SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON
MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

VOLUME 5 NO 4, 2017
ISSN 2054-7390

Feature-rich PoS Tagging through Taggers Combination : Experience in Arabic

Imad Zeroual and Abdelhak Lakhouaja
Computer Sciences Laboratory, Faculty of Sciences, Mohammed First University, Oujda, Morocco
mr.imadine@gmail.com, abdel.lakh@gmail.com

ABSTRACT

Since words can play different syntactic roles in different contexts, it is not trivial to assign the appropriate morphosyntactic category to each word according to the context. Part of Speech (PoS) tagging is the task which manage this issue. Several probabilistic methods have been adapted for PoS tagging such as Hidden Markov Models, Support Vector Machines, and Decision Tree. Based on these methods, language-independent PoS taggers have been developed such as TnT, SVMTool, and Treetagger. The main purpose of this work is to combine automatically the output of these standard PoS taggers and investigate several options for how to do this combination. The experiments are applied to one of the morphologically complex languages, Arabic. In this paper, we highlight the use of these taggers via various experiments. In fact, the evaluations involve several tests on both Classical and Modern Standard Arabic, trained/untrained and tagged/untagged data. Finally, a deeper investigation of Arabic PoS tagging through these language-independent taggers combination is performed.

Keywords: Part of speech, Tagging, Treetagger, SVMTool, TnT, Arabic.

1 Introduction

The Part of Speech tagging is the basis of well-known natural language processing (NLP) fields. It is a preliminary task that reflects directly the performance of any other subsequent text processing [1]. Further, pos tagging is a key input feature for other NLP tasks. For example, it is very useful for spell checking and correcting, named entity recognition, information retrieval, building dictionaries, phrase chunking, and text-to-speech synthesis systems [2].

Due to these convincing reasons, several methods have been proposed for automatic pos tagging. The most known are stochastic methods that used Hidden Markov Models (HMM), Support Vector Machines (svms), and Decision Tree (DT). Based on these methods, language-independent pos taggers have been developed. The finest freely available taggers with a considerable accuracy are tnt [3], svmtool [4], and Treetagger [5]. These taggers are data-driven which mean they learn from pre-annotated corpora and lexicon.

Arabic is a resource-poor language when it comes to finding freely available lexical resources and pre-tagged training corpora and so on. De facto, not a single tagged modern standard Arabic corpus was a freely or publicly available until the fall of 2011 [6]. Further, only a few works suggested a tagset for a

DOI: 10.14738/tmlai.54.2981
Publication Date: 15th August, 2017
URL: http://dx.doi.org/10.14738/tmlai.54.2981

standard use such as [7], [8], and [9]. Thus, to evaluate different approaches and tools on a common ground becomes hard to realize.

Basically, the standard taggers are adaptable to any language if a lexicon and a tagged training data are available. Fortunately, a recent adaptation of Treetagger for Arabic [10] is available for public . The adaptation used a universal common tagset that covers 22 different languages including Arabic [11]. In this regard, we adopt the same language model used for the Treetagger to adapt the other two taggers (TnT and SVMTool).

To construct a combination system, two or more individual taggers should be involved. Usually, a combination of taggers obtains a higher accuracy than the application of just a single tagger [12]. The reason is that different taggers eventually produce different errors and these differences can be exploited to yield better results. Thus, when building combined taggers it is important to use taggers based on different methods [13].

The aim of the present paper is to shed light on the performance of three selected taggers via various experiments. Further, an evaluation of these taggers and their combinations is performed by measuring the accuracy rate at the word level. The corpora used for the experiments are; those which the taggers trained on, Al-Mus'haf [14] and NEMLAR [15]. Al-Mus'haf corpus covers the Quranic text where all the words are annotated using a semi-automatic method by applying "AlKhalil Morpho Sys" [16] and hand-corrected by linguistics afterwards. NEMLAR is an Arabic written corpus produced and tagged by RDI, Egypt for the Nemlar Consortium. The third data is untagged and extracted from the Arabic part of a multilingual corpus [17]. Therefore, the tagging results were validated manually and compared to their combination.

Finally, all possible options for how to do these dual combinations are performed and their results is discussed according to their individual performance. The obtained results prove that some of these combinations could have a negative effect on the system performance, but others yield a better result. However, the best accuracy is achieved by extending the dual combination to a trilogy.

The remainder of the paper is organized as follows: in the second section, background information is given about Arabic PoS tagging, mentioning several relevant works. We briefly describe the three standard taggers and the universal tagset implemented in the third section. In the fourth section, various experiments are presented and we discuss the obtained results. Finally, we conclude this paper in section five.

2 Arabic PoS tagging

The PoS tagging is selecting which is necessary to resolve ambiguity during text processing to estimate the exact morphosyntactic tags to fit the actual input context.

In order to begin with PoS tagging, There are certain requirements [18]:

To select the approaches that will be used for the automatic tagging process;

To prepare the linguistic resources required for training the tagger, and optional lexicon containing all possible tags for a particular word form;

To define the tagset, i.e., basic morphosyntactic tags attached to each word.

Different methods are designed for PoS tagging:

Statistical methods: they derive the probabilities from a large pre-tagged training data. The majority of these methods are based on HMM. For example, Banko and Moore [19] presented an HMM tagger that exploits context on both sides of a word to be tagged. It has been evaluated in both the unsupervised and supervised cases. It achieves an accuracy of about 96%. Another tagger has been proposed, tested, and it achieves a performance of 97% [20]. The last examples is a recent published study by [21]. Other statistical taggers are based on SVM such as [22] and Yamcha [23] that achieves 97.6% accuracy. Finally, a maximum entropy approach, that enrich the information sources used for tagging, has been adopted by the Stanford PoS tagger [24]. Its result accuracy on the Penn Treebank [25] achieves 96.86% overall, and 86.91% on previously unseen words.

Rule-based methods: they are based on rules defined by linguistics. These methods involve morphological analysis and lexicons research. For example, Brill's PoS tagger, a rule-based tagger for Arabic which uses a machine learning approach based on Brown corpus [26]. A similar work has been developed by [27] using transformation-based learning method, which is an error-driven approach to induce the retagging rules from a training Arabic Treebank corpus and the morphological analyser BAMA [28]. The achieved accuracy is 96.9%. Finally, an Arabic PoS tagger has been developed based on sentence structure, i.e., the relation between the untagged words and their adjacent [29].

Neural network models: These methods use learning models inspired from the artificial intelligence field based on understanding the operation of biological neural networks in brains. Typically, they use highly interconnected simple processing nodes [30]. To our knowledge, no implementation of neural network models has been done for Arabic yet. As Carneiro claimed [31], the developer of the mWANN-tagger (multilingual Weightless Artificial Neural Network tagger), the ability to successfully tag languages that possess non-concatenative morphology such as Arabic is left for a future investigation.

Hybrid Systems: guessing PoS tags for unknown word is certainly the main problem in probabilistic methods. This problem becomes more serious in resource-poor languages that have rich vocabulary and complex morphology such as Arabic [32]. In order to handle this problem, several taggers combine different statistical model with rule-based method like [33], [34], and [35]. Their accuracies achieved are respectively 97.4%, 94%, and 92.86%. It should be noted that the three taggers differ in terms of the selected tagset and the resources used in both training and evaluation. A different hybrid system has been implemented by Tlili-Guiassa [36] to tag an Arabic text based on memory-based learning and rules-based method. The modus operandi of this system is to apply rules (analyzing the affixes and the patterns of a word) to determine the appropriate tag of each word in the current context, then, refer to memory-based learning as a machine learning method that can handle the exceptions of these rules.

3 Standard taggers

Standard taggers have been successfully applied to several morphologically complex languages such Arabic, which have been yielded state-of-the-art results. In the following, we introduce those most relevant ones:

- TnT tagger: TnT (short form of Trigrams'n'Tags) is a PoS tagger developed by Brants [3]. TnT tagger uses the Viterbi algorithm [37] for second-order Markov models where the transition probability depends on two preceding tags. From a training data, the probabilities of the model are estimated using maximum likelihood estimation. New assignments of tags to a word are determined by optimizing the product of lexical probabilities and contextual probabilities. The states of the model represent tags while the outputs represent the words.

- Treetagger: it uses an unknown word PoS guesser similar to that of the TnT tagger. However, it is developed to avoid problems that HMM face in transition probabilities. Thus, Treetagger estimates the transition probabilities based on a decision tree; which mean that the probability of a given trigram is determined by following the corresponding path through the tree until a leaf is reached [5]. The Treetagger is probably the widely language-independent PoS tagger used, it has been officially and successfully used to tag more than 30 different languages.
- SVMTool: Giménez and Marquez [4] proposed a standard PoS tagger based on SVMs. the tagger implement five different models for training (0 (default),1, 2, 3, and 4) with a tagging direction that can be either “left-to-right”, “right-to-left”, or a combination of both. Concerning the models training, they are based on the SVMLight implementation of the Vapnik's SVM [38], [39] by Joachims [40].

All the three taggers come with two programs. The first one is for the training task that requires a tagged training corpus and auxiliary lexicon to generate the parameter file. The second program is the tagger itself. Up to date, there is no comparative study that evaluates the three taggers for Arabic on a common ground. However, the taggers have been individually implemented and evaluated for English under the same conditions. They were trained on two million words of the Wall Street Journal section of the Penn Treebank corpus [41]. The obtained accuracy rates are 96.06%, 96.36%, and 97.16% respectively for the TnT, Treetagger, and SVMTool [4], [42]. This shows that the SVMTool outperforms the other taggers.

3.1 The tagset

A tag is a string used as a label to represent information about morphosyntactical features (case, gender, etc.) of word forms. A tagset is a set of these tags.

Generally, every proposed PoS tagger is looking for his suitable objective. Therefore, it is hard to assume that the tagsets proposed up to date are for standard use; especially for a language with a highly inflectional grammar, Arabic. Since we don't want to reinvent the wheel, the recent adaptation of Treetagger, that use a tagset based on two studies [11], [43] is thus adopted in our experiments. Table 1 below presents the basic tags of the adopted universal tagset.

Table 1. The basic tags of the universal tagset.

Tags	Tag Symbols	Tag in Arabic	Examples
1. Verbs (all tenses and modes)	VERB	فعل	“كَتَبَ" (kataba “to Write”)
2. Nouns	NOUN	اسم	“مَدْرَسَة” (madrasap “School”)
3. Proper nouns	PN	اسم علم	“مُحَمَّد” (muHam~ad “Mohamed”)
4. Pronouns	PRON	ضمير	“هِيَ” (hiya “She”)
5. Adjectives	ADJ	صفة	“جَمِيل” (jamyl “Beautiful”)
6. Adverbs	ADV	ظرف	“بَعْدَ، فَوْقَ” (baEda, fawoqa “After, Above”)
7. Utilities words (Particles, Adpositions…)	PRT	أداة	“إلى، ذلك، الذي” (<ilY, *lk, Al*y “To, That, who”)
8. Disconnected letters (Quranic Initials)	DISL	حروف مقطعة	“الم، طه، كهيعص” (Alm, Th, khyES)
9. Speech-specific sounds	Uh	حرف صوت	“آه، هيهات” (|h, hayhAt)
10. Other: foreign words, typos, abbreviations…	X	أخرى	“أوبك، مانشستر” (>wbk, mAn$str “OPEC, Manchester”)
11. Punctuation marks	SENT	علامة ترقيم	

4 Experiments and discussion

In this section, we highlight the performance of the selected taggers via various experiments on text from both Modern Standard Arabic (MSA) and Classical Arabic (CA). Further, we evaluate the performance of the taggers on trained/untrained and tagged/untagged data. In addition, all possible combinations are investigated, presenting the best combination. Finally, the achieved results are discussed.

4.1 Experiences on tagged/trained corpora

In this section, we highlight the performance of the three taggers (TnT, Treetagger, and SVMTool) and explore the commons results and different errors. At the first stage, these taggers are evaluated individually using both tagged and trained corpora. The taggers are trained and tested on data from the NEMLAR (500,000 words) and Al-Mus’haf (78,121 words) corpora. 90% of the corpora are used for training and the rest 10% for testing. Table 2 exhibits the obtained accuracies from all the taggers

Table 2. Accuracy results

* Words are not included in training data

Corpora	*Unrecognized words**	*TnT*	*Treetagger*	*SVMTool*
Al-Mus’haf	942	93.97%	**94.70%**	93.42%
NEMLAR	6,276	94.74%	**95.12%**	94.88%

As seen in Table 2, Treetagger performs better than the other taggers when they are applied on both corpora. Whereas, the achieved accuracy by TnT is slightly better than the one achieved by SVMTool when it is applied on Al-Mus’haf corpus; and vice versa when they are applied on NEMLAR corpus.

To indicate the motivation for taggers combination, deeper investigation is required. Therefore, we checked the outputs of the three taggers to explore the commons results and different errors, and eventually to exploit these observations in further tasks. Table 3 shows detailed information about taggers outputs.

Table 3. Detailed information about taggers outputs

Corpora	***Common***		***Non-common***			
Taggers	*All*		*TnT*	*Treetagger*	*SVMTool*	*All*
Al-Mus'haf	*93.72%*		*6.28%*			
	Correct	*Incorrect*	*Correct*	*Correct*	*Correct*	*Incorrect*
	92.94%	0.78%	1.03%	1.76%	0.48%	3.01%
NEMLAR	*94,61%*		*5.39%*			
	Correct	*Incorrect*	*Correct*	*Correct*	*Correct*	*Incorrect*
	93.85%	0.76%	0.89%	1.27%	1.03%	2.20%

Several hints are observed in Table 3 above:

- The common outputs are not certainly correct; yet, the rate of incorrect ones remains very low (0.76%-0.78%).
- None of the common and correct outputs of the three taggers reach the accuracy rate of the three taggers individually.
- The non-common outputs are not certainly incorrect. In fact, more than half of them are correct.
- Based on these observations, we deduce that depending only on the common outputs is not effective, because it does not reach the performance level of each tagger individually. Also, we cannot abandon the non-common outputs, where there is an interesting percentage of correct results.

4.2 Combination algorithm

Through the previous investigation, it is possible to define an appropriate combination algorithm. However, the purpose of this work is not to propose a better combination algorithm. Instead, we would like to demonstrate that a combination system does effectively improve tagging accuracy. Here, we describe the algorithm implemented for the combination process. This combination algorithm determines the most appropriate tags in three steps:

- Tagging the input text with all taggers;
- Selecting for each token the most voted tag from the majority taggers (in these experiments, at least two taggers);
- If the given tags from all taggers are unlike. Then, the selected tag is the one proposed by the most accurate tagger (in these experiments, is Treetagger).

The evaluation of the algorithm is divided into two phases. In the first one, only two taggers are used as a combination. Consequently, we left with three possible combinations; while in the second phase, the three taggers are used as a combination. Table 4 shows the achieved accuracies of all combinations in these the two phases.

Table 4. Combinations accuracies

Combinations	***TnT & Treetagger***	***Treetagger & SVMTool***	***TnT & SVMTool***	***All taggers***
Al-Mus'haf	95.73% (+)	93.54% (-)	93.82% (-)	**95.79% (*)**
NEMLAR	95.23% (+)	95.00% (-)	94.93% (+)	**96.45% (*)**

By combining the outputs of two or three taggers using the proposed algorithm, the obtained results are as follows:

- (-): these combinations achieve an accuracy rate lower than the most accurate tagger involved in the combination.
- (+): these combinations achieve an accuracy rate higher than the most accurate tagger involved in the combination.
- (*): the best achieved results in all combinations, i.e., those involve all the three taggers.

4.3 Evaluation on untagged/untrained corpus

After testing and validation of the combination algorithm on available pre-tagged and trained corpora, it remains to evaluate this algorithm on a new untagged/untrained data which is the main objective of this work. For that reason, we have selected the data from a rich resource in term of variety of domains and topics. The data are extracted from the Arabic part of a new proposed multilingual corpus constructed based on the available subtitles of TEDx talks. The size of these data is 500,000 words.

Before applying the combination algorithm, it is required to determine the most accurate tagger among the three. Based on the idea that the tagger which outperforms the others in the non-common outputs, eventually, it will be the one that has the higher accuracy in the overall corpus. Hence, the first stage of this evaluation is to annotate the corpus with the three taggers to separate the common and non-common outputs. Finally, we verify manually and validate the achieved accuracies in two experimental samples: (1) all non-common outputs; (2) 10% of random common outputs. Table 5 presents the obtained results of this task.

Table 5. Accuracy analysis on experimental samples

	Common		**Non-common**			
Taggers	**All**		**TnT**	**Treetagger**	**SVMTool**	**All**
Percentages	86.98%		13.02%			
Experimental samples	10.00%		13.02%			
Correctness	**Correct**	**Incorrect**	**Correct**	**Correct**	**Correct**	**Incorrect**
Accuracy	84.85%	2.13%	**4.03%**	4.02%	3.54%	1.43%

The observations noted in the previous evaluations (Table 3) remain the same as they are in the experimental samples presented in Table 5. Thereby, the next step is to apply the combination algorithm and compare it to the performance of each tagger. In this task, only both experimental samples (23.02%)

are used instead of the overall corpus. Table 6 exhibits the obtained accuracy rate by all the taggers and the combination algorithm.

Table 5. Taggers accuracies on TED corpus

Taggers	***TnT***	***Treetagger***	***SVMTool***	***Combination***
Accuracy	88.88%	88.87%	88.39%	**90.63%**

As seen in Table 6, the achieved accuracies of the three taggers are approximately similar with relative progress of the TnT tagger. However, the combination algorithm outperforms the three taggers individually.

4.4 Discussion

Several experiments on various resources are performed in term of text form, trained/tagged and untrained/untagged. Based on a deeper investigation of these experiments, a combination algorithm is developed. Several evaluations and validations are done to demonstrate that the combination system does effectively improve tagging accuracy considering the number of taggers involved and their performance. To sum up the most important results obtained in this investigation, we state the following points:

- As seen in the evaluation experiments, the proposed combination system performs better than the other taggers when they applied individually on all three corpora.
- Usually, the PoS tagging is done by an automatic process and manually corrected afterwards. To minimize the hand-correction, the combination algorithm can be used to improve the accuracy rate; yet, to point the candidate mis-tagged words by indicating the unlike tags.
- By combining only two taggers, an accuracy rate reduction could be achieved. In our case, the rate was lower than the most accurate tagger involved in the combination algorithm.
- Improving the performance of the current combination algorithm is at hand. For instance, the improvement still possible if the number of involved taggers is augmented or different combination algorithms are adopted.

In addition, all obtained results shows that the accuracy of a common output is always lower than that achieved by the taggers separately. The reason is that the taggers produce different errors and these differences are exploited in the combination to yield better results. Therefore, we suggest combining taggers based on different methods while building a combined system.

5 Conclusion and perspectives

In this paper, we demonstrate the feature-rich functionality of PoS tagging through taggers combination. Arabic language was the case of this investigation. The combination algorithm achieves a state-of-the-art overall accuracy in Arabic PoS tagging and outperforms other taggers.

Here, we highlight pertinent tagging methods, primarily those implemented for Arabic language, mentioning relevant works that have been published in this field. Further, three standard taggers are introduced with a brief description; then, their performance is evaluated via various experiments by measuring the accuracy rate at the word level.

As it was observed in the experiments conducted, the proposed combination algorithm involves three language-independent PoS taggers. The performance is better in comparison with the other taggers separately. The proposed system increases the accuracy rate of the most accurate tagger by 1.09%, 1.33% and 1.75% respectively for the three different corpora Al-Mus'haf, NEMLAR and the Arabic part of TED corpus. The modus operandi of our PoS combination algorithm is in accordance with the observations concluded during the various experiments that we have made.

Finally, this work is another step to improve tagging accuracy for Arabic and to minimize the hand-correction. Yet, it is still possible to move performance levels up. Later, we look forward to combine a PoS taggers dedicated to the Arabic language (e.g., [34], [44], and [45]) with an application to fine-grained PoS tagging. Further, we plan to investigate other combining approaches and their application to PoS tagging.

REFERENCES

[1] M. Albared, N. Omar, and M. J. Ab Aziz, "Developing a competitive HMM arabic POS tagger using small training corpora," in Intelligent Information and Database Systems, Springer, 2011, pp. 288–296.

[2] R. A. Abumalloh, H. M. Al-Sarhan, O. Ibrahim, and W. Abu-Ulbeh, "Arabic Part-of-Speech Tagging," J. Soft Comput. Decis. Support Syst., vol. 3, no. 2, pp. 45–52, 2016.

[3] T. Brants, "TnT: a statistical part-of-speech tagger," in Proceedings of the sixth conference on Applied natural language processing, 2000, pp. 224–231.

[4] J. Giménez and L. Marquez, "SVMTool: A general POS tagger generator based on Support Vector Machines," 2004.

[5] H. Schmid, "Treetagger| a language independent part-of-speech tagger," Inst. Für Maschinelle Sprachverarbeitung Univ. Stuttg., vol. 43, p. 28, 1995.

[6] H. S. Rabiee, "Adapting Standard Open-Source Resources To Tagging A Morphologically Rich Language: A Case Study With Arabic.," in RANLP Student Research Workshop, 2011, pp. 127–132.

[7] S. Alqrainy, "A morphological-syntactical analysis approach for Arabic textual tagging," De Montfort University, 2008.

[8] M. Sawalha, "Arabic Morphological Features Tag set." University of Leeds, 2009.

[9] I. Zeroual, A. Lakhouaja, and R. Belahbib, "Towards a standard part of speech tagset for the Arabic language," J. King Saud Univ. - Comput. Inf. Sci., 2017.

[10] Z. Imad and L. Abdelhak, "Adapting a decision tree based tagger for Arabic," in 2016 International Conference on Information Technology for Organizations Development (IT4OD), 2016, pp. 1–6.

[11] S. Petrov, D. Das, and R. McDonald, "A universal part-of-speech tagset," ArXiv Prepr. ArXiv11042086, 2011.

[12] N. P. P. Khin and T. N. Aung, "Analyzing Tagging Accuracy of Part-of-Speech Taggers," in International Conference on Genetic and Evolutionary Computing, 2015, pp. 347–354.

[13] V. Henrich, T. Reuter, and H. Loftsson, “CombiTagger: A System for Developing Combined Taggers.,” in FLAIRS Conference, 2009.

[14] I. Zeroual and A. Lakhouaja, “A new Quranic Corpus rich in morphosyntactical information,” Int. J. Speech Technol., pp. 1–8, Feb. 2016.

[15] M. Yaseen et al., “Building annotated written and spoken Arabic LR’s in NEMLAR project,” in Proceedings of LREC, 2006, pp. 533–538.

[16] M. Boudchiche, A. Mazroui, M. Ould Abdallahi Ould Bebah, A. Lakhouaja, and A. Boudlal, “AlKhalil Morpho Sys 2: A robust Arabic morpho-syntactic analyzer,” J. King Saud Univ. - Comput. Inf. Sci., 2016.

[17] I. Zeroual and A. Lakhouaja, “Towards a Multilingual Aligned Parallel Corpus,” presented at the International Conference of High Innovation in Computer Science, Kenitra, Morocco, 2016, pp. 1–4.

[18] M. Utvić, “Annotating the corpus of contemporary Serbian,” in Proceedings of the INFOtheca ‘12 Conference, 2011.

[19] M. Banko and R. C. Moore, “Part of speech tagging in context,” in Proceedings of the 20th international conference on Computational Linguistics, 2004, p. 556.

[20] F. Al Shamsi and A. Guessoum, “A hidden Markov model-based POS tagger for Arabic,” in Proceeding of the 8th International Conference on the Statistical Analysis of Textual Data, France, 2006, pp. 31–42.

[21] A. Kadim and A. Lazrek, “Bidirectional HMM-based Arabic POS tagging,” Int. J. Speech Technol., vol. 19, no. 2, pp. 303–312, 2016.

[22] M. Diab, K. Hacioglu, and D. Jurafsky, “Automatic tagging of Arabic text: From raw text to base phrase chunks,” in Proceedings of HLT-NAACL 2004: Short Papers, 2004, pp. 149–152.

[23] N. Habash and O. Rambow, “Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop,” in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, 2005, pp. 573–580.

[24] K. Toutanova and C. D. Manning, “Enriching the knowledge sources used in a maximum entropy part-of-speech tagger,” in Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13, 2000, pp. 63–70.

[25] M. Maamouri and A. Bies, “The Penn Arabic Treebank.” In Farghaly, A., Ed., Arabic Computational Linguistics. CSLI Publications, Stanford, CA., 2010.

[26] A. Freeman, “Brill’s {POS} tagger and a Morphology parser for {Arabic},” 2001.

[27] S. AlGahtani, W. Black, and J. McNaught, “Arabic part-of-speech tagging using transformation-based learning,” in Proceedings of the Second International Conference on Arabic Language Resources and Tools, Cairo, Egypt, 2009.

[28] T. Buckwalter, “Buckwalter Arabic morphological analyzer (BAMA) version 2.0. linguistic data consortium (LDC) catalogue number LDC2004L02,” ISBN1-58563-324-0, 2004.

[29] Y. El Hadj, I. Al-Sughayeir, and A. Al-Ansari, “Arabic part-of-speech tagging using the sentence structure,” in Proceedings of the Second International Conference on Arabic Language Resources and Tools, Cairo, Egypt, 2009.

[30] D. R. Wilson, “Advances in instance-based learning algorithms,” Citeseer, 1997.

[31] H. C. Carneiro, F. M. França, and P. M. Lima, “Multilingual part-of-speech tagging with weightless neural networks,” Neural Netw., vol. 66, pp. 11–21, 2015.

[32] M. Albared, T. Al-Moslmi, N. Omar, A. Al-Shabi, and F. M. Ba-Alwi, “Probabilistic Arabic part of speech tagger with unknown words handling,” J. Theor. Appl. Inf. Technol., vol. 90, no. 2, p. 236, 2016.

[33] M. Hadni, S. A. Ouatik, A. Lachkar, and M. Meknassi, “Hybrid Part-Of-Speech Tagger for Non-Vocalized Arabic Text,” Int. J. Nat. Lang. Comput., vol. 2, no. 6, pp. 1–15, 2013.

[34] N. Ababou and A. Mazroui, “A hybrid Arabic POS tagging for simple and compound morphosyntactic tags,” Int. J. Speech Technol., vol. 19, no. 2, pp. 289–302, 2016.

[35] A. H. Aliwy, “Arabic Morphosyntactic Raw Text Part of Speech Tagging System,” Repozytorium Uniwersytetu Warszawskiego, 2013.

[36] Y. Tlili-Guiassa, “Hybrid method for tagging Arabic text,” J. Comput. Sci., vol. 2, no. 3, pp. 245–248, 2006.

[37] D. L. Neuhoff, “The Viterbi algorithm as an aid in text recognition (Corresp.),” Inf. Theory IEEE Trans. On, vol. 21, no. 2, pp. 222–226, 1975.

[38] V. Vapnik, The nature of statistical learning theory. Springer Science & Business Media, 2013.

[39] V. N. Vladimir and V. Vapnik, The nature of statistical learning theory. Springer Heidelberg, 1995.

[40] T. Joachims, “Svmlight: Support vector machine,” SVM-Light Support Vector Mach. Httpsvmlight Joachims Org Univ. Dortm., vol. 19, no. 4, 1999.

[41] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, “Building a large annotated corpus of English: The Penn Treebank,” Comput. Linguist., vol. 19, no. 2, pp. 313–330, 1993.

[42] H. Schmid, “Probabilistic part-ofispeech tagging using decision trees,” in New methods in language processing, 2013, p. 154.

[43] O. Rambow et al., “Parallel syntactic annotation of multiple languages,” in Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC2006). Genoa, Italy, 2006.

[44] A. Pasha et al., “Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic,” in Proceedings of the Language Resources and Evaluation Conference (LREC), Reykjavik, Iceland, 2014.

[45] K. Darwish, A. Abdelali, and H. Mubarak, “Using Stem-Templates to Improve Arabic POS and Gender/Number Tagging.,” in LREC, 2014, pp. 2926–2931.