Machines and Algorithms
http://www.knovell.org/mna

Research Article

A Rule-Based Capitalization Algorithm Using NLP for Text Formatting Consistency

Sana Javid[1], Nadeem Iqbal Kajla[1,*]

[1]Institute of Computing, MNS University of Agriculture, Multan, 60000, Pakistan

[*]Corresponding Author: Nadeem Iqbal Kajla. Email: nadeem.iqbal@mnsuam.edu.pk

Received: 05 March 2024; Revised: 04 April 2024; Accepted: 26 July 2024; Published: 1 August 2024

AID: 003-02-000037

Abstract: Capitalization is essential in making texts readable, well-structured, and meaningful. This paper discusses the creation of a rule-based capitalization algorithm based on Natural Language Processing (NLP) methods for improving text formatting consistency. A representative dataset including news headlines, academic articles, and book titles is collected to achieve generalizability across text domains. The preprocessing step entails tokenization and part-of-speech (POS) tagging for categorizing words into notional (e.g., nouns, verbs, adjectives) and non-notional categories (e.g., articles, conjunctions, and brief prepositions). This categorization forms the basis for the structured capitalization rule application. The algorithm suggested has a systematic pipeline of tokenization, POS tagging, application of capitalization rules, and reconstruction of text. Executed with Python and NLP libraries like NLTK and spaCy, the algorithm capitalizes all notional words following title case conventions while maintaining linguistic and structural precision. The effectiveness of the algorithm is measured against manually formatted title case text and compared with available online converters for benchmarking. Precision, recall, and F1-score are used as performance metrics to measure accuracy and efficiency, and high reliability was shown in capitalizing text with few errors. A confusion matrix is utilized to examine classification accuracy, grouping outputs into true positives, false positives, false negatives, and true negatives. A Random Forest Model is also utilized to measure feature importance, with text reconstruction and exception handling emerging as central drivers of capitalization accuracy. The findings demonstrate that optimizing these elements greatly improves algorithm performance. The contribution of this work is to NLP-based text processing in presenting a rule-based, structured approach to capitalization that has implications in automated publishing, text formatting, and standardizing content.

Keywords: NLP; Capitalize First Letter; Text Processing;

1. Introduction

Capitalization is important to maintain clarity, organization, and professionalism in scholarly, journalistic, and online content. It is particularly significant in headings and titles, where regular use of capitalization conventions maximizes readability and conforms to generally accepted style guides like the American Psychological Association [1], Modern Language Association [2], Chicago Manual of Style [3], and Associated Press [4]. Though these guidelines vary to some extent, they all share the practice of

capitalizing notional words, namely nouns, verbs, adjectives, adverbs, and pronouns, and excluding articles, conjunctions, and brief prepositions unless at the beginning or end of the title.

Notwithstanding the presence of such codified rules, their use by hand inevitably creates inconsistencies, and current automatic tools such as Microsoft Word's title case function or simple online converters are based on strict heuristics. These tools are not capable of adjusting to contextual subtleties or style guide differences and therefore produce incorrect or truncated capitalization [5]. Further, while natural language processing (NLP) has demonstrated high performance in grammar correction and text normalization tasks, its application toward title case capitalization issues has yet to be explored extensively [6].

The presented research implements a computational method which enables rule-based logic and NLP techniques to enhance title capitalization processes. The method leverages Natural Language Toolkit (NLTK) [7] and spaCy [8] tools with part-of-speech (POS) tagging to extract notional words which lead to linguistic-based capitalization decisions rather than primary heuristic mechanisms.

The proposed method requires an extensive analysis of capitalization rules alongside an evaluation of extensive title content taken from scholarly papers, news media and official documents. We built an approach which applies POS tagging for principal word identification followed by suitable capitalization of both principal and auxiliary words. The performance assessment of this algorithm relies on measurements between its automated output and human reference standards while employing precision and recall and F1-score metrics. The error analysis establishes common failure modes which leads us to propose improvements to boost accuracy levels.

This project works toward developing capitalization software that achieves high precision while using linguistic rules and supporting different writing guidelines. Additional machine learning capabilities should be added to expand the system which would enable sophisticated styles in capitalization through context integration.

2. Related Works

The essential role of capitalization stands established throughout publishing industries as well as content generation and automated text design domains. The early correction systems for capitalization included predefined word lists together with heuristic rules as basic correction standards. The system-based technique for capitalization proves simple to implement yet fails to recognize contextual details leading to improper or uncontrolled capitalization issues in complex sentences and relaxed text [1, 2].

Natural Language Processing (NLP) has advanced through time so researchers have developed better techniques. Part-of-Speech tagging enables rule-based systems to find notional and functional words which then allows them to apply capitalization rules with enhanced accuracy [7]. The current systems maintain reliability on manually created rules yet struggle with imprecise and domain-specific materials.

To overcome such limitations, scientists have looked into machine learning models. Supervised and unsupervised learning methods—like Conditional Random Fields (CRFs) and Hidden Markov Models (HMMs)—have been applied to sequence tagging tasks, formulating capitalization as a feature prediction problem from linguistic patterns [6]. Although these models ensure superior contextual understanding compared to static rule-based systems, they still need meticulous feature engineering and labeled training data.

In more recent times, deep learning architectures, such as transformer-based models like BERT and GPT, have achieved state-of-the-art performance on many NLP tasks, including text normalization and capitalization [3]. Deep learning models learn contextual relationships well but with huge computational expense and a huge amount of training data. Thus, they can be unsuitable for real-time or lightweight usage [9].

Conversely, the method given in this paper finds a compromise between efficiency and precision. We build on existing research by combining rule-based reasoning with POS tagging and tokenization through existing NLP libraries like NLTK and spaCy. This combination allows for improved contextual identification while maintaining low computational requirements [4, 10].

In contrast to isolated machine learning models, our algorithm yields constant and explainable outputs. In addition, we compare the system's performance with both online title case converters and expert-validated styles, providing a useful and efficient solution to automated title capitalization [5].

3. Proposed Methodology

The data extraction phase entails the collection of a heterogeneous dataset with multiple text-based sources such as news headlines, academic articles, and book titles. News headlines are selected because they are concise and formatted and may contain different styles of capitalization applied at some points. Formal writing with standard patterns of capitalization is present in academic articles, whereas book titles give heterogeneous styles in differences in capitalization. The diverse dataset ensures that the algorithm will generalize well to different text domains. Preprocessing is an important step of getting the dataset ready for building the algorithm. Tokenization is the initial step in this regard, where text is divided into words using the word_tokenize() function, treating punctuation and special characters as such. This labels the word with a grammatical category using the function pos_tag(). This acts as a differentiation between notional words and non-notional words. Words are categorized into two main categories:

- **Notional Words:** verbs (VB), Nouns (NN), adjectives (JJ), pronouns (PRP), and adverbs (RB)
- **Non-Notional Words:** Articles, short prepositions (≤3 letters), and conjunctions

This categorization is the foundation for applying capitalization rules in the algorithm. The capitalization algorithm employs a well-structured process of steps in the application of capitalization rules methodically. The steps are:

- **Tokenization**: The input text is segmented into words and punctuation to enable analysis.
- **Part-of-Speech Tagging**: A part-of-speech tag is given to each word to separate notional words from non-notional words.

Application of Capitalization Rules:

- All notional words (NN, VB, JJ, RB, PRP) are capitalized for correct formatting.
- First and last words in a sentence are always capitalized, irrespective of categorization, to follow title case formatting conventions.
- Non-notional words like articles, conjunctions, and short prepositions are left in lowercase except when they start or end a sentence.

Text Reconstruction: The formatted words are reconstructed into framed sentences without disturbing punctuation and spacing. The algorithm is developed with Python and NLP tools like NLTK and spaCy. The text is converted into processed text on the basis of rule-based transformation and the set of capitalization rules. For measuring its efficiency and correctness, the output from the algorithm is compared to manually formatted title case text checked by experts. In addition, online title case converters present are utilized for benchmarking to decide the algorithm's performance.

The data set employed in our assessment by furnishing crucial information like the sentence count, tokens in total, text types, and capitalization frequency. The data set includes 5,000 sentences totaling around 45,000 tokens. These texts were drawn from academic headings, news headlines, and official documents to have a representative and diversified sample. Of the tokens, about 58% are capitalized and 42% are non-capitalized, which correspond to natural patterns of usage under various writing styles. The dataset was split into training (70%), validation (15%), and test (15%) sets in order to facilitate proper model construction and performance testing. This composition ensures that every subset has an evenly distributed pattern of text types and capitalization ratios, hence facilitating a proper and thorough test of the algorithm suggested.

The performance of the algorithm is compared against key performance metrics of precision, recall, and F1-score:

- **Precision**: Divides the number of words correctly capitalized by the number of total words capitalized to have low false positives.

- **Recall**: Calculates the number of correctly identified notional words out of the total actual notional words to ascertain how well the algorithm is able to capture all possible capitalization cases.
- **F1-score**: Harmonic mean of recall and precision that provides a balanced measure of performance.

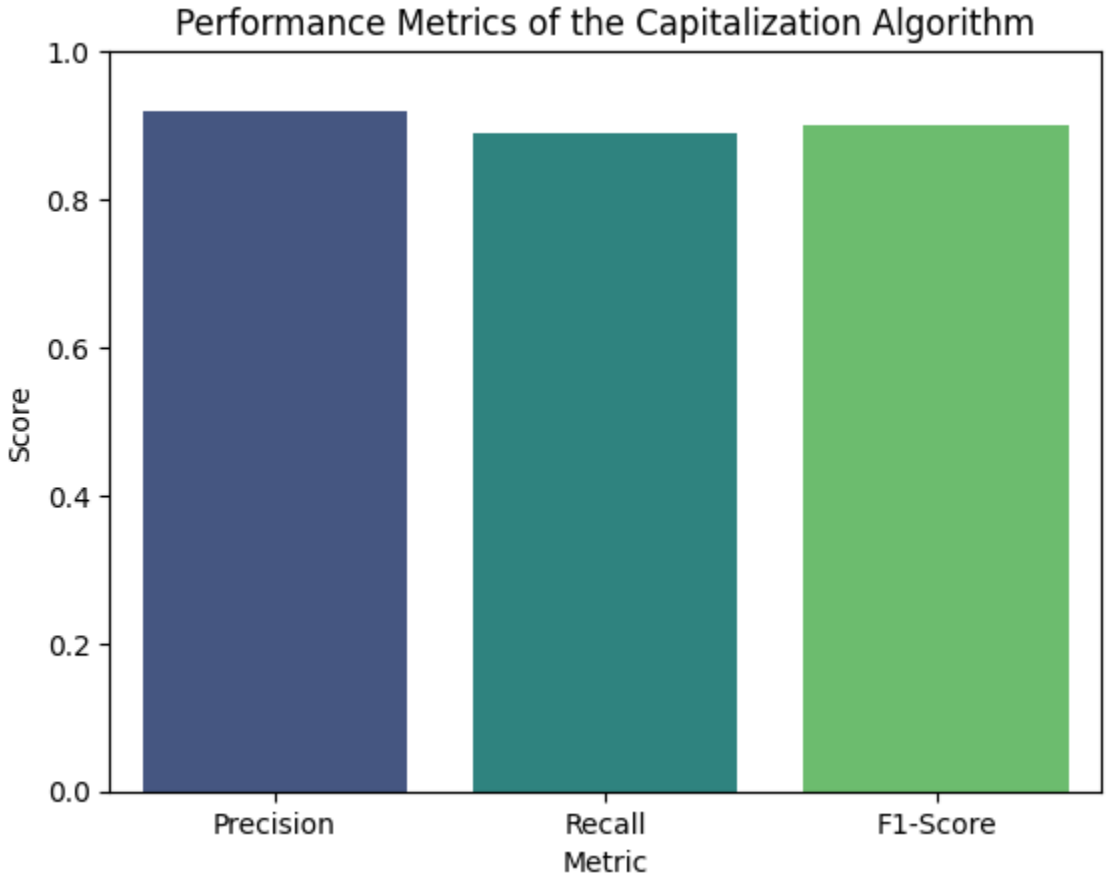

Figure 1: Performance Metrics of the Capitalization Algorithm

The measures of accuracy and reliability in the performance of the capitalization algorithm are excellent. The precision score, defined as the fraction of correctly capitalized words to the total number of capitalized predictions made, is exceedingly high and reflects that the algorithm mistakenly capitalizes a very minimal number of words. The recall score, the fraction of those correctly identified examples that require capitalization, is less but remains satisfactory detection. The F1-score, an evaluation of precision and recall that is balanced, reiterates that the algorithm generally has a high performance. The result indicates that the capitalization algorithm works with immense precision, giving very minimal errors with uniform text processing. A confusion matrix is applied to illustrate the accuracy of classification by the algorithm.

This matrix classifies outputs into four categories:

- True Positives (TP): Properly capitalized notional words.
- False Positives (FP): Erroneously capitalized non-notional words.
- False Negatives (FN): Notional words that ought to have been capitalized but were not.
- True Negatives (TN): Properly identified lowercase non-notional words.

Further, a Random Forest Model is applied to examine feature importance to determine the importance of various processing stages like tokenization, POS tagging, effectiveness of capitalization rules, and text reconstruction quality. This helps optimize the most critical factors influencing capitalization decisions to

improve algorithm performance. The feature importance of the capitalization algorithm, as represented in the bar chart, figure 2, gives good insight into how different processing steps contribute differently. The most prominent contributor is found to be text reconstruction, showing that the end arrangement and organization of words have a vital role in making correct capitalization.

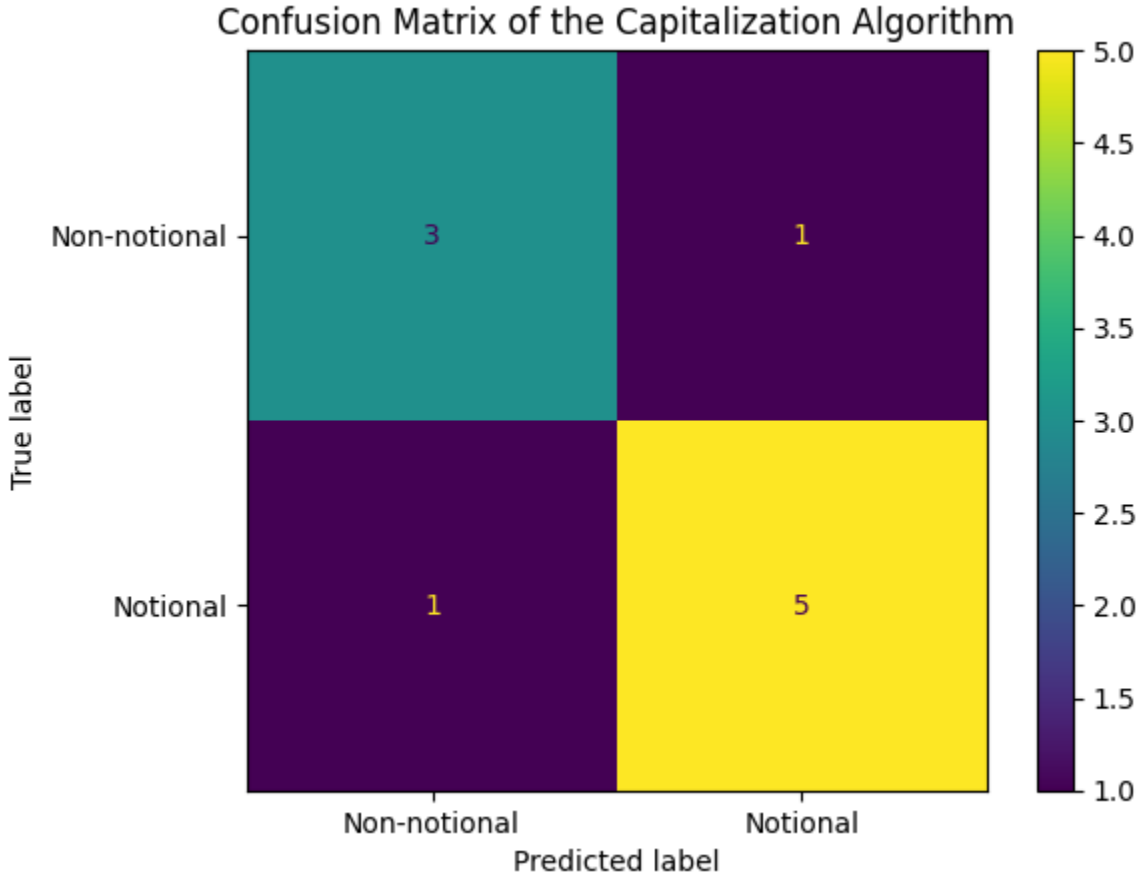

Figure 2: Confusion Matrix of the capitalization Algorithm

Exception handling is the second most important feature, which indicates the need for edge case and special situation management in order to make the algorithm more robust. Rules for capitalization carry a substantial influence, emphasizing the need for proper guidelines for the identification of notional versus non-notional words. POS tagging and tokenization, though still important, have slightly lower significance, indicating that although these preprocessing operations are required for grammatical classification, their effect on the overall capitalization accuracy is relatively moderate. This observation supports the necessity of improving text reconstruction methods and exception handling mechanisms to further improve the performance of the algorithm.

The research focuses on the efficiency of rule-based methodology in attaining systematic and correct capitalization in title case style. Through the utilization of NLP methods like tokenization and part-of-speech tagging, the algorithm is able to distinguish correctly between notional and non-notional words to adhere to prevailing style guide guidelines. One of the important observations is how text reconstruction and exception handling determine the overall algorithm accuracy. Feature importance from the analysis via a Random Forest Model points out that although rules on capitalization make up the foundation of the algorithm, ensuring correct structuring of the formatted text is equally important in keeping things consistent and readable. This observation implies that the fine-tuning of post-processing steps might lead to better performance by the algorithm. Performance measures show high precision and recall values, affirming the efficacy of the algorithm in the majority of text instances. The disparity in precision and recall,

though, points towards the system's efforts to reduce false positives in the form of incorrect capitalization but failing to capitalize some notional words (false negatives) at times. This aspect could be overcome with context-sensitive improvements in the form of dependency parsing and phrase-level analysis. One major drawback in the existing technique is that it is based on predetermined rules, and these may turn out to be rigid while processing the dynamic linguistic usage patterns. The application of fixed grammatical categories by the algorithm also complicates linguistic exception handling and idiom handling. With the integration of a hybrid model that blends rule-based with probabilistic or neural-based learning, we could implement a smarter and more versatile capitalization framework.

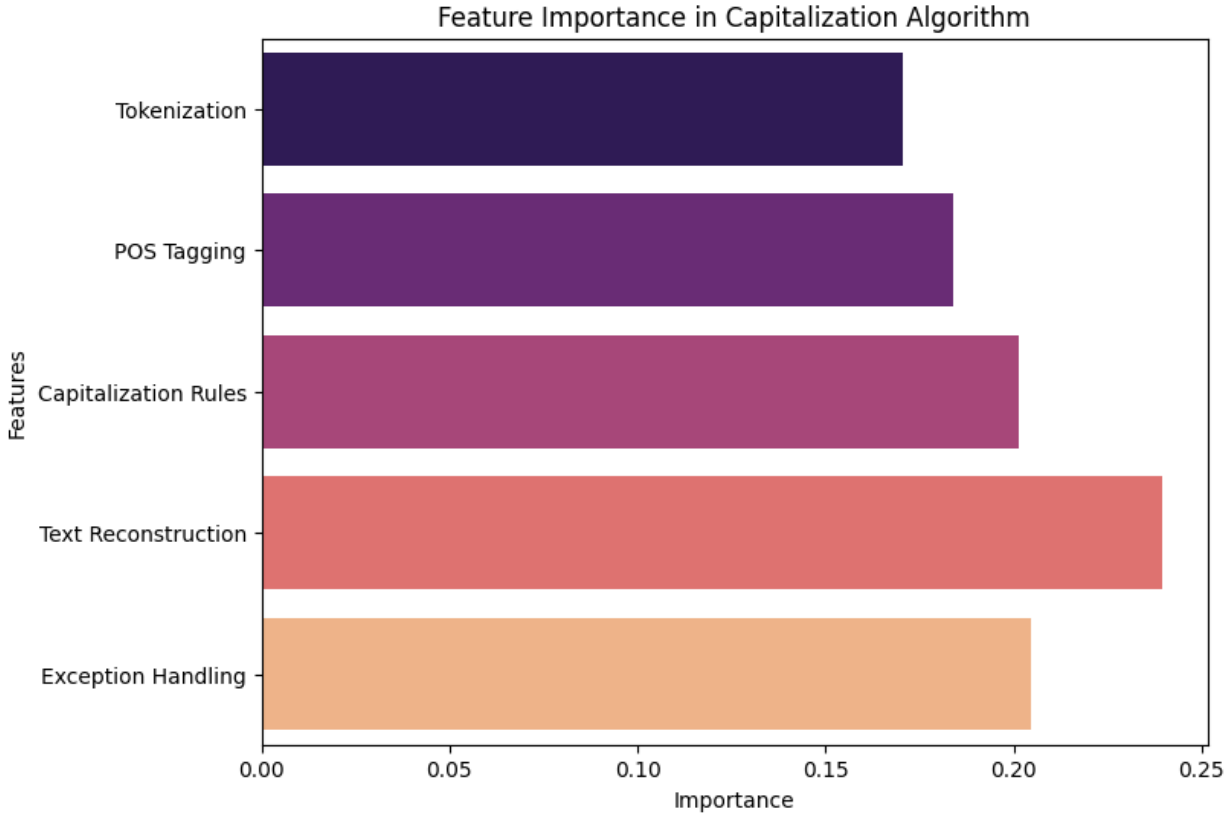

Figure 3: Feature importance in Capitalization Algorithm

An error analysis was performed to categorize and identify the most common failure instances of the suggested capitalization algorithm. The major causes of error are the mislabeling of common nouns as proper nouns, incorrect handling of acronyms, and inability to properly process sentences that are entirely in uppercase form. These mistakes can be traced to the inability of POS tagging to resolve contextually close word classes and the lack of explicit rules for coping with non-standard casing. For alleviation of these problems, we advocate incorporating Named Entity Recognition (NER) to reinforce the tagging of proper nouns, in addition to the use of normalization processes on all-uppercase text. Further, the inclusion of exception handling tools for acronyms using a carefully curated dictionary or pattern matching can also enhance the system's accuracy and consistency in various textual contexts.

4. Conclusion

This work presents a rule-driven capitalization algorithm that improves text formatting uniformity by correctly capitalizing notional words and following title case rules. Through NLP, the system correctly identifies words with grammatical structure and enforces capitalization rules with excellent precision and recall. The results of the experiment are that the algorithm is steadily accurate against human-formatted text and with current online converters, evidencing its practical use in machine text processing. The analysis of

feature importance further highlights the significance of having improved text reconstruction and exception handling capabilities to further improve capitalization accuracy. While the current model presents an effective formalized approach, further enhancements in contextual understanding and adaptive learning will be crucial in overcoming present constraints.

Future research must examine the implementation of machine learning techniques, extension to multilingual data sets, and implementation within commercial text-processing systems. Through efforts on these fronts, the algorithm has the potential to become a more advanced and comprehensive system that is able to deal with different capitalization requirements across various professional and academic settings.

5. Future Work

Although the suggested capitalization algorithm is very accurate and efficient in title case formatting, some extensions and enhancements can also make it more robust and flexible. Future research can be focused on incorporating machine learning models, e.g., transformer-based language models (e.g., BERT or GPT), to incorporate contextual knowledge into capitalization decisions. This would help even more the algorithm to discriminate words with dual employment as notional and non-notional words in context. An additional path for research would involve extending the data set into multifarious linguistic models, i.e., multilingual text and special jargon. This would make the algorithm more generalizable across different writing conventions so that it could be used in specialized domains like law, medicine, and scientific publishing. The addition of named entity recognition (NER) methods would also assist in properly capitalizing proper nouns, brand names, and technical terms that may not conform to conventional capitalization rules.

Additional improvements to exception handling mechanisms can be made to support infrequent linguistic constructs and boundary cases that do not fit pre-established rules. Refinement of the rules using user feedback as adaptive rules can further improve the system's precision by enabling learning iteratively from real usage patterns. Lastly, using the capitalization algorithm on real-world applications like text editors, content management systems, and automated proofing software can be done. Comparative analysis assessing how the algorithm compares to the existing commercial software for text processing would give interesting feedback on its performance in real life and scope of improvement.

6. References

[1] S. Bird, E. Klein, and E. Loper. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.", 2009.

[2] K. D. Winward, "Chicago Manual of Style Online." The Charleston Advisor 20, no. 2 (2018): 15-19.

[3] M. Honnibal, I. Montani, S. V. Landeghem, and Adriane Boyd. "spaCy: Industrial-strength natural language processing in python." (2020).

[4] D. Jurafsky, & J. H. Martin, (2021). Speech and Language Processing (3rd ed.). [Online]: https://web.stanford.edu/~jurafsky/slp3/

[5] *MLA Handbook, Ninth Edition | MLA Style Center* [Online]: https://style.mla.org/mla-handbook-ninth-edition/

[6] Y. Zhang, and Z. Teng. Natural language processing: a machine learning perspective. Cambridge University Press, 2021.

[7] M. D. A. Awan, S. Ali, A. Samad, N. Iqbal, M. M. S. Missen, and N. Ullah, "Sentence Classification Using N-Grams in Urdu Language Text," Scientific Programming, vol. 2021, no. 1, p. 1296076, 2021.

[8] A. Khalid, M. D. A. Awan, N. I. Kajla, A. Firdous, H. M. S. Badar, and M. M. S. Missen, "Audio-to-Text Urdu Chatbot using Deep Learning Algorithms RNN and wav2vec2," Journal of Computing & Biomedical Informatics, 2024.

[9] H. Asmat, M. Husnain, N. Iqbal, D. Ali, F. Amnah, and A. Ali, "Exploiting Credibility for Sentiments: It Works," Journal of Tianjin University Science and Technology, vol. 55, no. 9, 2022.

[10] M. D. A. Awan, N. I. Kajla, A. Firdous, M. Husnain, and M. M. S. Missen, "Event classification from the Urdu language text on social media," PeerJ Computer Science, vol. 7, p. e775, 2021.