A Multi-Agent System for POS-Tagging Vocalized Arabic Texts

Chiraz Ben Othmane Zribi, Aroua Torjmen, and Mohamed Ben Ahmed
RIADI laboratory, University of La Manouba, Tunisia

Abstract: *In this paper, we address the problem of Part-Of-Speech(POS) tagging of Arabic texts with vowel marks. After the description of the specificities of Arabic language and the induced difficulties on the task of POS-tagging, we propose an approach that combines several methods (stochastic and rule-based). For the implementation of these methods and the global POS-tagging system, we adopted a multi-agent architecture. In which, five tagger agents work in parallel, each one applies its own method, in order to propose for each word in a sentence the suitable tag among those proposed by the morphological analyzer. The tagger agents cooperate together and with the unknown words solver agent to resolve unknown words. A voting agent decides in the end, which tag to affect to each word. Finally, we present the experimental protocol we used to evaluate the system carried out in this work and the obtained results that we consider very satisfactory.*

Keywords: *Natural Language Processing (NLP), Arabic language, morphological analyzer, part-of-speech tagging, hybrid methods, multi-agent system.*

Received February 3, 2006; accepted April 22, 2006

1. Introduction

POS-tagging is the process that consists in assigning grammatical characteristics (substantive, verb, adjective, etc.) to words in their contexts.
Example: The (*definite article*) boy (*substantive*) eats (*verb*) a (*indefinite article*) cake (*substantive*).

The process of POS tagging was widely automated for English and French and for many others European languages giving a rate of accuracy ranging from 95 % to 98 %. On the web, we find many tagged corpora as well as programs of POS-tagging for these languages. The methods used by these POS-taggers are various, namely stochastic approaches such as the Hidden Markov Model [4], the decision trees [9], the maximum entropy model [8], rules-based approaches inspired in their majority of the transformation rules-based POS-tagging [3], hybrid approaches [7] (statistics and rules-based), or combined ones [2, 10].

Unfortunately, the situation is different for Arabic as there are, in our knowledge, neither POS-taggers nor tagged corpuses available. Nevertheless, some Arabic POS-taggers [5, 6, 11] started to appear with an average accuracy going from 85% to 90% for texts with vowel marks and by about 65% for texts without vowel marks.

This gap noted for Arabic language is especially due to, its particular characteristics, which, involve firstly, a rate of grammatical ambiguity relatively more significant than for other languages, and secondly, make impossible the application of existing POS-taggers without any change. Thus, obtaining a high accuracy remains a challenge to reach for Arabic language.

Accordingly, we propose a POS-tagging system for Arabic texts, getting as an input a morphological analyzed text. Due to the complexity of the problem, and in order to decrease grammatical ambiguity, we have restricted the scope of our investigation: we only treat texts with vowel marks in words. Although the majority of Arabic texts are without vowel marks, there are as well semi-vocalized or completely vocalized texts. These latter are generally schoolbooks or stories for children.

The remainder of this paper is organized as follows. Section 2 presents the Arabic language characteristics making the task of POS-tagging more difficult. Sections 3 and 4 show briefly the morphological analyzer and the tag sets we used. In sections 5, we establish the general principle of our combined approach. In section 6, we show the architecture of our multi-agent system and present a detailed description of the work of each agent. Section 7 explains the method we used to evaluate the efficiency of our system and the results obtained. Finally, in section 8 we conclude the paper and present future possible work.

2. Difficulties of Arabic Language

In Arabic, the problem of POS-tagging is much more complicated than in other languages. Indeed, Arabic is a morphologically complex language that has numerous writing constraints such as vowels, agglutination and grammatical ambiguity, which can

lead to ambiguities.

2.1. Vowel Marks

The vowel marks in words, are vocalic signs that allow the reading and the comprehension of texts written in Arabic. Without vowels, the reader has to see the context to find the good vowels of a given textual form, because Arabic words are vocalically ambiguous. This vocalic confusion involves naturally much more grammatical ambiguity.

Table 1. Example of vocalic ambiguity.

كتب		
كَتَّبْ	*Kattib*	*Make write*
كُتِّبَ	*Kuttiba*	*Has been made written*
كُتِبَ	*Kutiba*	*Has been written*
كَتَبَ	*Kataba*	*Wrote …*

2.2. Agglutination

Arabic is an agglutinative language. Textual forms are made of the agglutination of proclitics (e. g., articles, prepositions, conjunctions) and enclitics (e. g., linked pronouns) called clitics, to stems (inflected forms called also surface forms).

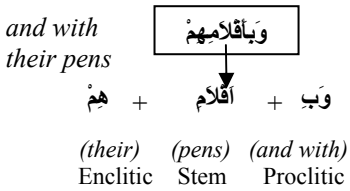

Figure 1. Example of an agglutinative word.

A textual form, without considering any context, can be segmented in different ways. The ambiguities of decomposing textual forms induce a significant ambiguity of tagging. When the text is without vowel marks the decomposing ambiguity increases generally. For example, the unvocalized textual form: المهم can have these three segmentations:

Table 2. Example of segmentation ambiguity.

المهم		
أ+لم+هم	*A+lamma+hum*	*Did it collect them?*
+ألم+هم	*Alamuhum*	*Their pain*
ال+مهم+	*Almuhim*	*The important*

2.3. Grammatical Ambiguity

Arabic words are grammatically ambiguous. The statistics carried out [1] in definition[1] confirm this ambiguity. The author noted the importance of the rate of grammatical ambiguity for the lexical forms with vowel marks, which is equal to 2.8 on average. This rate increases by the absence of the vowel marks to reach 5.6 possible tags per lexical form. Because of the agglutination of affixes to lexical forms, the rate of grammatical ambiguity is more significant for textual forms. According to the counting (considering comparable tags) carried out [5] on texts with vowel marks this grammatical ambiguity rate is equal to 5.6 on average, and could reach an average of 8.7 for texts without vowel marks.

3. Morphological Analyzing

Because of the ambiguities of decomposing textual forms on inflected forms and clitics, we generally need a morphological analyzer to obtain automatically the different segmentations.

For our tagging purpose, we use a rule- based morphological analyzer as shown in Figure 2, that we have developed in a previous work. This analyzer is well-tried and is able to analyze vocalized words, partially vocalized words and non-vocalized words.

It makes use of two dictionaries. The first and the larger one, has been constructed semi-automatically. It contains inflected Arabic forms with vowel marks (it counts about 1 615 159 entries corresponding to 577 546 unvocalized forms) such: كَتَبَ (*he wrote*), كَتَبَتْ (*she wrote*), كَتَبُوا (*they wrote*),….. Each form has different linguistic information us: lemma, POS-tag, gender, number, pronoun, etc.

Apart from the dictionary of inflected forms, this morphological analyzer uses a small dictionary that contains all clitics (it counts 89 entries corresponding to 75 unvocalized forms) such: الْ (*the*), وَبِ (*and with*), هُ (*his*),… and applies a set of rules for the research of all possible segmentations in proclitic, stem and enclitic with their associated information.

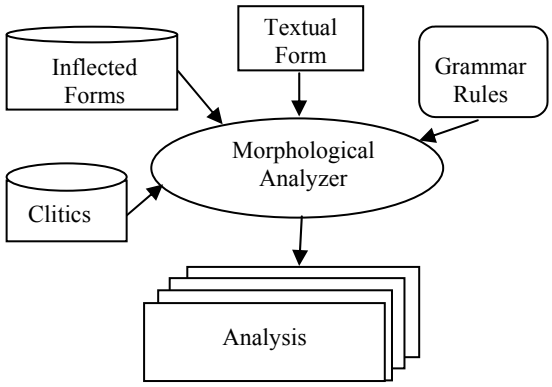

Figure 2. General principle of the morphological analyzer.

4. Tag Sets

The problem of the definition of tag sets is subtle and remains up to now topical. Even, for one language, we can find different tag sets. The cardinality of the tag sets varies, in general, from about ten to few hundreds. To define the tag sets, one must consider both parts of speech and distributional character of words in a sentence in order to assign differ
ent tags to words having different distributional behaviours [5].

[1] For an Arabic dictionary

Defining a tag set with great cardinality amounts handles fine tags. This allows having a higher distributional accuracy but risks to slow down the process of training and to weaken the rates of accuracy rate of tagging. However, the use of coarser tags improves generally the accuracy rates but contains less linguistic information affecting thus the quality of tagging. The ideal thing would be to find a good compromise between the fineness of tags and their distributional acuity.

The tagging system suggested here, exists in two versions using two principal tag sets. The first set contains grammatical tags of fine granularity, called micro-tags, and counting 223 tags for inflected forms and 65 tags for enclitics. These tags are listed in our two dictionaries (previously described) and used by our morphological analyzer. To define them we took into account various criteria being able to be discriminating for the words distribution. We cite for example: parts of speech (substantive, verb...), the inflection for names (nominative, accusative, genitive), the aspect for verbs (accomplished, unaccomplished, imperative), etc.

Examples:
اسم معرف مرفوع (definite nominative substantive)
فعل مضارع منصوب مخاطب (unaccomplished subjunctive interlocutor verb).

The second tag set relates to complex grammatical tags (hyper-tags), such as for a given word we consider the licit combination of the proclitic tag, the tag of the stem (inflected form) and the enclitic tag. We thus count 465 well-formed complex tags.

Examples:
أداة تعريف + اسم معرف مرفوع *(definite particle+definite nominative substantive)*
فعل مضارع منصوب مخاطب+ضمير متصل *(unaccomplished subjunctive interlocutor verb+linked pronoun)*

The use of this tag set is argued by the fact that it contains complex tags having a larger range than that of the simple tags. This, in our belief, will reinforce the efficiency of grammatical tagging by improving its results.

Two other sets of tags were useful for the evaluation of our system and resulted from the regrouping of the simple tags (micro-tags). These are macro-tags set counting 22 tags for inflected forms and basic parts of speech set holding the following three tags: Substantive, Verb and Particle.

These two tag sets were not used in the tagging process. Their use allows a simple adjustment of the results obtained by tagging with simple tags. This matching allowed the adaptation of the results of our tagging system to the various needs of applications requiring grammatical tagging as a pre-treatment.

5. Suggested Approach

The method that we propose here is a supervised training method: a training corpus is pre-treated in order to extract from it the training data handled by the tagging system and a testing text is used for experimentations and the evaluation of the system.

The treatment of the training corpus consists in semi-automatic labelling it. First, the corpus is analyzed automatically by the morphological analyzer, which provides for each word the set of its possible analyses (e. g., segmentations, tags...), then it is manually disambiguated. However, it is important to say that we made the analyzer providing only linguistic data which is directly useful for us (e. g., segmentation, tag) while omitting to provide other linguistic data which we didn't take into account (ex. lemma, gender and a number, transitivity, pronoun).

The testing text is morphologically analyzed. We obtained thus, for each textual form different possibilities of analyses. Although, the principal goal of our tagging system is to assign the appropriate tag to each form, it is indirectly constrained to solve other ambiguities such as those related to the segmentations, as we will see it later.

Here is an example of analysis provided for a textual form extracted from the testing text. The tags are represented by numbers, which we have affected to them.

Analysis of the word n°15: [وَلَـٰكِنَّهُ]
Segmentation 1: [وَلَ/كِنَّ/هُ]
Proclitic: [وَلَ]|Tags: (59, 60, 4)
Stem: [كِنَّ]|Tags: (193)
Enclitic: [هُ]|Tags: (47, 38)
Segmentation 2: [وَ/لَـٰكِنَّ/هُ]
Proclitic: [وَ]|Tags: (2, 26, 29, 3, 53, 55)
Stem: [لَـٰكِنَّ]|Tags: (58)
Enclitic: [هُ]|Tags: (47, 38)

Figure 3. Example of an input for the POS-tagging system.

To achieve our POS-tagging system, we opted for combining methods in a multi-agent architecture.

5.1. Combined Method

We combine different methods trying to benefit from the advantages for each method used and to improve the tagging accuracy. This implies the construction of a number of POS-taggers where every one operates according to the principle of the method that it represents. Each POS-tagger proposes one tag for the treated word and by voting, the best one will be assigned as the final tag to the target word.

5.2. A Multi-Agent Architecture

In addition to its originality, the following arguments justify the choice of this architecture:

- Combination of several methods.

- Competition and parallel work of agents.
- Communication and cooperation between agents.

6. POS-Tagging System

Some agents participate to accomplish the global objective of our POS-system that consists in assigning an appropriate tag to each textual form of a given text. We cite:

- Sentences' extracting agent.
- Tagger agents.
- Voting agent.
- Unknown words solver agent.

The following figure illustrates the general architecture of this system.

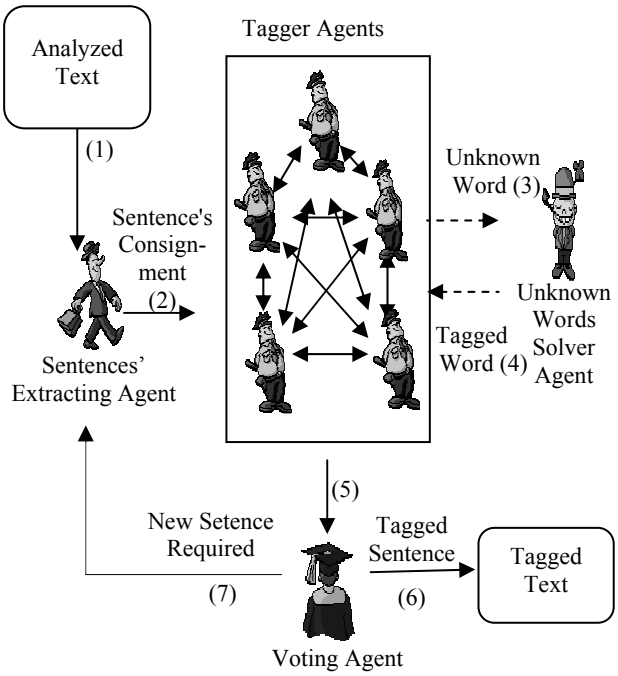

Figure 4. General architecture of the POS-tagging system

6.1. Sentences' Extracting Agent

This agent is responsible for the extraction of sentences from the analyzed text to tag. For each textual form in a sentence, the morphological analyzer proposes a set of segmentations (i.e. proclitic+stem+enclitic) and for each such segmentation a set of tags. When it loads a sentence, the sentences' extracting agent activates all tagger agents to start the tagging of this sentence.

6.2. Tagger Agents

Given a sentence, five POS-taggers agents will work in parallel, each one applies its own method, aiming to find for each word of the sentence the suitable tag among those proposed by the morphological analyzer.

6.2.1. Resolving Ambiguities of Segmentations

In addition to its principal function of grammatical tagging, each tagger agent has to choose one above all the correct segmentation relative to a given textual form when this latter, has different possibilities of segmentation.

It should be noted that when it is a question of handling completely vocalized texts, the phenomenon of segmentation is negligible but this does not prevent its existence. A counting which we carried out in a former work, showed, showed that the number of possible segmentations can reaches 5 for an unvocalized textual form and 2 for a vocalized one.

For a given textual form, to choose the correct segmentation, each tagger agent works according to the following principle:

1. Recovers the set of tags proposed by the morphological analyzer for each proposed segmentation
2. Adds the number of licit binary successions between each tag of the given segmentation and the tag of the preceding word. The licit successions are obtained by consulting a binary matrix of precedence frequencies built from the training corpus.

The two stages (1) and (2) are as many repeated as there are segmentations for the treated textual form. The tagger agent will then keep the segmentation of which the number of licit successions is the highest. Once the correct segmentation is found, it will work on the disambiguation of the obtained tags according to its own method.

6.2.2. Unigram Tagger Agent

For each word of a received sentence, the unigram tagger:

1. Accedes to a lexicon which is containing the different words of the training corpus and their tags with their occurrence's frequencies;
2. Seeks the target word in this lexicon and chooses the most frequent tag for this word.

6.2.3. Bigram Tagger Agent

This tagger uses the binary succession probabilities recovered from the training corpus and saved in a binary succession matrix. We calculate the binary succession probability as follows:

$$p(t_i \setminus t_{i-1}) = \frac{number\ of\ occurrences\ of\ the\ succ\ (t_{i-1}, t_i)}{number\ of\ occurrences\ of\ t_{i-1}}$$

The bigram tagger follows theses steps to tag a word, it:

1. Recovers the tag of the word preceding the target word.

2. Accedes to the matrix of binary succession probabilities;
3. Chooses the tag belonging to the set of tags proposed by the morphological analyzer having the higher binary transition probability considering the tag of the previous word and assigns it to the target word.

6.2.4. Trigram Tagger Agent

This trigram tagger agent works similarly to the precedent one, but it takes into account ternary succession probabilities recovered from the training corpus and saved in a ternary succession matrix. We determine the ternary succession probability as follows:

$$p(t_i \setminus t_{i-2}, t_{i-1}) = \frac{\text{numberof occ. of the succ}(t_{i-2}, t_{i-1}, t_i)}{\text{numberof occ. of the succ}(t_{i-2}, t_{i-1})}.$$

Here, the principle of tagging a word in a given sentence consists in:

1. Recovering the two previous tags in relation with the target word;
2. Acceding to the matrix of ternary transition probabilities;
3. Assigning to the target word the tag, which belongs to the tags proposed by the morphological analyzer and has the higher ternary transition probability considering the two tags of the two previous words.

6.2.5. Hidden Markov Model Tagger Agent

This tagger agent operates according to Hidden Markov Model's principle.

Given a sequence of n words $W = w_1 \ldots w_n$, this tagger tries to find the tag sequence $T = t_1 \ldots t_n$, that maximizes the conditional probability $p(T\setminus W)$.

We note:

$$Max\ T = arg\ Max_T\ p(T\setminus W)$$

By some assumptions[1]:

$$Max\ T = arg\ Max_T \prod_{i=1}^{n} p(w_i \setminus t_i) \times p(t_i \setminus t_{i-1})$$

where:

$p(wi\setminus ti)$ is the emission probability that is calculated with the following formula:

$$p(w_i \setminus t_i) = \frac{\text{number of occ. of } w_i \text{ tagged with } t_i}{\text{number of occurrences of } t_i}$$

and $p(t_i \setminus t_{i-1})$ is the transition probability that is determined as follows:

$$p(t_i \setminus t_{i-1}) = \frac{\text{number of occ. of the succ.}(t_{i-1}, t_i)}{\text{number of occurrences of } t_{i-1}}$$

with:

$p(t_1\setminus t_0) = p(t_1)$ called initial probability.

When it receives a sentence, including for each word all the tags proposed by the morphological analyzer, this tagger agent applies the VITERBI algorithm[2]. This latter takes all the needed frequencies from the training corpus and tries to find the tag sequence that has the maximum likelihood.

6.2.6. Agent Based on Sentence Patterns

We present here a new method based on sentence patterns that has not been approached before (at least for tagging).

We define a sentence pattern as a model of sentence made of a succession of tags.

The sentence: تلعب البنت في الحديقة (*the girl plays in the garden*)

is segmented as: تلعب+ال+بنت+في+ال+حديقة and matches with the following pattern:

فـ.مض.ـُ +أداة تع.+اسم مع.ـُ +حرف جر+أداة تع.+اسم مع.ـِ

(unaccomplished indicative verb +definite article +definite nominative substantive + genitive particle+ definite article + definite genitive substantive)

In the practice, the possession of all sentence patterns for a language is difficult. That is why this tagger agent manipulates the longest successions of tags of adjustable size and considers the positional character of tags in the sentence (1st tag, 2nd tag, 3rd tag …).

The principle of tagging the words in a sentence consists in:

1. considering the first word of the sentence and extracting the tags that have been assigned by the morphological analyzer;
2. seeking from the set of sentence patterns, the patterns which start with one of the tags proposed by the morphological analyzer;
3. going to the second word and seeking among models found in patterns, in which the second tag correlates to one of tags proposed by the morphological analyzer for this word.

This process is repeated until the words of the sentence are tagged completely. During the matching between the tags proposed by the morphological analyzer for the focused word and the tags of patterns, the position of words is token into account. Thus, the number of the candidates patterns decreases progressively when the tagger goes forward in the treatment of the sentence.

If for a given word w_i, no tags proposed by the morphological analyzer appears in one of the training patterns at the same position *i*, the tagger examines all the training patterns to extract the longest succession tags called "sub-patterns" whose the first tag matches with one of these tags (Figure 5)

[1] Independency assumption and Markov assumption k=1 (using binary successions)

[2] http://en.wikipedia.org/wiki/Viterbi_algorithm.

When the extraction is made, the tagger joins the "sub-patterns" to the segments previously retained, to form new patterns that are going to serve to the tagging of following words according to the principle cited previously.

To every sentence pattern, we assign a weight that is calculated from the product of ternary succession probabilities. The values of theses probabilities are recovered from the ternary succession matrix (previously cited).

When all words of the sentence have been treated, and if several candidate patterns were kept as result of the tagging, the tagger agent chooses the pattern having the highest weight. If several patterns have the same weight then it keeps the one that is most frequent in the training corpus.

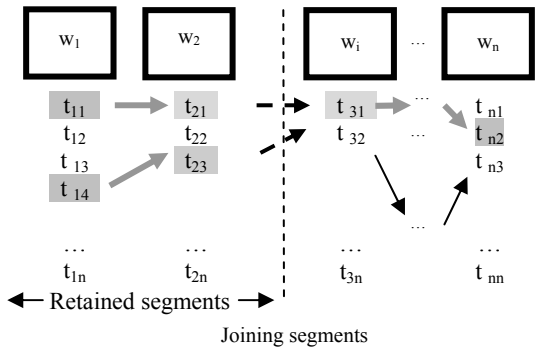

Figure 5. Example of concatenation of patterns.

6.3. Unknown Words Solver Agent

We have two cases of an unknown word:

1. Word receiving no tags from the morphological analyzer.
2. Word receiving tags from the morphological analyzer which are not found in the training data.

For the two cases, the tagger agent in work asks the assistance help of the other tagger agents. Thus, these tagger agents cooperate and communicate mutually to resolve unknown word. If one of them solves this problem, it sends to the tagger agent asking help, the founded tag. If no tagger could help it, the tagger agent calls the unknown word resolution agent.

Once the unknown word is resolved, the proposed tag is assigned to the unanalyzed word. However, in the second case, the agent tagger proceeds firstly by some checking: so, it verifies if the proposed tag exists among the set of tags proposed by the morphological analyzer, and then assigns it to the target word. Otherwise, the tagger obtains from to the training data the most frequent tag among the set of tags proposed by the morphological analyzer and assigns it to this word.

To guess a tag, the unknown words resolution agent works according to this principle: It uses templates of verbs and proper names as well as lexical rules to determine the nature of the treated word (substantive, proper name, verb…). If the word to tag has a scheme of a verb or of a proper name as shown in Table 3 the solver agent takes the corresponding tag from a training list containing templates with their corresponding tags.

Table 3. Examples of templates for proper names and verbs.

Templates	Examples	POS Tags
فَعْلاَنُ	عَدْنَانُ	اسم علم مرفوع *Nominative proper noun*
فُوعِلَ	تُونِسَ	اسم بلاد منصوب *Accusative country noun*
إِفْعَلُوا	إِذْهَبُوا	فعل أمر *Imperative verb*

Otherwise, it considers the unknown word as a substantive and applies lexical rules to find the suitable tag.

Example:

A word starting with الْ (definite article) and ending with ضمّة ُ(nominative case) is likelier to be اسم معرّف مرفوع (definite nominative substantive).

6.4. Voting Agent

The tagger agents are synchronized in such way that after achieving their works, the voting agent is started to decide which tag to assign for a word. We have two cases: If all taggers (or the majority) elect the same tag then this tag is affected to the target word. Otherwise, if all taggers are in a total disagreement, the voting agent uses heuristics to decide and to choose one and only one tag to assign to this word. These heuristics are:

- *The Reliance Degree*: Voting agent considers the tag of the tagger having the higher reliance degree. For each tagger an indicator is provided and incremented each time the voting process considers its tag in the vote.
- *The Historic*: In case two or several taggers have the same highest reliance degree, the voting agent sees the historic of every tagger in competition and chooses the one, which previously achieved the best tagging accuracy.

7. Experimentations and Results

Our experiment was carried out in two stages: one stage of training during which, a textual corpus[1] containing about 18 000 textual forms was manually annotated and probabilities were collected. The second stage was the testing, which consists in using these probabilities to tag automatically a testing text (out of the training data) containing about 1 200 words.

[1] Children stories, collected by Al-Sulaiti L. http://www.comp.leeds.ac.uk/eric/latifa/arabic-corpora.htm

For the system evaluation, we used the accuracy rate that is calculated as follows:

$$\text{Tagging accuracy} = \frac{\text{number of correctly tagged words}}{\text{total number of tagged words}}$$

As shown in Table 4, the probabilistic taggers are not very efficient (maximum accuracy of 93.65% for micro-tags and 93.21% for composed tags) since they require a big training data. In general, the accuracy for all taggers increases (except for the Trigram tagger), when we manipulate the composed tags (hyper-tags). This improvement can be explained by the fact that complex tags have a larger range than that of the simple tags (as we previously mentioned).

Table 4. Percent tagging accuracy.

Taggers	Micro Tags	Hyper Tags	Macro Tags	S	V	P
Unig.	90.59	90.81	91	95	80	99
Big.	91.88	93.21	93.02	96.48	82.25	98.02
Trig.	93.65	90.27	95.32	95.32	89.21	100
HMM	87.43	91.19	90.82	90.82	82.18	96.55
Patt.	94	93.9	95.5	90	98.3	97
Vote	96.01	96.2	98.1	99.8	92.02	99.1

We observe also that the use of macro tags increases significantly the accuracy of the taggers. This phenomenon shows clearly how coarser tags improve the accuracy rates. This is due in our case, to the nature of mistakes caused by taggers that confuses tags belonging to the same class of tags.

Furthermore, we notice that the new method of tagging based on sentence patterns gives the best results for the three first tag sets. And because of the diversity of mistakes that taggers provoke, the accuracy of the global system is generally higher than the tagging accuracy of each tagger.

However, we consider this accuracy (using the simple and composed tags) satisfactory compared to results achieved by other Arabic tagging systems in a rather similar environment of experimentation.

Finally, it is important to say that even if accuracy rates give us an idea on the performances of POS-taggers, it is very difficult to make a good comparison between them because neither the tag sets nor the data of training and testing are generally similar.

8. Conclusion

Our POS-tagging system is based on a combined approach. The efficiency of this approach was proved by the accuracy generated by the global system. In fact, this accuracy is generally higher than the tagging accuracy of each tagger. The new method of tagging based on sentence patterns gives also satisfactory results and proves to be promising. In spite of the lack of our training data and the ambiguous specificities of the Arabic language, the choices that we adopted enabled us to reach our initially drawn up goals. However, we can ameliorate the results by improving largely the training data. Moreover, we plan to treat semi-vocalized and unvocalized texts in a further work.

References

[1] Ben Othmane C., "De la Synthèse Lexicographique A La Détection Et La Correction Des Graphie Fautives Arabes," *PhD Thesis*, Université de Paris XI, Orsay, 1998.

[2] Brill E. and Wu J., "Classifier Combination for Improved Lexical Disambiguation," *in Proceedings of 36th ACL and 17th COLING*, Montreal, Canada, pp. 191-195, 1998.

[3] Brill E., "Some Advances in Transformation-Based Part of Speech Tagging," *in Proceedings of the 12th National Conference on Artificial Intelligence*, pp. 722-727, 1992.

[4] Cutting D., Kupiec J., Pedersen J., and Sibun P., "A Practical Part-Of-Speech Tagger," *in Proceedings of the 3rd Conference on Applied Natural Language Processing*, pp. 133-140, 1992.

[5] Debili F., Achour H., and Souissi E., "La Langue Arabe Et L'Ordinateur: De L'étiquetage Grammatical A La Voyellation Automatique," *Correspondances,* Institut de Recherche Sur Le Maghreb Contemporain, CNRS, no. 71, pp. 10-28, 2002.

[6] Khoja S., "APT: Arabic Part-of-speech Tagger," *in Proceedings of the Student Workshop at the Second Meeting of The North American Chapter of the Association for Computational Linguistics (NAACL'01)*, Carnegie Mellon University, Pennsylvania, pp. 20-26, 2001.

[7] Marshall I., "Choice of Grammatical Word-class without Global Syntactic Analysis: Tagging Words in the LOB Corpus," *Computers and the Humanities*, no.17, pp. 139-50, 1983.

[8] Ratnaparkhi A., "A Maximum Entropy Model for Part of Speech Tagger," *in Proceedings of the 1st Empirical Methods in Natural Language Processing Conference*, Philadelphia, USA, pp. 133-142, 1996.

[9] Schmid H. and Stein A., "Etiquetage Morphologique de Textes Français Avec Un Arbre de Décision," Le Traitement Automatique Des Langues: Traitements Probabilistes Et Corpus, vol. 36, no. 1-2, pp. 23-35, 1995.

[10] Sjöbergh J., "Combining POS-Taggers for Improved Accuracy on Swedish Text," *NoDaLiDa*, Reykjavik, 2003.

[11] Zemirli Z. and Khabet S., "TAGGAR : Un Analyseur Morphosyntaxique Destiné A La Synthèse Vocale Des Textes Arabes Voyellés," *Traitement Automatique de l'Arabe*, Fès, 2004.

Chiraz Ben Othmane Zribi received her PhD in computer science in 1998 from PARIS XI University, France. Currently, she is an associate professor at the Department of Computer Science at Faculty of Sciences of Bizerte in Tunisia. She is actually a member of the RIADI Research Laboratory at University of La Manouba. Her current research interests are in the area of Arabic language processing. Her recent work has focused on natural language parsing based on hybrid learning (neuro-symbolic), detection and correction of errors, generation of dictionaries and knowledge retrieval. Her research interests center around computational learning theory and hybrid models of language.

Aroua Torjmen received her Master degree in computer science from the National School of Computer Science, University of La Manouba, Tunisia, in December 2005. She is an assistant professor in the Higher School of Computer Science, University of Tunis-El Manar, Tunisia. Currently, she is a PhD student at the RIADI research laboratory, University of La Manouba.

Mohamed Ben Ahmed received his PhD in computer science in 1978 at Paris IX University, France, with first class honour degree. Currently, he is an emeritus professor in computer science at University of La Manouba, Tunisia. He is the head of RIADI research laboratory from 1992 till now. He occupied higher administrative positions from 1993 to 1999. He is the author of seven books and of a great number (about 200) of scientific publications and reports.