A Hidden Markov Model –Based POS Tagger for Arabic

Fatma Al Shamsi, Ahmed Guessoum
Department of Computer Science – University of Sharjah
P.O. Box 27272 – Sharjah – UAE
Emails : {fshamsi, guessoum}@sharjah.ac.ae

Abstract

This paper presents a Part-of-Speech (POS) Tagger for Arabic. The POS tagger resolves Arabic text POS tagging ambiguity through the use of a statistical language model developed from Arabic corpus as a Hidden Markov Model (HMM). The paper presents the characteristics of the Arabic language and the POS tag set that has been selected. It then introduces the methodology followed to develop the HMM for Arabic. The proposed HMM POS tagger has been tested and has achieved a state-of-the-art performance of 97%.

Keywords : Arabic Morphological Analysis, Part-of-Speech Tagger, Stemmer, Hidden Markov Model, Statistical Language Model.

1. Introduction

Interest in the Arabic language is growing fast. Despite the fact that Arabic is the language of almost 300 million people and the language used by 1.2 billion Muslims in religious ceremonies (prayers, holy book, etc.), natural language processing tools for Arabic are yet to achieve the quality and robustness levels that would support the need for and the growing interest in the language.

Arabic uses diacritics to disambiguate words. There are four diacritics in Arabic, giving short sounds : the fatHa is a character put on the top of a letter to give the "a" sound (as in apple) ; the Dhamma is a character put on the top of a letter to give the "u" (as in rudimentary) ; the kasra is put under a letter to give the "i" sound (as in interest) ; whereas the sukuun is a small circle put on the top of a letter to reflect the absence of any of the first three sounds. The presence of diacritics allows lexical disambiguation since many words in Arabic can have the same constituent letters but different pronunciations, thus meanings, depending on the diacritics used. For instance, دَخَلَ (dhakhala[1]) means "he entered" while دَخْلْ (dakhl) means "income". Likewise, كَتَبَ (kataba) means "he wrote", while كُتِبَ (kutiba) means "(it-masculine) was written". Leaving out the diacritics is very common in Modern Standard Arabic. Sometimes, the diacritic used with the last syllable is kept. This, of course, adds a lot of lexical ambiguity. In this case, the reader would use contextual information to lexically disambiguate the words.

POS tagging is the process of assigning a part-of-speech tag such as noun, verb, pronoun, preposition, adverb, adjective or other tags to each word in a sentence. It reflects the word syntactic category based on its context for the purposes of resolving lexical ambiguity

[1] In the sequel, any Arabic sentence will be followed by its transliteration using the Qalam transliteration scheme (as described in **http://eserver.org/langs/qalam.txt**) followed by its English translation.

(Jurafsky and Martin, 2000). POS-tagging is considered as one of the basic tools needed in speech recognition, natural language parsing, information retrieval and information extraction. It is also one of the main tools needed to develop any language corpus. Consider the following Arabic sentence

دخل باسم قبل شاكر / Bassem entered before Shakir

Each word in the above sentence has more than one morphological analysis (see Table 1). The POS tagger is responsible for assigning to each word the most appropriate morphological tag. In Table 1, the correct POS tag for each word is given in bold font. The first word is actually a verb and hence the second word is more likely to be a proper noun instead of an adjective or a preposition that is attached to a noun.

Word	Transliteration	POS Tag	Meaning
دخل	dakhala	**Verb**	(He) entered
	dakhl	Noun	Income
باسم	baasim	**Proper Noun**	Bassem/Bassim
	baasim	Adjective	Smiling
	bi-smi	Preposition +Noun	By/with + name
قبل	qabla	**Preposition**	Before
	qabila	Perfect Verb	(he) accepted
	qabbala	Perfect Verb	(He) kissed
	qubila	Passive form, past tense	(It-masculine) was accepted
شاكر	shaakir	Adjective	Thankful
	shaakir	**Proper Noun**	Shakir

Table 1 : Part-of-Speech tagging for the sentence دخل باسم قبل شاكر

There are two approaches of POS taggers : rule based and trained ones. In the rule-based approach, a knowledge base of rules is developed by linguists to define precisely how and where to assign the various POS tags. In the trainable approach, statistical language models are built, refined and used to POS tag the input text automatically. In this paper, our approach focuses on employing the Arabic phrase structure in resolving the POS ambiguity of Arabic text. By the phrase structure we mean the valid sequence of POS tags that forms the grammatical structure of the noun and the verb phrases. To capture the grammatical structure of the Arabic phrase, we need to build and train statistical language models. Then the trained models are used to POS tag the input text automatically. One of the robust approaches in statistical models is the use of Hidden Markov models (HMM). In this paper, we report on the building and use of an HMM-based POS tagger for Arabic. We have favored Hidden Markov Models over other statistical models for a number of reasons. First, HMM model make use of the history events in assigning the current event some probability value and that suits our approach philosophy. Second, HMM is superior to other models with regard to training speed. Hence, HMM is suitable for applications that have to process large amounts of text.

A lot of research has been done on POS-tagging for English. A rule-based tagger has been developed in (Brill, 1994). Various statistical taggers were built during the last decade, a hidden Markov models were used in (Brants, 2000 ; Garside et al., 1997 ; Cutting et al., 1992 ; Merialdo, 1994). Neural networks have been used in developing POS taggers such as (Olde et al., 1999 ; Marques and Lopes, 1996). There has not been much work done in POS tagging for Arabic. A likely reason is that Arabic is rich in morphology and most of the information for POS tagging is available as inflections or affixes attached to the word.

Recently, a rule-based POS tagger was developed in (Freeman, 2001). A framework of a hybrid Arabic POS tagger has been introduced in (Khoja, 2001) without specifying a particular statistical method. A Support Vector Machine (SVM) based POS tagger was described in (Diab et al., 2004). SVM method estimates a classification function using training data so that error on unseen data is minimized. The classification function returns either +1 if the test data is a member of the class, or -1 if it is not. The SVM tagger is using the Linguistic Data Consortium (LDC) (LDC, 2005) POS tagset, which consists of 24 tags. Also, the input to the SVM tagger has to be a transliterated (written in Latin letters) Arabic text. While we were writing up this paper, we read about an Egyptian dialect HMM POS tagger (Duh and Kirchhoff, 2005) where a performance of a 68.48 % was reported. Similar to the SVM tagger, it uses the LDC tagset. Our HMM POS tagger has achieved a state-of-the art performance of 97 %. The input to our HMM tagger is the original raw Arabic text processed from right to left and hence we are not enforcing any transliteration schema. Moreover, our HMM tagger uses finer grained tag set than the reported above taggers. This finer grained tag set enables our HMM POS tagger to deliver more morpho-syntactic information about the input text.

The lexical characteristics of Arabic and the selected POS tag set are introduced in Section 2. The approach for building the HMM POS tagger is described in Section 3. The training corpus and experimental results are presented in Section 4. Finally, a conclusion and future work are presented in Section 5.

2. Arabic Language Lexical Characteristics and POS Tag Set Description

In the rest of this section, we will present an analysis of the lexical characteristics of Arabic. On the basis of this analysis, we will explain the POS tags that we have selected. Note that the main goal behind the development of our POS tagger is to use it for Named Entity extraction. As such, in selecting the tags we have made sure that the tag set is rich enough to allow a good training and a good performance of the HMM-based POS tagger. At the same time, we have tried to be careful that the tag set is small enough to make the training of the POS tagger computationally feasible.

Arabic is a widely spoken language in the world. The Arabic alphabet consists of twenty eight letters, twenty five of which are consonants and the remaining three letters are long vowels. A distinguishing feature of Arabic is that no letters are used to represent short vowels. Instead, they are presented by short strokes called diacritics, which are placed either above or below the preceding consonant, as explained in the previous section. Arabic is read from right-to-left, and transliterated Arabic is read from left-to-right. The Arabic language is an inflectional language that is known for its rich vocabulary and complex morphology. In this section, we shall present the Arabic morphology and give a description of our proposed POS tag set.

Arabic has two genders, masculine and feminine. It also has three persons, one to describe the speaker (first person), one to describe the person being addressed (second person) and one to describe the person that is not present (third person). Arabic differs from other languages like English and French in that it has three numbers instead of two. So, in addition to the singular and plural, there is also the dual that is used for describing the actions of two people. We propose to realize two genders (F, M), three persons (1, 2, 3) and three numbers (S, D, P). See appendix A for the whole POS Tag set. Arabic words such as nouns and verbs consist of a stem surrounded by prefix and suffix parts, which indicate grammatical categories such as person, number and gender. An example is given in Table 2.

	Suffix	Stem	Prefix
Arabic	ين (iyn)	أكل ('kul)	ت (ta)
Morphological Analysis	Suffix, 2nd person, feminine	Verb	Prefix, 2nd person
Meaning	You feminine	eat	you

Table 2 : Morphological Structure of ين ت أكل *(ta'kuliyna/you eat)*

2.1. Nouns

Arabic nouns can be subcategorized into adjectives, proper nouns and pronouns. A noun can be definite or indefinite. The definite state is marked by the article ال(al / the) as in الـرجل ('alrajul / the man). The indefinite state is marked by ending the noun with what is known as "tanwiyn", i.e. a doubled diacritic, as in رسميًا (rasmiyyaN / officially). The Noun category and its subcategories shall be tagged with the following tags : NOUN (noun), ADJ (adjective), PNOUN (proper noun), PRON (pronoun), INDEF (indefinite noun), DEF(definite noun).

There are three grammatical cases in Arabic : the nominative (الــرفع), the accusative (الـنصــب) and the genitive (الجر). A noun is in the nominative case when it is a subject, in the accusative case when it is the object of a verb, and in the genitive case when it is the object of a preposition. These cases are distinguished based on the noun suffixes. For example, the suffix ون (uwn) is used with plural masculine nouns that are in the nominative case ; the suffix يـن (iyn) is used with plural masculine nouns that are in the accusative or genitive case ; and the suffix ات (aat) is used with plural feminine nouns in all cases. Also, the dual suffix ان ('aan) is used with nouns in the nominative case and the dual suffix يـن ('ayn) is used with nouns in the genitive or accusative case. We will use one tag SUFF for all suffixes plus the gender and number tags to form a combined tag as illustrated in Table 3.

Case	Nominative	Genitive	Accusative	All
Word	مسـلمون	مسـلمين	مسـلمان	مسـلمات
Transliteration	muslimuwn	muslimiyn	muslimaan	muslimaat
Meaning	Muslims (masc., plural)	Muslims (masc., plural)	Two Muslims (masc., dual)	Muslims (fem., plural)
Suffix POS tag	ون/ SUFF_M_P	يـن/ SUFF_SUBJ_ALL	ان/ SUFF_M_D	ات/SUFF_F_P

Table 3 : Different plural and dual forms of the word مسـلم *(muslim)*

We have chosen to tag the suffix يـن (iyn or 'ayn) as SUFF_SUBJ_ALL because this suffix can stand for different sub-categorizations and could thus be tagged differently as shown in Table 4. However, since the main use of our tagger is intended to be for Named Entity extraction, there is no need to have that fine-grained a tag set. Note that "ALL" in SUFF_SUBJ_ALL indicates that the suffix يـن could stand for any of the genders and numbers for nouns in the nominative case.

Word	Suffix	Morphology Analysis	POS Tag
مسلمين	ين	Suffix Subject Masculine Plural	SUFF_SUBJ_ALL
مسـلمين	ين	Suffix Subject Masculine Dual	SUFF_SUBJ_ALL
مسلمتين	ة + يـن	Suffix Subject Feminine Dual	SUFF_SUBJ_ALL
تأكلين	ين	Suffix Subject Feminine Single	SUFF_SUBJ_ALL

Table 4 : POS of various words ending with the suffix يـن

Also, the suffix ة (taa' marbuwTah) is found attached to the end of nouns and adjectives to express that the noun is feminine, singular ; hence the noun gets tagged as SUFF_F_S. So, the feminine adjective قليلة (qaliylah / little) will be POS tagged as ADJ+SUFF_F_S

2.2. Pronouns

Pronouns can be stand-alone words such as the demonstrative pronoun هذا (haadhaa / this-masculine). Also, pronouns can be attached to a word in the form of affixes. They can also be attached to nouns to indicate possession. Moreover, they can be attached to verbs as direct objects, or attached to prepositions such as فيه (fiyhi / in it"). We have selected to tag demonstrative, possessive and direct object pronouns with the following tags : DPRON, PPRON and SUFFDO. The demonstrative pronouns (two of which are shown in Table 5) can be tagged using the proposed tagset as follows :

Pronoun	Morphological Analysis	POS Tag
هذا	Singular, masculine, demonstrative pronoun	DPRON_MS
هذه	Singular, feminine, demonstrative pronoun	DPRON_FS

Table 5 : Sample tagging of two demonstrative pronouns

The pronoun أنت ('ant / you) can be tagged as second person singular masculine or second person singular feminine pronoun depending on the diacritic that would be used on the last letter : fatha (short /a/) or kasra (short /i/). If fatha is used, the pronoun is masculine whereas with kasra the pronoun is feminine. Since diacritics are usually not used in the written media and publications, we can generalize both tags to one tag (PRON_2S) that can be used to tag the same word surface as second person singular pronoun (Table 6).

Word	Morphology Analysis	POS Tag
أنـت	Second person singular feminine/masculine pronoun	PRON_2S

Table 6 : Tagging of the pronoun أنت *(you)*

2.3. Verbs

Verbs in Arabic can be perfect, imperfect or imperative verbs. Perfect verbs are used to describe completed actions ; imperfect verbs indicate uncompleted actions ; while imperative verbs express an order. The imperfect verbs have three moods indicative (الـــرفع), subjunctive (الـنصـــب), and jussive (الجـزم). Perfect verbs can have pronoun suffixes and subject indicators that indicate the person, gender and number of the subject. For example, the perfect verb قالـت (qaalat / she said) is for the third person, feminine, singular subject. The word علّمته (she taught him) has a third person, feminine, singular subject ت (she) and a pronoun suffix ه (him). Verbs can be attached to a prefix indicating the future س (sa, will) or preceded by a word indicating the futureسـوف (sawfa, will). We have selected to tag the prefix س and the word سـوف as FUTURE. The following POS tags shall be used for verbs : PVERB (perfect verb), IVERB (imperfect verb), CVERB (imperative verb), MOOD_SJ (subjunctive or jussive), MOOD_I (indicative), SUFF_SUBJ (suffix subject), FUTURE (future).

2.4. Particles

There is a group of particles (إنَّ 'indeed / verily', أنَّ' that', كأنَّ' it seems that', لكنَّ' but', ليتَ' I wish', لعلَّ'maybe') known in Arabic grammar as إن وأخواتهـا ("'inna wa-akhawaatuhaa" / inna and its sisters). The grammatical function of these words is to come before a noun and change its case from nominative to accusative. These words were morphologically tagged as function words in (Buckwalter, 2002). For our POS tag set, we have kept the same tag name represented as FUNC_WORD. Also**,** Arabic particles include interrogation, conjunction, preposition, and negation particles. Interrogation words such as ماذا (maadhaa/what) and لماذا (limaadhaa/why) shall be assigned the INTERROGATE tag. Some other Arabic particles can be used for negation or denial purposes such as ليس (laysa / not). Negation words shall be

assigned the NEGATION tag whereas conjunction and preposition will be represented as CONJ and PREP respectively.

In Arabic, numeral quantities can be written in two different ways : numerically and alphabetically. For example, the 11th of October can be expressed alphabetically as الحادي عشـــر مـن أكـتوبـر ('al-Haadiy `ashar min uktuwbar) and numerically as 10/11. Numeral quantities that are expressed alphabetically can be tagged as shown in Table 7.

Word	Meaning	POS Tag
الحادي	The first of	DEF+ADJ
عشـر	Ten	NOUN
من	From	PREP
أكـــتوبـر	October	PNOUN

Table 7 : Tagging of the numeral sentence الـحادي عشـر مـن أكـتوبر *(the 11th of October)*

Numeral quantities that are expressed numerically can be given a single tag NUM. Punctuation marks are tagged as PUNC. We have now introduced and explained the basic tag set. The reader is referred to appendix A for the entire Arabic tag set. In the next section, we will present the design of the HMM model that we have developed for POS tagging Arabic text.

3. The HMM-Based POS Tagger

3.1. Approach

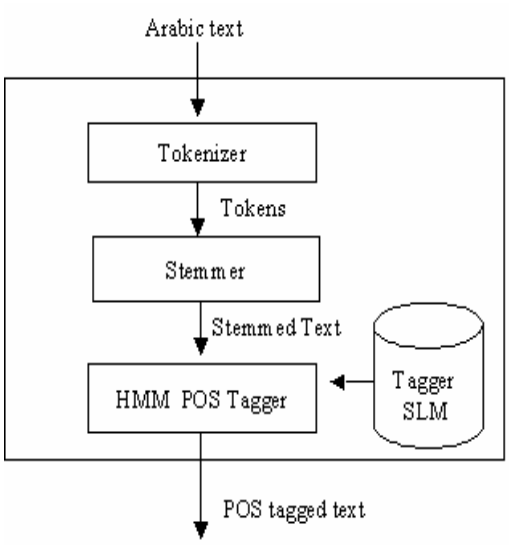

Figure 1 : HMM POS Tagger architecture

As explained in the previous section, in Arabic, prepositions, conjunctions and pronouns can be attached to a word stem as prefixes or suffixes. This results in a large vocabulary. Thus if one needs a corpus which is rich enough to account for the various morpho- syntactic phenomena of Arabic, the corpus would need to be very large indeed. To avoid this, we can separate affixes from stems and POS tag them along with the stems. Prior to the stemming process, pre-processing steps are also needed to prepare the input text for the tagging module. To this end, we have developed a tokenizer and a stemmer to prepare the input text for POS tagging as shown in Figure 1. Let us now present the different modules of the POS tagging system.

3.1.1. The Tokenizer

Since punctuation marks need to be tagged in the POS- tagging task (Jurafsky and Martin, 2000), we must extract them and pass them to the POS tagger so that it tags them as PUNC. The purpose of the tokenization phase is to go through some pre-processing steps in order to prepare the input text for the remaining modules. Various punctuation marks such as the full stop are usually found attached to the preceding word. In order to process each word correctly, we have developed a tokenizer to separate the punctuation marks from the words. Then the tokenizer converts the input text into a list of words using the space as a delimiter. The resulting list is passed to the stemmer

3.1.2. The Stemmer

Stemming is the process of segmenting and separating affixes from a stem to produce prefix, stem, and suffix parts (see Table 8).

Un-stemmed	Stemmed		
	Prefix	Stem	Suffix
المعلمون (teachers)	ال	معلم (teacher)	ون
'almu`allimuwn	'al	mu`allim	uwn

Table 8 : Stemming Arabic text

Stemming of Arabic text was discussed in (Chen and Gey, 2002 ; Larkey et al., 2002 ; Khoja and Garside,1999). The stemmer in (Larkey et al., 2002) uses a predefined lexicon of prefixes and suffixes to produce a stemmed form as prefix/stem/suffix. This stemmer segments only up to a three-character prefix and up to a two-character suffix. Thus, it fails to remove prefixes that are more than three characters long and suffixes that are more than two characters long. On the other hand, the stemmer in (Buckwalter, 2002) returns all valid segmentations based on the fact that an Arabic prefix length can go from zero to four characters, the stem can consist of one or more characters, and the suffix can consist of zero to six characters. In our case, we want a stemmer that can segment words containing prefixes and suffixes of any length and which, at the same time, returns one valid solution. To evaluate the heuristic of selecting the first segmentation as the correct one for a given word, we have run Buckwalter's stemmer on a 9k corpus and the result has been very encouraging. The percentage that the first segmentation solution is the correct one was 96%. So, we have used Buckwalter's stemmer to stem the training data. All invalid segmentations have been manually corrected.

3.1.3. The POS Tagger

N-gram language model is needed to capture and determine language regularities such as the frequency of some word or the probability of some phrase being followed by some word. The history of previous N-1words is used to obtain the N-gram language model. For example, the probabilities of a bigram model are obtained using the previous word only but in a trigram model the probabilities are obtained using the last two words. To build our POS tagger we have constructed trigram language models and used the trigram probabilities in building the HMM model. An HMM model can be expressed by four parameters : the set of states S, the observation sequence O, a matrix A which stores transition probabilities between states (where a state is a tag), and matrix B which stores state observation probabilities (called emission probabilities).

For the POS-tagging problem, we consider a sequence of words as the observation sequence. The transition probabilities are obtained from the contextual trigram model and the emission probabilities are obtained from the lexical trigram model. The states of the HMM model are the POS tags to which two special states are added : start and end. These last two states are non-emitting states, which means that the emission probabilities at these states are 0 and they are used to denote the start and the end of an observation sequence. The use of these special states avoids the use of the initial parameter which holds the probability that the HMM will start at some specific state (Jurafsky and Martin, 2000). In the HMM model, an "emission probability" is the probability of observing the input sentence or sequence of words W given the state sequence T, that is P(W|T). Also, from the state transition probabilities we can calculate the probability P(T) of forming or following the state sequence T. Lets consider the HMM model (Figure 2) of the sentence : البيت كبير (al-baytu kabiir / the house is big).

The transition probability from the state Noun to the state Adjective is P(ADJ | DEF NOUN) which is formally $P(n_i|n_{i-2}\ n_{i-1})$ and P(كبير |ال DEF بيت NOUN ADJ) is the observation probability that كبير (kabiir / big) is an adjective which is formally $P(w_i|w_{i-2}n_{i-2}\ w_{i-1}n_{i-1}\ n_i)$.

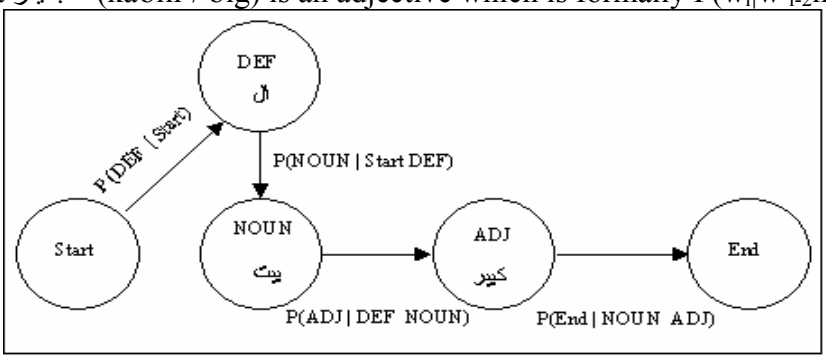

Figure 2 : HMM diagram of the Arabic sentence : البيت كبير (al-baytu kabiir / the house is big)

Formally, for a sequence of words W the problem is to find the tag sequence T which maximizes the probability P(T|W),

$$\text{argmax } P(T|W) \quad (3.1)$$

Using the Bayes' rule for conditional probabilities, we can calculate P(T|W) as

$$P(T|W) = [P(W|T) * P(T)]/P(W)$$

Since the probability of the word sequence P(W) is the same for each name sequence, we can ignore it giving P(N|W) = P(W|N) * P(N) and, hence, P(N|W) needs to be maximized as

$$\text{argmax } P(W|T) * P(T) \quad (3.2)$$

A training corpus of Arabic news articles has first been stemmed using the stemming component and then tagged manually with our proposed tag set. Then a trigram language model was built for the tagged training corpus. The trigram language model computes lexical probabilities. Then, we obtained the POS tag sequences from the training corpus and created a trigram Arabic language model based on the POS tag corpus. The benefit of this tag model is that it allows calculating the probability of one tag following another tag (contextual model). Since we use trigram models it is possible that some trigrams were never observed in the training corpus. The probability of unseen trigrams cannot be assigned to zero because this will cause the (computed) probability of an entire observed sequence to be zero. We rather use the back-off smoothing technique (Chen and Goodman, 1996) so that the model backs off to a bigram model. Similarly, if the bigram was never observed during training, then we would back off to a unigram model. For the purposes of this smoothing, we have created unigram, bigram and trigram lexical and contextual language models. Next, lexical and contextual probabilities were used to build the HMM model's parameters as follows : contextual probabilities were stored in parameter A as transitions probabilities and lexical probabilities were stored in parameter B as the emission probabilities. Once matrices A and B are computed, search needs to be performed to find the POS tag sequence that maximizes the product of the lexical and contextual probabilities. For example, if the observed (input) sequence has three words, two of which can be assigned two POS tags then the search space will be 2*2*1 = 4. The Viterbi (Forney, 1973) algorithm is used to decode or find the optimal path in the search space.

3.2. Constructing the HMM Model :

There are two types of phrases in Arabic : noun phrase and verb phrase. A noun phrase starts with a noun and is followed by a noun or adjective as in the following sentences :

1 الـقمر مـنـير ('al qamaru muniyruN / the moon is shining)
2 السـمـاء صـــافـيـة ('al samaa'u Saafiyah / the sky is clear)

A verb phrase starts with a verb and is followed by a noun or proper noun as in the following sentences :

1 أكـــل الولد الـــتفـاحـــة ('akala 'al-waladu 'al-tuffaaHah / The boy ate an apple)
2 يـــأكل حســـن الـتفاحـة (ya'kulu Hassan 'al-tuffaaHah / The boy is eating the apple)

Every noun can be preceded by the definite article ('al), which can be preceded by prepositions such as (li and bi). Also conjunctions such as (wa and fa) can come before any combination of these. To summarize a noun phrase structure formally, we have designed the following noun phrase structure[2] expression :
[*CONJ *PREP *DEF *FUNC_WORD *[NEGATION INTERROGATE]] [NOUN PNOUN ADJ] [*SUFF% *%PRON%]
Likewise, a verb phrase structure can be formally summarized by the following expression :
[*CONJ *PREP *[NEGATION INTERROGATE] *FUTURE *IV%] [PVERB IVERB CVERB] [*SUFF% *%PRON%]

Using the noun and verb phrase structures helps us in designing the HMM model. From the phrase structure, we can define the valid state transitions (POS tags). For example, from state CONJ to state NOUN, there must be a direct arc (transition) and, at the same time, indirect arc via states PREP and DEF. Consider the (stemmed) sentence in Table 9.The word شـخص can be a noun (shakhS) that means "person" or a perfect verb (shakhkhaSa) that means "to diagnose". To resolve this ambiguity, let us check the tags of the previous two-words, which are DPRON_MS DEF. From the language model, the trigram DPRON_MS DEF NOUN is 0.459 but the trigram DPRON_MS DEF PVERB is not estimated because it was not seen in the training corpus. Thus, the word شـخص is tagged as NOUN and the whole sentence is POS tagged as in Table 9.

Word	فرنسي	شخص	ال	هذا
Transliteration	faransiyy	shakhS	al	haadhaa
Meaning	French	person	is	This
POS Tag	ADJ	NOUN	DEF	DPRON_MS

Table 9 : POS tagging of sentence : فرنسي شخص ال هذ (haadhaa 'alshakhS faransiyy)

As illustrated in the previous example, the HMM POS tagger requires contextual probabilities of the valid POS tag sequences. From the annotated training data we have deleted all the Arabic words and kept only the POS tag sequences. We then created a trigram model for these POS tag sequences so as to calculate the contextual probabilities. In other words, we have made use of the Arabic sentence structure by creating a trigram language model for the different noun and verb phrase structures (valid tag sequences). All the tags in the POS tag set are forming the HMM model states plus the two special states : Start and End. It would be difficult to draw the whole POS HMM model that we have designed since we would have to draw all the POS tags as states and add arcs with transition probabilities between any two states. For readability purposes, we only show in Figure 3 an example of simple noun and verb phrases HMM graph.

[2] The star (*) symbol is used to denote optional, [ ] is used to denote disjunction, and % to denote an arbitrary number of characters.

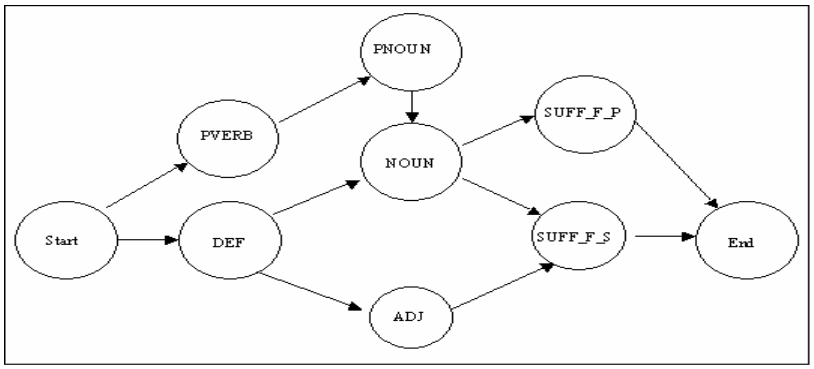

Figure 3 : Partial POS HMM model

4. Experimental Results

4.1. Training and Test corpus

We have examined LDC's Arabic TreeBank corpus (LDC, 2005) that consists of 734 news articles. The TreeBank is transliterated and POS tagged. Since we have developed our own tag set, we could not use the Treebank to train the tagger. Instead, we have developed a 9.15 MB corpus of native Arabic articles, which we have manually tagged using the developed tag set (see Section 2). The tagger was trained on 27594 nouns, 23554 verbs, 5722 adjectives and 5384 proper nouns. To evaluate our HMM POS tagger we need to compare it to an answer key corpus that is a hand tagged corpus. We have prepared and tagged a test corpus of size 6k (944 words).

4.2. Evaluation and Metrics

We have used the F-measure to evaluate the POS tagger performance. We compute the *F-measure* as : [2 x Precision x Recall] / [Precision + Recall]

where Precision = Ncorrect / Nresponse
and Recall = Ncorrect / Nkey

Ncorrect is the total number of correct tags assigned by the tagger ; Nresponse is the total number of assigned tags (correct and incorrect) ; and Nkey is the total number of the assigned tags in the answer key test corpus. Also, the tagger errors could be analyzed using the confusion matrix (Jurafsky and Martin, 2000) where the row labels indicate correct tags and column labels indicate the tagger response tags.

4.3. Results

The HMM tagger achieved 97 % using the F-measure given above. By this score, our HMM tagger outperforms the other Arabic statistical POS taggers that we are aware of. We can analyze the POS tagger performance using the confusion matrix shown in Table 10. For example, incorrectly tagging a NOUN as ADJ amounted for 3.5% of the errors.

	NOUN	PNOUN	ADJ	PVERB	IVERB
NOUN	-	1.6	3.5	0.6	-
PNOUN	2.0	-	0.5	-	-
ADJ	2.3	1.5	-	-	-
PVERB	0.5	-	-	-	1.8
IVERB	-	-	-	2.3	-

Table 10 : The HMM tagger confusion matrix

Using the contextual model, we were able to tag the word عـالم (`aalam/ world) as noun (see Figure 4) instead of adjective (the knower) because the contextual probability of the trigram PREP DEF NOUN is 0.2155998 whereas the contextual probability of the trigram PREP DEF ADJ is 0.0439055.

وال/CONJ+DEF شـرودر/PNOUN ألمانـي/ADJ ال/DEF مسـتشـار/NOUN ال/DEF افـتـتـح/PVERB
ال/DEF دول/NOUN ال/DEF ة/SUFF_F_S جامع/NOUN ل/PREP عام/ADJ ال/DEF أمين/NOUN
فـــرانكفورت/PNOUN معرض/NOUN موسى/PNOUN عمرو/PNOUN ة/SUFF_F_S عـربـي/ADJ
ضـد/NOUN مشـتـرك/ADJ نـداء/NOUN ب/PREP كـتـاب/NOUN لـل/PREP+DEF دولـي/ADJ ال/DEF
عرب/NOUN ال/DEF ة/SUFF_F_S صـور/NOUN تشـــويـه/NOUN ات/SUFF_F_P محاول/NOUN
وال/CONJ+DEF مسـلم/ADJ يـن/ SUFF_SUBJ_ALL فـي/PREP ال/DEF عـالم/NOUN

Figure 4 : POS tagged sample text

The performance of the POS tagger decreased to 55 % when it was used to tag a non-stemmed text. Some minor misspellings can change the meaning of a word and hence its POS tag. For example, the composed word بـأن (bi-'anna / in that) must be tagged as PREP+FUNC_WORD but if it is written as بـان (i.e. without "hamza" on the middle letter), it becomes a perfect verb PVERB that means "became clear". In such cases, we consider the tagger response as a correct one.

5. Conclusion

In this paper, we have presented a statistical approach that uses HMM to do POS tagging of Arabic text. We have analyzed the Arabic language quite systematically and have come up with a good tag set of 55 tags. We have then used Buckwalter's stemmer to stem Arabic corpus and we manually corrected any tagging errors. Having done this, we designed and built an HMM-based model of Arabic POS tags, which we trained on the annotated corpus. This has led to an HMM-based POS tagger that has an F-measure of 97 %, which is a very good result indeed. One of the greatest advantages of having a trainable POS tagger is that it will speed up the process of tagging huge corpora. Out of this work, we have developed almost 10 MBs of Arabic corpus and we are in the process of enlarging the size of this corpus to reach one million words. Having produced a performant POS tagger,, we will use it in a variety of ways in our research in information retrieval, machine translation project transfer module, etc.

Appendix A : POS TAG Set Used

ADJ	EXCEPT	PPRON_2FP	PRON_3D	SUFF_M_P
CONJ	FUNC_WORD	PPRON_3FP	PRON_3FP	SUFF_SUBJ_1P
CVERB	FUTURE	PREP	PRON_3FS	SUFF_SUBJ_2D
DEF	INTERROGATE	PRON	PRON_3MP	SUFF_SUBJ_2FP
DPRON_F	IV1P	PRON_1P	PRON_3MS	SUFF_SUBJ_2MP
DPRON_FD	IV2	PRON_1S	PVERB	SUFF_SUBJ_2S
DPRON_FP	IV3	PRON_2	SHORT_FORM	SUFF_SUBJ_3FD
DPRON_FS	IVERB	PRON_2D	SUFF_F_D	SUFF_SUBJ_ALL
DPRON_MD	NEGATION	PRON_2FP	SUFF_F_P	SUFF_SUBJ_FP
DPRON_MP	NOUN	PRON_2MP	SUFF_F_S	SUFF_SUBJ_MP
DPRON_MS	PNOUN	PRON_2S	SUFF_M_D	SUFF_S_INDEF

References

Jurafsky D. and Martin J. (2000). *Speech and Language Processing*. Prentice Hall.

Brill E. (1994). Some Advances in Transformation Based Part of Speech Tagging. In *proc. Of ICAI'94 (The Twelfth International Conference on Artificial Intelligence) :* 722-727.

Brants T. (2000). TnT : a statistical part of speech tagger. In *proc. of ANLP'2000 (the 6th Conference on Applied Natural Language Processing) :* 224-231.

Garside R., Leech G. and McEnery A. (1997). *Corpus Annotation : Linguistic Information from Computer Text Corpora*. Addison Wesley Longman Inc.

Cutting D., Kupiec J., Pedersen J.O. and Sibun P. (1992). A practical part–of–speech tagger. In *proc. of ANLP'94 (the 3rd Conference on Applied Natural Language) :* 133-140.

Merialdo B. (1994). Tagging English Text with a Probabilistic Model. *Computational Linguistics*, vol.(20) :155-172.

Olde B. A., Hoener J., Chipman P., Graesser A.C. and the Tutoring Research Group. (1999). A Connectionist Model for Part of Speech Tagging. In *proc. of FLAIRS'99 (the 12th International Florida Artificial Intelligence Research Society Conference) :* 172-176.

Marques N. M. and Lopes J. G. P. (1996). Using Neural Nets for Portuguese Part-of-Speech Tagging. In *proc. of CSNLP'96 (the 5th International Conference on the Cognitive Science of Natural Language Processing*) : 172-176.

Freeman A.T. (2001). Brill's POS tagger and a Morphology parser for Arabic. In proc. Of ACL'2001 (the 39th Annual Meeting of Association for Computational Linguistics & 10th Conference of the European Chapter, Workshop on Arabic Language Processing) : 7.

Khoja S. (2001). APT : Arabic Part-of-speech Tagger. In *proc. of NAACL'2001 (the Student Workshop at the Second Meeting of the North American Chapter of the Association for Computational Linguistics) :* 20-26.

Diab M., Hacioglu K. and Jurafsky D. (2004). Automatic Tagging of Arabic Text : From Raw Text to Base Phrase Chunks. In *proc. of HLTNAACL'04* (*Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*) : 149–152.

The Linguistic Data Consortium (LDC), available from http://www.ldc.upenn.edu/.

Duh K. and Kirchhoff K. (2005). POS Tagging of Dialectal Arabic : A Minimally Supervised Approach. In *proc. ACL2005 (43rd Annual Meeting of the Association for Computational Linguistics) :* 55-62.

Buckwalter T. (2002). Buckwalter Arabic Morphological Analyzer. LDC Catalog No. LDC2002L49, the Linguistic Data Consortium, University of Pennsylvania.

Chen A. and Gey F. (2002). Building an Arabic Stemmer for Information Retrieval. In *proc. Of TREC'2002 (the 11th Text Retrieval Conference :* 47.

Larkey L. S., Ballesteros L. and Connell M. E. (2002). Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-Occurance Analysis. In *proc. of SIGIR'2002 (the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval) :* 269-274.

Khoja S. and Garside R. (1999). Stemming Arabic Text. Technical report, Lancaster University, Lancaster, U.K.

Chen S. and Goodman J. (1996). An Empirical Study of Smoothing Techniques for Language Modeling. In *proc. of ACL'96 :* 310-318.

Forney G. D. (1973). The Viterbi Algorithm. In *proc. of the IEEE Transactions on Information Theory* : 263-278.