MATHEMATICAL MODELING AND COMPUTING, Vol. 11, No. 3, pp. 741–751 (2024)

Improving Amazigh POS tagging using machine learning

Amri S.[1], Bani R.[2], Zenkouar L.[2], Guennoun Z.[2]

[1] *ENSAM School, Moulay Ismail University, Meknes, Morocco*
[2] *ERSC, EMI School, Mohammed V University, Rabat, Morocco*

(Received 8 December 2023; Revised 20 July 2024; Accepted 25 July 2024)

Tamazight, Berber, and Amazigh are the multiple names for the same language. It covers a great geographical area including the north of Africa, Sahara Sahel. It is spread principally in Morocco, Algeria, Tunisia, and Mali. In terms of natural language processing, it is considered a low-resource language. This paper presents multiple applications of different machine learning algorithms for part-of-speech tagging Amazigh for the first time. Those algorithms include trigrams 'n' tags (TnT), Brill tagging, hidden Markov model (HMM), Unigram, Bigram, Unigram + Bigram,and conditional random fields (CRF). Also, we present a part-of-speech tagger using CRF with our function of extracting features from the Amazigh language. The importance of finding a performant POS tagger for the Amazigh is to enrich its corpus, which is a main step for other NLP applications. In this research, we used 60000 tokens of annotated Amazigh corpus with 28 tags, and we realized the necessary processing step on it to be in an adequate form for feeding each model. A detailed comparison of the performance results is presented to establish the best one and the results show that our application of CRF model outperforms other techniques.

Keywords: *POS tagging; NLP; Amazigh language; machine learning.*

2010 MSC: 68T50 **DOI:** 10.23939/mmc2024.03.741

1. Introduction

Natural language processing (NLP) is the field that uses the computer science applications in language. NLP aims to find the most accurate machine learning models that can simulate human capacities of talking and understanding natural language like English, Arab, Amazigh, etc. Nowadays, NLP has become the center of research in multiples language and has multiple subfields starting from basic tools like part-of-speech tagging and named entity recognition to sophisticated applications like sentiment analysis and translation etc.

In this paper, we present multiple Amazigh tagging models including models that have not been applied to Amazigh yet, such as Trigrams-n-tags (TnT) [1], n-gram, as well as combining some models to seek the better performance, including conditional random fields (CRF) [2], Brill's tagging [3] and hidden Markov model (HMM) [4]. Part-of-speech tagging (POS tagging) is the first primary step for all NLP applications. It aims to attribute each word in a sentence or text, via a machine learning model, to its corresponding part of speech.

Table 1. Example of an annotated sentence.

Word	Tag	Description
Mom	NN	Noun
loves	VB	Verb
her	PPZ	Possessive pronoun
son	NN	Noun

First POS tagging systems were rule-based, which means that they used the characteristics and the grammar rules to tag the text [3]. Then, with the growth of big data, statistical and probabilistic POS tagging models become more attractive [1, 5]. Moreover, there are technics that combine the two approaches to profit from the best performance of each [5]. In NLP problems, the process begins with a dataset or corpus, and it depends on the problem we are going to solve. In POS tagging, the dataset is an annotated text or sentences where each word is labeled with its appropriate grammatical class. Those grammatical classes are organized in a list of tag-set which depends on the language. For example, in Table 1, the word "mom" is tagged with "NN", which means "a common noun".

The Amazigh language, known as Tamazight and Berber, is a branch of the family language called Afro-Asiatic (called Hamito-Semitic) [6,7]. It is spread in North Africa that includes the Sahara (from Niger to the Mediterranean, the Canary Isles). With the intention of establishing a writing system and make it standard, the Tifinagh-version IRCAM, which is a graphical system, was initiated to write the Amazigh language of Morocco. The Tifinaghe-IRCAM is a system that has multiple letters: 27 consonants, two semi-consonants, 4 vowels.

There are several syntactic classes of the Amazigh language and most of them are Nouns which is a lexical unit that is a result of combining a root according to a certain pattern. Nouns have either simple forme like ('asif', the river), or compound form ('butaghat', the goat owner), or even a derived form ('ighimi', the stay). Like other languages, Amazigh has two genders: masculine and feminine, and they could be singular or plural. Nouns in the Amazigh language have two cases: free and construct. As for Verbs, in Amazigh, it could be basic or derived verbs. Verb in basic form is formed by a root and a radical. In the case of the derived form,it has a basic form combined with one of the following prefixes morphemes: 's'/'ss' for the factitive form, 'tt' indicating the passive form, and 'm'/'mm' for the reciprocal form. In the conjugation of Amazigh Verbs, there are four aspects: perfect aspect, negative perfect aspect, aorist aspect, and imperfective aspect. Finally,we consider Particle, which is a functional word that can not designate verbs or nouns. Particles include pronouns, conjunctions, aspectual, orientation, prepositions, negative particles, adverbs and subordinates. Generally, particles are the word not inflected, except for the possessive and demonstrative pronouns (this (mas.),'wa', 'win' these (mas.)). For more information about Amazigh grammar see [25].

In this work, we present new applications of machine learning algorithms for part-of-speech tagging the Amazigh language which are TnT [1], n-gram, hidden Markov model (HMM) [4], Brill'tagger [3] and CRF [2]. To find the best performance model, we have made a detailed comparison of the results of those machine learning models. For this experimentation, we used the existing Amazigh corpus [8]. The rest of this paper is organized as follows: section 2 presents the related works on POS tagging Amazigh; section 3 describes the machines learning approaches; section 4 describes the methodology of POS tagging Amazigh; section 5 presents the results; section 6 shows the conclusion and perspectives.

2. Related works

POS tagging is an essential step in natural language processing of any language. The first research in this field was based on classical machine learning approaches. In English, early tagging is based on HMM [9] and [10], where the bidirectionality was added to achieve 97%. With the application of SVM classification algorithm in NLP, other taggers based on this model have appeared in work [11] and [12] with accuracy of 97%. Before that, the statistical tagger TnT [1] reached 96.7% of accuracy. Same in other known languages, the accuracy reaches more than 97% using those classical machine learning approaches. Using CRF, in [13] they reached 97% in French POS tagging.

On the other hand, the low-resource language, knows an augmentation on natural language processing works and POS is not an exception. In the Indian language, especially the Maithili language, [14] describes a CRF based tagger with accuracy of 85.88%. In [15], they developed a SVM-based tagger for Malayalam language reaching 94%. As for Urdu, multiple works have been done on POS tagging such as [16], where they use n-gram model with an accuracy of 95%. Also, [17] tested different machine learning models (SVM, TnT, TreeTagger, RF-tagger) with the best accuracy of 95.66% with SVM.

For the Amazigh language, we note the fact that it is a very low-resources language and the interest of developing its natural language processing is quite new (from 2011), and especially in POS tagging the Amazigh. With the corpus of 60k tokens [8], they realized different part-of-speech tagging models such as SVM, CRF, Tree-Tagger [19] with the best accuracy of 89.26% for TreeTagger [18]. The HMM model and the decision tree have been tested on Amazigh POS tagging in [20] with 80% accuracy. As well as finding the best performance model, the combination of three tagging systems (SVM, CRF, TreeTagger) which called Combitagger [21] has been test achieving 89% of tagging accuracy [22].

3. Methodology

In this section, we present the different machine learning models that we used in our experimentation on Amazigh part-of-speech tagging to find the most accurate.

3.1. Hidden Markov model

Starting from a sequence of words (sentence), the objective is to find the most probable sequence of tags for this sentence. HMM tagger aims to find the sequence of tags that maximize the probability:

$$p(\text{word} \,|\, \text{tag}) \times p(\text{tag} \,|\, \text{previous tags}).$$

HMM [23] is characterized by tagging a sequence rather than one single word. Starting from a sequence of words $= w_1, w_2, \ldots, w_n$, we want to find the tags $= t_1, t_2, \ldots, t_n$ that maximize $p(\text{tags} \,|\, \text{words})$ which is $p(\text{tags}) \times p(\text{words} \,|\, \text{tags})$ by applying byes law. And using Markov [24] assumption,

$$p(\text{tags} \,|\, \text{words}) = p(t_1)\, p(t-2 \,|\, t_1) \prod_{i=3}^{n} p(t_i \,|\, t_{i-1}) \left(\prod_{i=1}^{n} p(w_i \,|\, t_i) \right).$$

3.2. *N*-gram model

In this section, we work with the trigram and bigram models, the probability $p(t_1, t_2, \ldots, t_n)$ of the apparition of this sequence $t_1, t_2, \ldots, t_n$ is given in these equations:
– For the trigram model:

$$p(t_1, t_2, \ldots, t_n) = \prod_{i=1}^{n} q(t_i \,|\, t_{i-2} t_{i-1}),$$

– For the bigram model:

$$p(t_1, t_2, \ldots, t_n) = \prod_{i=1}^{n} q(t_i \,|\, t_{i-2}).$$

Using the probability estimation

$$q(t_i \,|\, t_{i-2}, t_{i-1})) = \frac{c(t_{i-2}, t_{i-1}, t_i)}{c(t_{i-2}, t_{i-1})},$$

$$q(t_i \,|\, t_{i-1}) = \frac{c(t_{i-1}, t_i)}{c(t_{i-1})},$$

where $c(t_{i-2}, t_{i-1}, t_i)$ is the number of times the trigram t_{i-2}, t_{i-1}, t_i has been seen in the training corpus, and $c(t_{i-2}, t_{i-1})$ is the number of times the bigram t_{i-2}, t_{i-1} has been seen in the training corpus, and $c(t_{i-1}, t_i)$ is the number of times the bigram t_{i-1}, t_i has been seen in the training corpus, and $c(t_{i-1})$ is the number of times the unigram t_{i-1} has been seen in the training corpus.

3.3. Conditional random fields (CRF)

Conditional random fields (CRF) is a framework developed by Lafferty [2] to build probabilistic models for segmenting and tagging a sequence of data. CRF has multiple advantages compared to Hidden Markov Model and does not assume the total independence of probability distribution. HMMs are generative models, which means that they compute a joint probability of both observation and POS tag. This requires the calculation of all possible observation sequences, which is not a simple task. Thus, the interest of using a conditional model instead. Conditional model aims to compute the probability of a possible tag given a certain observation. Figure 1 presents a graphical representation of CRF structure.

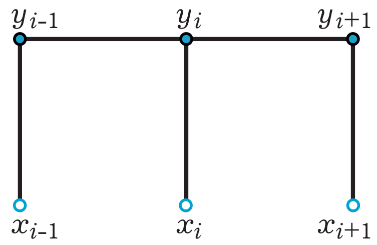

Fig. 1. Graphical structure of chained structured conditional random fields.

Let $X = X_1, X_2, \ldots, X_n$ be a sequence of words or tokens and $Y = Y_1, Y_2, \ldots, Y_n$ be a sequence of part-of-speech tags. The joint distribution of Y given X is

$$p_\theta(Y \,|\, X) \propto \exp\Bigg(\sum_{e\in E,k} \lambda_k f_k(e, y \,|\, e, x) + \sum_{v\in V,k} \mu_k g_k(v, y \,|\, v, x)\Bigg),$$

where $g = (V = 1, 2, \ldots, m, E = (i, i+1))$ is a chain, $y \,|\, s$ is a set of the components of y associated with the vertices in subgraph S. The features f_k and g_k are assumed to be given and fixed. The problem to resolve is the estimation of parameters $\theta = (\lambda_1, \lambda_2, \ldots; \mu_1, \mu_2, \ldots)$ from the dataset of training $\wp = \big\{(x^{(i)}, y^{(i)})\big\}_{i=1}^N$ characterized with empirical distribution $\hat{p}(x, y)$. The log-likelihood objective function is given as follow:

$$\begin{aligned}\mathcal{T}(\theta) &= \sum_{i=1}^{N} \log p_\theta\big(y^{(i)} \,|\, x^{(i)}\big)\\ &\propto \sum_{x,y} \hat{p}(x, y) \log p_\theta(x \,|\, y).\end{aligned}$$

Although it encloses the HMM, CRF model is more expressive because it takes into consideration arbitrary dependencies in the sequence of observations. Also, the model can perform well even with less training data. The training of the data is done using the convexity of the loss function. To simplify some expressions, two states were created, the start state $Y_0 =$ start and the stop state $Y_(n+1) =$ stop. The conditional probability of sequence of tags is computed using a matrix form. For each position i of an observation sequence x, we define $\wp \times \wp$ a matrix of random variable $M_i(x) = [M_i(y', y \,|\, x)]$ with

$$M_i(y', y \,|\, x) = \exp\big(\psi_i(y', y \,|\, x)\big),$$

where $\psi_i(y', y \,|\, x) = \sum_k \lambda_k f_k(e_i, y \,|\, e_i = (y', y), x) + \sum_k \mu_k g_k(v_i, y \,|\, v_i = y, x)$, e_i is the edge (Y_{i-1}, Y_i) and v_i is the vertex Y_i. Finally, the conditional probability of a sequence of tags y is given by

$$p_\theta(y \,|\, x) = \frac{\prod_{i=1}^{n+1} M_i(y_{i-1}, y_i \,|\, x)}{Z_\theta(x)},$$

where $Z_\Theta(x)$ is the normalization or partition function of (start, stop):

$$Z_\theta(x) = \Bigg(\prod_{i=1}^{n+1} M_i(x)\Bigg)_{\text{start,stop}},$$

$y_0 =$ start, $y_{n+1} =$ stop.

Parameters estimation in CRF. Given a data training data $T = \big\{X^{(i)}, Y^{(i)}\big\}_{i=1}^N$, where each $X^{(i)} = \big\{x_1^{(i)}, x_2^{(i)}, \ldots, x_T^{(i)}\big\}$ is a sequence of inputs and $Y^{(i)} = \big\{y_1^{(i)}, y_2^{(i)}, \ldots, y_T^{(i)}\big\}$ is the corresponding prediction. To find the best estimation of parameter θ that maximize the log-likelihood of the training data:

$$\ell(\theta) = \sum_{i=1}^{N} \log p\big(y^{(i)} \,|\, x^{(i)}\big)$$

using the CRF probability expression:

$$\ell(\theta) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K} \theta_k f_k\big(y_t^{(i)}, y_{t-1}^{(i)}, x_t^{(i)}\big) - \sum_{i=1}^{N} \log Z\big(x^{(i)}\big).$$

When there is a great number of parameters like hundreds of thousands, we use a penalty regularization to avoid overfitting. We choose the Euclidean norm of θ

$$\ell(\theta) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K} \theta_k f_k\big(y_t^{(i)}, y_{t-1}^{(i)}, x_t^{(i)}\big) - \sum_{i=1}^{N} \log Z\big(x^{(i)}\big) - \sum_{i=1}^{K} \frac{\theta_k^2}{2\sigma^2},$$

$$Z(x^{(i)}) = \sum_{y^{(i)}} \prod_{t=1}^{T} \exp\Bigg(\sum_{k=1}^{K} \theta_k f_k\big(y_t^{(i)}, y_{t-1}^{(i)}, x_t^{(i)}\big)\Bigg),$$

σ^2 is a free parameter that plays the role of determining how much of large weights to penalize.

Using partial derivative to find the parameters optimization.

$$\frac{d\ell}{d\theta_k} = \sum_{i=1}^{N}\sum_{t=1}^{T}\theta_k f_k\big(y_t^{(i)}, y_{t-1}^{(i)}, x_t^{(i)}\big) - \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{y,y'} f_k\big(y, y', x - t^{(i)}\big)\, p\big(y, y' \,|\, x^{(i)}\big) - \frac{\theta_k}{\sigma^2}.$$

3.4. Transformation-based tagging (Brill tagger)

Brill's tagger is based on rules or transformation, as he defines them, in which the grammar rules are induced from the training dataset without human intervention or linguist expertise. It is a sort of hybrid approach because the tagger uses the statistical techniques first to understand information from the training and then programs an algorithm that learns rules to reduce the statistical errors. It is called transformation-based-error-driven learning because it learns by detecting errors. The learning begins with assigning an initial annotation to a text, this annotation is compared with the true hand annotated text to induce the rules to improve the annotation. Each rule includes two parts, a condition or trigger and a resulting tag.

3.5. Trigrams 'n' Tags (TnT)

Trigrams 'n' Tags (abbreviation TnT) [1] is a statistical part-of-speech tagger that can be trainable on multiple languages with a tag set. The TnT is based on second-order Markovian using states as POS tags and outputs as words. The model aims to calculate the sequence of tags $t_1, t_2, \ldots, t_T$ that maximizes the probability for a certain sequence of tokens $w_1, w_2, \ldots, w_T$:

$$\left(\prod_{i=1}^{T} P(t_i \,|\, t_{i-1}, t_{i-2}) p(w_i \,|\, t_i)\right) P(t_{T+1} \,|\, t_T)$$

with t_{-1}, t_0 are the markers of "beginning sequence" and t_{T+1} is marker of "the end sequence". To estimate the probabilities of the transition and the outputs, a tagged corpus is used based on the maximum likely probabilities $\hat{p}$:

- Unigram:

$$\hat{P}(t_3) = \frac{c(t_3)}{N},$$

- Bigram:

$$\hat{P}(t_3 \,|\, t_2) = \frac{c(t_2, t_3)}{c(t_2)},$$

- Trigram:

$$\hat{P}(t_3 \,|\, t_2, t_1) = \frac{c(t_1, t_2, t_3)}{c(t_1, t_2)},$$

- Lexical characteristics:

$$\hat{P}(w_3 \,|\, t_3) = \frac{c(w_3, t_3)}{t_3}.$$

Smoothing. In the case of the trigram model, the probability could be zero because of there are not sufficient trigram instance to estimate the probability in reliable way. Thus, the interest of employing smoothing technics. TnT applies linear interpolation on Unigrams and Bigrams and also Trigrams to estimate the probability of the trigram:

$$P(t_3 \,|\, t_2, t_1) = \mu_1 \hat{P}(t_3) + \mu_2 \hat{P}(t_3 \,|\, t_2) + \mu_3 \hat{P}(t_3 \,|\, t_2, t_1),$$

$\hat{P}$ represents maximum likelihood and P represent the probability of distribution and $\mu_1 + \mu_2 + \mu_3 = 1$. The value of μ_1, μ_2 and μ_3 are defined using the suppression of the interpolation. This is done by removing consecutively the trigrams from the training dataset and computes best values of μ_s from the existing n-gram in the dataset. Knowing unigram, bigram, and trigram frequency, we can define efficiently the weights by time linear processing of the number of trigrams.

Unknown words. In inflected language, the best technic to handle the unknown words is the suffix analysis. The probability of a tag is assigned according to the ending of the word. Using the training dataset, the probability of a suffix is extracted from all words that have the same suffix. The term 'suffix' does not designate necessarily the meaning of suffix in linguistic. It can be the last two, three or more characters. To compute probabilities, the smoothing by chained abstraction is applied. Starting by computing the probability, t knows the last m characters c_i of a n-character word $P(t \mid c_{n-m+1}, \ldots, c_n)$. In a recursive way, the suffix omits characters for i from 0 to m:

$$P(t \mid c_{n-i+1}, \ldots, c_n) = \frac{\hat{P}(t \mid c_{n-i+1}, \ldots, c_n) + \theta_i P(t \mid c_{n-i}, \ldots, c_n)}{1 + \theta_i}.$$

The calculation of the maximum likelihood $\hat{P}(t \mid c_{n-i+1}, \ldots, c_n)$ is extracted from the corpus of training:

$$\hat{P}(t|c_{n-i+1}, \ldots, c_n) = \frac{f(t, c_{n-i+1}, \ldots, c_n)}{f(c_{n-i+1}, \ldots, c_n)}.$$

To identify the best value with respect to m, the longest value is used, TnT uses the approach that assumes that it depends on word itself. Thus, use the long suffix that we could find in the training dataset. It is an empirical choice. As for θ_i, it is determined without taking in consideration the context, as in case of μ_i. In TnT tagger, they choose θ_i to be the standard deviation of the unconditioned maximum likelihood probabilities of the tags in dataset of training,

$$\theta_i = \frac{1}{1-l} \sum_{j=1}^{l} \left(\hat{P}(t_j) - \bar{P}\right)^2.$$

For i from 0 to m, where l is the number of tags in the tagset and

$$\bar{P} = \frac{1}{s} \sum_{j=1}^{l} \hat{P}(t_j).$$

3.6. Model architecture

In Figure 2, we present the workflow of the process of tagging text in Amazigh. After collecting the desired text, the system proceeds to transform it into sentences. And each sentence is tokenized into words. Then, the tokenized sentences are proceeded by the model sentence by sentence to deliver tagged words in the form of sentences.

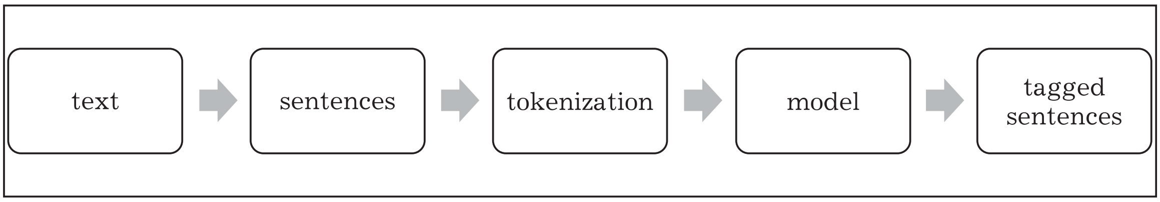

Fig. 2. Workflow of the tagging system.

The system architecture presented in Figure 2 is the same used in the machine learning models proposed in this work except for the CRF model using our function of extracting features. The proposed models in NLTK such as TnT, Brill, HMM and CRF require a text in form of tagged sentences to train the model. However, in our CRF model, we propose a special function to extract the special features of Amazigh texts. This function takes into consideration the characteristics of the Amazigh language such as non-capitalization and some special particles. The architecture of this approach is presented in Figure 3.

4. Experiments and results

After presenting the different algorithms that we have used in this research, in this section, we present different steps in experimenting each model. We start with data preparation by transforming the available corpus into an adequate format for each model. The dataset and the tag-set used in this experiment are also presented in this section.

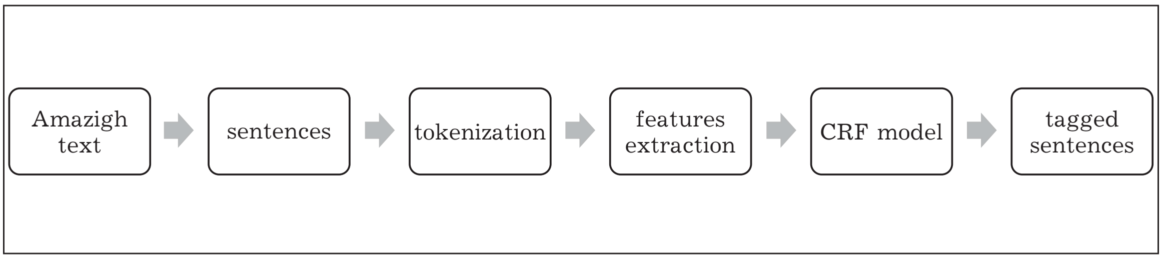

Fig. 3. CRF model with function of extracting features from Amazigh languge.

4.1. Dataset and tagset

The corpus that we used in this research is the one elaborated by [8], it is a 60k tagged tokens in csv format presented in Table 2. To explore it in our experimentation, we process it in the form of sentences of tagged word such as: [(word1, tag1), (word2, tag2), ..., (wordN)] like the common corpus such as treebank. The tag set that we used in our experimentation is presented in Table 3. The training and test dataset used in this work are respectively 80% and 20% of the global dataset, which give us 1090 sentences in training datasets and 242 sentences in test dataset.

Table 2. Characteristics of the Amazigh corpus.

Dataset	size	type	number of sentences	number of tokens
Monolingual corpus	700 MB	Utf-8	3231	60000

Table 3. The Amazigh tag-set.

Tag	Designation
NN	noun
NNK	noun represent kinship
NNP	proper noun
VB	verb in its base form
VBP	participle form of verb
ADJ	the adjective
ADV	the adverb
C	the conjunction
DT	the determiner
FOC	the focalizer
IN	the interjection
NEG	particle for negation
VOC	the vocative
PRED	particle for the predicate
PROR	particle for the orientation
PRPR	particle precedes verb
PROT	other particle
PDEM	pronoun, demonstrative
PP	pronoun, personal
PPOS	pronoun, possessive
INT	the interrogative
REL	the relative
S	the preposition
FW	strange word
NUM	the numeral
DATE	the date
ROT	the residual
PUNC	the punctuation

4.2. Experimental results

In this section, we present the results that we obtained using different tagging techniques and combining some taggers. As for the CRF model, we used a function for extracting features to be fed for the model. The features considered are the word itself, the three first and last letters. The word is numbered, and we do the same for the two words before and after. The results of these experiments are raised in Table 4 and Figure 4 below.

Table 4. Accuracy of different tagging technics for Amazigh language.

Model	Recall	F1 score	Precision	Accuracy (%)
Unigram	72.44	78.45	88.41	72.44
Bigram	9.04	16.10	88.2	9.04
Unigram+Bigram	79.30	78.91	81.75	79.3
HMM	39.5	37.2	73.5	39.5
TnT	72.96	79.21	89.18	72.86
Brill	80.36	79.74	82.5	80.36
CRFsuite NLTK	82.18	81.67	81.9	82.18
CRFsuite with our function of features extraction	87.5	87.1	87.2	87.46

As shown in Table 4 and Figure 4, the performance of some statistical taggers is not suitable for the Amazigh subtypes such as Bigram (with just 9% of accuracy) and HMM (with just 39% of accuracy). Brill's tagger shows good results with 80.36% accuracy compared to TnT which has 72.86% of accuracy, with an approximative results with Unigram+Bigram that has 79.3%. CRFsuite NLTK has greater results (82.18% accuracy), however, our CRFsuite model with our function of extraction features outperforms it with 87.46%.

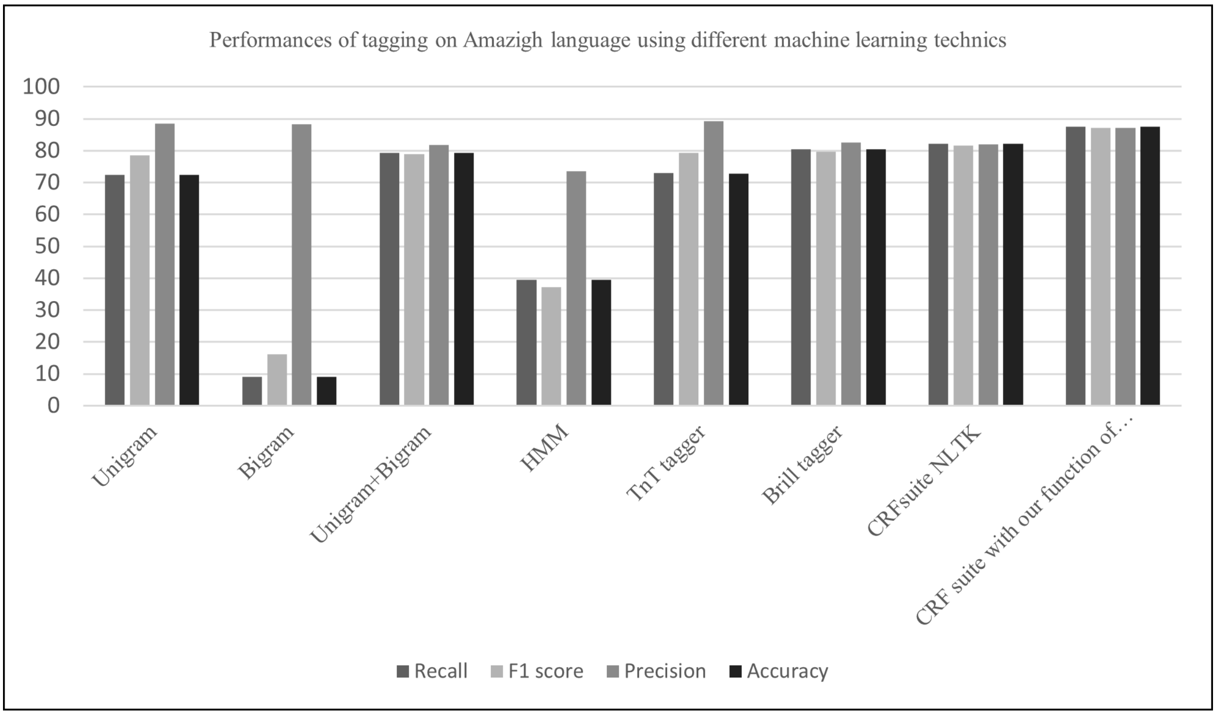

Fig. 4. Performances of the different tagging systems for Amazigh language.

To highlight the effect of the size of the available dataset on the performance of the tagging, we realize multiple experiments on tagging systems using different sizes of datasets. The accuracy of each system is presented in Table 5 and Figure 5. As we can see in Table 4 and Figure 5, the size of the dataset influences the performance of the tagging system, the increase of its size drives to the increase in the accuracy of all taggers. In CRFsuites, in the 20000 tokens dataset the accuracy is 82.3%. In the 40000 tokens dataset, the accuracy is 84.2%. And finally, in the dataset of 60000 tokens, the accuracy is 87.5%.

Table 5. Accuracy of the taggers on different sizes of dataset (%).

Model	20000 tokens	40000 tokens	60000 tokens
Unigram	63.13	66.2	72.4
Bigram	5.14	7.19	9.04
Unigram+Bigram	74.4	76	79.3
HMM	33.8	34.6	39.5
TnT tagger	65.9	66.2	72.86
Brill	74.3	76.3	80.36
CRFsuite NLTK	77.8	78.2	82.18
CRFsuite with our function of features extraction	82.3	84.2	87.5

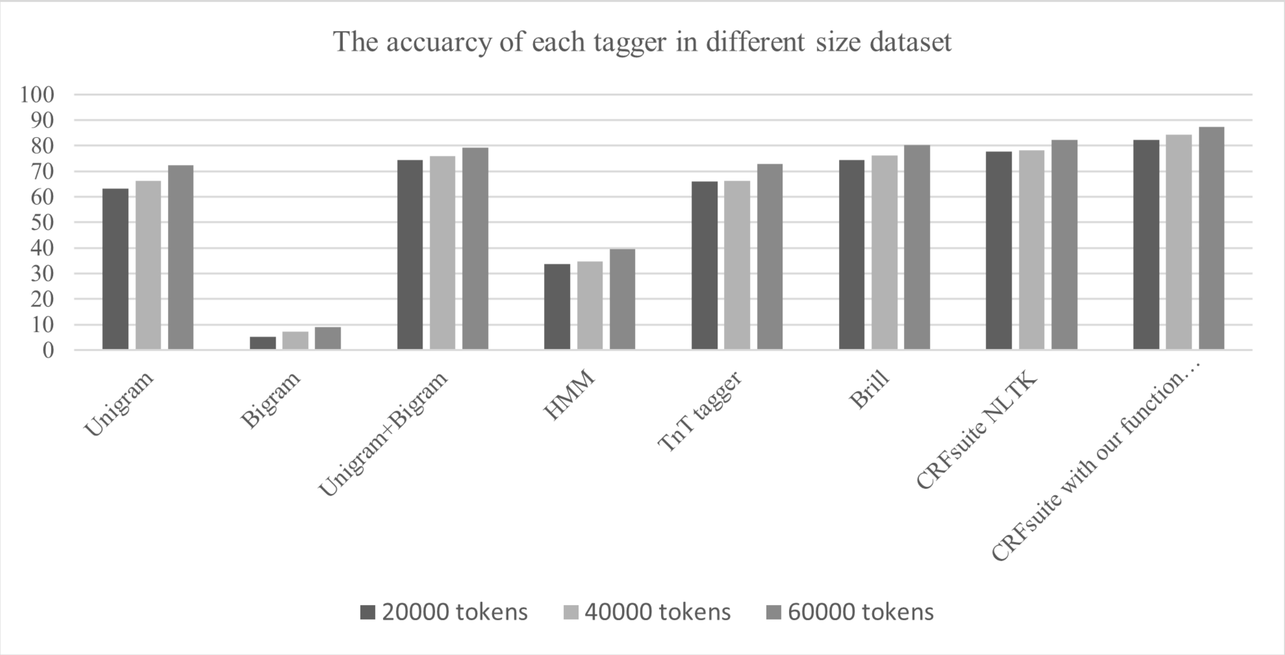

Fig. 5. The accuracy of the studied taggers on different sizes of dataset.

4.3. Baseline comparison

To evaluate our work in comparison with other researches in Amazigh language tagging, we present, in Table 6 below, a comparison of our results with the existing researches in this field. The used corpus in those works is the same that we used. The comparison to find the best accurate tagger is based on global accuracy.

As we can see in Table 6, our proposed work‘s performance on Amazigh POS tagging, using machine learning, is competitive with the existent ones. Moreover, our proposed CRF model with our function of extracting features performs better than the one proposed in [18], however the using of Combitagger offers the best results achieving 89%.

Table 6. Accuracy of the taggers on different sizes of dataset (%).

	Model	Results(%)
Existent works	Tree-tagger [18]	87.2
	CRF [18]	86.7
	SVM [18]	86.4
	Combitagger [22]	89
Our work in this paper	CRFsuit NLTK	82.18
	Brill	80.36
	CRFsuite with our feature function	87.5

5. Conclusion

In this paper, we experimented with the part-of-speech tagging of the Amazigh language using different machine learning techniques to study the performances of each technique. The results show that some techniques perform better than others, however, the size of the dataset is an important parameter in any model. We also presented an application of the CRF model using our special function of extracting Amazigh features and the results outperform the existing CRF tagger. We must insist on the fact that the performance of the classical machine learning tagging in the Amazigh language is far more than those of other rich languages. So, as a perspective, in the future, we will investigate other machine learning techniques, especially the deep learning models, for POS tagging the Amazigh language to try to find the best accuracy.

[1] Brants T. TnT – A Statistical Part-of-Speech Tagger. Preprint arXiv:cs/0003055 (2000).

[2] Lamport L., Lafferty J., McCallum A., Pereira F. C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001).

[3] Baum L. E., Petrie T. Statistical inference for probabilistic functions of finite state Markov chains. The Annals of Mathematical Statistics. **37** (6), 1554–1563 (1966).

[4] Ratnaparkhi A. A maximum entropy model for part-of-speech tagging. Conference on Empirical Methods in Natural Language Processing (1996).

[5] Spoustová D., Hajič J., Votrubec J., Krbec P., Květoň P. The best of two worlds: Cooperation of statistical and rule-based taggers for Czech. Proceedings of the Workshop on Balto–Slavonic Natural Language Processing: Information Extraction and Enabling Technologies. 67–74 (2007).

[6] Greenberg J. H. The Languages of Africa. The Hague (1966).

[7] Ouakrim O. Fonética y fonología del Bereber. Servei de Publicacions de la Universitat Autònoma de Barcelona (1995).

[8] Amri S., Zenkouar L., Outahajala M. Build a Morphosyntaxically Annotated Amazigh Corpus. BDCA'17: Proceedings of the 2nd international Conference on Big Data, Cloud and Applications. 1–7 (2017).

[9] Cutting D., Kupiec J., Pedersen J., Sibun P. A practical part-of-speech tagger. ANLC'92: Proceedings of the third conference on Applied natural language processing. 133–140 (1992).

[10] Toutanova K., Klein D., Manning C. D., Singer Y. Feature-rich part-of-speech tagging with a cyclic dependency network. NAACL'03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. 173–180 (2003).

[11] Giménez J., Màrquez L. Fast and accurate part-of-speech tagging: The SVM approach revisited. Recent Advances in Natural Language Processing III: Selected papers from RANLP 2003. 153–163 (2003).

[12] Giménez J., Màrquez L. SVMTool: A general POS tagger generator based on Support Vector Machines. Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04). (2004).

[13] Constant M., Sigogne A. MWU-aware part-of-speech tagging with a CRF model and lexical resources. Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World. 49–56 (2011).

[14] Priyadarshi A., Saha S. K. Towards the first Maithili part of speech tagger: Resource creation and system development. Computer Speech & Language. **62**, 101054 (2020).

[15] Antony P. J., Mohan S. P., Soman K. P. SVM based part of speech tagger for Malayalam. 2010 International Conference on Recent Trends in Information, Telecommunication and Computing. 339–341 (2010).

[16] Anwar W., Wang X., Li L., Wang X. L. A statistical based part of speech tagger for Urdu language. 2007 International Conference on Machine Learning and Cybernetics. 3418–3424 (2007).

[17] Sajjad H., Schmid H. Tagging Urdu text with parts of speech: a tagger comparison. EACL'09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. 692–700 (2009).

[18] Amri S., Zenkouar L., Outahajala M. A Comparison of Three Machine Learning Methods for Amazigh POS Tagging. International Journal of Scientific and Engineering Research. **8** (2), 83–87 (2017).

[19] Schmidt H. Probabilistic part-of-speech tagging using decision trees. Proceedings of International Conference on New Methods in Language Processing. 44–49 (1994).

[20] Amri S., Zenkouar L., Outahajala M. Amazigh part-of-speech tagging using Markov models and decision trees. International Journal of Computer Science & Information Technology (IJCSIT). **8** (5), 61–71 (2016).

[21] Henrich V., Reuter T., Loftsson H. CombiTagger: A System for Developing Combined Taggers. Proceedings of the Twenty-Second International FLAIRS Conference (2009).

[22] Amri S., Zenkouar L., Outahajala M. Combination POS taggers on Amazigh texts. I2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech). 1–6 (2017).

[23] Baum L. E., Petrie T., Soules G., Weiss N. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. The Annals of Mathematical Statistics. **41** (1), 164–171 (1970).

[24] Markov A. A. Essai d'une recherche statistique sur le texte du roman "Eugene Onegin" illustrant la liaison des epreuve en chain. Bulletin de l'Académie Impériale des Sciences de St.-Pétersbourg. VI serie. **7**, 153–162 (1913).

[25] Boukhris F., Boumalk A., El Houssaïn El Moujahid, Souifi H. La Nouvelle Grammaire de L'amazighe. Publications de l'Institut Royal de la Culture Amazighe (2008).

Удосконалення амазигського POS-тегування за допомогою машинного навчання

Амрі С.[1], Бані Р.[2], Зенкоуар Л.[2], Гуеннун З.[2]

[1] *Школа ENSAM, Університет Мулая Ісмаїла, Мекнес, Марокко*
[2] *ERSC, Школа EMI, Університет Мохаммеда V, Рабат, Марокко*

Тамазайтська, берберська, амазигська — це декілька назв для однієї мови, яка охоплює велику географічну територію, включаючи північ Африки, Сахару–Сахель. Поширена переважно в Марокко, Алжирі, Тунісі, Малі. З точки зору обробки природної мови, вона вважається мовою з низьким ресурсом. У цій статті вперше подано численні приклади застосування різних алгоритмів машинного навчання для тегування частин мови амазигів. Ці алгоритми включають триграми 'n' тегів (TnT), тегування Брілла, приховану модель Маркова (HMM), уніграму, біграму, уніграму + біграму, умовні випадкові поля (CRF). Крім того, представлено можливість подання тегеру частини мови за допомогою CRF із нашою функцією завантаження ознак амазигської мови. Важливість пошуку ефективного POS-тегера для амазигської мови полягає в збагаченні його корпусу, і це головний крок для інших додатків NLP. У цьому дослідженні використали 60000 токенів анотованого корпусу амазигської мови із 28 тегами і реалізували необхідний етап обробки, щоб він був у адекватній формі для подачі кожної моделі. Подано детальне порівняння результатів продуктивності, щоб визначити найкращий підхід, і результати показують, що наше застосування моделі CRF перевершує інші методи.

Ключові слова: *POS тегування; NLP; амазигська мова; машинне навчання.*