INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Named Entity Recognition for Sheko Language Using Bidirectional LSTM

Desalegn Ashebir[1], **Prabhakar Gantela**[2]*

1 Department of Software Engineering, Mizan Tepi University, Ethiopia
2 Department of Computer Science and Engineering, Mizan Tepi University, Ethiopia

OPEN ACCESS

Received: 20-03-2022
Accepted: 09-05-2022
Published: 24-06-2022

Citation: Ashebir D, Gantela P (2022) Named Entity Recognition for Sheko Language Using Bidirectional LSTM. Indian Journal of Science and Technology 15(23): 1124-1132. https://doi.org/10.17485/IJST/v15i23.642

* **Corresponding author**.

gantelaprabhakar@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: This study aims to advance Sheko language name entity Recognition first of its kind. Named Entity Recognition (NER) is one of the most important text processing in machine translation, text summarization, and information retrieval. Sheko language named entity recognition concerns in addressing the usage of the bidirectional Long Short-Term Memory (LSTM) model in recognizing tokens into predefined classes. **Methods**: A bidirectional long short-term memory is used to model the NER for sheko language to identify words into seven predefined classes: Person, Organization, Geography, Natural Phenomenon, Geopolitical Entity, time, and other classes. As feature selection plays a vital role in long short-term memory framework, the experiment is conducted to discover the most suitable features for Sheko NER tagging task by using 63,813 words to train and test our model. Out of which is 70% for training and 30% for testing. Datasets were collected from Sheko Mizan Aman Radio Station (MARS), Sheko southern region mass media, Language, and Literature Department. **Findings:** Through several conducted experiments, Sheko NER has successfully achieved a performance of 97% test accuracy. From the experimental result, it is possible to determine that tag context is a significant feature in named entity recognition and classification for Sheko language. Finally, we have contributed a new architecture for Sheko NER which uses automatically features for Sheko named entity recognition which is not dependent on other NLP tasks, and added some preprocessing steps. We provide a comprehensive Comparison with other traditional NER algorithms.

Keywords: Named Entity Recognition; Sheko language; Recurrent Neural Network; Bidirectional layer; embedding layer

1 Introduction

With the advancement of electronic media, more and more regional language text is now being produced [(1)]. Mining relevant information from unstructured electronic data has become a difficult task for organizations such as newspapers, medical institutions, educational institutions, and others, and text information cannot be used directly by natural language processing tasks to achieve predictable results. The application of entity name recognition made it simple to solve the above challenge. Information extraction is an important pre-processing approach that greatly improves text mining

potential.[1,2]. Named Entity Recognition (NER) is one of the most important aspects of data extraction. NER is an information extraction subtask that detects and categorizes named entity references in unstructured text into pre-defined classes such as people's names, places, organizations, time expressions, quantities, and monetary values.[3,4]. It was also regarded as one of the most important preprocessing procedures in domains like as question answering, entity linking, and machine translation[5]. Sheko language is a member of the Omotic branch of the Afro-Asiatic language family[6]. It is spoken in southern Ethiopia, particularly in the Gambella Region and the Ethiopian Southern Nations, Nationalities, and Peoples' Region's Bench Maji Zone. Sheko, also known as Shak, Shako, Sheka, Shekka, Shekko, or Tschako, had roughly 143,800 speakers in 2019. Tepi and Guraferda are two dialects[6].

In 2009, a method of writing Sheko language using the Latin alphabet was introduced, and educational materials based on this orthography were developed. Before that, some people used the Ethiopic script to write Sheko. Sheko is a language that is taught in select primary schools and broadcast on the radio[6]. · Furthermore, educational institutions, the media, and the healthcare industry are all expanding and in need of computational resources in the language. The lack of a structured corpus makes it difficult to conduct NLP research on this language.

The current study is likely to play a part in resolving such a dilemma first kind. However, there are three types of NER models: Rule-based, Machine learning, and Hybrid techniques. Handcrafted local language rules underpin the rule-based method. The rule-based approach has several disadvantages, including the need for a linguistic expert to identify all of the rules, the time-consuming effort of establishing those rules, lack of portability, and high maintenance costs when new rules are added and old rules are removed.[7]. The machine-learning approach uses NEs to extract some features from the prepared corpus and then learns based on those feature sets. Machine learning is a simple, versatile, and easy-to-maintain method.[7]. Data were collected from three separate sources and preprocessed so that it could be fed into the BiLSTM model in a machine-readable format. NER with the core component that takes plain text as input, then performs entity identification, the features that are extracted and stored during the training phase are supplied to the recognizer to identify NERs from the text. Despite a large number of Named Entity classes, our model can only classify seven types of entities: person, organization, geography, time, geopolitical, natural, and other.

Sample text in Sheko Kookn shadnka daanta kib S'oku yaabonka, dacha ítira sasku qarnsab bengiqa Bengs náng təkə azqa ń qaynbab, náng s'oon qoy ant faadu ń kaysnbab. unabab qyaast qarns footut ,ń dadu tamarstə ń noogu arut wota saanta ń baznbab ń dadukn maakut, unaka haaqastaka bazxabara sesutə ń noogu ń angusə qoy footutə·[6]

Sample ENR for Sheko NER in Sheko language is becoming harsher and more exciting as compared to English language, Amharic, and Afan Oromo. For instance, in this sentence Kookn shadnka daanta kib S'oku yaabonka, dacha ítira sasku qarnsab bengiqa. This statement is equivalent to the English version: Sheko people in far places and nearby in the new era which got you there suitably the year is ours. Here, **Kookn** is name of a Geography, **shadnka** is the name of time, **daanta** is the name of Time, yaabonka is the name of person, dacha is another class, **bengiqa** is the name of a Natural Phenomenon but the first letter does not block and **hib** is a verb in the sentence.[6]

We have also presented different experiments in Sheko's text and analyzed errors in evaluation results. In this study, we used the bidirectional LSTM technique for the development of the named entity recognizer system for the Sheko language

In our study, we concentrated on six named entities, and if an entity is not among the seven NEs, it is labeled as "others." We also attempt to provide the uniqueness of Sheko texts as well as an examination of the experimental evaluation results for flaws. We employed the recurrent neural network approach in this investigation. There has been a lot of research done on NERs. The traditional solution to the NER problem mainly includes three methods: Rule-based method, Statistics-based method, and Dictionary-based method[8]. The method based on rules and dictionaries requires professional linguists to write rules by hand, requires a lot of time, and has poor portability in different areas. In Recurrent neural network-based entity name recognition the accuracy rate is improved compared with the method based on rules and dictionaries

[9–11]. Researchers began to use deep learning for NER tasks as deep learning technology progressed. In comparison to traditional models, neural network models can learn more semantic feature information and do not need feature engineering or domain knowledge. However, currently, the Bi LSTM-GRU model, boosts entity recognition accuracy[12].

In[13] introduced an approach that combines Conditional Random Field (CRF) and Bi-RNN with Support Vector Machine (SVM) and outperformed the baseline by a large margin. The major factor that hindered the performance lies in the redundant or missing modifier of the entity to be recognized. Likewise, the neural network architecture is presented in Chen L's work. In[14] the lack of unambiguous labels in Chinese text made it difficult to establish the borders of words, they proposed a NER model based on character-word vector fusion. The provided model was shown to lessen reliance on the word segmentation algorithm's accuracy and make the greatest use of the semantic features of the words. To solve the issue of failing to accurately identify an entity's border, they maintained the entity's integrity and accuracy by storing text information in both directions. Muralikrishna et al.[2] proposed a model to identify spoken language, a bidirectional LSTM-based technique. They employed two datasets for

this experiment: the Oregon institute's multi-language telephone speech corpus, which has 450 training samples and contains 10 Indian languages for utterances and the duration of hours for each language. Second, they used 13 Indian languages from an Indian institute of technology.

In [15] , followed an approach for nested entity name recognition, a Transfer-Learning based technique, with pertained BERT being utilized for recognition. They conducted the experiment using three datasets: GENIUS, Germ Eval 2014, and JNLPBA, with 16,992, 26,202, and 20,546 sentences used to train and test the model respectively. They achieved 74.30 % test accuracy from the GENIUS dataset, 85.92% test accuracy from the Germ Eval 2014 dataset, and 80.68 % test accuracy from the JNLPBA dataset.

The authors of [9] used Chinese character strokes as features and proposed the stroke n-gram model, which not only excavated the feature information of Chinese character strokes but also more effectively used the semantic information of Chinese characters to train word vectors. They Suggested solving the ambiguity of Chinese words and the lack of word boundaries and proposed a novel Complexity character-level representation method to capture the semantic information of Chinese characters. [16] Proposed Anyuak Language Named Entity Recognition Using Deep Learning Approach to detect and classify words into five predefined classes: Person, Time, and Organization reported that result of precision, recall, and F1-measure values of 98%, 90, and 94% respectively. But for Location and Others (non-named entity words) they didn't include compressive experiments like single bidirectional two bidirectional LSTM.

In [17], a hybrid approach combining rule-based and machine learning techniques was used to create NER for the Afan Oromo language. In two scenarios, they acquired 77.41 % recall, 75.8% precision, and an F1 measure of 76.6% from their experimental results. The authors describe a rule generation strategy that allows the NER model to learn the structure of named entities in the Afan Oromo language. The NER methods presented by the authors are more reliant on the nature of features than on increasing the quantity of the training data set, although features are not automatically trained. Colbert uses a neural network forward feeds model to recognize and classify named entities using a limited number of contextual phrases. In [3] proposed a bidirectional LSTM technique for predicting positive or negative sentiment in Arabic text. Sentiment analysis was based on six benchmark datasets and a total of 61,582 datasets to evaluate the model's performance. Finally, they compared the model to CNN and LSTM approaches and found that the sentiment prediction accuracy in Arabic text was 83.15 % on an average from the six datasets.

Our paper has the following contributions: Technically simple architecture is proposed that utilizes feature embedding which focuses on relevant parts of the current token in a sequence of words to efficiently recognize entities. Effective word representation by incorporating normalization, cleaning, and the design of a high-level feature embedding bi-LSTM, that effectively captures the local and global information of a token. Finally, a Comparison with CRF and LSTM based on single and two bidirectional layers. We studied the impact of each module (size of embedding layer, and LSTM unit) and the effectiveness of their combinations.

2 Material and methods

2.1 Annotation: Named entity tagging

We have annotated the dataset with seven tags; named entity tags Person, Organization, geography, Natural Phenomenon, Geopolitical Entity, time, and other Classes, It was carried out by a linguist with extensive knowledge and experience. The part of speech tagging algorithm takes input in the form of a phrase's tokens and assigns Parts of Speech (POS) tags to the words in that sentence. It provides a token's grammatical structure depending on the token's syntax and the semantics surrounding it.

2.2 Pre-processing

Tasks took place on the whole dataset. These phases are comprised of Cleaning, Tokenizer, and normalizer modules.

Cleaning

Input: raw unprocessed Sheko_NER dataset

Output: processed clean Sheko_NER dataset

INIT: **1**. Read Sheko_NER dataset

While (it is not the end of file):

If text contain special characters and symbols [')', '(', '-', '|', '$', '&', '/', '[', ']', '>','%', '=', '#', '*', '+', '\\',
'● ', '~', '@', '£', '_', '{', '}', '©', '^', '®', '°', '<', '→', '○', '€', '™', '›', '←', '×', '§"]

Then Remove characters

If the text contains emojis [☒, ☒, ☒, ☒…]

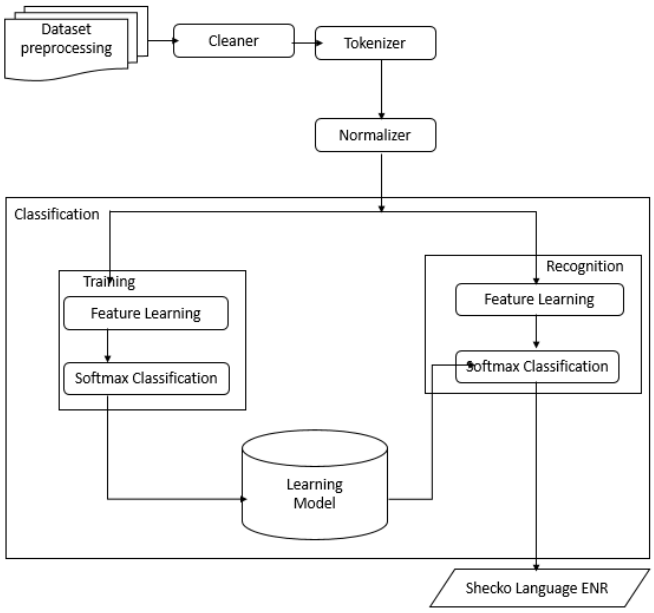

Fig 1. High-level architecture of the proposed system

Then Remove emojis
If the text contains Nan (not a number)
Then Drop Nan and fill na
Return processed text
End if
Normalization
There are many words with the same meaning but have different written forms; words must be in grammatical form
Input: unnormalized Sheko_NER dataset
Output: normalized Sheko_NER dataset
INIT:
Read Sheko_NER dataset
WHILE (it is not the end of the file :
IF dataset contains [ayi], **THEN**
Replace with 'nayi'
IF dataset contains [S'oku], **THEN**
Replace with 'Sheko'
IF dataset contains [Shak], **THEN**
Replace with 'Sheko"
IF dataset contains [saanta], **THEN**
Replace with 'wota saanta'
ENDIF

2.3 Feature learning

To train a neural network, first, we will use two mappings,{token} to {token id}: address the row in the embedding's matrix for the current token.{tag} to {tag id}: one-hot ground truth probability distribution vectors for computing the loss at the network's output.

Kookn shadnka daanta kib S'oku yaabonka,tokenize into "Kookn","shadnka","daanta"," kib "S'oku","yaabonka",

The second step is to transform columns to extract sequential data into sequential arrays that Split the dataset into a train, and test after padding. Padding: The LSTM layers accept sequences of the same length only. Therefore, every sentence represented as integers ('Word_idx') must be padded to have the same length. We have worked with the largest sequence's maximum length

and pad the shorter sequences. Thirdly, construct the architectural model. To do so, first, design the architecture and determine the input and output dimensions for each layer. RNNs can handle a variety of input and output combinations. **Table 1** shows that for this request, many-to-many architecture is used. At each step, we must output a tag (y) for a token (X) ingested. To output, we generally use three layers (embedding, bi-lstm, and lstm layers) and a fourth layer (Time Distributed Dense layer).

Layer 1 — the maximum length (104) of the padded sequences is specified in the embedding layer. After the network has been trained, the embedding layer will convert each token into an n-dimensional vector. We've picked n as the number of dimensions (64)

Layer 2— Bidirectional LSTM: A recurrent layer (the first LSTM layer) is passed as an argument to bidirectional LSTM. The output from the preceding embedding layer is used in this layer (104, 64).

Layer 3 — LSTM Layer: This layer accepts the output dimension (? 104, 128) and outputs (? 104, 256) from the previous bidirectional LSTM layer.

Layer 4 — Time Distributed Layer: This layer takes the previous LSTM layer's output dimension (104, 256) and outputs the maximum sequence length (104) and maximum tags (max tags) of (7).

3 Results and Discussion

In this section, the dataset and implementation of the Sheko NER model are described and also compared in terms of recall, precision and F1 score. Datasets was collected from Sheko Mizan Aman Radio Station (MARS), Sheko southern region mass media, Language, and Literature Department, and prior research have been used. 4254 sentences, each sentence contains 12-15 words, having a total of 63,813 words were used to train and test our model. Out of which 70 % were used for training and 30% for testing. The proposed model is implemented with Keras (Tensor Flow as a backend) using Python programming language. The experiment performed used 50 epochs, max length 104, LSTM unit 32, and recurrent dropout of 0.1. Since this parameter is selected by empirical optimization (manual searching)

Comparison of LSTM with two bi-directional layers in Sheko ENR

Network type recurrent dropout LSTM unit size of Embedding layer max length accuracy 0 32 100

Comparison of LSTM with a single bi-directional layer in Sheko ENR

The performance of Sheko ENR in the training and validation phase is compared. In addition, the performance of single directional LSTM and Bidirectional LSTM models were evaluated. The training process indicates that as the number of epochs rises, the training accuracy improves while the loss decreases. The loss is the sum of errors for each sample in the training and validation sets. The better the model and recognition outcome, the lesser the loss.

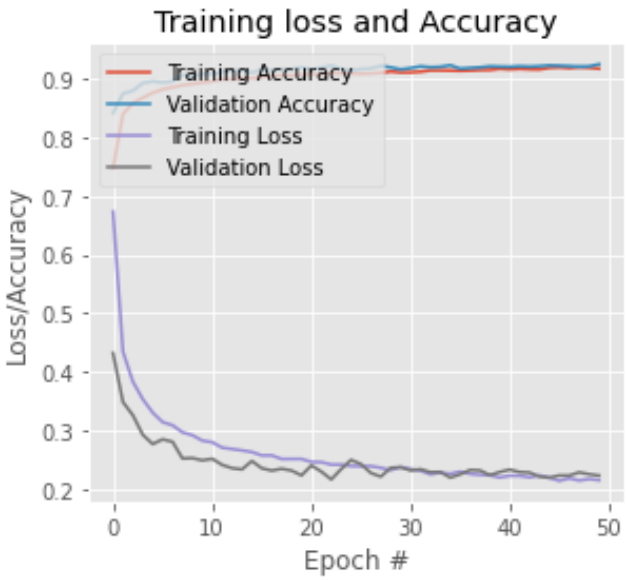

Fig 2. LSTM with two bi-directional layer curve recurrent dropout=0.1

As we depicted in **figure 2** the proposed Sheko language entity name recognition in the case of LSTM with two bidirectional layer training accuracy increased linearly and validation accuracy oscillates a little and reached 93% and finally achieved 97% test accuracy. Training accuracy increased linearly and passed over 98% after epoch 48, while the validation accuracy oscillates a little and reaches 98% accuracy in some instances. The training loss also decreases intensely down to 0.76, while validation loss oscillates and reaches 0.8. This experiment clearly shows that applying two bidirectional LSTM improves the learning capability of the proposed Sheko ENR system.

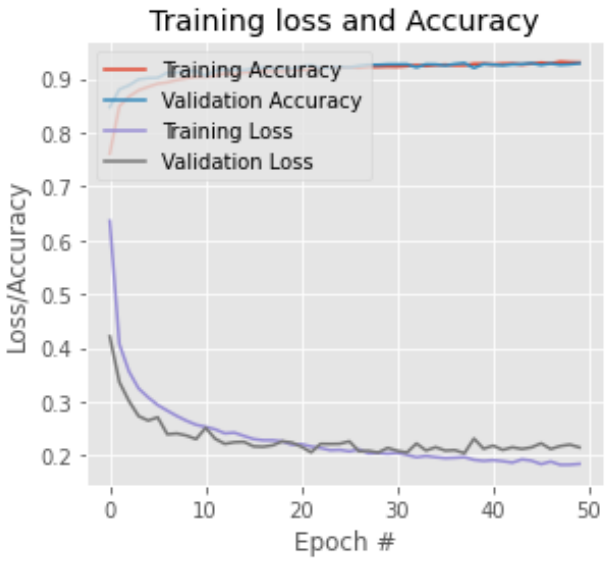

Fig 3. LSTM with two bi-directional layer curve recurrent dropout=0

As depicted in the training loss and accuracy curve in Figure 3, the training accuracy is above 90% for many epochs. The validation accuracy is lower than training accuracy throughout the curve. However, the gap is higher than LSTM with two bi-directional layer curve recurrent dropout=0.1 (figure 3). The gap between the training loss and validation loss is visible, which is a typical sign of overfitting occurrence though the gap is too small. The validation loss was very high throughout the curve.

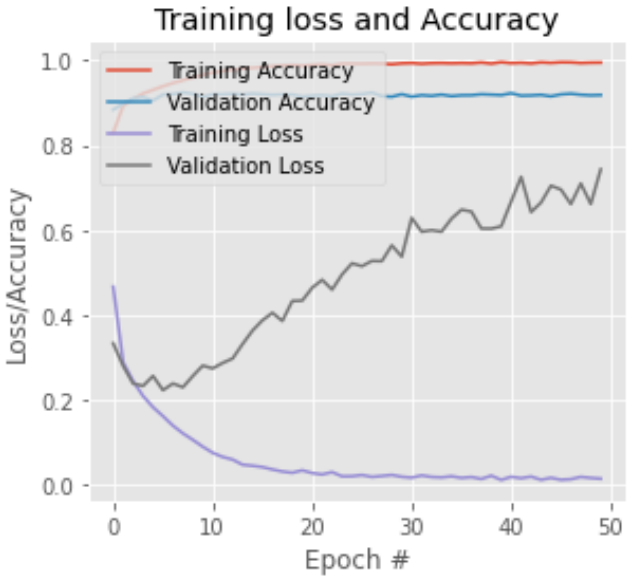

Fig 4. LSTM with single bi-directional layer curve recurrent dropout=0.1

As shown in the training loss and accuracy curve in Figure 4, training and validation accuracy increases while training and validation loss decreases nearly linearly up to epoch 40. Between epochs 45 and 48, training and testing accuracies decrease too much and run constantly while training and testing loss increase and remain constant respectively. These two signs are a typical indication of overfitting occurrences.

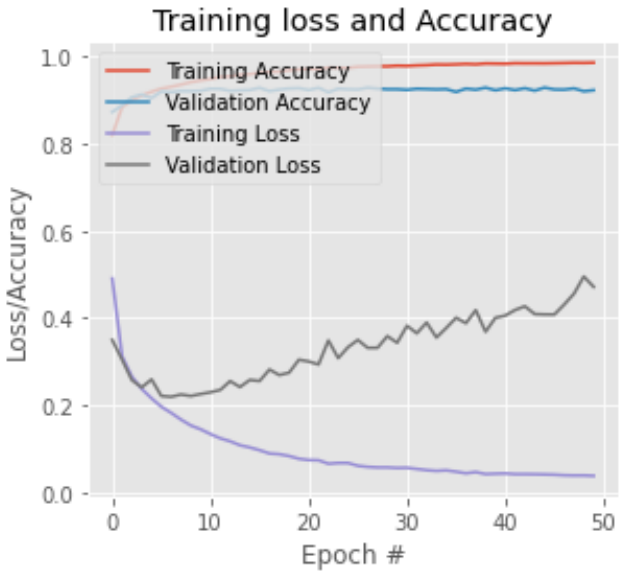

Fig 5. LSTM with single bi-directional layer curve recurrent dropout=0

As shown in the training loss and accuracy curve in Figure 5, training accuracy is greater than validation accuracy throughout the curve. However, the gap between the training accuracy and validation accuracy is higher as compared to the training phase. The LSTM used is with a single bi-directional layer curve recurrent dropout=0.1. Next, when compared with the Random Forest classifier, the CRF classifier did better as the scores improved. However, the precision and recall metrics of the classes individually have improved but not as expected.

Finally we have done comparison of condtional random forst algorithm (CRF) and BiLSTM model using Sheko language dataset , wh have selected recall, precision, f1 measure parmeters for tge compariono of the model, we can see clearelly about **Table 4**, the result from two model and performance of each model on individual class in sheko name entity recognition.

Comparison of LSTM and CRF in sheko ENR (in terms of recall, precision, f1 measure)

Classifier Class Precision R ecall F1 measure

BiLSTM Person 0.84 0.90 0. 87

Organization 0. 81 0.78 0.79

Geography 0.81 0.80 0.80

Natural 0.60 0.30 0.40

Geopolitical 0.87 0.50 0.64

Time 0.84 0.75 0.79

Other 0.99 0.99 0.99

Weighted Averages 0.97 0.97 0.97

Person 0.67 0.66 0.66

Organization 0.44 0.55 0.49

Geography 0.66 0.39 0.49

CRF Natural 0.00 0.00 0.00

Geopolitical 0.00 0.00 0.00

Time 0.98 0.19 0.32

Other 0.93 0.93 0.93

Weighted Averages 0.93 0.93 0.93

As we experiment with two different approaches depicted in **table 4**, LSTM, and conditional random field. The results obtained from the experiments conducted show the lower F-measures could be due to three possible reasons, the first one is the train test split used in these experiments discards 30% of training data for the test. This affects the accuracy of the model. The other reason is that deep neural networks need a very large amount of data to give better performance and the data set we used is large enough. But in the CRF model if the dataset is very large well performed in the training and slowdown in the testing phase and not generalized well because difficult to memorize that amount of data. Other issues are the network parameters optimization technique used in our experiments is manually searching optimized value to train and validate the mode so this case leads to minimizing the performance of the model. The accuracy obtained in the experiment is 97% in LSTM and 93% in CRF. As we have observed in the table, precision, recall, and f1-measure for the Geopolitical and natural class are Zero due to the lack of named entities in the class. There is a lack of resources in the class; we have collected these data from student books in the language, and people who teach the language in the south nation and nationality regional state, Ethiopia

In the Natural and Geography categories class precision, recall and F1 measure are 0.00 in CRF not occurred in BiLSTM. BiLSTM considers the current input plus (some) previous inputs, each previous input is down-weighted a bit compared to the one after it. When we use a Bi-LSTM, we are running the input through two LSTMs, one reading our input left to right and another right to left, and then concatenating the outputs. The first LSTM learns the effect of previous words and the second learns the effect of future words at this time the largest sequence's maximum length and pads shorter sequences leading our BiLSTM model performed better than CRF. Bi-LSTMs usually provide slightly better results than using a CRF model for most Entity name Recognition tasks, because a word's context in a sentence includes future words as well as previous words. There are several ways of encoding the output, but they typically encode at least: Beginning, inside, and Outside of an entity in BiLSTM. CRF can model dependencies over any observations in the sequence and the current label, in this case, the performance of the CRF model in the Natural and Geography categories leads to zero

When we compared our works with other related work in Anyuak Language Named Entity Recognition Using Deep Learning Approach to detect and classify words into four predefined classes: and achieved 94% test accuracy [16]. In a hybrid approach combining rule-based and machine learning techniques that were used to create NER for the Afan Oromo language and they achieved 76.6% test accuracy this fail to local minima. Another work was Arabic text recognition using CNN and LSTM and finally, they achieved 83.15% test accuracy. However, from our study, we have achieved 97% test accuracy. The reason we achieved our model better performed from previous work is preprocessing steps like normalization, tokenization, and cleaning the datasets before feeding them into the LSTM model this leads our model easily learn features. Adjusting parameters by considering the number of datasets in each class improved the performance of the BiLSTM model. And finally shown applied bidirectional LSTM which proved to be far better than single LSTM in Sheko NER. We studied the impact of each module (size of embedding layer, and LSTM unit) and the effectiveness of their combinations.

4 Conclusion

In this study, we tried to develop named entity recognition for the Sheko language, which is the first of its kind in the language. It is an under-resourced language among the other African languages. In the current study, we have designated the Named Entity Recognition Model for the Sheko language. To attain our objective, data had been gathered from different sources and our corpus was prepared of 63813 words. The corpus has nearly populated NE instances for seven classes. The preprocessed corpus needs to represent in machine-understandable form, and we used LSTM recurrent neural network to train and build an identification model and got 97% test accuracy for Sheko language entity name recognition. The model might be incorporated in advanced language-processing applications like Machine translation, information retrieval, and opinion mining for the language. Not just this, but the model also had great significance for mass Media those using the language by extracting NE from unstructured text inputs. It would be well reasonable in the future to conduct research works in the language using a large corpus to develop language-specific features to enhance the performance. Additionally, it might be possible to develop and utilize word embedding for the language.

References

1) Patil N, Patil A, Pawar BV. Named Entity Recognition using Conditional Random Fields. *Procedia Computer Science*. 2020;167:1181–1188. Available from: http://dx.doi.org/10.1016/j.procs.2020.03.431.
2) Muralikrishna H, Sapra P, Jain A, Dinesh DA. Spoken Language Identification Using Bidirectional LSTM Based LID Sequential Senones. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. 2019;p. 320–326. Available from: https://doi.org/10.1109/ASRU46091.2019.9003947.
3) Elfaik H, Nfaoui EH. Deep Bidirectional LSTM Network Learning-Based Sentiment Analysis for Arabic Text. *Journal of Intelligent Systems*. 2020;30(1):395–412. Available from: http://dx.doi.org/10.1515/jisys-2020-0021.

4) Abafogi A. Boosting Afaan Oromo Named Entity Recognition with Multiple Methods. *Int J Inf Eng Electron Bus*. 2021;13(5):51–59. Available from: http://dx.doi.org/10.5815/ijieeb.2021.05.
5) Tran Q, Mackinlay A, Yepes AJ. Named Entity Recognition with stack residual LSTM and trainable bias decoding. 2019;p. 566–575. Available from: http://arxiv.org/abs/1706.07598.
6) "sheko-hellenthal-0328 | Endangered Languages Archive.". . Available from: https://www.elararchive.org/uncategorized/SO_9306e3ab-c010-4f03-bacc-9a139ec2ce43/.
7) Bose P, Srinivasan S, Sleeman WC, Palta J, Kapoor R, Ghosh P. A Survey on Recent Named Entity Recognition and Relationship Extraction Techniques on Clinical Texts. *Applied Sciences*. 2021;11(18):8319–8319. Available from: https://doi.org/10.3390/app11188319.
8) Zhang N, Xu G, Zhang Z, Li F. MIFM: Multi-Granularity Information Fusion Model for Chinese Named Entity Recognition. *IEEE Access*. 2019;7:181648–181655. Available from: https://doi.org/10.1109/ACCESS.2019.2958959.
9) Wu G, Tang G, Wang Z, Zhang Z, Wang Z. An Attention-Based BiLSTM-CRF Model for Chinese Clinic Named Entity Recognition. *IEEE Access*. 2019;7:113942–113949. Available from: https://doi.org/10.1109/ACCESS.2019.2935223.
10) Cho M, Ha J, Park C, Park S. Combinatorial feature embedding based on CNN and LSTM for biomedical named entity recognition. *Journal of Biomedical Informatics*. 2020;103:103381–103381. Available from: https://doi.org/10.1016/j.jbi.2020.103381.
11) Chen T, Xu R, He Y, Wang X. Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Syst Appl*. 2019;72:221–230. Available from: https://doi.org/10.1016/j.eswa.2019.10.065.
12) Habibi M, Weber L, Neves M, Wiegandt DL, Leser U. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*. 2017;33(14):i37–i48. Available from: https://doi.org/10.1093/bioinformatics/btx228.
13) Wei Q, Chen T, Xu R, He Y, Gui L. Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks. *Database*. 2016;2016:baw140–baw140. doi:10.1093/database/baw140.
14) Ye N, Qin X, Dong L, Zhang X, Sun K. Chinese Named Entity Recognition Based on Character-Word Vector Fusion. *Wireless Communications and Mobile Computing*. 2020;2020:1–7. doi:10.1155/2020/8866540.
15) Agrawal A, Tripathi S, Vardhan M, Sihag V, Choudhary G, Dragoni N. BERT-Based Transfer-Learning Approach for Nested Named-Entity Recognition Using Joint Labeling. *Applied Sciences*. 2022;12(3):976–976. Available from: https://doi.org/10.3390/app12030976.
16) Journal I, Science OF. Anyuak Language Named Entity Recognition Using Deep Learning Approach. 2021. Available from: https://doi.org/10.17485/IJST/v14i39.1163.
17) Cao S, Lu W, Zhou J, Li X. CW2Vec: Learning Chinese word embeddings with stroke n-gram information. *32nd AAAI Conf Artif Intell*. 2019;p. 5053–5061. Available from: https://doi.org/10.1016/j.eswa.2019.10.003.