Comparing the accuracy of text classification using multiple stemming and machine learning method

1st Mithila Arman
Department of Computer Science And Engineering
Brac University
mithila.arman@g.bracu.ac.bd

2nd Md Humaion Kabir Mehedi
Department of Computer Science And Engineering
Brac University
humaion.kabir.mehedi@g.bracu.ac.bd

3rd Mehnaz Ara Fazal
Department of Computer Science And Engineering
Brac University
mehnaz.ara.fazal@g.bracu.ac.bd

4th Annajiat Alim Rasel
Department of Computer Science And Engineering
Brac University
annajiat@bracu.ac.bd

***Abstract*—Natural Language Processing (NLP) plays a pivotal role in text-based data analysis and classification. This research explores the application of NLP techniques in conjunction with machine learning algorithms for document classification. The study utilizes the 20 Newsgroups dataset and focuses on preprocessing text data through tokenization, part-of-speech tagging, stemming, and stopword removal. The incorporation of TF-IDF features, both unigrams and bigrams, is employed for feature extraction. Additionally, Principal Component Analysis (PCA) is utilized to reduce feature dimensionality. The research compares the performance of two classifiers, namely Naive Bayes and Support Vector Machines, in classifying the preprocessed and feature-engineered text data. Experimental results demonstrate the effectiveness of the proposed approach in achieving high classification accuracy. The findings provide valuable insights into the synergy of NLP and machine learning for text classification tasks.**

***Index Terms*—Text Classification, Natural Language Processing (NLP), Term Frequency-Inverse Document Frequency (TF-IDF), 20 Newsgroups Dataset, Stemming, Stopword Removal, Support Vector Machines (SVM), Naive Bayes Classifier, Feature Extraction, PCA, n-gram**

I. Introduction

In the era of information overload, effective text classification has become a critical component of numerous applications, ranging from information retrieval to sentiment analysis. Natural Language Processing (NLP) techniques, coupled with machine learning algorithms, have emerged as powerful tools for extracting meaningful insights from vast textual datasets. This research focuses on the application of NLP and machine learning in the context of document classification, leveraging the widely-used 20 Newsgroups dataset.

The primary objective of this study is to investigate the impact of various preprocessing techniques on text data, including tokenization, part-of-speech tagging, stemming, and stopword removal. These techniques aim to transform raw textual information into a structured format conducive to subsequent machine learning analysis. The research also explores the use of TF-IDF (Term Frequency-Inverse Document Frequency) features, incorporating both unigrams and bigrams, to represent the textual content of documents.

Furthermore, dimensionality reduction is addressed through Principal Component Analysis (PCA), providing a means to distill essential features and enhance the efficiency of the subsequent classification algorithms. The classification task is approached using two distinct algorithms: Naive Bayes and Support Vector Machines (SVM). These classifiers are trained and evaluated on the preprocessed and feature-engineered text data, allowing for a comparative analysis of their performance.

This investigation aims to contribute insights into the interplay between NLP techniques and machine learning algorithms, shedding light on their effectiveness in document classification tasks. The outcomes of this study have implications for diverse applications, including information retrieval, content categorization, and automated document organization.

The subsequent sections of this paper delve into the methodology employed, detailing the preprocessing steps, feature extraction, and the chosen classifiers. Experimental results and their implications are then discussed, followed by a comprehensive analysis in the conclusion, providing a holistic perspective on the presented approach.

II. Literature review

In the realm of text classification, the significance of Python programming and the Natural Language Toolkit (NLTK) for preprocessing activities, including case folding, tokenizing, and stop word removal, has been emphasized [1]. The study specifically employed stemming methods in the context of non-formal Indonesian conversation, utilizing the CACM collection for evaluation [2]. Stemming algorithms such as Paice-Husk, Porter's, and Lovin's, along with lemmatization, were explored to enhance retrieval performance, with Mean Average Precision (MAP) used for evaluation.

The literature review extends to Turkish text classification, revealing the prevalence of Support Vector Machines (SVM)

and LIWC in marketing research studies, suggesting the need for alternatives like Naive Bayes (NB) and Random Forest (RF) [3] [4]. The paper also introduces statistical stemmers, such as the N-Gram Stemmer and the HMM Stemmer, and briefly touches on inflectional and derivational methods in stemming [5].

Studies by Leopold Kinermann, Lin et al., Nguyen Luong, and Pham Ta demonstrate the application of machine learning algorithms, including SVM, in various text classification tasks, showcasing their efficiency in different languages [6]. Stemming, as discussed, finds applications in text classification, clustering, sentiment analysis, and various other tasks [7].

The paper highlights the advantages of Natural Language Processing (NLP), such as automatic summary generation and co-reference resolution, while acknowledging its disadvantages, including the loss of visual context and challenges with synonyms [8]. The importance of deep linguistic understanding in lemmatization for accurate outcomes is emphasized, contrasting with stemming, which is described as a process that simply chops off word endings without considering meaningful information.

Furthermore, the literature encompasses the exploration of phonetic algorithms for conflation in stemming algorithms [9]. Relevant papers by Krovetz, Lovins, Mayfield and McNamee, Majumder et al, Singh, Tamah, UzZaman and Khan, and Xu and Croft are mentioned, providing insights into morphology, stemming algorithm development, and phonetic support.

The comparison of rule-based approaches, such as YASS and GRAS, and their evaluation for different languages, including English, French, Bengali, and Marathi, is detailed, with a focus on stemmer strength and computation time [10].

III. Dataset

The research employs the widely recognized 20 Newsgroups dataset, a collection of approximately 20,000 documents spanning 20 different newsgroups. Each document is associated with a specific category, making it an ideal benchmark for text classification tasks. The chosen subset, encompassing all categories and excluding headers, footers, and quotes, forms the basis of our investigation.

IV. Methodology

A. Data Preprocessing

Tokenization and Part-of-Speech Tagging
The preprocessing pipeline initiates with the application of tokenization to break down each document into individual words. Part-of-speech tagging is then performed using the Natural Language Toolkit (NLTK) to assign grammatical categories to each token.

Stemming
The Porter Stemmer, Lancaster Stemmer and Snowball Stemmer is employed to reduce words to their root form, ensuring a uniform representation of semantically similar terms. This step aims to enhance the efficiency of subsequent feature extraction.

Stopword Removal
Stopwords, common words that contribute little to the overall meaning, are removed to reduce noise in the dataset. This step is crucial in refining the dataset for meaningful feature extraction.

B. Feature Extraction

TF-IDF Vectorization

The preprocessed text data is transformed into TF-IDF features using the sklearn TfidfVectorizer. n-grams is considered, capturing the frequency of terms and their significance across documents.

Principal Component Analysis (PCA)

To address the curse of dimensionality, PCA is applied to reduce the feature space. The number of components is determined to strike a balance between preserving information and computational efficiency.

C. Training and Evaluation

Splitting

The dataset is divided into training and testing sets using a standard 80-20 split. This ensures the model is trained on a representative portion of the data and evaluated on unseen instances.

Naive Bayes Classifier

A Multinomial Naive Bayes classifier is trained on the TF-IDF features of the training set. The model is evaluated on the testing set, and performance metrics, including accuracy and classification report, are recorded.

Support Vector Machines (SVM) Classifier
A Support Vector Machines classifier is trained and evaluated in a manner similar to the Naive Bayes approach. SVM is chosen for its ability to handle high-dimensional data efficiently.

V. Result and Analysis

A. Result Summary

In this study, the performance of Naive Bayes and Support Vector Machines (SVM) algorithms was evaluated using different text processing techniques, including Porter, Lancaster, and Snowball stemming, both before and after applying part-of-speech (POS) tagging, principal component analysis (PCA), and n-gram features. Prior to feature augmentation, Naive Bayes exhibited accuracy rates of 63.37%, 59.34%, and 63.82% for Porter, Lancaster, and Snowball, respectively, while SVM achieved accuracies of 60.85%, 51.91%, and 61.72% for the corresponding techniques. Following the application of POS tagging, PCA, and n-gram features, Naive Bayes demonstrated improved accuracy for Porter (63.71%) and Snowball (64.22%) but exhibited a significant decrease for Lancaster (13.58%). Similarly, SVM showed enhanced

accuracy for Porter (62.68%) and Snowball (63.55%) but a substantial reduction for Lancaster (5.12%).

B. Result Table and Comparison

TABLE I
THE ACCURACY LEVEL OF CLASSIFICATION MODEL BY USING "PORTER", "LANCASTER" AND "SNOWBALL" STEMMERS.

*before applying (POS) tagging, PCA, n-gram

Classification Algorithm	Stemmers		
	Porter	***Lancaster***	***Snowball***
Naive Bayes	0.6337	0.5934	0.6382
Support Vector Machines	0.6085	0.5191	0.6172

Fig.1 Comparing the results

TABLE II
THE ACCURACY LEVEL OF CLASSIFICATION MODEL BY USING "PORTER" AND "LANCASTER" STEMMERS

*after applying (POS) tagging, PCA, n-gram

Classification Algorithm	Stemmers		
	Porter	***Lancaster***	***Snowball***
Naive Bayes	0.6371	0.1358	0.6422
Support Vector Machines	0.6268	0.0512	0.6355

Fig.2 Comparing the results

VI. CONCLUSION

Our study on document classification using NLP techniques and machine learning algorithms with the 20 Newsgroups dataset highlights the pivotal role of our preprocessing pipeline. Tokenization, part-of-speech tagging, stemming, and stopword removal contribute to a refined format, influencing feature extraction and classification stages.In feature extraction, TF-IDF vectorization with n-grams captures document essence, while PCA efficiently addresses dimensionality challenges.

Exploring different stemming techniques such as Porter, Lancaster, and Snowball reveals, Snowball's consistent superiority, emphasizing its effectiveness. The decrease in accuracy with the Lancaster Stemmer underscores the importance of careful preprocessing decisions. Comparing Naive Bayes and SVM classifiers, Snowball Stemmer stands out with 64.22% and 63.55% accuracy, respectively, showcasing the methodology's efficacy and versatility.

Our research offers a comprehensive methodology, emphasizing reproducibility through Google Colab and Google Drive. The insights encourage further exploration of diverse preprocessing techniques, classifier evaluation, and the method's applicability to various text corpora.

VII. FUTURE WORK

To enhance the dimensionality reduction process, future work could consider alternative techniques, like t-Distributed Stochastic Neighbor Embedding (t-SNE) or manifold learning methods, which may better preserve local and global structures in high-dimensional data. The study's focus on individual classifiers, Naive Bayes and Support Vector Machines, opens the possibility for future exploration of ensemble methods or hybrid models that combine multiple classifiers to leverage their strengths and improve overall classification accuracy.

Expanding the evaluation scope to diverse datasets representing various domains, languages, or text styles will contribute to assessing the generalizability of the proposed approach. Further, optimizing hyperparameters, employing interpretability and explainability techniques for classifier decisions, and addressing scalability and efficiency concerns in the face of growing textual data volumes are important avenues for future investigation.

*ACKNOWLEDGEMENT

I extend my heartfelt gratitude to my academic advisors for their unwavering support and guidance throughout this research project. Special appreciation goes to the broader academic community for fostering an environment conducive to intellectual exploration. The open-source contributions of the Natural Language Toolkit (NLTK) and scikit-learn developers significantly facilitated the implementation of NLP techniques and machine learning algorithms. Acknowledgment is extended to authors referenced in the literature review for laying the groundwork in text classification and stemming algorithms. Access to datasets, particularly the 20 Newsgroups dataset, provided by institutions, greatly contributed to the study.

REFERENCES

[1] E. P. W. Rianto1*, Achmad Benny Mutiara2 and P. I. Santosa, "Improving the accuracy of text classifcation using stemming method, a case of non-formal indonesian conversation," *Springer*, p. 16, 2021.

[2] V. Balakrishnan and E. Lloyd-Yemoh, "Stemming and lemmatization: A comparison of retrieval performances," *semantic scholar*, vol. 2, no. 3, p. 6, 2014.

[3] . Yehia Ibrahim Alzoubi 1, Ahmet E. Topcu 2 and A. E. E. 3, "Machine learning-based text classification comparison: Turkish language context," *Applied Science, MDPI*, no. 2, p. 22, 2023.

[4] C. S. M. H. Jochen Hartmann∗, Juliana Huppertz, "Comparing automated text classification methods," *ELSEVIER*, October 2018.

[5] M. G. Jivani, "A comparative study of stemming algorithms," *ISSN*, vol. 2(6), p. 1938, 2011.

[6] L. T. M. Nguyena, "Text classification based on support vector machine," *Mathematics*, vol. 9, no. 2, p. 19, 2019.

[7] M. I. T. S. H. A. A. Abdul Jabbar1, Sajid Iqbal2, "Empirical evaluation and study of text stemming algorithms," *Springer Nature*, 2020.

[8] N. N. M. D. B. M. . B. I. o. T. . . . Divya Khyani1, Siddhartha B S2, "An interpretation of lemmatization and stemming in natural language processing," *ISSN*, vol. 22, January 2021.

[9] D. Sharma, "Stemming algorithms: A comparative study and their analysis," *International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868*, vol. 4, no. 3, p. 6, 2012.

[10] E. P. W. Rianto1*, Achmad Benny Mutiara2 and P. I. Santosa3, "Improving the accuracy of text classifcation using stemming method, a case of non-formal indonesian conversation," *springer*, p. 16, 2021.