iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80301

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Comparative Analysis of Text Preprocessing Models

Shradha Soni[1], Shraddha Masih[2]
[1]*Research Scholar,* [2]*Professor, School of Computer Science & Information Technology Devi Ahilya Vishwavidyalaya, Indore (M.P)*

Abstract: ***The initial step of text preparation before applying privacy protection methods is crucial. This process enhances accuracy in identifying sensitive data, reduces computational complexity, and increases the efficacy of anonymization procedures.Text preprocessing constitutes a critical component in natural language processing (NLP), exerting significant influence on model performance across various tasks. This paper evaluates the efficacy of diverse text preprocessing models for large datasets. Applied to benchmark datasets, these methodologies are assessed for efficiency and accuracy. The findings elucidate performance trade-offs, thereby providing insights to optimize preprocessing strategies for diverse NLP applications.***

I. INTRODUCTION

Text preprocessing is essential before applying privacy protection techniques, as it improves the precision of sensitive information detection, minimizes processing demands, and enhances the effectiveness of methods like anonymization and differential privacy for safeguarding data. By normalizing and purifying information, preprocessing ensures adequate protection while preserving data usefulness, thereby avoiding excessive or insufficient information shielding. This crucial step facilitates compliance with privacy laws such as GDPR and HIPAA, ultimately enabling efficient and powerful privacy safeguards.

Text preprocessing in natural language processing (NLP) encompasses the processes of cleaning, normalizing, and transforming raw text data to enhance its quality for analysis and modeling (Arpita et al., 2020; Chai, 2022; Kunilovskaya & Plum, 2021). The selection of preprocessing techniques is contingent upon the NLP task, data characteristics, and research objectives (Chai, 2022). Preprocessing decisions substantially influence text mining outcomes, affecting both content and style, and may introduce biases or alter data distribution (Hickman et al., 2020; Kunilovskaya & Plum, 2021). For complex languages such as Arabic, specialized methods are requisite to address unique morphological and grammatical challenges (Nafea et al., 2024). Meticulous evaluation of preprocessing techniques and their implications is essential for enhancing the quality and reliability of NLP applications.

Machine learning is fundamental in NLP, enabling computers to comprehend, interpret, and generate human language (N, 2023), and is crucial for automatic text analysis and classification (Teufl et al., 2010). These technologies have revolutionized NLP, facilitating various applications such as sentiment analysis, language translation, named entity recognition, and text classification (Ii, 2018; N, 2023). NLP and machine learning also extend to malware analysis, known as Machine Language Processing (MLP) (Teufl et al., 2010). Explainability and data bias challenges in NLP machine learning models are critical for ensuring accuracy and fairness (Gholizadeh & Zhou, 2021; Raja et al., 2023). Machine learning in NLP overcomes the knowledge acquisition bottleneck through empirical methods (Daelemans et al., 1997), leading to applications such as chatbots, text summarization tools, and sentiment analysis across various domains (Rajendran et al., 2024; Raza et al., 2023). Addressing data distortion and contextual ambiguity is essential for further progress in NLP and machine learning (Ali Raza et al., 2023).

Text preprocessing techniques, including data cleaning (Keerthi Kumar & Harish, 2018), normalization (Avasthi et al., 2022; Kumar & Harish, 2018), tokenization (Nafea et al., 2024), stop-word removal, and stemming (Avasthi et al., 2022; Nafea et al., 2024), are crucial for preparing raw text for natural language processing tasks. The efficacy of these techniques is dependent on the language and domain (Avasthi et al., 2022), and their selection significantly influences text classification accuracy (Arisha et al., 2023; Avasthi et al., 2022). As textual data proliferates, efficient and scalable preprocessing methods, such as MapReduce-based parallel data preprocessing, become increasingly significant for managing large datasets (He et al., 2010). This paper evaluates the performance of different text preprocessing models for large data set. Further organization of this paper is, Section II describes various processes included in text preprocessing, Section III is about the experimental setup, Section IV explains the outcome received and Section V is conclusion.

II. TEXT PREPROCESSING TECHNIQUES

Textual data can be preprocessed using several techniques. The following are some commonly utilized approaches (figure 1):

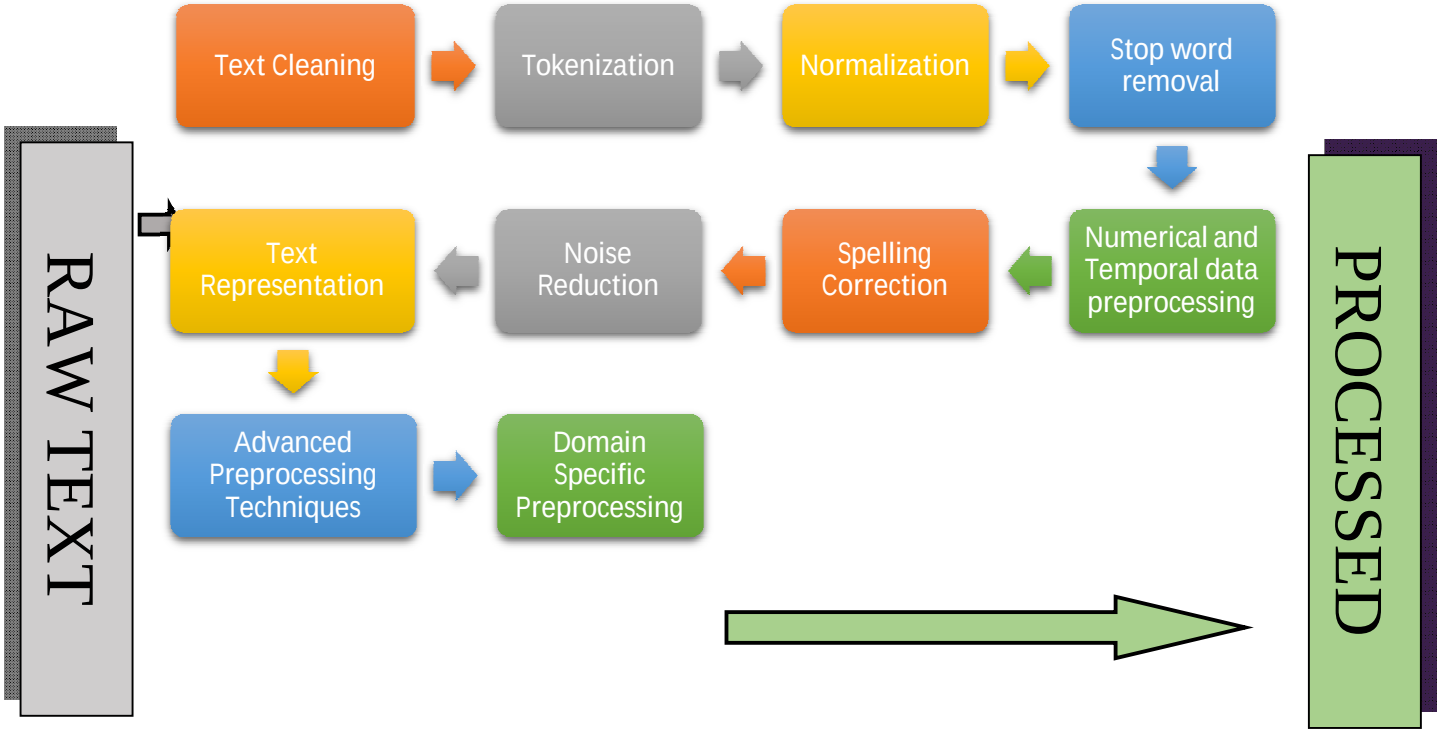


Figure 1: Various Text Preprocessing Techniques

1) Text cleaning focuses on enhancing the quality of textual data through the removal of undesirable elements. This process encompasses the elimination of special characters and punctuation that may introduce noise into the dataset. It also involves the management of HTML tags, other markup, and formatting inconsistencies. Furthermore, text cleaning addresses encoding challenges, such as resolving issues related to UTF-8. These procedures ensure that the data is refined, standardized, and prepared for subsequent analysis or machine learning applications.

2) Tokenization is the process of segmenting text into smaller units for analysis. This process encompasses word tokenization, which divides text into individual lexical units, and sentence tokenization, which delineates text into discrete sentences. Furthermore, it incorporates subword tokenization methodologies, such as Byte Pair Encoding (BPE) and WordPiece, which address the challenge of rare words by decomposing them into meaningful subword components. These tokenization techniques are fundamental for the preparation of textual data for various natural language processing applications.

3) Normalization of textual data entails the standardization of text to enhance consistency and facilitate processing. This process typically encompasses the conversion of all text to lowercase to ensure uniformity. Additionally, it involves the application of stemming algorithms, which reduce words to their root forms through the removal of affixes, and lemmatization, which maps words to their base forms according to linguistic principles. These procedures serve to mitigate variability in text while preserving its semantic content, thereby rendering it suitable for natural language processing tasks.

4) Stop word removal is a text preprocessing technique that eliminates common, non-informative words (e.g., the, and, in) to reduce noise, enhance efficiency, and focus on meaningful content. Tools such as NLTK, spaCy, and Gensim provide predefined or customizable stop word lists. While this technique improves tasks such as text classification and topic modeling, its application depends on the specific context, as certain scenarios may necessitate the retention of stop words.

5) The preprocessing of numerical and temporal data in text involves ensuring consistency and standardization. Numeric normalization entails converting numbers into a standardized format, such as eliminating commas or normalizing decimal representations, to facilitate more effective analysis. Date and time standardization ensures uniform representation of date formats (e.g., converting "Jan 19, 2025" to "2025-01-19") and time formats across the dataset. These procedures are essential for enhancing the interpretability and usability of numerical and temporal data in natural language processing tasks. Through the standardization of these elements, models can more effectively recognize patterns and relationships.

6) Spelling correction focuses on identifying and rectifying misspelled words to enhance text quality and analysis. Methodologies include utilizing dictionaries for reference, rule-based approaches, and machine learning models such as SymSpell or Hunspell. Advanced techniques employ context-aware tools, including Transformers or n-grams, to propose corrections based on surrounding lexical items. Challenges in automated spelling correction encompass addressing ambiguous cases, contextual misspellings (e.g., their vs. there), and domain-specific terminology. Notwithstanding these obstacles, accurate spelling correction remains essential for improving the reliability of natural language processing tasks.

7) Noise reduction in text processing entails the elimination of extraneous or superfluous information that does not contribute to the analysis. This process encompasses the filtration of filler words, excessive punctuation, or irrelevant content that may dilute the substantive data. When addressing social media text and informal language, additional complexities arise, such as colloquialisms, abbreviations, emojis, and non-standard grammatical structures. Specialized methodologies, including text normalization and emoji parsing, are frequently employed to address these challenges. Efficacious noise reduction facilitates the enhancement of data quality and augments the performance of natural language processing models.

8) Text representation is a methodology employed to transform textual data into numerical format for utilization in machine learning models. The Bag-of-Words model represents text as an aggregation of word frequencies, disregarding syntactical structure. TF-IDF (Term Frequency-Inverse Document Frequency) enhances this approach by considering the significance of words within a specific document relative to the entire corpus. Word embeddings, such as Word2Vec and GloVe, represent words as dense vectors in continuous space, capturing semantic relationships between lexical units. These methodologies offer diverse approaches to representing textual data, with embeddings providing a more nuanced understanding for complex Natural Language Processing tasks.

9) Advanced preprocessing techniques enhance text analysis by providing more comprehensive insights into the structure and semantics of the text. Named Entity Recognition (NER) identifies and classifies entities such as names, locations, and dates, facilitating the extraction of structured information. Part-of-Speech (POS) tagging assigns grammatical labels to words, such as nouns, verbs, or adjectives, enabling improved syntactic comprehension. Dependency parsing analyzes sentence structure to identify relationships between words, elucidating their interdependencies. These techniques are fundamental for more complex Natural Language Processing tasks, including information extraction, sentiment analysis, and machine translation.

10) Domain specific preprocessing adapts text cleaning and analysis methodologies to address the distinctive characteristics of various fields, such as medical, legal, or social media text. In domains such as medicine, specialized terminologies and abbreviations (e.g., BP for blood pressure) necessitate careful handling to ensure accuracy. Legal texts require the management of complex language and structure, while social media text frequently incorporates colloquialisms, emojis, and informal language. Preprocessing steps for these domains involve customizing tokenization, stopword removal, and the management of jargon or abbreviations to enhance data quality. This approach ensures that domain-specific nuances are appropriately addressed, thereby improving the accuracy of NLP models.

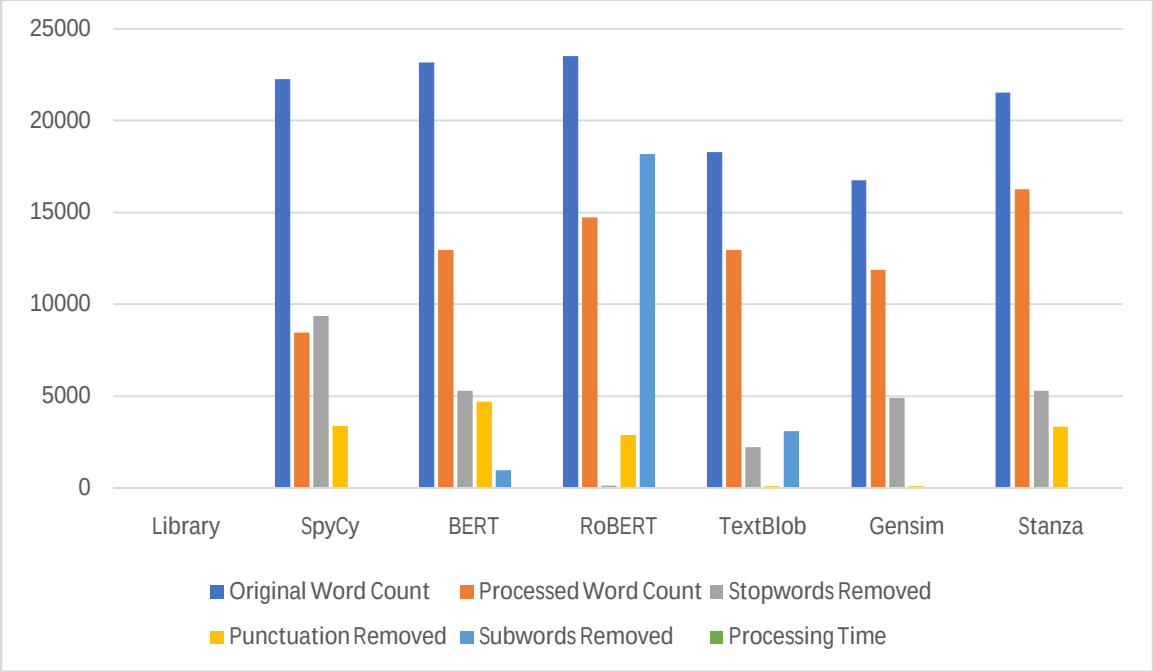
III. EXPERIMENTAL SETUP

An experimental study was conducted to evaluate the performance of various text preprocessing pipelines across different environments, including BERT, spaCy, RoBERTa,NLTK, Gensim, Stanza and TextBlob . The objective was to compare the efficacy of each tool in handling tasks such as tokenization, lemmatization, stopword removal etc. for specific type of data. Through the evaluation of these libraries in multiple environments, the experiment aimed to elucidate the relative strengths and limitations of each tool for specific natural language processing (NLP) tasks. The findings provide valuable insights into the suitability of each library for various applications in the field of natural language processing. The dataset utilized for the experiment is the European Court of Human Rights (ECHR). This dataset comprises approximately 11,500 cases (1.74 GB), including the unprocessed textual content. The in initial operation was performed on the single file.

IV. RESULTS AND FINDINGS

In this experiment, essential text preprocessing tasks, including tokenization, punctuation removal, stopword removal, lemmatization, and spelling correction, were conducted with different libraries. The efficacy of these operations was evaluated in terms of their capacity to clean and standardize text for subsequent analysis. The comparison of the results is shown in Table 1 and graph.

Model/ Library	Original Word Count	Processed Word Count	Stopwords Removed	Punctuation Removed	Subwords Removed	Processing Time
SpyCy	22281	8465	9369	3336	none	56.7 ms ± 14.3 ms per loop
BERT	23199	12949	5274	4696	929	49.5 ms ± 12.2 ms per loop
RoBERT	23541	14719	110	2877	18205	93.2 ms ± 31.5 ms per loop
TextBlob	18295	12961	2203	68	3063	30.5 ms ± 657 µs per loop
Gensim	16743	11857	4886	84	0	55.3 ms ± 1.01 ms per loop
Stanza	21534	16272	5262	3306	0	45.6 ms ± 11.2 ms per loop



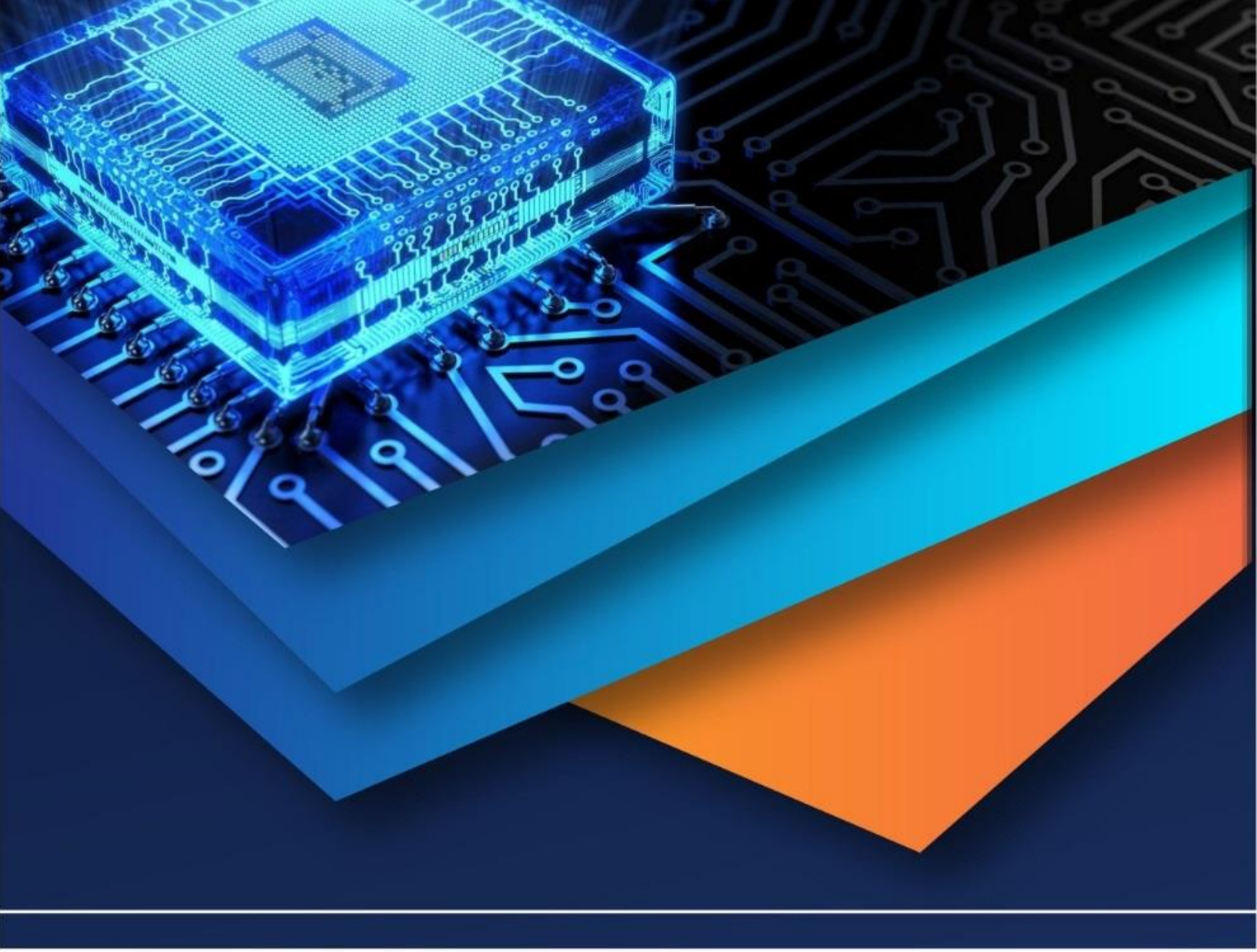
The original document contained a total of 18,674 words. In this experiment, according to the data, TextBlob determined the original word count with high accuracy, whereas SpaCy, BERT, RoBERTa, and Stanza produced significantly higher counts, and Gensim produced a lower count. For other parameters, TextBlob also demonstrated greater accuracy. In terms of processing time, the use of TextBlob was found to be reasonable.

V. CONCLUSION

Based on the analysis of this data, it can be concluded that TextBlob, while demonstrating utility, exhibits potential for enhancement. Through additional refinement, such as the adjustment of parameters or the training of the model on domain-specific data, its performance could be substantially improved. Such refinement may enable the model to better comprehend context and nuances, resulting in more accurate outcomes. The experiment suggests that, with further optimization, TextBlob could yield superior results for specific text preprocessing tasks. Overall, the findings indicate that refinement holds the potential to enhance the model's efficacy in processing text data. As BERT and RoBERT are being widely utilized, they could also be reevaluated with modified parameter settings.

REFERENCES

[1] Ali Raza, A., Parveen, U., Asghar, A., Aslam, H., Fatima, K., Qamar, K., Arslan, A., Fatima, S., & Tehseen, H. (2023). Review to unfold the role of Machine Learning Algorithms in Natural Language Processing. Journal of Policy Research, 9(4), 152–162. https://doi.org/10.61506/02.00136
[2] Avasthi, S., Acharjya, D. P., & Chauhan, R. (2022). Significance of Preprocessing Techniques on Text Classification Over Hindi and English Short Texts (pp. 743–751). springer nature singapore. https://doi.org/10.1007/978-981-19-4831-2_61
[3] Chai, C. P. (2022). Comparison of text preprocessing methods. Natural Language Engineering, 29(3), 509–553. https://doi.org/10.1017/s1351324922000213
[4] Daelemans, W., Bosch, A., & Weijters, T. (1997). Empirical learning of Natural Language Processing tasks (pp. 337–344). springer berlin heidelberg. https://doi.org/10.1007/3-540-62858-4_97
[5] He, Q., Tan, Q., Shi, Z., & Ma, X. (2010). The High-Activity Parallel Implementation of Data Preprocessing Based on MapReduce (pp. 646–654). springer berlin heidelberg. https://doi.org/10.1007/978-3-642-16248-0_88
[6] Keerthi Kumar, H. M., & Harish, B. S. (2018). Classification of Short Text Using Various Preprocessing Techniques: An Empirical Evaluation (pp. 19–30). springer singapore. https://doi.org/10.1007/978-981-10-8633-5_3
[7] N, R. (2023). Machine Learning for Natural Language Processing: Techniques and Applications. https://doi.org/10.59646/csebookc6/004
[8] Nafea, A. A., Khalaf, M. A., Sami, A. B. N., Steiti, A., Ali, A., Majeed, R. R., Bashaddadh, O. M., & Muayad, M. S. (2024). A Brief Review on Preprocessing Text in Arabic Language
[9] Dataset: Techniques and Challenges. Babylonian Journal of Artificial Intelligence, 2024, 46–53. https://doi.org/10.58496/bjai/2024/007
[10] Teufl, P., Lackner, G., & Payer, U. (2010). From NLP (Natural Language Processing) to MLP (Machine Language Processing) (pp. 256–269). springer berlin heidelberg. https://doi.org/10.1007/978-3-642-14706-7_20



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY