Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks*

Mona Diab Linguistics Department Stanford University mdiab@stanford.edu

Kadri Hacioglu Center for Spoken Language Research University of Colorado, Boulder hacioglu@colorado.edu

Daniel Jurafsky Linguistics Department Stanford University jurafsky@stanford.edu

Abstract

To date, there are no fully automated systems addressing the community's need for fundamental language processing tools for Arabic text. In this paper, we present a Support Vector Machine (SVM) based approach to automatically tokenize (segmenting off clitics), part-of-speech (POS) tag and annotate base phrases (BPs) in Arabic text. We adapt highly accurate tools that have been developed for English text and apply them to Arabic text. Using standard evaluation metrics, we report that the SVM-TOK tokenizer achieves an $F_{\beta=1}$ score of 99.12, the SVM-POS tagger achieves an accuracy of 95.49%, and the SVM-BP chunker yields an $F_{\beta=1}$ score of 92.08.

1 Introduction

Arabic is garnering attention in the NLP community due to its socio-political importance and its linguistic differences from Indo-European languages. These linguistic characteristics, especially dialect differences and complex morphology present interesting challenges for NLP researchers. But like most non-European languages, Arabic is lacking in annotated resources and tools. Fully automated fundamental NLP tools such as Tokenizers, Part Of Speech (POS) Taggers and Base Phrase (BP) Chunkers are still not available for Arabic. Meanwhile, these tools are readily available and have achieved remarkable accuracy and sophistication for the processing of many European languages. With the release of the Arabic Penn TreeBank 1 (v2.0),[1] the story is about to change.

In this paper, we propose solutions to the problems of Tokenization, POS Tagging and BP Chunking of Arabic text. By Tokenization we mean the process of segmenting clitics from stems, since in Arabic, prepositions, conjunctions, and some pronouns are cliticized (orthographically and phonological fused) onto stems. Separating conjunctions from the following noun, for example, is a key first step in parsing. By POS Tagging, we mean the standard problem of annotating these segmented words with parts of speech drawn from the 'collapsed' Arabic Penn TreeBank POS tagset. Base Phrase (BP) Chunking is the process of creating non-recursive base phrases such as noun phrases, adjectival phrases, verb phrases, preposition phrases, etc. For each of these tasks, we adopt a supervised machine learning perspective using Support Vector Machines (SVMs) trained on the Arabic TreeBank, leveraging off of already existing algorithms for English. The results are comparable to state-of-the-art results on English text when trained on similar sized data.

2 Arabic Language and Data

Arabic is a Semitic language with rich templatic morphology. An Arabic word may be composed of a stem (consisting of a consonantal root and a template), plus affixes and clitics. The affixes include inflectional markers for tense, gender, and/or number. The clitics include some (but not all) prepositions, conjunctions, determiners, possessive pronouns and pronouns. Some are proclitic (attaching to the beginning of a stem) and some enclitics (attaching to the end of a stem). The following is an example of the different morphological segments in the word وبحسناتهم which means *and by their virtues*. Arabic is read from right to left hence the directional switch in the English gloss.

	enclitic	**affix**	**stem**	**proclitic**	**proclitic**
Arabic:	هم	ات	حسن	ب	و
Translit:	hm	At	Hsn	b	w
Gloss:	their	s	virtue	by	and

The set of possible proclitics comprises the prepositions {*b,l,k*}, meaning *by/with, to, as*, respectively, the conjunctions {*w, f*}, meaning *and, then*, respectively, and the definite article or determiner {*Al*}, meaning *the*. Arabic words may have a conjunction and a prepostition and a determiner cliticizing to the beginning of a word. The set of possible enclitics comprises the pronouns and (possessive pronouns) {*y, nA, k, kmA, km, knA, kn, h, hA, hmA, hnA, hm, hn*}, respectively, *my (mine), our (ours),*

*This work was partially supported by the National Science Foundation via a KDD Supplement to NSF CISE/IRI/Interactive Systems Award IIS-9978025.

[1] http://www.ldc.upenn.edu/

your (yours), your (yours) [masc. dual], your (yours) [masc. pl.], your (yours) [fem. dual], your (yours) [fem. pl.], him (his), her (hers), their (theirs) [masc. dual], their (theirs) [fem. dual], their (theirs) [masc. pl], their (theirs) [fem. pl.]. An Arabic word may only have a single enclitic at the end. In this paper, stems+affixes, proclitics, enclitics and punctuation are referred to as tokens. We define a token as a space delimited unit in clitic tokenized text.

We adopt a supervised learning approach, hence the need for annotated training data. Such data are available from the `Arabic TreeBank`,[2] a modern standard Arabic corpus containing `Agence France Presse` (AFP) newswire articles ranging over a period of 5 months from July through November of 2000. The corpus comprises 734 news articles (140k words corresponding to 168k tokens after semi-automatic segmentation) covering various topics such as sports, politics, news, etc.

3 Related Work

To our knowledge, there are no systems that automatically tokenize and POS Arabic text as such. The current standard approach to Arabic tokenization and POS tagging — adopted in the `Arabic TreeBank` — relies on manually choosing the appropriate analysis from among the multiple analyses rendered by `AraMorph`, a sophisticated rule based morphological analyzer by Buckwalter.[3] Morphological analysis may be characterized as the process of segmenting a surface word form into its component derivational and inflectional morphemes. In a language such as Arabic, which exhibits both inflectional and derivational morphology, the morphological tags tend to be fine grained amounting to a large number of tags — `AraMorph` has 135 distinct morphological labels — in contrast to POS tags which are typically coarser grained. Using `AraMorph`, the choice of an appropriate morphological analysis entails clitic tokenization as well assignment of a POS tag. Such morphological labels are potentially useful for NLP applications, yet the necessary manual choice renders it an expensive process.

On the other hand, Khoja (Khoja, 2001) reports preliminary results on a hybrid, statistical and rule based, POS tagger, `APT`. `APT` yields 90% accuracy on a tag set of 131 tags including both POS and inflection morphology information. `APT` is a two-step hybrid system with rules and a Viterbi algorithm for statistically determining the appropriate POS tag. Given the tag set, `APT` is more of a morphological analyzer than a POS tagger.

[2]http://www.ldc.upenn.edu

[3]http://www.ldc.upenn.edu/myl/morph/buckwalter.html

4 SVM Based Approach

In the literature, various machine learning approaches are applied to the problem of POS tagging and BP Chunking. Such problems are cast as a classification problem where, given a number of features extracted from a predefined linguistic context, the task is to predict the class of a token. `Support Vector Machines` (SVMs) (Vapnik, 1995) are one class of such model. SVMs are a supervised learning algorithm that has the advantage of being robust where it can handle a large number of (overlapping) features with good generalization performance. Consequently, SVMs have been applied in many NLP tasks with great success (Joachims, 1998; Kudo and Matsumato, 2000; Hacioglu and Ward, 2003).

We adopt a tagging perspective for the three tasks. Thereby, we address them using the same SVM experimental setup which comprises a standard SVM as a multiclass classifier (Allwein et al., 2000). The difference for the three tasks lies in the input, context and features. None of the features utilized in our approach is explicitly language dependent. The following subsections illustrate the different tasks and their corresponding features and tag sets.

4.1 Word Tokenization

We approach word tokenization (segmenting off clitics) as a one-of-six classification task, in which each letter in a word is tagged with a label indicating its morphological identity.[4] Therefore, a word may have $0 \leq 2$ proclitics and $0 \leq 1$ enclitic from the lists described in Section 2. A word may have no clitics at all, hence the 0.

Input: A sequence of transliterated Arabic characters processed from left-to-right with "break" markers for word boundaries.

Context: A fixed-size window of -5/+5 characters centered at the character in focus.

Features: All characters and previous tag decisions within the context.

Tag Set: The tag set is {*B-PRE1, B-PRE2, B-WRD, I-WRD, B-SUFF, I-SUFF*} where *I* denotes inside a segment, *B* denotes beginning of a segment, *PRE1* and *PRE2* are proclitic tags, *SUFF* is an enclitic, and *WRD* is the stem plus any affixes and/or the determiner *Al*.

Table 1 illustrates the correct tagging of the example above, *w-b-hsnAt-hm*, 'and by their virtues'.

4.2 Part of Speech Tagging

We model this task as a 1-of-24 classification task, where the class labels are POS tags from the collapsed tag set in

[4]For the purposes of this study, we do not tokenize the proclitic determiner *Al* since it is not tokenized separately in the `Arabic treebank`.

Arabic	Translit.	Tag
و	w	B-PRE1
ب	b	B-PRE2
بح	H	B-WRD
ســ	s	I-WRD
ـنـ	n	I-WRD
ـا	A	I-WRD
ـتـ	t	I-WRD
ـه	h	B-SUFF
ـم	m	I-SUFF

Table 1: Sample SVM−TOK tagging

the Arabic TreeBank distribution. The training data is derived from the collapsed POS-tagged Treebank.
Input: A sequence of tokens processed from left-to-right.
Context: A window of -2/+2 tokens centered at the focus token.
Features: Every character N-gram, $N \leq 4$ that occurs in the focus token, the 5 tokens themselves, their 'type' from the set {*alpha, numeric*}, and POS tag decisions for previous tokens within context.
Tag Set: The utilized tag set comprises the 24 collapsed tags available in the Arabic TreeBank distribution. This collapsed tag set is a manually reduced form of the 135 morpho-syntactic tags created by AraMorph. The tag set is as follows: {*CC, CD, CONJ+NEG_PART, DT, FW, IN, JJ, NN, NNP, NNPS, NNS, NO_FUNC, NUMERIC_COMMA, PRP, PRP$, PUNC, RB, UH, VBD, VBN, VBP, WP, WRB*}.

4.3 Base Phrase Chunking

In this task, we use a setup similar to that of (Kudo and Matsumato, 2000), where 9 types of chunked phrases are recognized using a phrase IOB tagging scheme; Inside *I* a phrase, Outside *O* a phrase, and Beginning *B* of a phrase. Thus the task is a one of 19 classification task (since there are I and B tags for each chunk phrase type, and a single O tag). The training data is derived from the Arabic TreeBank using the ChunkLink software.[5]. ChunkLink flattens the tree to a sequence of base (non-recursive) phrase chunks with their IOB labels. The following example illustrates the tagging scheme:

Tags:	O	B-VP	B-NP	I-NP
Translit:	w	qAlt	rwv	$wArtz
Arabic:	و	قالت	روث	شوارتز
Gloss:	and	said	Ruth	Schwartz

Input: A sequence of (word, POS tag) pairs.
Context: A window of -2/+2 tokens centered at the focus token.
Features: Word and POS tags that fall in the context along with previous IOB tags within the context.

[5]http://ilk.uvt.nl/ sabine/chunklink

Tag Set: The tag set comprises 19 tags: {*O, I-ADJP, B-ADJP, I-ADVP, B-ADVP, I-CONJP, B-CONJP, I-NP, B-NP, I-PP, B-PP, I-PRT, B-PRT, I-SBAR, B-SBAR, I-UCP, B-UCP, I-VP, B-VP*}

5 Evaluation

5.1 Data, Setup and Evaluation Metrics

The Arabic TreeBank consists of 4519 sentences. The development set, training set and test set are the same for all the experiments. The sentences are randomly distributed with 119 sentences in the development set, 400 sentences in the test set and 4000 sentences in the training set. The data is transliterated in the Arabic TreeBank into Latin based ASCII characters using the Buckwalter transliteration scheme.[6] We used the non vocalized version of the treebank for all the experiments. All the data is derived from the parsed trees in the treebank. We use a standard SVM with a polynomial kernel, of degree 2 and C=1.[7] Standard metrics of Accuracy (Acc), Precision (Prec), Recall (Rec), and the F-measure, $F_{\beta=1}$, on the test set are utilized.[8]

5.2 Tokenization

Results: Table 2 presents the results obtained using the current SVM based approach, SVM−TOK, compared against two rule-based baseline approaches, RULE and RULE+DICT. RULE marks a prefix if a word starts with one of five proclitic letters described in Section 4.1. A suffix is marked if a word ends with any of the possessive pronouns, enclitics, mentioned above in Section 4.1. A small set of 17 function words that start with the proclitic letters is explicitly excluded.

RULE+DICT only applies the tokenization rules in RULE if the token does not occur in a dictionary. The dictionary used comprises the 47,261 unique non vocalized word entries in the first column of Buckwalter's dictStem, freely available with the AraMorph distribution. In some cases, dictionary entries retain inflectional morphology and clitics.

System	Acc.%	Prec.%	Rec.%	$F_{\beta=1}$
SVM−TOK	**99.77**	**99.09**	**99.15**	**99.12**
RULE	96.83	86.28	91.09	88.62
RULE+DICT	98.29	93.72	93.71	93.71

Table 2: Results of SVM−TOK compared against RULE and RULE+DICT on Arabic tokenization

Discussion: Performance of SVM−TOK is essentially perfect; $F_{\beta=1} = 99.12$. The task, however, is quite easy,

[6]http://www.ldc.upenn.edu/myl/morph/buckwalter.html
[7]http://cl.aist-nara.ac.jp/ taku-ku/software/yamcha
[8]We use the CoNLL shared task evaluation tools available at http://cnts.uia.ac.be/conll2003/ner/bin/conlleval.

and SVM−TOK is only about 5% better (absolute) than the baseline RULE+DICT. While RULE+DICT could certainly be improved with larger dictionaries, however, the largest dictionary will still have coverage problems, therefore, there is a role for a data-driven approach such as SVM−TOK. An analysis of the confusion matrix for SVM−TOK shows that the most confusion occurs with the *PREF2* class. This is hardly surprising since *PREF2* is an infix category, and thus has two ambiguous boundaries.

5.3 Part of Speech Tagging

Results: Table 3 shows the results obtained with the SVM based POS tagger, SVM−POS, and the results obtained with a simple baseline, BASELINE, where the most frequent POS tag associated with a token from the training set is assigned to it in the test set. If the token does not occur in the training data, the token is assigned the *NN* tag as a default tag.

System	Acc.%
SVM−POS	95.49
BASELINE	92.2

Table 3: Results of SVM−POS compared against BASELINE on the task of POS tagging of Arabic text

Discussion: The performance of SVM−POS is better than the baseline BASELINE. 50% of the errors encountered result from confusing nouns, NN, with adjectives, JJ, or vice versa. This is to be expected since these two categories are confusable in Arabic leading to inconsistencies in the training data. For example, the word for *United* in *United States of America* or *United Nations* is randomly tagged as a noun, or an adjective in the training data. We applied a similar SVM based POS tagging system to English text using the English TreeBank. The size of the training and test data corresponded to those evaluated in the Arabic experiments. The English experiment resulted in an accuracy of 94.97%, which is comparable to the Arabic SVM−POS results of 95.49%.

5.4 Base Phrase Chunking

Results: Table 4 illustrates the results obtained by SVM−BP

BPC	Acc.%	Prec.%	Rec.%	$F_{\beta=1}$
SVM−BP	94.63	92.06	92.09	92.08

Table 4: Results of SVM−BP on base phrase chunking of Arabic text

Discussion: The overall performance of SVM−BP is $F_{\beta=1}$ score of 92.08. These results are interesting in light of state-of-the-art for English BP chunking performance which is at an $F_{\beta=1}$ score of 93.48, against a baseline of 77.7 in CoNLL 2000 shared task (Tjong et al., 2000). It is worth noting that SVM−BP trained on the English TreeBank, with a comparable training and test size data to those of the Arabic experiment, yields an $F_{\beta=1}$ score of 93.05. The best results obtained are for VP and PP, yielding $F_{\beta=1}$ scores of 97.6 and 98.4, respectively.

6 Conclusions & Future Directions

We have presented a machine-learning approach using SVMs to solve the problem of automatically annotating Arabic text with tags at different levels; namely, tokenization at morphological level, POS tagging at lexical level, and BP chunking at syntactic level. The technique is language independent and highly accurate with an $F_{\beta=1}$ score of 99.12 on the tokenization task, 95.49% accuracy on the POS tagging task and $F_{\beta=1}$ score of 92.08 on the BP Chunking task. To the best of our knowledge, these are the first results reported for these tasks in Arabic natural language processing.

We are currently trying to improve the performance of the systems by using additional features, a wider context and more data created semi-automatically using an unannotated large Arabic corpus. In addition, we are trying to extend the approach to semantic chunking by hand-labeling a part of Arabic TreeBank with arguments or semantic roles for training.

References

Erin L. Allwein, Robert E. Schapire, and Yoram Singer. 2000. *Reducing multiclass to binary: A unifying approach for margin classifiers*. Journal of Machine Learning Research, 1:113-141.

Kadri Hacioglu and Wayne Ward. 2003. *Target word Detection and semantic role chunking using support vector machines*. HLT-NAACL 2003.

Thorsten Joachims. 1998. *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. Proc. of ECML-98, 10th European Conf. on Machine Learning.

Shereen Khoja. 2001. *APT: Arabic Part-of-speech Tagger*. Proc. of the Student Workshop at NAACL 2001.

Taku Kudo and Yuji Matsumato. 2000. *Use of support vector learning for chunk identification*. Proc. of the 4th Conf. on Very Large Corpora, pages 142-144.

Erik Tjong, Kim Sang, and Sabine Buchholz. 2000. *Introduction to the CoNLL-2000 shared task: Chunking*. Proc. of the 4th Conf. on Computational Natural Language Learning (CoNLL), Lisbon, Portugal, 2000, pp. 127-132.

Vladamir Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, USA.