Parallel HMM-Based Approach for Arabic Part of Speech Tagging

Ayoub Kadim and Azzeddine Lazrek
Department of Computer Science, Faculty of Science, Cadi Ayyad University, Morocco

Abstract: *In this paper we try to go beyond the classical use of the Hidden Markov Model for Part Of Speech Tagging, particularly for the Arabic language. In fact, most available Arabic tagging systems and tagsets are derived from English and do not make use of the linguistic richness of Arabic. Our new proposed tagging system will consist of two Hidden Markov Models working in parallel: In addition to the main model, a second model is added to serve as a reference for low probabilities tags. Of course, a dual corpus is required to train both models. To do so, we restructure the Nemlar Arabic corpus and extract a new tagset from diacritics and grammatical rules. The approach is implemented by using Java programming environment and several experimentations are conducted to evaluate it. The results of this approach, which are promising, as well as its limitations, are deeply discussed and future possible enhancements are also highlighted. This work will open the door for new promising research perspectives, particularly for the Arabic language processing, and more generally for the applications of Hidden Markov Models.*

Keywords: *Part of speech tagging, hidden Markov model, Viterbi algorithm, natural language processing, corpus, arabic language.*

Received May 31, 2014; accepted December 21, 2015

1. Introduction

Part Of Speech (POS) Tagging aims to assign each occurrence of a corpus a symbol representing a grammatical category (noun, verb, etc.,) and associated morphological information (masculine, singular, etc.,) [30]. POS tagging is one of the well-known Natural Language Processing (NLP) research areas as it represents an important precursor to various natural language processing tasks [9]. In fact, in text processing steps, once the text is tagged, it could be used for deeper analysis [3]. Nevertheless, as the first works in POS tagging have been designed for English, we noticed that the Arabic taggers are often based on classifications (tagsets) derived from the English language that differs from Arabic. Indeed, the Arabic language is characterized by several difficulties that represent a challenge in POS tagging and require special processing, such as vowels, agglutination and grammatical ambiguity [31]. In addition, the Arabic taggers are intended primarily to Modern Standard Arabic (MSA), which can lead to bad results when used for the Traditional or Classical Arabic (CA), which employs sequences of words rarely encountered in the learning corpora of these taggers. Therefore, we propose through this work a new approach for Arabic language processing by manipulating words from several aspects and taking into account several tagsets from various classifications supported by the Arabic language to increase the tagging accuracy. A new approach of parallel tagging has been proposed: a basic tagger, performing initial tagging, is combined with a second tagger used to introduce corrections in low probability cases. We also added a three-dimensional matrix to link the two tagsets underlying the aforementioned taggers. The Nemlar corpus, which appeared large enough and rich in detail, was used as a basis of our work by restructuring its content to extract a new tagset based on final diacritics and other grammatical data.

2. Works Related to Arabic POS Tagging

Several works for developing Arabic taggers based-on different approaches have been recently emerged. Besides commercial products developed by specialized companies (Xerox, Sakhr, Research and Development International (RDI), etc.) research efforts are developing in the scientific community (e.g. [8, 14, 18, 23, 24, 28, 31]).

Kübler and Mohamed [24] proposed two approaches for POS tagging of Arabic using the full POS tagset of the Penn Arabic TreeBank. The first approach uses complex tags that describes full words and does not require any word segmentation. The second approach is segmentation-based, using machine learning for segmenting. The indicated accuracy of these approaches is respectively 93.93% and 93.41%. They report that word based tagging gives best results on known words, while the segmentation-based approach gives better results on unknown words. To deal with the unknown words problem, El-Jihad *et al*. [17] proposed a tagging system based on the patterns of unknown words and

the Hidden Markov Model, with a set of 3800 patterns and 52 POS tags. The experimental results gave that 99.14% of the sentences containing unknown word have been correctly tagged [17]. Khoja [23] uses a combination of both statistical and rule-based techniques with a tagset of 131 basically derived from the British National Corpus (BNC)[1] tagset. She worked on three stages:

1. An initial stage in which every word is looked up in the lexicon to assign all possible tags for words specified in the lexicon.
2. A stemming stage is used for words that are not found in the dictionary in the initial stage; affixes are used to help determine tags of those words.
3. Finally, a statistical tagger based on the Viterbi algorithm has been developed and used to disambiguate words that have more than one tag (ambiguous words and unknown words).

An accuracy of 85% was reported. Diab *et al*. [14] uses Support Vector Machines (SVM) as classification approach to model Arabic POS tagging using a manually reduced tagset of 24 POS tags from the Penn Treebank [26]. They used a large feature set and report an accuracy of 95.5% on all tokens drawn from the Khoja's [23] Arabic Tagger. Following this work, Habash and Rambow [18] use SVM to perform a real POS tagging instead of a POS classification. In a first step, a morphological analyzer is used to produce all possible morphological forms of a word, and then a classifier is used in a second step to choose an appropriate solution from all propositions given by the analyzer. Habash and Rambow [18] use the same tagset to compare their work with the previous one and report an accuracy of 97.6%. Tlili-Guiassa [29] uses a combination of rule based and memory-based learning method for tagging Arabic words. A tagset derived from Khoja's [23] tagger is used and a performance of 85% was reported. Daelemans *et al*. [8] use also memory-based learning method for POS tagging. They use the segmentation given in the Arabic TreeBank (ATB), and report an overall accuracy of 91.5%. AlGahtani *et al*. [2] use transformation-based learning as given in the Brill tagger [6] for Arabic POS tagging with segment-based tags. They use the segmentation in ATB for training and the segmentation performed by Buckwalter Arabic Morphological Analyzer (BAMA) [7] for testing. To deal with the multiple solutions generated by BAMA, AlGahtani *et al*. [2] use bigram information from the morphological analyses to select the preferred solution, which is then passed to the Brill tagger. They evaluate their approach on the whole ATB and reach an accuracy of 96.9%. Dukes *et al*. [15] propose a syntactic annotation of the Quranic text using the Quranic corpus[2] developed at the Leeds University.

As a conclusion, we notice that many works with different techniques and approaches are being developed for Arabic POS tagging. However, these works are mainly directed to MSA and might not be suitable for CA, used in the Holy Quran and in classical texts. Moreover, almost all of these taggers are based on tagsets derived from English which may not also be appropriate for an accurate description of CA. For these reasons, works have been started on this direction by moving towards an appropriate POS tagger for classical texts, especially for the Holy Quran [16] and its related sciences. In this paper, we describe a more elaborated approach based on Parallel Hidden Markov Models (HMM).

3. Proposed Approach

3.1. Needs and Motivation

Due to the nature of the Arabic language, which is a highly inflectional language, the traditional classification-into nouns, verbs and particles-may not be enough for detailed description. Detailed tagsets describing in-depth morphological and grammatical features (e.g. all sub-categories, person, number, gender, case, mood, etc.), are generally considered more appropriate [4]. However, large or fine-grain tagsets may cause problems for automatic tagging, since:

- Some words can change grammatical tag depending of function and context [4].
- The time complexity of the tagging process may be largely increased.

New approaches were recently proposed to deal with fine-grain tagsets in highly inflectional languages, such as German and Arabic for examples. Schmid and Laws [28] use HMM POS tagger with a fine-grain tagset for German and Czech. They split the POS into attribute vectors and estimate the conditional probabilities of the attribute with decision trees. Sawalha and Atwell [27] designed a detailed morphological feature tagset that captures long-established traditional morphological features of Arabic.

In our vision, it would be better to design separate relatively compact tagsets based on different classifications of morphological and grammatical features of the Arabic language and then to run POS tagger on each of them to drive available information. Preliminary results of this approach highlight many benefits [21].

[1] http://corpus.byu.edu/bnc/

[2] http://corpus.quran.com/

3.2. Proposed Tagsets

The idea behind our choice of working on different tagsets comes from the fact that Arabic words may have different classifications depending on the type of considerations we want to deal with. For example, the same word might be classified in a given category for a specific linguistic characterization, but it may belong to another category when we consider it from another characterization point of view. These different characterizations are in many cases important to be known and thus required for an accurate tagging. However, it is not appropriate to combine them as they might be linguistically inhomogeneous. Let's take the following example to clarify this concept: We often find in Arabic, particles that have several functions, such as:

إنَّ : حرف نصب وتوكيد

(Particle which introduce the verb in Accusative Case, and expresses the affirmation)

لم : حرف جزم ونفي وقلب

(Particle which introduce the verb in Jussive Case, and expresses the negation, and returns the present to past)

أمّا : حرف شرط وتوكيد وتفصيل

(Particle which introduce the sentence in Condition Case, and expresses the affirmation, and used for detailing)

From these examples, it becomes clear that we can create two classifications for particles: one for the grammatical function (العمل) and another for the meaning function (المعنى), as follow:

العمل : {نصب ، جر ، جزم ، لا يعمل}

المعنى : {توكيد ، نفي ، قلب ، شرط ...}

So, a particle like إنَّ will have: the tag نصب in the first tagset, and the tag توكيد in the second one.

This concept, just explained for particles, can be extended to all Arabic terms. Indeed, we can create several classifications based on different kinds of characterizations or features. For example, a word (noun, verb or particle) can belong to the following classifications (see Figure 1):

{مركب ، بسيط}, {مشتق ، جامد}, {مبني ، معرب},

{مرفوع ، منصوب ، مجرور ، مجزوم},...

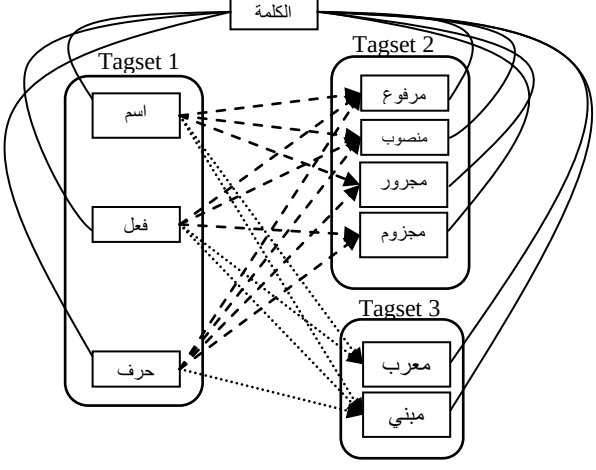

Figure 1. Example of proposed tagsets.

Following these concepts, several tagsets can be created and used to tag words on different contexts.

To simplify the proof of concept of this approach, we will be limited in this paper to the two following tagsets: {*V* (فعل), *N* (اسم), *P* (حرف)} and {*Marf* (مرفوع), *Manss* (منصوب), *Magr* (مجرور), *Majz* (مجزوم)}. In addition, we considered the following tags:

Manss_Magr (can be either *Manss* or *Magr*) for three cases: the feminine plural ending with *kasra*: ◌ِ, the masculine plural ending with "ينَ" and the dual ending with "يْنِ".

Marf_Magr (can be either *Marf* or *Magr*) and *x* for words ending with vowels, but are not used for our case as we will see.

Start for marking the beginning of a sequence; it corresponds to a special character created to separate sequences: <s>.

So, the final tagsets that we will use are presented in Table 1.

Table 1. Used tagsets.

Tagset1	**Tagset2**
V (فعل)	Marf (مرفوع)
N (اسم)	Manss (منصوب)
P (حرف)	Magr (مجرور)
Start	Majz (مجزوم)
	Manss_Magr (منصوب أو مجرور)
	Start

3.3. Parallel Approach for Tagging

Tracking a word through several contexts will give more information about this word and will help accurately tagging it. For example, knowing the morphological function of a word will give an idea about its grammatical function by referring to linguistic rules or statistical approach (as for our case).

By looking back to Figure 1, we can see that certain tags are mutually exclusive. For example, a Noun (اسم) cannot be *Majz* (مجزوم), and a Particle (حرف) cannot be Case-marked (معرب). In fact, the majority of names in Arabic are usually Case-marked, contrary to verbs. So, we can deduce some kind of relations between the different proposed tagsets, and thus identifying a tag from a given tagset may help to infer a corresponding one from another tagset [21].

Thus, our approach can be formulated as follows:

- First, identifying words into two different contexts via parallel part of speech taggers, which use two different tagsets;
- Second, in the case of a low estimation of a tag, use the relation between the two tagsets to help extracting the appropriate tag.

3.4. Proposed Application of HMM in POS Tagging

In the case of POS tagging, the observed sequence in HMM is the words to tag, and hidden sequence is the tags. The model could be represented as in Figure 2.

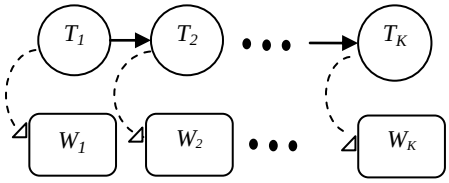

Figure 2. HMM representation for the POS tagging.

Two HMMs will be used in this work to build our parallel tagger as mentioned before. The proposed models can be then presented as in Figure 3.

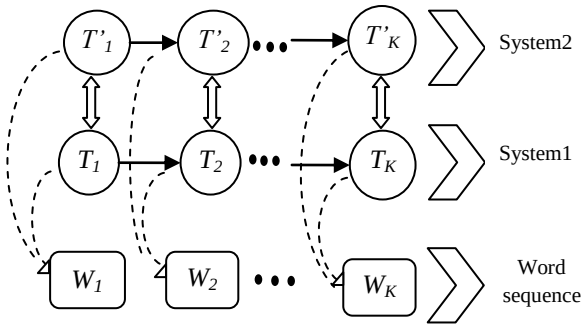

Figure 3. Proposed models representation.

The topologies of these models are depicted in Figure 4 and 5.

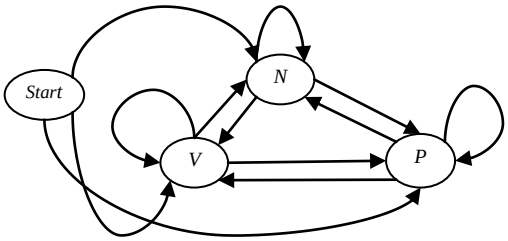

Figure 4. Topology of the first HMM.

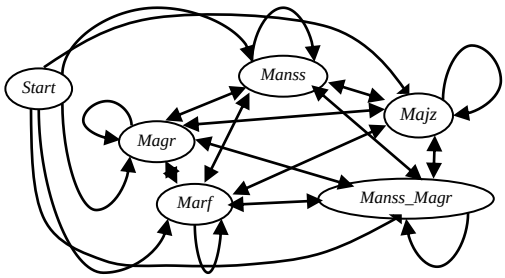

Figure 5. Topology of the second HMM.

In the Viterbi algorithm, we calculate, for each length *k* of words, the sequence of tags that maximizes the joint probability $P(T_1 \ldots T_k , W_1 \ldots W_k)$ - which is saved in a matrix δ. Therefore, the higher this value is, the larger we are sure of the choice of tag and vice versa.

For that, our approach is to add another tagger in parallel (for the same sequence of words) to act as an auxiliary. Before extracting results in the main tagger, we compare for every word the two values of δ (in both taggers); if δ value in the auxiliary tagger is higher, its corresponding tag will be the reference for redefining the tag in the main tagger, and then all the tags that precede it will change (as we will explain in Figure 6).

4. Implementation

To better understand how the approach works, we first describe how the Viterbi algorithm is implemented in POS tagging, and then explain our proposed algorithm.

4.1. Viterbi Implementation

For the POS tagging case, the Viterbi algorithm is practically implemented through four steps:

1. Initialization step:

1.1. Initializing the matrices (δ_{ij}) and (ψ_{ij}):

$$\delta_{ij} = 0 \; ; \; \psi_{ij} = -1 \; , \; 1 \leq i \leq N \; , \; 1 \leq j \leq K$$

Given *N* the tagset's length (number of possible tags) and *K* the word sequence length (number of words to tag).

1.2. Initializing first column of the matrices (δ_{ij}) and (ψ_{ij}):

$$\delta_1(T_i) = b_i(W_1) \times A(st,i)$$
$$\psi_{i1} = st$$

With:

- T_i: the tag having the index *i* in the tagset;
- W_i: the word in the observed sequence at the position (time) *i*;
- *A(i,j)*: the transition probability is the probability that the tag indexed *i* is followed by the tag indexed *j*:

$$A(i,j) = P(T_t = j | T_{t-1} = i)$$

- $b_i(j)$: the emission probability is the probability that the word indexed *j* is tagged (emitted) bay the tag indexed *i*:

$$B(i,j) = b_i(j) = P(W_t = j | T_t = i)$$

- *st*: the index of the state *Start*.

2. Recursion step:

$$\delta_j(T_i) = b_i(W_j) \times max_{0 \leq q \leq N} \{A(q,i) \times \delta_{j-1}(T_q)\}$$

3. Termination step:

$$\delta_K(T_i) = max_{0 \leq q \leq N} \{A(q,i) \times \delta_{K-1}(T_q)\}$$
$$\psi_{iK} = arg(max_{0 \leq q \leq N} \{A(q,i) \times \delta_{K-1}(T_q)\})$$

4. Extraction of the best path (R_i):

$$R_K = arg(max_{1 \leq i \leq N} \{\delta_K(T_i)\})$$

And for *k=K-1* down-to *k=1*:

$$R_k = \psi_{R(k+1),k+1}$$

4.2. Proposed Algorithm

After extracting the best path in both HMM1 and HMM2, we will have two matrices: respectively (δ_{ij})

and (δ'_{ij}), and two tag sequences as results: R and R'. Thus, for each word W_i we compare the corresponding $\delta_j(T_i)$ with $\delta'_{j'}(T'_i)$, if it is the lowest we change the tag T_i to another tag based on the linking matrix $L(i,j,k)$ -as we will explain in the experimentations part; the tag sequence that precede it will be also changed (see Figure 6).

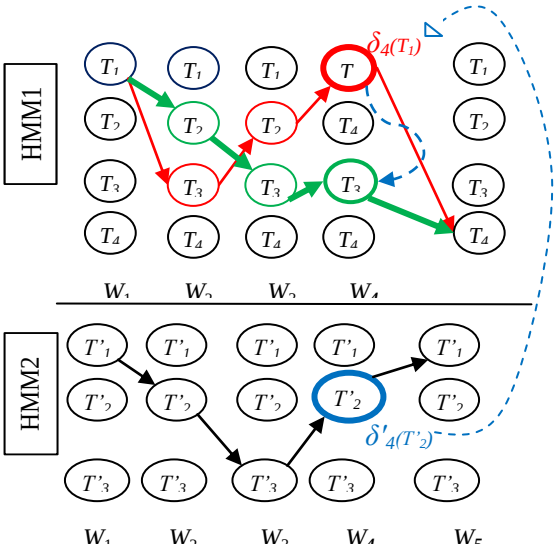

Figure 6. Model process example.

Practically, the change of tags is done in the fourth step of Viterbi algorithm (Extraction of the best path). In fact, for each node (i,j) in the Trellis, ψ_{ij} points to the previous tag (to draw the best path ending in the node (i,j)). So, if we want to change a resulting tag in extraction iteration, we save the new tag in the result sequence, so we can base on it in the next iteration. These steps are gathered in the proposed model algorithm as in Figure 7.

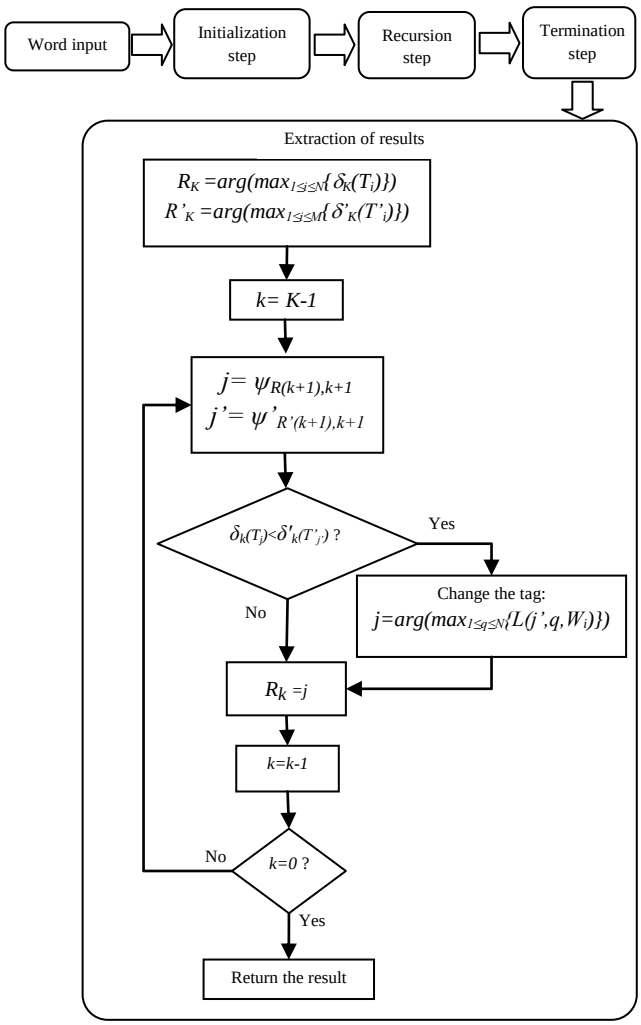

Figure 7. Proposed approach algorithm.

5. Experimentations

Before going to the evaluation, we have to talk about the corpus we used and to explain how it was prepared. The size of the corpus and its nature are very important issues for the evaluation of our proposed approach.

5.1. Corpus Importance and Availability

Corpora represent the heart of statistical approaches for natural language processing. Many specific -and/or general- purposes corpora were developed for several languages around the world (e.g. the Corpus Of Contemporary American English (COCA) [13], American National Corpus (ANC), British National Corpus (BNC) [12], Corpus for Spanish [11], Corpus for Portugal [10], etc.). For the Arabic language, unfortunately, corpora are still limited in terms of size, coverage, and availability compared to European languages. Despite the availability of some general Arabic corpora (e.g. CLARA Corpus, Al-Hayat Corpus, An-Nahar Corpus, Arabic Gigaword Corpus, etc. [1]), tagged Arabic corpora are still difficult to be obtained for research purposes.

5.2. Overview Of The Used Corpus: Nemlar

To implement the approach we looked for a corpus, large enough and rich in information in order to extract different kinds of tagset; we chosen the Nemlar corpus. Nemlar corpus contains annotated written corpus of MSA derived from the European NEMLAR (*Network for Euro-Mediterranean Language Resources*) project [25]. It counts about 500,000 words from 13 different domains. The corpus is organized in different formats (from different points of views) with 489 text files each:

- *Raw corpus*: diacriticized Arabic texts.
- *Fully vowelized corpus*: same texts with pronunciation information.
- *Lexically analyzed corpus*: lexical analysis, including the type of word, prefix, root, pattern and suffix.
- *POS tagged corpus*: (used in this work) provides additional information related to morphosyntactic analysis by associating to each word a tags sequence containing prefix tags (or "NullPrefix" if not exist), stem tags, and suffix tags ("NullSuffix" if not exist).

Example:

{(اَلْأَلْسِنَةِ) Definit Noun Plural Femin Single}
{(يَخُو@ضُو@ا×) Present Active Verb Manss_*Majz* SubjPro}

5.3. Proposed Adaptation of the Corpus

First, a work on restructuring the Nemlar corpus content is made to optimize the learning process [22]. To simplify the experimentation, we were limited to attributing a tag for the whole word, without considering its segmentation (prefixes, stem and suffixes). To apply our proposed approach we need to create a doubly tagged corpus, where each word will have two tags taken from our previously described tagsets (see Table 1).

- *Creation of the corpus based on Tagset1*:

The following operations are applied to create a new tagged corpus using Tagset1 (see Table 2):

- If we find the tag "noun" in the sequence of tags belonging to a word, we give it the tag N (noun).
- If we find the tag "verb" in the sequence of tags belonging to a word, we give it the tag V (verb).
- Otherwise, we give it the tag P (particle).

Table 2. Simple corpus example.

Word	Tag
أوضح	*V*
أن	*P*
الشعب	*N*
البربري	*N*
يعتز	*V*
بعروبته	*N*
وإسلامه	*N*

- *Creation of the corpus based on Tagset2*:

As previously seen, Tagset2 will be created based on the final diacritics (*Marf*, *Manss*, …), which will be called the diacritical status of word. Nevertheless, in Arabic, diacritical status is not always defined from the final diacritical marks. Indeed, there are certain categories of words in which the diacritical status manifests in letters (such as in dual nouns) or does not manifest at all (for words with vowels at the end). So, for each word of Nemlar corpus (4th format), we proceeded as follows:

- If we have a tag (*Manss*, *Magr*, …) in the word's tag sequence, we consider it as its tag (the second tag).
- If it is a word without suffix, we check the last diacritic:
  - if it equals to " ً" or " ّ ً", we give the tag *Manss*;
  - if it equals to " ُ " or " ّ ُ", we give the tag *Marf*;
  - if it equals to " ِ " or " ّ ِ", we give the tag *Magr*;
  - if it equals to " ْ ", we give the tag *Majz*.
- If it is a word with suffix:
  - if it has the suffix "ي" (ياء النسبة) (having the tag "Reladj" in the tag sequence) or "ـة" (having the tag "Femin+Single" in the tag sequence), we check the last diacritic as above;
  - if it is in the plural masculine form (جمع المذكر السالم) it has the tag *Marf* for the suffix "ون" and *Manss_Magr* for the suffix "ين" in the sequence tags;
  - if it is in the dual form (المثنى) it has the tag *Marf* for the suffix "ـا" and *Manss_Magr* for the suffix "ـي" in the sequence tags;
  - we added that for the plural feminine form (جمع المؤنث السالم) with the suffix "ـات" (having the tag "Plural" and "Femin" in the tag sequence), we check the latest diacritic:
    - if it equals to " ُ " or " ّ ُ", we give the tag *Marf*;
    - if it equals to " ِ " or " ّ ِ", we give the tag *Manss_Magr*[3];
  - if it has another suffix, we check the last diacritic, as seen in the first case.
- We have also encountered a difficulty with words ending with the vowels: "ا", "و" or "ي" (called *weak letters* "حروف العلة"), where diacritics are often not displayed which causes an ambiguity in the determination of the diacritic states of these words.

[3]If a feminine plural ended with the suffix "ـات" and has in the end the diacritic Kasra (" ِ" or " ّ ِ") it is either in oblique *Magr* or accusative *Manss* [20].

To deal with these special cases, we have firstly considered the tag *Marf_Magr* for words ending with vowels "و" or "ي" as "القَاضِي" (for nouns) and "يَدْعُو", "يَرْمِي" (for verbs)[4]; Actually, the state *Manss* is manifested by the diacritic " ً " on these two vowels, such as "رَأَيْتُ الْقَاضِيَ", "حَتَّى يَدْعُوَ", while if there is no diacritic on this vowels the word state will be either *Marf*, such as "جَاءَ الْقَاضِي" and " زَيْدٌ يَدْعُو", or *Magr* such as "مَرَرْتُ بِالْقَاضِي". So we created for these cases the tag *Marf_Magr* (*Marf* or *Magr*). For the words ending with the vowel **"ا"**, we have considered the tag *x* since this vowel is always without diacritic. This is what we did in first time. However, to simplify the experimentation and focus in the main approach, we give the tag *Manss* for words ending with "ا", *Marf* for those ending with "و" and *Magr* for those ending with "ي".Based on the above algorithms for Tagset1 and Tagset2, we built a program: to extract data from Nemlar corpus and to transform them to our own formatting in order to build an appropriate corpus suitable for the proposed tagging approach (see Table 3).

Table 3. Parallel corpus example.

Word	Tag1	Tag2
أوضح	V	Manss
أن	P	Manss
الشعب	N	Manss
البربري	N	Manss
يعتز	V	Marf
بعروبته	N	Magr
وإسلامه	N	Magr

And this is the statistics of the tags in the used corpus see Table 4.

Table 4. Tags statistics.

Tag	Frequency
Magr	193131
Majz	46028
Manss	137358
Manss_Magr	16699
Marf	81793
Start	19016
P	108665
N	306317
V	60027

5.4. Training the HMMs

The HMM learning phase is aiming to estimate its parameters from statistics related to the tagged corpus as follows:

- The set of states *S* is the set of possible tags in the corpus (for the first model it is {*Start*, *N*, *V*, *P*}, and for the second one it is {*Start*, *Manss*, *Marf*, *Magr*, *Majz*, *Manss_Magr*}).
- The transition matrix *A(i,j)* represents the transition probability from tag T_i to tag T_j; it can be estimated from the corpus as follows:
  Count (tag T_i is followed by tag T_j) / Count (tag T_i)
  For more accuracy, we used the trigram model *A(i,j,k)*:
  Count (tag T_i is followed by tag T_j and T_j is followed by tag T_k) / Count(tag T_i is followed by tag T_j)
  We integrated also a smoothing technique with the deleted interpolation [5].
- The emission matrix *B(i,j)* represents the probability that a tag T_i emit the word W_j; it can also be estimated from the corpus as follows:
  Count(word W_j is tagged by tag T_i) / Count(tag T_i)
- The initial probability is fixed to *1* for the tag *Start*: $\pi(Start)=1$, and *0* for the other tags to enforce the model starting from this state.

For our approach, two parallel tagging systems will be run on the same sequence of words using two different tagsets (Tagset1 and Tagset2). So, we need to build two HMMs based on a corpus tagged with these two tagsets.

The parameters of the first HMM (HMM1) are extracted from the version of the corpus tagged with Tagset1 (described above); while the second HMM (HMM2) is built from version of the corpus tagged with Tagset2 (described above).

5.5. Linking Matrix

To estimate a tag in the Tagset2 from another tag in the Tagset1 we make use the relation between the two tagsets by calculating intra-linking probability between HMM1 tags and HMM2 tags; we name it the *Linking matrix* (*L*). It represents the probability that a word tagged T_i (in HMM1) is tagged T'_j (in HMM2). It is calculated first time as follow:

$$L(i,j)= P(T'_j \mid T_i)$$

$$=Count(\text{tagging the same word with tag } T_i \text{ and tag } T'_j) / Count(\text{tag } T_i)$$

But it did not give good results. In fact, for example, if the most adequate tag for a particle (tag *P*) is the tag *Majz* so every time we find a word that we are more sure that is a particle it will have always the tag *Majz*, which is not reasonable because we have several particles with tags other than *Majz* (أنَّ *Manss*, فِي *Magr*, ...). So, in addition to the tag of the other system, we must also take into consideration the word itself. Thus the linking matrix should be three-dimensional by adding a third column for the words, and it is calculated as follow:

$$L(i,j,k)= P(T'_j \mid T_i, W_k)$$

$$=Count(\text{tagging the word } W_k \text{ with tag } T_i \text{ and tag } T'_j) / Count(\text{tagging the word } W_k \text{ with tag } T_i)$$

The formulation above can be schematized as in Figure 8.

[4]There are no Arabic Case-marked nouns ending with "و" (without diacritic) [19].

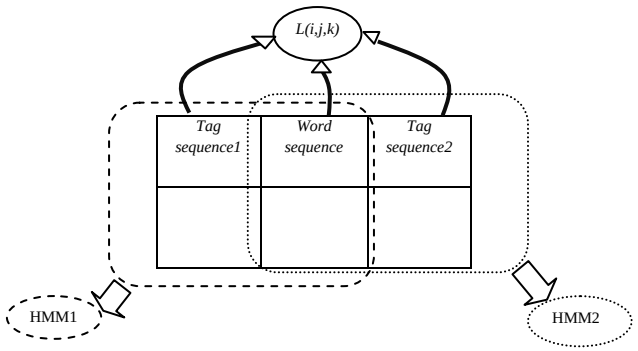

Figure 8. Extracting information from the tagged corpus.

So, the actions in our proposed model are (see Figure 9):

- From the Word sequence, the HMM1 returns Tag sequence1 belonging to the Tagset1;
- From the Word sequence, the HMM2 returns Tag sequence2 belonging to the Tagset2;
- Given a tag belonging to the Tagset1 and the tagged word, the linking matrix *L(i,j,k)* returns a tag belonging to the Tagset2.

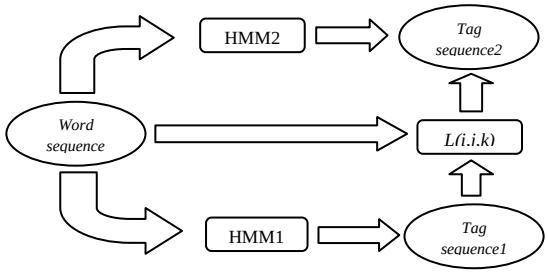

Figure 9. Actions in the proposed model.

Example:
Consider this sentence: عمر زيد طويلا.
S_1={*N*, *V*, *P*, *Start*} and S_2={*Manss*, *Marf*, *Magr*, *Majz*, *Manss_Magr*, *Start*}
If we tag the word عمر with *Manss* (عُمِّر or عَمَّر), the meaning will be: "Zaid had long age".
If we tag it *Marf* (عُمَر or عُمُر), we will have wrong meanings: "*Omar Zaid long" or "*Zaid Age long".
And it may happen that -in the tagged corpus- the word عمر is frequently tagged with *Marf*, so the tag *Marf* has a high emission probability for عمر, and the system may select it (see Figure 10).

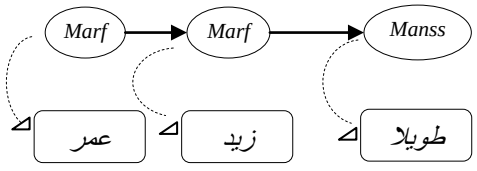

Figure 10. Simple POS tagging of عمر زيد طويلا.

But after introducing the second tagger, the result would not be the same. In fact, after referring to the linking matrix, we find that the most appropriate tag is *Manss* (see Figure 11).

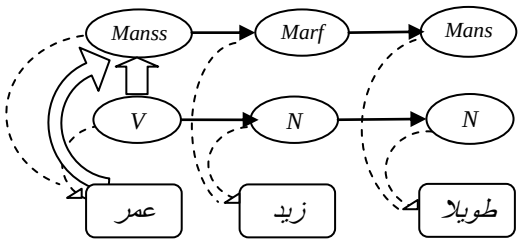

Figure 11. Parallel POS tagging of عمر زيد طويلا.

5.6. Results of the Model Testing

All described processes were implemented in a Java project consisting of various classes according to the concerned process (corpus operations, matrices creation, Viterbi algorithm, etc.). We tested the model on a set of Arabic sentences with different lengths. It was taken from various web pages (Islamic, politic, sportive, etc.) and from some Shamela books. If necessary, we introduced some changes to have all words belonging to the corpus lexicon, and in the same time meaningful sentences. For each sentence, we apply simple and double tagging and the results are extracted from the Java application output (see application output example in Figure 12) and compared in both cases, given the initial tagger (on which the tag changes are made) is Tagger2 with Tagset2: {*Marf*, *Manss*, *Magr*, *Majz*, *Manss_Magr*, *Start*} and the auxiliary tagger is Tagger1 with Tagset1: {*N*, *V*, *P*, *Start*}. The comparison and evaluations-that we will see later - were made by a person with a good level in Arabic grammar.

For each word, before parallel-HMM, we manually evaluate the tag result of Tagger2, as well as for Tagger1. Then, after executing the parallel-HMM, we evaluate the changes (if they exist) between simple and parallel-HMM-in Tagger2-. Each tag change is evaluated manually: if it is a positive change, we add +1 to the "changes result" of the sentence, otherwise we add -1. The experimentation was conducted on 40 sentences, where results are presented on graphs. For each sentence, the accuracy rate (percentage of correct tags) of the Tagger1 and Tagger2 are presented independently (see Figure 13). The "changes result" and the number of tag changes between simple and parallel-HMM are also presented (see Figure 14).

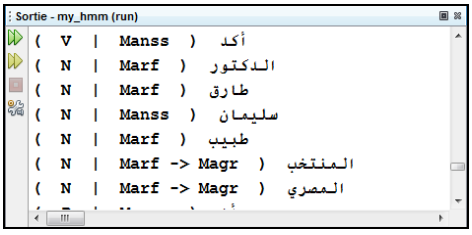

Figure 12. Experimentation result example.

For example, in Figure 12, we can see two changes in Tagger2 for the two words المنتخب (*Marf* ➔ Magr) and المصري (*Marf* ➔ Magr), so the "changes count"

will be incremented by two. Since the two changes are positives, we add +2 to the "changes result". We then collect these parameters for the 40 sentences which are presented in Figure 14.

The Table 5 summarizes the experimentation statistics.

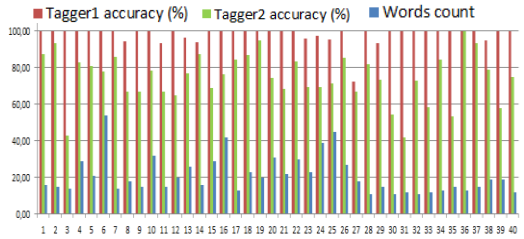

Figure 13. Accuracy of Tagger1 and Tagger2 and word number of each sentence.

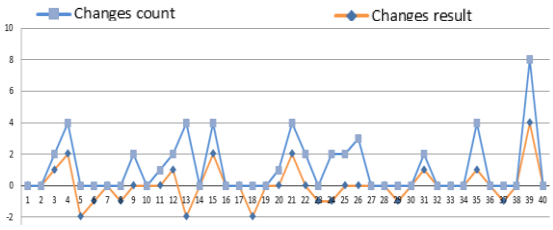

Figure 14. Number of tag changes (*changes count*) in Tagger2 and evaluation of these changes (*changes result*) for each sentence.

Table 5. General statistics of the experimentation.

Number of sentences	40
Number of words	845
Average accuracy of Tagger1	98.22%
Average accuracy of Tagger2	75.12%
Average accuracy of Tagger2 with Parallel HMM	75.38%

5.7. Result Discussion

From the graph in Figure 14, we can see that there are positive and negative results, and that these results do not depend on the length of the input sentence. So after this work, we can say that we got our hands on positive results that invoke a deep study to improve the application accuracy. We also mention in this regard, that for more credible results it was necessary to apply the approach on a manually tagged corpus with the two tagsets, while in our attempts we worked on a tagset automatically extracted therefore less precise. Indeed, although the rules of extraction of the second tagset - described above- apply to the majority of Arabic words, there were significant cases that do not respond to these rules. For example, the five nouns[5] (الأسماء الخمسة) and the five verbs[6] (الأفعال الخمسة) are from words whose diacritic state is manifested by letters [20]. However, according to the extraction rules described above, it will not have the right tags, because it will be treated according to the last diacritic, and it is not obvious to define it's states automatically (especially for the five verbs whose forms vary greatly depending on the verbs and amendments made to the vowels - called in Arabic "الإعلال"); in addition to words ended with a vowel that we cannot have the exact diacritic state[7].

6. Conclusions and Future Works

This paper represents a novelty in the HMM domain by combining two parallel sets of hidden states in order to enhance the model. Moreover, it opens up the way for an extension to the use of multi HMMs which could revolutionize research in this domain. For POS Tagging, it gives a new vision for the traditional method. By working with several tagsets, we can thus divide the tagging process into modules, and choose the modules according to the processing context. In addition, it gives more importance to the Arabic language, by introducing more details, in order to have best results even with low probability cases existing in the traditional Arabic. Also, by defining several aspects of the Arabic word, we expect that this work could be a first step of a traditional grammar analyzer for Arabic words (الإعراب الآلي).

However, this new approach still needs to be tested on a large manually doubly tagged corpus to measure its real impact and performance - as we previously explained in the Result discussion section.

To improve the performance of this approach, we are planning to work on different tracks:

- *Enlarging the corpus*: the whole Nemlar corpus may be transformed to a doubly tagged corpus, based on our mentioned tagsets, by doing automatic creation using the proposed algorithms followed by manual verification and validation. This semi-automatic approach will help creating a big size corpus using different tagsets with reduced cost in both time and effort.
- *Parallel implementation of the HMMs*: one big concern about our proposed tagging approach is its time complexity compared to the classical use of a single HMM-based tagging. To overcome this deficiency, we are proposing to use a multithreading approach to implement the taggers

[5]five nouns {"ذُو"، "فُو"، "حَمٌ"، "أَخٌ"، "أَبٌ"}; and there are others that add a sixth name: "هَنٌ" [19].

[6]verbs having this patterns:
{"يَفْعَلَانِ"، "تَفْعَلَانِ"، "يَفْعَلُونَ"، "تَفْعَلُونَ"، "تَفْعَلِينَ"}; and it is better to call it "the five examples" (الأمثلة الخمسة) [19].

[7]We mention that, in this paper, we have not compared our method with other existing methods. Indeed, the current study was focused on the presentation of a new tagging approach with a concrete application of it. This work is not a new tagging system that needs to be compared with existing systems. So we just compared the results before and after the parallel tagging to evaluate the approach.

in order to be run simultaneously. With this approach, the time complexity of the whole model will be in the same order of using a single HMM tagger. This will have a great importance as we can enlarge the parallel tagger to combine many HMMs when needed without incurring a significant complexity.

References

[1] Alansary S., Nagi M., and Adly N., "Building an International Corpus of Arabic: Progress of Compilation Stage," *in Proceedings of the 7th International Conference on Language Engineering*, Cairo, pp. 1-30, 2007.

[2] AlGahtani S., Black W., and McNaught J., "Arabic Part-of-Speech-Tagging using Transformation-based Learning," *in Proceedings of the 2nd International Conference on Arabic Language Resources and Tools*, Cairo, pp. 66-70, 2009.

[3] Al-Taani A. and Al-Rub S., "A Rule-Based Approach for Tagging Non-Vocalized Arabic Words," *Information Technology*, vol. 6, no. 3, pp. 320-328, 2009.

[4] Atwell E., *Development of Tag Sets for Part-of-Speech Tagging, In Ludeling A., and Kyto M., Corpus Linguistics*, Walter de Gruyter, 2008.

[5] Brants T., "TnT: a Statistical Part-of-Speech Tagger," *in Proceedings of the 6th Conference on Applied Natural Language Processing*, Washington, pp. 224-231, 2000.

[6] Brill E., "A Report of Recent Progress in Transformation-based Error-Driven Learning," *in Proceedings of the Workshop on Human Language Technology*, Plainsboro, pp. 256-261, 1994.

[7] Buckwalter T., *Arabic Morphological Analyzer Version 2.0*, Linguistic Data Consortium, 2004.

[8] Daelemans W., Zavrel J., Van der Sloot K., and Van den Bosch A., TiMBL: "Tilburg Memory Based Learner," Technical Report Induction of Linguistic Knowledge, 2007.

[9] Dale R., Moisl H., and Somers H., *Handbook of Natural Language Processing*, CRC Press, 2002.

[10] Davies M. and Michael F., Corpus do Português: 45 million words, 1300s-1900s, http://www.corpusdoportugues.org, Last Visited 2014.

[11] Davies M., Corpus del Español: 100 million words, 1200s-1900s, http://www.corpusdelespanol.org, Last Visited 2014.

[12] Davies M., British National Corpus from Oxford University Press, http://corpus.byu.edu/bnc, Last Visited 2014.

[13] Davies M., Corpus of Contemporary American English: 425 million words, 1990-present, http://corpus.byu.edu/coca, Last Visited 2014.

[14] Diab M., Hacioglu K., and Jurafsky D., "Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks," *in Proceedings of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Boston, pp. 149-152, 2004.

[15] Dukes K., Atwell E., and Habash N., "Supervised Collaboration for Syntactic Annotation of Quranic Arabic," *Language Resources and Evaluation Journal*, vol. 47, no. 1, pp. 33-62, 2013.

[16] Elhadj Y., "Statistical Part-of-Speech Tagger for Traditional Arabic Texts," *Computer Science*, vol. 5, no. 11, pp. 794-800, 2009.

[17] El-Jihad A., Yousfi A., and Aouagh S., "Morpho-Syntactic Tagging System Based on the Patterns Words for Arabic Texts," *Information Technology*, vol. 8, no. 4, pp. 350-354, 2011.

[18] Habash N. and Rambow O., "Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in one Fell Swoop," *in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Michigan, pp. 573-580, 2005.

[19] Ibn Aqil, *Charh Ibn Aqil ala Alfiati Ibn Malik*, Al Maktaba al-Asria, 2002.

[20] Ibn Hichâm, *Charh Qatru Nadawa Ballu as-Sada*, Al Maktaba al-Asria, 1994.

[21] Kadim A., Lazrek A., and Elhadj Y., "Dual Hidden Markov Model-New Approach for an Accurate Arabic Part-of-Speech Tagging," *International Journal of Computational and General Linguistics*, vol. 5, pp. 57-74, 2013.

[22] Kadim A., Lazrek A., and Elhadj Y., "The Nemlar Arabic Written Corpus, a new Optimized version and Proposal of an Arabic POS Tagger Based on It," *in Proceedings of 17th International Conference on Intelligent Text Processing and Arabic Computational Linguistic, Co-located with CICLing*, Konia, 2016.

[23] Khoja S., Arabic Part-of-Speech Tagger, Carnegie Mellon University, Pennsylvania, 2001.

[24] Kübler S. and Mohamed E., "Part of Speech Tagging for Arabic," *Natural Language Engineering*, vol. 18, no. 4, pp. 521-548, 2012.

[25] Maegaard B., Choukri K., Mokbel C., and Yaseen M., *Language Technology for Arabic*, Nemlar, 2005.

[26] Santorini B., "Part-of-Speech Tagging Guidelines for the Penn Treebank Project,"

Technical Report Department of Computer and Information Science, 1990.
[27] Sawalha M. and Atwell E., "Fine-Grain Morphological Analyzer and Part-of-Speech Tagger for Arabic Text," *in Proceedings of the Language Resource and Evaluation Conference*, Valleta, pp. 1258-1265, 2010.
[28] Schmid H. and Laws F., "Estimation of Conditional Probabilities with Decision Trees and an Application to Fine-Grained POS Tagging," *in Proceedings of the 22nd International Conference on Computational Linguistics*, Manchester, pp. 777-784, 2008.
[29] Tlili-Guiassa Y., "Hybrid Method for Tagging Arabic Text," *Computer Science*, vol. 2, no. 3, pp. 245-248, 2006.
[30] Valli A. and Véronis J., *Etiquetage Grammatical Des Corpus De Parole: Problèmeset perspectives*, Revue française de linguistique appliquée, 1999.
[31] Zribi C., Torjmen A., and Ahmed M., "A Multi-Agent System for POS-Tagging Vocalized Arabic Texts," *The International Arab Journal of Information Technology*, vol. 4, no. 4, pp. 322-329, 2007.

Ayoub Kadim has completed preparatory classes in 2006. He had state engineering diploma from INPT, Rabat, Morocco in 2009. (In telecommunication and computer science). He got his PHD in computer science at Cadi Ayyad University, Morocco in 2017. He works on Arabic Natural Language Processing and computerization of Quranic grammar, in which he had several publications and international conference communications.

Azzeddine Lazrek is full Professor in Computer Science at Cadi Ayyad University in Marrakesh at Morocco. He holds a Ph.D. in Computer Science from Lorraine Polytechnic National Institute in France, awarded in 1988, and a State Doctorate awarded in 2002. Prof. Lazrek has more than 20 years of work in the field of multimedia communications for multilingual electronic documents, in the digital domain. Research interests include electronic publishing, digital typography, Arabic language processing and history of sciences. He has lead a multilingual e-document composition project with some international organizations: the King Abdulaziz City for Science and Technology and the King Fahd Glorious Quran Printing Complex in Saudi Arabia, the Education Ministry in Libya, the Research Group in Intelligent Machine in Tunisia, Support for Research Activities in Computer Science and Mathematics in Africa in France, the Institute of Mathematics and its Applications, the American Mathematics Society and the TEX Users Group in the United States. He is an expert in the World Wide Web Consortium (W3C), and adviser and consultant to the Unicode Consortium. He was director of Formation and search Unit Ph.D. Informatics, and deputy director of Information systems engineering Laboratory, and supervisor of the research group in the Processing of multilingual e-document, and a former responsible of the research team in Information systems and communications networks. He has participated in some international scientific manifestations at Egypt, Germany, Tunisia, Jordan, France, Greece, Qatar, USA, Saudi Arabia and Morocco. He supervises some PhD students in informatics. He also leads and teaches modules at both BSc and MSc levels in computer science and software engineering. He contributes to scientific journals and is a member of several national and international scientific associations.