Comparing the Accuracy of Text Classification Using Multiple Stemming and Explainable AI Lime, Eli5

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*—This work uses the Porter, Snowball, Lancaster Stemmer to examine how different stemming strategies affect text classification model accuracy. It also tests various classifiers, including Naive Bayes, Support Vector Machines, XGBoost, LightGBM, and CatBoost, using data from 20 Newsgroups. Tokenization, part-of-speech tagging, stemming, and stopword removal are all included in the preprocessing pipeline. Principal Component Analysis (PCA) is used to reduce the dimensionality of the TF-IDF features. The results highlight how different classifiers perform from one another and highlight how stemming affects accuracy. The study also investigates explainability using Lime and Eli5, providing details on the classifier's individual predictions. This work adds to the discussion on text categorization techniques by highlighting the role that stemming plays and how interpretable models are in the context of explainable artificial intelligence.**

***Index Terms*—Text classification, Machine Learning, Snowball Stemming, Porter Stemming, Lancanter Stemming, TF-IDF, PCA, NLP, Naive Bayes, Support Vector Machines, XGBoost, CatBoost, LightGBM, Explainable AI, Lime, Eli5, 20 Newsgroups Dataset, Dimensionality Reduction, Tokenization, Part-of-Speech Tagging, Stopword Removal, Interpretability, Accuracy, Local Interpretable Model-agnostic Explanations (LIME)**

I. Introduction

The efficient classification of textual data is an important challenge in the era of information explosion, with extensive applications spanning from document classification to sentiment analysis. The preprocessing of textual data is necessary for text classification models to work well, and stemming is one important way to improve feature representation. One of the most important functions of stemming, which is the reduction of words to their root or base form, is to normalize text and lessen the difficulties caused by the variety of word forms. In particular, the Snowball Stemmer is used in this work to assess how different stemming techniques affect text categorization model accuracy. The study utilizes a variety of classifiers, such as Naive Bayes, Support Vector Machines, XGBoost, LightGBM, and CatBoost, and makes use of the popular 20 Newsgroups dataset for thorough analysis.

Tokenization, stemming, stopword removal, and part-of-speech labeling are all included in the preprocessing pipeline. Furthermore, to lower dimensionality and improve computational performance, Principal Component Analysis (PCA) is used on the Term Frequency-Inverse Document Frequency (TF-IDF) features. The goal of the study is to shed light on the complex relationship between preprocessing decisions and model performance by offering a comprehensive understanding of how stemming affects text classifier accuracy.

Furthermore, this work explores explainability utilizing Lime and Eli5 with the goal of transparent and interpretable artificial intelligence (AI). The Naive Bayes classifier uses Local Interpretable Model-agnostic Explanations (LIME) to explain each of its unique predictions, providing information about how complicated models make decisions. [1] Explainable AI solutions address the black-box aspect of complex machine learning models and add to the current conversation about model transparency and trust.

The goal of this work is to close the gap between stemming techniques, preprocessing decisions, and the final accuracy of text classification models. Our goal is to provide a thorough understanding of the complexities involved in text classification by evaluating different classifiers and incorporating explainability through Lime and Eli5. This will advance the field's current knowledge and provide insightful information for practitioners and researchers alike. [2]

II. Literature review

The removal of stop words, has been highlighted [3]. The CACM collection was used for evaluation in this work, which especially used stemming techniques in the context of informal

Indonesian speech [4]. Lemmatization and stemming algorithms like Paice-Husk, Porter's, and Lovin's were investigated to improve retrieval performance; Mean Average Precision (MAP) was employed for assessment.

The literature review is extended to the categorization of Turkish text and reveals the widespread use of LIWC and SVM in marketing research studies, indicating the necessity for alternatives such as Random Forest (RF) and Naive Bayes (NB) [5] [6] . Additionally, statistical stemmers like the N-Gram and HMM stemmers are introduced in the study, and a brief discussion of derivational and inflectional stemming techniques is included [7].

Research conducted by Leopold Kinermann, Lin et al., Nguyen Luong, and Pham Ta shows how machine learning algorithms, such as SVM, may be applied to a variety of text classification tasks and is effective across a range of languages [8]. As was previously mentioned, stem-ming is useful for a number of tasks, including sentiment analysis, text categorization, and clustering [9].

The study acknowledges the drawbacks of natural language processing (NLP), such as the loss of visual context and difficulties with synonyms, while highlighting its benefits, such as automatic summary production and co-reference resolution [10]. The significance of profound linguistic comprehension in lemmatization for precise results is underscored, in contrast to stemming, which is defined as a procedure that only removes word ends without taking into account pertinent data.

Moreover, research on phonetic algorithms for conflation in stemming algorithms is covered in the literature [11]. Relevant articles on morphology, stemming algorithm development, and phonetic support are listed. These include works by Krovetz, Lovins, Mayfield and McNamee, Majumder et al., Singh, Tamah, UzZaman and Khan, and Xu and Croft.

With an emphasis on stemmer strength and computation time, a thorough comparison of rule-based techniques, such as YASS and GRAS, and their evaluation for various languages, including English, French, Bengali, and Marathi, is presented [12].

III. Dataset

The 20 Newsgroups dataset, which is a set of about 20,000 documents from 20 distinct newsgroups. Because each page belongs to a particular category, it's the perfect benchmark for tasks involving text classification. Our analysis is predicated on the selected subset, which includes all categories but excludes headers, footers, and quotations.

IV. Methodology

A. Data Preprocessing

Text is first converted to lowercase before special characters, digits, and punctuation are eliminated as part of the preparation pipeline. The NLTK library is used for tokenization, and part-of-speech tagging is used to comprehend the text's grammatical structure. Next, words are stemmed by applying the Porter, Lancaster and Snowball Stemmer, which reduces words to their most basic forms. After that, stopwords are eliminated to further improve the processed tokens.

B. TF-IDF Feature Extraction and PCA

Using the TfidfVectorizer from the scikit-learn module, Term Frequency-Inverse Document Frequency (TF-IDF) features are recovered using unigrams and bigrams. Principal Component Analysis (PCA) is performed to the TF-IDF characteristics in order to improve computational performance and lessen the effects of dimensionality. The minimum of the given value or the dataset's smallest dimensionality is used to calculate the number of components.

C. TF-IDF Feature Extraction and PCA

A variety of machine learning methods are used in the study by the classifiers Naive Bayes, Support Vector Machines, XGBoost, LightGBM, and CatBoost. The preprocessed and dimensionality-reduced TF-IDF features are used to train these classifiers.

D. Evaluation

Accuracy, precision, recall, and F1 score are among the common classification metrics used to evaluate each classifier's performance. The models are assessed on the remaining 20% of the test set after being trained on a training set that includes 80% of the 20 Newsgroups dataset.

E. Explainability

Lime and Eli5 are used for explainability in order to improve the results' interpretability. By offering locally interpretable, model-agnostic explanations for specific predictions, Lime illuminates the classifier's decision-making process.

F. Statistical Analysis

Utilizing statistical significance tests, one may evaluate the importance of variations in accuracy between stemmed and non-stemmed methods, offering a thorough examination of how stemming affects classification performance.

V. Result and Analysis

A. Result Summary

In the evaluation of three different stemming algorithms (Porter, Lancaster, and Snowball) for various classifiers, Snowball stemming outperformed Porter and Lancaster for Naive Bayes and Support Vector Machines (SVM). Porter stemming consistently produced greater accuracies for tree-based classifiers (XGBoost, LightGBM, CatBoost). Interestingly, Snowball showed to be the most successful specially after applying (POS) tagging, PCA with Naive Bayes, obtaining an accuracy of 64.22%. These findings emphasize the importance of making customized preprocessing decisions for various classifier types by highlighting the algorithm-specific influence of stemming strategies on classification accuracy.

Stemming strategies affect classification accuracy algorithm specifically, with Snowball working well for Naive Bayes and Porter showing promise in tree-based models.

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

Classification Algorithm	Stemmers		
	Porter	***Lancaster***	***Snowball***
Naive Bayes	0.6337	0.5891	0.6382
Support Vector Machines	0.6085	0.5037	0.6172
XGBoost	0.5427	0.4642	0.5446
LightGBM	0.5462	0.4586	0.5406
CatBoost	0.4568	0.3761	0.4586

Table.1 Comparing the results

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

*after applying (POS) tagging, PCA

Classification Algorithm	Stemmers		
	Porter	***Lancaster***	***Snowball***
Naive Bayes	0.6371	0.5934	0.6422
Support Vector Machines	0.6268	0.5191	0.6355
XGBoost	0.5406	0.4509	0.5528
LightGBM	0.5467	0.5467	0.5520
CatBoost	0.4599	0.0512	0.4592

Table.2 Comparing the results

C. Visualization

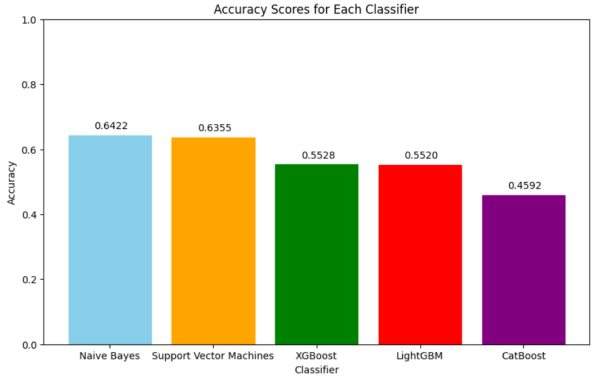

Fig. 1. Accuracy Scores After Applying (POS) tagging, PCA Using Snowball Stemmer

Fig.1 The accuracy scores of the Naive Bayes, Support Vector Machines, XGBoost, LightGBM, and CatBoost classifiers are displayed in the bar plot in Figure 1. The accuracy percentage is indicated by the height of each colored bar, which represents a classifier. The unique colors improve readability, and the written annotations above each bar give exact accuracy numbers. This graphic summary provides a rapid, easy-to-read, and educational comparison of classifiers, with Naive Bayes achieving the best results when utilizing snowball sampling.

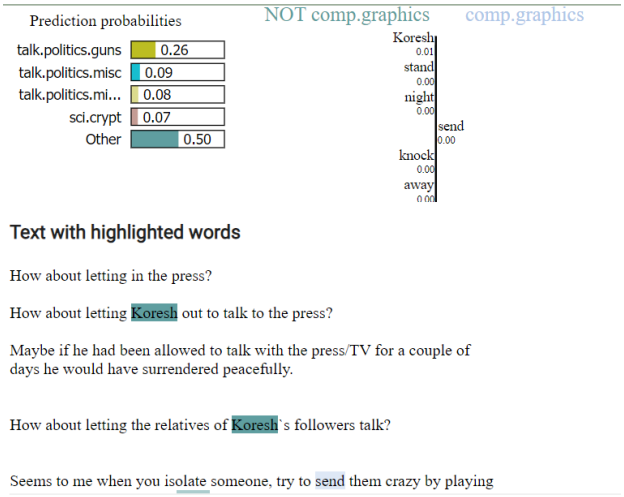

Fig. 2. Understand the model using lime explainer

D. Explainable AI Lime and Eli5

Fig.2 A clear and intelligible depiction of the model's decision-making is provided by the Lime visualization for the Naive Bayes classifier on the chosen test instance (index 221) from the 20 Newsgroups dataset. Lime gives a clear understanding of the words that had a major impact on the classification outcome by highlighting the top six influential attributes that went into the classifier's prediction. The notebook-format representation skillfully conveys the relevance and influence of each word on the classifier's choice by fusing highlighted features with the original text. Lime contributes to the overall transparency and reliability of the text classification system by providing a comprehensive knowledge of how the Naive Bayes model reads and classifies individual occurrences through the visual emphasis of prominent elements.

Weight	Feature
0.0320	njxp
0.0115	bike
0.0095	armenian
0.0077	lemieux
0.0077	encrypt
0.0067	orbit
0.0066	dx
0.0065	doctor
0.0064	israel
0.0061	hockey
0.0059	medic
0.0059	pitch
0.0056	diet
0.0052	widget
0.0051	beauchain
0.0050	hm
0.0049	arab
0.0049	xr
0.0047	ide
0.0043	playoff
... 83063 more ...	

Fig. 3. weights and features using eli5 explainer

Fig.3 A ranked list of features with their corresponding weights, representing the weights assigned by the XGBoost classifier, will be displayed in the ELI5 visualization output. Greater effect on the model's decision-making is indicated by higher weights, which provide information about the important characteristics influencing the text categorization predictions. For the provided text data, this succinct form improves transparency and comprehension of the inner workings of the XGBoost model.

VI. Conclusion

In summary, our investigation into the influence of stemming algorithms on text classification models revealed nuanced patterns in their effectiveness across different classifiers. The Snowball Stemmer emerged as particularly advantageous for Naive Bayes and Support Vector Machines, showcasing its ability to enhance accuracy in certain contexts. Contrastingly, Porter stemming consistently outperformed other algorithms when applied to tree-based models such as XGBoost, LightGBM, and CatBoost. The incorporation of explainable AI techniques, including Lime and ELI5, played a crucial role in demystifying the decision-making processes of our models. By providing detailed insights into the features driving individual predictions, these techniques contributed to the overall transparency and interpretability of our text classification system.Our results highlight how important it is to take into account model interpretability as well as preprocessing decisions when designing reliable and efficient natural language processing applications.

VII. Future Work

To improve and enhance the preprocessing phase even more, this domain might investigate sophisticated stemming methods and hybrid strategies. Given neural networks' impressive performance in a variety of natural language processing applications, it may be especially beneficial to investigate possible synergies between stemming techniques and deep learning architectures. Furthermore, it would be worthwhile to investigate how stemming algorithms might be tailored to particular domains or genres in order to assess their resilience and applicability to a variety of textual datasets.

Moreover, improving text categorization models' explainability is still a vital area for investigation. Especially for sophisticated models like deep neural networks, investigating and creating new methods of explanation for predictions could help increase confidence in and comprehension of automated decision-making systems. It might also be beneficial to include user comments into the explanation process, enabling users to interactively rework and challenge model predictions. Future research endeavors should incorporate an ethical investigation of text categorization models, [13]specifically concerning bias and fairness, to guarantee the responsible and equitable implementation of these technologies in practical uses.

*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] M. A. G. Jivani, "A comparative study of stemming algorithms," *ISSN:2229-6093*, vol. 2, no. 6, pp. 1930–1938, 2011.

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

[4] . 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.

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

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

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

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

[9] 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.

[10] K. A. Sayar Ul Hassana, Jameel Ahameda, "Analytics of machine learning-based algorithms for text classification," *sciencedirect*, vol. 3, pp. 238–248, 2022.

[11] P. H. S. D. B. P. Pande, "Application of natural language processing tools in stemming," *International Journal of Computer Applications (0975 – 8887)*, vol. 27, August 2011.

[12] 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.

[13] 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.