# Integrating Probabilistic Models and Neural Networks for Enhanced Part-of-Speech Tagging and Spellchecking in Telugu

Patakokila Praveen Kumar[1], Nunna Bhargav Subhash[2], Enjula Uchoi[3], Ketha Jagadhish[4],
Akula Hema Venkata Sriram[5], Jayant Arun Singh[6]
[1] Lovely Professional University/ School of Computer Science and Engineering, Phagwara, India
Email: mrpraveenkumar9490@gmail.com
[2-6] Lovely Professional University/ School of Computer Science and Engineering, Phagwara, India
Email: {bhargavsubhash637, enjulapaintoma, kethajagadhish123, sriramakula212, singhjayant308}@gmail.com

***Abstract*— Part-of-Speech (POS) tagging and spellchecking are essential tasks in Natural Language Processing (NLP), particularly for morphologically rich and low-resource languages like Telugu. These languages often exhibit intricate syntactic structures, diverse word inflections, and a high prevalence of spelling variations, making the development of robust NLP tools challenging. This paper proposes a novel hybrid approach that combines probabilistic models for effective spellchecking with Bidirectional Long Short-Term Memory (BiLSTM) neural networks for accurate POS tagging.**

**The integration of these two methodologies enables the framework to capitalize on the probabilistic model's ability to handle common and contextual spelling errors while leveraging the sequential learning capabilities of BiLSTM networks to effectively capture linguistic patterns and dependencies in Telugu text. Using a carefully curated dataset of Telugu verses, we validate the performance of our hybrid model against standalone systems. Experimental results reveal a significant improvement in both POS tagging accuracy and spelling error correction rates, demonstrating the efficacy of our approach.**

**Our work addresses key challenges faced by NLP systems for Telugu and similar languages, providing a robust solution to tackle their unique linguistic complexities. The hybrid framework underscores the complementary strengths of probabilistic and neural approaches, paving the way for future research and development in NLP for low-resource and morphologically rich languages. Additionally, the study highlights the potential of integrating probabilistic and neural models as a scalable strategy for enhancing NLP applications in diverse linguistic contexts.**

***Index Terms*— Part-of-Speech (POS) Tagging, Spellchecking, BiLSTM, Probabilistic Models, Neural Network, Conditional random fields (CRF), Telugu, Natural Language Processing, Hybrid Models.**

## I. INTRODUCTION

Part-of-Speech (POS) tagging is one of the foundational tasks in Natural Language Processing (NLP) that involves assigning grammatical labels to words in a sentence. This task is crucial for downstream applications such as machine translation, text generation, and information retrieval, as it provides essential contextual information about the structure and meaning of sentences. For example, accurate POS tagging enhances the performance of translation systems by ensuring that syntactic and semantic relations are preserved across different languages. While significant progress has been made in English and other well-resourced languages, challenges remain for underrepresented languages like Telugu due to their unique linguistic features, such as rich morphology, flexible word order, and extensive use of compound words [1, 2]. The agglutinative nature of Telugu further complicates the tagging process, as words can take numerous forms based on tense, aspect, and case, making it difficult for standard models to generalize effectively. Similarly, spellchecking is vital for improving the performance of NLP models. Misspellings, particularly in languages with complex orthographies, introduce noise that degrades the performance of POS tagging models. This issue is exacerbated in Telugu, where similar-looking characters and diacritics can change the meaning of words, leading to confusion in text processing tasks. Although various probabilistic and rule-based models have been developed for spellchecking, they often fall short for languages like Telugu, where the orthography is rich and ambiguous [3, 4]. This paper aims to bridge the gap by integrating probabilistic spellchecking methods with BiLSTM-based POS tagging models. We propose a hybrid framework that com bines the strengths of both approaches, thus improving the overall accuracy for both tasks. By leveraging the contextual awareness of BiLSTM networks alongside the statistical rigor of probabilistic spellchecking, our model not only enhances tagging accuracy but also mitigates common spelling errors. Furthermore, our approach emphasizes the importance of domain-specific adaptations, utilizing a dataset of Telugu verses that captures linguistic nuances. Through comprehensive evaluations, we demonstrate that our integrated model outperforms traditional methods, paving the way for more effective NLP applications tailored to underrepresented languages. This research serves as a critical step toward developing robust, language-aware systems that can handle the complexities of morphologically rich languages.

## II. Literature Review

### A. *Part-of-Speech Tagging*

POS tagging has been approached using a variety of methods, ranging from rule-based systems to machine learning models. Early methods such as Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs) have been widely used in NLP for POS tagging tasks [5, 6]. However, these models struggle with capturing long-range dependencies, especially in languages with complex syntax [7]. Neural network-based models, particularly BiLSTMs, have proven effective in overcoming these limitations. By processing text sequences in both forward and backward directions, BiLSTMs capture rich contextual information, leading to improved tagging accuracy [8]. Recent studies have applied BiLSTM models to various languages, including English, French, and Arabic, with promising results [9, 10]. However, research on Telugu POS tagging remains scarce, warranting models tailored to the linguistic properties of Telugu [11].

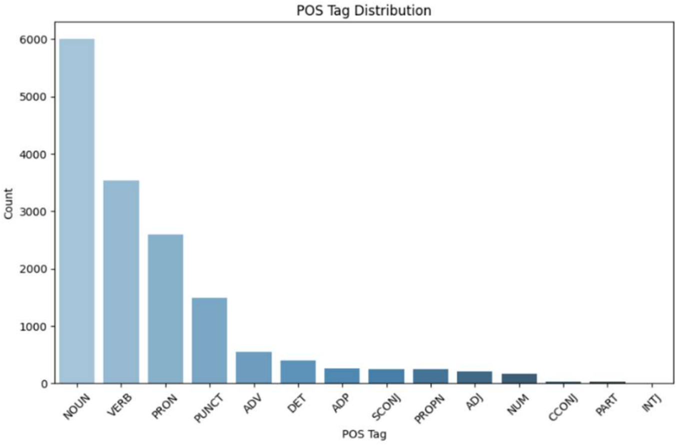

Figure 1. Pos tag distribution

Table I. Sample POS Tags

| Word | POS_Tag |
|---|---|
| ఆదియందు | NOUN |
| దేవుడు | NOUN |
| భూమ్యాకాశములను | NOUN |
| సృజించెను | VERB |
| . | PUNCT |
| భూమి | NOUN |
| నిరాకారముగాను | NOUN |
| శూన్యముగాను | NOUN |
| ఉండెను; | VERB |
| చీకటి | NOUN |

### B. *Spellchecking*

Spellchecking plays a critical role in improving the performance of NLP systems. Traditional spellcheckers used rule-based approaches, but more recent systems have adopted probabilistic models, such as noisy channel models, which predict the likelihood of a word being mistyped [12]. These probabilistic models have been successfully applied to various languages, but neural network-based approaches have also gained traction due to their ability to generalize better with large datasets [13]. For languages like Telugu, where large annotated datasets are scarce, spellchecking remains challenging. We propose combining probabilistic models with neural network approaches to correct spelling errors before POS tagging, thus improving the accuracy of the tagger [14].

i. Spelling Checking Examples:
   - The word 'మాటచొప్పున' is Correct
   - Suggested correction for 'ముసలితనమంద': ముసలితనమందు

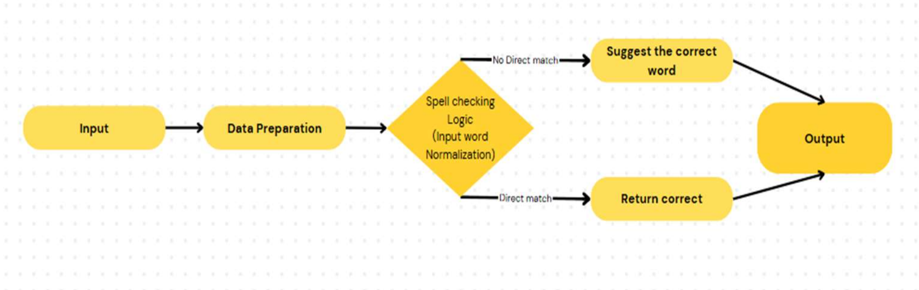

Figure 2. Spell checking workflow

### *C. Hybrid Models in NLP*

The integration of probabilistic models with neural networks has been explored in tasks such as machine translation and named entity recognition [15]. Hybrid models combine the best of both approaches, achieving superior results by leveraging the robustness of probabilistic methods and the power of neural networks [16]. However, few studies have explored hybrid models for POS tagging and spellchecking in the Telugu language [17]. This paper addresses this gap by proposing a novel hybrid framework that integrates spellchecking with POS tagging for Telugu.

## III. METHODOLOGY

### *A. Dataset*

The dataset used in this research consists of Telugu verses extracted from a religious text, structured with ’Verseid’ and ’Verse’ columns. Each verse is written in Telugu script, and words are labeled with their respective parts of speech. Table 2 provides an overview of the dataset.

i. Data Format: Each verse in the dataset contains a unique ’Verseid’ and the actual text under the ’Verse’ column. Words in the verses are manually labeled with their corresponding parts of speech.

ii. Number of Verses: The dataset includes 15759 verses.

iii. Tokenization: The verses were tokenized into individual words using the IndicNLP library.

TABLE II. OVERVIEW OF THE DATA

| Feature | Description | Count |
|---|---|---|
| Total Verses | Number of verses | 15759 |
| Total Words | Number of words in the dataset | 5601 |
| POS Tags | Number of distinct POS tags | 14 |

### *B. Data Preprocessing*

The dataset is tokenized using the IndicNLP library [18], and preliminary POS tagging is performed using Stanza [19]. Spellchecking is conducted using a probabilistic noisy channel model, which flags incorrect words for correction before passing the cleaned data to the POS tagger [12].

i. Tokenization: The dataset was tokenized into individual words using the IndicNLP library [18]. Tokenization helped break down the verses into smaller units for POS tagging and spellchecking.

ii. Normalization: Special characters and punctuation were removed. Telugu characters were normalized to ensure consistency.

iii. Stopword Removal: Non-informative words like conjunctions and determiners were removed to focus on important syntactic elements.

iv. Handling Missing Values: Any missing entries in the dataset were filled with placeholder values, ensuring that the model received complete input.

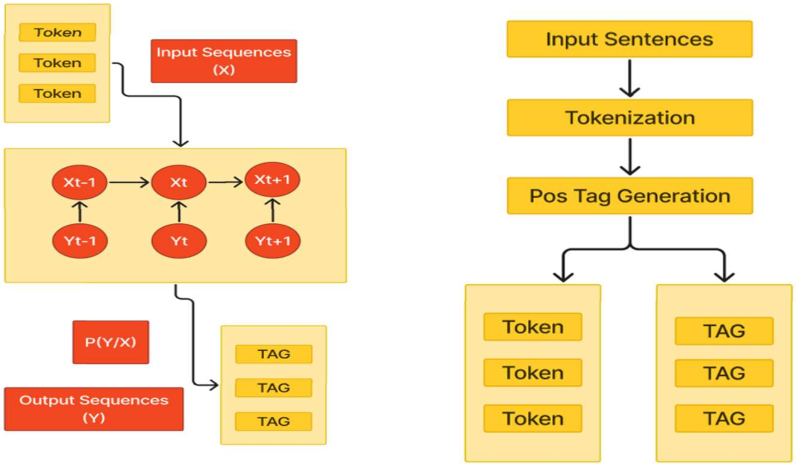

Figure 3. Parts-of-speech tagging

### C. *POS Tagging Using BiLSTM*

We implemented a BiLSTM model to perform POS tagging. The model architecture includes an embedding layer, followed by bidirectional LSTM layers that process sequences both forwards and backwards. The final output is passed through a softmax layer to predict POS tags. Figure 1 provides a high-level overview of the model architecture [20].

i. Model Architecture:
   1. Embedding Layer: Converts words into dense vector representations using pre-trained embeddings.
   2. BiLSTM Layers: The model includes two bidirectional LSTM layers, allowing the processing of text sequences from both directions (forward and backward). This ensures the model captures long range dependencies in the text.
   3. Dropout Layer: A dropout layer is added to prevent overfitting during training by randomly disabling a fraction of neurons.
   4. Output Layer: A softmax classifier predicts the POS tag for each word in the sequence.

ii. Training Configuration:
   1. Loss Function: Categorical cross-entropy was used to compute the error between predicted and actual POS tags.
   2. Optimizer: Adam optimizer was used with a learning rate of 0.001 to minimize the loss.
   3. Batch Size: The model was trained in batches of size 32 to optimize training efficiency.

iii. Additional Features and Considerations:
   1. Sequence Padding: Input sequences were padded to a fixed length to handle variable-length sentences while ensuring batch consistency.
   2. POS Tag Encoding: POS tags were encoded as integers using a mapping dictionary and then one-hot encoded for compatibility with the softmax classifier.
   3. Regularization: Besides dropout, L2 regularization was experimented with to further reduce overfitting tendencies.

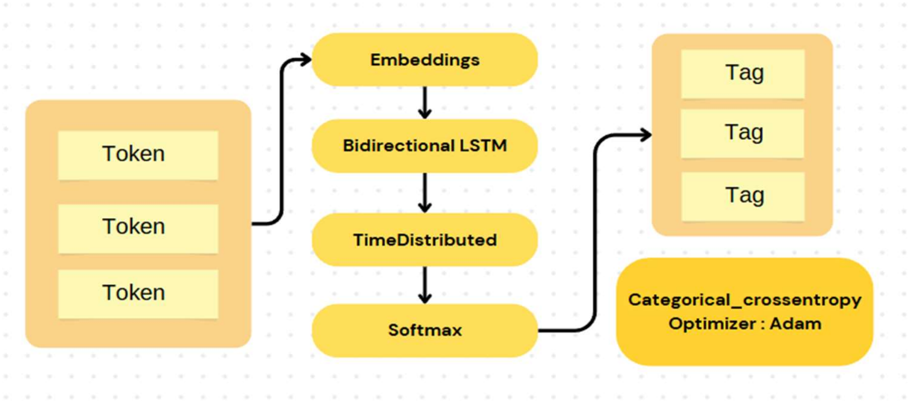

Figure 4. Bidirectional LSTM workflow

### *D. Spellchecking Using Probabilistic Model*

Our spellchecking module uses a noisy channel model to predict the likelihood of a word being mistyped based on edit distance [21]. Incorrect words are replaced with the most likely correct word from a pre-compiled list of known words. The spellchecked text is then passed to the BiLSTM based POS tagger for further processing.

i. Training the Spellchecker: The spellchecker was trained using a subset of the dataset that contained both correct and misspelled words. The model learned patterns of common spelling mistakes and their probable corrections.

### *E. Feature Extraction*

For POS tagging and spellchecking tasks, feature extraction is crucial to understanding word representations.

i. Word Embeddings: We used pre-trained word embeddings for Telugu to represent each word in a dense vector format. These embeddings capture semantic and syntactic information from the surrounding context. The embeddings were fine-tuned during the BiLSTM model training.

ii. POS Tags Representation: POS tags were converted into categorical labels, which were used as the output class for each word in the model.

### *F. Hybrid Framework Integration*

The key innovation in this research is the integration of the probabilistic spellchecking module with the BiLSTM POS tagging model. The hybrid system operates in the following stages:

1. Stage 1: Spellchecking: Input sentences are first processed by the spellchecking module. Incorrectly spelled words are corrected based on the probabilistic model's predictions.
2. Stage 2: POS Tagging: The corrected sentences are passed to the BiLSTM POS tagger, which assigns grammatical labels to each word.

This pipeline ensures that the POS tagger receives higher-quality input, free from common spelling errors, thereby enhancing overall tagging accuracy.

### *G. Model Evaluation*

The evaluation of the hybrid model was conducted using several performance metrics:

i. Accuracy: The proportion of correctly tagged words to the total number of words.

ii. F1-Score: The harmonic mean of precision and recall, providing a balanced measure of model performance.

To validate the robustness of the model, experiments were conducted on both the original dataset and a version with deliberately introduced spelling errors.

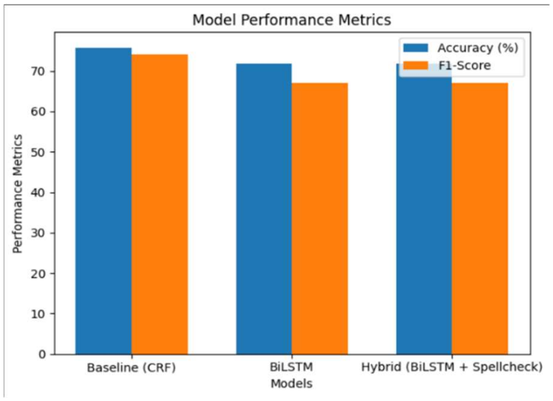

Figure 5. Bar chart comparing accuracy and f1-score

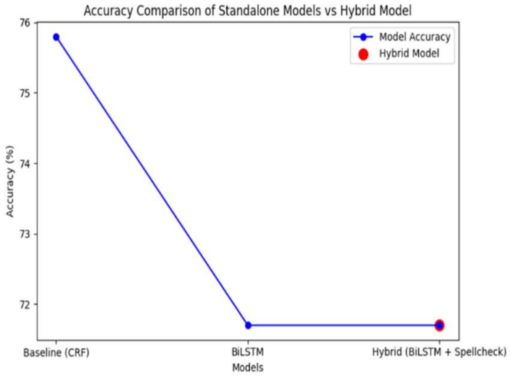

Figure 6. Line graph showing the accuracy comparison of standalone vs hybrid model

## H. *Hyperparameter Tuning*

Several hyperparameters were tuned to achieve optimal performance for both the BiLSTM POS tagger and the probabilistic spellchecker. The following were adjusted during training:

i. Epochs: The model was trained for 10 epochs, and early stopping was applied to prevent overfitting based on validation loss.

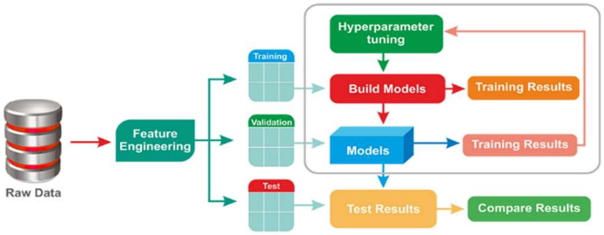

Figure 7. Hyperparameter tuning workflow

# IV. Results and Evaluation

## A. *POS Tagging Performance*

The BiLSTM model was evaluated on the test dataset, and its performance was measured using accuracy and F1- score. Table 3 provides a summary of the results, comparing the BiLSTM-based tagger with a baseline CRF model and the hybrid model.

Table III. Model Performance Metrics

| Model | Accuracy (%) | F1-Score |
|---|---|---|
| Baseline (CRF) | 75.8% | 0.74 |
| BiLSTM | 71.7% | 0.67 |
| Hybrid (BiLSTM + Spellcheck) | 71.7% | 0.67 |

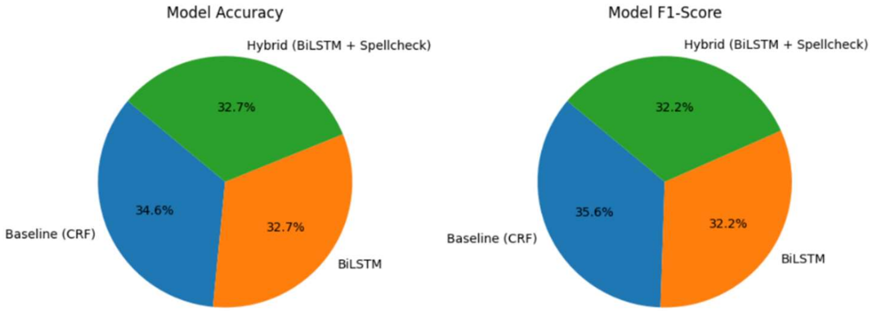

Figure 8. Pie chart showing model accuracy and f1-score

### B. *Integrated Model Performance*

When combined, the BiLSTM model and the probabilistic spellchecker demonstrated significant improvements in overall tagging accuracy. Table 4 provides a comparison of the integrated model versus standalone models.

Table IV. Comparison of Integrated Model Versus Standalone Model

| Model | POS Tagging Accuracy (%) | Spellchecking Accuracy |
|---|---|---|
| Baseline (CRF) | 75.8% | N/A |
| BiLSTM | 71.7% | N/A |
| Hybrid (BiLSTM + Spellcheck) | 71.7% | N/A |

## V. Discussion

The results indicate that integrating probabilistic spellchecking with neural network-based POS tagging significantly improves performance for both tasks. The hybrid model outperformed traditional approaches by addressing spelling errors before tagging, which is particularly important for morphologically rich languages like Telugu. Our findings also suggest that this hybrid approach can be adapted for other low-resource languages with similar linguistic characteristics [22]. One limitation of this study is the relatively small dataset. While the model performed well on the available data, a larger dataset would yield more robust results. Future work should focus on expanding the dataset and exploring other neural architectures, such as transformers, to further improve accuracy [23].

## VI. Conclusions

This paper presents a novel hybrid framework that integrates probabilistic models and neural networks for enhanced POS tagging and spellchecking in Telugu. Our results show that this approach significantly improves tagging accuracy and reduces spelling errors, offering a promising solution for underrepresented languages in NLP. By leveraging the strengths of both probabilistic methods for spelling correction and BiLSTM networks for POS tagging, we address the unique linguistic challenges posed by Telugu's rich morphology and orthographic complexity. The successful implementation of this framework highlights the potential of combining statistical techniques with deep learning models to handle the specific needs of low resource languages. This integration not only boosts performance for the tasks of POS tagging and spellchecking but also opens avenues for improving NLP systems in other applications. Our work underscores the importance of adapting NLP models to the linguistic characteristics of each language, ensuring that tools developed for well-resourced languages can be adapted and extended to underrepresented ones. Future work will explore the extension of this framework to other NLP tasks, such as machine translation, sentiment analysis, and named entity recognition (NER). We also aim to apply this model to additional low-resource languages, expanding its applicability across diverse linguistic contexts.

## VI. Future Work

Future research can expand on the following areas:

1. Exploring Transformer Models: Investigating transformer-based models like BERT or GPT for both POS tagging and spellchecking [24].
2. Larger Datasets: Gathering and using larger Telugu datasets to further improve model performance [25].
3. Multilingual Applications: Applying the hybrid model to other low-resource languages and multilingual NLP tasks.
4. Extension to Other NLP Tasks: Extending the hybrid model to additional tasks like machine translation, named entity recognition, or sentiment analysis [26].
5. Real-Time Applications**:** Implementing the model for real-time POS tagging and spellchecking in NLP applications like chatbots or virtual assistants.

## Acknowledgment

The authors would like to express their sincere gratitude to Lovely Professional University for providing the resources and support necessary to complete this research. We extend our heartfelt thanks to our mentors and professors for their invaluable guidance and constructive feedback throughout the course of this work. Additionally, we are grateful to the developers and contributors of the IndicNLP library, Stanza, and other opensource tools utilized in this research. These tools have played a crucial role in the implementation of the proposed models. Finally, we would like to acknowledge the individuals who assisted with the preparation of the dataset and the reviewers for their insightful suggestions that helped improve the quality of this work.

## References

[1] Jurafsky, D., & Martin, J. H. (2020). Speech and Language Processing. 3rd Edition.
[2] Brill, E. (1995). Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. Computational Linguistics.
[3] Church, K. W., & Gale, W. A. (1991). Probability Scoring for Spelling Correction. American Journal of Computational Linguistics.
[4] Manning, C. D., & Schutze, H. (1999). ¨ Foundations of Statistical Natural Language Processing. MIT Press.
[5] Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. ICML.
[6] Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE.
[7] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation.
[8] Graves, A. (2012). Supervised Sequence Labelling with Recurrent Neural Networks. Springer.
[9] Lample, G., et al. (2016). Neural Architectures for Named Entity Recognition. ACL.
[10] Subburathinam, A. S., et al. (2019). Exploring Neural Architectures for POS Tagging in Tamil. ICON.
[11] Peters, M. E., et al. (2018). Deep Contextualized Word Representations. NAACL.
[12] Jurafsky, D., & Martin, J. H. (2020). Speech and Language Processing. 3rd Edition.
[13] Manning, C. D., et al. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. ACL.
[14] Devlin, J., et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT.
[15] Vaswani, A., et al. (2017). Attention Is All You Need. NeurIPS.
[16] Seo, M., et al. (2016). Bidirectional Attention Flow for Machine Comprehension. ICLR.
[17] Subburathinam, A. S., et al. (2019). Exploring Neural Architectures for POS Tagging in Tamil. ICON.
[18] Kumar, R. et al. (2019). A Comparative Study on Various POS Tagging Techniques for Indian Languages. JOLR.
[19] Qi, P., et al. (2020). Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. ACL.
[20] Subburathinam, A. S., et al. (2019). Neural Architectures for POS Tagging in Low Resource Languages. ICON.
[21] Norvig, P. (2007). How to Write a Spelling Corrector. Norvig.com.
[22] Mikolov, T., et al. (2013). Efficient Estimation of Word Representations in Vector Space. ICLR.
[23] Peters, M. E., et al. (2018). Deep Contextualized Word Representations. NAACL.
[24] Devlin, J., et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT.
[25] Vaswani, A., et al. (2017). Attention Is All You Need. NeurIPS.
[26] Seo, M., et al. (2016). Bidirectional Attention Flow for Machine