Comparing the accuracy of text classifcation 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

The paper "Improving the accuracy of text classifcation using stemming method, a case of non-formal Indonesian conversation" emphasizes the importance of Python programming and the Natural Language Toolkit (NLTK) for natural language processing tasks, including preprocessing activities like case folding, tokenizing, and stop word removal [1].

The study used the Communications of the Association for Computing Machinery (CACM) collection, which contains 3204 documents, 64 queries, and relevance judgments. [2] The authors also mentioned various stemming algorithms, such as the PaiceHusk stemmer, Porter's stemmer, and Lovin's stemmer, which have been developed to reduce words to their

root forms. Lemmatization has been used in several languages for information retrieval and has shown to improve retrieval performance.Mean Average Precision (MAP) was used to evaluate document relevancy, which combines both recall-oriented and precision-oriented aspects of the search engine.

The paper focuses on Turkish text classification and reviews the available literature on this topic. [3]The paper analyzes the performance of various algorithms in classifying customer inquiries received by an institution. The findings of this study and the text classification technique utilized can be applied to data in dialects other than Turkish.

The paper highlights that many marketing research studies rely on SVM or LIWC without providing a rationale for their method choice, suggesting a lack of guidance from computer science research on text classification performance. The authors note that lexicon-based approaches, particularly LIWC, perform poorly compared to machine learning methods, with accuracies sometimes only slightly exceeding chance. They suggest that marketing research should consider alternatives such as NB and RF. [4]

The paper mentions two specific statistical stemmers: the N-Gram Stemmer and the HMM Stemmer. The N-Gram Stemmer uses a string-similarity approach and is language independent but requires significant memory and storage. The HMM Stemmer is based on the concept of Hidden Markov Models. [5] The paper also briefly mentions inflectional and derivational methods in stemming, which involve analyzing both inflectional and derivational morphology. The Krovetz Stemmer is mentioned as an example of a linguistic lexical stemmer.

Leopold Kinermann (2002) studied different weight schemes for text representation in input space using SVM . Lin et al. (2006) used SVM for question classification in Chinese . [6] Nguyen Luong (2006) achieved a classification accuracy of 80.72% for Vietnamese text classification using SVM. Pham Ta (2017) achieved a classification accuracy of 99.75% for Vietnamese text classification using a neural network method

Stemming has various applications such as text classification, text clustering, sentiment analysis, text compression, question answering, text summarization, machine translation, and detecting wicked websites. [7]The authors conducted experiments using Urdu language and concluded that existing evaluation metrics can only measure an average conflation of words and cannot perfectly measure stemmer performance for all languages.

This paper highlights the advantages of NLP, such as automatic summary generation, co-reference resolution, and improved efficiency in document retrieval.The paper also mentions the disadvantages of NLP, including the loss of visual context, difficulties in generalized searches, and challenges with synonyms. [8] It emphasizes the importance of deep linguistic understanding in lemmatization for accurate outcomes, while stemming simply chops off word endings without considering meaningful information.

Many machine learning algorithms have been applied to create automatic text classifiers, trained on classified training documents. Support Vector Machine (SVM), k-Nearest Neighbor (k-NN), Logistic Regression (LR), Multinomial Naive Bayes (MNB), and Random Forest (RF) are machine learning algorithms used in this work for text classification. The efficiency of these algorithms on different datasets is analyzed and compared, with Logistic Regression and Support Vector Machine outperforming other models on the IMDB dataset, and k-NN outperforming other models on the SPAM dataset. [9]

[10]Previous studies have shown that no stemming algorithm has utilized the science of phonology, which motivated the authors to explore the use of phonetic algorithms for conflation. Some relevant papers mentioned here also: Krovetz (1993) discusses morphology as an inference process. Lovins (1968) presents the development of a stemming algorithm. Mayfield and McNamee (2003) propose a single N-gram stemming approach. Majumder et al. (2007) introduce YASS, a suffix stripper for stemming. Singh (2003) discusses search algorithms. Tamah (2008) presents work on error-free stemming. UzZaman and Khan (2005) propose T12, a text input system with phonetic support. Xu and Croft (1998) discuss corpus-based stemming using co-occurrence of word variants

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 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.

Both unigrams and bigrams are 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

Dataset 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

VI. Conclusion

VII. Future Work

Acknowledgement

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

[10] 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.