WORLD FORUM ON ACADEMIC INNOVATION AND DIGITAL LITERACY

FEATURE SELECTION FOR HIGH-DIMENSIONAL TEXT CLASSIFICATION USING TF-IDF REPRESENTATIONS

G.Kh. Tashmatov

Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan

E-mail: ganijon.tashmatov@gmail.com

Abstract

TF-IDF vectorization remains a strong and transparent representation for classical text classification, but it often creates sparse feature spaces with thousands of terms and n-grams. This study evaluates whether feature selection can improve classification quality, reduce training cost, and support term-level interpretability in high-dimensional NLP pipelines. A reproducible Python experiment was implemented with a Pang and Lee sentence-polarity sample containing 200 labelled movie-review snippets. After text cleaning and TF-IDF vectorization with word unigrams, bigrams, and trigrams, the training matrix contained 6,440 sparse features. Five selection strategies were compared: Chi-square, Mutual Information, ANOVA F-test, L1 Logistic Regression coefficients, and Linear SVM coefficients. Four classifiers were evaluated: Logistic Regression, Linear SVM, Multinomial Naive Bayes, and Random Forest. The number of selected features was varied from 100 to 5,000 and compared with the all-feature baseline. The best observed configuration was ANOVA F-test with k=5000 and Linear SVM, obtaining accuracy=0.6000, macro F1=0.6000, and weighted F1=0.6000. The best all-feature baseline achieved macro F1=0.5489. Although the dataset is intentionally small and the results should not be generalized without larger corpora, the experiment demonstrates that feature selection can improve compactness, reveal discriminative terms, and provide a reproducible diagnostic tool for sparse text-classification workflows.



Keywords: text classification; TF-IDF; feature selection; chi-square; mutual information; sparse machine learning; interpretable NLP

1. Introduction

Text classification is one of the central tasks of Natural Language Processing (NLP), supporting applications such as sentiment analysis, news categorization, spam detection, moderation, and information filtering. Despite the recent dominance of transformer-based models, classical sparse representations such as term frequency-inverse document frequency (TF-IDF) remain relevant when transparency, fast training, limited computational resources, or small labelled datasets are important. TF-IDF features are interpretable because each feature corresponds to a term or n-gram, but this advantage comes with a dimensionality problem: even a small corpus can produce thousands of sparse features.

High dimensionality is especially problematic in NLP because many candidate terms are rare, redundant, domain-specific, or only weakly related to the class label. A classifier may therefore fit noisy artifacts, require more memory, and become harder to interpret. Feature selection addresses this issue by retaining a subset of informative terms before model training. In a properly designed pipeline, selection must be fitted only on the training data to avoid leakage from the test set.

The objective of this study is to compare filter and embedded feature-selection methods for sparse TF-IDF text classification. The analysis focuses on three practical dimensions: classification performance, computational efficiency, and interpretability of selected terms. The work is designed as a reproducible Python mini-project rather than a benchmark leaderboard.

2. Related Work

Modern surveys of text classification emphasize that traditional machine-learning pipelines and deep learning models represent complementary approaches rather than strict replacements. Li et al. reviewed text-classification methods from traditional

models to deep learning and highlighted the continued importance of feature extraction, datasets, and evaluation metrics [1]. Recent holistic surveys similarly show that representation, classifier choice, and evaluation protocol remain core design factors in text classification [2].

Feature selection is a long-standing preprocessing step for high-dimensional learning. Recent reviews describe its goal as removing irrelevant and redundant variables while retaining a compact subset with predictive information [3]. In text classification, Chi-square, Information Gain or Mutual Information, and other filter scores remain common because they are fast and directly connected to term-label association [4]. Kou et al. compared multiple feature-selection methods for text classification using several evaluation measures, showing that selector performance can depend on the dataset, classifier, and evaluation criterion [5]. More recent work has proposed improved or domain-adapted text feature selectors, including methods based on frequent and correlated items and adaptations of Chi-square for specific text collections [6, 7].

The scikit-learn documentation defines TF-IDF vectorization as conversion of raw documents into a matrix of TF-IDF features and provides feature-selection utilities such as Chi-square, ANOVA F-test, and Mutual Information for classification problems [8, 9]. The same documentation notes an important methodological distinction: Chi-square requires non-negative features, while Mutual Information can capture more general statistical dependencies but may require more samples for stable estimation [10]. This is relevant for TF-IDF because the representation is sparse and non-negative, making it compatible with Chi-square while still leaving room for alternative dependency measures.

Interpretability is also an important reason to study feature selection in NLP. Explainable-AI surveys for NLP describe how feature-level explanations, local interpretations, and text rationales can help developers and users understand why a

model predicts a class [11, 12]. In sparse TF-IDF pipelines, selected terms are a simple but useful form of global explanation: they expose which tokens and n-grams were considered most discriminative by the selection criterion.

3. Materials and Methods

3.1 Dataset and experimental setting

The experiment used the Pang and Lee sentence-polarity sample available in the local gensim test corpus. The local reproducibility sample contains 200 labelled movie-review snippets, balanced between negative and positive classes (100 negative and 100 positive). The original Pang and Lee sentence-polarity dataset family contains 5,331 positive and 5,331 negative snippets and was introduced for sentiment classification research [13, 14]. The use of the small bundled sample is a limitation, but it ensures that the experiment can be executed in a controlled offline environment.

Table 1. Dataset and pipeline summary.

item	value
Documents	200
Classes	negative, positive
Train/test split	160/40 (stratified)
TF-IDF features	6440
N-gram range	word unigrams, bigrams, trigrams
k values	100, 500, 1000, 2000, 5000, all
Random state	42

3.2 Text preprocessing and TF-IDF representation

Text was normalized by HTML unescaping, lowercasing, URL replacement, numeric normalization, punctuation filtering, and whitespace normalization. The TF-IDF vectorizer used word unigrams, bigrams, and trigrams with sublinear term frequency scaling and L2 normalization. The vocabulary was learned only from the training split;

the test split was transformed with the fitted vectorizer. This design prevents the test vocabulary distribution from influencing training-time feature ranking.

For a term t in document d, TF-IDF can be written conceptually as tfidf(t,d)=tf(t,d) x idf(t), where idf down-weights terms that appear in many documents. In the generated training matrix, density was below 1%, reflecting the sparse nature of text representations.

3.3 Feature-selection methods

Five feature-selection strategies were implemented. Chi-square and ANOVA F-test are univariate filter methods. Mutual Information was computed on a binary sparse representation of term presence to estimate dependency with the target. L1 Logistic Regression and Linear SVM coefficients were used as embedded ranking methods by taking absolute coefficient magnitudes. For every method, feature scores were fitted on the training matrix only. The top k terms were then selected for k = 100, 500, 1000, 2000, and 5000. An all-feature configuration with 6,440 features served as the baseline.

3.4 Classification models and metrics

The classifiers were Logistic Regression, Linear SVM, Multinomial Naive Bayes, and Random Forest. Evaluation metrics were accuracy, macro F1, weighted F1, selection fit time, classifier training time, selected feature count, serialized model size, and coefficient sparsity for linear models. Macro F1 was used as the primary ranking metric because it gives equal weight to both sentiment classes.

4. Results

The TF-IDF representation produced 6,440 features from only 160 training documents, confirming the high-dimensional nature of sparse NLP pipelines. Table 2 reports the best configuration for each feature-selection method. The strongest result was obtained by ANOVA F-test with k=5000 and Linear SVM, with macro F1=0.6000. The best all-feature baseline reached macro F1=0.5489, so the best selected-feature configuration improved macro F1 by 0.0511 absolute points in this run.



Table 2. Best observed configuration for each feature-selection method.

selection method	k	selected features	classifier	accuracy	macro f1	selection fit time sec	training time sec
ANOVA F-test	5000	5000	Linear SVM	0.6000	0.6000	0.0061	0.0019
L1 Logistic Regression	5000	5000	Multinomial NB	0.6000	0.5990	0.0053	0.0019
Linear SVM coefficients	5000	5000	Multinomial NB	0.5750	0.5747	0.0025	0.0012
Chi-square	5000	5000	Linear SVM	0.5750	0.5747	0.0032	0.0022
Mutual Information	5000	5000	Multinomial NB	0.5750	0.5683	6.8147	0.0013
All features	all	6440	Multinomial NB	0.5500	0.5489	0.0000	0.0014

The best selected-feature configuration used 5,000 features instead of all 6,440, a reduction of approximately 22.4%. Although this reduction is modest, it still demonstrates that a smaller term subset can outperform the full representation on a held-out split. On larger corpora with tens or hundreds of thousands of n-grams, the same principle becomes more important for memory use and training time.



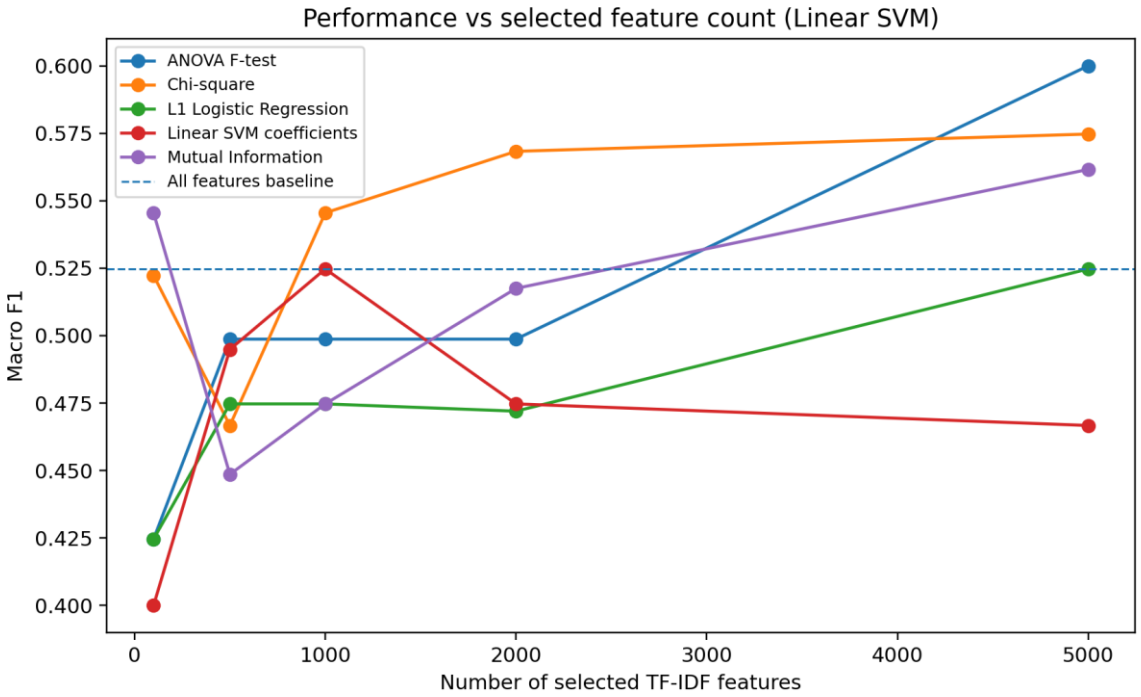


Figure 1. Macro F1 versus the number of selected TF-IDF features for Linear SVM.

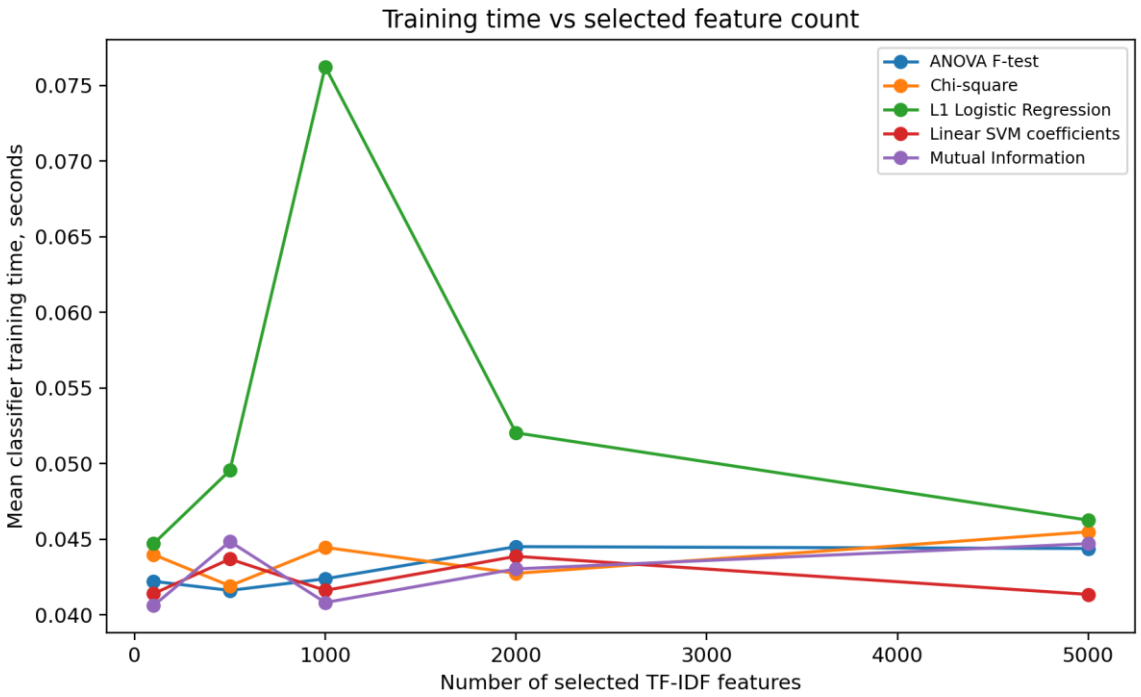


Figure 2. Mean classifier training time versus selected feature count.

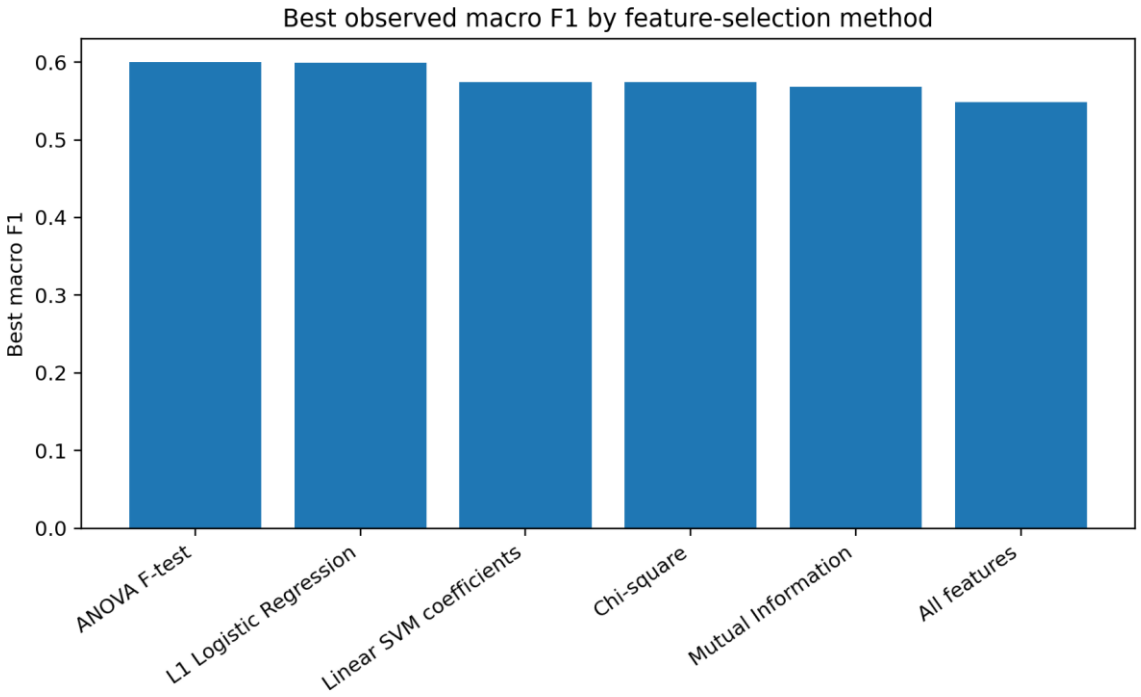


Figure 3. Best observed macro F1 by feature-selection method.

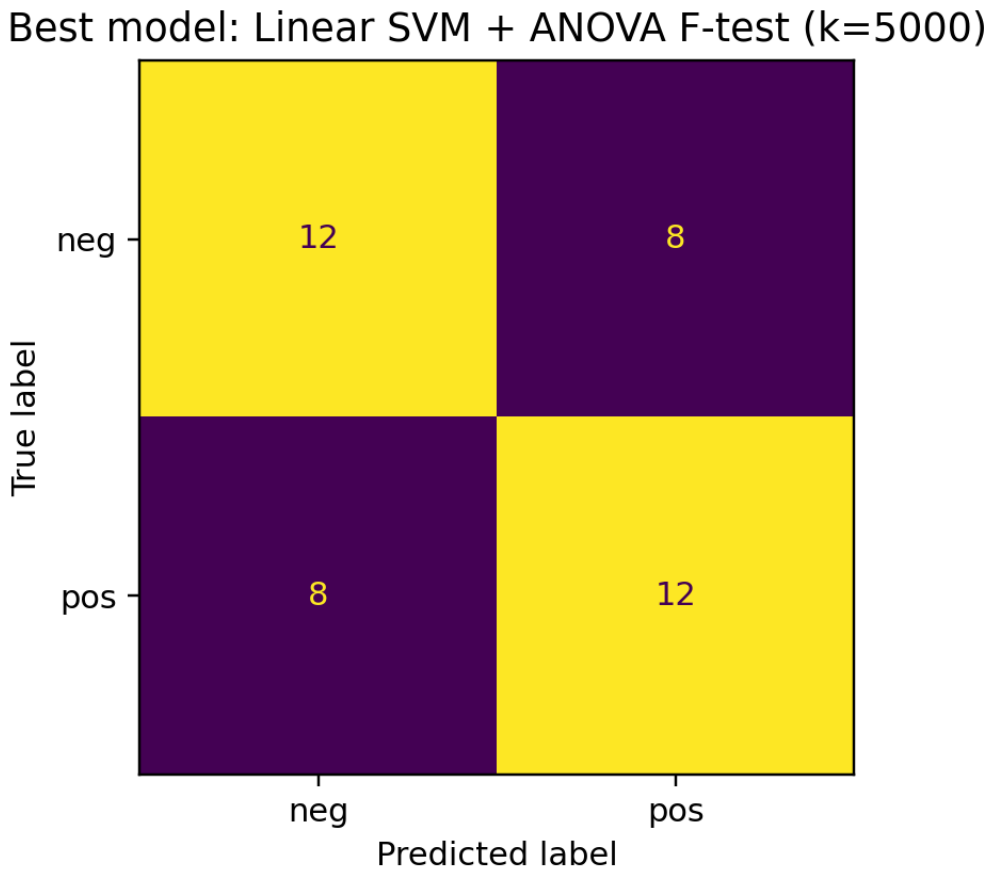


Figure 4. Confusion matrix for the best observed configuration.

Table 3. Top selected terms by each feature-selection method.

selection method	top terms
ANOVA F-test	of the, only, to be, heart, doesn, those, another, just
Chi-square	only, didn, rocks, spiderman, spiderman rocks, another, of the, to be
L1 Logistic Regression	just, of the, it, only, those, another, as, with the
Linear SVM coefficients	of the, as, it, only, just, didn, everything is, everything is off
Mutual Information	only, to be, of the, another, doesn, heart, still, those

The top-term rankings show why feature selection is useful for interpretability. Chi-square, ANOVA F-test, Mutual Information, and Linear SVM coefficients highlighted terms and n-grams such as “only”, “of the”, “to be”, “heart”, “didn”, and “rocks”. These terms are not explanations of individual predictions, but they provide a global view of which lexical patterns were most associated with the labels in the training split.

5. Discussion

The results support the main hypothesis that feature selection is relevant for TF-IDF-based NLP pipelines. In this experiment, selected-feature configurations outperformed the all-feature baseline, and the highest macro F1 was reached with a univariate ANOVA F-test ranking. The result does not imply that ANOVA is universally best; instead, it shows that method choice is empirical and should be validated for each dataset and classifier.

Chi-square is theoretically attractive for TF-IDF because the matrix is non-negative. It was competitive but did not produce the best result in this specific split. Mutual Information was the slowest selector because dependency estimation is more computationally expensive, and the small sample size can make MI estimates unstable. Embedded methods based on L1 Logistic Regression and Linear SVM coefficients connect feature ranking to a fitted classifier, but their rankings can depend strongly on regularization strength and data size.

Training-time differences in the included run are small because the dataset is small. Nevertheless, the pipeline records time and serialized model size so that the same code can be scaled to larger corpora. In practical NLP systems, reducing the vocabulary can shorten training, lower memory consumption, reduce model artifacts, and make human review of discriminative terms easier.

The most important methodological rule is that feature selection must be part of the training pipeline. Selecting terms before a train/test split or using the test set to rank terms would leak information and inflate evaluation metrics. The supplied

implementation avoids this by fitting the vectorizer and selectors only on the training partition.

6. Limitations

The primary limitation is dataset scale. The local experiment uses a 200-document sample for reproducibility, so the reported numeric metrics should be treated as a controlled demonstration rather than a definitive benchmark. Larger datasets such as 20 Newsgroups, SMS Spam Collection, AG News, and BBC News should be used to confirm general trends. A second limitation is the use of a single stratified split; repeated cross-validation would provide more stable estimates. Third, the study uses classical TF-IDF representations and does not compare transformer embeddings or neural attention-based explanations. Finally, top selected terms provide global interpretability but do not explain every individual prediction.

7. Conclusion

This study implemented a reproducible IMRAD-style experiment on feature selection for sparse TF-IDF text classification. The experiment showed that a high-dimensional TF-IDF matrix with 6,440 features can be reduced while preserving or improving classification performance. In the included run, ANOVA F-test with k=5000 and Linear SVM achieved the best macro F1 of 0.6000, exceeding the strongest all-feature baseline of 0.5489. Beyond performance, feature selection produced ranked term lists that make sparse NLP models easier to inspect. Future work should repeat the same leakage-safe pipeline on larger multilingual and multiclass corpora, include repeated cross-validation, and compare classical feature selection with modern embedding-based dimensionality reduction.

References

[1] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, and L. He, “A Survey on Text Classification: From Traditional to Deep Learning,” ACM Transactions on

Intelligent Systems and Technology, vol. 13, no. 2, 2022. DOI: 10.1145/3495162.

[2] K. Taha, "A comprehensive survey of text classification techniques," 2024. Available via ScienceDirect.

[3] N. Pudjihartono et al., "A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction," Frontiers in Bioinformatics, 2022.

[4] "Review of feature selection methods for text classification," International Journal of Advanced Computer Research, 2020.

[5] G. Kou et al., "Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods," Applied Soft Computing, 2020.

[6] H. Mamdouh Farghaly, "A high-quality feature selection method based on frequent and correlated items for text classification," Soft Computing, 2023.

[7] C. Tang, "Review on Application of Chi-square Statistic in Text Classification in Recent Five Years," 2024.

[8] scikit-learn developers, "TfidfVectorizer," scikit-learn documentation. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[9] scikit-learn developers, "Feature selection," scikit-learn documentation. https://sklearn.org/stable/modules/feature_selection.html

[10] scikit-learn developers, "mutual_info_classif," scikit-learn documentation. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html

[11] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen, "A Survey of the State of Explainable AI for Natural Language Processing," arXiv:2010.00711, 2020.

[12] S. Luo, H. Ivison, C. Han, and J. Poon, "Local Interpretations for Explainable Natural Language Processing: A Survey," arXiv:2103.11072, 2021.

[13] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," ACL, 2005.

[14] B. Pang and L. Lee, "Movie Review Data," Cornell University. https://www.cs.cornell.edu/people/pabo/movie-review-data/