Which Algorithm and Approach for Arabic Part Of Speech Tagging

Mourad MARS[1,2], Georges Antoniadis[1], Mounir Zrigui[2]

[1] Lidilem Laboratory, University of Grenoble,
BP 25, 38040 Grenoble cedex 9, France
[2] UTIC laboratory, University of Monastir, Tunisia
Mourad.mars@e.u-grenoble3.fr
Georges.antoniadis@u-grenoble3.fr
Mounir.zrigui@fsm.rnu.tn
Paper received on 06/08/10, Accepted on 05/10/10

Abstract. This paper presents a recent study aiming to develop a Part Of Speech (POS) tagger for Arabic language. Part-of-speech (also known as POS, Word classes, morphological classes, or lexical tags) of Arabic is an active topic of research in recent years, it is the process of assigning to each sequence of words the right POS tags. The paper is organized as follows. In the first section, an overview of recent work and a briefly introduce to different configurations and tested approaches in POS tagging. In the following section, we describe the different kinds of languages models with various smoothing techniques, as well as several resources and tested approaches to justify the methodology and the appropriate setting used to build our POS tagger. We also tested and evaluate the robustness of the tagger. Furthermore, conclusions, open areas, and future directions are provided at the end.

Keywords: Part-Of-Speech tagging, POS tagger, Arabic, disambiguation, HMM, Algorithm, Language Model.

1 Introduction

Arabic is one of the Semitic languages; known for its rich and systematic but very complex morphological structure and also for its several problems in natural language processing (agglutinative form, run-on word, free concatenation and orthographic variation) [21]. Part-of-speech (also known as POS, Word classes, morphological classes, or lexical tags) of Arabic language is an active topic of research in recent years, it is the process of assigning to each sequence of words the right POS tags.

Many computational methods and algorithms for assigning POS to words have been used including rule-based tagging (hand-written rules), statistical methods (HMM tagging and maximum entropy tagging) as well as other methods like transformation-based, memory-based tagging and also hybrid method for tagging.

Advances in Intelligent and Information Technologies.
Research in Computing Science 50, 2010, pp. 235 -245

Before turning to present our approach in POS tagging and our tests, let's begin with an overview of recently published work.

Table **1.** Sample of Arabic tagged.

Arabic sentence	كتب جميلة .
Transliteration	ktb jmylp .
POS tag	NN Adj PUNCT
Meaning	Nice books.

2 Recent Work on POS Tagging: A Brief Overview

In this section, a briefly overview of different POS taggers for Arabic language recently achieved is presented. Khoja [12] combines both statistical and rule-based techniques and uses a tagset of 131 derived from the BNC tagset. Maamouri and Cieri system [14] achieved an accuracy of 96%, it consist of the automatic annotation output produced by AraMorph, which is the morphological analyzer of Tim Buckwalter [4]. Yahya O. Mohamed Elhadj [21] presented an Arabic POS tagger that uses a HMM model, the evaluation shown an accuracy of about 96%. Freeman's tagger [7] uses a machine learning (ML) approach as in Brill's POS tagger [3], a tagset of 146 tags extracted from the Brown corpus for English, is used. Mona Diab et al. [6] use Support Vector Machine (SVM) method to build their tagger for Arabic and the LDC's POS tagset with the number of 24 tags. Fatma al Shamsi and Guessoum [2] HMM POS tagger has been tested and has achieved a state-of-the-art performance of 97%. Tlili Guiassa [22] uses a hybrid which consists of combining based-rules and memory-based learning methods. This system reports a performance of 85%.

3 Part Of Speech Tagging

What is the best sequence of tags which corresponds to a given sequence of words? The use of a POS tagger can answer this question; it attempts to assign the correct tag or lexical category to all words in a text. This is an example of a tagged text (POS tagger output):

ktb/VB rsAlp/NN ./PUNCT (1)

ktb/NN jmylp/ADJ ./PUNCT (2)

In both sentences, automatically assigning one tag per word is not easy. For example, ktb is ambiguous. That is, it has more than one possible tag. It can be a

verb (write) as in (1) "Write a letter" or a noun (books) as in (2) "Nice Books". The objective of POS tagging is to resolve these ambiguities, choosing the proper tag for the word.

This table reports the problem of ambiguity, tag number per word for ambiguous corpus [5], [10].

Table 2. The amount of tag ambiguity for word types in a ≈ 135000 words corpus.

Tag per word	% in Arabic corpus
Unambiguous (1 tag per word)	24.15%
Ambiguous (more than 1 tag per word)	24.15%

Most tagging algorithms fall into one of two classes : Rule-Based taggers [3], [1] which consists of coding the necessary knowledge in a set of hand written disambiguation rules database (3) or Stochastic Taggers which resolve problem by the use of a training corpus to calculate probability of a given words and tags.

VB VB - Rule
Given tags: Vb or NN

if(-1 is Vb); /*if previous tag is Verb*/ (3)

then eliminate Vb tags

Most experimental results, in Arabic and other languages, demonstrate that statistical theories are the commenly used methods in order to obtain the best accuracy and precision in POS tagging.

Use of a Hidden Markov Model (HMM) to Part Of Speech tagging is a special case of Bayesian inference. POS tagger tries to choose the tag sequence t_1^n which is most probable given the observation sequence of n words w_1^n [24]

$$t_1^n = \arg\max P\ \ t_1^n / w_1^n\ \ . \tag{4}$$

We apply Bayes' rule, (4) will be transformed to (5), a simpler formula easier to compute.

$$t_1^n = \arg\max P\ \ w_1^n / t_1^n\ \ P\ \ t_1^n\ \ . \tag{5}$$

HMM taggers make other simplifying assumption is that the probability of a tag appearing is dependent on the n-1 previous tags. To summarize, the most probable tag sequence t_1^n given some word w_1^n can be computed by the following equation:

$$t_1^n = \arg\max P\ \ t_1^n / w_1^n\ \ \approx \arg\max \prod_{i=1}^{n} P(w_i / t_i) P(t_i / t_{i-1}, t_{i-2}, ..., t_{i-n+1}). \quad (6)$$

3.1 Part of Speech Data

The achieved POS tagger gets its input from our morphological analyzer [15], [16], [17] module which segments lexical units and gives, for each surface form, all informations consisting of lemma, lexical category and morphological inflection information.

To calculate all statistical parameters for our POS tagger, we make use of a hand-tagged corpus of Arabic sentences collection. We start by dividing the data into a training set which is approximately about 120,000 words (TrSet≈2.2Mo) and a test set (TestSet≈38ko). The training consists in estimating two types of parameters: lexical P(W/T) saved in Map file and contextual P(Ti/Ti-1,…,T-n+1) saved in language model (LM) file.

3. Lexical Model

Due to data sparseness and in order to overcome this problem, it is necessary to introduce smoothing techniques, while using with HMM [11]. All experiments illustrate that overall accuracy of the tagger improves significantly with the introduction of smoothing techniques. Thus we find that the HMM together with different smoothing methods shows improved results than ordinary Hidden Markov models.

Smoothing methods have been proposed and described in the literature, these methods aimed at re-evaluating and harmonizing the probabilities of words which are quite unlikely to occur and then assign them appropriate non-zero probabilities [11], [23]. These methods adjust downward high probabilities. Two smoothing methods was tested, Good-Turing and Chen and Goodman's modified Kneser-Ney discounting [8] [9], [13], [18], [19].

Using the same training set (TrSet) and under the identical test environment, we tried to compare different smoothing methods for different size of training corpus in order to select the best method and get which techniques improve with more data, and which get worse. The comparison results of different smoothing techniques are shown that all technique improves with more data and the Chen and Goodman's modified Kneser-Ney smoothing method gives best result.

3.3 Contextual Parameters (Language Model)

For our generative modeling approach, the first step consists of creating an N-gram language model from the tagged corpus. The estimation of the n-gram probabilities was carried out using the SRI Language Modeling toolkit1[1] [20] for language models (LM), we compare Chen and Goodman's modified Kneser-Ney discounting, as implemented by the ngram-count tool, and Good-Turing smoothing method. The LM is given in ARPA format. We measure the size of language models in total number of n-grams which is the sum of all n-gram orders [19] (n from 1 to 3).

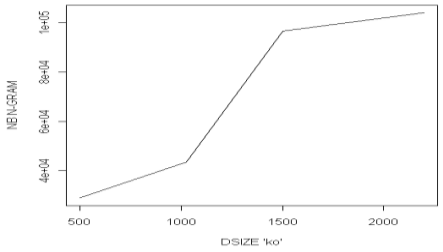


Fig. 1. Number of n-grams for varying training data size.

Now, we first describe and justify our use of perplexity as an evaluation technique. We then describe the experimental techniques used, in POS tagging, in the following sections.

The performance evaluation of a language model is usually based on the probability the language model assigns for an independent test text. The most commonly used method for measuring language model performance is perplexity which is equal to the geometric average of the inverse probability of the words measured on test data [8]:

$$\sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_i / w_1 \ldots w_{i-1})}}. \quad (7)$$

[1] SRILM*:* SRI Language Modeling Toolkit is a toolkit for building and applying statistical language models (LMs)

Perplexity has many properties that make it attractive as a measure of language model performance; among others, the "true" model for any data source will have the lowest possible perplexity for that source. Thus, the lower the perplexity of our model, the closer it is, in some sense, to the true model.

We measured perplexity for different tested smoothing techniques using various training data size and our results show that perplexity is lower with Chen and Goodman's modified Kneser-Ney, Namely when the data size is high.

In general, smoothing is an important factor in language modeling especially when the data size is too small and we haven't enough corpora; it is the case of Arabic language [14].

3.4 Viterbi Algorithm

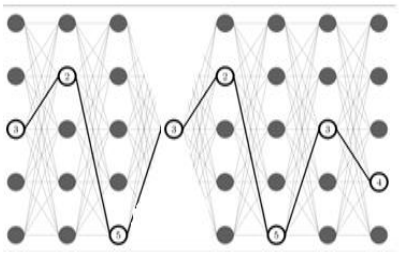
Having computed the transition and emission probabilities and assigning all possible tag sequences to all the input words, now we are in the situation to construct a lattice showing association of different tags to the input words, we need an algorithm that can search the tagging sequence and maximize the product of transition and emission probabilities. For this purpose we use Viterbi algorithm, a dynamic programming process, which compute the maximized as well as optimal tag sequence with best score.

3.5 Disambiguation Approach

In this work, a combination of statistical and linguistic approaches is employed, and our approach is to use variable windows. In other way, the text sequence $w_1 \ldots w_n$ being disambiguated is always boundered by the beginning and the end of a sentence. We disambiguate each sentence separately, because as in other languages, the sentence structure is independent from the previous and the next sentence, so there is no grammatical relation between the last word of a sentence and the first one of the upcoming sentence. Computing probabilities for these sequences have no meaning and can reduce POS tagger performance. Other point, our POS Tagger divide sentences to sub sentences delimited by unambiguous words having only one tags (see figure 2). The best sequence of tags for each sub sentence is finded by a modified classical use of viterbi algorithm. For each sequence, we try to maximize this probability (n=3):

$$\arg\max \prod_{i=1}^{n} P(w_i / t_i) P(t_i / t_{i-1}, t_{i-2}, ..., t_{i-n+1}). \quad (8)$$

This is a sample of a sentence having one unambiguous word.



Sub sentence1 Sub sentence2

Fig. 2. Sentence decomposition.

3.6 Architecture of POS Tagger

The diagram hereafter gives the architecture of our POS tagger for Arabic language. morphological system of analysis of Arab texts. The system comprises three modules: the morphological analyzer making it possible to assign with each lexeme one or several labels, the theory of probability which make it possible to assign a probability with each potential sequence of labels, and modulate it clarification making it possible to calculate the most probable sequence of labels for each sequence of the text to be analyzed.

In support of this and other works, we have described an architecture for our POS tagging system for Arabic language [16]. This architecture defines two components for disambiguation systems:

Training, in which we compute HMM parameters from an unambiguous tagged corpus (lexical model and Language model). These parameters are respectivly saved in Map file and LM file.

POS Tagger, it is implemented as an analysis module. Figure 2 illustrates the overall architecture; it attempts to choose the correct tag or lexical category, from a list of tags, for each words in a given text. This operation is achieved by the use of Viterbi Algorithm and all needed files like Map file and LM file computed in traning level.

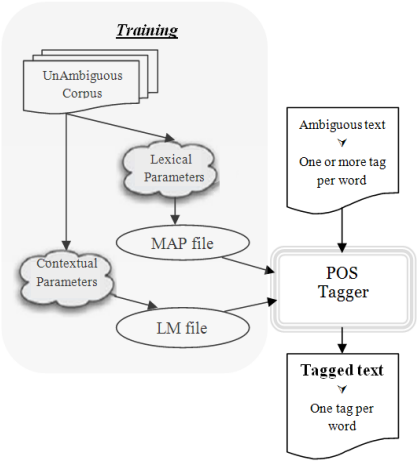


Fig. 3. Our Arabic POS tagger architecture.

3.7 Evaluation

Taggers are often evaluated by comparing them with a hand tagged test set, based on accuracy. We have prepared and tagged a test corpus (TestSet). Experimental results have been evaluated through standardize formulation: Precision, Recall and F-measure [8].

- *Precision= Correct number of token tag pair occurrence / Total number of token tag pair.*
- *Recall= Correct number of token tag pair occurrence / Number of correct token tag pair that is possible.*

$$F = 2 * \frac{precision * recall}{precision + recall}.$$

The proposed POS tagger has been tested and has achieved a state of the art performance of 96% which is very encouraging.

A confusion matrix contains information about correct and predicted tag done by the POS tagger. Performance of such systems can be represented using the following matrix. The following table shows a part of the confusion matrix.

Table3. Part of the confusion matrix

Tag	VB	NN	DT
VB	98.20%	01.40%	00.80%
NN	01.38%	92.30%	04.32%
DT	01.51%	3.39%	94.10%

4 Conclusion and Future Work

To summarize, POS tagging of Arabic language is not an area for the faint of heart or easily depressed. It is sometimes very difficult even for human to identify the correct tag for a given word. A great work was done and our results compare favorably and very promising to other recently reported works results.

As future work, we will try to proceed to test disambiguation in two steps, or in hierarchic manner, to give more information for each word. We disambiguate the first level of tags and then the second. For example, for the word "ktb" it can be a VB, as word class, while it can be also a NN. The first level of disambiguation will disambiguate the word in its context and return us "ktb" as VB. The second steps decide if this VB is in the Past or in its Passive Form. This choice is due to the use of our morphological analyzer in platform of language learning for Arabic which needs largest informations about words in text to allow the automatic generation of pedagogical activities.

Other future work is to combine taggers in series (rule-based-tagger and stochastic tagger) by using the rule-based approach to remove some of the impossible tag possibilities for each word, and then the HMM tagger to choose the best sequence from the remaining tags. Other possibility is to use the HMM tagger to choose the best tag for each word in corpus, or multiple tags in certain cases. Carrying some ambiguity from one processing step into the next, in order not to prune good solutions, will be performed by the use of a confidence interval, and then we use rules to keep only one tag per word.

The proposed approach can also be applied to other NLP processing. We estimate that this additional work we planned will improve our system's performance even more.

Acknowledgments. This work is part of three large research projects, the first one titled "Oreillodule" which is a tool for automatic speech recognition, translation, and synthesis for Arabic language. The others are "MIRTO" & "@rabLearn": two Intelligent Computer Assisted Language Learning (ICALL) platforms. The integration of our POS tagger in both ICALL systems will help teachers to generate, automatically, pedagogical activities and scenarios for learner and to migrate towards new generation of intelligent feedback.

Glossary

Corpus, pl. corpora. A collection of text, typically representing either a random sample from numerous genres, or a large archive of text from a single source (such as the archives of a single newspaper).

Hidden Markov Model (HMM). A stochastic finite-state model whose output is known, but for which the state sequence that produced is unknown.

Informations theory. A branch of mathematics that quantifies information's content of communications, with applications that include compression, error correction, and encryption.

Language model. A statistical model defining a probability distribution over the word sequences of a language.

Smoothing. Statistical estimation technique that are used when the training data is insufficient for estimating all parameters of a large model.

Supervised (unsupervised) learning. In supervised learning, the learning algorithm is provided with training data that has been manually labeled with the correct answers. In unsupervised learning, only unlabeled data is provided.

Viterbi algorithm. An efficient algorithm which, given the output sequence produced by an HMM, computes the most-likely sequence of states that the HMM passed through.

References

1. Ahmad, A., Salah, A.: A Rule-Based Approach for Tagging Non-Vocalized Arabic Words: In The International Arab Journal of Information Technology, Vol. 6, No. 3. (2009)
2. Fatma, A., Ahmed, G.: A Hidden Markov Model –Based POS Tagger for Arabic: In JADT&06. (2006)
3. Eric, B.: Transformation-based error-driven learning and natural language processing: a case study in part of speech tagging: In Computational Linguistics 21, pages 543-565. (1995)
4. Tim, B.,: Arabic Morphological Analyzer Version 1.0. LDC: University of Pennsylvania. (2002)
5. Debili, F., Emna, S: Etiquetage grammatical de l'arabe voyellé ou non: In Proceedings of the Workshop on Computational Approaches to Semitic Languages, Montreal, Quebec, Canada. (1998)
6. Mona, D., Kadri, H., Daniel, J.: Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks: In HLT-NAACL, pages 149-152. (2004)
7. Freeman, A.: Brill's POS tagger and a morphology parser for Arabic: In ACL'01 Workshop on Arabic language processing. (2001)
8. Goodman, J.: A bit of progress in language modeling: Extended Version, Microsoft Research Technical Report MSR-TR-72. (2001)
9. Jelinek, F.: Aspects of the Statistical Approach to Speech Recognition: In IEEE International Symposium on Information Theory, Washington D.C. (2001)
10. Karin, C.R.: A Reference Grammar of Modern Standard Arabic: Cambridge University Press, ISBN: 0521777712, 736 pages. (2005)

11. Katz S.M.: Estimation of probabilities from sparse data for the language model component of a speech recognizer: In IEEE Transactions on Acoustics, Speech and Signal Processing, volume ASSP-35, pages 400-401. (1987)
12. Khoja, S.: APT: Arabic part-of-speech tagger: In proceeding of the Student Workshop at the 2nd Meeting of the NAACL, (NAACL'01), Carnegie Mellon University, Pennsylvania. (2001)
13. Kneser R., Ney H.: Improved backing-off for n-gram language modeling: In Conference on Acoustics, Speech and Signal Processing, volume 1, pages 181-184. (1995)
14. Maamouri, M,. Ann, B.: Developing an Arabic treebank: Methods, guidelines, procedures, and tools: In Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages (COLING), Geneva. (2004)
15. Mourad, M., Mohamed, B.: Development Of A Morphological Analyzer For Arabic Language, Tool For Creation Of Educational Activities for Training Of Arabic: In International Conference in Technology Enhanced Learning (TEL) in Working Context, Grenoble France. (2006)
16. Mourad, M., Mounir, Z., Mohamed, B., Anis, Z., Georges, A.: A Semantic Analyzer for the Comprehension of the Spontaneous Arabic Speech: In 9th International Conference on Computing CORE08, Journal Research in Computing Science (Journal RCS), ISSN: 1870-4069, Vol 34, pp 129-140, CORE0, Mexico. (2008)
17. Mourad, M., Georges, A., Mounir, Z.: Nouvelles ressources et nouvelles pratiques pédagogiques avec les outils TAL: In TICEMED 08, Journal of Information Sciences for Decision Making (ISDM Journal), ISDM32, N°571. (2008)
18. Ney, H., Essen, Kneser, R.: On structuring probabilistic dependencies in stochastic language modeling: In Computer Speech and Language, volume 8(1), pp. 1-28. (1994)
19. Rosenfeld, R.: Two decades of Statistical Language Modeling: Where Do We Go From Here? In Proceedings of the IEEE, 88(8) (2000)
20. Stolcke, A.: SRILM - An Extensible Language Modeling Toolkit: in Proceeding of International Conference in Spoken Language Processing, Denver, Colorado. (2002)
21. Yahya, O., Mohamed, E.: Statistical Part-of-Speech Tagger for Traditional Arabic Texts: Journal of Computer Science 5 (11): 794-800,. (2009)
22. Yamina, T.G.: Hybrid Method for Tagging Arabic Text: Journal of Computer Science 2 (3): 245-248, ISSN 1549-3636. (2006)
23. Witten, I.H., Bell, TC: The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression: IEEE Transactions Information Theory, vol. 34, number 4, pp. 1085-1094. (1991)
24. Waqas, A., Xuan, W., LuLi, X.: Hidden Markov Model Based Part of Speech Tagger For Urdu: Information Technology Journal Vol: 6, 1190-1198. (2007)