Neural Network and Transformer-Based PoS Tagger for Low Resource Languages

***Endrit Fetahi*[1], *Mentor Hamiti*[1], *Arsim Susuri*[2], *Besnik Selimi*[1], *Deshira Imeri Saiti*[1]**

1 Faculty of Contemporary Sciences and Technologies South East European University Tetovo, North Macedonia
2 Faculty of Computer Science, University of Prizren Ukshin Hoti, Prizren Kosovo

presented at InfoTech 2024, Sofia, BULGARIA (Virtual Forum)
http://infotech -bg.com/

Outline

- I. Introduction
- II. Materials and methods
- III. Results and Discussion
- IV. Conclusion

Introduction

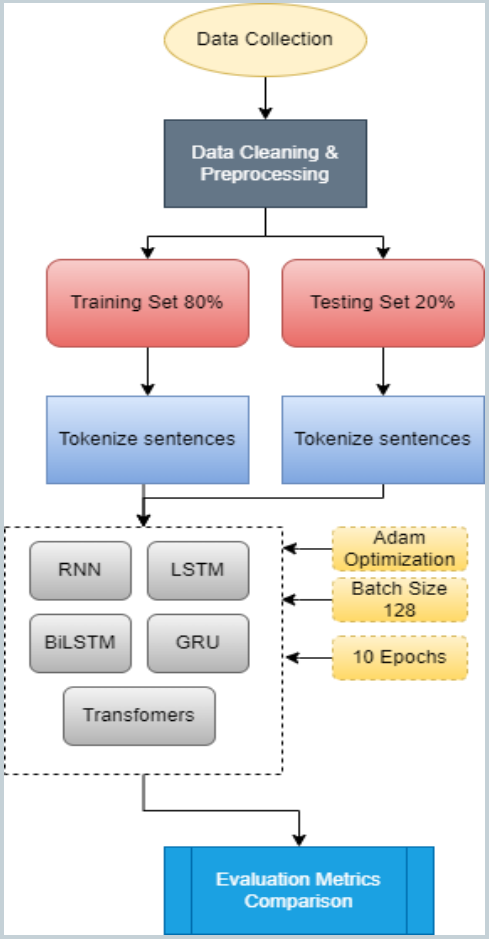
- NLP (Natural Language Processing) is a wide area of research nowadays since many different technologies are involved in order for machines to understand and process information. This field presents particular challenges regarding low-resource languages, as there is a low amount of research done. This shows a significant gap in relation to Computational Linguistics in these languages.
- Part-of-Speech (PoS) tagging is a key focus in low-resource languages as it serves as a foundation for further investigation
- PoS Tagging is particularly helpful for many NLP applications, including Text-to-Speech systems, Corpus Linguistics, Information Extraction and Retrieval (e.g., Document Categorization in Internet Search Engines), as a pre-processing step to Syntactic Parsing, etc.
- The aim of this paper is to implement and observe up-to-date deep learning techniques to realize PoS Tagging for low-resource languages, with the case study of the Albanian language.

  The main objectives of present study are as follows:

  - **Analyze and Implement Deep Learning Techniques for PoS Tagging**
  - **Optimize the PoS Tagging process by using Embedded Weights**
  - **Evaluate the models and propose the best approach.**

Materials and Methods

- The low-resource language selected for the experiments in this research paper is Albanian.
- For the modeling, evaluation, and optimization of the Tagging Pipeline that we propose in our experiments, we make use a manually annotated corpora
- This corpus is a combination of several resources, such as The Leipzig Corpora Collection, which is based on web-crawled text and Wikipedia and contains sentences sampled from the Albanian section.



Corpus

- The dataset used contains 117,686 tokens (6,644 sentences) of written Albanian that were gathered from various text sources. This corpus has gone through two phases before being implemented and validated in our experiments, such as data cleaning and data preparation for the implementation of the experiments.
- We eliminated all sentences which contained characters not recognized in the Albanian language, as well as sentences that had unfilled PoS Tags.
- Sentences that had various anomalies such as words without their corresponding tag, or lengthy sentences (as can be seen in the following figure) were excluded from the database. In total, there are 5,208 rows and 93,068 tokens after this phase.

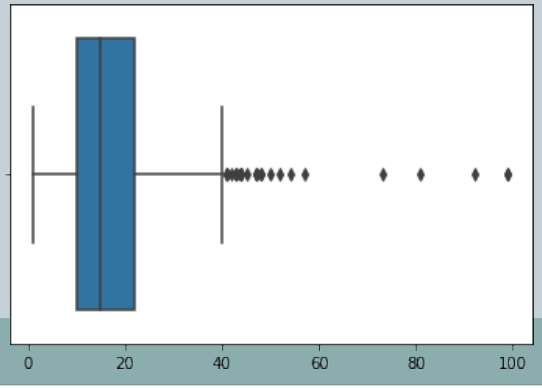


Figure.1 Sentences Length of the Dataset

Data Preparation

- In order to facilitate the implementation of the experiments, various Python libraries have been utilized, each of which has its own form of corpus reading.
- Initially, the corpus was converted from the CoNLL-U format, which is the universal format for datasets, to a text file, wherein each sentence is represented by a new line.
- This file is then read by the NLTK library, which houses the TaggedCorpusReader.
- Each word, along with its corresponding tag, is read by the library. The tag set used in the research paper is presented in Table 1.

TABLE I. TAGSET USED FOR THE POS TAGGER [15]

Part of Speech	*POS Tagg*
Noun	NOUN
Proper Noun	PROPN
Adjective	ADJ
Numerals	NUM
Pronoun	PRON
Verb	VERB
Adverb	ADV
Preposition	ADP
Conjuction	CCONJ
	SCONJ
Part	DET
Interjection	INTJ
Symbol	SYM
Punctuation	PUNCT

Experimental Setup and evaluation

- Our experiments were conducted using deep learning models such as RNN, LSTM, Bi-LSTM, GRU, and Transformers. We implemented the experiments using the Python Programming Language, as well as the NLTK, Keras, SKLearn, and Huggingface Transformers libraries. Additionally, within the scope of our tag experiments, we will undergo a process of converting them into a vector space. This conversion aims to optimize our models and facilitate the attainment of higher levels of accuracy.
- To evaluate the model, we divide the dataset into a training set (80%) and a test set (20%).
- We conducted the experiments in three different setups such as:

Experimental Setup and evaluation

- Setup A – the above Neural Networks will be used, with each tag encoded as an integer. The Neural Networks shall be configured to have 64 cells, with Softmax activation and Adam optimization. The model shall run for 10 Epochs, with a batch size of 128.
- Setup B – Neural Networks with the same configuration as in Setup A will be utilized, however, for optimal results, Embedded Weights will be employed in all Neural Networks.
- Setup C – Fine-tuned Hugging Face Transformers with the BERT model will be used. Therefore, in this setup, 10 epochs with the Adam optimizer will be executed.
- The evaluation metrics we use to measure the performance of the models are:
  - **Recall**
  - **Precision**
  - **F1**

Results

- The results of Setup A experiments are described in Table 2, wherein several types of neural networks were used for the implementation of the experiments.
- The results were measured using the aforementioned metrics. From these results, it is evident that almost all neural networks perform similarly to one another except of RNN having lower performance. However, it is worth noting that BiLSTM performs best with an F1 of 87%, with a minimal difference from the others.

TABLE II. SUMMARY OF THE RESULTS FROM EXPERIMENTS ON SETUP A

Model	*Recall*	*Precision*	*F1*
RNN	0.74	0.73	**0.72**
LSTM	0.85	0.86	**0.85**
BiLSTM	0.87	0.87	**0.87**
GRU	0.86	0.86	**0.86**

Results

- In Setup B, Embedded Weights pre-calculated in vector form are utilized, with the intention of increasing the accuracy of neural network models. In Table 3, we can observe the results obtained from the experiments.
- Even in Setup B, neural networks perform similarly to one another.

TABLE III. SUMMARY OF THE RESULTS FROM EXPERIMENTS ON SETUP B

Model	*Recall*	*Precision*	*F1*
RNN	0.85	0.85	**0.85**
LSTM	0.86	0.86	**0.86**
BiLSTM	0.86	0.86	**0.86**
GRU	0.86	0.86	**0.86**

- However, while employing embedded weight in such experiments, it is demonstrated that they do not effectively increase in performance, rather staying in the same range of accuracy. Notable the only model having an affect was the RNN model, increasing in accuracy with 13%, which shows that embedded weighs are effective only for the RNN model.

Results

- In Table 4, the results from the implementation of Fine-Tuned Transformers are presented, indicating that this model is the most superior among all Deep Learning methodologies.
- These models are a significant improvement over RNN-based models, as they process input sequences as a whole rather than token by token, allowing the model to be accelerated using GPUs
- The accuracy of the PoS Tagger is already the highest, at 95% F1. This shows that this model is demonstrating to be the best model for performance in order for PoS tagging.

TABLE IV. SUMMARY OF THE RESULTS FROM EXPERIMENTS ON SETUP C

Model	*Recall*	*Precision*	*F1*
Fined Tuned Transformers	0.95	0.95	**0.95**

Conclusions

- In this research, we have developed Part of Speech Taggers based on deep learning models, for low-resource languages, taking the Albanian Language Dataset for the experiments.
- Known Neural Networks architectures as RNN, LSTM, BiLSTM, GRN, and Transformers were used for developing the PoS Tagger. For the optimization of the PoS Tagger, the PoS tags were converted into vectorial form with the word2vec library, which enhanced the accuracy of the PoS tagger overall.
- The dataset was divided into an 80% training set and a 20% testing set. These models were trained with 10 Epochs and reached high accuracy. The models were evaluated using different metrics such as precision, F1, and recall.
- It was observed that for all the different combinations made in the experiments, the Fined Tuned BERT Transformers performed significantly better than the other models with high F1 of 95%.
- Not bypassing the BiLSTM model can also be considered for future work. Taken together, we can conclude that, for low-resource languages, deep learning models are proven to be adequate with high accuracy, given that iterative methods require a lot of work which its realization is unrealistic for these types of languages given the fact that they are low resourced.

THANK YOU

Q&A

Endrit Fetahi
ef30456@seeu.edu.mk