DEEP LEARNING-BASED PART-OF-SPEECH TAGGING OF THE ETHIOPIC-LANGUAGE

BY: SENAIT GEBREMICHAEL TESFAGERGISH
SUPERVISOR: JURGITA KAPUCIUTE DZIKIENE

RESEARCH OBJECTIVE

- Part-of-Speech tagger is a prerequisite for many NLP applications

- Tigrinya is
  - morphologically rich,
  - low- resource language

- POS tagging for Tigrinya:
  - Limited resources (Keleta et al..,2016)
  - Solved with traditional ML approaches (Teklay Gebregzabiher Abreha, 2010; Keleta et al., 2016)

INTRODUCTION

GOALS

- Solve POS tagging task with the state-of-the-art (i.e., deep learning) approaches

TASKS

- Related Research Analysis
- Data Preparation
- Experiment with different approaches (classifiers and vectorization type)
- Manual and automatic tuning of hyper-parameters
- Recommendation formulation

Scientific novelty:

- POS tagging task for Tigrinya has never been solved before using Deep Neural Networks and neural word embeddings

Practical value:

- Part-of-Speech tagger could be used for many NLP applications

TIGRINYA AND ITS MORPHOLOGY

- Belongs to the Semitic language- Afro-Asiatic family:

  Hebrew, Amharic, Maltese, Tigre and Arabic

- Characterized by rich derivational and influential morphology.

- Distinguishing feature lies in the ‘root-template’ morphological pattern that is often composed of trilateral roots.

- Ge’ez script is adapted to write other mostly Semitic languages, Particularly Amharic and Tigrinya.

Token: እንተዘይሓተትካዮ ‘IntezeyHatetIkayo’

Gloss: if you did not ask him

Word order: if not asked you him

Morphology:

Inte | zI | ayI | HatetI | ka | Yo

Category: VERB

Sub Category: REL NEG TAM: perf.

Clitcs:

CON

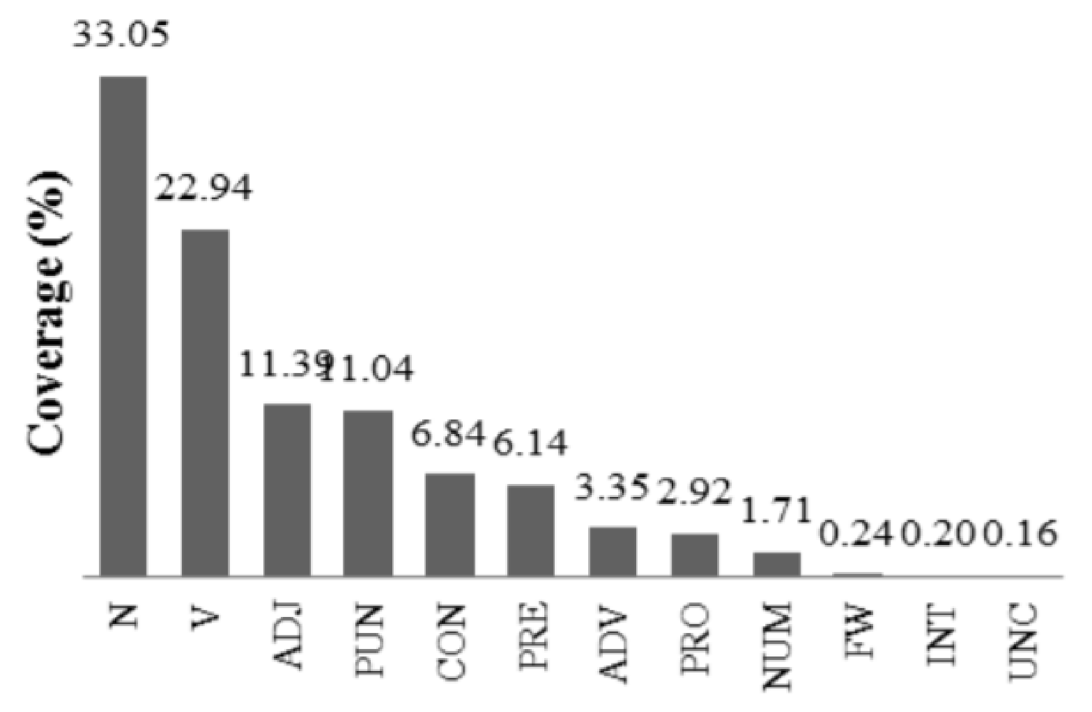
Attributes: SUBJ.PRO.1st, Sg, M OBJ. PRO. 3rd,Sg,M

Stem: HatetI (CaCeCI) Root: Htt

POS: V_PRF_C(Perfective Verb with Conjugation)

THE CORPORA

- Nagaoka Tigrinya Corpus (Keleta et al., 2016) is the only Corpus available publicly.
- Consists of:
  - 72,080 tokens
  - 4656 sentences.
  - 20 types of POS tags: V_PRF, UNC, V_AUX, V_IMV, N, PUN, V_REL, ADV, INT, N_V, ADJ, NUM, N_PRP, FW, V, V_GER, CON, V_IMF, PRE, PRO



- Nagaoka Tigrinya Corpus Snippet and its class distribution

VECTORIZATION

Word Embedding

- DNN applied on a top of the vectorized words.
- Words represented with real values vector
- Similar words are projected closer in vector space.
- Word2Vec approach is used in this experiment
- Window size equal to 3 and with 100 dimensions.
- Pre-trained word embeddings saved and afterwards used in all the experiments.

EXPERIMENTAL SETUP

Evaluation metrics

- Accuracy = $\frac{tp+tn}{tp+tn+fp+fn}$

where, tp (true positive)

tn (true negatives)

fn (false negatives)

fp (false negatives)

- Loss = $-\sum_{c=1}^{M} y_{o,c} \log \log(p_{o,c})$

Where, M is number of classes (POS tags)

Y is binary indicator (0,1) if class label c is the correct classification for observation o

P is a predicted probability of observation o in class c

EXPERIMENTAL SETUP

Baselines

- Random baseline = $\sum P(c_i)^2 = 0.127821$
- Majority baseline = $\max P(c_i) = 0.270475$

Statistical significance measure:

- McNemar(1945) test with significance level =95%

EXPERIMENTAL SETUP

Tools

- Python programming language, TensorFlow, Keras, Hyperas
- Tested methods:
  - FFNN-Feed Forward Neural Network
  - CNN- Convolutional Neural Network
  - LSTM- Long Short Term Memory
  - BiLSTM – Bidirectional LSTM
- Tested hyper-parameters:
  - Activation function
  - Epochs
  - Batch sizes

EXPERIMENTAL SETUP

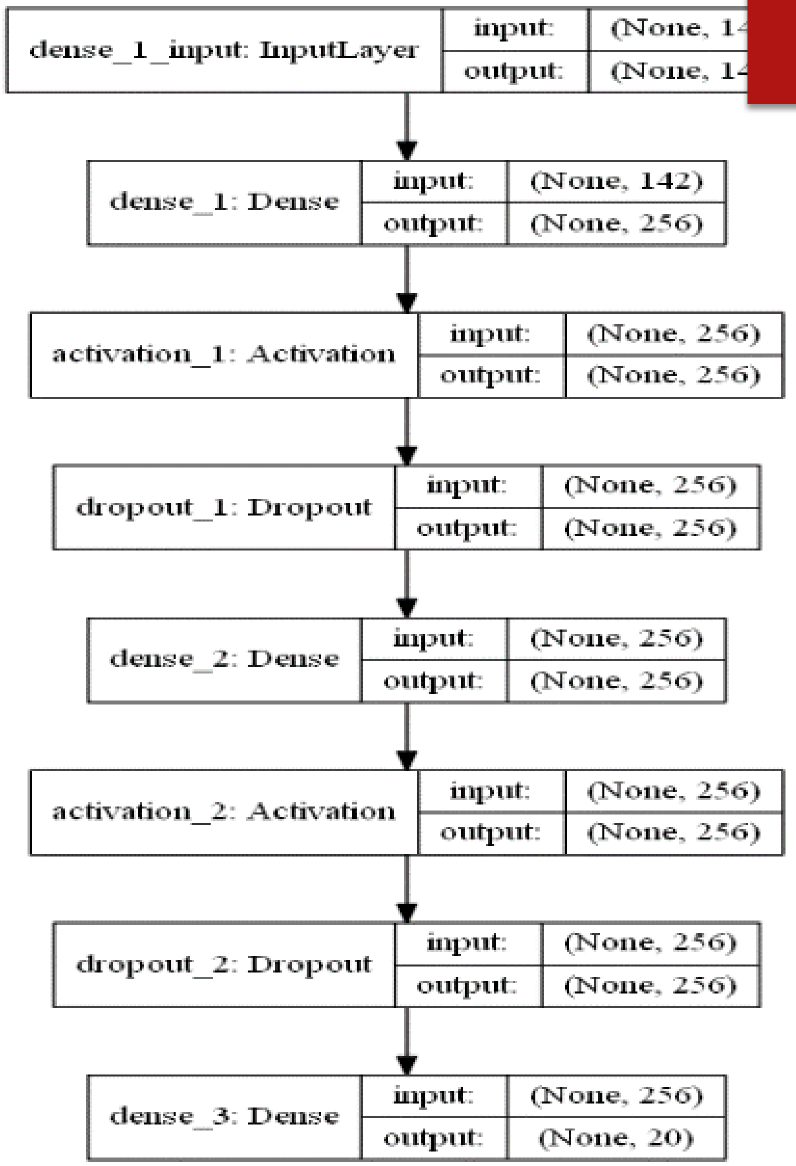
- Hyper- parameter tuning approaches:
  - Manual hyper-parameter tuning
  - Automatic hyper-parameter tuning

- Manual tuning: 80% of dataset for training ; 20% for testing

- Automatic tuning: 80% for training (of which 20% for validation); 20% for testing

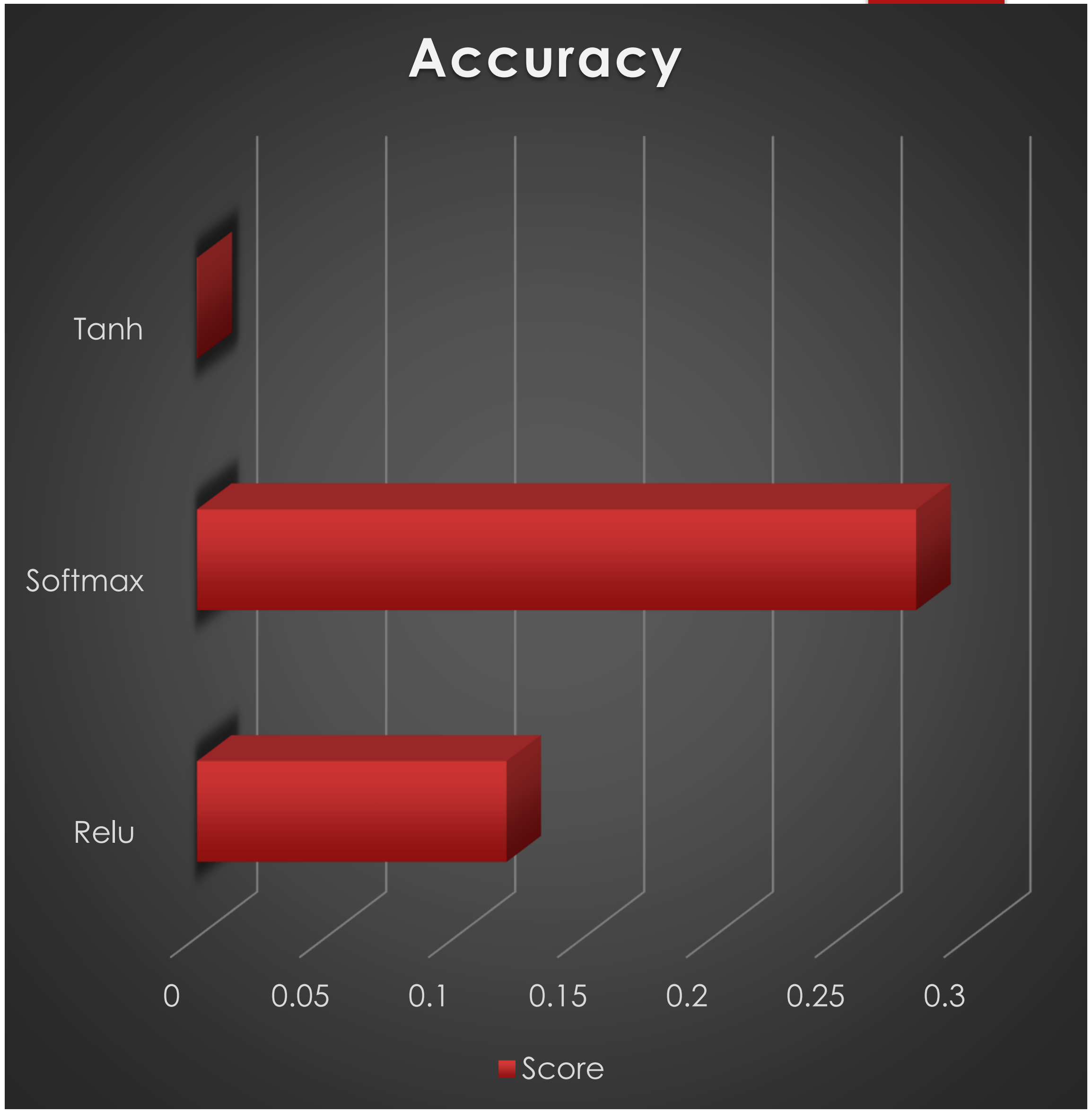
FFNN

- Tuning type: manually
- Vectorization : One-hot encoding
- Hyper-parameters:
  - Up to 3 hidden layers
  - of neurons of 256, 512, 1024
  - Epochs =100
  - Batch_size = 256



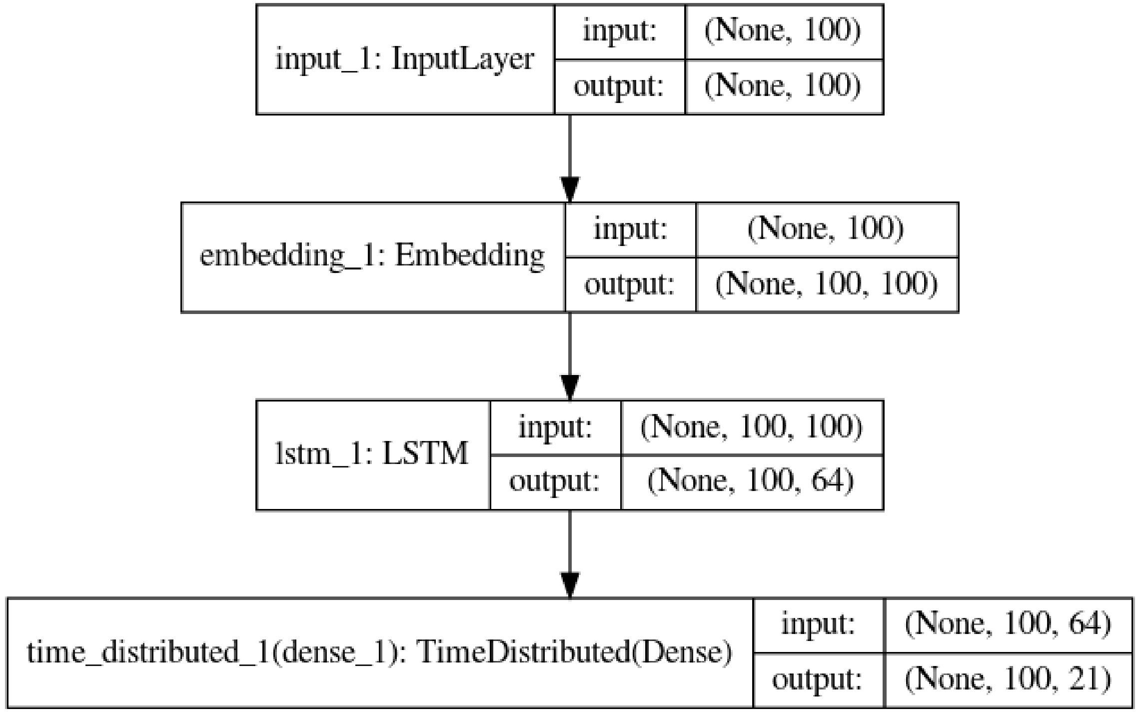
FFNN RESULTS

- 28% of accuracy
- Softmax Activation function
- Different Hidden layers and neurons didn’t show any significant impact on the accuracy



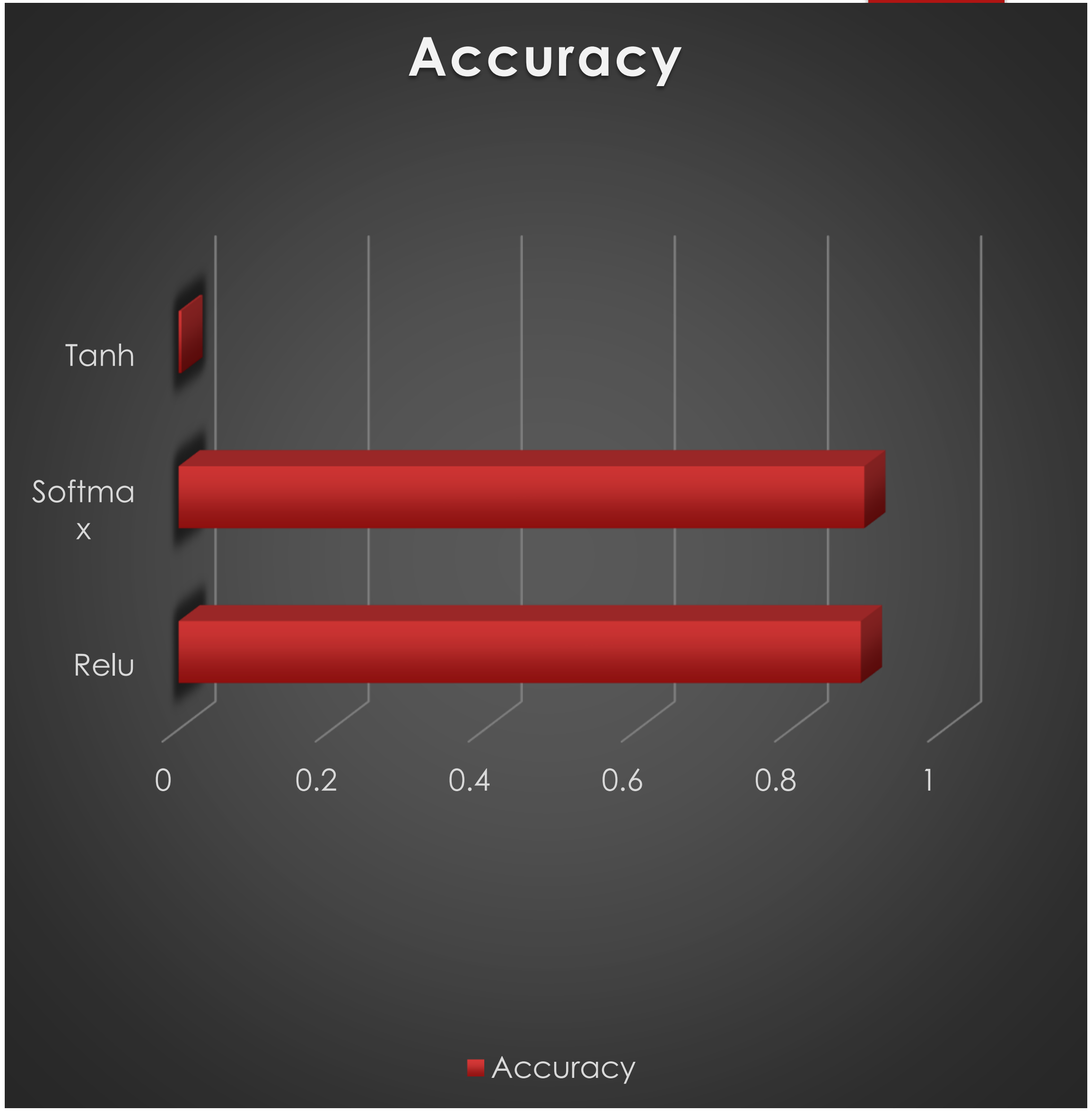
LSTM

- Training type : manually
- Vectorization : neural word embeddings
- Hyper-parameters:
  - Epochs = 100
  - Batch_size = 32



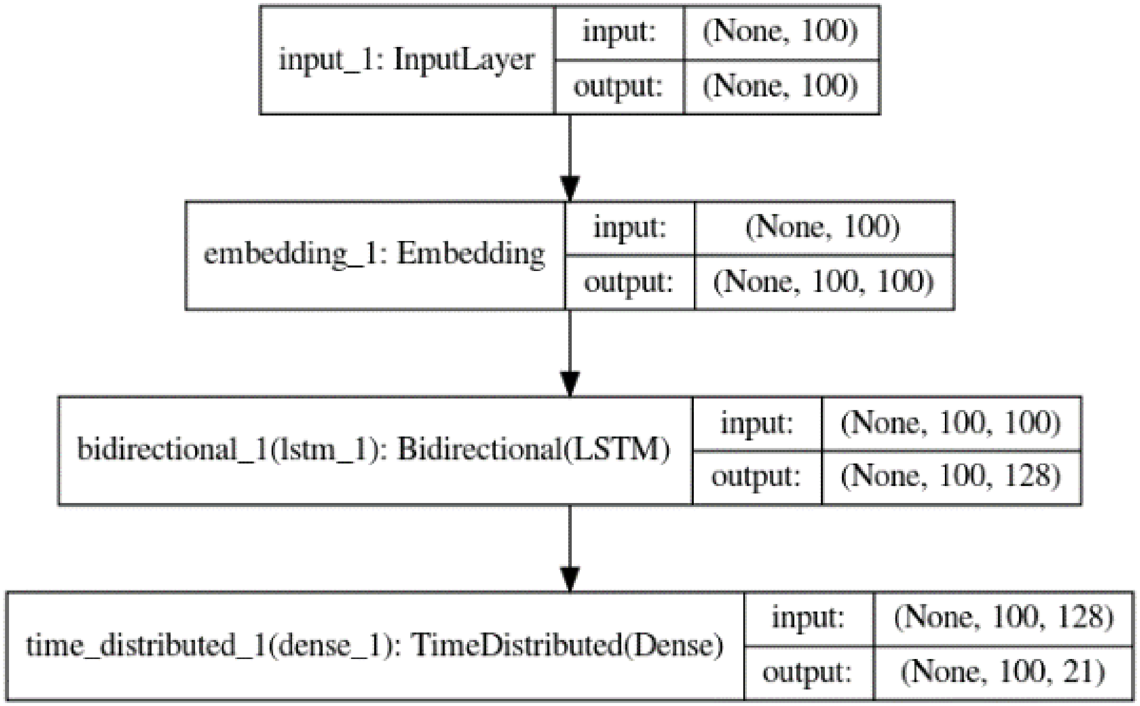
LSTM RESULTS

- 89% of accuracy
- Softmax Activation Function
- Optimizer = rmsprop
- 1 layer
- 64 neurons



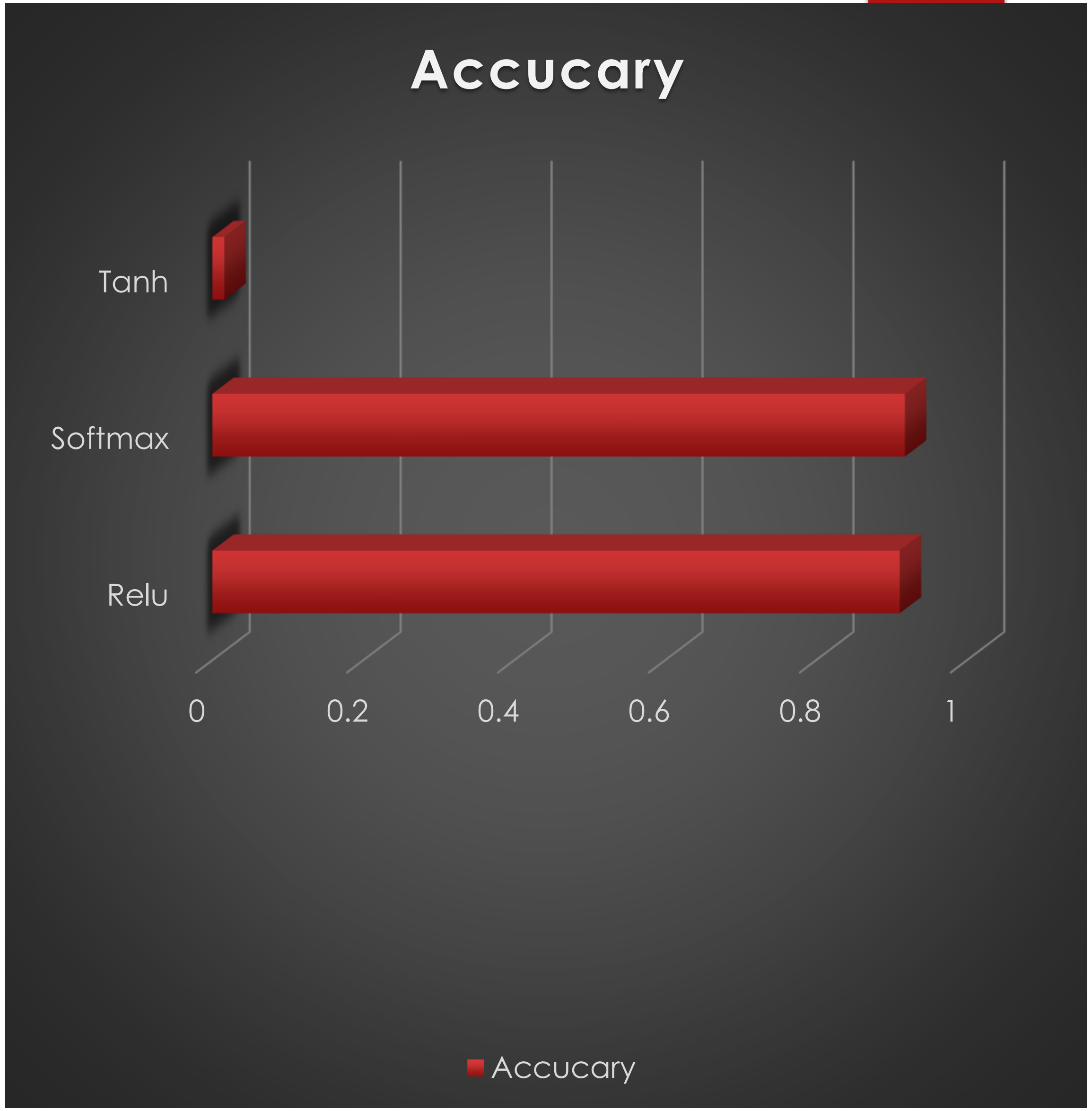
BiLSTM

- Tuning type : manually
- Vectorization : neural word embeddings
- Hyper-parameters:
  - Epochs = 100
  - Batch_size 32



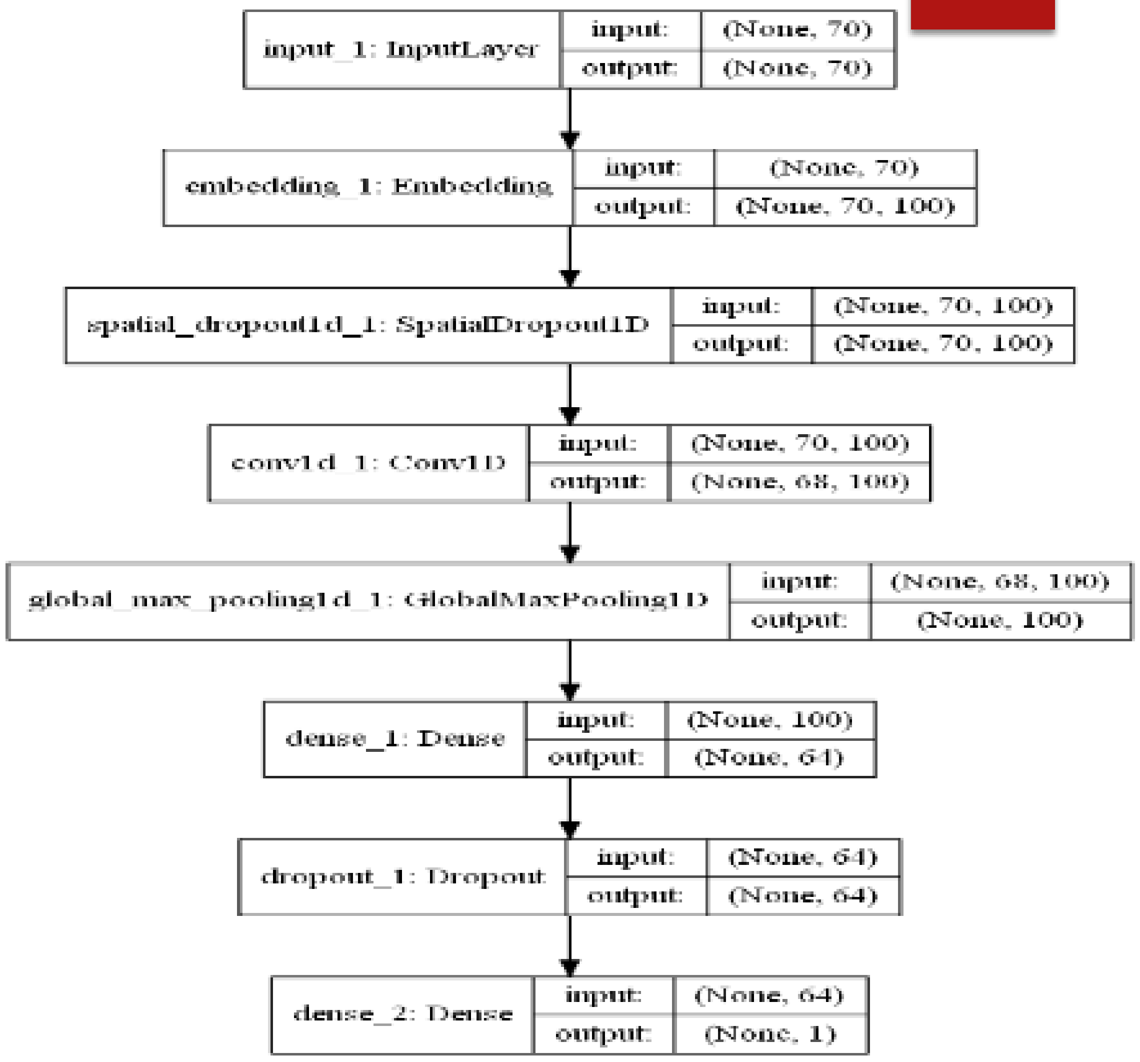
BiLSTM RESULTS

- Best accuracy achieved using the Softmax Activation function that is 91%
- 1 layer
- 128 neurons



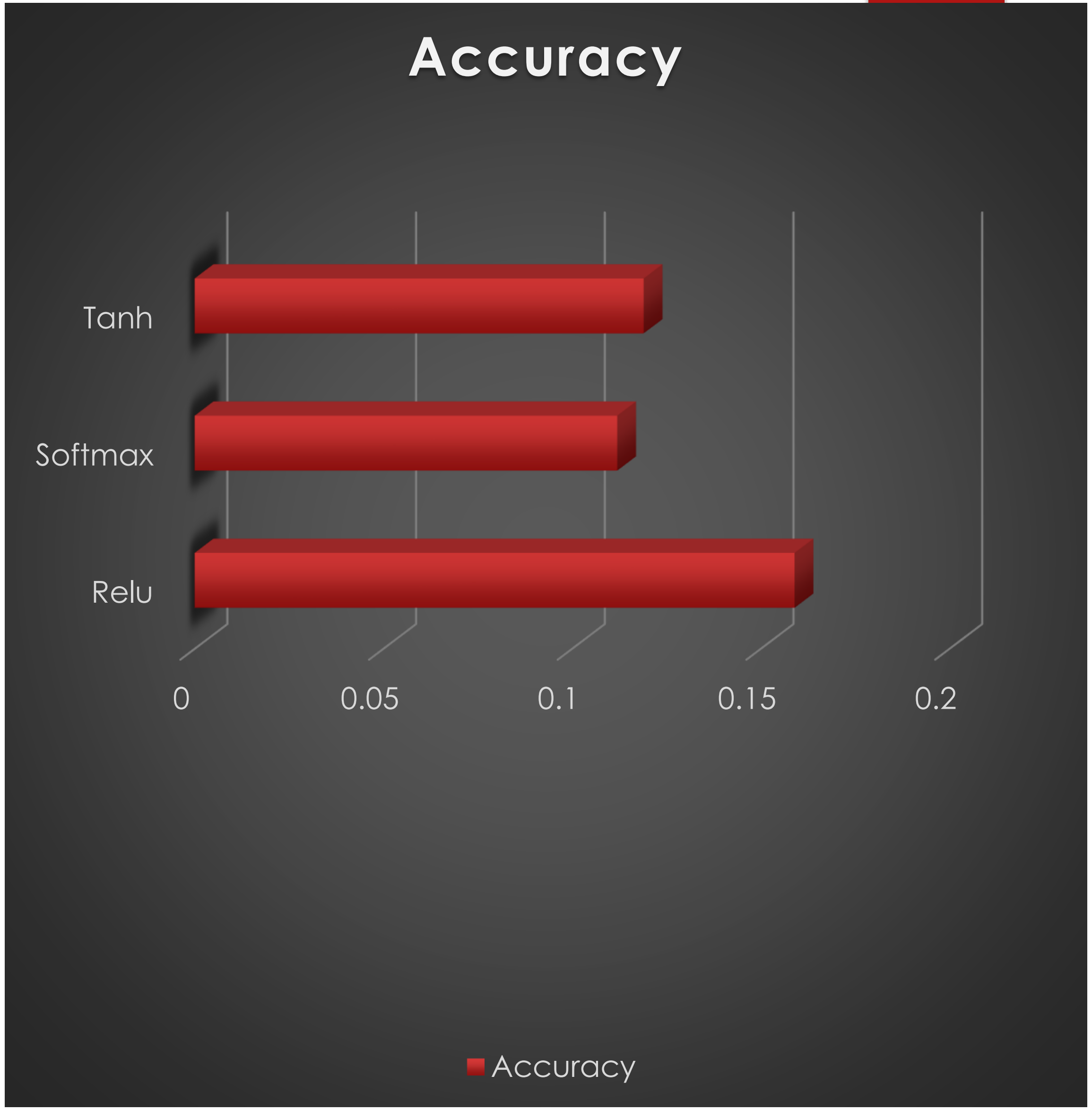
CNN

- Tuning type : manually
- Vectorization : neural word embeddings
- Hyper-parameters:
  - Dimension: 1D
  - Kernel Size = 3
  - Filters = 100
  - Neurons 64 and 1



CNN RESULTS

- 15 % of accuracy
- ReLU Activation function



Manually tuned Classifiers hyper-parameter and accuracies

CLASSIFIERS	LAYERS	OPTIMIZERS	NEURONS	ACTIVATION FUNCTION	ACCURACY
FFNN	1,2,3	Adam	256	Softmax	28%
LSTM	1	rmsprop	64	Softmax	89%
BiLSTM	1	rmsprop	128	Softmax	91%
CNN	2	Adam	64 + 1	ReLU	15%

Automatic DNN Hyperparameter Optimization

Automatic Hyper-parameter tuning

- Tuned methods:
  - LSTM
  - BiLSTM
  - CNN

- Tested parameters and their values:
  - Activations: Sigmoid, Softmax, tanh, ReLU , Swish, SeLU
  - Optimizers: adam, sdg, rmsprop
  - Batch size: 32, 64, 128
  - Neurons : 16, 32, 64, 128
  - Layers: Up to 3 layers

- Tuning strategy:
  - *tpe.suggest* strategy
  - Optimization in 20 iterations

Method	Activation	Layers	Neurons	Batch_size	Optimizer	Accuracy
LSTM	Sigmoid	1	32	32	rmsprop	0.89
BiLSTM	Sigmoid	1	64	32	rmsprop	0.91
CNN	Sigmoid, Softmax	1	32	32	adam	0.61

Hyperparameter optimization result

DISCUSSION

- Previous work for Tigrinya POS tagging:
  - use the traditional methods of CRFs and SVMs.
  - 90% accuracy achieved by enriching contextual features with morphological and affix features.
  - DNN BiLSTM outperformed previous work with traditional machine learning approaches (Keleta et al., 2016) for Tigrinya language

- Results with DNN methods are above random and majority baselines.

- Tuning hyper-parameters automatically, achieved the best result using the BiLSTM, yet didn’t beat the achievement of manually tuned hyper-parameter in BiLSTM model

DISCUSSION

- CNN model shows a good improvement using the automatically tuned hyper-parameters, which increases to 61% from 15%.
- FFNN performs poorly and is not suitable for the solving task.
- Working with small amount of data requires rule-based method and with NTC 1.0 more feature extraction is needed for better accuracy with other methods
- The McNemar test show that the difference between the best and the second achieved results are statistically significant

Conclusion and Future Work

- The best achieved accuracy = 91% for the Tigrinya POS tagging was achieved with BiLSTM method and neural word embeddings.
- The contribution of this research:
  - The first time state-of-the-art methods were applied for the Tigrinya language
  - Neural word embeddings have been trained and used for the first time for the Tigrinya language
  - The recommendations about the classifier and vectorization should be beneficial for the other Semitic languages (such as Tigre, Saho, Afar)
- Future work:
  - More data; more diverse data
  - Further NLP tasks that could not be initiated without POS tagger (e.g., dependency parsing)

Thank you