University of Sindh Journal of Information and Communication Technology (USJICT)
Volume 7, Issue 1
ISSN-E: 2523-1235, ISSN-P: 2521-5582
Website: http://sujo.usindh.edu.pk/index.php/USJICT/
© Published by University of Sindh, Jamshoro

Improving Part of Speech Tagging for Learner English.

Taiwo Kolajo, Emeka Ogbuju, Abubakar Aliyu, Victoria Yemi-Peters, Kayode Atteh, Francisca Oladipo

Department of Computer Science, Federal University Lokoja; Thomas Adewumi University
taiwo.kolajo@fulokoja.edu.ng; emeka.ogbuju@fulokoja.edu.ng; abubakar.aliyu@fulokoja.edu.ng; Victoria.yemi-peters@fulokoja.edu.ng; attehkayode2@gmail.com; francisca.oladipo@tau.edu.ng

Abstract: As advanced algorithms for natural language processing are being developed, fundamental tasks like part of speech tagging remain crucial and challenging. As a crucial step in natural language processing, part of speech tagging assigns each word in a phrase to the correct part of speech. Spelling mistakes are a frequent blunder we make when utilizing computer software. Spelling mistakes have a negative impact on part-of-speech tagging performance. In this paper, the integration of spell-checking into part-of-speech tagging was proposed. In our comparison of four spell checkers—Norvig, Jamspell, Hunspell, and Dummy—Jamspell outperformed the others in terms of accuracy, speed, and the number of words it broke during the correction. This influenced our decision to use Jamspell to correct spelling mistakes. Treebank, Brown, and Conll2000 NLTK corpora were utilized for the project's Part of speech tagging. We applied a softmax function on the hidden state of the LSTM after passing an LSTM over the tokens to make predictions. The outcome is a vector of tag scores from which we can determine the projected tag for a word based on the highest value in this distribution of tag scores. The results show an accuracy of 97%, a macro average of 98% and a weighted average of 97% when tested on 212 tagged sentences after spelling correction was applied. The results also show that integrating spell-checking into part-of-speech tagging process improves its accuracy.

Keywords: Part-of-speech tagging; Spelling checker; LSTM; Learner English

I. INTRODUCTION

Technology development in our society now generates a significant amount of daily information, notably in emails and weblogs. The creation of these electronic papers has altered how we interact today, making information organization efficient for everybody. As advanced algorithms for NLP (Natural language processing) are being developed, some fundamental tasks, such as part of speech tagging, continue to be significant and difficult [1]. Assigning words in a phrase to the correct part of speech is a job that stands out as being important in natural language processing.

Spelling checkers are automated tools used in the computer industry to find misspelt words and provide suitable corrections for each one. Spelling mistakes are a frequent blunder we make when utilizing computer software. For instance, typing errors might be fatal in the field of medicine and are common owing to the abundance of scientific terminology employed there that can be challenging to type even when written in English [2]. Automatic spelling correction is a crucial function in search engines since user-generated material frequently contains spelling mistakes. Many websites or search engines offer suggestions for typographical mistakes in user inquiries [3].

Part of Speech Tagging (POS), which assigns the appropriate POS tags from a predetermined language tag or a lexical category to each word in a document or phrase, e.g., verb, adverb, noun, pronoun, etc., is one of the well-known problems in Natural Language Processing (NLP). The POS tagger's accuracy is used to assess NLP systems. A variety of NLP applications, including chunking, text creation, machine translation, speech recognition, knowledge extraction, language comprehension, and word meaning disambiguation, heavily rely on the POS tagger [4][5]. POS tagging assists in automatically evaluating any document or text in addition to assigning grammatical categories to words in a text.

Reading, spelling, and writing are fundamental abilities that are required in modern culture. These abilities are necessary to complete forms, compose letters, and edit documents. Typographical errors are virtually always present while editing and typing manuscripts, which is a challenge for many overworked editors. There are many people whose spelling or writing abilities are affected by learning disorders like dyslexia. Dyslexia is characterized by poor spelling and decoding skills as well as trouble with accurate and/or fluent word recognition. Secondary effects might include reading comprehension issues and a lack of reading experience, both of which can inhibit vocabulary development and acquisition [6]. When determining which part of speech a word belongs to or when spelling a word, English learners frequently make mistakes, especially those for whom English is a second language [7]. In certain situations, these mistakes are caused by the writer's mother tongue. This could result in weak composition.

To enhance POS tagging results, this work proposes to develop a model for correcting misspelt words in sentences and documents. Spelling mistakes, which reduce the effectiveness of natural language processing directed at English learners, are one of the main characteristics of English Language Learners (ELL) [8]. An effective approach that can help those who are learning English, as well as those for whom English is a second language, must be developed.

II. Background and Related Work

This section presents the background and research efforts in spelling correction and part-of-speech tagging.

A. Spelling Correction and Part of Speech Tagging

Many pieces of software, including text editors, word processors, and online search engines, have spell checkers. When looking for a certain subject, search engines' "Did you mean?" recommendations for misspelt words may be quite helpful. Correct spelling not only eases the user's cognitive burden on the task but also increases the search engine's accuracy as it tries to anticipate the user's purpose [9]. High linguistic proficiency is typically necessary for research writing. For those who are not native to a nation, this is a significant challenge [10].

One of the most important components, applications, and features in natural language processing is part of speech tagging. Taggers are used to do POS tasks. Assigning the appropriate POS tag to each word in a text from a predetermined tag collection for a language, such as a verb, adverb, noun, pronoun, etc., is the task of these taggers, which are made up of a variety of sets of linguistics rules [4][11]. When tagging, ambiguity frequently causes issues, especially with statements that have many contextual meanings. Machine translation, question-answering, word meaning disambiguation, parsing, and many more fields have found a use for POS tagging [12].

B. Spelling Error Detection and Correction Techniques

To find, fix, and POS tag words in a text, there are several techniques or procedures. Edit distance, similarity keys, rule-based, n-gram-based, neural network, noisy channel models, stochastic, and hybrid are a few strategies. In this section, several strategies are briefly explained.

- N-gram evaluation: An N-gram is a collection of n words drawn from various grams, with n being a fixed number. The approach is not language-specific and may be used in any language [13].
- Dictionary/Wordnet: A lexical database that contains a list of acceptable terms in a specific language is called a dictionary or wordnet. By checking every word against a dictionary, which includes nouns, verbs, adjectives, and adverbs, one may quickly identify non-word mistakes [14].
- Levenshtein edit distance: The least amount of editing operations, such as insertions, deletions, and substitutions, between two strings required to change one word into another is known as the. It is one of the easiest approaches since it only requires one incorrect character operation (insertion, deletion, and replacement) to change a dictionary word into a non-word, assuming that most people make a few mistakes [5].
- Similarity keys: This method or methodology compares each key computed for a non-word with the key assigned to each word in the dictionary. With an accurate algorithm transformation like BioWordVec, this method can address keyboard spelling errors [15].
- Rule-based: One of the first approaches to natural language processing is rule-based. A rule-based approach offers a set of criteria for identifying typographical and orthographic mistakes, which are then applied to the wrong word. By adding up the probabilities for the applicable rules, the rules employ probabilities to rank the suggestions [16].
- Artificial neural networks: An artificial neuron is made to try to mimic the structure and actions of a natural neuron. In most situations, back-propagation networks are utilized, with an output node for each word in the dictionary and an input node for each probable n-gram in each position of the word, where n is often one or two [17]. It is one of the most effective NLP approaches.
- Noisy channel model: Finding the intended word from a word with part of the letters jumbled is the goal of a noisy channel model. The technology is utilized for machine translation, question-answering, speech recognition, and spell-checking [18][19].

C. Techniques for Part-of-Speech Tagging

- Rule-based: One of the first methods for POS tagging was rule-based. Regular expressions or context-pattern rules can be used as the rules. When developing rules, the language's linguistic characteristics—including lexical, morphological, and syntactical data—are taken into account. These guidelines are used to prune out potential POS tags [20].
- Stochastic POS tagging: This technique may be used with statistical models. The word frequency technique and tag sequence probability are the two most common and straightforward ways for stochastic taggers [21]. The hidden Markov model and the maximum entropy Markov model are two more often utilized strategies [12].
- Hybrid approach: This combines rule-based and stochastic-based techniques to achieve POS tagging. One of the advantages of using a hybrid approach is to achieve highly reliable models that can adapt to different forms of languages [22].

D. Related Work

This section presents the research efforts in spelling correction and part of speech tagging.

For spelling correction of Indic languages (SCMIL) as in [3] presented a character-based recurrent sequence-to-sequence architecture comprising an LSTM encoder and an LSTM decoder. By accumulating highly likely spelling errors with the inclusion of a noisy clean corpus, the experiment was conducted on synthetic datasets of noisy and accurate word mappings for Hindi and Telugu. The experiment demonstrates that SCMIL outperforms Statistical Machine Translation (SMT) with an accuracy of 85.4% for Hindi languages and 89.3% for Telugu languages after comparison with cutting-edge methods like Statistical Machine Translation (SMT) and Rule-based methods. For spelling correction in foreign languages, a model that uses a multi-layer recurrent neural network as an encoder and decoder layer was developed by [9]. Internal datasets that were created automatically from user logs were used in the tests. The model demonstrated competitive performance to the cutting-edge techniques, with an accuracy of 62.5%. Reference [23] researched the creation of a spell checker for the Icelandic language that could identify and repair spelling problems and was targeted at those with dyslexia and those for whom Icelandic is a second language. Approximately 20% of spelling mistakes produced by dyslexic persons are real-word errors, which may be identified and fixed by looking at the context in which the term was used. The dataset employed for the experiment was the Icelandic Confusion Set Corpus containing about 1.3 billion running words of text which are tagged morphologically. A decision tree binary classification was used to build the system giving an accuracy average of 73% to 86%.

In the study of the issue of real-world mistake detection and correction based on statistical machines for the Assamese language, [24] used the n-gram language model. The model entails obtaining a confusion set unique to each testing word in the phrase and identifying the word from the confusion set with the highest probability given the sentence's context. The development of the n-gram language model, which assists in determining the context of a word in a phrase, was done using a standard corpus dataset for the Assamese language. Based on accuracy and recall, the system's performance for word recommendations and spell-checking was evaluated. The system was noted to have an overall performance of 76%. To improve performance and accuracy, [25] used a hybrid approach to develop a spell checker for the Punjabi language. A word dictionary, a tree-based algorithm with n-grams, LSTM, a rule-based approach, and minimal edit distance make up the hybrid system. By using a letter-to-letter prefix search on a tree-based dictionary with a tree-based algorithm, error detection is made feasible. However, when the term is not included in the tree-based dictionary, the model turns to n-gram analysis. Based on the relationship between the misspelt words and prior word sequence, LSTM was used to suggest replacement terms for misspelt words. Reference [26] used an annotated corpus comprising roughly 6k spelling errors in context, gathered from essays authored by English language learners, to address the lack of datasets that contain plausible spelling errors. The method of correcting spelling errors involves three steps: candidate generation, candidate ranking, and candidate detection. A spelling correction system that requires little supervision and achieves an accuracy of 88.12% was developed. It was claimed that the model could function at a respectable level when applied to other areas. More precisely, the model was evaluated using a medical dataset, and its performance was found to be competitive with that of a model evaluated and trained using the same dataset. Reference [27] suggested a method for correcting spelling errors. The authors combined Levenshtein distance, double metaphone algorithm, and global vectors (GloVe). The results showed improved performance over individual methods.

The focus of several research projects has been POS tagging. Reference [28] look at how POS labelling might be modified to sufficiently and effectively assist English students. They demonstrate that the deep neural network outperformed the most advanced POS tagger, with an accuracy of 0.96, in handling POS tagging. It was emphasized that several POS tagging problems were the consequence of grammatical and orthographic mistakes. The model is made up of three main components: a softmax layer, a bidirectional long short-term memory, and an embedded layer. The training text input contains spelling mistakes, which impacted the POS-tagger model's accuracy and performance. A model created by [29] was able to offer word-choice recommendations for writing by an ESL learner, also known as an English second language learner. The model was implemented using a bidirectional LSTM tagger, and it was claimed to perform better than state-of-the-art techniques, recommending words like "show" as a suitable replacement for the printed word "indicate." Reference [30] researched the Urdu language to develop part-of-speech (POS) tagging. A challenging issue is presented by the Urdu language's extensive morphological inflectional and derivational structure. To train the conditional random field (CRF), support vector machine (SVM), recurrent neural network (RNN), long short-term memory (LSTM), and hidden Markov models utilized in the study, data from the Center for Language Engineering (CLE) and Bushra Jawaid (BJ) datasets were used. The LSTM-RNN model performed much better than the other models in a cross-validation experiment carried out in the study on the CLE dataset, whereas the CRF model fared far better than other models.

A deep neural network was suggested by [31] for a POS tagging system. It comprises a pre-trained static model for representing words with increased morphological information. To handle both consistently prevalent and well-known words as well as unusual and out-of-vocabulary terms, the FastText model approximated the Word Embedding representation learned from an unlabeled corpus. A character-level representation of words is dynamically learnt and concatenated with the prior one, according to the POS tagging job. The main network, which consists of a Bi-LSTM layer trained to link a series of tags with a sequence in terms of language, receives this combined representation as input. An extensive experimental program shows the effectiveness of the inputs to the suggested system in comparison to the state-of-the-art. Reference [32] annotated

a POS-tagged corpus due to a shortage of available training data. A 52,190-word Maithili corpus was carefully annotated by the authors, who also generated a POS tagset. Using the corpus, a conditional random fields (CRF) classifier was trained. The word2vec CBOW model was trained for CRF training, and the word vectors produced were used. A large raw corpus of 110 K phrases from various online data sources was gathered, and neural word embedding of the Maithili words was learned. The system's accuracy increased dramatically with this modification, reaching 85.88%. Additionally, tests using a variety of context and orthography characteristics in CRF yielded an accuracy of 82.67%.

The taggers for eight languages were subjected to several methodologies by [33], who then presented a novel study revealing the taggers to be size-performance optimum. The methods offered a range of size measures and a multilingual comparison of part-of-speech taggers. The methods include calculating the accuracy of tokens and sentences as well as their size imprints. To identify which taggers are effective and which are not, size and accuracy were sufficient indicators. The results demonstrate that some conventional taggers prioritize size performance across languages. Additionally, generic knowledge may be learnt and encoded from a range of languages and datasets with a little computational footprint. Using deep learning, [34] created a joint word segmentation and part-of-speech tagging model for the Khmer language, the nation of Cambodia's official tongue. Text is written in the language from left to right, with the option of a space. When writing in Khmer, space is not used to delimit words; rather, it is employed to make the text easier to read or for other reasons. The suggested model was trained and tested using the openly available Khmer POS dataset. A bidirectional LSTM network, which accepts character-level inputs and produces a series of POS tags, makes up the deep learning model. The segmentation accuracy of the proposed model is 97.11%, and the POS tagging accuracy is 94.00%, which is equivalent to the accuracy of the conventional two-stage Khmer POS tagging.

Reference [35] proposed a deep neural network-based method for better POS labelling in Amharic. For Amharic POS tagging, the authors assessed various models including BiLSTM, CNN, and CRF as well as their combinations. 31 POS tags were manually added to 321K tokens to evaluate each model's performance. The maximum accuracy, 97.23%, was attained by the CNN_BiLSTM-CRF combination.

Reference [36] proposed the use of part-of-speech tagging with the use of morphological analysis. Reference [37] suggested feature selection as an improvement an improvement to POS tagging results.

Multiple languages have been used to accomplish the identification and repair of misspelt words or part of speech tagging (POS) using various machine learning methodologies and techniques. One of the key drawbacks was the size of the dataset needed for POS tagging and auto-spell checkers, which was necessary for high accuracy but might be restricted depending on the language. Deep learning has been the prevalent approach used in recent times to build systems that perform POS tagging or auto-correction of words or sentences. There is a need for an efficient POS tagging model and an effective POS tagging for better results [12].

III. METHODOLOGY

Enhancing the accuracy and precision of applying the proper Part-of-Speech (POS) tags to words in sentences written by non-native English speakers is necessary to improve POS tagging for learner English. Incorporating a spell checker into the POS tagging process is a practical way to do this. This section explains how to achieve the research's main objective, which is to enhance POS taggers with spelling mistake connections, as well as the architecture for the suggested model (see Figure 1). The suggested model attempts to offer a paradigm that may be utilized for POS tagging as well as context-sensitive spell-checking.

The Cross-Industry Standard Process for Data Mining (CRISP-DM) is the approach chosen for use in these research. The scientific procedures necessary to implement the chosen technique are systematically followed.

First, an understanding of the research work, which is the first stage of CRISP-DM was carried out. Knowing exactly what has to be done—in this example, a spelling detection, correction, and POS tagging system that can be applied to the English language—is a requirement for business knowledge. The spell checker module may be applied in typical situations like phrase correction and POS tagging. Each word in a phrase or text will be tagged according to its proper part of speech by the POS tagging model.

Figure 1. Architectural design of improved part of speech tagging model.

An analysis of the data is done in the subsequent stage of CRISP-DM. Books make up a sizable portion of the datasets, which were gathered from a variety of sources. The methodology's third phase includes. The model is created using the Python programming language in the fourth step. The description of the architecture of an LSTM POS tagger is presented in Section 3.1.

A. The Architecture of an LSTM POS Tagger

Long short term memory (LSTM) [38] neural networks are a type of recurrent neural network (RNN) that are used in POS tagging. RNNs are capable of processing sequential data and storing information from earlier stages in their recurrent hidden layers for use in subsequent steps. Recurrent neural networks are a suitable classifier for NLP problems because of this ability. Recurrent neural networks (RNNs) showed promise in solving problems such as language modeling and POS tagging after their introduction. However, the length of sequences that RNNs could learn was restricted due to a phenomenon known as vanishing or exploding gradient [39][40], which occurs when a long sequence of data is fed into the network. With minor RNN modifications, LSTM neural networks were developed as a solution to this issue. An input gate, a forget gate, and an output gate make up an LSTM block, which functions as a memory cell in recurrent hidden layers neural networks.

The forget gate layer, a sigmoid in the LSTM unit, informs us what has to be forgotten. Subsequently, the input gate layer—a sigmoid layer once more—determines which values require updating. The LSTM unit then generates a vector of fresh candidate values using a tan h layer. Next, the LSTM unit's status is updated. The output layer then uses a sigmoid function to determine what should be output. The output layer, which is really the LSTM unit's ultimate output, is finally multiplied by the output of this sigmoid layer. Each class is then given a probability using a softmax layer.

According to the design, the spelling correction module corrects errors in spelling before building a word dictionary and point-of-sale system (POS). The phrases are prepared by a helper function and then sent to the LSTM tagger for training. Next is a presentation of the step-by-step procedure from dataset to model training.

B. Dataset

The datasets used were compiled from different sources, of which books form a large part of the datasets. We obtained 72202 sentences with part-of-speech tags from various sources using the NLTK corpora. It was downloaded to use the Treebank, Brown, and Conll2000 nltk corpora. For training, the three datasets were then pooled.

C. Data Preprocessing

The noise was eliminated from the obtained data during cleaning. Unwanted characters like [@_*>()#%+=[]] and superfluous spaces from the text were eliminated The splits() function in Python was used to further divide each phrase into tokens of words before changing each sentence's case to lower case. As can be seen in the code snippets below, a dictionary tag was then made for each component of speech:

tag2idx = {"NOUN": 0, "ADJ": 1, "NUM": 2,}

Every word in a sentence was assigned a number, starting from 0,1,2,3,…. The first unique word found in the sentence was assigned 0 and the second unique word was assigned 1 and so on…

{'the': 0, 'cat': 1, 'ate': 2, 'cheese': 3, 'she': 4, 'read': 5, 'that': 6, 'book': 7, 'dog': 8, 'loves': 9, 'art': 10, 'elephant': 11, 'answers': 12, 'phone': 13 ...}.

D. Spelling Correction

JamSpell was compared against Norvig, Hunspell, and dummy spell checkers. A spelling checker library called JamSpell provides context-sensitive correction suggestions in addition to dictionary-based corrections. You may also utilize and train your own manually collected datasets with this multilingual package. The Norvig spell checker searches the dictionary for every phrase that may be generated from the query term with an edit distance (deletes + transposes + replacements + inserts). A high-level wrapper for locating spelling mistakes in a text document is the Hunspell function. It reads text from a character vector in any format—plain, latex, man, HTML, or XML—parses the words, and outputs a list of misspelled terms for every line [41].

A total of 300,000 sentences from the English news and 300,000 words from Wikipedia were used to train the spell checkers. 95 per cent of the time was devoted to training, and just 5 per cent to evaluation. The typo_model.py's mistakes model was used to create error-filled text from the original text. The best performance spell-checker was used for the part of speech tagging.

E. Word Embedding

An LSTM requires an expected input size and hidden dim. As illustrated in Figure 2, we designed an embedding layer that accepts the size of our vocabulary and generates an embedding dim vector for each word in an input word list.

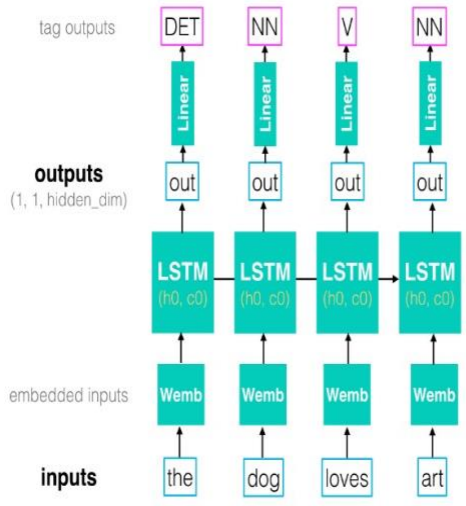


Figure 2. Layers of LSTM tagger

F. Defining Parameters

To train the model, we specified the loss and optimizers that would be used. The first thing is to define the size of the word embedding. For the embedding dimension, hidden dimension and embedding dimension, 128 was used.

G. Loss and Optimization

The Negative Loss Likelihood (NLL) Loss was employed since the LSTM's output is a collection of tag scores with a softmax layer. When combined with a softmax layer, NLL Loss produces the kind of cross-entropy loss that is typically used to evaluate the distribution of class scores. For this work, we explored optimizers like Adams (Adaptive Moment Estimation) and gradient descent.

H. Loss and Optimization

The training data was repeatedly looped through using an epoch size of 30. We zeroed the gradients to get the model ready for training. The LSTM's hidden state was initialized. After that, data for the training was prepared. It performs a forward pass on the inputs to obtain tag_scores, then computes the loss between tag_scores and the true tag. Backpropagation was used to update the weights.

IV. Results and Discussion

The section presents the results of the comparison of the four spell checkers and the part of speech tagging. The result also shows part of speech tagging before and after the application of the spell checker.

A. Comparison of Spelling Checkers

We compared four spell checkers: Norvig, Hunspell, spell checker, dummy, and JamSpell. The result of the comparison is shown in Table 1.

TABLE I. Comparison between Jamspell, Norvig Hunspell, and Dummy

Spell Checker	*Errors (%)*	*Top 7 Errors (%)*	*Fix Rate (%)*	*Top 7 Fix Rate (%)*	*Broken (%)*	*Speed (words/second)*
JamSpell	3.25	1.27	79.53	84.10	0.64	4854
Norvig	7.62	5.00	46.58	66.51	0.69	395
Hunspell	13.10	10.33	47.52	68.56	7.14	163
Dummy	13.14	13.14	0.00	0.00	0.00	-

From Table 1, JamSpell had the best performance in terms of speed and low error rate. The most important metric is the error rate because it shows the percentage of incorrect words after the spell checker is applied. Getting the correct possible output is the main desire of the user. This informed the choice of JamSpell for the spell checking in this work.

B. Part-of-Speech Tagging Results

The part-of-speech tagging model was applied to 100 tagged sentences from the train set and 112 tagged sentences from manually generated sentences. The model tagged 170 sentences correctly. The sentences were first applied to the model without a spell checker and then with a spell checker. Tables 2 and 3 show the results obtained before and after applying spellchecker, respectively.

TABLE II. Precision, Recall, F1-score, and Support for POS Tagging before applying spell checker

	Precision	*Recall*	*F1-score*	*Support*
.	1.00	1.00	1.00	19
ADJ	0.72	0.78	0.75	17
ADP	0.89	0.89	0.89	30
ADV	1.00	1.00	1.00	4
DET	1.00	1.00	1.00	22
NOUN	1.00	1.00	1.00	64
NUM	1.00	1.00	1.00	5
PRON	1.00	1.00	1.00	5
PRT	1.00	1.00	1.00	9
VERB	1.00	0.96	0.98	27
X	1.00	1.00	1.00	10
Accuracy			0.85	212
Macro avg	0.98	0.98	0.86	212
Weighted Avg	0.97	0.97	0.85	212

TABLE III. Precision, Recall, F1-score, and Support for POS Tagging after applying spell checker

	Precision	*Recall*	*F1-score*	*Support*
.	1.00	1.00	1.00	19
ADJ	0.83	0.88	0.85	17
ADP	0.90	0.90	0.90	30
ADV	1.00	1.00	1.00	4
DET	1.00	1.00	1.00	22
NOUN	1.00	1.00	1.00	64
NUM	1.00	1.00	1.00	5
PRON	1.00	1.00	1.00	5
PRT	1.00	1.00	1.00	9
VERB	1.00	0.96	0.98	27
X	1.00	1.00	1.00	10
Accuracy			0.97	212
Macro Avg	0.98	0.98	0.98	212
Weighted Avg	0.97	0.97	0.97	212

After applying a spell checker to the 212 tagged phrases, the model had an accuracy of 97%, compared to 85% without the spell checker. In tags like adverbs, determinants, nouns, numerals, pronouns, and verbs, the model fared better. The adjective and adposition (a cover name for prepositions and postpositions) were tagged incorrectly in the model in a few instances.

V. Conclusion and Further Work

Part of speech tagging is a fundamental part of natural language processing, text summarization and speech recognition. In this paper, a spelling correction based on real word errors and a part-of-speech tagging system was developed. The central idea of the research is to improve the accuracy of part of speech tagging by applying a spell checker. Four spell-checkers, which include Norvig, Hunspell, Jamspell, and Dummy were compared. Jamspell

spell checker has the best performance. The Part of Speech tagger was built using Long short-term memory and achieved an accuracy of 97%, a macro average of 98% and a weighted average of 97% on 212 tagged sentences when spellchecker was applied. Without a spell checker, the model had an accuracy of 85%. Applying a spellchecker before POS tagging improves the accuracy of the POS tagger.

The improved part-of-speech tagging system is highly recommended for people for whom English is a second language and dyslexic persons, to assist them in writing. This could also be used in other research areas such as text summarization, language modelling, sentiment analysis and speech recognition. The research can be improved by providing more datasets and exploring other optimizers. The system can also perform better with context-sensitive spell-checking. A possible direction for future study is to provide an efficient POS tagging model by using less sophisticated deep learning algorithms and efficient POS tagging in terms of detection mechanism.

DECLARATION OF COMPETING INTEREST

The authors declare that no competing interests, either financial interests or personal relationships inappropriately influence this research work.

ACKNOWLEDGMENT

The authors would like to thank the Department of Computer Science, Federal University Lokoja, Nigeria for providing an enabling environment to carry out this research.

REFERENCES

[1] R. Stankovic, B. Sandrih, C. Krstev, M. Utvic, and M. Skoric, "Machine learning and deep neural network-based lemmatization and morphosyntactic tagging for Serbian," Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), ELRA, Marseille, 2020, pp. 3954-3962.

[2] N. Zukarnain, B. S. Abbas, S. Wayan, A. Trisetyarso, and C. H. Kang, "Spelling Checker Algorithm Methods for Many Languages," 2019 International Conference on Information Management and Technology (ICIMTech), Jakarta/Bali, 2019, pp. 198-201,.

[3] M. Chinnakotla, P. Etoori, and R. Mamidi, "Automatic spelling correction for resource-scarce languages using deep learning," Proceedings of ACL 2018, Student Research Workshop, Melbourne, 2018, pp. 146–152.

[4] T. Dalai, T. P. Mishra, P. K. Sa, "Part-of-speech tagging of Odia language using statistical and deep learning based approaches," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 22(6), 2023, 167. https://doi.org/10.1145/3588900.

[5] S. Rani, H. F. Naqvi, F. H. Khoso, A. Agha, D. N. Hakro, M. Hameed, "Named entity recognition for Urdu Language: the UNER system, a hybrid approach," University of Sindh Journal of Information and Communication Technology(USJICT), vol. 6(3), 2022, pp. 108-114.

[6] J. Zhang and Q. Wang, "Reading and writing difficulties in bilingual learners," Annals of Dyslexia, vol. 73, 2023, pp. 1-5. https://doi.org/10.1007/s11881-023-00282-8.

[7] L. F. Wanderley, and C. D. Epp, "Identify negative language transfer in learner errors using POS information. Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications," ACL, 2021, pp. 64-74.

[8] M. Imtiaz, K. H. U. Hassan, and F. Akmal, "Analyzing spelling errors committed by English as a second language (ESL) learners at secondary school level," Journal of Social Sciences Review, vol. 3(2), 2023, pp. 181-189. https://doi.org/10.54183/jssr.v3i2.246.

[9] Y. Zhou, U. Porwal, and R. Konow, "Spelling correction as a foreign language," in Proceeding of the SIGIR 2019 eCom workshop, J. Degenhardt, S. Kallumadi, U. Porwal, A.rotman Eds, CEUR: Paris, France, 2019.

[10] E. Fischer, P. E. Engelhardt, J. Horvath, and H. Ohtani, "Experimental ordinary language philosophy: a cross-linguistic study of defeasible default inferences," Synthese, vol. 198, 2021, pp. 1029-1070, https://doi.org/10.1007/s11229-019-02081-4.

[11] A. Gopalakrishnan, K. P. Soman, and B. Premjith, "Part-of-Speech tagger for biomedical domain using deep neural network architecture," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, pp. 1-5. IEEE, 2019. doi:10.1109/ICCCNT45670.2019.8944559.

[12] A. Chiche, and B. Yitagesu, "Part of speech tagging: a systematic review of deep learning and machine learning approaches," Journal of Big Data, vol. 9, 2022, 10, https://doi.org/10.1186/s40537-022-00561-y.

[13] Y. Chaabi and F. A. Allah, "Amazigh spell checker using Damerau-Levenshtein algorithm and N-gram," Journal of King Saud University Computer and Information Sciences, vol. 34(8, Part B), 2022, pp. 6116-6124. https://doi.org/10.1016/j.jksuci.2021.07.015.

[14] T. Kolajo, O. Daramola, A. Adebiyi, and A. Seth, "A framework for pre-processing of social media feeds based on integrated local knowledge base," Information Processing & Management, vol. 57(6), 2020, 102348.

[15] T. Kim, S. W. Han, M. Kang, S. H. Lee, J. Kim, H. J. Joo, and J. W. Sohn, "Similarity-based unsupervised spelling correction using biowordvec: development and usability study of bacteria culture and antimicrobial susceptibility reports," JMIR Medical Informatics, vol. 9(2), 2021, e25530.

[16] L. Wiechetek, F. A. Pirinen, M. Hamalainen, and C. Argese, "Rules ruling neural networks vs. rule-based grammar checking for a low resource language," Proceedings of Recent Advances in Natural Language Processing, 2021, pp. 1526-1535. https://doi.org/10.26615/978-954-452-072-4_171.

[17] K. Rakesh, B. Minu, and S. Kumar, "A study of spell checking techniques for Indian Languages," JK Research Journal in Mathematics and Computer Sciences, 2018, 105-113.

[18] D. Jurafsky, J. H. Martin, "Spelling correction and the noisy channel," Speech and Language Processing, 2023.

[19] A. Shar, A. K. Soomro, H. F. Naqvi, F. H. Khoso, M. Hameed, D. N. Hakro, "Balochi Speech Recognition using Android Based Smart Phone," University of Sindh Journal of Information and Communication Technology(USJICT), vol. 5(4), 2021, pp. 208-212.

[20] H. Li, H. Mao, and J. Wang, "Part-of-speech tagging with rule-based data preprocessing," Electronics, vol. 11(1), 2021, 56. https://doi.org/10.3390/electronics11010056.

[21] S. G. Withanage, and T. Silva, "A stochastic part of speech tagger for the Sinhala language based on social media data mining," 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer), IEEE, Colombo, Sri Lanka, 2020, pp. 137-142.

[22] G. L. Izzi, and S. Ferili, "UniBA @ KIPOS: a hybrid approach for part-of-speech tagging," CEUR-WS, vol. 2765, 2020, 125.

[23] A. K. Ingason, and S. R. Friðriksdóttir, "Disambiguating confusion sets as an aid for dyslexic spelling," Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI), European Language Resources Association, Marseille, 2020, pp. 1-5.

[24] R. Choudhury, N. Deb, and K. Kashyap, "Context-sensitive spelling checker for Assamese Language," in Recent Developments in Machine Learning and Data Analytics. Advances in Intelligent Systems and Computing, Kalita, vol. 740, J. Balas, V. Borah, S. and

R. Pradhan, Eds. Springer: Singapore, 2019. https://doi.org/10.1007/978-981-13-1280-9_18

[25] G. Kaur, K. Kaur, and P. Singh, "Spell checker for Punjabi Language using deep neural network," 2019 5th International Conference on Advanced Computing Communication Systems (ICACCS), IEEE, Coimbatore, India, 2019, pp. 147-151,

[26] A. Rozovskaya, M. Fried, and M. Flor, "A benchmark corpus of English misspellings and a minimally supervised model for spelling correction," Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics, Florence, Italy, 2019, pp. 76-86.

[27] G. Huang, J. Chen, and Z. Sun, "A correction method of word spelling mistakes for English text," Journal of Physics: Conference Series, vol. 1693, 2020, 012118.

[28] R. Nagata, T. Mizumoto, Y. Kikuchi, Y. Kawasaki, and K. Funakoshi, "A POS tagging model adapted to learner English," Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text, ACL, Brussels, Belgium, 2018, pp. 39-48.

[29] V. Makarenkov, L. Rokach, and B. Shapira, "Choosing the right word: using bidirectional LSTM tagger for writing support system," Engineering Applications of Artificial Intelligence, Vol. 84, 2019, pp. 1-10, https://doi.org/10.1016/j.engappai.2019.05.003.

[30] W. Khan, A. Daud, K. Khan, J. A. Nasir, and F. S. Alotaibi, "Part of speech tagging in Urdu: comparison of machine and deep learning approaches," IEEE Access, vol. 7, 2019, pp. 38918-38936.

[31] M. Pota, F. Marulli, M. Esposito, G. De Pietro, and H. Fujita, "Multilingual POS tagging by a composite deep architecture based on character-level features and on-the-fly enriched word embeddings," Knowledge-Based Systems, vol. 164, 2019, pp. 309-323. https://doi.org/10.1016/j.knosys.2018.11.003

[32] A. Priyadarshi, and S. K. Saha, "Towards the first Maithili part of speech tagger: resource creation and system development," Computer Speech & Language, vol. 6, 2019, 101054. doi:https://doi.org/10.1016/j.csl.2019.101054

[33] M. Jacobsen, M. H. Sørensen, and L. Derczynski, "Optimal size-performance tradeoffs: weighing PoS tagger models," arXiv:2104.07951 [cs.CL], 2021.

[34] R., Buoy, N. Taing, and S. Kor, "Joint Khmer word segmentation and part-of-speech tagging using deep learning," arXiv:2103.16801 [cs.CL], 2021. https://doi.org/10.48550/arXiv.2103.16801

[35] S. Hirpassa, and G. S. Lehal, "Improving part-of-speech tagging in Amharic language using deep learning," Heliyon, vol. 9(7), 2023, e17175. https://doi.org/10.1016/j.heliyon.2023.e17175.

[36] J. Liu, R. Chen, C. Liang, "Improving extraction of Japanese functional expressions with discontinuous types through part-of-speech tagging," Proceedings of the 2023 6th International Conference on Big Data and Education (ICBDE '23),. ACM, Jinan, China June, 2023, pp. 131-135. https://doi.org/10.1145/3608218.3608230

[37] B. Mounica, and K. Lavanya, "Feature selection method on Twitter dataset with part-of-speech (PoS) pattern applied to traffic analysis," International Journal of System Assurance Engineering and Management, vol. 15(1), 2024, pp. 110–123. https://doi.org/10.1007/s13198-022-01677-3.

[38] S. Hochreiter, J. Schmidhuber, "Long short-term memory," Neural Comput. vol. 9(8), 1997, pp. 1735–1780.

[39] Y. Bengio, P. Simard, P. Frasconi, "Learning long-term dependencies with gradient descent is difcult," IEEE Trans. Neural Netw., vol. 5(2), 1994, pp. 157–166.

[40] S., Besharati, H., Veisi, A., Darzi, S. H. H. Saravani, "A hybrid statistical and deep learning based technique for Persian part of speech tagging," Iran Journal of Computer Science, vol. 4, 2020, pp. 35–43. https://doi.org/10.1007/s42044-020-00063-1.

[41] M. Näther, "An in-depth comparison of 14 spelling correction tools on a common benchmark," Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), Marseille, May 2020, pp. 1849–1857.