

# Agent Based Arabic Language Understanding

Muhammad Taha<sup>\*</sup>, Tarek Helmy<sup>\*\*</sup> and Reda Abo Alez<sup>\*\*\*</sup>

<sup>\*</sup> *Cairo University, Faculty of Science, Mathematical Department, Cairo, Egypt.*

<sup>\*\*</sup> *College of Computer Science and Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Mail Box 413, Kingdom of Saudi Arabia.*

<sup>\*\*\*</sup> *Al Azhar University, Faculty Of Engineering Computers & Systems Engineering Department, Nasr City, Cairo, Egypt.*

*Emails: mtahacom@yahoo.com, helmy@kfupm.edu.sa.*

## Abstract

*Arabic Language understanding (ALU) computing is considered an AI-hard task. In this paper, we propose an Agent model for ALU problem. This agent is detailed in this paper. An ALU system is developed for 'Voice Activated Drawing Interface'. Our experiment shows that agent-based ALU can be very robust and reliable in comparison to text analysis by using rules of Arabic language, parts of speech and structure of sentence.*

## Keywords

Language understanding, Hidden Markov Model (HMM), Semantic, and Tagging.

## 1. Introduction

Russell and Norvig [10] have defined an agent as being anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors. The Multi-Agent paradigm is one which promotes the interaction and cooperation of intelligent autonomous agents in order to deal with complex tasks [6]. Language understanding systems use a large set of rules approach to explain the syntactic and semantic possibilities. Moreover, this approach is much more complicated in the cases of spoken languages recognized from long distance, geriatric utterances, grammatically incorrect sentences, and in a very noisy environment [13]. An alternative approach is to use some statistical methods to map directly from word strings to the intended meaning structures. In this approach, hand-crafted grammars and rules are replaced by statistical models that are automatically learned from some training data [13].

This paper is organized as follows: Section 2 describes the hidden markov model. Section 3 previews the Viterbi algorithm. In section 4, we propose Arabic lan-

guage understanding agent based on hmm semantic tagging. In next section we present the experiments. In last section we show the conclusions

## 2. Hidden Markov Model

The hidden Markov model (HMM.) is a very powerful statistical method of characterizing the observed data samples of a discrete-time series.

The basic HMM theory was published in a series of classic papers by Baum and his colleagues[2]. The HMM has become one of the most powerful statistical methods for modeling speech signals. Its principles have been successfully used in automatic speech recognition, formant and pitch tracking, speech enhancement, speech synthesis, statistical language modeling, part-of-speech tagging, spoken language understanding, and machine translation [1], [2], [3], [4], [5], [7], [8], and [9].

A hidden Markov model is defined by:

$O = \{o_1, o_2, \dots, o_M\}$  — An output observation alphabet: The observation symbols correspond to the physical output of the system being modeled.

$\Omega = \{1, 2, \dots, N\}$  — A set of states representing the state space. Here  $s_t$  is denoted as the state at time  $t$ .

$A = \{a_{ij}\}$  — A transition probability matrix, where  $a_{ij}$  is the probability of taking a transition from state  $i$  to state  $j$ , i.e.,  $a_{ij} = P(s_t = j \mid s_{t-1} = i)$

$B = \{b_i(k)\}$  — An output probability matrix where  $b_i(k)$  is the probability of emitting symbol  $o_k$  when state  $i$  is entered. Let  $X = X_1, X_2, \dots, X_t$  be the observed output of the HMM. The state sequence  $S = s_1, s_2, \dots, s_t$ , is not observed (hidden), and  $b_i(k)$  can be rewritten as follows:

$$b_i(k) = P(X_t = o_k \mid s_t = i)$$

$\pi = \{\pi_i\}$ — A initial state distribution where  
 $\pi_i = P(s_o = i) \quad 1 \leq i \leq N.$

### 3. The Viterbi Algorithm

In many applications, it is desirable to find best path (or state sequence). As a matter of fact, finding the best path (state sequence) is the cornerstone for searching in continuous speech recognition. Since the state sequence is hidden (unobserved) in the HMM framework, the most widely used criterion is to find the state sequence that has the highest probability of being taken while generating the observation sequence. In other words, we are looking for the state Sequence  $S = (s_1, s_2, \dots, s_T)$  that maximizes  $P(S, X | \Phi)$ . This problem is very similar to the optimal-path problem in dynamic programming. As a consequence, a formal technique based on dynamic programming, known as Viterbi algorithm [12], can be used to find the best state sequence for an HMM.

The Viterbi algorithm can be regarded as the dynamic programming algorithm applied to the HMM. Instead of summing up probabilities from different paths coming to the same destination state, the Viterbi algorithm picks and remembers the best path. To define the best-path probability:

$$V_t(i) = P(X_1^t, S_1^{t-1}, s_t = i | \Phi)$$

$V_t(i)$  is the probability of the most likely state sequence at time  $t$ , which has generated the observation  $X_1^t$  (until time  $t$ ) and ends in state  $i$ . A similar induction procedure.

Step 1: Initialization

$$V_1(i) = \pi_i b_i(X_1) \quad 1 \leq i \leq N$$

$$B_1(i) = 0$$

Induction

$$V_t(j) = \max_{1 \leq i \leq N} [V_{t-1}(i) a_{ij}] b_j(X_t) \quad 2 \leq t \leq T; \quad 1 \leq j \leq N$$

$$B_t(j) = \text{Arg Max}_{1 \leq i \leq N} [V_{t-1}(i) a_{ij}] \quad 2 \leq t \leq T; \quad 1 \leq j \leq N$$

Termination

$$\text{The best score} = \max_{1 \leq i \leq N} [V_T(i)]$$

$$s_T^* = \text{Arg Max}_{1 \leq i \leq N} [B_T(i)]$$

Backtracking

$$s_t^* = B_{t+1}(s_{t+1}^*) \quad t = T-1, T-2, \dots, 1$$

$S^* = (s_1^*, s_2^*, \dots, s_T^*)$  is the best sequence.

### 4. Arabic Language Understanding Agent Based On Hmm Semantic Tagging

Understanding agent can be described as follows:

*Percepts:* This agent gets Arabic text words.

*Actions:* There are four main actions available: Calculate the probability of the sequence of semantic tags, calculate the probability of each semantic tag generating a word, and, find the best sequence of semantic tags for a given sequence of words.

*Goals:* The goal for this agent is to map each Arabic word to semantic tag.

*Environment:* The environment consists of Arabic users.

Now, Consider some observation(s) (for example sequence of Arabic words), and then a classification task is to determine which of a set of classes it belongs to. Semantic tagging is viewed as a sequence classification task. So here the observation is a sequence of Arabic words, and it is our task to map them a sequence of semantic tags.

The task of semantic tagging is the unique annotation of a word with a semantic category. Let  $S$  be defined as the set of all semantic tags for specific domain, and  $\Sigma$  the set of all words. In a statistical tagging task, one is given a sequence of Arabic words  $W = w_1 \dots w_k \in \Sigma^*$ , and is looking for a sequence of semantic tags  $S = s_1 \dots s_k \in S^*$  that maximizes the conditional probability  $p(S | W)$ , hence one is looking for

$$\arg \max_s p(S | W) = \arg \max_s \frac{p(S) p(W | S)}{p(W)}$$

$p(W)$  is independent of the chosen tag sequence, thus it is sufficient to find  $\arg \max_s p(S) p(W | S)$ .

In an  $n$ -gram model for each pair  $(w, s) \in \Sigma \times S$ , the lexical probabilities  $p(w | s)$ , and for each  $n$ -tuple  $(s_1 \dots s_n) \in S \times \dots \times S$  the transition probabilities  $p(s_n | s_1 \dots s_{n-1})$  are defined. These approximation the lexical and conditional probabilities with

$$p(W | S) \approx p(w_1 | s_1) \dots p(w_k | s_k), \text{ and}$$

$$p(S) \approx \prod_{i=1}^k p(s_i | s_{i-n+1} \dots s_{i-1})$$

Now, the probability of a sequence of words  $W = w_1 \dots w_k \in \Sigma^*$  having a sequence of semantic tags  $S = s_1 \dots s_k \in S^*$  is the product of their lexical and transition probabilities

$$p(W, S) = p(S) p(W | S)$$

$$\approx \prod_{i=1}^k p(s_i | s_{i-n+1} \dots s_{i-1}) p(w_i | s_i)$$

Therefore, the best sequence of semantic tags  $S$  for a given sequence of words  $W$  is done by finding

$$\arg \max_{s_1 \dots s_k} \prod_{i=1}^k p(s_i | s_{i-n+1} \dots s_{i-1}) p(w_i | s_i)$$

In reality, the probabilities  $p(s_i | s_1 \dots s_{i-1})$  are impossible to estimate for even moderate values of  $i$ , since most histories  $s_1 \dots s_{i-1}$  are unique or have occurred only a few times. A practical solution to this problem is to assume that  $p(s_i | s_1 \dots s_{i-1})$  depends only on some equivalence classes. The equivalence class can be simply based on the several previous semantic tags  $s_{i-n+1} \dots s_{i-1}$ . This leads to an  $n$ -gram language model. If the tag depends on previous two tags, we have tri-gram:  $p(s_i | s_{i-2}, s_{i-1})$ . Similarly, we can have uni-gram:  $p(s_i)$ , or bi-gram:  $p(s_i | s_{i-1})$  language models. To estimate  $p(s_i | s_{i-1})$ , the frequency with which the tag  $s_i$  occurs given that the last tag  $s_{i-1}$ , we simply count how often the sequence  $(s_{i-1}, s_i)$  occurs in some text and normalized the count by the number of times  $s_{i-1}$  occurs.

The tag transition probabilities

$$p(s_i | s_{i-1}) = \frac{C(s_{i-1}, s_i)}{C(s_{i-1})}$$

represent the probability of

a tag, given the previous tag, and the word likelihood

$$p(w_i | s_i) = \frac{C(s_i, w_i)}{C(s_i)}$$

probabilities represent the

probability, given that we see a given tag, that it will be associated with a given word.

## 5. Experiments

To evaluate HMM model as a semantic tagging Agent approach to Arabic language understanding system, we consider a natural user interface of mobile computer applications like "simple drawing program". There are some the user's attempts to draw a shape in a drawing application from recorded corpus: "ارسم دائرة حمراء", "ارسم مستطيل", "ارسم خط", "ارسم مربع أصفر", "ارسم دائرة". Let HMM be defined by the two tables 1 and 2. Table 1 expresses the *transition* probabilities between states (i.e. semantic tags). Table 2 expresses the  $b_j(o_i)$  probabilities, the *observation* likelihoods of words given tags.

**Table 1. Semantic tag transition probabilities.**

	Verb	Shape	Color
<S>	0.82	0.42	0.0
Verb	0.0	0.92	0.0
Shape	0.05	0.0	0.91
Color	0.0	0.003	0.0

**Table 2. Observation likelihoods computed from the corpus without smoothing.**

	'ارسم'	'دائرة'	'حمراء'
Verb	0.99	0.0	0.0
Shape	0.0	0.99	0.0
Color	0.0	0.0	0.99

Again, HMM tagging algorithm chooses as the most likely tag sequence the one that maximizes the product of two terms; the probability of the sequence of tags

$$p(s_i | s_{i-1}) = \frac{C(s_{i-1}, s_i)}{C(s_{i-1})}$$

and the probability of each

$$\text{tag generating a word: } p(w_i | s_i) = \frac{C(s_i, w_i)}{C(s_i)}$$

According to the Viterbi algorithm, each cell of the trellis in the column for the word "ارسم" is computed by multiplying the previous probability at the start state (1.0), the transition probability from the start state to the tag for that cell, and the observation likelihood of the word "ارسم" given the tag for that cell. Next, each cell in the "دائرة" column gets updated with the maximum probability path from the previous column. We have shown only the value for the SHAPE cell. That cell gets the max of three values; as it happens in this case, two of them are zero (since there were zero values in the previous column). The remaining value is multiplied by the relevant transition probability, and the (trivial) max is taken. In this case the final value, 0.75, comes from the VERB state at the previous column.

## 6. Results

In this section, we report results from Arabic language understanding agent using HMM semantic tagging. The corpus data used in these experiments were obtained from user's attempts to draw a shape in a drawing application. The data were divided into two sets; training set A and testing set B. Training set A has 15 command-sentences. Testing set B has 25 command-sentences.

Training set A is used to estimate the transition probabilities between states (i.e. semantic tags), and to estimate the observation likelihoods of words given tags.

We have used the accuracy measure to evaluate the semantic tagger performance for testing set B. We compute the accuracy measure as: Accuracy = (no. of correct semantic tags assigned by the tagger) / (no. of assigned tags found). The Accuracy of HMM semantic tagging in the Testing set B is shown in Table 3.

**Table (3): The accuracy results of HMM semantic tagging.**

Arabic Semantic Tags	Testing Set B
	Accuracy
Verb	92%
Shape	96%
Color	88%

The error rate in Arabic understanding in our model can be attributed to the complex morphological structure of Arabic language.

## 7. Conclusions

In this paper, we described Arabic language understanding for voice-command interface system. Our approach is based on HMM. We showed by detailed example, the capabilities of using HMM in simulating Arabic meaning. HMM model semantic taggers have been applied successfully for Arabic text. We showed that HMM is a mechanism that can robustly, and with relatively low training costs, provide the needed estimations. With this capability, voice interface system can carry on an extended conversation, fostering user construction of knowledge and enabling richer evaluation of that knowledge.

## 8. Acknowledgment

The authors would like to express their sincere appreciation to Professor Hany Ammar for his many useful comments.

## 9. References

- [1] Baker J K. "The DRAGON System—An Overview," *Trans. on Acoustics, Speech and Signal Processing*, 1975, 23(1), pp. 24-29.
- [2] Baum L E. and Eagon J A, "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology," *Bulletin of American Mathematical Society*, 1967, 73, pp. 360-363.
- [3] Brown P F, et al, "The Mathematics of Statistical Machine Translation: Parameter Estimation," *Computational Linguistics*, 1995, 19(2), pp. 263-312.
- [4] Church K. "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text," *Proc. of the Second Conf. on Applied Natural Language Processing*, 1988, Austin, Texas, pp. 136-143.
- [5] DeRose S J, "Grammatical Category-Disambiguation by Statistical Optimization," *Computational Linguistics*, 1988(1), pp. 31 -39.
- [6] Ferber J. *Multi-Agent Systems - An Introduction to Distributed Artificial Intelligence*. Addison-Wesley, 1999.
- [7] Huang X D, Arifa Y, and Jack M A, *Hidden Markov Models for Speech Recognition*, 1990, Edinburgh, U.K., Edinburgh University Press.
- [8] Jelinek F. "Continuous Speech Recognition by Statistical Methods." *Proc. of the IEEE*, 1976, 64(4), pp. 532-556.
- [9] Rabiner L R. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. of IEEE*, 1989, 77(2), pp. 257-286.
- [10] Russell, Stuart J. and Peter Norvig. *Artificial Intelligence: A Modern Approach (The Intelligent Agent Book)*. Prentice Hall. 1995; p31.
- [11] Sakoe H and Chiba S, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1978, 26(1), pp. 43-49.
- [12] Viterbi A J, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. on Information Theory*, 1967, 13(2), pp. 260-269.
- [13] Yamina T, "Hybird Method for Tagging Arabic Text", *Journal of Computer Science*, 2006; 2(3), pp. 245-248.