Indonesian Journal of Data and Science

Volume 6 Issue 2
ISSN 2715-9936
https://doi.org/10.56705/ijodas.v6i2.245

Research Article

Enhanced NER Using Relative Positional Encoding in Transformers

Jerome Aondongu Achir [1,*], Muhammad Abdulkarim [2], Mohammed Abdullahi [3]

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

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

Received 10 January 2025; Accepted 23 May 2025; Published 31 July 2025

Abstract:

Named Entity Recognition remains pivotal for structuring unstructured text, yet existing models face challenges with long-range dependencies, domain generalisation, and reliance on large, annotated datasets. To address these limitations, this paper introduces a hybrid architecture combining a transformer model enhanced with relative positional encoding and a rule-based refinement module to increase tokens' context-awareness as well as enhance the model's ability to discover token dependencies in long-range text. Relative positional encoding is applied to the attention head of the transformer model to improve contextual understanding by capturing the dynamic relationships between tokens, while rule-based post-processing corrects inconsistencies in entity tagging. After being evaluated on the Groningen Meaning Bank and Wikipedia Location datasets, the proposed model achieves state-of-the-art performance, with validation accuracies of 98.91% for Wikipedia and 98.50% for GMB with rule-based refinement, surpassing existing benchmark research of 94.0%. Notably, the system demonstrates a higher robustness in cross-domain tests compared to the purely data-driven baselines, reducing error propagation in nested entity recognition. The relative positional encoding contributes 34.42% to the attention mechanism's ability to identify and classify entities, underscoring its efficacy in modelling token interactions. Results demonstrate that integrating transformer-based architectures with rule-based corrections significantly enhances entity classification accuracy, particularly in complex and morphologically diverse contexts. This work highlights the potential of hybrid approaches to optimise sequence labelling tasks across domains while mitigating annotated data scarcity, through a semi-supervised rule-based system, and still provides an optimised system for entity classification.

Keywords: Named Entity Recognition; Transformer Model; Relative Positional Encoding; Rule-Based Fine-Tuning; Long-Range Dependencies; Hybrid Architecture; Contextual Understanding.

1. Introduction

Named Entity Recognition (NER) is a component of Natural Language Processing (NLP) that identifies and classifies entities [1]. It is an essential task in NLP aimed at the identification of entities like individuals, geographical locations, institutions, etc, in unstructured text [1], [2]. It enables the creation of structured knowledge that supports decision-making activities [3]. However, current NER models often depend on high-quality annotated diverse datasets [4], potential for bias amplification [2], [5] and difficulty in capturing long-range dependencies, which limits their capacity to generalise well across various domains, as well, their inability to capture contexts information with changes in complexity [6], [7]. To mitigate the aforementioned shortcomings, this research aims to overcome the limitations and provide a more robust NER tagging and classification system through improved transformer-based models using relative positional encoding in combination with rule-based fine-tuning, thus enhancing the accuracy and generalizability of NER systems. By modifying transformer architectures to incorporate a relative positional encoding

scheme for improved contextual understanding, integrating a rule-based refinement module to finetune entity tagging outcomes, the context awareness of the research model is greatly enhanced.

The paper discusses related works, introduces the proposed method, details the experimental setup, presents results, and concludes with key findings and future research directions.

Related Work

In this section, NER literature as well as Transformers models will be reviewed. This includes literature on tagging schemes, transformer models and NER-related work reviews.

Tagging Scheme

A common sequence tag of Begin-Inside-Outside (BIO), also referred to as IOB in other literature, is adopted [8]. The entities adopted also form the number of classes in the model, as captured in **Table 1**.

Table 1. Named entity classes

S/No	Entity tags	Description
1	O	Outside: The token is not relevant to the entities being identified.
2	B-GEO	Begin-Geographical Entity
3	I-GEO	Inside-Geographical Entity
4	B-GPE	Begin-Geopolitical Entity
5	I-GPE	Inside -Geopolitical Entity
6	B-PER	Begin-Person
7	I-PER	Inside -Person
8	B-ORG	Begin-Organization
9	I-ORG	Inside -Organization
10	B-TIM	Begin-Time
11	I-TIM	Inside -Time
12	B-ART	Begin-Artifact
13	I-ART	Inside -Artifact
14	B-NAT	Begin-Natural Object
15	I-NAT	Inside -Natural Object
16	B-EVE	Begin-Event
17	I-EVE	Inside -Event

Where **B** captures the beginning of the entity, **I** the inside and **O** refers to the unidentified entity respectively.

Transformer Model

Natural Language Processing (NLP) has changed significantly, especially after the introduction of the transformer model [9]. The model introduced a significant breakthrough in deep learning, constituting a serious difference from conventional architecture frameworks [10]. In contrast to standard sequence models that apply recurrent or convolutional structures, the transformer uses attention mechanisms, thereby establishing a new standard for NLP performance [11].

A transformer architecture basically consists of two components: a fully connected feedforward network and a multi-head self-attention mechanism [12]. The model uses an attention mechanism developed based on the Query–Key–Value (QKV) structure [12], [13] to amplify the self-attention mechanism to effectively identify the relationships between tokens in a sentence, regardless of the distance [14]. The attention mechanism receives a query and a list of key-value pairs and outputs, where all the queries, keys, values, and outputs are vector representations. The input

structure consists of queries and keys of dimension dk and values of dimension dv. It computes the dot products of the query with all keys, divides each by dk, and applies a SoftMax function to obtain the weights on the values [15] as outlined in Equation 1.

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

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

Rule-based tagging

Rule-based entity taggers utilise manually created linguistic rules to assign a tag to a token [14], [16]. For instance, if an entity precedes a token plc, it should be tagged as a company.

This approach significantly reduces the amount of information storage since knowledge is represented in the form of rules, easier to understand, and highly portable from one text corpus to another [17]. Despite its effectiveness, there are difficulties with the rule-based method, such as the necessity to understand linguistic background, inability to capture all linguistic rules [14], lack of flexibility in language model transformation and many more [13].

NER Tagging

Lample [18] introduced a neural network architecture combining bidirectional LSTM and CRF for NER. It uses character-level word representations via LSTM to improve entity detection. It outperformed traditional feature-based models on multiple datasets but struggles with rare or unseen entities. Liu [19] aims to improve NER performance in low-resource settings by leveraging adversarial training techniques. Traditional NER models rely heavily on large, annotated datasets, which are often unavailable for many languages and domains. This research explores how adversarial learning can enhance generalisation and robustness in low-resource environments. It significantly improved performance in low-resource settings, but was less effective on high-resource datasets. Devlin [6] introduced the Bidirectional Encoder Representations from Transformers (BERT) model, which is a transformer-based model that utilises bidirectional training of transformers to develop more informative contextual word representations by reading text from both directions at once. This method surpasses the previous unidirectional approaches, facilitating the comprehension of deeper linguistic context. Tested on tasks such as named entity recognition (NER), BERT obtained state-of-the-art performance after fine-tuning on the Conll-2003 dataset, outperforming previous models in terms of accuracy. Yet its efficacy relies on computationally costly pretraining with plentiful data, which is challenging to deploy in resource-scarce settings. Akbik [20] introduced contextual string embeddings for improved NER performance. It utilised character-level models trained bidirectionally to generate word embeddings, and it outperformed traditional word embeddings like Word2Vec and FastText on Conll-2003, but it fails with domain-specific datasets. Li [21] explored BERT-based models for biomedical NER by Fine-tuning BioBERT on multiple biomedical datasets. The research achieved good results in biomedical NER but requires large, labelled datasets and is computationally expensive. Lin [22] introduced a multi-task learning approach for cross-domain NER. A single model was trained for multiple NER tasks to improve generalisation. While the model generalised better than the traditional domain-specific models, it struggled with low-resource domains.

2. Method:

This section outlines the architecture and implementation of the proposed NER model. The model integrates a transformer-based encoder, enhanced with relative positional encoding to capture dynamic contextual relationships between tokens. A rule-based refinement module is also applied post-classification to correct tagging inconsistencies and improve accuracy. The proposed model is designed with a hybrid approach, integrating a transformer model with an attention mechanism using a relative positioning encoder with rule-based refinement to enhance NER performance. This architecture effectively captures contextual relationships between tokens while ensuring consistency in entity classification. This is illustrated in a simple architecture in **Figure 1**.

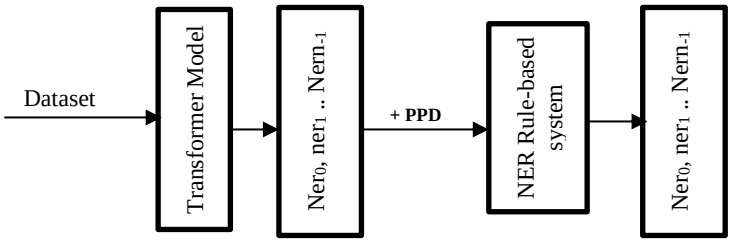

Figure 1. Architecture of NER model

As shown in **Figure 1**, the dataset comprising sentences, POS tags and entity tags forms the input to a transformer model, and the output is a trained model used to classify a set of entities into entity tags for each token. The entity tags and pre-processed dataset (PPD) are fed into the NER rule-based system, which is a refinement module for fine-tuning of entity tagging. The rule-based system refines the accuracy of the entity tagging system.

The transformer model as employed to achieve this objective is modified to consist of positional information encoding, transformer layer encoder, multi-head attention mechanism, and transformer block. It employs the relative positioning encoder against the absolute positioning to increase the availability of context information about the tokens in consideration. The algorithm of relative position encoding is captured in Algorithm 1.

Algorithm 1: pseudocode for relative positioning encoder.

```
# --- Relative Positional Encoding Module ---
# Addresses Challenge: Long-range dependencies via dynamic token relationships
def get_relative_position_matrix(seq_length):
    # Captures relative distances between tokens (i,j)
    i = np.arange(seq_length)
    j = np.arange(seq_length)
    return j - i[:, np.newaxis]  # Vectorized computation
# Addresses Challenge: Domain generalization via adaptive positional semantics
def relative_position_embedding(relative_positions, embedding_dim):
    seq_length = relative_positions.shape[0]
    max_relative_pos = 2 * seq_length - 1  # Covers all possible relative offsets
    embedding_table = np.random.rand(max_relative_pos, embedding_dim)  # Learnable embeddings
    shifted_positions = relative_positions + seq_length - 1  # Shift to non-negative indices
    return embedding_table[shifted_positions]  # Direct indexing replaces nested loops
# Addresses Challenge: Contextual ambiguity via token-position interaction
def apply_relative_position_embedding(token_embeddings, relative_embeddings):
    # Fuses token and positional information (additive attention mechanism)
    return np.sum(token_embeddings[:, np.newaxis, :] + relative_embeddings, axis=1)
# --- Rule-Based Refinement Module ---
# Addresses Challenge: Annotation scarcity via syntactic/lexical constraints
def rule_based_refinement(entity_tags, text):
    # Example correction: Resolve ambiguous LOC/ORG conflicts using preposition rules
    for i in range(1, len(entity_tags)):
        if entity_tags[i] == "LOC" and text[i-1] in ["near", "from"]:
            entity_tags[i] = "GPE"  # Geographic entity heuristic
    return entity_tags
```

The algorithm enhances named entity recognition by combining two core components. The relative positional encoding module uses get_relative_position_matrix to calculate how far apart words are in a sentence, relative_position_embedding to convert these distances into meaningful numerical patterns, and apply_relative_position_embedding to blend these patterns with the words' original meanings, thereby helping the system understand relationships between distant or complex terms. The rule-based

refinement module then applies rule_based_refinement, using simple language rules to correct mislabeled entities, reducing dependence on large labelled datasets. Put together, these steps address challenges like interpreting long sentences, adapting to diverse contexts, and working with limited training data.

The algorithm of the NER model is captured in Algorithm 2.

Algorithm 2: Pseudocode for the NER module.

```
Accept sentences, POS tags, and entity tags as inputs.
Build vocabularies for words, POS tags, and entity tags.
Encode and pad sequences.
Optionally augment data with entity swapping.
Methods:
    build_vocab(data): Build vocabulary for input data.
    encode_and_pad_data(data, vocab): Encode and pad sequences.
    augment_with_entity_swapping(sentences, entity_tags): Swap entities of the same type
        within sentences for augmentation.
MultiHeadAttention:
    Implements scaled dot-product attention.
    Projects input to query, key, and value tensors.
    Computes attention weights and outputs.
TransformerEncoderLayer:
    Composed of multi-head attention and feed-forward layers.
    Uses LayerNorm and Dropout for stability.
TransformerEncoder:
    Stack multiple encoder layers to process input.
Combines embedding layers for words and POS tags.
Includes a transformer encoder and CRF for sequence labeling..
```

Here, the algorithm accepts sentences, POS tags, and entity tags as inputs, then builds vocabularies and encodes sequences for NER. It constructs vocabularies for words, POS tags, and entity tags, converts input sequences into numerical representations and ensures uniform length with padding. Thereafter, it enhances training data by swapping entities of the same type within sentences and goes ahead to implement scaled dot-product attention by projecting inputs into query, key, and value tensors and computing attention weights. A transformer layer combines multi-head attention, feed-forward networks, LayerNorm, and Dropout for stable training, while transformerencoder stacks multiple encoder layers, integrates word and POS embeddings, and incorporates a conditional random field (CRF) for sequence labelling tasks.

The research brings to the fore an implementation of relative positional encoding in transformers for NER tagging as well as a hybrid architecture combining transformer predictions with rule-based corrections to refine transformer model outputs and address inconsistencies. Rule-based corrections are applied after the Transformer's predictions.

- Consistency Rules: These rules enforce valid sequence tagging by ensuring that a "B-X" (beginning) tag is always followed by an "I-X" (inside) tag, preventing errors where entity labels are incorrectly split.
- Keyword and Prefix/Suffix Rules: Specific patterns are used to improve classification accuracy. For instance, recognising "Ltd." as an indicator of an organisation ("ORG"), assigning "Dr." to the person ("PER") category.

Each refinement in the rule-based approach is added to the correct_predict of the transformer model that holds a total of correct predictions [23, 24]. In the end, the overall accuracy was evaluated using in Equation 2.

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

Dataset

Two diverse and well-established datasets were utilised to evaluate the effectiveness of the proposed Named Entity Recognition (NER) and relationship extraction models: the GMB and the Wikipedia Location Dataset. These datasets provide a comprehensive benchmark for assessing the model's ability to recognise, classify, and extract entities across different domains. The GMB dataset consists of 47,958 sentences, annotated across 17 distinct entity classes, making it a rich resource for training and evaluating NER models [25]. On the other hand, the Wikipedia Location Dataset is a large-scale collection comprising 42,222 articles, specifically focused on geo-political entities, such as countries, cities, regions, and landmarks [1].

Training

The model is trained using the Adam optimiser with a CrossEntropy loss, a learning rate of 0.001, word embedding layer is initialised to 100-dimensional embeddings. The Transformer encoder consists of two stacked layers, each with eight attention heads in the multi-head attention sublayer. The feed-forward sublayer is set to an input and output dimension of 256, while the softmax layer in the multi-head attention has a hidden size of 48 for multiclass classification. The batch size is set to 32, and the Transformer model has three layers. The input dimension is 23,698 (vocabulary size), while the output dimension is 17, representing the number of classes.

3. Results and Discussion

Results

The model is evaluated using accuracy, precision, recall, and the F1 score. Accuracy measures the correctness of predicted tags for individual tokens [26],[27]. Precision is computed as the ratio of correctly classified data to the total dataset assigned to a specific class [28]. Recall measures the proportion of positive samples correctly identified, while the F1 score is the harmonic mean of precision and recall [29],[30].

The evaluation metrics obtained from the model after training and evaluation are summarised in **Table 2**.

Table 2. Model Accuracy

Dataset	Training Accuracy	Validation Accuracy
Wikipedia	0.9910	0.9891
GMB	0.9698	0.9606

Table 3. Model Loss

Dataset	Training Loss	Validation Loss
Wikipedia	0.5147	0.6440
GMB	1.3147	1.8450

The model demonstrates high accuracy across both training and validation datasets, indicating its strong generalisation capability.

As captured in **Table 2**, the model achieves a training accuracy of 99.10% and a validation accuracy of 98.91%, suggesting that it learns effectively from Wikipedia data with minimal overfitting. Table 3 captures the training loss of 0.5147 and validation loss of 0.6440 for the same Wikipedia dataset, showing a relatively low loss, thereby reinforcing that the model converged well.

The GMB dataset results are slightly lower, and this can be attributed to the complexity of the dataset compared to Wikipedia. It hit a training accuracy of 96.98% and validation accuracy of 96.06% with training loss of 1.3147 and validation loss of 1.8450, respectively.

To enhance performance on the general dataset (GMB dataset), a rule-based module was applied and accuracy computed using Equation 2, leading to an accuracy improvement to 98.50%, demonstrating that integrating rule-based refinements with the Transformer model can significantly boost performance in challenging datasets

Comparing the existing model

Table 4 captures the accuracy, f1-score, precision and recall and positional encoding ntion of both the existing and newly developed model.

Table 4. Model accuracy and positions encoding magnitude

Model	Accuracy (%)	F1-Score	Precision	Recall	Position Encoder Magnitude (%)
Berragan *et al.*, (2023)	94.0	0.92	0.90	0.94	18.5%
Proposed Model with Wikipedia dataset	98.91	0.95	0.96	0.95	34.42
Proposed Model with GMB dataset	98.5	0.90	0.92	0.89	34.42

The accuracies of the models are visualised in **Figure 2**.

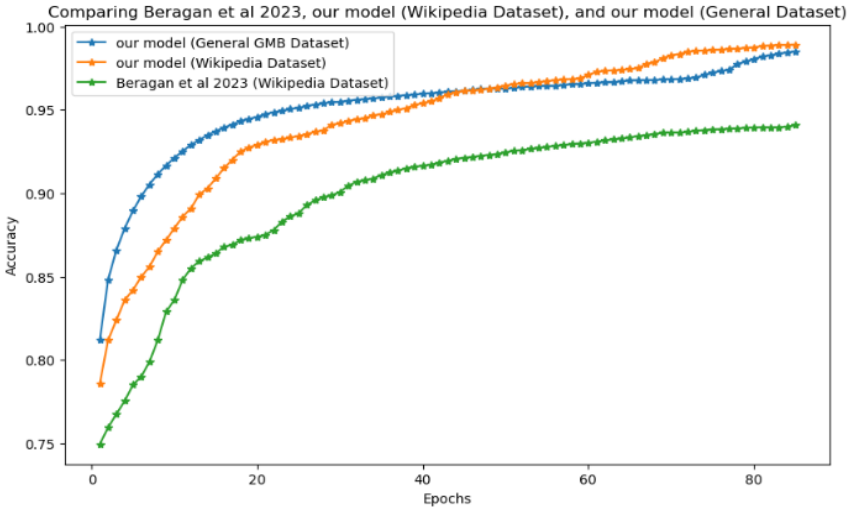

Figure 2. Accuracies of the models

Table 4 presents a comparison of the proposed transformer-based model's performance on the Wikipedia and GMB datasets against the benchmark model developed by Berragan [1], based on accuracy, F1-score, precision, and recall. The benchmark model achieved an accuracy of 94.0%, with an F1-score of 0.92, precision of 0.90, and recall of 0.94. While these metrics indicate strong performance, they are lower compared to those achieved by the proposed model on the same Wikipedia dataset. The proposed model trained on the Wikipedia dataset marked an accuracy of 98.91%, F1-score of 0.95, precision of 0.96, and recall of 0.95, significantly outperforming the base model. On the one hand, the proposed model trained on the GMB dataset achieved 96.06% accuracy, slightly lower than its performance on Wikipedia but still superior to the base model. The superior performance of the model on the Wikipedia dataset is largely attributed to its domain homogeneity and less context complexity, which contrasts with the more linguistically diverse and informal GMB dataset. The F1-score of 0.90, precision of 0.92, and recall of 0.89 suggest a slight drop in recall, due to the greater variability and complexity of the GMB dataset.

Overall, the proposed model outperforms the base model across all evaluation metrics, demonstrating the effectiveness of transformer-based architectures with relative position encoding. The application of a rule-based enhancement significantly improved the GMB dataset results, as the rules enhance entity classification, most especially the person entity, thereby boosting accuracy to 98.50%. This suggests that integrating rule-based refinements with deep learning models can further optimise sequence labelling performance.

Table 4 also highlights the relative position scores of 34.42% contribution to the attention mechanism's capability for context-aware computation compared to the 18.5% contribution from absolute positioning. This substantial impact highlights the importance of relative positioning in enhancing context understanding.

4. Conclusion

The proposed hybrid model, integrating relative positional encoding in transformers with rule-based refinements, outperforms the existing benchmark model of Berragan [1], achieving 98.91% accuracy and 0.95 F1-score on the Wikipedia dataset. The more complex GMB dataset with rule-based refinement elevated accuracy to 98.50%, underscoring its value in handling variability. Relative positional encoding significantly enhanced context capture, contributing 34.42% to attention mechanism context-aware capability versus 18.5% from absolute positioning. The hybrid architecture ensured tagging consistency by enforcing valid BIO sequences and leveraging linguistic patterns, thereby demonstrating the robustness of the model. This approach highlights the efficacy of combining deep learning with rule-based systems to address challenges in NER, particularly in complex or low-resource domains. Future work should apply multilingual dataset and refine domain-specific morphological handling. Despite strong performance, the model's reliance on rule-based refinement may limit adaptability to domains with highly irregular linguistic patterns or where such rules are harder to define.

References:

[1] C. Berragan, A. Singleton, A. Calafiore, and J. Morley, "Transformer based named entity recognition for place name extraction from unstructured text," *International Journal of Geographical Information Science*, vol. 37, no. 4, pp. 747–766, 2022, doi: 10.1080/13658816.2022.2133125.

[2] P. Basile, A. Caputo, and G. Semeraro, "An enhanced Lesk word sense disambiguation algorithm through a distributional semantic model," in *Proceedings of 24th International Conference on Computational Linguistics (COLING)*, 2014, pp. 1591–1606.

[3] K. Pakhale, "Comprehensive overview of named entity recognition: models, domain-specific applications and challenges," *arXiv preprint arXiv:2309.14084v1*, Sep. 2023.

[4] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proceedings of 57th Annual Meeting of the Association of Computational Linguistics (ACL)*, 2019, pp. 2978–2988, doi: 10.18653/v1/P19-1285.

[5] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT): Human Language Technologies (NAACL-HLT)*, 2018, pp. 464–468, doi: 10.18653/v1/N18-2074.

[6] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," ," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2019, pp. 4171–4186, doi: 10.48550/arXiv.1810.04805.

[7] B. Y. Lin, C. Tan, Y. Ji, and X. Ren, "RockNER: A simple method to create adversarial examples for evaluating the robustness of named entity recognition models," in *Findings of the Association for Computational Linguistics: EMNLP*, 2021, pp. 3729–3744, doi: 10.18653/v1/2021.emnlp-main.302.

[8] E. T. K. Sang and S. Buchholz, "Introduction to the CONLL-2000 shared task: Chunking," in *Proceedings of the. 4th Conference on Computational Natural Language Learning (CONLL/LLL 2000)*, Lisbon, Portugal, Sep. 2000, pp. 127–132, doi: 10.3115/1117601.1117631.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] A. Chiche and Y. 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.

[15] A. M. Rush. The Annotated Transformer. In Proceedings of Workshop for NLP Open-Source Software, pages 52–60, 2018. Melbourne, Australia, doi: 10.18653/v1/W18-2509.

[16] T. Dalai, T. K. Mishra, and P. K. Sa, "Deep learning-based POS tagger and chunker for Odia language using pre-trained transformers," *Association of Computing Machinery*, vol. 23, no. 2, 2024. doi: 10.1145/3637877.

[17] S. E. Abdulkareem, M. Abdullahi, and A. E. Evwiekpaefe, “Parts of speech tagging: A review of techniques,” *FUDMA Journal of Science (FJS)*, vol. 4, no. 2, 2020, doi: 10.33003/fjs-2020-0402-325.

[18] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural architectures for named entity recognition,” Transactions of the Association for Computational Linguistics, vol. 4, pp. 1–17, 2018, doi: 10.1162/tacl_a_00001.

[19] Y. Liu, W. Li, X. Zheng, and X. Sun, “Adversarial training for low-resource named entity recognition,” in *Proceedings of the* Association for Computational Linguistics, 2019, pp. 2171–2181, doi: 10.18653/v1/P19-1210.

[20] A. Akbik, D. Blythe, and R. Vollgraf, “Contextual string embeddings for sequence labelling,” in *Proceedings of 27th International Conference on Computational Linguistics (COLING)*, 2018, pp. 1638–1649.

[21] F. Li, Y. Jin, W. Liu, B. P. S. Rawat, P. Cai, and H. Yu, “Fine-tuning bidirectional encoder representations from transformers (BERT) improves biomedical named entity recognition,” *Bioinformatics*, vol. 36, no. 15, pp. 4236–4242, 2020, doi: 10.1093/bioinformatics/btaa375.

[22] H. Lin, Z. Lu, X. Han, and M. Sun, “A multi-task learning framework for cross-domain named entity recognition,” in *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, doi: 10.1609/aaai.v34i05.6509.

[23] M. Abdulkarim, M. Abdullahi, and J. A. Achir, “Improving part-of-speech tagging with relative positional encoding in transformer models and basic rules,” *Indonesian J. Data Sci.*, vol. 6, no. 1, 2025, doi: 10.56705/ijodas.v6i2.184.

[24] P. Basile, J. Bos, K. Evang, and N. Venhuizen, “Groningen Meaning Bank” [Dataset]. University of Groningen, 2012. https://gmb.let.rug.nl/.

[25] 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,” *Sci. Rep.*, vol. 12, no. 1, 2022, doi: 10.1038/s41598-022-09954-8.

[26] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008.

[27] D. M. W. Powers, “Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation,” *Journal of Machine Learning Technologies*, vol. 2, pp. 37–63, 2011, doi: 10.9735/2229-3981.

[28] J. Juba and H. S. Le, "Precision-Recall versus Accuracy and the Role of Large Data Sets," in *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)* vol. 33, no. 01, pp. 4039–4048, 2019, doi: 10.1609/aaai.v33i01.33014039.

[29] Z. D. Vujovic, "Classification Model Evaluation Metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021, doi: 10.14569/IJACSA.2021.0120670.

[30] 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, p. 5979, 2022, doi: 10.1038/s41598-022-09954-8