

Amazigh part-of-speech tagging with machine learning and deep learning

Otman Maarouf¹, Rachid El Ayachi², Mohamed Biniz³

^{1,2}Department of Computer Science, Faculty of Science and Technology, Sultan Moulay Slimane University, Beni Mellal, Morocco

³Department of Computer Science Polydisciplinary, Faculty Sultan Moulay, Slimane University, Beni Mellal, Morocco

Article Info

Article history:

Received Jun 11, 2021

Revised Oct 14, 2021

Accepted Oct 27, 2021

Keywords:

Amazigh language

Conditional random fields

Decision tree

Deep learning

Machine learning

Part of speech

Tifinagh

ABSTRACT

Natural language processing (NLP) is a part of artificial intelligence that dissects, comprehends, and changes common dialects with computers in composed and spoken settings. At that point in scripts. Grammatical features part-of-speech (POS) allow marking the word as per its statement. We find in the literature that POS is used in a few dialects, in particular: French and English. This paper investigates the attention-based long short-term memory (LSTM) networks and simple recurrent neural network (RNN) in Tifinagh POS tagging when it is compared to conditional random fields (CRF) and decision tree. The attractiveness of LSTM networks is their strength in modeling long-distance dependencies. The experiment results show that LSTM networks perform better than RNN, CRF and decision tree that has a near performance.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Otman Maarouf

Department of computer science

Sultan Moulay Slimane University, Beni Mellal, Morocco

Email: maarouf.otman94@gmail.com

1. INTRODUCTION

Part-of-speech (POS) labeling is the way toward doling out grammatical feature markers to each word in an informational text in different languages and dialects. [1] The contribution to a labeling calculation is a succession of (tokenized) words and a label set, and the yield is an arrangement of labels, one for each token [2]. Thusly, POS taggers are an import bug module for gigantic open applications, for instance, the questions-taking notes of systems, information extraction, information recuperation, machine understanding. They can be used in various applications, for instance, substance to talk or like a preprocessor for a parser; the parser can improve, in any case, dynamically expensive. In this paper, we decided to focus on the POS naming for the Amazigh language. These days, POS Tagging is finished with respect to computational semantics using a couple of counts by a ton of clear marks. POS-labeling figuring fall into three specific social affairs: rule-based, quantifiable, and hybrid-based taggers. A standard-based tagger uses etymological rules to consign the correct marks to the words in the sentence or record. Authentic Part of Speech tagger relies upon the probabilities of occasions of words for a given explicit tag through a decision tree and conditional random fields (CRF) approach. Cream-based Part of Speech tagger is a blend of a rule-based system and factual philosophy. Syntactic structure naming is a huge gadget of normal language arrangement. It is used in a couple of Natural Language's readiness-based programming executions. For all natural language processing (NLP) tasks like language structure, checker machine, and translation, the precision depends on the accuracy of the part of speech tagger.

Tagger accepts a critical occupation in talk affirmation, normal language parsing and information recuperation. This undertaking seeks after the measurable based technique for POS Tagging, much more expressly the CRF Model of the real system [3]. As of late, neural organizations have been acquiring notoriety in the field of computerized reasoning. The headway is because of the discovery in the calculations that learn and perceive extremely complex examples utilizing profound layers of neural organizations or normally known as the profound neural organizations deep neural network (DNN) [4], furthermore, the presentation of various kinds of neural organization, namely, convolutional neural organization and repetitive neural organization (RNN). For example, convolutional neural organizations, which are a unique kind of feed-forward neural organizations with two measurement organizations, have demonstrated colossal exactness in groups pictures through nearby open fields, shared loads, pooling, from straightforward transcribed digit acknowledgment to more perplexing face acknowledgment. In the demonstrating of consecutive examples, like, phoneme acknowledgment programmed discourse acknowledgment, discourse combination, discourse translation, chatbot, and numerous others, RNN or the more specific sort of RNN and the long-short term memory (LSTM) networks [5], [6] have demonstrated to be superior to large numbers of the conventional methodologies. This paper presents a similar investigation of three strategies to take care of the issue of Amazigh grammatical feature (POS) labeling. These techniques are LSTM organizations, CRF, and decision trees. The goal is to analyze the exhibition of the present status of LSTM organizations while contrasted with CRF and Decision trees in POS labeling. POS labeling is a language-preparing task that allocated a POS tag to each word in a sentence [4]. Concerning the organization of paper, there are three sections, in the first section we talk about the different approaches used in the Amazigh part of speech, in the second section we dispute the methodology of the article talking about the Amazigh language, the Amazigh tag set and the corpus using in this study. The third section is divided in three parts, the first part describes the machine learning algorithms (CRF and decision tree), the second part presents the architecture of the recurrent neural network and long-short term memory, but in the last part, the obtained results are discussed and the conclusion is given.

2. RELATED WORKS

In the domain of POS labeling, numerous investigations have been made. It arrived at incredible degrees of execution using discriminative models, for example, greatest entropy models (MaxEnt) [7], bolster vector machines (BVM) [8], support vector machine (SVM) is an example of supervised learning algorithms that his function is to find a hyperplane that separates data in two classes [9], or Markov CRF [10], [11]. Among stochastic models, bigram and trigram hidden Markov models (HMM) are very prominent. Dynamite is a generally utilized stochastic trigram HMM tagger which utilizes a suffix examination system to assess lexical probabilities for obscure tokens dependent on properties of the words in the preparation corpus which share the equivalent suffix [12]. The improvement of a stochastic tagger requires a lot of comment on content. Stochastic taggers with over 95% word-level exactness have been produced in English, German and other European dialects, in which huge named information is accessible. At that point, choice trees have been utilized for POS labeling and parsing. A choice tree initiated from labeled corpora was used for grammatical feature disambiguation. For Amazigh POS labeling [13]. Manufactured a POS-tagger for Amazigh, as an under-resourced language. The information used to achieve the work was physically gathered and commented on. To help increase the presentation of the tagger, they utilized AI methods (SVM and CRF) and different assets or apparatuses, for example, lexicons and word division devices to process the content and concentrate highlights sets comprising of lexical setting and character n-grams. The corpus contained 20,000 tokens and was utilized to prepare their POS-tagger model. In this manner, there is a squeezing need to build up a programmed part-of-speech tagger for Amazigh. Under the current work, a character recognition system is presented for recognizing English characters extracted from images/graphics embedded text documents such as business card images.

In the same way, we find [14], that applied the tree tagger on Amazigh texts show that provides overall tagging accuracy of 93.19%, specifically, 94.10% on known words and 70.29% on unknown words. Applying the new NLP techniques, we find [15] are used long short-term memory (LSTM) on the Amazigh part of speech in the case of texts written in Latin characters, they reach a score of 92.7% in a training of 30 epochs. We found also [16] used CRF, decision tree, and long short term memory, they reach a score of 87.6% using DT, 93.4% by CRF, and 97.87% in a training of 10 epochs.

3. METHODOLOGY

3.1. Amazigh language

Be first, Amazigh language is one of the Hamito-Semitic/"Afro-Asiatic" dialects with a rich templatic morphology. In semantic terms, the language is described by the multiplication of tongues because of

verifiable, geological, and sociolinguistic elements. In Morocco, one may recognize three significant vernaculars: Tarifit in the North, Tamazight in the middle, and Tashlhiyt in the southern regions of the country; half of the Moroccan populace speaks Amazigh yet as indicated by the last legislative demo-linguistic information by 2004, the Amazigh language is spoken only by some 28% of the Moroccan population (around 10 million citizens), demonstrating a significant diminishing of its use.

Amazigh standardisation can't be accomplished without embracing a practical procedure that consider its phonetic variety. All things considered, and as a result of recorded and social reasons, Tifinagh has become the official script for the writing Amazigh language. IRCAM kept just relevant phonemes for Tamazight, so the quantity of the sequential phonetic substances is 33, yet Unicode codes just 31 letters in addition to a modifier letter to shape the two phonetic units: ⵍ (g^w) and ⵎ (k^w). The entire scope of Tifinagh letters is partitioned into four subsets: the letters utilized by IRCAM, an all-inclusive set used by IRCAM, other new Tifinagh letters being used, and some authenticated current Touareg letters. The number reaches 55 characters [17]. Amazigh NLP presents many challenges for researchers. Its major features are:

- It has its own script: the Tifinagh which is written from left to right.
- It does not contain uppercase.
- Like other natural languages, Amazigh presents for NLP ambiguities in grammar classes, named entities, and meaning. For example, grammatically; the word (illi) depending on the context can mean a noun in this sentence (tfulki illi: my daughter is beautiful) or a verb in this sentence (ur illi walou: there is nothing).
- As most languages whose research in NLP is new, Amazigh is not endowed with the linguistic resources and NLP tools.
- Amazigh, like most of the languages which have only recently started being investigated for NLP, still suffer from the scarcity of language processing tools and resources [18].

3.2. Amazigh tag set

Characterizing the adequate label set is a middle endeavor in building a customized POS tagger. It targets describing a measurable label set to a reasonable level of warranty, for instance, not highly-elaborated nor unreasonably shallow for the potential joins structures that will use it. The used corpus contains a once-over of compositions isolated from a collection of sources, for instance a couple of books, similarly as specific works from IRCAM's site. We had the choice to show up at an all dwarf of words in a way that is better than 60k tokens. This corpus is remarked on morphologically using the mark set introduced in [19].

3.3. Corpus

A corpus is a set of language information that is chosen and coordinated by unequivocal semantic standards to fill in as an example of occupations decided on a language. Generally speaking, a corpus contains a huge number of words and can be lemmatized and clarified with data about the grammatical forms. Among the corpus, there is the British National Corpus (100 million words) and the American National Corpus (20 million words). A reasonable corpus would give a wide choice of various kinds of writing and from different sources, for example, papers, books, reference books or the web. For the Moroccan Amazigh language, it was very hard to find instant assets. We can simply make reference to the physical commented on the corpus of Outahajala *et al.* [20]. This corpus contains 20k words, which is why we have decided to use it in this study with some modifications.

- We enriched the corpus with other words, we collected more than 40k words from Moroccan agency of press (MAP) news so we have in global corpus more than 60k words
- We have added the type symbol in the types of tags signifying the special characters that they are considered punctuation in the corpus realized by Outahajala;
- We have eliminated the date, type because it is an entity;
- We use Tifinagh characters instead of Latin writing.

The advantage of using the Tifinagh character is to optimize the program by avoiding the use of transcoding. In this step, we have encountered some errors in the writing of the words in Latin; there is a character in the corpus of Outahajala written in an erroneous way like (E=T A=H M=Z H=E E=D Q=R O=S U=G Gw= X^w) (Table 1), but the correct equivalence of this character is (E=t A=h M=z H=e E=d Q=r O=s U=y X^w=gw). After the translation into Tifinagh character, it gives us meaningless words or words with a different meaning from the origin. The Table 1 cites some examples of the erroneous words existing in the corpus written in Latin/Tifinagh characters, and gives the correct words in the two last columns.

Table 1. Example of the erroneous word in the corpus

Word In Corpus	Transcoded Word	Tifinaghe Correct Word	Latin Correct Word
amaziG	ⵎⵣⵓⵎⵉⵎⵓ	ⵎⵣⵓⵎⵉⵎ	amaziŷ
nG	ⵎⵓ	ⵎⵓ	nŷ
Akkw	ⵎⵓⵕⵓⵎⵓ	ⵎⵓⵕⵓ	akkw
tdggwat	ⵜⵏⵏⵓⵎⵓⵎⵓⵜ	ⵜⵏⵏⵓⵎⵓⵎⵓⵜ	tdggwat
iD	ⵉⵎ	ⵉⵎ	id
iZlmD	ⵉⵎⵓⵕⵓⵎⵓ	ⵉⵎⵓⵕⵓⵎ	izlmd
iRaZaGn	ⵉⵕⵓⵎⵓⵎⵓⵎⵓ	ⵉⵕⵓⵎⵓⵎⵓⵎⵓ	irazaŷn
lmasiH	ⵎⵓⵎⵓⵎⵓ	ⵎⵓⵎⵓⵎ	lmasih
miSR	ⵎⵓⵎⵓⵎ	ⵎⵓⵎⵓⵎ	miŷr
Eica	ⵉⵎⵓ	ⵉⵎⵓ	eica
iHyaWi	ⵉⵎⵓⵎⵓⵎⵓ	ⵉⵎⵓⵎⵓⵎⵓ	ihyaw
uTumatik	ⵉⵎⵓⵎⵓⵎⵓ	ⵉⵎⵓⵎⵓⵎⵓ	utumatik

4. ALGORITHMS AND RESULTS

4.1. Deep learning algorithms

4.1.1. Recurrent neural network (RNN)

A recurrent neural network (RNN) is a class of artificial neural networks where associations between nodes structure a coordinated diagram along a transient arrangement. This allows it to show fleeting unique conduct. As obtained from the feedforward neural network, RNNs can use their inner state (memory) to handle variable length groupings of information sources. This makes them material to different tasks, for example, unsegmented, connected handwriting recognition or speech recognition [21].

The following equation shows the unrolled RNNs, the yield h_t at specific time step "t".

$$h_t = \sum_{t=0}^T X_t W_t + U h_{t-1} + b \tag{1}$$

h_{t-1} is previous output and X_t is current input and W_t represents weight at the time step t , also U represents weight associated with output h_{t-1} and b represents bias terms.

4.1.2. Long short-term memory networks (LSTM)

Hochreiter and Schmidhuber [22] proposed the long short-term memory cell (LSTM) for its favorable position in the quicker union and identifying and remembering long-term dependencies. Notwithstanding the short-term state, the LSTM cell h_t has a long-term state. The two states are elements of the current output and the past states as summed up in the accompanying C_t functions. The cell yield is just the short-term state.

$$\begin{aligned}
 f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}C_{t-1}), \\
 i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}C_{t-1}), \\
 o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}C_{t-1}), \\
 C_t &= f_t \odot C_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1}), \\
 h_t &= o_t \odot \tanh(C_t),
 \end{aligned}
 \tag{2}$$

x is the input vector, h is the output vector and c are the cell state. The subscript t represents the current time and $t-1$ is the last time. σ Is a sigmoid function, \odot is the Hadamard product and W represents undetermined parameters [23]. In (2), f is the forget gate that decides what information needs to be discarded from the cell state. i is the input gate that decides what information needs to be stored in the cell state. O Is the output gate that decides what information to output [22].

4.2. Training techniques

Train–test split: the primary thought (train/test split) is to divide the dataset into two sections: a training set, and a test set. A while later, use the training set to prepare the model [24] This will permit us to ascertain the presentation of the test set, and the outcome will be a decent estimate of the exhibition on obscure information. The informational index is isolated into a training set and a test set. The test set is not used to train the model, however, just to assess it. In our case the division we make is not arbitrary, for instance, a subset of our information incorporated the lone content of a specific class. We trained all the algorithms on 80% of the corpus and we tested them by 20% of the corpus. This will result in an "overfitting" overfitting is a measurable examination that compares to intently or precisely to a specific arrangement of

information. Consequently, this investigation may not relate to extra information or dependably anticipate future perceptions. We are attempting to stay away from that with cross-approval.

Cross-validation: is a strategy that permits you to use the dataset for preparation and validation, it fills in as follows: The dataset is parted into k parts or overlap. Every one of the k folds is used then as a test set. The rest (the association of the k-1 different parts) is used for training. Eventually, each point (or perception) is used once in a test set (k-1) is used in a training set [25].

4.3. Proposed model for Amazigh part of speech

We have the privilege to apply the deep learning techniques on the Amazigh language to predict part of speech of words written in Tifinagh characters using the open-source library Keras. In this article, we used two algorithms for deep learning which have the same layers. The first layer is input layer, it accepts vectors of shape (28) and matches our X variable (we have 28 tokens in each of our sequences train and test).

Next, we have the embedding layer. This layer will take each of our tokens/words and turn it into a dense vector of size 100. Think of it as a giant lookup table (or dictionary) with tokens as keys and the actual vectors as values. This lookup table is trainable, i.e., each epoch during the model training; we updated those vectors to match out the input. After the embedding layer, our input turns from a vector of length 28 to a matrix of size (28, 100). Each of the 28 tokens now has a vector of size 100.

Once we have this, we can use the LSTM layer (or the simple RNN layer) that for each token will look both ways in the sentence. The output of this layer is a matrix of size (28, 64). Next, we have dropout layer that is a matrix of size (28, 64). Finally, we have a time distributed dense layer. It takes the (28, 64) matrix of the LSTM (or simple RNN) layer output. Table 2 and Table 3 shows our model layers. In the model we chose to use BatchNormalization layer, the rectified linear unit (ReLU) function in the hidden layer, and the softmax function in the output layer, we used Adam methods as an optimizer.

Table 2. Proposed LSTM model architecture

Layer	Output Shape	parameters
InputLayer	[(None,28)]	0
Embedding	(None,28,100)	809100
LSTM	(None,64)	42240
Dropout	(None,28,64)	0
Dense	(None,28,32)	2080

Table 3. Proposed RNN model architecture

Layer	Output Shape	parameters
InputLayer	[(None,28)]	0
Embedding	(None,28,100)	809100
SimpleRNN	(None,64)	10560
Dropout	(None,28,64)	0
Dense	(None,28,32)	2080

The embedding layer that received as an input a vector of length 28 and turns to a matrix of size (28, 100), has 809,100 parameters, the LSTM layer has 42,240, the last layer has 2,080 parameters. So, the number total of parameters is: 853 420, the trainable parameters: 853 420, and non-trainable parameters: 0. The embedding layer received as an input a vector of length 28 and turns to a matrix of size (28, 100) has 809,100 parameters, the simple RNN layer has 10,560 and the last layer has 2,080 parameters. So, the total number of parameters is: 821,740, the trainable parameters: 821,740, and non-trainable parameters: 0.

4.4. Discussion

In this study, we fixed the size cells of RNN layer in 64, and we varied the batch size between 32 and 128. We chose the number of epochs when the accuracy stopped improving on the test sets. Training usually stops after 10 epochs.

The Table 4 contains the results of the different size batch using RNN, these results show that the accuracy using recurrent neural networks (RNN) gave 95.11% when we chose 32 in the size batch and 95.20% in 64 and 95.05% in 128. Therefore, the best size batch is 64. The Figure 1 indicates the training and testing accuracy of recurrent neural network algorithm using 64 size cells and 64-size batch in 10 epochs. We observed that the average loss in the last epochs is 4.79%.

For the LSTM networks, we fixed the size cells in 64, and we varied the batch size between 32 and 128 in 10 epochs. The Table 5 contains the results of the different size batch using LSTM, these results show that the best accuracy, using Long short-term Memory networks (LSTM) is 97.88% in batch size 128, where the accuracy given is 97.81% by 32-size batch and 64-size batch. Therefore, the best size batch is 128.

Table 4. POS tagging accuracy using RNN

Size batch	RNN accuracy
32	95.11%
64	95.20%
128	95.05%

Table 5. POS tagging accuracy using LSTM

Size batch	LSTM accuracy
32	97.81%
64	97.81%
128	97.88%

The Figure 2 indicates the training and testing accuracy of Long Short-term memory algorithm using 64 size cells and 128-size batch in 10 epochs (Figure 2). We observed that the average loss in the last epochs is 2.11%. The Figure 3 indicates the different variations of the Precision, Recall and F1-score in the two proposed models (RNN and LSTM), for the part of speech tag set. We observed that the majority of POS tag has the similar values in precision, recall, F1-score, except particle (PROT) residual (ROT), and demonstrative pronoun (PDEM), which have low values, that is caused by the missing words of these types.

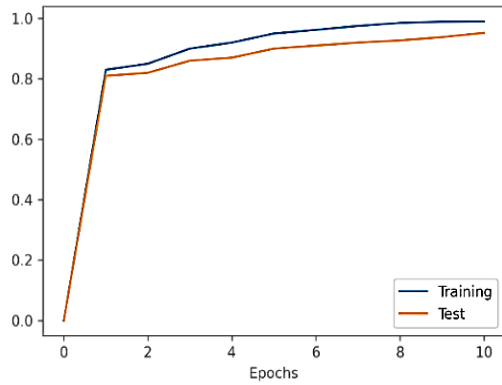


Figure 1. RNN training and testing accuracy using 64 size cells and size batch in 10 epochs

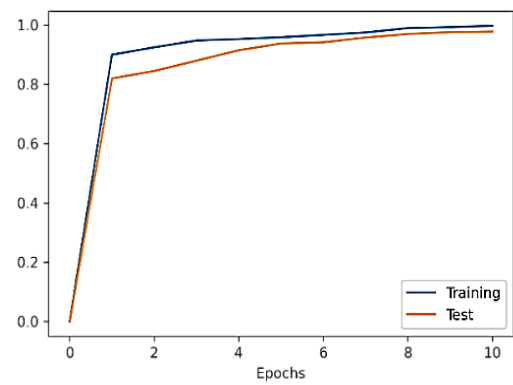


Figure 2. LSTM training and testing accuracy using 64 size cells and 128 size batch in 10 epochs

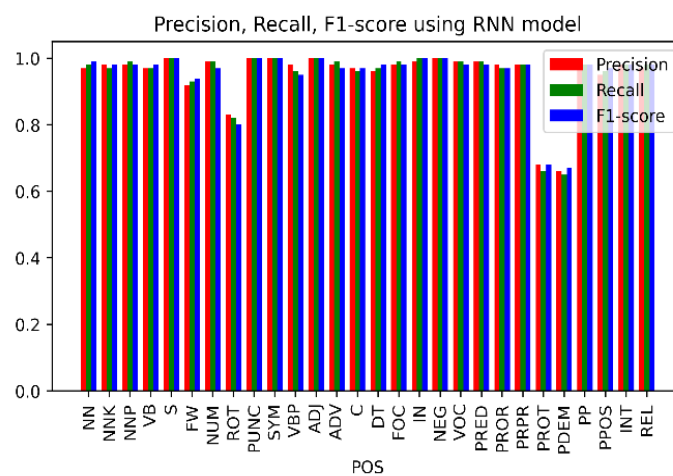
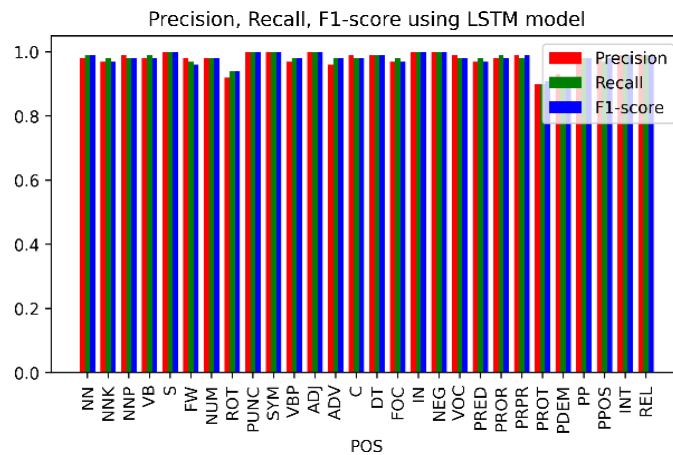


Figure 3. Precision, recall, F1-score for RNN and LSTM models

To be objective, we compared our LSTM model and RNN model to other model which used the same dataset in the same conditions, for the epoch number we used; 10 epochs for all systems. The table below summarizes the obtained results. As shown in Table 6, in terms of accuracy, precision, recall, or f1-score, our LSTM gives better results compared to other systems 97.88% in f1-score. This indicates that our system is a very powerful classifier in the Amazigh part of speech. The architecture proposed in [16] closer to this model in terms of accuracy gave 97.87%. But the difference in the number of layers used in the two models. The model proposed in [16] used 6 layers and our model used just 5 layers and is used in the hyperbolic tangent as an activation function in the input layer in our model we used ReLU function. In the output layer, we used the softmax function, the model proposed in [16] used the sigmoid function, as an optimizer we used the Adam optimizer in the two models.

Table 6. Comparison of our LSTM and RNN models with other methods (accuracy, precision, recall, f1-score)

Methods	Accuracy	Precision	Recall	F1-score
Decision Tree[16]	0.876	0.876	0.842	0.832
CRF[16]	0.934	0.934	0.935	0.933
LSTM[16]	0.9787	0.9786	0.9787	0.9787
Our RNN model	0.9520	0.9520	0.9521	0.9520
Our LSTM model	0.9788	0.9789	0.9789	0.9788

Regarding the execution time during training in Table 7, our systems have shown their efficiency. In the training phase, our RNN system very fast compared to LSTM models: 44.36% compared to the proposed LSTM in [16] and 30.40% compared to our LSTM model, and our LSTM model is very fast by 20.05% compared to the LSTM proposed in [16]. This big difference in execution time is explained by the large number of parameters used in LSTM architecture [16] which used 2,523,777 parameters compared to our models. In the RNN, we used 821,740 parameters and in the LSTM model, we used 853,420 parameters.

Table 7. Comparison of our LSTM and RNN models with other methods (number of parameters, training time)

Methods	Parameters	Training time (s)
LSTM[16]	2,523,777	65.32
Our RNN model	821,740	36.34
Our LSTM model	853,420	52.22

5. CONCLUSION

Amazigh part of speech is one of the most interesting subjects treated by several researchers, in particular Moroccan researchers. In this article, we used deep learning algorithms to label words written in Tifinagh characters. We compared our RNN and LSTM models with existing deep learning methods and machine learning methods, we found that our RNN model is better than CRF and decision tree (machine learning algorithms) that it applied in the same dataset and in the same conditions.

By comparing the deep learning algorithms with each other, we see that our recurrent neural network model is weak compared to the other Long short-term memory LSTM model in terms of accuracy but time of execution is the best one. This is because the gradient of the loss function decays exponentially with time (the vanishing gradient problem). This shows that LSTM networks can capture knowledge of a language because LSTM units include a 'memory cell' that can maintain information in memory for long periods. A set of gates is used to control when information enters the memory when it is output, and when it is forgotten. This architecture lets them learn longer-term dependencies, when we compared our LSTM model and the the other LSTM model, we found that our model is better in terms the accuracy and in terms of execution time. This has a tremendous preferred position, particularly for being used on dialects on which we do not have many language studies. As a perspective, we improved the obtained results by proposing other approaches to label words written in Tifinagh characters and treat other tasks in the same domain.

REFERENCES

- [1] M. Boukabous and M. Azizi, "A comparative study of deep learning based language representation learning models," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 2, pp. 1032-1040, 2021, doi: 10.11591/ijeecs.v22.i2.pp1032-1040.

- [2] D. Jurafsky and J. H. Martin, *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*. New Jersey: USA: Prentice Hall, 2000.
- [3] V. Gupta, N. Joshi, and I. Mathur, "CRF based Part of Speech Tagger for Domain Specific Hindi Corpus," *IJCA Proceedings on National Conf. on Contemporary Computing*, no. 2, 2017, pp. 14-18.
- [4] T. Tan, B. Ranaivo-Malançon, L. Besacier, Y. Yeong, K. H. Gan, and E. K. Tang, "Evaluating LSTM Networks, HMM and WFST in Malay Part-of-Speech Tagging," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 9, no. 2-9, pp. 79-83, 2017.
- [5] K. Kumar and D. P. Gandhmal, "An intelligent indian stock market forecasting system using LSTM deep learning," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 2, pp. 1082-1089, 2021, doi: 10.11591/ijeecs.v21.i2.pp1082-1089.
- [6] A. M. S. Omar, M. K. Osman, M. N. Ibrahim, Z. Hussain, and A. F. Abidin, "Fault classification on transmission line using LSTM network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, pp. 231-238, 2020, doi: 10.11591/ijeecs.v20.i1.pp231-238.
- [7] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, vol. 1, 2003, pp. 173-180. doi: 10.3115/1073445.1073478.
- [8] J. Giménez and L. Márquez, "SVMTool: A general POS Tagger Generator Based on Support Vector Machines," in *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, 2004, pp. 43-61.
- [9] M. Biniz and R. El Ayachi, "Recognition of Tifinagh Characters Using Optimized Convolutional Neural Network," *Sens Imaging*, vol. 22, no. 28, 2021, doi: 10.1007/s11220-021-00347-1.
- [10] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001, pp. 282-289.
- [11] Y. Tsuruoka, J. Tsurui, and S. Ananiadou, "Fast Full Parsing by Linear-Chain Conditional Random Fields," in *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, 2009, pp. 790-798, doi: 10.3115/1609067.1609155.
- [12] T. Brants, "TnT: A Statistical Part-of-speech Tagger," *Sixth Applied Natural Language Processing Conference*, 2000, pp. 224-231. doi: 10.3115/974147.974178.
- [13] M. Outahajala, Y. Benajiba, P. Rosso, and L. Zenkour, "POS Tagging in Amazighe Using Support Vector Machines and Conditional Random Fields," *International Conference on Application of Natural Language to Information Systems*, 2011, pp. 238-241, doi: 10.1007/978-3-642-22327-3_28.
- [14] S. Amri and L. Zenkour, "Amazigh POS Tagging Using TreeTagger: A Language Independent Model," *International Conference on Advanced Intelligent Systems for Sustainable Development*, pp. 622-632, 2019, doi: 10.1007/978-3-030-11928-7_56.
- [15] S. Amri, L. Zenkour, and R. Benkhouya, 'Amazighe POS tagging using Long Short Term Memory Networks', in *Proceedings of the 4th International Conference on Big Data and Internet of Things*, no. 43, 2019, pp. 1-5, doi: 10.1145/3372938.3372981.
- [16] O. Maarouf and R. El Ayachi, "Part-of-Speech Tagging Using Long Short Term Memory (LSTM): Amazigh Text Written in Tifinaghe Characters," *International Conference on Business Intelligence*, pp. 3-17, 2021, doi: 10.1007/978-3-030-76508-8_1.
- [17] A. Samir, Z. Lahbib, and O. Mohamed, "Amazigh PoS Tagging Using Machine Learning Techniques," *Proceedings of the Mediterranean Symposium on Smart City Applications*, pp. 551-562, 2018, doi: 10.1007/978-3-319-74500-8_51.
- [18] S. Amri, L. Zenkour, and M. Outahajala, "Amazigh Part-of-Speech Tagging Using Markov Models and Decision Trees," *International Journal of Computer Science & Information Technology*, vol. 8, no. 5, pp. 61-71, 2016, doi: 10.5121/ijcsit.2016.8505.
- [19] M. Outahajala, L. Zekouar, P. Rosso, and M. M. Antònia, "Tagging Amazigh with AnCoraPipe," *Proceeding of the Workshop on Language Resources and Human Language Technology for Semitic Languages*, 2010, pp. 52-56.
- [20] M. Outahajala, L. Zekouar, and P. Rosso, "Building an annotated corpus for Amazighe". In *Will appear In Proc. of 4th International Conference on Amazigh and ICT*, 2011.
- [21] R. M. Hanifa et al., "Voiced and unvoiced separation in malay speech using zero crossing rate and energy," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 2, pp. 775-780, 2019, doi: 10.11591/ijeecs.v16.i2.pp775-780.
- [22] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [23] M. Zulqarnain, R. Ghazali, Y. M. Mohamad Hassim, and M. Rehan, "A comparative review on deep learning models for text classification," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 1, pp. 325-335, 2020, doi: 10.11591/ijeecs.v19.i1.pp325-335.
- [24] I. Jamaledyn and M. Biniz, "Contribution to Arabic Text Classification Using Machine Learning Techniques," *International Conference on Business Intelligence*, pp. 18-32, 2021, doi: 10.1007/978-3-030-76508-8_2.
- [25] C. Bergmeir, R. J. Hyndman, and B. Koo, "A note on the validity of cross-validation for evaluating autoregressive time series prediction," *Computational Statistics & Data Analysis*, vol. 120, pp. 70-83, Apr. 2018, doi: 10.1016/j.csda.2017.11.003.

BIOGRAPHIES OF AUTHORS

Otman Maarouf: received his master's degree in business intelligence in 2018 from the Faculty of Science and Technology, University Sultan Moulay Sliman Beni-Mellal. He is currently a PhD degree student. His research activities are located in the area of natural language processing specifically; it deals with Part of speech, named entity recognition, and limatization-steaming of the Amazigh language.



Rachid El Ayachi obtained a degree in Master of Informatic Telecom and Multimedia (ITM) in 2006 from the Faculty of Sciences, Mohammed V University (Morocco) and a Ph.D. degree in computer science from the Faculty of Science and Technology, Sultan Moulay Slimane University (Morocco). He is currently a member of laboratory TIAD and a professor at the Faculty of Science and Technology, Sultan Moulay Slimane University, Morocco. His research focuses on image processing, pattern recognition, machine learning, and semantic web.



Mohamed Biniz: received his master's degree in business intelligence in 2014 and Ph.D degree in computer science in 2018 from the Faculty of Science and Technology, University Sultan Moulay Sliman Beni-Mellal. He is a professor at polydisciplinary faculty University Sultan My Slimane Beni Mellal morocco. His research activities are located in the area of the semantic web engineering and deep learning specifically, it deals with the research question of the evolution of ontology, big data, natural language processing, and dynamic programming.