Using Confidence and Informativeness Criteria to improve POS-Tagging in Amazigh

Mohamed Outahajala[a,*], Yassine Benajiba[b] , Paolo Rosso[c], and Lahbib Zenkouar[a]

[a]*LEC, EMI, Univérsité Mohammed V, Avenue Ibnsina B.P. 765 Agdal Rabat, Morocco*
[b]*Symanto Research, New York, USA*
[c]*NLE Lab., PRHLT research center, Universitat Politècnica de València, Spain*

Abstract. Amazigh is used by tens of millions of people mainly for oral communication. However, and like all the newly investigated languages in natural language processing, it is resource-scarce. The main aim of this paper is to present our POS taggers results based on two state of the art sequence labeling techniques, namely Conditional Random Fields and Support Vector Machines, by making use of a small manually annotated corpus of only 20k tokens. Since creating labeled data is very time-consuming task while obtaining unlabeled data is less so, we have decided to gather a set of unlabeled data of Amazigh language that we have preprocessed and tokenized. The paper is also meant to address using semi-supervised techniques to improve POS tagging accuracy. An adapted self training algorithm, combining confidence measure with a function of Out Of Vocabulary words to select data for self training, has been used. Using this language independent method, we have managed to obtain encouraging results.

Keywords: POS-tagging, Amazigh, Conditional Random Fields, Support Vector Machines, Out Of Vocabulary, Self training

1. Introduction

The POS-tagging task consists of annotating each word in a sentence with its lexical category, i.e., part-of-speech. It is the first layer above the lexical level and the lowest level of syntactic analysis. Hence, most of the Natural Language Processing (NLP) tasks dealing with higher linguistic levels require POS tags, for instance: phrase chunking, word sense disambiguation, grammatical function assignment and named entity recognition [27]. In conjunction with partial parsing, POS-tagging is used in more complex tasks such as: lexical acquisition, information extraction, finding good indexing terms in information retrieval and question answering [19].

Most of the newly investigated languages in NLP are resource-scarce. Amazigh is one of the endangered languages of West Africa. However, with the emergence of an increasing sense of identity and militancy, it has been introduced in mass media and in the educational system in Morocco. On the first July 2011, Moroccans voted favorably for the new constitution; therefore, the Amazigh language became an official language along with Arabic. In the last ten years, The Royal Institute for Amazigh Culture (IRCAM), together with other associations and authors have published an important number of books related to the Amazigh language and culture. However, this language, and like most non-European languages, still suffers from the scarcity of language processing tools and resources.

Enhancing the performance of taggers when trained on small manually annotated datasets is of great importance. However, obtaining hand labeled data is time consuming and requires significant human effort in the annotation process [28, 42], especially for languages with scarce resources such as Amazigh. To overcome these issues, other techniques are used, namely: unsupervised strategies where no data is labeled and all annotations are discovered [21], and semi-supervised learning paradigms, where labeled data are used to annotate unlabeled data. Examples of these techniques include self-training [11, 43] and co-training [6]. Active learning, which can be seen as an interactive semi-supervised technique, is also used to reduce annotation cost [35, 36]. In this paper we present Amazigh POS-tagging results based on a small corpus with a

[*] Corresponding author. E-mail: outahajala1@yahoo.fr.

tag set of 28 tags, then we experiment some confidence measure variants used to select unlabeled data for self-training our model, and an adapted form of self training algorithm heavily relying on Out Of Vocabulary (OOV) and confidence measure.

The rest of this paper is organized as follows: in Section 2 we give an overview of Amazigh language and its NLP related works. The third section presents manual annotation process and POS-tagging results. In Section 4 we present confidence measure effectiveness for self training our model based on unlabeled data, informativeness effectiveness and how we combine confidence and informativeness criteria for self training. Finally, in Section 5 we draw some conclusions.

2. Amazigh Language and NLP Related works

In this section we will present a brief description of the Amazigh language and NLP Amazigh related works.

2.1. Amazigh language brief description

Amazigh is spoken in Morocco, Algeria, Tunisia, Libya, and the Egyptian Oasis Siwa; it is also spoken by many other communities in parts of Niger and Mali and by immigrant Amazigh communities in Europe and over the world. In Morocco, it is used by tens of millions of people mainly for oral communication, and has been introduced in mass media and in the educational system. Its writing system for Amazigh is Tifinagh. It does not have capitalization in its script and it is written from left to right. The total number of Tifinagh letters after the two amendments reaches 59 characters, and are occupying 2D30-2D7F plage in Unicode.

The Amazigh language belongs to the Hamito-Semitic/“Afro-Asiatic” languages [9, 13], with rich templatic morphology [8]. In linguistic terms, the language is characterized by the proliferation of dialects due to historical, geographical and socio-linguistic factors. For instance, one may distinguish three major dialects in Morocco: Tarifit in the North, Tamazight in the center and Tashlhit in the southern parts of the country; it is a composite of dialects of which none has been considered the national standard.

Most Amazigh words may be conceived of as having consonantal roots. They can have one, two, three or four consonants, and may sometimes extend to five. Words are made out of these roots by following a pattern. For example the common noun ⴰⵎⵣⴷⴰⵖ (inhabitant) “amzdaG” is built up from the root ⵣⴷⵖ (live) “zdG” by following a definite pattern (Figure 1.) ⴰⵎ12ⴰ3 “am12a3”; where the number 1 is replaced by the first consonant of the root, number 2 is replaced by the second consonant of the root and number 3 is replaced by the 3rd consonant of the root.

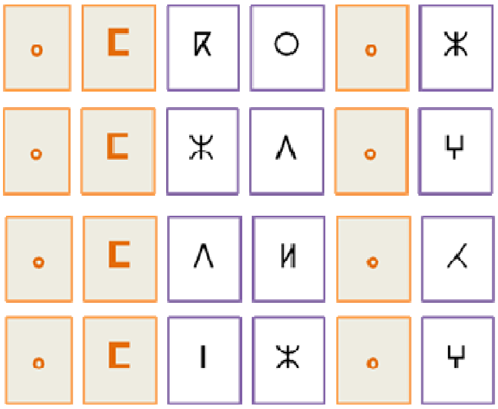

Fig. 1. Making words following a pattern.

Also, verb derivation is very rich. A reciprocal verbal derivation sample is generate d by adding ⵎⵎ “mm”. For instance, for the verb ⵕⵥ “RZ” (break), its reciprocal derivation is ⵎⵎⵕⵥ “mmRZ” (mutual break).

2.2. Challenges in POS-tagging and related works

One of the challenges of POS-tagging is ambiguity; the same surface form might be tagged with a different POS tag depending on how it has been used in the sentence. Like most of the languages that have only recently started being investigated for the NLP tasks, Amazigh lacks of annotated corpora and processing tools and resources. Very few linguistic resources and tools have been developed up to now for this language. In this part of the paper, we introduce existing works related to the introduction of this language into information and communication technology. Existing works in NLP are a spelling corrector based on the algorithm of Hunspell [17], a concordance [7], a light stemmer [4], some tools and resources achieved by LDC/ELDA under a relationship of partnership with IRCAM as an encoding converter [2], a word and sentence segmenter, and a named-entity tagger and tagged text with named entities, and a morphological analyzer/generator for Amazigh nouns [34].

We think that the development of a POS-tagger system is the first step needed for automatic text processing. In line with this, in the preliminary experiments on POS-tagging for Amazigh [30, 31], we have trained two sequence classification models using Support Vector machines (SVMs) and Conditional Random Fields (CRFs). SVMs out-performed CRFs on the fold level (91.66% vs. 91.35%) and CRFs outperformed SVMs on the 10 folds average level (88.66% vs. 88.27%), based on a tag set containing 15 elements (verb, noun, ad-verb...etc.), in addition to S_P and N_P referring respectively to prepositions and kinship nouns when followed by personal pronouns. Using a tokenization step, results for a tag set of 13 tags showed as well that the performance of SVMs and CRFs are very comparable. Across the board, SVMs outperformed CRFs on the fold level (92.58% vs. 92.14%) and CRFs outperformed SVMs on the 10 folds average level (89.48% vs. 89.29%).

In the next section we will present AMTS, AMazigh Tag Set a larger tag set of 28 tags and POS-tagging results using SVMs and CRFs.

3. Amazigh POS-tagging with CRFs and SVMs

In this section we introduce the manually annotated corpus, the employed machine learning techniques, baselines, and the used feature set. Then we present the POS-tagging results obtained on the basis of the manually labeled data and the used unlabeled data in self training experiments.

3.1. Manual Annotation and AMTS tag set

Manual annotation was achieved following a 5-step process:

1. Raw texts: to constitute an annotated corpus, we have chosen a list of corpora extracted from a variety of sources such as some novels, as well as some texts from IRCAM's web site.
2. Transliteration: Amazigh corpora produced up to now are written on the basis of different writing systems, most of them use Tifinagh-IRCAM (Tifinagh-IRCAM makes use of Tifinagh glyphs but Latin characters) and Tifinagh Unicode. It is important to say that the texts written in Tifinagh Unicode are increasingly used. Even though, we have decided to use a specific writing system based on ASCII characters for technical reasons [29].
3. Manual annotation: this corpus is annotated morphologically using AncoraPipe annotation tool[1]. Four annotators were involved in this task and annotation speed was between 80 and 120 tokens/hour. Our Inter Annotator Agreement is around 94.98%.
4. Revision: we have used XSLT to generate output files that allow validation of the annotated corpora. Annotation speed was between 80 and 120 tokens/hour. Randomly chosen texts were revised by three other linguists, their common remarks were generalized to the whole corpora in the second validation by a different annotator.
5. Annotated texts: output documents have an XML format, allowing representing tree structures. As XML is a wide spread standard, there are many tools available for its analysis, transformation and management.

Since defining the adequate tag set is a core task in building an automatic POS-tagger, we have decided to use a state-of-art machine learning technique, namely CRFs, on the basis of a fine-grained tag set called AMTS. It contains 28 tags and it is presented in Table 1. The tag set is richer than the one used by (Outahajala et al., 2011). For instance we have split the N corresponding tag to the nouns into NN for common nouns, NNK for kinship nouns and NNP for proper nouns. Also, S_P and N_P referring respectively to prepositions and kinship nouns when followed by personal pronouns where split. Thus, a tokenization system was achieved to assume tokenization preprocessing task before POS-tagging [32]. PROT represents all particle kinds apart from orientation, vocative, negative, predicate and preverbal particles. ROT label stands for attributes like currency, and mathematical marks.

[1] http://clic.ub.edu/mbertran/tbfeditor/instalar_en.html

Table 1. AMTS tag set

N°	POS	Designation
1	NN	Common noun
2	NNK	Kinship noun
3	NNP	Proper noun
4	VB	Verb, base form
5	VBP	Verb, participle
6	ADJ	Adjective
7	ADV	Adverb
8	C	Conjunction
9	DT	Determiner
10	FOC	Focalizer
11	IN	Interjection
23	NEG	Particle, negative
12	VOC	Vocative
13	PRED	Particle, predicate
14	PROR	Particle, orientation
15	PRPR	Particle, preverbal
16	PROT	Particle, other
17	PDEM	Demonstrative pronoun
18	PP	Personal pronoun
19	PPOS	Possessive pronoun
20	INT	Interrogative
21	REL	Relative
22	S	Preposition
24	FW	Foreign word
25	NUM	Numeral
26	DATE	Date
27	ROT	Residual, other
28	PUNC	Punctuation

3.2. Our Machine learners

In this subsection we describe SVMs and CRFs, being proved to give good results for sequence classification [12, 20, 22, 23].

SVMs were first introduced by Vapnik [41]; they are known for their good generalization performance and have been used for different recognition problems. For instance, in NLP SVMs are applied to text categorization [22], name entity recognition [5], base phrase chunking [15] and others. Many POS taggers based on SVMs have been achieved for many languages, such as: Arabic [14, 15], Bengali [16] etc .They are reported to have achieved a high accuracy without over fitting even with a large number of features. SVMs are also known for copying well with sparse and noisy data.

With respect to the task of POS tagging in Amazigh, the training process has been carried out by YamCha[2], an SVM based toolkit. For classification, we have used the TinySVM-0.094[3] classifier, a publicly available toolkit for the problem of pattern recognition.

CRFs are undirected graph models. They are a generalization of Maximum Entropy Markov Models (MEMMs) and are oriented toward segmenting and labeling data [23]. Conditional model specify the probabilities of possible label sequences given an observation sequence. In addition to having the advantages of MEMMs, CRFs also overcome the label bias problem. We can think of CRFs as a finite state model with unnormalized transition probabilities. CRFs are applied to many NLP fields such as name entity recognition [25], shallow parsing [37], information extraction from tables [33]. CRFs were used for POS-tagging in many languages, such as Amharic [1] and Tamil [24].

We have used CRF++[4], an open source implementation of Conditional Random Fields for segmenting and labeling data, using the same data set as the one used with YamCha.

3.3. Experiments settings and baselines

Throughout this paper, all the described statistical models will use the same feature-set. The choice of the below described features has been reached through empirical results. The employed features are the following:

1- The current token;

2- Lexical features: these consist of the last and first ’i’ character n-grams, with ’i’ spanning from 1 to 4;

3- Lexical context: the surrounding words in a window of -/+2; and

4- Tag context: this consists of the predicted tags of the two previous words.

Regarding baselines, frequency based baseline (Freq-Base.) is a non learning algorithm. It predicts the tag for a certain token is the most frequent POS tag that has been associated with it in the training data. Thus, this baseline completely ignores the sur-

[2] http://chasen.org/~taku/software/yamcha/

[3] http://chasen.org/~taku/software/TinySVM/

[4] http://crfpp.sourceforge.net/

rounding context and resolves the ambiguous cases using only frequency. Such baseline has been already used in competition tasks such as CoNLL for named entity recognition[5].

Best-Base: To study the best case scenario using CRFs and SVMs, we start with an initial model M_{init} trained on an initial training set and aggregate data from the remaining 30% of the manually annotated data in blocks of 2k (see Figure 4). This will be helpful to provide a contrast to the models that will be generated using automatically annotated data.

3.4. POS-tagging 10-fold cross-validation results

In this first experiment set, we have run 10-fold cross validation. We use the whole manually annotated data. The obtained best F-measure is in the fifth fold. Table 2 presents 10-fold cross validation results for a total of 1,438 sentences. Based on AMTS, CRFs outperformed SVMs on the fold level (87.95% vs. 87.11%) and on the 10 folds average level (91.18% vs. 90.75%).

The test set of fold 5 is the one used in the rest of the experiments of this paper.

The obtained results are very promising considering that we have used a corpus of only 20k tokens and compared to previous results based on 13 tags. We have more than doubled the tag set size in return we lost 1.34% in precision.

In comparison with the old tag set, most classes' performance increased. We obtained 96.24% vs. 94% for prepositions class, 65.38% vs. 60.7% for adverbs, 87.02% vs. 84.6% for determinants, 75% vs. 60% for focalizers, 100% vs. 45% for interjections. The adjective and conjunctions classes precision decreased in the new tag set. Regarding classes that we have split into several subclasses such as N corresponding to nouns, that we split into NN for common nouns, NNK for kinship nouns and NNP for proper nouns, NN precision is 95.15% vs. 94.60% for N. However obtained accuracy for proper nouns is just 54.16%, due essentially to insufficient examples in the training set. Concerning verbs base form precision is 94.22%.

Table 2. 10-fold cross validation results using tokenization

Fold#	BASELINE	SVMs	CRFs
0	79.70	85.12	86.02
1	77.36	83.25	84.28
2	84.03	**90.75**	89.48
3	81.00	87.89	88.2
4	80.11	88.36	89.35
5	81.47	90.24	**91.18**
6	77.29	83.18	84.27
7	76.95	83.84	85.32
8	84.22	89.33	90.31
9	86.45	89.20	91.12
AVG	**80.85**	**87.11**	**87.95**

Analyzing training and test sets, some classes are difficult to distinguish in Amazigh such as adjectives, nouns and participles. Also, we observed that unseen words in the test set are significant due to the small size of the data set.

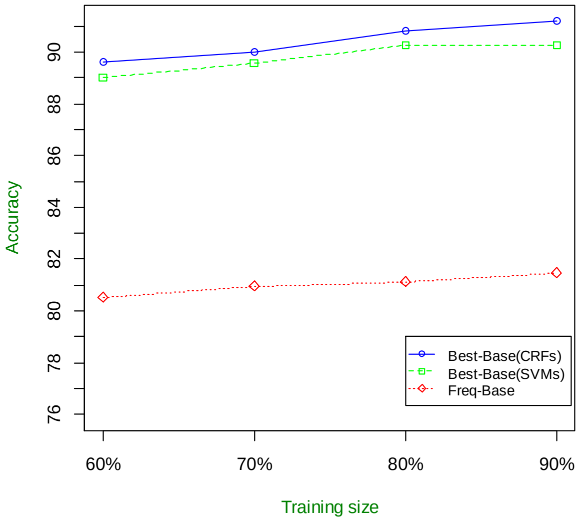

Fig. 2. Performance of the tagger when trained with manually annotated data

Figure 2 shows the obtained results for 'Best-Base' and 'Freq-Base' both using manually annotated data. The learning curve is increasing along the training corpus size. The 'Freq-Base' is at least 8 points below CRFs and SVMs across the curve. We started with an initial model (M_{init}) and each time we added 10% from annotated data. The precision difference between the model trained on the basis of 60% and the model trained on the basis of 90% of hand labeled data for CRFs and SVMs is 1.55% and 1.23% respectively.

[5] http://www.cnts.ua.ac.be/conll2002/

Creating labeled data is a hard task. The only way to obtain labeled data, especially for languages with scarce resources, is following manual annotation process. However, obtaining unlabeled data, although needing most time preprocessing, is less difficult. In this way, we collected all the available corpora in Amazigh and we experimented their use with the manually labeled data presented in subsections 3.5 and 3.6 below.

3.5. Amazigh unlabeled corpora

The used corpora, in these experiments, were collected from some published IRCAM novels, 113,474 words from the collected corpora by LDC [10], plus available sentences translated into Amazigh. The collected corpora needed preprocessing of many kinds:

- For texts written in tifinaghe-IRCAM (Tifinaghe-IRCAM makes use of Tifinagh glyphs but Latin characters), to correct some elements such as the character ^ which exists in some texts due to input error in entering emphatic letters;
- Transliteration to the chosen writing system based on ASCII characters, of texts written in tifinaghe-IRCAM and official tifinagh transcription of Amazigh language;
- Correction of misplaced tifinagh letter yey "ⴻ ", which is one of the frequent errors in the collected texts. A script was used in order to help fixing it;
- Orthographic writing revision by linguists following IRCAM rules; indeed many orthographic writing rules exist for this language. One frequent mistake is space misplacing. A Perl script was used in order to correct space misplacing. This program uses a lexicon of more than 41,000 distinct Amazigh words. Many texts and lexicons were used to constitute this lexicon such as [3, 18, 38]. The total number of words of the revised corpus is 218,073 words ;
- Tokenization, using the Amazigh tokenizer described in [32]. We have obtained a total of 225,901 tokens from the raw collected corpus.

In order to compute the quality and the reading complexity of this Amazigh collected and tokenized corpus, the three measures defined in [26] were considered. The complexity is 8.37, variety is 1884,35, and correctness of the corpus tokens frequency distribution, based on "the principal of least-ffort" [44] is presented in figure 3.

Graphical representation in log-log scale of ideal Zipf's law is a straight line with negative slope. Amazigh tokens distribution is around Zipf's law ideal curve (Figure 3). Hence, we showed empirically that this Amazigh corpus respects the Zipf's law principle.

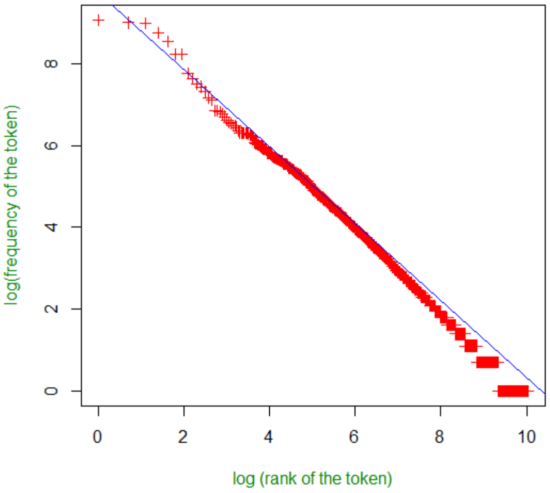

Fig.3. Tokens frequency distribution and the Zipf's ideal curve

The preprocessed and tokenized collected data is denoted by U in the following experiments.

3.6. Data selection for automatic training

In order to study the usefulness of system word confidence to select data, we have conducted experiments using M_{init} and the unlabeled data presented above.

Unlabeled data were annotated automatically and we kept only corresponding sentences to 1295 sentences (the same size as the manually labeling data training set). The data selection criterion is based on CRFs sentence confidence measure. As it is shown on Figure 4, the selected corpus was divided into 9 parts: U_1, U_2, U_3, U_4, U_5, U_6, U_7, U_8, and U_9. Where each part of them has 144 sentences for each U_i where *i* varies from 1 to 9 (the equivalent to 10% of the total number of manually annotated sentences).

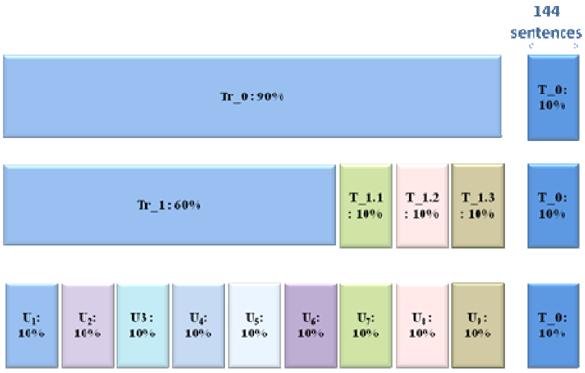

Fig. 4. Data splits for preliminary self training experiments

In this experiment, we started by training our model from an intial model M_{init} and at each time we added U_i, 10% from automatically annotated data using M_{init}, on the basis of the best system confidence measure. The achieved error reduction is 1.37%. The used measure to choose the best sentences to constitute automatically unlabeled data is system confidence for sentences. Each point of Figure 5(a) represents the obtained model accuracy with training based on 60% of manually annotated corpus (Tr_1) and its accumulation with U_i, where *i* varies from 1 to 3.

In order to study the effect of ignoring confidence and to see whether confidence is important or not, we conducted experiments on training our model from M_{init} and at each time we add 10% from automatically annotated data on the basis of data selected randomly from unlabeled data set. Self training our model on the basis of data selected randomly from unlabeled data set is less than self training on the basis of data selected using confidence measure. We have also conducted experiments on training our tagger on automatically annotated corpora. In this experiment, and instead of using M_{init}, we used U_1, U_2, ..., U_6 to generate the initial model M_{init}, afterwards, at each time we added 10% from automatically annotated data. The achieved accuracy improvement between 60% and 90% of the trained data is 5.9%. Figure 5(b) shows the improvement evolution and summarizes results.

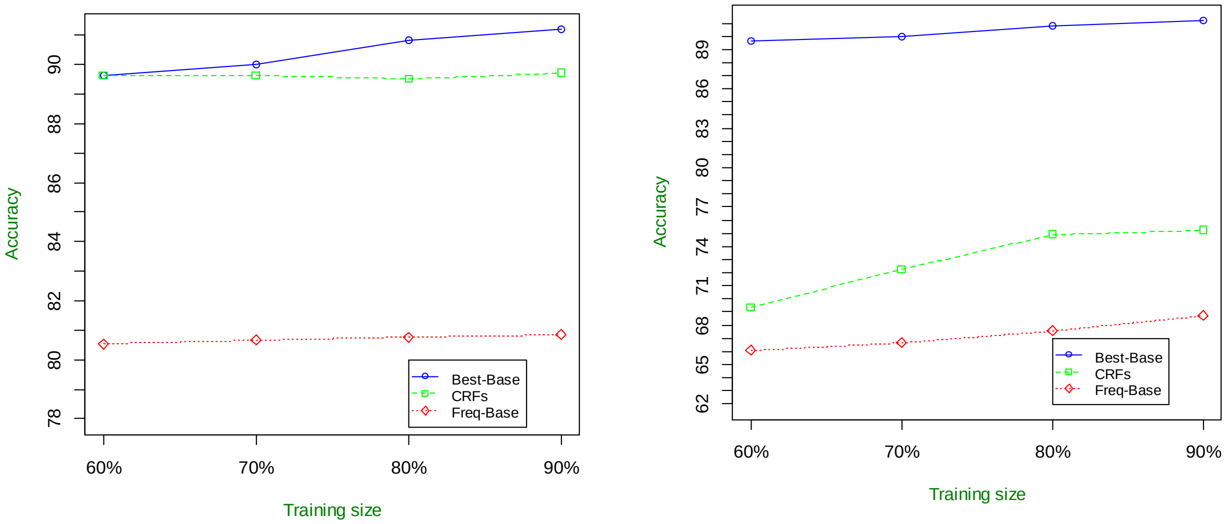

Fig. 5. Training on data selected using (a) system sentence confidence measure and (b) training on automatically annotated data

This experiment shows the importance of the learned information in spite of the fact that M_{init} is obtained by the use of U_1, U_2, ..., U_6 and that we add at each iteration tagged data obtained automatically from U.

In the next section we will present semi-supervised learning experiments to improve our POS tagger accuracy. In these experiments, we will use the CRFs tool used above because it provides probabilities for each tag and a conditional probably for the whole sentence.

4. Experiments

In this section we will study the usefulness confidence measure and its effectiveness when using unlabeled data for improving our POS tagger accuracy.

4.1. Confidence measure effectiveness

A selection criterion that we want to explore in this research work is confidence measure. We want, however, to start with an assessment of the validity of this approach. To do so, we have opted to estimate the correlation between the ’confidence’ and the ’probability of correctness’. That is to assess the odds of the automatic tag assigned to a token being correct when the system ’word confidence’ is high. We believe this estimate is important because as the observed correlation tends to 0 the probability of the selected data point to enhance the performance tends to 0.5, i.e. random. From a noise filtering perspective, we can say that in the case of absence of correlation between the two terms in question it is not possible to filter noise on the base of system confidence. In order to obtain the required information we have automatically tagged 10% from the training set using M_{int} a trained model based on 60% of manually annotated corpus. The obtained tags served as a data set to compute the correlation. In Figure 6 we show a plot of the data point together with a line obtained through linear regression. The data set shows that there is a correlation of 0.78 between these two terms. From the figure, it can be appreciated that there is a clear positive correlation with few outliers.

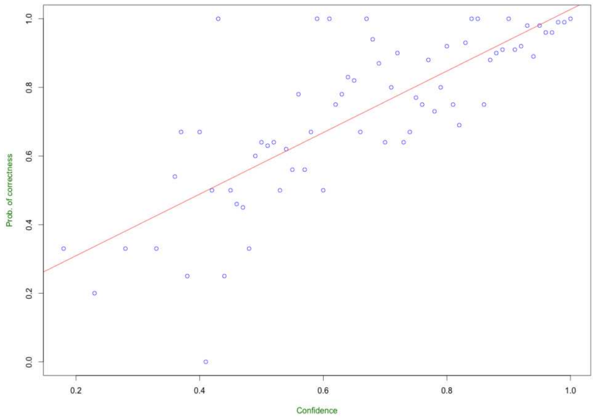

Fig. 6. Scatter plot of system confidence and probability of confidence

Computing a similar statistic for the ‘sentence confidence’ has been stymied by the skeweness of the distribution of correctly vs. incorrectly tagged sentences, i.e. the number of correctly tagged sentences (where all the words are correctly tagged) is very smaller relatively to the number of the incorrectly tagged ones (where at least one word is incorrectly tagged). However, we find the results encouraging running further experiments using both word and sentence confidence as self-training criteria. We present the design and results of these experiments in the upcoming subsections.

4.2. Self training algorithm and confidence as criterion to select data

Semi supervised learning techniques which consist of using labeled as well as unlabeled data can be very useful. The goal is to map inputs to labels by taking advantage of available unlabeled data, in order to build a more accurate classifier.

Two main algorithms are used in semi-supervised learning:

- Self training begins with a classifier trained with some labeled data, and at each iteration the labeled data are increased with the new labeled data. It has been applied to many NLP fields. For instance, Yarowsky [43] uses it for word sense disambiguation;
- Co-training proposed by Blum and Mitchell [6], starts with some labeled data, train one classifier using the first view of labeled data, and the second classifier using a second view of labeled data. The classifiers are used to label some new data. The

most confident labeled data are kept and added to the training set. The process is iterated until a stopping criterion is achieved. Since unsupervised approach have proven to give interesting results, when using automatically annotated data, we have opted to implement self training algorithm to generate more accurate POS-tagger using many variants of CRF++ confidence measure and an adapted form of self training algorithm.

4.2.1. *Self training algorithm*

The algorithm consists in training a first classifier M_{init} with a small amount of labeled data and expanding the labeled data with the addition of the classified-labeled data, and re-training new classifiers. The process is iterated until a stopping criterion is met. Its basic form is presented in Algorithm 1. The select function takes a confidence-labeled data set and selects best sentences to be trained.

Algorithm 1. Informativeness(L_0, U)

```
$L_0$ is labeled data, $T_0$ test file, U is Unlabeled data
$M_{init}$<-- train($L_0$)
For each $OOV_i$ in OOV
    Automatically_tag(U, Model)
    Tf = Identify most frequent tag to $OOV_i$
    $ST_i$= Assign Tf to select(U, $OOV_i$))
    U<-- U- $ST_i$
    L <--$L_0$ + $ST_i$
    $Model_i$<--train (L)
    Test($T_0$)
End For each
Return Model
Function Select (U, Word)
    For each sentence of U
        If (sentence contains Word) then
            selected_sentences = selected_sentences + sentence
        End If
    End For each
Return selected_sentences
```

We have implemented the self training algorithm using 60% of hand labeled data constituting L_0. The accuracy of the trained model M_{init} based on L_0 is 89.63%, and collected unlabeled data is *U* presented in Subsection 5.1.

4.2.2. *Self training using confidence to select data*

In this experiment we have used 10% of the manually annotated data for the test. Taking as confidence measure the mean of sentence words confidences given by the model, the best obtained performance is 89.86% achieved after 155 iterations. The error reduction in this experiment is 2.24%. Analyzing selected sentences, we have observed that they are small. In line with this, we have combined this measure with a weight of sentences length.

At each iteration of the experiment we add the best sentence based on a confidence with word system probability and the weight of the sentence following the formula:

$$Conf\_M = \frac{\frac{Words\_Conf}{Sentence\_lenght} + \alpha\left(\frac{Sentence\_lenght}{max\_sentence\_lenght}\right)}{(1+\alpha)Sentence\_lenght}$$

Where *Conf_M* is the computed confidence measure, *Words_Conf is* the total of the given sentence words confidences, *Sentence_length* represents the tokens number of the sentence and *max_sentences_lenght* is the number of tokens of the longest unlabeled data sentences.

By varying *α*, we obtained a slightly better accuracy of 89.89% and an error reduction of 2.53% after 11 iterations with a value of α equal to 3. However, the accuracy decreases slowly after some little iteration.

Using system confidence for sentences as a criterion to choose the best sentences to add to labeled data, our model reaches the accuracy of 89.96% after 840 iterations which is the maximum of the self training curve when we select at each iteration 1 best sentence. With this experiment we obtain an error reduction of 3.20%. The performance goes down after 1200 iteration.

Since that OOV are an important source of errors, in the next subsection we will study the informativeness impact on POS-tagging.

4.3. *Informativeness*

In order to study informativeness impact on our system, we studied the performance with respect to OOVs rate. Table 3 summarizes the OOV and baselines results; as the performance goes up it becomes harder to improve. However, the difference in improvement does not decrease; instead it slightly fluctuates.

Table 3. OOV rate with respect to performance

Training file size	**OOV rate**	**Freq-Base accuracy**	**Best-Base accuracy**
60%	15%	80.53	89.63
70%	13%	80.95	90.00
80%	11%	81.14	90.81
90%	10%	81.47	91.18

For instance, improvement going from 70 to 80% (0.81) is greater than the improvement from 60 to 70% (0.66) when we train our models on data manually annotated data. The main reason of this improvement is OOV rate decreasing. For instance, the OOV for 80% is 11% besides 15% for 60%. Also, analyzing the output files of our tagger, we observed that returned tags by the system are sometimes correct. In this way, we have conducted experiments on informativeness by looking for unseen instances with frequency higher than some threshold and for each of these instances we identified the most frequent tag and assigned it to all the retrieved sentences containing the given OOV. Afterwards adding the sentences to the training data and re-training (Algorithm 2) did not give good results. It has not an important impact on performance; in fact the maximum obtained error reduction is 1.37%.

```
Algorithm 2. selfTrain($L_0$, U)
$L_0$ is labeled data, U is Unlabeled data
$M_{init}$ <-- train($L_0$)
Loop until stopping criterion is met
    L <--$L_0$ + select(U, Model)
    Model<--train (L)
End loop
Return Model
Function Select(U, Model)
    selected_sentences= best sentences based on a confidence measure
Return selected_sentences
```

In Algorithm 2, identifying the most frequent tag consists in looking for the most frequent tag of the given OOV among sentences; then it is assigned to all the sentences containing the given OOV word.

The obtained results are more interesting when we select data based on the combination of confidence measure and unseen tokens frequencies feature; that is why we decided to use these two features with semi-supervised learning techniques.

In the next section we will combine the use of confidence measure and informativeness.

4.4. Combining Confidence and Informativeness

Since OOVs are an important source of errors, we have implemented a new algorithm (Algorithm 3.), which exploits, at each iteration, frequencies of given tags to OOVs and confidence measure.

```
Algorithm 3. InformativenessConfidence($L_0$, U)
$L_0$ is labeled data, $T_0$ test file, U is Unlabeled data
$M_{init}$<-- train($L_0$)
For each $OOV_i$ in OOV sorted by confidence
    $ST_i$<- select(U, $OOV_i$)
    $BS_i$<- select_BS($ST_i$, $Model_i$)
    Tf = Identify most frequent tag of $OOV_i$
    $BS_i$= Assign Tf, to select(U, $OOV_i$))
    U<-- U- $BS_i$
    L <--$L_0$ + $BS_i$
    $Model_i$ <--train (L)
    Test($T_0$)
End For each
Return Model
```

In this algorithm, ST_i are selected sentences from U for the i^{th} OOV sorted by confidence, BS_i represents best selected sentences selected from ST_i using confidence measure of $Model_i$.

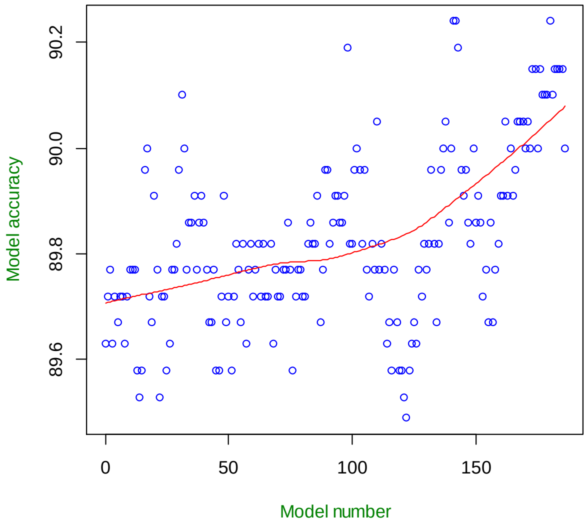

Fig. 7. Results of self training using unlabeled data OOV frequencies

Figure 7 shows the important impact of using unlabeled data OOV frequencies and CRFs confidence measure. Although the fact that we have only 41% of the OOVs in the unlabeled data, we obtained an error reduction of 5.90%. As shown in figure 7, model accuracy is improving each time we added selected data containing OOVs with high confidence. However, analyzing the learning curve evolution and output files, it is hard to increase the performance when it is already high. The obtained Error reduction on this small Amazigh corpus is slightly better than the one obtained in related works on semi-supervised POS tagging, e.g. [39, 40], where obtained error reduction is between 4 and 5% on Wall Street Journal.

OOVs is an import source of errors that's why in the near future we will investigate further the use of lexicons together with supervised learning techniques to improve Amazigh POS-tagging results.

5. Conclusions

Very few linguistic resources have been developed so far for Amazigh and we believe that the development of a POS-tagger system is the first step needed for automatic text processing. In line with this, we presented AMTS tag set. Using CRFs we obtained a performance of 91.18% in accuracy; these results are very promising considering that we have used a corpus of only 20k tokens. In this way, since creating labeled data is a hard task and the fact that obtaining unlabeled data is less difficult, although needing a lot of time for its preprocessing especially for languages with scarce resources, we have gathered a set of unlabeled data of Amazigh language that we have preprocessed and tokenized. We have obtained a total of 225,901 tokens. The collected unlabeled corpus was used with self training in order to have a more accurate POS-tagger. Analyzing the learning curve evolution, it is possible to notice that when the performance is already high, it is hard to increase it especially if we add automatically tagged data. The automatically tagged data, even if it is filtered by looking only at the data with high confidence, do not help increasing the performance such as when hand labeled data is used. In this way, using an adapted form of self training algorithm heavily relying on CRFs confidence measure and OOVs frequencies of the unlabeled corpus, we obtained an error reduction of 5.9%.

Acknowledgments

The first author wants to grant CODESRIA. The work of the third author was carried out in the framework of the VLC/CAMPUS Microcluster on Multimodal Interaction in Intelligent Systems and the European Commission WIQ-EI IRSES (no. 269180) and DIANA-APPLICATIONS-Finding Hidden Knowledge in Texts: Applications(TIN2012-38603-C02-01) research projects.

References

[1] S. F. Adafre, Part of Speech Tagging for Amharic using Conditional Random Fields. In Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, 2005, pp. 47–54.

[2] Y. Ait Ouguengay and A. Bouhjar, For Standardized Amazigh Linguistic Resources, in: Proceedings of the 7th International Conference on Language Resources and Evaluation. LREC'10, Malta, May 17-23, 2010, pp. 2699–2701.

[3] F. Agnaou, A. Bouzandag, M. El Baghdadi, E. El Gholb, A. Khalafi, K.. Ouqua, M. Sghir. Lexique Scolaire. Publications de l'IRCAM. 2011.

[4] F. Ataa Allah and S. Boulaknadel, Pseudo-racinisation de la langue amazighe, in: Proceedings of TALN 2010, Montréal, 2010, pp. 19–23.

[5] Y. Benajiba, M. Diab and P. Rosso, Arabic Named Entity Recognition: A Feature-Driven Study, IEEE Transactions on Audio, Speech and Language Processing, vol. 15, num. 5. Special Issue on Processing Morphologically Rich Languages, 2010, pp. 926–934.

[6] A. Blum and T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT'98), 1998, pp. 92–100.

[7] S. Boulaknadel, Amazigh ConCorde: an appropriate concordance for Amazigh, in: Proceedings of 1er Symposium International sur le Traitement Automatique de la Culture AMazighe (SITACAM). Agadir, Morocco.

[8] M. Chafiq. أربعة وأربعون درسا في الأمازيغية [Forty four lessons in Amazigh]. éd. Arabo-africaines. 1991.

[9] S. Chaker, Textes en linguistique berbère - introduction au domaine berbère, éds du CNRS, 1984, pp. 232–242.

[10] C. Cieri and M. Liberman, 15 Years of Language Resource Creation and Sharing: A Progress Report on LDC Activities, in: Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC'08, Marrakech. 2008.

[11] S. Clark, J.R. Curran and M. Osborne, Bootstrapping POS Taggers Using Unlabelled Data, in: Proceedings of CoNLL'03. 2003.

[12] M. Constant, M., I. Tellier, D. DUCHIER, D., Y. Dupont, A. SIGOGNE, and S. BILLOT, Intégrerdes connaissances linguistiques dans un CRF : application à l'apprentissage d'un segmenteur-étiqueteur du français, 2011, in Proceedings of TALN'11.

[13] D. Cohen, Chamito-sémitiques (langues), in: Encyclopaedia Universalis. 2007.

[14] M. Diab, K. Hacioglu, and D. Jurafsky, Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. Proceedings of Human Language Technology-North American

Association for Computational Linguistics (HLT-NAACL), 2004.

[15] M. Diab, K. Hacioglu, and D. Jurafsky, Arabic Computational Morphology: Knowledge-based and Empirical Methods, 2007, chapter 9. Springer.

[16] A. Ekbal, and S. Bandyopadhyay, Part of Speech Tagging in Bengali Using Support Vector Machine. In Information Technology, ICIT '08, 2008, pp. 106-111.

[17] Y. Es Saady, Y. Ait Ouguengay, A. Rachidi, M. Elyassan and D. Mammass, Adaptation d'un correcteur orthographique existant à la langue Amazighe: cas du correcteur Hunspell, in: Proceedings of 1er symposium international sur le traitement automatique de la culture amazighe(SITACAM), Agadir, Morocco. 2009.

[18] L. El Gholb, La Conjugaison du Verbe en Amazighe: Elément Pour Une Organisation, Editions Universitéaires Européennes, Sarrebruck, Allemagne. 2011.

[19] D. Jurafsky and J.H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, And Speech Recognition, 2nd Ed. New Jersey: Prentice Hall. 2009.

[20] J. Giménez, L. Márquez, SVMTool: A general POS tagger generator based on support vector machines. In Proceedings of the 4th International Conference on Language Resources and Evaluation, Lisbon, Portugal, 26–28 May 2004, pp. 43–46.

[21] D. Klein and C. Manning, A Generative Constituent Context Model For Improved Grammar Induction. In Proceedings of the 40th Annual Meeting of the ACL. 2002.

[22] T. Kudo and Y. Yuji Matsumoto, Use of Support Vector Learning for Chunk Identification, in: Proceedings of CoNLL-2000 and LLL-2000. 2000.

[23] J. Lafferty, A. McCallum and F. Pereira, Conditional Random Fields: Probabilistic Models for Segmenting And Labeling Sequence Data, in: Proceedings of ICML-01, 2001, pp. 282–289.

[24] S. Lakshmana Pandian and T.V. Geetha, CRF Models For Tamil Part of Speech Tagging and Chunking, in: Proceeding of ICCPOL'09. Springer-Verlag Berlin, Heidelberg. 2009.

[25] W. Li and A. McCallum, Rapid Development of Hindi Named Entity Recognition Using Conditional Random Fields and Feature Induction, in: ACM Transactions on Computational Logic, 2003, pp. 290–294.

[26] P. Makagonov, and M. Alexandrov, Some Statistical Characteristics for Formal Evaluation of the Quality of Text books and Manuals. In Computing Research: Selected papers., 1999, pp 99--103.

[27] C. Manning and H. Schütze, Foundations of Statistical Natural Language Processing, the MIT Press. 1999.

[28] M. Marcus, B. Santorini and M. Marcinkiewicz, Building a large annotated corpus of English: the Penn treebank, Computational Linguistics, Vol. 19, 1993, pp 313–330.

[29] M. Outahajala, L. Zenkouar, P. Rosso and A. Martí, Tagging Amazigh with AncoraPipe, in: Proceedings of the Workshop on LR & HLT for Semitic Languages, 7th International Confe-rence on Language Resources and Evaluation, LREC'10, Malta, May 17-23, 2010, pp. 52–56.

[30] M. Outahajala,Y. Benajiba, P. Rosso and L. Zenkouar, POS tagging in Amazigh using Support Vector Machines and Conditional Random Fields, in: Proceedings of 16th International Conference on Applications of Natural Language to Information Systems, NLDB 2011, LNCS(6716), Springer-Verlag, 2011, pp. 238–241.

[31] M. Outahajala, Y. Benajiba, P. Rosso and Zenkouar, L'étiquetage grammatical de l'amazighe en utilisant les propriétés n-grammes et un prétraitement de segmentation, e-TI-la revue électronique des technologies d'information, Numéro 6. 2012.

[32] M. Outahajala, L. Zenkouar, Y. Benajiba and P. Rosso, The Development of a Fine Grained Class Set for Amazigh POS Tagging, in: Proceedings of the 10th ACS/IEEE conference. AICCSA 2013. Fes, Morocco. 2013.

[33] D. Pinto, A. McCallum, X. Wei and W.B. Croft, Table Extraction using conditional random fields, in: Proceedings of the 26th annual international of SIGIR'03, New York, USA, 2003, pp. 235–242.

[34] H. Raiss and V. Cavalli-Sforza, ANMorph: Amazigh Nouns Morphological Analyzer, in: Proceedings of the 5th Int. Conference on Amazigh and ICT, NTIC-2012, Rabat, Morocco. 2012.

[35] E. Ringger, P. McClanahan, R. Haertel, G. Busby, M. Carmen, J. Carroll, K. Seppi, and D. Lonsdale, Active learning for part-of-speech tagging: Accelerating corpus annotation. in Proceedings of the Linguistic Annotation Workshop, 2007,pages 101–108.

[36] B. Settles, Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin-Madison. 2009.

[37] F. Sha and F. Pereira, Shallow Parsing with Conditional Random Fields, in: Proceedings of Human Language Technology. 2003.

[38] M. Sghir, Essai de Confection d'un Dictionnaire Monolingue Amazighe: Méthodologie et Application, Parler de la Vallée du Dadès (Sud-Est du Maroc). Thèse de doctorat, FLSH Saïs-Fès. 2014.

[39] A. Søgaard, Simple semi-supervised training of part-of-speech taggers, in: Proceedings of the ACL 2010 Conference Short Papers, 2010, 205–208.

[40] D. Spoustova, J.Hajic, J. Raab, and M. Spousta, Semi-supervised training for the averaged perceptron POS tagger, in EACL, 2009, Athens, Greece.

[41] V. N. Vapnik, The Nature of Statistical Learning Theory. Springer Verlag, New York, USA, 1995.

[42] N. Xue, F. Xia, F.D. Chiou and M. Palmer, The Penn Chinese Treebank: Phrase structure annotation of a large corpus, Natural Language Engineering, Vol. 10(4), 2004, pp. 1–30.

[43] D. Yarowsky, Unsupervised Word Sense Disambiguation Rivaling Supervised Methods, in: Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, 1995, pp. 189–196.

[44] G.K. Zipf. Human Behaviour And The Principle Of Least Effort. Addison-Wesley. 1949.