The Development of a Fine Grained Class Set for Amazigh POS Tagging

MOHAMED OUTAHAJALA, Laboratoire Electronique et Communication, Ecole Mohammadia d’Ingénieurs, Université Mohammed V Agdal, Morocco

LAHBIB ZENKOUAR, Laboratoire Electronique et Communication, Ecole Mohammadia d’Ingénieurs, Université Mohammed V Agdal, Morocco

YASSINE BENAJIBA, Thomson Reuters, New York, USA

PAOLO ROSSO, Natural Language Engineering Lab – EliRF, Universitat Politècnica de València, Spain

Abstract—Like most of the languages which have only recently started being investigated for the Natural Language Processing (NLP) tasks, Amazigh lacks annotated corpora and tools and still suffers from the scarcity of linguistic tools and resources. The main aim of this paper is to present a tokenizer tool and a new part-of-speech (POS) tagger based on a new Amazigh tag set (AMTS) composed of 28 tag. In line with our goal we have trained two sequence classification models using Support Vector Machines (SVMs) and Conditional Random Fields (CRFs) to build a tokenizer and a POS tagger for the Amazigh language. We have used the 10-fold technique to evaluate and validate our approach. We report that POS tagging results using SVMs and CRFs are very comparable. Across the board, CRFs outperformed SVMs on the fold level (91.18% vs. 90.75%) and CRFs outperformed SVMs on the 10 folds average level (87.95% vs. 87.11%). Regarding tokenization task, SVMs outperformed CRFs on the fold level (99.97% vs. 99.85%) and on the 10 folds average level (99.95% vs. 99.89%).

Keywords—Annotation process, Tokenization, POS tagging, supervised learning, SVMs, CRFs;

I. INTRODUCTION

Most of the newly investigated languages in NLP are resource-scarce. According to [25], as many as half of the estimated 6,000 languages spoken on earth are spoken only by adults who no longer teach them to the next generation. 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 Arabic1. 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 of the languages which have only recently started being investigated for the NLP, still suffers from the scarcity of language processing tools and resources.

The very first POS-taggers were mainly rule-based systems. Building such systems requires huge manual effort in order to handcraft the rules and to encode the linguistic knowledge that governs the order of their application. For instance, in 1970

1 See the 5th article of the Moroccan Constitution, http://www.sgg.gov.ma/constitution_2011_Fr.pdf
The Amazigh language belongs to the Hamito-Semitic/Afro-Asiatic languages [13][15], with rich templatic morphology [12]. In linguistic terms, the language is characterized by the proliferation of dialects due to historical, geographical and sociolinguistic 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.

Due to its complex morphology as well as to the use of the different dialects in its standardization (Tashlhit, Tarifit and Tamazight being the three more used ones), the Amazigh language presents interesting challenges for NLP researchers. Some of these characteristics are presented below.

1) **Amazigh language encoding**

The official graphic system for writing Amazigh is Tifinagh. It does not have capitalization in its script and it is written from left to right. IRCAM kept only pertinent phonemes for Amazigh language, so the number of the alphabetical phonetic entities is 33 (consisting of: 27 consonants, 2 semi-consonants and 4 vowels), however Unicode codes only 31 letters plus a modifier letter to form the two phonetic entities: \(\text{거든} (g^w) \) and \(\text{カー} (k^w)\). The whole range of Tifinagh letters is subdivided into four subsets [5][41]: the basic letters used by IRCAM, an extended set used also by IRCAM, other neo-Tifinagh letters in use and some attested modern Touareg letters [35]. The total number of Tifinagh letters after the two amendments reaches 59 characters, and are occupying 2D30-2D7F plage in Unicode².

2) **Amazigh language morphology**

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 \(\text{نامكرزي} \) (farmer) “amkraz” is built up from the root \(\text{كرزي} \) (cultivate) “krz” by following a definite pattern (Figure 1.) \(\text{ايامكراز} \) “aamkraz”; 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. Also, verb derivation is very rich [12].

Concerning spelling, the system used by IRCAM is based on a set of rules and principles applied to “words” along which the parsing of pronounced speech into written separated words is effected [3][4][8]. A grapheme, a written word, according to the spelling system is a succession of letters which can sometimes be one letter delimited by whitespace or punctuation.

Here we summarize these rules to make the reading of the rest of this paper easier:

- Nouns consist of a single word occurring between two blank spaces. To the noun are attached the morphological affixes of gender (masculine/ feminine), number (singular/plural) and state (free/construct) as shown in the following examples:
  - \(\text{امكرزي} \) (a dweller) (singular masculine), “amzdaG” in the defined latin transcription presented in (Outahajala et al., 2010), \(\text{امكرزي} \) (a dweller) (singular feminine) “tamzdaGt”;
  - \(\text{امكرزي} \) (dwellers) (plural masculine) “imzdaGn”. The construct state of the noun \(\text{امكرزي} \) (a dweller) is \(\text{امكرزي} \) “umzdaGn”

- Kinship names constitute a special class since they are necessarily determined by personal pronouns which form with them one word, for example: \(\text{امكرزي} \) (your father), “babak”.

- Quality names, called also adjectives, constitute a single word along with the morphological indicators of gender (masculine/ feminine), number (singular/plural), and state (free/construct);

- Verbs are single graphic words along with their inflectional (person, number, aspect) or derivational morphemes. For example:
  - \(\text{امكرزي} \) (you run (imperfective)”, “ttazzl”;

  - The verb is separated by a blank space from its predecessors and successors, i.e.: \(\text{امكرزي} \) (he took)(them)/ “yasi tn”. \(\text{امكرزي} \) (you run (imperfective)”, “ttazzl”;

- Pronouns are isolated from the words they refer to. Pronouns in Amazigh are demonstrative, exclamative, indefinite, interrogative, personal, possessive, or relative. An example of their use is:
  - \(\text{امكرزي} \) (that) “nna” in the phrase: \(\text{امكرزي} \) (the way) \(\text{nna} \) “nna tkkit”, is an example of a relative pronoun;

- An adverb consists of one word which occurs between two blank spaces. Adverbs are divided into adverbs of place, time, quantity, manner, and interrogative adverbs. An example of an adverb in Amazigh is:

  \[\text{امكرزي} \] (that) “nna” in the phrase: \(\text{امكرزي} \) (the way) \(\text{nna} \) “nna tkkit”, is an example of a relative pronoun;
We think that the development of a POS-tagger tool is the first step needed for automatic text processing. In line with this, we have dedicated the following Subsection to the presentation of a morphologically annotated corpus and some preliminary results in Amazigh NLP.

3) Challenges in POS tagging

One of 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. To give some examples of different categories, extracted from the annotated corpus presented in [32], we present the following examples:

- **Orientation particle** like (and) "imān d ubrid"; an orientation particle. For instance, in the sentences below, the orientation particle "d" might function as a preposition, a coordination conjunction, a predicate particle or an orientation particle. For instance, in the sentences below, the word "d" might be:

  - A coordination conjunction:  "d īlīāt(d) (Amazigh) īlīāt(technologies) īlīāt(new), "tamaziGt d tiknuljijin timaynutin";
  - A preposition:  "d īlīāt(he went) īlīāt(with) īlīāt(the road), "iman d ubrid";
  - A predicative particle: īlīāt(he is) īlīāt(a man), "d argaz"; or
  - An orientation particle:  "d īlīāt(bringing) īlīāt(to here) īlīāt(bowl) īlīāt(large), "asi d tiknit tamjahdit".

4) Scarcity of resources

Like most of the languages that have only recently started being investigated for the NLP tasks, Amazigh lacks 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 may be subdivided into:

- General computational resources, which can be subdivided into three subdomains:

  - Tifinagh promotion works: Tifinagh is used more and more since the creation of the fonts and the keyboards for its transcription [36][37]. Also a lot of supporting materials have been achieved for its learning and use such as educational cd-roms, multimedia and other teaching materials;
  - Dictionaries and corpora: few resources about electronic dictionaries and corpora exist such as a web dictionary [22], a valence dictionary [29], a tool about terminology data allowing the compilation, and the management of existing terminology [17], and a corpus morphologically annotated of about 20K tokens described in [31];
  - Optical character recognition: many works related to optical character recognition have been carried out about Amazigh language [2][19][20].

- NLP resources: few NLP tools have been developed for this language which are a spelling corrector based on the algorithm of Hunspell [18], a concordance [9], a light stemmer [6], some tools and resources achieved by LDC/ELDA under a relationship of partnership with IRCAM as an encoding converter, a word and sentence segmenter, and a named-entity tagger and tagged text with named entity, and a morphological analyzer/generator for Amazigh nouns [38].

We think that the development of a POS-tagger tool is the first step needed for automatic text processing. In line with this, we have dedicated the following Subsection to the presentation of a morphologically annotated corpus and some preliminary results in Amazigh POS tagging.

B. Preliminary results on Amazigh POS tagging

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 NLP tasks dealing with higher linguistic levels require POS tags, for instance: phrase chunking, word sense disambiguation, grammatical function assignment and named entity recognition [28]. Across languages, POS tagging was used for many tasks, for example: named entity recognition and parsing [1][28]. 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 [21].

In the first experiments on POS tagging for Amazigh [32], the authors have trained two sequence classification models using SVMs and CRFs. SVMs outperformed 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 13 elements (verb, noun, adverb...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 [34] showed as well that 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%). Analyzing the most frequent errors in the two confusion matrices given by SVMs and CRFs for the folds that gave the best scores, the authors showed that the POS-tagger based on CRFs has better results in tagging nouns, adjectives and verbs. Besides SVMs based POS-tagger achieves better results in tagging pronouns, determinants, adverbs, focalizers and particles.

C. Manual annotation and AMTS tag set

Based on the Amazigh language features presented above, Amazigh tag set may be viewed to contain 13 parts-of-speech with two common attributes to each one: word and lemma, whose values depend on the lexical item they accompany. The features of the Amazigh POS tag set with their attributes are presented in [31].

The numbers between brackets in Table I. represent the numbers of subcategories of the attribute or the subattribute. The number of the theoretical Amazigh tag set is more than 1900 tags. More details about the attributes values and the whole tag set in general may be found in [33].

Manual annotation was achieved following a process of many steps (Figure 2.):

1- Raw texts: to constitute an annotated corpus, we have chosen a list of corpora brought from a variety of sources such as some novels, 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 [31]. A transliteration tool¹ was built in order to handle transliteration to and from the chosen writing system and to correct some elements such as the character “^” which exists in some texts due to input error in enting some Tifinagh letters;
3- Manual annotation: this corpus is annotated morphologically using AncoraPipe² annotation tool³. AncoraPipe [7] allows different linguistic levels to be annotated. The interface is fully customizable to allow different tag sets defined by the user. In line with this, we have defined a specific tag set to annotate Amazigh corpora;
4- Revision: we have used XSLT to generate output files that allow validation of the annotated corpora. Annotation speed is 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

¹ http://www.outamed.com/downloads/TranAmazighEv1.0.rar
² http://clic.ub.edu/ancora/
³ http://clic.ub.edu/mbertran/tbfeditor/instalar_en.html
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 compare two machine learning techniques, namely SVMs and CRFs, on the basis of a fine-grained tag set called AMTS. It is a new tag set for the Amazigh language; it contains 28 tags. These tags are presented in Table I.

### Table I. AMTS tag set

<table>
<thead>
<tr>
<th>Nº</th>
<th>Old POS</th>
<th>New POS</th>
<th>Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N</td>
<td>NN</td>
<td>Common noun</td>
</tr>
<tr>
<td>2</td>
<td>NNK</td>
<td>Kinship noun</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>NNP</td>
<td>Proper noun</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>V</td>
<td>VB</td>
<td>Verb, base form</td>
</tr>
<tr>
<td>5</td>
<td>VBP</td>
<td>Verb, participle</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>A</td>
<td>ADJ</td>
<td>Adjective</td>
</tr>
<tr>
<td>7</td>
<td>ADV</td>
<td>Adverb</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>C</td>
<td>C</td>
<td>Conjunction</td>
</tr>
<tr>
<td>9</td>
<td>DT</td>
<td>DT</td>
<td>Determiner</td>
</tr>
<tr>
<td>10</td>
<td>FOC</td>
<td>FOC</td>
<td>Focalizer</td>
</tr>
<tr>
<td>11</td>
<td>I</td>
<td>IN</td>
<td>Interjection</td>
</tr>
<tr>
<td>12</td>
<td>PR</td>
<td>NEG</td>
<td>Particle, negative</td>
</tr>
<tr>
<td>13</td>
<td>PRED</td>
<td>Particle, predicate</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>PROR</td>
<td>Particle, orientation</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>PRPR</td>
<td>Particle, preverbal</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>PROT</td>
<td>Particle, other</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>PDEM</td>
<td>Demonstrative pronoun</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>PP</td>
<td>Personal pronoun</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>PPOS</td>
<td>Possessive pronoun</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>INT</td>
<td>Interrogative</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>REL</td>
<td>Relative</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>S</td>
<td>S</td>
<td>Preposition</td>
</tr>
<tr>
<td>23</td>
<td>R</td>
<td>FW</td>
<td>Foreign word</td>
</tr>
<tr>
<td>24</td>
<td>NUM</td>
<td>Numerical</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>DATE</td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>ROT</td>
<td>Residual, other</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>F</td>
<td>PUNC</td>
<td>Punctuation</td>
</tr>
</tbody>
</table>

Table I shows how we have enriched the tag set used in [33]. For instance we have split the N corresponding to the nouns into NN for common nouns, NNK for kinship nouns and NNP for proper nouns.

PROT represents all particle kinds apart from orientation, vocative, negative, predicate and preverbal particles. ROT label stands for attributes like currency, and mathematical marks.

### III. AMAZIGH TOKENIZATION

Amazigh tokenization consists of breaking a stream of text into phrases, words and symbols, based on spaces and punctuation; afterwards we break each word into its different morphemes, i.e. tokens. This step is necessary to obtain the required performance for further processing such as POS tagging [34] and parsing. In order to build our tokenizer, we have trained CRFs. The training process has been carried out by CRF++\(^6\), an open source implementation of CRFs. To our knowledge, this is the first tokenizer for this language.

We have trained this sequence labeling tools on the same corpus, yet using the following 5 classes: {B-WORD, I-WORD, B-SUFF, I-SUFF, O}; similarly Diab used 10 classes to achieve a tokenizer for Arabic [16]. An extract of training corpus used in the experiments is presented in Figure 3. Our feature set consisted of the surrounding characters and their tags in a context window of -/+4 without undertaking any change for the other parameters. The size of the context window has been chosen on the basis of empirical experiments.

### Table II. Amazigh Tokenizer 10-fold cross validation results

<table>
<thead>
<tr>
<th>Fold#</th>
<th>SVMs</th>
<th>CRFs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>99.76</td>
<td>99.62</td>
</tr>
<tr>
<td>1</td>
<td>99.6</td>
<td>99.5</td>
</tr>
<tr>
<td>2</td>
<td>99.78</td>
<td>99.72</td>
</tr>
<tr>
<td>3</td>
<td>99.66</td>
<td>99.62</td>
</tr>
<tr>
<td>4</td>
<td>99.85</td>
<td>99.72</td>
</tr>
<tr>
<td>5</td>
<td>99.76</td>
<td>99.56</td>
</tr>
<tr>
<td>6</td>
<td>99.72</td>
<td>99.59</td>
</tr>
<tr>
<td>7</td>
<td>99.75</td>
<td>99.66</td>
</tr>
<tr>
<td>8</td>
<td>99.92</td>
<td>99.85</td>
</tr>
<tr>
<td>9</td>
<td>99.95</td>
<td>99.89</td>
</tr>
</tbody>
</table>

**AVG** 99.77 99.67

As it is shown in Table II, 10-fold cross validation performance results of SVMs and CRFs are very comparable; SVMs outperformed CRFs on all level folds and on 10 folds average level (99.77% vs. 99.67%). Analyzing confusion matrices results for the folds where the tokenizers have achieved the best performance, we noticed that the tokenizer based on SVMs has slightly better results than the one based on CRFs for the five classes presented above.

IV. POS TAGGING EXPERIMENTS AND RESULTS

In this section we present POS tagging experiments baselines, used features and a comparison between performance of the new and the old tag sets.

A. Baselines

In these experiments, we have chosen to use two baselines in order to better understand our results. These baselines are, namely:

1- Frequency-based baseline (Freq-Base.): This is a non-learning algorithm. The predicted tag for a certain token is simply the most frequent POS tag that has been associated with it in the training data. Thus, this baseline completely ignores the surrounding 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. Its source code is freely available.

2- Best-case baseline (Best-Base.): To study the best case scenario, we start with \( M_{\text{init}} \) and aggregate data from the remaining 30% of the manually annotated data in blocks of 2k. This will be helpful to provide a contrast to the models that will be automatically annotated data. In order to choose the best \( M_{\text{init}} \) for our experiments, we have taken 10 corpora size points, with a step of 125 sentences between each two points. In this experiment set, we have run 10-fold cross validation. In the tenth point, where we use the whole manually annotated data, we have obtained the best F-measure in the fifth fold using CRFs. Table III presents 10-fold cross validation results for the tenth point containing 1438 sentences.

B. Experiments settings and POS tagging results

The employed features are the following:

1- The current token;

2- Lexical features: this consists of the last and first ‘i’ character n-grams, with ‘i’ spanning from 1 to 4. Figure 4 presents an example of what this features look like;

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

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

Table III. 10-fold cross validation results using tokenization.

<table>
<thead>
<tr>
<th>Fold#</th>
<th>BASELINE</th>
<th>SVMs</th>
<th>CRFs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>79.7</td>
<td>85.12</td>
<td>86.02</td>
</tr>
<tr>
<td>1</td>
<td>77.36</td>
<td>83.25</td>
<td>84.28</td>
</tr>
<tr>
<td>2</td>
<td>84.03</td>
<td>90.75</td>
<td>89.48</td>
</tr>
<tr>
<td>3</td>
<td>81.00</td>
<td>87.89</td>
<td>88.2</td>
</tr>
<tr>
<td>4</td>
<td>80.11</td>
<td>88.36</td>
<td>89.35</td>
</tr>
<tr>
<td>5</td>
<td>81.47</td>
<td>90.24</td>
<td>91.18</td>
</tr>
<tr>
<td>6</td>
<td>77.29</td>
<td>83.18</td>
<td>84.27</td>
</tr>
<tr>
<td>7</td>
<td>76.95</td>
<td>83.84</td>
<td>85.32</td>
</tr>
<tr>
<td>8</td>
<td>84.22</td>
<td>89.33</td>
<td>90.31</td>
</tr>
<tr>
<td>9</td>
<td>86.45</td>
<td>89.20</td>
<td>91.12</td>
</tr>
<tr>
<td>AVG</td>
<td>80.85</td>
<td>87.11</td>
<td>87.95</td>
</tr>
</tbody>
</table>

Figure 5 shows the obtained results for ‘Best-Base’ and ‘Freq-Base’ both using manually annotated data. The learning curve is increasing along training corpus size. The ‘Freq-Base’ is at least 8 points below CRFs and SVMs across the curve.

We started with \( M_{\text{init}} \) 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 is 1.55% and 1.23% for CRFs and SVMs respectively.

---

7 http://www.outamed.com/downloads/baseline.txt
C. AMTS performance in POS tagging

AMTS tag set was conceived in order to capture finer grammatical distinctions, and to be used in wide range of applications. The following criteria were used in the development of this POS tag set:
- Been based on the underlying linguistic theory presented above, but not too shallow as the one used previously;
- Using mnemonic tag names;
- Taking into consideration tokenization issues.

Using the old tag set containing 13 tags, SVMs and CRFs results 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%). Based on this more fine grained tag set of 28 tags, CRFs outperformed SVMs on the fold level (87.95% vs. 87.11%) and on the 10 folds average level (91.18% vs. 90.75%). These results are very promising considering that we have used a corpus of only ~20k tokens.

Regarding classes that we have split to 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% against 94.6% for N. However obtained accuracy for proper nouns is just 54.16%, due essentially to insufficient examples in training set. Concerning V class corresponding to verbs, which were split into VB for verbs base form and VBP for participles, VB precision is 94.22% against 93.3%.

CONCLUSIONS

Very few linguistic resources have been developed so far for Amazigh and we believe that the development of a POS-tagger tool is the first step needed for automatic text processing. In line with this, we presented how we provided this scarce resource language with a more fine-grained tag set of 28 tags. Then, we have addressed the basic principles we followed in the tagging process. Afterwards, we have presented the tokenizer and POS tagging experiments and results for Amazigh based on two state-of-the-art supervised learning technique, namely SVMs and CRFs.

Results show that the tokenizer performance of SVMs and CRFs are very comparable. Across the board, SVMs outperformed CRFs on the fold level (99.77% vs. 99.67%) and on the 10 folds average level (99.95% vs. 99.89%). Regarding POS tagging results, performance of SVMs and CRFs are also very comparable in this task. Across the board, CRFs outperformed SVMs on the fold level (87.95% vs. 87.11%) and on the 10 folds average level (91.18% vs. 90.75%). These results are very promising considering that we have used a corpus of only ~20k tokens.

In order to obtain a more accurate POS tagger using semi-supervised learning techniques, we have gathered a set of unlabeled data that we have used. The total size of the collected corpus is 225,240 tokens. This corpus is freely available. Preliminary experiments on amazigh POS tagging using self-training were conducted. The achieved error reduction is 1% using system confidence for sentences as measure to choose the best sentences for self-training.

In the near feature, we aim at exploring also the possibility of using confidence measure and informativeness criteria for self-training.

ACKNOWLEDGMENTS

The work of the fourth 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


www.outamed.com/downloads/Amazigh_Corpus.txt


