Indonesian Journal of Data and Science

Volume 6 Issue 1
ISSN 2715-9936
https://doi.org/10.56705/ijodas.v6i1.184

Research Article

Improving Part-of-Speech Tagging with Relative Positional Encoding in Transformer Models and Basic Rules

Abdukareem Mohammad [1], Abdullahi Mohammed [2], Achir Jerome Aondongu [3,*]

[1] Ahmadu Bello University, Zaria 810211, Kaduna, Nigeria, mmmhammmad@gmail.com
[2] Ahmadu Bello University, Zaria 810211, Kaduna, Nigeria, moham08@gmail.com
[3] Joseph Sarwuan Tarka University, Makurdi 970101, Benue, Nigeria, achir.jerome@uam.edu.ng

Correspondence should be addressed to Achir Jerome Aondongu; achir.jerome@uam.edu.ng

Received 02 December 2025; Accepted 27 March 2025; Published 31 March 2025

Abstract:

Introduction: Part-of-speech (POS) tagging plays a pivotal role in natural language processing (NLP) tasks such as semantic parsing and machine translation. However, challenges with ambiguous and unknown words, along with limitations of absolute positional encoding in transformers, often affect tagging accuracy. This study proposes an enhanced POS tagging model integrating relative positional encoding and a rule-based correction module. **Methods**: The model utilizes a transformer-based architecture equipped with relative positional encoding to better capture token dependencies. Word embeddings, POS tag embeddings, and relative position embeddings are combined and processed through a multi-head attention mechanism. Following the initial classification by the transformer, a corrective rule-based module is applied to refine misclassified tokens. The approach was evaluated using the Groningen Meaning Bank (GMB) dataset, comprising over 1.3 million tokens. **Results**: The transformer model achieved an accuracy of 98.50% prior to rule-based corrections. After applying the rule-based module, overall accuracy increased to 99.68%, outperforming a comparable model using absolute positional encoding (98.60%). Additional evaluation metrics, including a precision of 0.92, recall of 0.89, and F1-score of 0.90, further validate the model's effectiveness. **Conclusions**: Incorporating relative positional encoding significantly enhances the transformer's contextual understanding and performance in POS tagging. The addition of a rule-based correction module improves classification accuracy, especially for linguistically ambiguous tokens. The proposed hybrid model demonstrates robust performance and adaptability, offering a promising direction for future multilingual POS tagging systems.

Keywords: Corrective rule-based Module, NLP, Part-of-Speech tagging, Self-attention, Transformer, Word Embedding.
Dataset link: https://developer.ibm.com/exchanges/data/all/groningen-meaning-bank/

1. Introduction

Parts of speech (POS) also known as word classes, or syntactic categories [1] are useful because they reveal a lot about a word and its neighbours. POS tagging is one of the most important tasks in the field of natural language processing (NLP), as it assigns a POS tag to each word in a sentence [2], [3]. The performance of any NLP system depends on the accuracy of a POS tagger [4] and two main issues that affect the accuracy of POS tagger are unknown words and ambiguity. A POS grammatically classifies words that commonly includes verbs, adjectives, adverbs, nouns, etc [5]–[7]. Generally, POS tagging is an upstream task for other NLP tasks, such as semantic parsing, machine translation [8], and relation extraction [9], to improve their performance and accuracy. It is one of the challenging research areas in NLP as it requires good knowledge of a particular language with large amounts of data or corpora for feature engineering, which can lead to achieving a good performance of the tagger [10].

POS tagging have been implemented by several researchers using different techniques [11] such as Bayesian Models, Markov Models, Maximum Entropy, and Transformation-Based Learning (TBL). Traditional methodologies involve rule-based and statistical POS taggers, and transformation-based techniques [12], [13].

The oldest POS tagging technique is rule-based tagging system. It utilizes carefully crafted lingual established laws approach to allot tags to words [14]. A set of manually composed rules are used, and additionally contextual data is utilized to allocate POS tags to words in the rule-based POS tagging. This technique is very complex and time consuming [2].

Stochastic POS technique is a corpus-based technique that harnesses the probabilities of occurrences of words for a particular tag [2], [3]. Since it is based on likelihood of word occurrence [15], it is possible that two words are same but according to the context in a sentence, both words will have different POS tags.

Transformation-based part-of-speech tagging is an in-stance of the transformation-based learning (TBL) approach to machine learning. It is an algorithm that automatically extract rule or linguistic information from a sample of manually annotated corpus and learns lexical or morphological and contextual information from correctly annotated corpus without human intervention or expert knowledge [16]. The aim of the research work is to adopt relative positional encoding in transformer multihead attention to improve token context information. Normally, traditional POS taggers face challenges with ambiguity and unknown words, which significantly affect tagging accuracy. Further, absolute positional encodings in transformer models limit their contextual understanding of token dependencies. To improve POS tagging accuracy, this research introduces relative positional encoding in transformers and then augment predictions with a rule-based correction module.

The research is limited to English POS tagging using the GMB dataset. The proposed architecture relies on predefined rules, which may limit adaptability to new languages.

Related Works

In this section, POS literatures will be reviewed as well as Penn Treebank Tagset and Transformers will be discussed.

Tag Set

A well-chosen tag set is also important in representing POS and consist of syntactic classes [5]. In POS tagging, suitable tags are attached to each word of a sentence and the set of tags is called tag-set and tagging means assigning a single POS to each word or punctuation marker in a corpus [5], [10].

Penn Treebank Tagset

The Penn Treebank POS tagset contains 36 POS tags and 12-character tags [2] for characters and its widely used to annotate large corpora.

Table 1. Penn Tagtree POS Tagset

Number	Tag	Description	Number	Tag	Description
1.	CC	Coordinating conjunction	19.	PRP$	Possessive pronoun
2.	CD	Cardinal number	20.	RB	Adverb
3.	DT	Determiner	21.	RBR	Adverb, comparative
4.	EX	Existential *there*	22.	RBS	Adverb, superlative
5.	FW	Foreign word	23.	RP	Particle
6.	IN	Preposition or subordinating conjunction	24.	SYM	Symbol
7.	JJ	Adjective	25.	TO	*to*
8.	JJR	Adjective, comparative	26.	UH	Interjection
9.	JJS	Adjective, superlative	27.	VB	Verb, base form
10.	LS	List item marker	28.	VBD	Verb, past tense
11.	MD	Modal	29.	VBG	Verb, gerund or present participle
12.	NN	Noun, singular or mass	30.	VBN	Verb, past participle
13.	NNS	Noun, plural	31.	VBP	Verb, non-3rd person singular present
14.	NNP	Proper noun, singular	32.	VBZ	Verb, 3rd person singular present
15.	NNPS	Proper noun, plural	33.	WDT	Wh-determiner
16.	PDT	Predeterminer	34.	WP	Wh-pronoun

Number	Tag	Description	Number	Tag	Description
17.	POS	Possessive ending	35.	WP$	Possessive wh-pronoun
18.	PRP	Personal pronoun	36.	WRB	Wh-adverb

The character tags are of no value to this research work, hence **Table 1** contains the 36 tags of interest adopted for this research work.

POS Tagging

Wang [17] used Bidirectional Long Short-Term Memory (Bi-LSTM) for POS tagging. A word embedding layer and a function is applied to keep track of the original case of words. Ling, Luís [18] proposed a character to word (C2W) model that is based on Long Short-Term Memory (LSTM), and it composes representations of characters into representations of words and the result showed that the C2W model had better performance compared to the word lookup tables in POS tagging. Plank [19] also applied Bi-LSTM as base model for POS tagging. The model inputs are word level embeddings and character-level embeddings. Wang [20] applied the concept of transferred learned to solve the problem of lack of large corpora of training samples. Qi [21] proposed a POS tagger that adopts a Bi-LSTM with inputs from the concatenation pretrained word embedding, trainable frequent word embedding and character level embedding. It also uses affine classifiers for each kind of tag [22]. The research of Warjri [23] and AlKhwiter [24] combined Bi-LSTM with Conditional Random Field (CRF) to propose a model since CRF can learn sentence-level tag information. Li [3] use data pre-processing rules to tag words, mask the untagged words and pass it to a transformer to predict POS for the tags and the transformer uses absolute position encoding. The pruning rules focus on which candidate POS tags to eliminate rather than which candidate POS tag is correct, thereby creating a more complex system.

Transformers

Natural Language Processing has revolutionized the world especially after the advent of transformers [25]. The transformer model represents a serious evolutional moment of deep learning models [26]. Different from conventional sequence models, which usually involve recurrent or convolutional layers, the transformer model harnesses attention mechanisms, thereby setting a new performance precedent in natural language processing tasks [27]. Transformer block consists of two main components: a multi-head self-attention mechanism and a fully connected feedforward network [28]. Transformer adopts attention mechanism with Query–Key–Value (QKV) model [3], [29] and the self-attention mechanism in transformer captures relationship between words in a sentence, regardless of the distance [3]. An attention function is computed by mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The input consists of queries and keys of dimension dk, and values of dimension as shown in Equation 1.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

Where q is as a set of queries simultaneously, packed together into a matrix Q while the keys and values a real also packed together into matrices K and V.

Rule-based Tagging

Rule-based part of speech taggers assign a tag to a word based on manually created linguistic rules [7] to assign tags to words in a sentence [12]. For instance, a word that follows adjectives should be tagged as a noun.

This approach significantly reduces the amount of information storage since knowledge is represented in the form of rules, easier to understand, highly portable from a text corpus to another [11].

Despite being just, there are some difficulties found with the rule-based method; the necessity of linguistic background, lack of guarantee that all linguistic rules are captured, lack of flexibility if language model transportation and many more [3].

The research brings to fore an implementation of relative positional encoding in transformers for POS tagging as well as a hybrid architecture combining transformer predictions with rule-based corrections. The paper discusses related works, introduces the proposed method, details the experimental setup, presents results, and concludes with key findings and future research directions.

2. Method:

The text corpus is tokenized and fed into the transformer model and classified. Output of the model then fed into a rule-based system for corrective adjustment as shown in **Figure 1**.

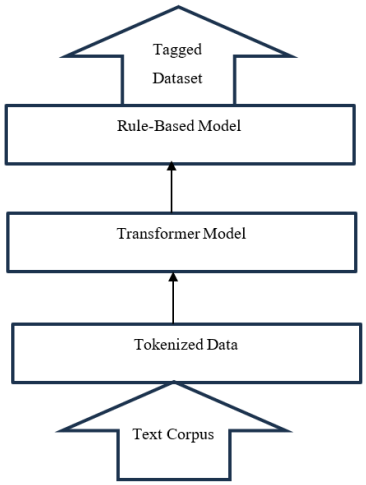

Figure 1. General architecture of the proposed system

Transformer-based Tagging

Given a corpus which contains sentences S_1, S_2 ..S_n. For each word W_i in S_i, each W_i has a corresponding POS tag T_i and the T_i of W_i is much dependent on the position of such W_i in the sentence. Es in this research, the transformer model captures this dependency information with the help of relative positional encoding algorithm, in order to identify the right T_i for each W_i at every given time. The algorithm for determining relative position is stated in Algorithm 1.

Algorithm 1: pseudocode for relative positioning encoder.

```
def get_relative_position_matrix(seq_length):
    relative_positions = np.zeros((seq_length, seq_length), dtype=int)
    for i in range(seq_length):
        for j in range(seq_length):
            relative_positions[i, j] = j - i
    return relative_positions


def relative_position_embedding(relative_positions, embedding_dim):
    max_relative_position = 2 * seq_length - 1
    embeddings = np.random.rand(max_relative_position, embedding_dim)
    relative_position_embeddings = np.zeros((seq_length, seq_length, embedding_dim))
    for i in range(seq_length):
        for j in range(seq_length):
            relative_position_embeddings[i, j] = embeddings[relative_positions[i, j] + seq_length - 1]
    return relative_position_embeddings
def apply_relative_position_embedding(input_embeddings, relative_position_embeddings):
    seq_length, embedding_dim = input_embeddings.shape
    output_embeddings = np.zeros_like(input_embeddings)
    for i in range(seq_length):
        for j in range(seq_length):
            output_embeddings[i] += input_embeddings[j] +relative_position_embeddings[i, j]
    return output_embeddings
```

As shown in Algorithm 1, the relative position matrix encodes the relative distances between all pairs of tokens in the sequence.

a. Generate Relative Position Matrix

- The relative positions matrix is initialized with zeros
- For each pair of positions (*i*, *j*) in the sequence, the relative position *j* - *i* is calculated and stored

b. Generate Relative Position Matrix

These embeddings are used to represent the relative positions in a high-dimensional space.

- The embeddings matrix stores the embeddings for all possible relative positions, ranging from -seq_length + 1 to seq_length - 1.
- For each pair (i, j), the corresponding relative position embedding is assigned based on the precomputed relative positions

c. Incorporate Relative Position Embeddings into Self-Attention

- The function iterates over each token pair (i, j) in the sequence
- It accumulates the contributions from both the input embeddings and the relative position embeddings into the output embeddings

The relative tokens positioning is then applied to transformer model for proper classification of tokens in documents.

Architecture

In the architecture, the unstructured dataset is pre-processed and fed into a transformer model which applies attention mechanism to classify each token into the various classes of POS. The resulting POS tags from the model, together with the pre-processed data is again fed into a POS rule-based system for corrective classification, producing comprehensive POS tags for each token in the document as illustrated in **Figure 2**.

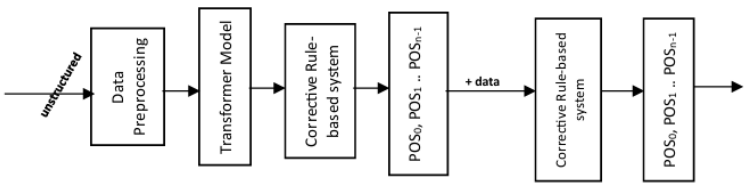

Figure 2. Architecture of POS model

The model uses a transformer encoder multi-head attention mechanism as base to capture dependency information in each given sentence. The input includes the token (TK), POS tag and token position. The token, POS tag, and position are both embedded, and the embedded tokens are concatenated to the embedded tags and then added to the embedded positions, which now form the input for the transformer training as shown in **Figure 3**.

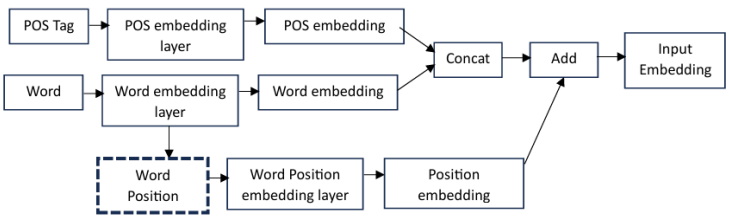

Figure 3. Transformer token embedding layer (14)

Figure 3 illustrates the complete architecture of the transformer-based word embedding system designed for POS tag prediction. The model processes tokens and POS tags at the input stage, converting them into input embeddings within the input layer. Leveraging the self-attention mechanism, the Transformer encoder incorporates bidirectional contextual information, further enhanced by the relative position encoding, depicted as dotted lines. Following the encoder's

computations, a linear layer with a softmax function generates predictions for each POS tag. These predicted tags, alongside the input tokens, are subsequently passed into a rule-based module to refine the tagging process

Rule-based Tagging Module

The output of the transformer model is fed into the corrective rule-based module to rectify perceived irregularities in the classification where necessary for a better classification output.

The rule-base module is applied on the test data contained in the val_loader tensor. The tensors are decoded with their corresponding POS tags, the rules are then applied to the tokens contained in the incorrect_token.

During transformers model training and evaluation, a record of tokens whose evaluations are negative are stored in incorrect_token variable and an incremental value added to incorrect_predict to keep record of number of incorrectly predicted tokens, while the sentence and token index stored in sentence_array and tonken_index respectively. The sentence refers to the sentence number from which the token belongs, while the token index refers to the index of the token in the sentence.

After the rules are applied to the test data, the values of both incorrect_predict and correct_predict are adjusted appropriately to accommodate the total number of incorrectly predicted and correctly predicted token.

The rules as applied in this research are stated in Algorithm 2.

S/No	***Algorithm 2: pseudocode for rule-based module.***
1	*If tg_0 is TO and tg_{+1} is DET, then tg_0 is switched to IN*
2	*If tg_0 is TO and t_{+1} is CAP, then tg_0 is switched to IN*
3	*If t_0 is to (IN) and tg_{+1} or tg_{+2} is VBG, then tg_0 is switched to TO*
4	*If tg_0 is VBN and tg_{-1} is CAP, then tg_0 is switched to VBD*
5	*If tg_0 is VBD and tg_{-1} is PRE 1 or 2 token is had, then tg_0 is switched to VBN*
6	*If tg_0 is VB and $tg_{-1,}$ or tg_{-2} is DET, then tg_0 is switched to NN*
7	*If tg_0 is NN and $tg_{-1,}$ is TO, then tg_0 is switched to VB*
8	*If tg_0 is NN and $tg_{-1,}$ is MD, then tg_0 is switched to VB*
9	*If tg_0 is JJ and $tg_{+1,}$ is not NN, NNS, NNP or NNPS, then tg_0 is switched to NN*
10	*If tg_0 is VBP and $tg_{0,}$ is first token in sentence, then tg_0 is switched to VB*
11	*If tg_0 is VBP and $tg_{0,}$ is first token in sentence, then tg_0 is switched to VB*
12	*If t_o is CAP and t_o Not first token in the sentence, then t_o is switched to NNP*

Algorithm 2 states the rules applied to output data of the transformer model to adjust some ill-classified data item.

If a base tag (tg0) is TO and if preceded by a determinant, then the TO should be an IN tag.

If a base tag (tg0) is TO and if preceded by a token whose first letter is upper case, then the TO should be an IN tag and that implies to all rules in the algorithm.

Each correction in the rule-based approach is added to the correct_predict of the transformer model that holds a total of correct predictions. In the end, the overall accuracy was evaluated using in equation.

$$\boldsymbol{accuracy} = \frac{\boldsymbol{correct\_predict}}{\boldsymbol{len(test\_data)}} \quad (2)$$

In all, metrics such as accuracy, precision, recall, and F1-score were employed to evaluate the model's performance

3. Results and Discussion

Dataset

The dataset was obtained from Groningen Meaning Bank (GMB) [30]. It is a large semantic annotated corpus of millions of tokens with various tags, but we only use the POS tags. In the dataset, there are 62,010 sentences comprising 1,354,149 tokens.

Training

The model was trained using the Adam optimizer with a CrossEntropy loss function and a learning rate of 0.001. The word embedding layer was initialized with 100-dimensional embeddings, while the POS embedding layer was configured

with a dimensionality of 10 to represent the 34 Penn Treebank POS tags utilized. For the position embedding layer, relative position embeddings were applied to encode positional information, with a dimensionality of 110. Since sentence lengths varied, with a maximum of 86 and a minimum of 16, the maximum position was set to 64, which corresponds to the average sentence length in the training dataset. Sentences exceeding this length were truncated, while shorter ones were padded with zeros to reach the desired length.

The Transformer encoder consisted of two stacked identical layers. In the multi-head attention sublayer, the number of attention heads was set to 8, and the dimensionality of the input and output in the feed-forward sublayer was 110. The linear layer of the multi-head attention employed a softmax function for multi-class classification, with a hidden size of 48. The batch size was set to 32, and the Transformer's architecture included four layers. The input dimension was set to 23,698, representing the vocabulary size, and the output dimension was set to 34, corresponding to the number of classes. The dataset was shuffled and split into 80% for the training set and 20% for the test set used for validation.

Results

Our approach is evaluated using token accuracy, precision, recall, and the F1 score. Token accuracy measures the accuracy of predicted tags for individual tokens. Precision reflects the proportion of retrieved samples that are relevant and is computed as the ratio of correctly classified samples to the total samples assigned to a specific class [31]. Recall, also referred to as sensitivity or the true positive rate, measures the proportion of positive samples correctly identified. The F1 score is the harmonic mean of precision and recall, offering a balanced assessment of both metrics (Dalianis, 2018).

The evaluation metrics obtained from the Transformer model after training are summarized in **Table 2**.

Table 2. Classification Report Metrics

Accuracy	Precision	Recall	F1-Score
0.9850	0.92	0.89	0.90

The POS tagging model achieved an accuracy of 0.9850, a precision of 0.92, a recall of 0.89, and an F1 score of 0.90 prior to the application of the rule-based module, demonstrating the model's effectiveness. During training, the validation accuracy steadily improved from 0.7686 to 0.9850, while the loss consistently decreased from 0.4932 to 0.0101, as detailed in **Table 3**.

Table 3. Training and Loss

Epoch	Accuracy	Loss
1	0.7686	0.4932
80	0.9850	0.0101

Plotting the accuracy and loss over the number of epochs results in **Figure 4** and **5** respectively.

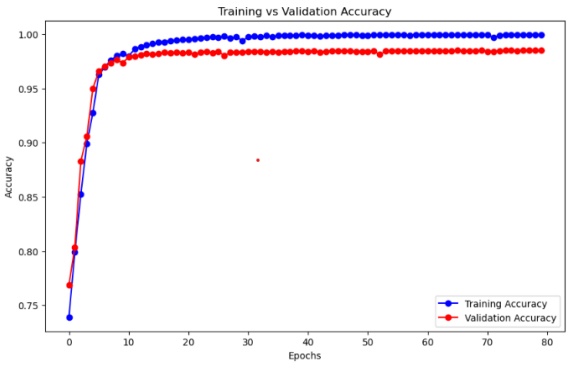

Figure 4. Model Accuracy Function

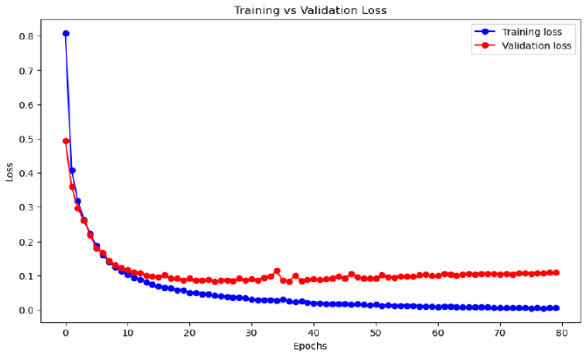

Figure 4. Model Loss Function

Table 3 shows the accuracy of the transformer deep learning model by Li et al., (2022) and our model.

Table 4. Models accuracy and positions encoding magnitude

Model	Transformer model	Rule-Base	Accuracy (%)	Positioning magnitude (%)
Model with absolute positioning encoding [3]	68%	30.6%	98.60	19.8
Our Model (relative positioning encoding)	98.5%	1.18%	99.68	33.44

The accuracy of the overall model is jointly determined by the transformer model and the rule-based module.

Evaluation and Comparison

For comparison, the study Part-of-Speech Tagging with Rule-Based Data Preprocessing and Transformer [3] employed absolute positional encoding within the attention head. In contrast, our approach utilizes relative positional encoding. Additionally, unlike [3], which applied a rule-based system for initial classification followed by masking certain tokens before passing the dataset to a transformer model, our method first processes tokens through the transformer model before applying the rule-based system. The masking approach in [3] led to the exclusion of masked tokens, thereby limiting the contextual information available to the transformer model.

In our method, processing tokens through the transformer model first allows for complete contextual information capture for each token and enables separate evaluation of the contributions of the transformer and the rule-based module, as presented in Table 4. The results indicate that the transformer model contributes 98.5% to the model's overall accuracy of 99.86%, while the rule-based module contributes an additional 1.18%. This sequence not only enhances the model's adaptability to unseen data but also fully leverages the capabilities of the transformer architecture.

Relative positional encoding proved to significantly enhance the self-attention mechanism, contributing 33.44% to its performance compared to the 19.8% contribution of absolute positional encoding. This improvement bolsters the attention mechanism's contextual understanding, thereby enhancing the overall model performance.

These findings underscore the advantages of our method in improving accuracy and contextual comprehension through relative positional encoding. By leveraging word embeddings, position embeddings, and POS embeddings in a unified framework, the proposed approach achieves superior performance.

4. Conclusion

This study highlights the effectiveness of incorporating relative positional encoding in transformer models for POS tagging tasks. By better capturing token dependencies within sentences, the proposed model achieves a precision, recall, and F1 score leading to an overall accuracy of 99.68%, outperforming models using absolute positional encoding, which achieve 98.60%. Integrating a rule-based module after the transformer further corrects misclassified tokens, enhancing the model's robustness. Relative positional encoding significantly amplifies the self-attention mechanism, improving the contextual representation and overall model performance. This novel approach, combining advanced transformer techniques with rule-based corrections, provides a robust solution for POS tagging in natural language processing tasks. Future work should explore automating rule extraction and extending the method's application to diverse languages for broader utility.

References:

[1] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2nd ed, 2019.

[2] S. G. Withanage and T. Silva, "A Stochastic Part of Speech Tagger for the Sinhala Language Based on Social Media Data Mining," in *Proceedings of the 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, Colombo, Sri Lanka, 2020, pp. 137-142, doi: 10.1109/ICTer51097.2020.9325456.

[3] H. Li, H. Mao, and J. Wang, "Part of Speech Tagging with Rule-Based Data Preprocessing and Transformer," *Electronics*, vol. 11, no. 56, 2022, doi: 10.3390/electronics11010056.

[4] K. S. Anbananthen, J. K. Krishnan, M. S. Sayeed, and P. Muniapan, "Comparison of Stochastic and Rule-Based POS Tagging on Malay Online Text," *American Journal of Applied Sciences*, vol. 14, no. 9, pp. 843-851, 2017, doi: 10.3844/ajassp.2017.843.851.

[5] A. Singh, C. Verma, S. Seal, and V. Singh, "Development of Part of Speech Tagger Using Deep Learning," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 1, 2019, doi: 10.35940/ijeat.A1531.109119.

[6] P. Lohe and V. Pandey, "Survey on Part of Speech Tagger for Hindi Language Using Rule-Based Approach," *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, no. 11, 2020.

[7] Chiche and Yitagesu, "Part of Speech Tagging: A Systematic Review of Deep Learning and Machine Learning Approaches," *Journal of Big Data*, vol. 9, no. 10, 2022, doi: 10.1186/s40537-022-00561-y.

[8] X. Yang, Y. Liu, D. Xie, X. Wang, and N. Balasubramanian, "Latent part-of-speech sequences for neural machine translation," 2019, doi: 10.48550/arXiv.1908.11782.

[9] Y. Tan, X. Wang, and T. Jia, "From syntactic structure to semantic relationship: Hypernym extraction from definitions by recurrent neural networks using the part of speech information," in *Proceedings of the 19th International Semantic Web Conference*, Athens, Greece, Nov. 2020, doi: 10.1007/978-3-030-62419-4_30.

[10] S. Warjri, P. Pakray, S. A. Lyngdoh, and A. K. Maji, "Part-of-Speech (POS) Tagging Using Deep Learning-Based Approaches on the Designed Khasi POS Corpus," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 3, 2022, doi: 10.1145/3488381.

[11] J. Awwalu, S. E. Abdullahi, and A. E. Evwiekpaefe, "Parts of Speech Tagging: A Review of Techniques," *FUDMA Journal of Sciences (FJS)*, vol. 4, no. 2, 2020, doi: 10.33003/fjs-2020-0402-325.

[12] B. Pham, "Parts of Speech Tagging: Rule-Based," 2020, doi: 10.3844/ajassp.2017.843.851.

[13] L. Galiano and A. Semeraro, "Part-of-Speech and Pragmatic Tagging of a Corpus of Film Dialogue: A Pilot Study," *Corpus Pragmatics*, vol. 7, pp. 17–39, 2023, doi: 10.1007/s41701-022-00132-9.

[14] S. Tyagi and G. S. Mishra, "Statistical Analysis of Part of Speech (POS) Tagging Algorithms for English Corpus," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 2, no. 3, 2016.

[15] G. Kaur and D. Sharma, "Development of Stochastic Part of Speech Tagger for Morphologically Rich Languages," *International Journal of Research in Engineering and Science (IJRES)*, vol. 9, no. 7, 2021.

[16] B. F. Shirko, "Part of Speech Tagging for Wolaita Language using Transformation-Based Learning (TBL) Approach," *International Journal of Engineering Science and Computing (IJESC)*, vol. 10, no. 9, 2020.

[17] P. Wang, Y. Qian, F. K. Soong, L. He, and H. Zhao, "Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network", 2015, doi: 10.48550/arXiv.1510.06168.

[18] W. Ling, T. Luís, L. Marujo, R. F. Astudillo, S. Amir, C. Dyer, A. W. Black, and I. Trancoso, "Finding function in form: Compositional character models for open vocabulary Word Representation," 2015, doi: 10.18653/v1/D15-1176.

[19] B. Plank, A. Søgaard, and Y. Goldberg, "Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss," 2016, doi: 10.18653/v1/P16-2067.

[20] H. Wang, J. Yang, and Y. Zhang, "From genesis to creole language: Transfer learning for Singlish universal dependencies parsing and POS tagging," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 18, no. 4, 2019, doi: 10.1145/3379142.

[21] P. Qi, T. Dozat, and C. Manning, "Stanford's Graph-Based Neural Dependency Parser at the CoNLL 2017 Shared Task," in *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Vancouver, BC, Canada, 2017, pp. 20-30. doi: 10.18653/v1/K17-3002

[22] P. Qi, T. Dozat, Y. Zhang, and C. D. Manning, "Universal dependency parsing from scratch," 2019, doi: 10.18653/v1/K18-2016.

[23] S. Warjri, P. Pakray, S. A. Lyngdoh, and A. K. Maji, "Part-of-speech (POS) tagging using Conditional Random Field (CRF) model for Khasi corpora," *International Journal of Speech Technology*, vol. 24, no. 3, pp. 415–423, 2021, doi: 10.1007/s10772-021-09854-6.

[24] W. AlKhwiter and N. Al-Twairesh, "Part-of-speech tagging for Arabic tweets using CRF and Bi-LSTM," *Computational Languages*, vol. 2021, no. 65, 2021, doi: 10.1016/j.csl.2020.101155.

[25] R. Dixit, "A Comprehensive Review of Transformer Models and Their Implementation in Machine Translation Specifically on Indian Regional Languages," *SSRN*, 2023.

[26] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021, doi: 10.1016/j.neucom.2021.03.091.

[27] S. R. Choi and M. Lee, "Transformer Architecture and Attention Mechanisms in Genome Data Analysis: A Comprehensive Review," *Biology (Basel)*, vol. 12, no. 7, p. 1033, Jul. 2023, doi: 10.3390/biology12071033.

[28] N. Patwardhan, S. Marrone, and C. Sansone, "Transformers in the Real World: A Survey on NLP Applications," *Information*, vol. 14, no. 4, p. 242, 2023, doi: 10.3390/info14040242.

[29] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A Survey of Transformers," *AI Open*, vol. 3, pp. 1-15, 2022, doi: 10.1016/j.aiopen.2022.10.001.

[30] J. Bos, V. Basile, K. Evang, N. J. Venhuizen, and J. Bjerva, "The Groningen Meaning Bank," in *Handbook of Linguistic Annotation*, N. Ide and J. Pustejovsky, Eds., Dordrecht, Netherlands: Springer, 2017, pp. 463–496, doi: 10.1007/978-94-024-0881-2_18.

[31] S. A. Hicks, I. Strümke, V. Thambawita, M. Hammou, M. A. Riegler, P. Halvorsen, and S. Parasa, "On Evaluation Metrics for Medical Applications of Artificial Intelligence," *Scientific Reports*, vol. 12, no. 1, 2022, doi: 10.1038/s41598-022-09954-8.

[32] H. Dalianis, "Evaluation Metrics and Evaluation," in *Clinical Text Mining*. Cham, Switzerland: Springer, 2018, pp. 6, doi: 10.1007/978-3-319-78503-5_6.