Transfer Learning for Named Entity Recognition in Setswana Language Using CNN-BiLSTM Model

Shumile Chabalala[1], Sunday O. Ojo[2], Pius A. Owolawi[3]
Dept. Computer Systems Engineering, Tshwane University of Technology, Pretoria, South Africa[1, 3]
Dept. Information Technology, Durban University of Technology, Durban, South Africa[2]

***Abstract*—This research proposes a hybrid approach for Named-Entity Recognition (NER) for Setswana, a low-resource language, that combines a bidirectional long short-term memory (BiLSTM) with a transfer learning model and a convolutional neural network (CNN). Among the 11 official languages of South Africa, Setswana is a morphologically rich language that is underrepresented in the field of deep learning for natural language processing (NLP). The fact that it is a language with limited resources is one of the reasons for this gap. The suggested NER hybrid transfer learning approach and an open-source Setswana NER dataset from the South African Centre for Digital Language Resources (SADiLaR), which contains an estimated 230,000 tokens overall, are used in this research to close this gap. Five NER models are created for the study and contrast with one another to determine which performs best. The performance of the top model is then contrasted with that of the baseline models. The latter three models are trained at sentence-level, whereas the first two are at word-level. Sentence-level models interpret the entire sentence as a series of word embeddings, while word-level models represent each word as a character sequence or word embedding. CNN is the first model, and CNN-BiLSTM transfer learning based on Word level is the second. Sentence-Level is the basis for the last three CNN, CNN-BiLSTM Transfer Learning, and CNN-BiLSTM models. With 99% of accuracy, the CNN-BiLSTM Transfer Learning sentence-level outperforms all other models. Furthermore, it outperforms the state-of-the-art models for Setswana in the literature that were created using the same dataset.**

Keywords*—*Natural language processing; named entity recognition; convolutional neural network; bidirectional long short-term memory; Setswana

I. Introduction

The fact that computers use binary code and humans use text, color, and conversation may indicate that communication between the two is impossible. However, computers and humans are already able to communicate successfully thanks to Natural Language Processing (NLP). Human speech, visual content, and search queries can all be understood by computers. Machine comprehension and processing of human language is made possible by NLP [7].

In NLP and information extraction (IE), Named Entity Recognition (NER) is a fundamental task that aims to recognize and extract entities from text, such as names of individuals, organizations, places, and more. It is challenging to create NER systems for languages like Setswana because there aren't many studies in this field and there aren't enough language resources available. These languages are categorized as low-resourced languages as a result. Most people in Southern Africa speak Setswana, a Bantu language. As the fifth most widely spoken language in South African households, Setswana is one of the eleven official languages of the country and is spoken by 8.8% of the population. Speaking Setswana, Sesotho (Southern Sotho), and Sepedi (Northern Sotho), the Sotho people in Southern Africa speaks this language, which belongs to the southern branch of the Bantu language family [5], [17], [18], [19].

This paper proposes a transfer learning-based Convolutional Neural Networks (CNN)-Bidirectional Long Short-Term Memory (BiLSTM) NER model that takes account of linguistic semantic nuances of named entities in Setswana language.

The main research question answered in this paper is as follows: How can a Setswana NER model be developed using a Transfer learning-based CNN-BiLSTM deep learning approach?

This leads to the following sub-questions:

RQ-1: What are the linguistic semantic nuances of named entities in Setswana language?

RQ-2: What are the limitations of existing NER models in accommodating these nuances of Setswana as a low resource language?

RQ-3: How can a Transfer learning-based CNN-BiLSTM NER model that addresses these limitations be developed?

RQ-4: How can the NER model be experimentally evaluated?

The goal is to address the lack of NER resources for Setswana and advance NER tools for languages with limited resources.

The NER system's performance is assessed using a Setswana NER corpus from the South African Centre for Digital Language Resources (SaDilar), and the outcomes are compared with those of state-of-the-art NER models for Setswana that were tested on the same dataset. The study contributes to the expanding body of knowledge in research on NER for low-resourced languages and provides insightful information about the opportunities and difficulties associated with creating NER systems for Setswana.

The rest of this paper is structured as follows: Related Works is covered in Section II, the linguistic semantic subtleties of named entities in Setswana and including CNN-BiLSTM based on transfer learning. The methodology, which includes data collection, preprocessing, bias analysis and CNN-BiLSTM

model design, is presented in Section III. The results of the investigation are presented in Section IV, and a discussion of the results is given in Section V. The Conclusion and Future Work are in Section VI.

II. Related Works

The difficulties in Named Entity Recognition (NER) for languages like Setswana have been brought to light by the increased interest in low-resource languages for natural language processing (NLP). With an emphasis on CNN, BiLSTM, CNN-BiLSTM, and transfer learning models, this literature review examines current NER methodologies. The review reveals gaps in literature.

An estimated four million South Africans speak Setswana, a Bantu language that is one of the country's eleven official languages. It is the predominant national language of Botswana, where there are an additional two million speakers, and Namibia and Zimbabwe have few speakers too [1]. The South African government departments of Arts and Culture and Science and Innovation have collaborated to support several Human Language Translation (HLT) projects. These projects involve the creation of NLP resources in the form of data, core technologies, and software. Although these resources were created for ten South African languages, English being the exception, these ten languages, including Setswana, are still regarded as resource-poor, having comparatively little data that can be utilized to create reliable NLP applications and technologies. Many of these resources are available via the South African Centre for Digital Language Resources (SADiLaR) [2].

Semantic nuances in NER for South African languages, including Setswana, concentrate on linguistic, contextual, and graphemic characteristics that facilitate the identification and classification of named entities [4]. In addition to contextual, syntactic, and semantic nuances that help in the identification and recognition of entities like people, places, and organizations, the linguistic features are designed to record information pertaining to NER and can support tasks like entity coreference resolution and word case analysis [22]. Words that have numerous meanings depending on the context can be handled with the help of contextual features, which help determine the meaning of words based on their surrounding terms [23]. "Noka ya Limpopo" (Limpopo River), for example, designates "Limpopo" as a named entity (Location). Contextual cues are words that surround a concept and provide clues about its meaning. For instance, names that are indicated by titles, like "Mma" for women or "Rra" for men, help to identify named entities.

The three main kinds of named entities in the Setswana language context are nested, non-continuous, and continuous named entities. These categories can be arranged based on their textual structure [16]. An entity embedded inside another entity is called a nested named entity [16]. The phrase "Country, South Africa" is a location entity nestled inside another (Country being the geographical area of South Africa) in the statement "Naga ya South Africa" (Country of South Africa). Compound sentences and multi-layered language formulations are common examples of this complexity. Non-continuous named entities, which refer to the same actual thing but only occur once in the text before being broken up by subsequent text, are the next category. For instance, the entity "Sefofane" is a non-continuous named entity in the phrase "Sefofane sa South African Airways" (South African Airways airplane). Traditional NER systems can usually analyze simple non-continuous entities using sequence-labelled techniques, as they often only need one boundary tag and no additional connections. The final category is a continuous named entity, in which the same entity must be consistently marked over multiple tokens [16]. For example, Melawana e mešwa e phasaladitswe ke Banka ya Aforika Borwa. Badirisi ba tla solegelwa molemo ke ditlhokego tseno tse di tlisiwang ke banka eo. The Bank of South Africa has announced new regulations. That bank claims that these requirements will benefit consumers. The first entity to appear is Banka ya Aforika Borwa (Bank of South Africa), which is categorized as B-ORG (Organization at the beginning). The same entity is referred to as banka eo (that bank), which is classified as I-ORG (Organization within a phrase), which is a linguistic continuation. Continuous named entities are necessary to maintain the integrity of multi-word assertions or phrases.

Ten low-resource South African languages were used in the study, which evaluated neural network implementations of basic language technologies using the SaDiLaR NER dataset. With a particular focus on neural network models for POStagging and NER, this study reevaluated the baseline models that were already in place. Setswana NER's results on Conditional Random Fields (CRF) Baseline, bidirectional long short-term memory with auxiliary loss function which is called (bilstm-aux), and bisltm-aux emb(embeddings) were compared. The CRF Baseline obtained an F1-Score of 78.06%, followed by a F1-Score of 75.74% on bilstm-aux, and finally a f1-csore of 74.07% on bilstm-aux emb [2].

Another study was conducted on ten low-resource South African evaluations using deep learning transformer architecture models for NER. Following that these models were compared to other neural networks and machine learning. The transformer architecture models' F1-Score continuously outperformed the methods of machine learning and neural networks. The models that were assessed against one another were CRF, XML-R Base, XML-R Large, bi-LSTM-aux, and bi-LSTM-aux-emb. The letter R in XML stands for RoBERTa (Robustly Optimized BERT Approach). The F1 Score for CRF was 78.06%, the bi-LSTM-aux was 75.74%, the bi-LSTM-aux emb earned a F1-Score of 74.07%, and the XLM-R_{Base} and XLM-R_{Large} scored 78.70% and 79.54%, respectively, when the model's performance was assessed on Setswana [3].

In a study using a CNN model for Setswana NER, evaluated on the SADiLaR NER dataset, the model achieved an overall F1-Score of 94%, outperforming previously constructed baseline models tested on the same dataset [7].

Using two BiLSTM layers to extract hidden features from word representations, a study was carried out to identify Vietnamese named entities in sequence labeling tasks. The outcomes were better than the top models previously created for Vietnamese NER. With a score of 95.61%, the developed model received the highest rating [9].

A CRF baseline model was utilized to conduct a research NER system as part of the National Centre for Human Language

Technology (NCHLT) Text Phase II development project, which was aimed at creating and advancing HLT. The project aimed to develop protocols, automatic NER systems, and 15,000 tokens with named entity annotations for ten of South Africa's official languages, including Setswana. NER for Setswana achieved a F1-Score of 78.06% in this study. In further work, the CRF technique was applied to Setswana NER using a Setswana Regex Annotator (SERxA) for initial entity classification. This was followed by annotation using the BRAT tool. The study utilized a corpus of 1,000 news stories, achieving an overall F1-Score of 82%, which is highly impressive [4], [6].

There have been other studies on various languages that have utilized CNN-BILSTM, such as Indonesian NER, where three neural network-based model architectures for Low Complexity NER were studied. They made use of the GitHub datasets from Indonesian-ner and nlp-experiments. They also used multi-sequence BiLSTM-CNNs, BiLSTM-CNNs, and BiLSTM. Combining BiLSTM, single CNNs, and word2vec embedding yields the greatest results, with a f1 score of 71.37% [5].

By using the third SIGHAN Bakeoff MSRA dataset, a Chinese NER based on the CNN-BiLSTM-CRF model was developed and tested. The experimental results indicate that their model achieves 91.09% in F-scores without the need for hand-designed features or domain expertise [8].

According to a Decade Survey of Transfer Learning (2010–2020), transfer learning does not need to learn from start with a vast amount of data because its goal is to solve the target problem by utilizing the knowledge gained from source tasks in other domains. A survey on Transfer Learning emphasized the characteristics of Inductive Transfer Learning, Transductive Transfer Learning, and Unsupervised Transfer Learning. Regression and classification tasks, as well as labeled data in the target domain, are part of the two forms of inductive transfer learning: self-taught learning (source domain labels unavailable) and multi-task learning (source domain labels available). Transductive transfer learning focuses on problems like domain adaptation and covariate shift when source domain labels are known but destination domain labels are unknown. Lastly, unsupervised transfer learning is used for tasks like clustering and dimensionality reduction where source and target domain labels are not available. Each category addresses a different set of challenges in knowledge transfer between domains [10],[11].

In a study that focused on patient note de-identification, two tests were conducted using neural networks for NER, specifically the LSTM layer, and transfer learning. In the studies, using 5% of the dataset as the train set for transfer learning, results increase in the F1-Score of about 3.1% points, from 90.12 to 93.21. It is demonstrated that transfer learning outperforms state-of-the-art, indicating that the method is beneficial for a target dataset with a limited number of labels [12].

This section's discussion of baseline and neural network model research in the literature shows that low resource models still require attention, and that performance can still be enhanced. The baseline models are limited by the availability of training data, as Setswana has a significantly smaller number of annotated NER datasets than high-resource languages like English. These baseline models use datasets like the SaDiLaR NER dataset presented for Setswana model in this paper, which is a limited dataset. The literature reviews, particularly those focused on Setswana, highlight the need for hybrid transfer learning models, which have yet to be explored and evaluated in the language. Furthermore, by investigating the relationship between CNN, BiLSTM, and transfer learning as documented in the literature, this study contributes to the existing body of knowledge on/ the Setswana NER.

III. METHODOLOGY

The study's methodology, including data gathering and analysis procedures, is described in this section. Additionally presented and addressed in this part is the CNN-BiLSTM transfer learning model architecture. Afterwards, the software tools and libraries that are utilized are also covered here.

A. Data Collection

This research makes use of the South African Centre for Digital Language Resource (SADiLaR) and the National Centre for Human Language Technology (NCHLT) Setswana Named Entity Annotated Corpus. This was one of the initiatives to provide annotated data for government papers, where the data was compiled from gazetteers, publications, and the internet. This public dataset, created for the NER, POS Tag project, is accessible to everyone. There are 230 000 parallel words in the used dataset that have tags attached. The tags belong to the LOC, ORG, MISC, and OUT categories. The ORG tag designates the term as Organization, whereas the LOC tag designates the word as Location. The final tag is associated with terms that do not fit within the previously mentioned groups. MISC is the term for miscellaneous words.

The text entity tagging technique employed in this study, known as the "BIO tagging scheme," is shown in Table I. When a token appears inside a named entity but not at the beginning, it is labeled as I-label, and if it appears at the beginning, it is labeled as B-label. This is one of the most effective techniques to label entities [14].

TABLE I. BIO TAGGING SCHEME

Table Column Head		
Tags	***Meaning***	***Example***
B-PERS	**Begin-Person:** Marks the beginning of a person's name	"Shumile Chabalala" → Shumile: B-PERS, Chabalala: I-PERS
I-PERS	**Inside-Person:** Marks the continuation of a person's name, e.g. Surname (following B-PERS).	"Shumi Chabalala" → Shumi: B-PERS, Chabalala: I-PERS
B-ORG	**Begin-Organization:** Marks the beginning of an organization's name.	"Tshwane University of Technology" → Tshwane: B-ORG, University: I-ORG, Of: I-ORG, Technology: I-ORG
I-ORG	**Inside-Organization:** Marks the continuation of an organization's name (following B-ORG).	"Microsoft Incorporation" →Microsoft: B-ORG, Incorporation: I-ORG
B-LOC	**Begin-Location:** Marks the beginning of a location name.	"South Africa" →South: B-LOC, Africa: I-LOC
I-LOC	**Inside-Location:** Marks the continuation of a location name (following B-LOC).	"Soshanguve North" →Soshanguve: B-LOC, North: I-LOC
B-MISC	**Begin-Miscellaneous:** Marks the beginning of an entity that doesn't fit other categories.	"PSL 2024" →PSL: B- MISC, 2024: I- MISC
I-MISC	**Inside-Miscellaneous:** Marks the continuation of a miscellaneous entity (following B-MISC).	"Section 9" →Section: B- MISC, 9: I- MISC

B. Data Preprocessing

Two different methods are used to process the data for the five produced models. The latter two models process data using a sentence-level technique, while the first three models use a word-level technique.

To extract words and their associated tags for a NER, the world level technique processes the dataset line by line, with spaces separating words and their respective tags on each line of the dataset under processing. It separates lines that are not empty (contain data) into words and tags by attaching them to the appropriate tag lists and sentences.

In a sentence-level technique, words and tags are extracted from a text document using a structured format. Words and tags are separated by a tab ("\t"). Each line of the data is iterated over, with each word-tag pair being split and appended to the temporary lists sentence and tag. It adds the sentence and its tags to the sentences and tags lists, respectively, when it comes across a full stop (.), indicating that a sentence has ended. It then resets for the following sentence. This guarantees that sentences and the tags that go with them are grouped appropriately. If the last sentence doesn't end in a period, a final check is made to capture it. One list for sentences and another for the tags that go with them are returned by the process.

C. Dataset Biases

The SADiLaR NER Setswana dataset exhibits bias in language balance, locational representation, and entity distribution as shown in Fig. 1, Setswana dataset named entity distribution. Comparing the dataset to other categories like B-PERS, I-PERS are 3,251 and organizations B-ORG, I-ORG are 2,764, the number of miscellaneous "B-MISC, I-MISC" entities is substantially larger with 14,955 instances. Because of this imbalance, the model may perform worse since it is more likely to classify items as miscellaneous rather than correctly differentiating between people, places, and organizations. Additionally, location "B-LOC" entities exhibit regional bias, with most locations such as Aforika, Potchefstroom, and Mafikeng concentrated in South Africa and locations from other countries, such as the USA, India, and Brazil, appearing much less frequently. This implies that the dataset favors the geography of Southern Africa, which may restrict the model's applicability to other Setswana-speaking countries, such as Namibia and Botswana.

Additionally, as shown in Fig. 2 "Setswana Dataset Linguistic Bias", Setswana exhibits a significant preference over English, with 217,296 entities compared to English's 13,438 entities, according to the linguistic bias in named entities. This is in line with the dataset's goal of training a Setswana NER model, but it can cause problems in practical applications where "mixing English and Setswana" code-switching is regular. A model trained with this dataset may have trouble recognizing English entities or may not generalize well in situations when many languages are used.

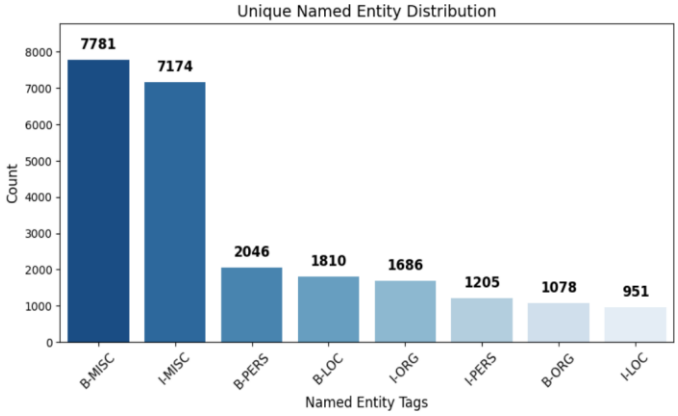

Fig. 1. Setswana dataset named entity distribution.

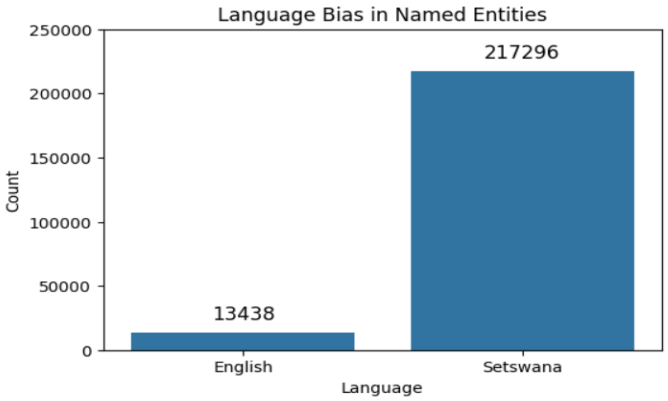

Fig. 2. Setswana dataset linguistic bias.

D. CNN-BiLSTM Hybrid Model

This section provides a detailed description and explanation of the CNN-BiLSTM architecture. This design is divided into two branches: the first branch displays the trained CNN model [6], while the second branch displays the recently created model. The CNN model parameters that have already been learned are coupled with the produced BiLSTM model. By changing the parameters of the specifically pre-trained CNN model developed for Setswana NER to the recently released BiLSTM model [2], which is displayed in Fig. 3 that illustrates the architecture of the suggested model, where the knowledge gained from the first model is applied to the second. To effect transfer learning, the CNN model that was trained on the same dataset as the suggested model is loaded, then the last classification layer is removed. Before incorporating the model into the new model, the pre-trained model layers are frozen to prevent any information from being lost during the training process. The model's workflow is as follows:

1) *Embedding layer*: Encodes the input tokens into dense vectors.
2) *Conv1D*: Extracts local contextual features.
3) *Dropout*: Mitigates overfitting.
4) *TimeDistributed (CNN Features)*: Expands feature representation.
5) *Transfer to sequential BiLSTM*: Processes sequential data to capture dependencies.
6) *Final TimeDistributed layers*: Classifies each token into NER tags.

The first layer of architecture uses a technique called Keras word embedding to transform words into dense vector representations that capture semantic information. Every token in the input sequence is transformed into a dense vector representation by this layer. Word indices, which are integer-encoded words, are sent into this layer, which then generates the matching embeddings. Keras is a Python module that operates on Tensorflow and is an open-source library. An open-source framework called TensorFlow is used to create deep learning applications [15].

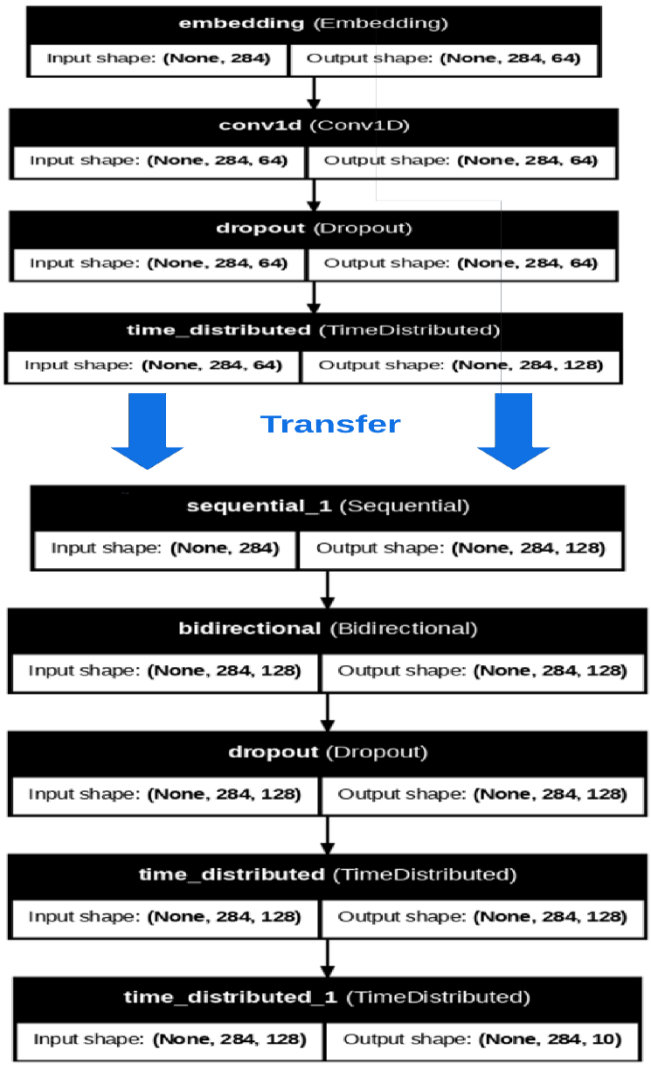

Fig. 3. CNN-BiLSTM hybrid architecture.

The CNN-BiLSTM Hybrid Architecture Fig. 3 shows the embedding input shape (None, 284), where "None" is the batch size and 284 is the sequence length. The output shape is (None, 284, 64), where 64 is the embedding size. The embedding equation is displayed below.

For each token t_i, it's embedding is:

$$\xi_t \in \square^{64}, \Xi = [\ \xi_1, \xi_2, \ldots, \xi_{284}\ ] \quad (1)$$

A convolutional layer employing a one-dimensional (Conv1D) CNN makes up the second layer of the model's design. The CNN Standard architecture standard, which includes the convolutional, pooling, and fully connected layers, is depicted in Fig. 4 to help visualize the layer. Before being sent to a fully connected layer, the implicit characters in the input data are fed into the pooling and convolution layers, where they are combined with gathered characteristics. In the final stage, the activation function processes the neuron's results [13].

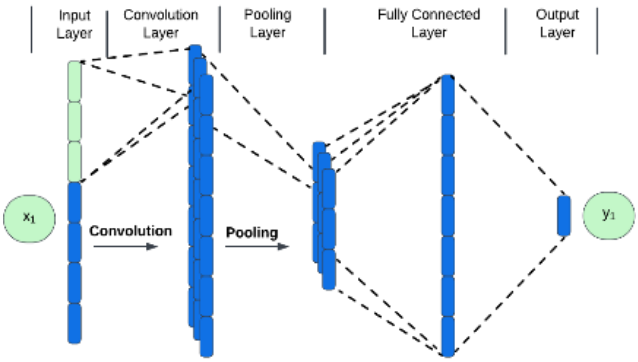

Fig. 4. CNN standard architecture (adopted from [7]).

Dropout, which stops feature detectors from simultaneously adjusting to the input space, is another method for tackling overfitting in big networks. When a classifier over adapts to the training set of data and performs poorly on untrained data, this is known as overfitting. Because of the time it takes for a network to settle to its ideal state and integrate, the dropout is introduced as a third layer that has no effect on the other layers. [20]. The model's use of dropout, which randomly sets some units to zero to avoid overfitting, is demonstrated in Eq. (2) below. The dropout mask m is applied elementwise:

$$Z' = m \odot Z, m \sim Bernoulli(p) \quad (2)$$

Where p is the dropout rate. The study uses the dropout rate of 0.5.

The TimeDistributed layer, the fourth layer employed in the study, applies a dense layer operation to each time step of a sequence separately. This is helpful since each input sequence comprises several time steps, and each time step requires the application of the same operation (dense layer).

$$h_t = f(W_h \cdot Z_t + b_h), W_h \in \mathbb{R}^{64X128} \quad (3)$$

Z_t is the embedding of a word in a sentence, with a dimensionality of 128.

W_h transforms the embedding into a hidden state h_t with dimension 64 capturing relevant contextual features for that word.

f is a tan function, commonly used in recurrent networks.

The hidden state h_t is then passed to the next layer which is the BiLSTM for further processing.

Equation overview:

Input: $z_h \in \mathbb{R}^{128}$

Weights: $W_h \in \mathbb{R}^{64X128}$

Bias: $b_h \in \mathbb{R}^{64}$

Output: $h_t \in \mathbb{R}^{64}$

The architecture's fifth layer, called BiLSTM, is made up of two LSTM layers placed next to each other. To record the past and future context of the sequence, the layer employs forward and backward LSTMs, as illustrated in Fig. 5 BiLSTM architecture. 64 units make up the return sequence, which is equivalent to the true.

The BiLSTM calculates:

$$h_t = concat(\overrightarrow{h_t}, \overleftarrow{h_t}) \quad (4)$$

where, $\overrightarrow{h_t}$ is a hidden state from forward LSTM and $\overleftarrow{h_t}$ is a hidden state from backward LSTM.

At the end of the model comes the Final TimeDistributed Layer, which assigns NER tags to every token. This layer functions as the model's classifier in essence. The layer's output shape is (None, 128, 9) with 9 representing the number of NER tags.

The probability at each timestep t are calculated by the SoftMax function:

$$p(y_t = c) = \frac{\exp(W_c.h_t + b_c)}{\sum_{c'} exp(W_{c'}.h_t + b_{c'})} \quad (5)$$

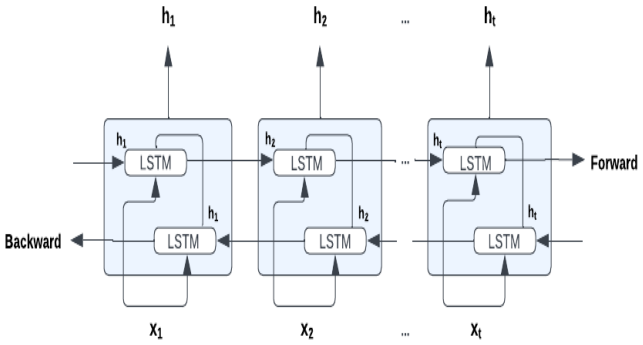

Fig. 5. BiLSTM achitecture (Adopted from [7]).

E. CNN-BiLSTM Model Algorithm Logic

Table II CNN-BiLSTM Model Algorithm, displays the NER method utilizing a hybrid CNN-BiLSTM model. The first step in the method is loading and preprocessing the dataset, which entails applying tokenizers and uniformity padding sequences to transform sentences and tags into numerical representations. After removing the final classification layer for transfer learning, a pretrained CNN model is loaded to extract spatial information. A BiLSTM layer is applied to the extracted features to record contextual data and temporal dependencies. The model iteratively adjusts weights depending on batches throughout training until convergence. Following the model's evaluation on the test set, metrics like as precision, recall, and F1-score are used to examine the findings, allowing for any necessary retraining or additional hyperparameter modification.

TABLE II. CNN-BILSTM MODEL ALGORITHM

Algorithm 1: CNN-BiLSTM Model Algorithm
Initialize
Load dataset, define hyperparameters, and preprocess
Compute
Create tokenizers, transform sequences, perform one-hot encoding and split the dataset into training and test sets
While (epochs not completed) do
For (each training batch) do
Train the CNN-BiLSTM model
Update weights of trainable layers
End For

Update and analyze
If validation accuracy improves, save the model weights.
Perform intermediate evaluations using metrics like accuracy, precision, recall, and F1-Score.
End
While
End

F. Performance Metrics

Performance metrics like precision, recall, and F1-Score are used to analyze the results once the model has been evaluated on the test set, as shown in the preceding section. This enables any necessary retraining or further hyperparameter change.

Using the metrics from the calculated formulas on Eq. (6), (7), and (8), the model generates a classification matrix. The F1-score, Accuracy, and Recall are determined by utilizing the true positives (TP), false negatives (FN), and false positives (FP). Appropriately defined situations are considered true positive. False positives are instances that were incorrectly labeled, and false negatives are instances that the system failed to detect. The F1-Score is the weighted mean of Precision and Recall [21]. These metrics are generated in the manner described below equations:

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

$$F = 2 * \frac{precision*recall}{precision+recall} \quad (8)$$

IV. Results

The results of the Setswana NER models are presented in this part along with an explanation of how evaluation indexes were used to calculate precision, recall, and F1-Score values. A classification report also includes these evaluation indices of named entities' performance on each of the five models created for the study.

A. Data Distribution

The study experiment's dataset splits 20% and 80% into test and training datasets, respectively. Using the train_test_split function from the sklearn library's model_selection module, it was split and randomized.

B. Performance Metrics

These results indicate the performance of different models in terms of Precision, Recall and F1-Score as shown in Table III Models 3, 4 and 5 have the maximum training accuracy of 99% although both Models 3 and 5 have lower average precision than Model 4.

C. Evaluation Measures

Table IV presents performance indicators that illustrate the accuracy and loss for training and validation across different models. Among them, Model 4, which integrates CNN and BiLSTM at the sentence level, demonstrates strong generalization ability. It achieves a well-balanced accuracy and loss, recording the lowest validation loss (0.0093) and the highest validation accuracy (0.9976).

TABLE III. Performance Comparison for 5 Developed NER Models

5 Setswana NER models				
Model	***Precision***	***Recall***	***F1-Score***	
Model 1: CNN-Word Level			**0.94**	**Accuracy**
Model 1: CNN-Word Level	0.75	0.59	0.65	Macro avg
Model 1: CNN-Word Level	0.93	0.94	0.93	Weighted avg
Model 2: CNN-BiLSTM transfer learning Word Level			**0.96**	**Accuracy**
Model 2: CNN-BiLSTM transfer learning Word Level	0.83	0.71	0.75	Macro avg
Model 2: CNN-BiLSTM transfer learning Word Level	0.95	0.96	0.95	Weighted avg
Model 3: CNN-Sentence Level			**0.99**	**Accuracy**
Model 3: CNN- Sentence Level	0.82	0.65	0.72	Macro avg
Model 3: CNN- Sentence Level	0.99	0.99	0.99	Weighted avg
Model 4: CNN-BiLSTM transfer learning Sentence Level			**0.99**	**Accuracy**
Model 4: CNN-BiLSTM transfer learning Sentence Level	0.85	0.70	0.76	Macro avg
Model 4: CNN-BiLSTM transfer learning Sentence Level	0.99	0.99	0.99	Weighted avg
Model 5: CNN-BiLSTM Sentence Level			**0.99**	**Accuracy**
Model 5: CNN-BiLSTM Sentence Level	0.78	0.69	0.73	Macro avg
Model 5: CNN-BiLSTM Sentence Level	0.99	0.99	0.99	Weighted avg

TABLE IV. PERFORMANCE INDICATORS

5 Setswana NER models Performance Indicators				
Model	***Training Accuracy***	***Validation Accuracy***	***Training Loss***	***Validation Loss***
Mode l : CNN-Word Level	0.9416	0.9601	0.2500	0.1165
Mode 2: CNN-BiLSTM Transfer Learning-Word Level	0.9566	0.9600	0.1323	0.1364
Mode 3: CNN-Sentence Level	0.9971	0.9937	0.0076	0.0315
Mode 4: CNN-BiLSTM Transfer Learning-Sentence Level	0.9983	0.9976	0.0046	0.0093
Mode 5: CNN-BiLSTM-Sentence Level	0.9997	0.9926	0.0012	0.0474

V. DISCUSSIONS

A discussion of the findings and a comparison of our model's results with those of the baseline models are presented in this section.

A. Performance Metrics Discussion

Models 3, 4, and 5 scored the greatest accuracy of 0.99 among the five Setswana NER models assessed, as from the results. Model 4 (CNN-BiLSTM transfer learning at the sentence level) scored better than the others for macro-average measures (accuracy of 0.85, recall of 0.70, and F1-Score of 0.76), which take class imbalance into consideration. The weighted averages for Models 3, 4, and 5 are extraordinarily high, with each model earning 0.99 for precision, recall, and F1, which represents the overall model performance across all classes. As the top-performing model for Setswana NER tasks, Model 4: CNN-BiLSTM transfer learning Sentence Level is suggested due to its high accuracy, weighted averages, and improved macro-average performance.

B. Evaluation Measures Discussion

When compared to sentence-level models, the performance indicators in Table IV demonstrates that Model 1 (CNN-Word Level) has the lowest validation accuracy (0.9601) despite achieving comparatively high accuracy. The training accuracy of Model 2 (CNN-BiLSTM Transfer Learning-Word Level) is somewhat higher than that of Model 1 (0.9566), while the validation accuracy is similar (0.9600). This suggests that while BiLSTM and transfer learning enhance training performance, they have no noticeable effect on word-level validation accuracy. When modeling sentence-level settings, Model 3 (CNN-Sentence Level) performs better, as seen by its significantly greater accuracy (0.9971 training, 0.9937 validation). The second highest accuracy for both training (0.9983) and validation (0.9976) is attained by Model 4 (CNN-BiLSTM Transfer Learning-Sentence Level), suggesting that integrating BiLSTM with sentence-level modeling and transfer learning significantly improves performance. Although Model 5 (CNN-BiLSTM-Sentence Level), the final model, has the highest training accuracy (0.9997), it may be overfitting because its validation accuracy (0.9926) is marginally lower than Model 4.

Model 4 (CNN-BiLSTM Transfer Learning-Sentence Level) is the most successful model when accuracy and loss are balanced since it obtains the lowest validation loss (0.0093) and the highest validation accuracy (0.9976), indicating good generalization capabilities.

C. Comparison with Previous Work

The models demonstrate different strengths in recall, F1-Score, and overall efficacy for Setswana NER, according to the performance comparison with baseline models that are already in the literature, as indicated in Table V, performance comparison with other existing work in Setswana NER. With an F1-Score of 78.06%, the Conditional Random Fields (CRF) baseline performs well overall, making it a solid benchmark. The BiLSTM-aux models have somewhat lower F1-Scores; their moderate efficacy is shown by their 74.07% BiLSTM-aux emb score. The CNN NER model in [7] shows inconsistent predictions, performing poorly in F1-score (62%) but well in recall (94%). In contrast, the CNN-BiLSTM model achieves a competitive F1-score (71%) while also improving recall (96%) [7]. But the CNN-BiLSTM transfer learning sentence-level model that was suggested works better than the others, with the highest recall (99%) and a high F1-Score (70%). The CNN-BiLSTM transfer learning sentence-level model is therefore suggested as the top-performing model for Setswana NER because of its competitive overall performance and superior recall.

TABLE V. PERFORMANCE COMPARISON WITH OTHER EXISTING WORK IN SETSWANA NER

Comparison Analysis				
Model	***Dataset***	***Recall***	***F1-Score***	
Conditional random fields (CRF) baseline NER for Setswana (Loubser, M. and Puttkammer, M.J., 2020)	NCHLT Setswana Named Entity Annotated Corpus	80.86%	75.47%	78.06%
bilstm-aux (Loubser, M. and Puttkammer, M.J., 2020)	“	74.14%	77.42%	75.74%
bilstm-aux emb (Loubser, M. and Puttkammer, M.J., 2020)	“	73.45%	74.71%	74.07%
CNN NER Model for Setswana NER (Chabalala, S., Owolawi, P. and Ojo, S., 2023)	“	77.00%	62.00%	94.00%
CNN-BiLSTM NER Model for Setswana (Chabalala, S., Owolawi, P. and Ojo, S., 2024)	"	83.00%	71.00%	96.00%
Our CNN-BiLSTM transfer learning Sentence Level	"	85.00%	70.00%	99.00%

VI. Conclusion

For the Setswana Language NER, the study suggested five models. The first two models, CNN and CNN-BiLSTM Transfer learning, are all based on word terms followed by the final three models, CNN, CNN-BiLSTM Transfer learning, and CNN-BiLSTM hybrid, which are all based on sentence-level. The evaluation was conducted on the South African Centre for Digital Language Resources (SADiLaR) NER dataset, and the top-performing model was ultimately compared to the state-of-the-art Setswana NER models that had already been published in the literature and were created using the same dataset. The top-performing model is Model 4 (CNN-BiLSTM Transfer Learning-Sentence Level), which attains the best macro averages, outstanding weighted averages, and high accuracy 99%. All classes, including minority ones, are balanced in terms of generality. As a result, sentence-level models perform better than word-level models since they can gather more contextual data, which enhances their efficiency. This model outperforms the state-of-the-art models in terms of accuracy (99%) and recall (85%), indicating its high precision and capacity to accurately identify most entities. Despite having a little lower macro average F1-Score (70%) than other models, its total performance, especially the accuracy and recall combination, makes it the best.

In addition to performing better than the other four models in this work, including models from literature, the suggested model has certain drawbacks because of its short dataset size, since deep learning algorithms often require huge datasets. Even though this work attempted to address this constraint through transfer learning, it was insufficient to completely address it; therefore, to further address this limitation, we recommend future research on the two domains indicated below as well as on the other fields.

Investigating cross-lingual transfer learning techniques may be valuable given the limitations of the current dataset and the generally scarce resources for the Setswana language. This study focused exclusively on NER in Setswana, with an emphasis on the South African Centre for Digital Language Resources (SaDilar) NER dataset, allowing for comparison with previous research conducted on the same dataset. Therefore, using tagged data from resource-rich languages could significantly improve the model's functionality in Setswana NER.

Error analysis: To identify any trends or problems the model shares, an error analysis can be conducted to inform future enhancements.

Consequently, it may be beneficial for future study to apply transformers, multilingual models like BERT, RoBERTa, or XLM-R, include larger and more varied datasets, and further optimize hyperparameters.

Dataset bias: The study has demonstrated that the dataset contains biases. Future studies should examine rebalanced datasets by adding more diverse named entities to alleviate these biases, particularly in underrepresented categories such as B-LOC, B-PERS, and B-ORG. Adding more foreign locales and English-Setswana mixed items to the dataset would also increase its robustness.

Acknowledgment

I would want to thank my family from the bottom of my heart for their unwavering support during this work.

References

[1] W. G. Bennett, M. Diemer, J. Kerford, T. Probert, and T. Wesi, ‘Setswana (South African)’, Journal of the International Phonetic Association, vol. 46, no. 2, pp. 235–246, 2016.

[2] M. Loubser and M. J. Puttkammer, ‘Viability of neural networks for core technologies for resource-scarce languages’, Information, vol. 11, no. 1, p. 41, 2020

[3] R. Hanslo, "Deep Learning Transformer Architecture for Named-Entity Recognition on Low-Resourced Languages: State of the art results," in 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS), Sofia, Bulgaria, 2022.

[4] R. Eiselen, ‘Government domain named entity recognition for South African languages’, in Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16), 2016, pp. 3344–3348.

[5] S. Sukardi, M. Susanty, A. Irawan, and R. F. Putra, ‘Low complexity named-entity recognition for Indonesian language using BiLSTM-CNNs’, in 2020 3rd International Conference on Information and Communications Technology (ICOIACT), 2020, pp. 137–142.

[6] B. Okgetheng and G. Malema, ‘Named Entity Recognition for Setswana Language: A conditional Random Fields (CRF) Approach’, in Proceedings of the 2023 7th International Conference on Natural Language Processing and Information Retrieval, 2023, pp. 240–244.

[7] S. Chabalala, P. Owolawi, and S. Ojo, ‘A Convolutional Neural Network Model for Setswana Named Entity Recognition’, in International Conference on Artificial Intelligence and its Applications, 2023, pp. 165–171.

[8] Y. Jia and X. Xu, ‘Chinese named entity recognition based on CNN-BiLSTM-CRF’, in 2018 IEEE 9th international conference on software engineering and service science (ICSESS), 2018, pp. 1–4.

[9] N. C. Lê, N.-Y. Nguyen, A.-D. Trinh, and H. Vu, ‘On the Vietnamese name entity recognition: A deep learning method approach’, in 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), 2020, pp. 1–5.

[10] S. J. Pan and Q. Yang, ‘A survey on transfer learning’, IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1345–1359, 2009.

[11] Niu, Y. Liu, J. Wang, and H. Song, ‘A decade survey of transfer learning (2010--2020)’, IEEE Transactions on Artificial Intelligence, vol. 1, no. 2, pp. 151–166, 2020.

[12] Lee, J.Y., Dernoncourt, F. and Szolovits, P., 2017. Transfer learning for named-entity recognition with neural networks. arXiv preprint arXiv:1705.06273.

[13] Shan, L., Liu, Y., Tang, M., Yang, M. and Bai, X., 2021. CNN-BiLSTM hybrid neural networks with attention mechanism for well log prediction. Journal of Petroleum Science and Engineering, 205, p.108838.

[14] Zhang, Z. Chen, D. Liu, and Q. Lv, ‘Building Structured Patient Follow-up Records from Chinese Medical Records via Deep Learning’, in 2022 2nd International Conference on Bioinformatics and Intelligent Computing, 2022, pp. 65–71.

[15] J. Brownlee, Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras. Machine Learning Mastery, 2016.

[16] I. Keraghel, S. Morbieu, and M. Nadif, ‘A survey on recent advances in named entity recognition’, arXiv preprint arXiv:2401. 10825, 2024.

[17] B. Okgetheng and G. Malema, ‘Named Entity Recognition for Setswana Language: A conditional Random Fields (CRF) Approach’, in Proceedings of the 2023 7th International Conference on Natural Language Processing and Information Retrieval, 2023, pp. 240–244.

[18] S. A. Stats, ‘Community survey 2016 in brief’, Statistics South Africa. Pretoria: SSA, 2016.

[19] R. Letsholo and K. Matlhaku, ‘The syntax of the Setswana noun phrase’, Marang: Journal of Language and Literature, vol. 24, pp. 22–42, 2014.

[20] J. van de Kerkhof, "Convolutional neural networks for named entity recognition in images of documents," Stockholm Sweden, 2016

[21] A. Goyal, V. Gupta, and M. Kumar, 'Recent named entity recognition and classification techniques: a systematic review', Computer Science Review, vol. 29, pp. 21–43, 2018.

[22] V. Harrison and M. Walker, 'Neural generation of diverse questions using answer focus, contextual and linguistic features', arXiv preprint arXiv:1809. 02637, 2018.

[23] T. T. H. Hanh, A. Doucet, N. Sidere, J. G. Moreno, and S. Pollak, 'Named entity recognition architecture combining contextual and global features', in International Conference on Asian Digital Libraries, 2021, pp. 264–276.