British Journal of Applied Science & Technology
19(1): 1-10, 2017; Article no.BJAST.29754
ISSN: 2231-0843, NLM ID: 101664541

SCIENCEDOMAIN *international*
www.sciencedomain.org

SCIENCEDOMAIN

Choosing the Optimal Segmentation Level for POS Tagging of the Quranic Arabic

Fadl Mutaher Ba-Alwi[1*], Mohammed Albared[1] and Tareq Al-Moslmi[2]

[1]Faculty of Computer and Information Technology, Sana'a University, P.O.Box 1247, Yemen.
[2]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Malaysia.

Authors' contributions

This work was carried out in collaboration between all authors. Author FMBA designed the study, performed the statistical analysis, wrote the protocol, wrote the first draft of the manuscript and managed the literature searches. Authors MA and TAM managed the analyses of the study and literature searches. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/BJAST/2017/29754
Editor(s):
(1) Kleopatra Nikolopoulou, School of Education, University of Athens, Athens, Greece.
Reviewers:
(1) Azman Bin Che Mat, UiTM Terengganu, Malaysia.
(2) Sazelin Arif, Universiti Teknikal Malaysia Melaka (UTeM), Malaysia.
Complete Peer review History: http://www.sciencedomain.org/review-history/17817

Original Research Article

Received 27th September 2016
Accepted 11th November 2016
Published 13th February 2017

ABSTRACT

As a morphologically rich language, Arabic poses special challenges to Part-of-Speech (POS) tagging. Words in Arabic texts often contain several segments; each has its own POS category. The choice of the segmentation level or the input unit, word-based or morpheme-based, is a major issue in designing any Arabic natural language processing system. In word-based approaches, words are used the atomic units of the language. In this case, composite POS tags are assigned to words. Therefore, large amounts of training data are required in order to ensure statistical significance. They suffer from the problems of data sparseness and unknown words. In case of morpheme-based approaches, morpheme components of words are used as the atomic units. This, however, results in high level of ambiguity rate and also small size of context for resolving such ambiguity because the span of the n-gram might be limited to a single word. This paper compares and contrasts the morpheme-based and word-based statistical POS tagging strategies. This paper evaluates the tagging performance of three statistical models, namely, the Arabic HMM POS tagger with the prefix guessing models, the Arabic HMM POS tagger with the linear interpolation guessing models and the TnT tagger, given training data from both morpheme-based and word-based tokenization levels. It also studies the influence of each choice on the

*Corresponding author: E-mail: dr.fadlbaalwi@gmail.com;

tagging performance of the Arabic POS tagging models, in terms of the tagging accuracy and the time complexity. In addition, this paper also evaluates the tagging performance of several stochastic models, given training data from both segmentation levels. Results show that the morpheme-based POS tagging strategy is more adequate for the purpose of training statistical POS tagging models as it provides a better overall tagging accuracy and a much faster training and tagging time.

Keywords: Arabic natural language processing; POS tagging; segmentation levels.

1. INTRODUCTION

Part of Speech (POS) disambiguation is the ability to computationally determine which POS of a word is activated by its use in a particular context [1]. Automatic text tagging is an important pre-processing step in many NLP applications. Arabic language is a morphologically rich language which offers some challenges to Natural Language Processing (NLP) systems due to the many forms a word can take, which leads to data sparseness (the insufficiency of data). Most of the current researches in NLP are based on supervised machine learning techniques in which the classifier learns from training sets which contain a fair amount of words and their associated annotation. These classifiers need a huge amount of training data to get a reasonable accuracy even with less morphological languages such as English. In morphologically rich languages, as the classifier will be faced by many forms of the same word that do not repeat enough for the tagger to learn the pattern (data sparseness problem). These languages have a high vocabulary growth rate which results in a large number of unknown words [2].

In Arabic and also in other Semitic languages, a word, a single orthographic space-delimited string, often consists of a concatenation of sub-tokens, up to four sub-tokens [3], which function as free morph-syntactic units, each sub-token with its own POS category. In fact, Arabic word consists of proclitics, stem with affixes (prefixes and suffixes) and enclitics. The clitics (proclitics and enclitics) have their own POS tags. Following previous works, the terms morpheme-level tagging pertain to morphemes as the word-segments which are assigned POS tags from a given tag set. According to this, the Arabic word “فَبِوُعُودكُم” (and + by/with + promises + your) consists of four morphemes (sub-tokens) “ف-ب-وعود-كم”. The POS of this word is a composite POS tag (Conj+Prep+Noun+Poss.Pron). Consequently, when designing POS taggers or any NLP application for Arabic language and other Semitic languages, a major architectural decision concerns the choice of whether we should analyze a word as a sequence of morphological units (morpheme-based) or we should treat space-delimited words as the primitive units of our analyses (word-based) [2,4]. From theoretical point of view, both methods have advantages and disadvantages. The use of the morpheme-based approach increases the level of ambiguity but it increases the coverage level and decreases the size of the unknown words. On the other hand, the word-based approach suffers from the data sparseness and large size of unknown words and large tag set with composite tags problems, and but it reduces less ambiguity.

In addition, the word formation process for Arabic words is quite complex. While the main formation process of English word is concatenative, the main word formation process in Arabic languages is non-concatenative [2,5]. As a Semitic language, the word in Arabic language can be described as combinations of two morphemes: a root and pattern. A root is a set of consonants (also called radicals) which has a basic lexical meaning. A pattern consists of a set of vowels which are inserted among the consonants of a root to form a stem. In addition to this non-concatenative morphological feature, Arabic uses different affixes to create inflectional and derivational word forms. Thus, the direct adoption of the NLP methods which are developed for western languages for Arabic is not an appropriate choice due to the specific features of the Arabic language [6].

The purpose of this paper is, therefore, to explore the influence of the different segmentation levels on the tagging performance, in terms of accuracy and time complexity, of the Arabic POS tagging models in order to determine the best segmentation level to be used for POS tagging when small amount of training data is available and a large size of unknown words exist in the test data. In addition, this paper evaluates the tagging performance of three fully supervised statistical models, namely, the Arabic HMM POS tagger with the prefix guessing models, the Arabic HMM POS tagger with the linear interpolation guessing models and the TnT tagger (Arabic version), given training data from both tokenization levels.

The rest of the paper is organized as follows, Section. 2 discuss related works. Section 3 describes the used corpora. Section 4 describes the HMM tagging approaches and also discusses the modifications to better handling unknown words POS tagging in Arabic text. Section 5 gives experimental results and discusses them. Finally, conclusions and future work appear in Section 9.

2. MATERIALS AND METHODS

2.1 Related Work

In Research on POS tagging has a long history. Numerous approaches have been successfully applied to POS tagging. The POS tagging techniques in the literature can be classified into the following:

- Rule-based POS tagging: this approach is based on a lexicon and a set of disambiguation rules [7,8].
- Supervised POS tagging: these approaches use machine-learning techniques to learn a classifier from labeled training sets such as maximum entropy model [9], Hidden Markov model [10], conditional random field [11], cyclic dependency networks [12] and support vector machine [13].
- Unsupervised POS tagging: these approaches do not require pre-tagged training data, but rely on dictionary information.

However, POS tagging for Arabic language has been an active topic of research in recent years. AlGahtani et al. [14] Yousif and Sembok [15], Al-Taani and Abu Al-Rub [16], Zribi et al. [17] and Alqrainy [18] are some examples for this line of work on Arabic. Similar to this work, the selection of the best segmentation level problem, using morphemes or words as input units in Semitic language NLP, has been studied before by [2,4,19,20]. Bar-Haim et al. [4,19] study the choice of the optimal architecture for the Hebrew POS tagging and other Semitic languages. They show that a model whose terminal symbols are word segments (morphemes), is advantageous over a word-level model for the task of POS tagging. Tachbelie [2] explored different ways of language modelling for Amharic, a morphologically rich Semitic language, using morphemes as units. The study showed that using morphemes in modelling morphologically rich languages is advantageous, especially in reducing the OOV rate. In contrast with these result, Mohamed and Kübler [21] and Kübler and Mohamed [20] come with different results and different conclusion. They state that word-based POS tagging approach is more appropriate than morpheme-based POS tagging approach for modern standard Arabic POS tagging. Unlike Mohamed and Kübler [21], this work evaluates the influence of the segmentation level on the tagging performance of the tagging models given a data from the Quranic Arabic (Classic Arabic). Ali and Jarray [22] used the Genetic algorithm to develop an Arabic part of speech tagging. They used a reduced tagset in their tagger. Hadni et al. [23] propose a Hidden Markov Model (HMM) integrated with Arabic Rule-Based method. Their POS tagger generates a set of three POS tags: Noun, Verb, and Particle. Albared et al. [24] present an approach based on the combination of several N-attributes probabilistic classifiers. First, the POS disambiguation problem is decoupled into several N-attributes tagging sub-problems. Then, several classifiers are used to solve each sub-problem. Finally, the outcomes of all N-attributes classifiers are combined. Several problem decomposition methods and classifiers combination algorithms are investigated. Kadim and Lazrek [25] present bidirectional HMM-based Arabic POS tagging in which they combine both direct and reverse taggers to tag the same sequence of words in both senses.

This work also evaluates the influence of the segmentation level on the tagging performance, not only on term of the tagging accuracy but also on term of the tagging time complexity. Moreover, this work evaluates the tagging performance of several fully supervised statistical tagging models, developed especially for Arabic text.

2.2 Methodology

The probabilistic tagging models used in this work are based on the trigram Hidden Markov Model (HMM). The HMM tagger assign a probability value to each pair $<w_1^n, t_1^n>$, where

$w_1^n = w_1, \ldots, w_n$ is the input sentence and $t_1^n = t_1, \ldots, t_n$ is the POS tag sequence.

In HMM, the POS problem can be defined as the finding the best tag sequence t_1^n given the word sequence w_1^n . The label sequence t_1^n generated by the model is the one which has highest probability among all the possible label sequences for the input word sequence. This is can be formally expressed as:

$$t_1^n = \arg \max_{t_1^n} \prod_1^n p(t_i | t_{i-1}, \ldots, t_1) p(t_i | w_i)$$

The first parameter $p(t_i | t_{i-1}, \ldots, t_1)$ is a known as the transition probability and second parameter $p(t_i | w_i)$ is known as the emission probability. These two model parameters are estimated from annotated corpus by Maximum Likelihood Estimation (MLE), which is derived from the relative frequencies. Given these two probabilities, we can find the most likely tag sequence for a given word sequence using the Viterbi algorithm. However, MLE is a bad estimator for statistical inference because data tends to be sparse. To handle the sparseness problem in this work, we use linear interpolation of unigram, bigram and trigram maximum likelihood estimates in order to estimate the trigram transition probability:

$$p(t_3 | t_2, t_1) = \lambda_1 p(t_3) + \lambda_2 p(t_3 | t_2) + \lambda_3 p(t_3 | t_2, t_1)$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$, so p represents a valid probability distribution. λs are estimated by deleted interpolation. To create a HMM POS tagger that can accurately tag unknown words, it is necessary to determine an estimate of the probability $p(w_i | t_j)$ for use in the tagger. As known, if a word does not occur in the training data the $p(w_i | t_j)$ lexical probability for that word is 0 for all t_j . This requires adding an algorithm to the HMM to approximate the probability that the current tag will emit given unknown words [10]. To handle the unknown words, we have used the following the suffix Probability algorithm [26], the prefix probability algorithm and the linear interpolation guessing algorithm [27].

2.3 Dataset

The data used in this work is the Quranic Arabic Corpus [28]. The Quranic Arabic Corpus is an annotated linguistic resource which shows the Arabic grammar, syntax and morphology for each word in the Holy Quran, the religious book of Islam which is written in classical Quranic Arabic (c. 600 CE). The research project is organized at the University of Leeds, and is part of the Arabic language computing research group within the School of Computing. The Quranic Arabic Corpus is consisting of 77,430 words of Quranic Arabic. For the purpose of this work, we have used two versions from the Quranic Arabic Corpus:

- The word-based version: An example from this version is shown in Table 1. The composite tag is consisting of multiple tags separated by "+", a tag for each word segment. The composite tag set is consisting of 375 tags.
- The morpheme-based version: An example from this version is shown in Table 1. The tag set of this version consists of 45 simple tags.

A brief statistical summary (the total number of words, the total number of unique words and the tag set) of the two versions are shown in Table 2.

Table 1. Examples from the word-based version and the morpheme-based version of the Quranic corpus

The word-based version		**The morpheme-based version**	
Word	**POS**	**Word**	**POS**
<V>		<V>	
الذين	REL	الذين	REL
يؤمنون	V+PRON	يؤمن	V
بالغيب	P+DET+N	ون	PRON
		ب	P
		ال	DET
		غيب	N

Table 2. Statistical summary of the two versions

	Number of words	Unique words	Size of tag set
Word-based version	77401	14835	375
Morpheme-based version	128219	7251	45

3. RESULTS AND DISCUSSION

In this section, we report an empirical comparison between the two levels of the segmentation presented in the previous sections, and also study the influence of the two segmentation levels on the tagging performance of Arabic POS tagging models when only small amount of training data is available.

3.1 Experimental Setting

The two training data are split into two sets, training set and testing set. Essentially, we have divided the word-based version randomly into 90.25% (69980 words, 5700 sentence) for training and 9.75% (7550 words, 536 sentences) for testing. The test data are chosen independently from the training data.

After that, the morpheme-based version is divided using the same setting, see Table 3. As shown from the table, the number of vocabularies is larger in case of the morpheme-based version than in the word-based version even when the training and testing sets are equally in both versions.

Furthermore, in order to study the effect of the size of the training data, we randomly portioned our training data from the two versions to construct seven training sets. Table 4 shows sizes of the training data sets and percentages of unknown words with respect to the test data set. The test set is the same as test set for all experiments. Although each training set from the morpheme-based version contains the same data as in its equivalent in the word-based version, the number of words and the percentages of unknown words are different. It is interesting to note that the number of words are larger and the percentages of unknown words are less in case of training sets which come from the morpheme-based version than their word-based counterparts (contains the same sentences).

3.2 Results and Discussion

First of all, several experiments are conducted using the TnT model. Table 5 presents the results (known accuracy, unknown words accuracy and the overall accuracy) obtained for each training data set from the two versions: the word-based version and the morpheme-based version. We can note that the unknown word accuracy of the TnT tagger over training data sets from the Word-Based Version are so low and it does not show any sensitivity to the increase of data size. However, an overall accuracy of 88.1% (96.2% on known words and 37.7% on unknown words) is obtained when the whole training data are used (training set 7).

Table 3. Statistical summary of the training and testing data from the two versions of the Quranic corpus

	Word-based version		Morpheme-base version	
	Training	**Testing**	**Training**	**Testing**
Percentage	90.25%	9.75%	89.1%	10.9%
# of sentences (verses)	5700	536	5700	536
# of words	96850	7750	115690	12529
# of unique words	13920	2855	6924	1820

Table 4. The sizes of the training sets from the two versions of the Arabic Quranic corpus, and the percentage of unknown words in each set with respect to the test set

Training set	Word-based version		Morpheme-based version	
	Training size	**% of unknown words**	**Training size**	**% of unknown words**
1	10000	33.5%	16673	12.08%
2	19997	26.27%	33181	8.28%
3	29990	21.82%	49881	6.44%
4	40002	18.88%	66427	5.07%
5	49997	16.79%	82958	4.2%
6	60000	14.41%	99511	3.62%
7	69851	13.55%	115690	3.28%

Using training data sets from the morpheme-based version, unknown words tagging results of the TnT tagger are much better than its results over those from the Word-Based Version. However, an overall accuracy of 93.8% (of 95.6% on known words and 73.4% on unknown words) is obtained when the whole training data are used.

In general, given TnT as tagging model, morpheme-based POS tagging yields much better results than full word- based tagging (93.8% vs. 88.41%).

Secondly, several experiments are conducted using the Arabic HMM POS tagger with the prefix guessing model. Table 6 presents the results obtained for each training data set from the two versions.

It has been observed from both Tables 5 and 6 that the Arabic HMM POS tagger with the prefix guessing model always performs significantly better than TnT tagger with the suffix guessing model regardless of the segmentation level used and also regardless of the training data set sizes.

The results in Tables 5 and 6 (the overall tagging results) also show that the morpheme-based POS tagging always yields much better results than the Word-based tagging regardless of the tagging model and the size of the training data set used.

It is very interesting to note that the word-based POS tagging produces slightly better known word accuracy than those of the morpheme-based POS tagging. This is actually due to that the morpheme-based approach increases the level of ambiguity. On the other hand, the morpheme-based POS tagging produces much better unknown word accuracy than those of the word-based POS tagging. In fact, these results show that dealing with segmentation as separate pre-processing step (using segmented text) is better for handling unknown words and for POS tagging in general especially when training data is small.

In addition, we compare the computational time cost (training and testing) of two POS tagging models (TnT tagger and the Arabic HMM POS tagger with prefix guessing model) when they are trained using different sized training data sets from the two versions: the word-based version and the morpheme-based version. First, we have found that both the TnT POS tagger and the Arabic HMM POS tagger with the Prefix guessing model have approximately the same computational time (training and testing) when they are trained and tested using the same training and test data. This means that both taggers are equally efficient with respect to the execution time. Due to this, we only study here the computational time cost of the Arabic HMM POS tagger with the Prefix guessing model when it is trained using different sized training data sets (and therefore different percentages of unknown words) from the two segmentation level approaches (and therefore different sizes of tag sets): the word-based version and the morpheme-based version.

Figs. 1 and 2 show the curves of the average training and testing time taken by the Arabic HMM POS tagger with the Prefix guessing model when it is trained using different sized training data sets from the two tokenization levels.

Table 5. Tagging accuracies of the TnT Tagger with the varying size of the training data form the two training Quranic versions

Tainting set	Word-based version			Morpheme-based version		
	Unknown	Known	Overall	Unknown	Known	Overall
1	37.5	91.9	73.2	68.1	92.3	86.7
2	39.2	94.2	79.4	72.4	94.2	90.5
3	38.5	94.8	82.2	72.1	94.5	91.4
4	36.6	94.9	83.7	71.7	94.9	92.3
5	37.3	95.5	85.5	71.9	95.3	92.9
6	38.2	96.0	87.6	72.0	95.5	93.5
7	37.7	96.2	88.1	73.9	95.6	93.8

Table 6. Tagging accuracies of the Arabic HMM tagger with the prefix guessing model with the varying size of the training data form the two training Quranic versions

Tainting	Word-based version			Morpheme-based version		
set	Unknown	Known	Overall	Unknown	Known	Overall
1	69.8	92.2	84.5	77.5	92.7	89.2
2	70.5	94.5	88.1	78.5	94.3	91.6
3	71.6	95.1	89.9	81.4	94.7	92.8
4	71.9	95.2	90.7	83.1	95.0	93.7
5	72.5	95.7	91.7	85.7	95.4	94.4
6	74.7	96.1	93.0	85.6	95.6	94.8
7	75.0	96.2	93.2	87.0	95.6	95

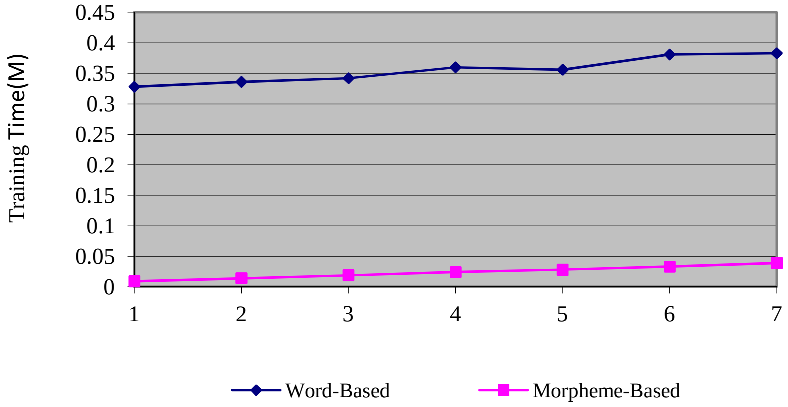

Fig. 1. The training time taken by the Arabic HMM POS tagger trained using different sized training data sets from both tokenization levels

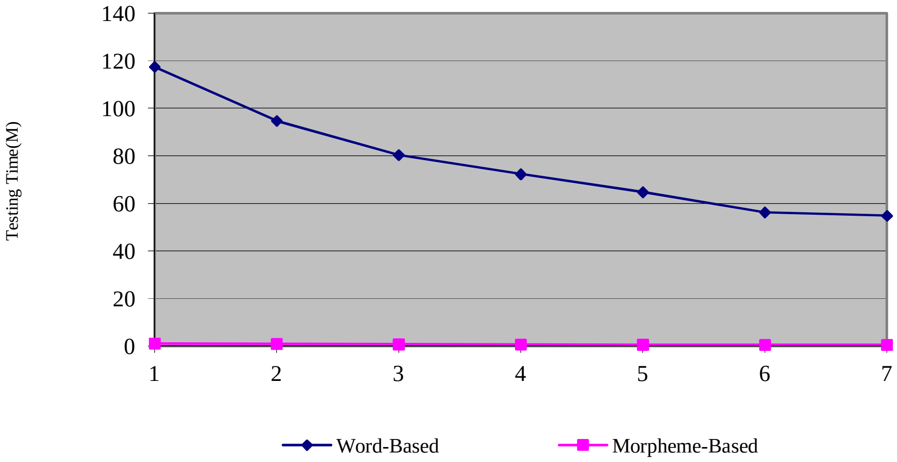

Fig. 2. The testing time taken by the Arabic HMM POS tagger trained using different sized training data sets from both tokenization levels

Table 7. The tagging performance (Time and accuracy) of the Arabic HMM POS tagger with the linear interpolation guessing model for each one of the two tokenization levels

Corpus	% of	Best	Time in minute		Accuracy		
	unknown	λ	Training	Testing	Unknown	Known	Overall
Word-based	13.5	0.9	0.40	51.1	75.6	96.2	93.4
Morpheme-based	8.02	0.7	0.03	0.58	87.4	95.6	95.0

From Figs. 1 and 2, we can draw several important observations. First, the training time is much lower than the testing time in spite of the training data set used and the corpus version used. Second, the training time in case of a training data set from the morpheme-Based version is lower than the training time in case of its counterpart from the word-based version. Third, the training time increased as the training data increased, see Fig. 1, and the testing time decreased as the training data increased, see Fig. 2. The explanation of this is that as the training data increased, the size of unknown words in the test data are substantially decreased, see Table 4, therefore less exceptional processing time and less tagging time. In fact, there is a strong positive correlation of 0.99 between the testing time and percentages of unknown words in the test sets regardless of the tokenization level used, which indicates that tagging time and the percentage of unknown words go in same directions.

Fourth, it is most importantly to note that the testing time of the word- based POS tagging (≈ 1hours to ≈ 2hours) is much larger than the testing time of the morpheme-based POS tagging (few seconds). From Figs. 1 and 2, we can readily observe that morpheme-based POS tagging would be an optimal choice as its tagging time is much larger than the tagging time of the word-based POS tagging.

Finally, several experiments are conducted using our HMM tagger with the linear interpolation guessing model which is trained using the whole training data (training set 7) from the two corpus versions. Varying the λ value from 0.0 to 1; the value is incremented by 0.1 each time. Table 7 summarizes the tagging results, the computational time needed and the best λ at which the model can give the best result, for each one of the two segmentation approach. The results also show morpheme-based POS tagging always yields better results than word- based tagging. In addition, as in previous models (TnT and Arabic Trigram HMM tagger with prefix guessing model) the tagging time of the word-based POS tagging (51 minutes) is much larger than the tagging time of the morpheme-based POS tagging (few seconds). Moreover, the linear interpolation guessing model performs better than the two previous models (TnT and Arabic HMM POS tagger with the prefix guessing model) for both tokenization levels.

4. CONCLUSION

Designing a POS tagger for Arabic with small training data is a challenging task due to the specific features of the Arabic language and the high degree of ambiguity in Arabic. In this paper, we compare and contrast morpheme-based POS and word-based POS tagging strategies and study the influence of each on the tagging performance of the Arabic POS tagging models, on term of the tagging accuracy and the time complexity. In addition, we also evaluate and compare several stochastic tagging models. We conducted a series of experiments using two versions of the Quranic Arabic corpus: morpheme-based version and word-based version. Results show that tagging models performs significantly better when their terminal symbols are word segments (morpheme-based), than when their terminal symbols are word (word-based).

In addition, the results show that the Arabic Trigram HMM POS tagger with the linear interpolation guessing algorithm substantially improve the tagging results over the TnT tagger regardless of the tokenization level used. However, our future direction is to study the influence of the segmentation level on another Arabic NLP process. Moreover, we plan to design a joint segmentation and POS tagging model which do both tasks simultaneously.

.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Albared M, Omar N, Ab Aziz MJ. Developing a competitive HMM arabic POS tagger using small training corpora. In Proceedings of the Third International Conference on Intelligent Information and Database Systems - Volume Part I, Daegu, Korea. 2011;288-296.
2. Tachbelie MY. Morphology-based language modeling for amharic. Ph.D., Department of Informatics, University of Hamburg; 2010.
3. Attia MA. Arabic tokenization system. Presented at the Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues

and Resources, Prague, Czech Republic; 2007.

4. Bar-Haim R, Sima'An K, Winter Y. Part-of-speech tagging of modern Hebrew text. Natural Language Engineering. 2008;14: 223-251.
5. Beesley KR, Karttunen L. Finite-state non-concatenative morphotactics. Presented at the Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, Hong Kong; 2000.
6. Farghaly A, Shaalan K. Arabic natural language processing: Challenges and solutions. ACM Transactions on Asian Language Information Processing (TALIP). 2009;8:1-22.
7. Loftsson H. Tagging Icelandic text: A linguistic rule-based approach*.* Nordic Journal of Linguistics. 2008;31:47-72.
8. Brill E. A simple rule-based part of speech tagger. Presented at the Proceedings of the Third Conference on Applied Natural Language Processing, Trento, Italy; 1992.
9. Ratnaparkhi A. Maximum entropy models for natural language ambiguity resolution. Ph.D., Computer and Information Science,University of Pennsylvania; 1998.
10. Thede SM, Harper MP. A second-order Hidden Markov Model for part-of-speech tagging. Presented at the Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, College Park, Maryland; 1999.
11. Lafferty JD, McCallum A, Pereira FCN. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Presented at the Proceedings of the Eighteenth International Conference on Machine Learning; 2001.
12. Toutanova K, Klein D, Manning CD, Singer Y. Feature-rich part-of-speech tagging with a cyclic dependency network. Presented at the Proceedings of NAACL '03, Edmonton, Canada; 2003.
13. Giménez J, Màrquez L. SVMTool: A general POS tagger generator based on support vector machines. In Proceedings of 4th International Conference on Language Resources and Evaluation (LREC), Lisbon, Portugal. 2004;43-46.
14. AlGahtani S, Black W, McNaught J. Arabic Part-of-speech tagging using transformation-based learning. Presented at the Proceedings of the Second International Conference on Arabic Language Resources and Tools, Cairo, Egyp; 2009.
15. Yousif JH, Sembok T. Arabic part-of-speech tagger based support vectors machines. In Information Technology, 2008. ITSim 2008. International Symposium. 2008;1-7.
16. Al-Taani A, Abu Al-Rub S. A rule-based approach for tagging non-vocalized Arabic words. The International Arab Journal of Information Technology. 2009;9: 320-328.
17. Zribi C, Torjmen A, Ahmed M. A multi-agent system for POS-tagging vocalized Arabic text. The International Arab Journal of Information Technology; 2007.
18. Alqrainy S. A morphological-syntactical analysis approach For Arabic textual tagging. Ph.D., De Montfort University, Leicester, UK; 2008.
19. Bar-Haim R, Sima'an K, Winter Y. Choosing an optimal architecture for segmentation and POS-tagging of modern Hebrew. Presented at the Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, Ann Arbor, Michigan; 2005.
20. Kübler S, Mohamed E. Part of speech tagging for Arabic. Natural Language Engineering. First View. 2011;1-28.
21. Mohamed E, Kübler S. Is Arabic part of speech tagging feasible without word segmentation? Presented at the The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, California, USA; 2010.
22. Ali BB, Jarray F. Genetic approach for Arabic part of speech tagging. International Journal on Natural Language Computing. 2013;2:1-12.
23. Hadni M, Ouatik S, Lachkar A, Meknassi M. Hybrid Part-of-speech tagger for non-vocalized Arabic text. International Journal on Natural Language Computing (IJNLC). 2013;2.
24. Albared M, Hazaa M. N-attributes stochastic classifier combination for Arabic morphological disambiguation. Saba Journal of information Technology And Networking (SJITN). 2015;3.
25. Kadim A, Lazrek A. Bidirectional HMM-based Arabic POS tagging. International Journal of Speech Technology. 2016;19: 303-312.
26. Brants T. TnT: A statistical part-of-speech tagger. Presented at the Proceedings of

the sixth conference on Applied natural language processing. Seattle, Washington; 2000.

27. Albared M, Omar N, Ab Aziz MJ, Nazri MZA. Automatic part of speech tagging for Arabic: An experiment using Bigram hidden Markov model. Presented at the Proceedings of the 5th International Conference on Rough Set and Knowledge Technology, Beijing, China; 2010.

28. Dukes K, Atwell E, Sharaf ABM. Syntactic annotation guidelines for the quranic Arabic Dependency Treebank. Presented at the Language Resources and Evaluation Conference (LREC 2010), Valletta, Malta; 2010.

Peer-review history:
The peer review history for this paper can be accessed here:
http://sciencedomain.org/review-history/17817