ASMA: A System for Automatic Segmentation and Morpho-Syntactic Disambiguation of Modern Standard Arabic

Muhammad Abdul-Mageed
Indiana University
Bloomington, IN, USA
mabdulma@indiana.edu

Mona Diab
George Washington University
Washington DC, USA
mtdiab@email.gwu.edu

Sandra Kübler
Indiana University
Bloomington, IN, USA
skuebler@indiana.edu

Abstract

In this paper, we present ASMA, a fast and efficient system for automatic segmentation and fine grained part of speech (POS) tagging of Modern Standard Arabic (MSA). ASMA performs segmentation both of agglutinative and of inflectional morphological boundaries within a word. In this work, we compare ASMA to two state of the art suites of MSA tools: AMIRA 2.1 (Diab et al., 2007; Diab, 2009) and MADA+TOKAN 3.2. (Habash et al., 2009). ASMA achieves comparable results to these two systems' state-of-the-art performance. ASMA yields an accuracy of 98.34% for segmentation, and an accuracy of 96.26% for POS tagging with a rich tagset and 97.59% accuracy with an extremely reduced tagset.

1 Introduction

Arabic raises various challenges to natural language processing (NLP): Arabic is a morphologically rich language (Tsarfaty et al., 2010), where significant information concerning syntactic units is expressed at the word level, which makes part of speech (POS) tagging a challenge since it involves morpho-syntactic disambiguation, including features like voice, number, gender (Diab, 2007; Diab et al., 2007; Habash et al., 2009).

We address the problem of full morpho-syntactic disambiguation of words in context. We devise a system, ASMA, that performs both inflectional morpheme segmentation and agglutinative clitic segmentation. For example, given a surface word in context such as وَبِحَسَنَاتِهِم (*wabiHasanaAtihim*, Eng. 'and by their virtues')[1], ASMA provides the following segmentation: وَ بِ حَسَن اَتِ هِم *wa bi Hasan aAti him*, with the prefixal clitics وَ بِ (*wa bi*, Eng. 'and' 'by'), the stem حَسَن (*Hasan*), the inflection morpheme اَتِ (*aAti*), and the suffixal pronominal morpheme هِم (*him*). ASMA then assigns each one of these resulting morphemes a POS tag. For an explanation of Arabic morphology, cf. section 2.

The most successful approaches to date that render this level of morphological segmentation (addressing both inflectional as well as agglutinative boundaries) typically rely on employing a morphological analyzer in the process (Habash et al., 2009). We show that it is possible to efficiently perform full morpho-syntactic disambiguation employing language-independent methods that are not based on a morphological analyzer. Our motivation is that dependence on a morphological analyzer comes at the cost of development since such an analyzer is generally based on manually written rules and an extensive lexicon.

ASMA performs both inflectional morpheme segmentation and agglutinative clitic segmentation, as well as fine grained POS tagging of Modern Standard Arabic (MSA). In ASMA, a *segment* is a stem, an inflectional affix, or a clitic. ASMA does not handle morphotactic boundaries, thereby potentially deriving stems which may not be smoothed into correct lexemic forms for the POS process. An example of the result of the *segmentation* in ASMA is as follows: the surface form الوِلَايَات (*AlwilaAyaAt*, Eng. 'the states') is segmented into ال + وِلَاي + اَت (*Al+wilaAy+aAt*) where *wilaAy* is a stem, *Al* is a clitic, and *At* is an affixival inflectional suffix. It should be noted that *wilaAy* is not a valid Arabic lexeme. For ASMA to convert it into a lexeme, it would have to process the morphotactics on the stem and render it as وِلَايَة

[1] For Arabic examples, we use both the Arabic script and the Buckwalter Arabic transliteration scheme (Buckwalter, 2004).

(*wilaAyap*) restoring the lexeme/lemma final ة *p*.

The remainder of the paper is structured as follows: Section 2 describes the pertinent facts about Arabic morphology. Section 3 describes related work, namely on AMIRA 2.1 and MADA+TOKAN 3.2. In section 4, we describe ASMA, the overall system, in section 5, we report results on the segmentation task, and in section 6 on the POS tagging task. In section 7 we provide an error analysis, and conclude in section 9.

2 Arabic Morphology

Arabic exhibits derivational, inflectional, and agglutinative morphology. Derivational morphology is mostly templatic where a word is made up of a root and a pattern, along with some idiosyncratic information. For example, a root such as ك ت ب (*k t b*) if combined with the pattern *1a2a3*, where the numbers [1,2,3] designate the root radicals, respectively, it results in the derivational form كَتَب (*katab*, Eng. 'to write'). Likewise for the same root when it combines with the pattern *1A2a3*, it result in the word كَاتَب (*kaAtab*, Eng. 'to correspond'). All derivation forms undergo inflection reflecting various types of functional features such as voice, number, aspect, gender, grammatical case, tense, etc. The resulting word is known as a *lexeme*. Therefore a lexeme such as كَتَبَت (*katabat*, Eng. 'she wrote') reflects feminine [gender], singular [number], past [tense], perfective [aspect], 3rd [person] inflections for the verb. Typically, one of the fully inflected lexemes is considered a citation form, and it is known as the *lemma*. The choice of a specific lexeme as a citation form is a convention, and it is typically the 3rd person masculine singular perfective form for verbs and the 3rd person singular form for nouns. Hence in this case the lemma is كَتَبَ (*kataba*, Eng. 'he wrote'). Arabic words often undergo clitic agglutination to form surface words. For example, the lexeme كَاتَبَت (*kAtabat*, Eng. 'she corresponded') could have an enclitic/suffixal pronoun as follows: كَاتَبَتهُم (*kAtabathum*, Eng. 'she corresponded with them'). The agglutination process results in morphotactic variations at the morpheme boundaries where the orthography is changed for the underlying lexeme. For example, in a noun such as وَبِحَسَنَتِهِم (*wabiHasanathim*, Eng. 'and by their virtue'), the underlying lexeme (same as lemma in this case) is the noun حَسَنَة (*Hasanap*), where the lexeme final Taa-Marbuta ة (*p*) is changed into a regular ت (*t*) when followed by a pronominal clitic. Accordingly, segmenting off agglutinative clitics without handling boundary morphotactics to restore the underlying lexeme form results in *stems*.

3 Related Work

AMIRA 2.1 (Diab et al., 2007; Diab, 2009) is a supervised SVM-based machine learning algorithm for processing MSA, including clitic tokenization and normalization, POS tagging, and base phrase chunking. Diab. et al. adopt the inside-outside-beginning (IOB) chunk approach (Ramshaw and Marcus, 1995) for clitic tokenization, i.e., each letter in a word is labeled as being at the beginning (B), the inside (I), or the outside (O) of a chunk. Note that the tokenization by Diab et al. does not split off inflectional morphology. For example, while ASMA would segment وبحسنَاتهم (*wbHsnAthm*) into *w+b+Hsn+At+hm*, AMIRA 2.1 would output *w+b+HsnAt+hm*, i.e., it does not split off the number and gender inflectional suffix ات *At* from the stem حسن (*Hsn*).

One advantage of ASMA over AMIRA 2.1 is thus that ASMA identifies inflectional morpheme boundaries. Similar to AMIRA 2.1, ASMA employs an IOB chunking approach on the character level for segmentation of words into morphemic chunks (clitics, stems, and inflectional affixes). AMIRA 2.1 achieves an F-measure of 99.15% for the entire word being segmented correctly. AMIRA 2.1 also performs POS tagging. It uses multiple POS tagsets ranging from a basic 24 tagset called Reduced TagSet (RTS) to an enriched tagset (ERTS) of 75 tags. AMIRA 2.1. achieves an accuracy of 96.6% for RTS and 96.13% for ERTS. ASMA, in contrast, uses a fuller tagset of 139 POS tags, which includes morphological information, e.g., on gender and number.

MADA+TOKAN 3.2 Habash et al. Habash and Rambow (2005; Habash et al. (2009) developed MADA, a system for the morphological disambiguation of MSA. MADA relies on the output of the Buckwalter Arabic Morphological Analyzer (BAMA) (Buckwalter, 2004) and uses 14 individual SVM classifiers for learning individual fea-

tures, which makes it computationally costly compared to ASMA which uses a single classifier for each of the two tasks of segmentation and morphological disambiguation. TOKAN, a separate tool, performs tokenization on the output of MADA. For tokenization, Habash et al. report 98.85% word level accuracy and for POS tagging, 96.1% accuracy. MADA+TOKAN 3.2 perform segmentation similar to ASMA. However, MADA+TOKAN 3.2 depend on the underlying morphological analyzer. In contrast to ASMA, MADA+TOKAN 3.2 perform POS tagging yielding the fully specified morphological analysis in the ATB, which comprises 440 unique tags.

4 ASMA

4.1 Method: Memory-Based Learning

For both segmentation and POS tagging, we use *memory-based learning* (MBL) (Aha et al., 1991) classifiers. MBL is a lazy learning method that does not abstract rules from the data, but rather keeps all training data. During training, the learner stores the training instances without abstraction. Given a new instance, the classifier finds the k nearest neighbors in the training set and chooses their most frequent class for the new instance. MBL has been shown to have a suitable bias for NLP problems (Daelemans et al., 1999; Daelemans and van den Bosch, 2005) since it does not abstract over irregularities or subregularities. For each of the two classification tasks (i.e., segmentation and POS tagging), we use MBT (Daelemans et al., 1996), a memory-based POS tagger that has access to previous tagging decisions in addition to an expressive feature set.

4.2 Data Sets and Splits

We use segmentation and POS data from the Penn Arabic Treebank (PATB) (Maamouri et al., 2004), specifically, we use the following parts: ATB1V4, ATB2V3, ATB3V3.1 and ATB3V3.2 with different splits as described below. The textual basis of the treebank consists of newswire articles covering political, economic, cultural, sports, etc. topics. Table 1 presents for each part the number of words, the number of tokens (i.e., only clitics are split off), the number of segments (i.e., clitic and inflectional morphology is split off), the number of news reports, and the source of the reports (i.e., the news agency)[2]. As mentioned above, Arabic is generally written without diacritics. While the ATB does have a version with diacritics restored, for our experiments, we use the version without diacritics, for both segmentation and POS tagging.

For a fair comparison of ASMA to both AMIRA and MADA, we adopt two different data splits, AMIRA-SPLIT and MADA-SPLIT, with each split corresponding to the data splits used in the evaluations of these systems. The same splits are used both for segmentation and POS tagging. For the AMIRA-SPLIT, we follow the procedure by Diab et al. (2004), but we use more recent releases of the ATB than Diab et al. We split each of the first three parts into 10% development data (DEV), 80% training data (TRAIN), and 10% test data (TEST). We then concatenate the respective splits from each part. For example, to create a single DEV set from the three parts, we concatenate the 10% DEV data from ATB1V4, ATB2V3, and ATB3V3.2, etc. For MADA-SPLIT, we follow the MADA manual[3]. For this split, ATB1V4 and ATB2V3 and the first 80% of ATB3V3.1 are used as the TRAIN set, the last 20% of ATB3V3.1 are divided into two halves, i.e. DEV and TEST (each making up 10% of ATB3V3.1) respectively. The development sets are used for parameter and feature optimization.

5 Segmentation

5.1 Setup

We define segmentation as an IOB classification task, where each letter in a word is tagged with a label indicating its place in a segment. The tagset is $\{$*B-SEG, I-SEG, O*$\}$, where B is a tag assigned to the beginning of a segment, I denotes the inside of a segment, and O spaces between surface form words.

Procedure: We performed a non-exhaustive search for optimal settings for the following MBT parameters: the MBL *algorithm*, the *similarity* metric, the *feature weighting* method, and the *value* of the k nearest neighbors. The best setting used the IB1 algorithm with *weighted overlap* as the similarity metric, gain ratio (GR) as a feature weighting method, and a value of $k = 1$.

[2]The information is based on the LDC documentation at `http://www.ldc.upenn.edu/Catalog/docs/*`.

[3]`http://www1.ccls.columbia.edu/MADA`

Data set	# words	# tokens	# segments	# texts	Source
ATB1V4	145,386	167,280	209,187	734	AFP
ATB2V3	144,199	169,319	221,001	501	UMMAH
ATB3V3.1	340,281	402,291	551,171	600	An Nahar
ATB3V3.2	339,710	402,291	512,932	599	An Nahar

Table 1: Data statistics and sources

Our complete feature set comprises the six preceding characters, the previous tag decisions of all the six preceding characters except the character immediately preceding the focus character, the focus character itself and its ambiguity tag (*henceforth*, *ambitag*), and the seven following characters. For features, we tested (1) left only, right only, and left and right contexts across various *window sizes* and (2) different *types of information*, e.g., feature sets with/without previous tag decisions for left context, feature sets with/without ambitags of right context. An ambitag is a combination of all tags of the ambiguity set of a word.

Evaluation: We evaluate segmentation in terms of character-based accuracy, word level accuracy, and precision, recall, and F-measure for segments. For example, the word الولَايَات (*AlwlAyAt*, Eng. 'the states') has the correct segmentation ال+ ولَاي + ات *Al+wlAy+At* and comprises 8 characters. If it is segmented incorrectly as ات + الولَاي (*Al-wlAy+At*), one of the 8 characters, the 'w' is incorrectly classified as I as opposed to B, and consequently, we have a character based accuracy of 7/8, a word level based accuracy of 0/8. On the segment level, precision is 50%, recall 33.33%, and the F-measure 41.65.

5.2 Segmentation Results

Table 2 shows the results for segmentation on the two data splits, AMIRA-SPLIT and MADA-SPLIT. For both data splits, the best features are the six preceding characters, the previous tag decisions of all the six preceding characters except the character immediately preceding the focus character, the focus character itself and its ambitag, and the seven following characters.

AMIRA-SPLIT: On the TEST data for this split, we reach an accuracy of 99.53%, a precision of 97.97%, a recall of 98.04% and an *F* of 98.01%. The segmentation accuracy is at 98.34% for words.

MADA-SPLIT: For this data set, we achieve an accuracy of 99.49%, precision of 97.72%, a recall of 97.85% and an *F* of 97.79%. Segmentation accuracy for words is at 98.10%

These experiments show that on the segmentation level, the MADA split is slightly more complex than the AMIRA split.

Our segmentation results are not fully comparable to the tokenization performance of AMIRA (Diab et al., 2004) since AMIRA does not split off inflectional morphology. MADA (Habash and Rambow, 2005; Habash et al., 2009), in contrast, does perform segmentation, but it is based on a morphological analyzer. ASMA, without the use of any external resources, achieved a word accuracy of 98.10% on the MADA-SPLIT, which is only slightly lower than MADA's 98.85% word accuracy.

6 POS Tagging

POS tagging is defined here so that each segment, rather than a full word (as in (Kübler and Mohamed, 2012)) or a token (as in (Diab et al., 2004)), is assigned a POS tag. For the experiments reported here, we modify the ATB tagset such that case and mood tags are removed since those are syntactic features that cannot be determined based on a local context. While AMIRA, similar to ASMA, does not predict case and mood, MADA does at the cost of some performance loss. The remaining tagset comprises 139 segment-based tags. The input for the POS tagger consists of gold segmented data. The reasons for this decision are mainly to allow us to compare our system to AMIRA, which also uses gold segmentation.

6.1 Setup

Procedure: We performed a non-exhaustive search for the best parameters described in section 5. We use the IGTREE algorithm. We identified the *modified value difference metric* (MVDM) as similarity metric, gain ratio (GR) as a feature weighting method, and $k = 1$ for known words

System	Split	Acc.	Precision	Recall	*F*	Word Acc.
ASMA	AMIRA-SPLIT	99.53	97.97	98.04	98.01	98.34
	MADA-SPLIT	99.49	97.72	97.85	97.79	98.10
MADA 3.2						98.85

Table 2: Segmentation results

and $k = 30$ for unknown words as optimal parameters. For both data splits, the following feature sets give optimal results on the DEV set: For known segments, the best feature set uses the focus segment, its ambitag, two previous segments, and the predicted tag of three previous segments. For unknown segments, the feature set consists of the five previous segments and their predicted tags, the focus segment itself and its ambitag, the first five characters and the last three characters of the focus segment, and six following segments and their ambitags.

Evaluation: We evaluate based on segments, i.e. on the units which were used for POS tagging, rather than on full words. We report overall accuracy as well as accuracy on known segments and on unknown segments.

6.2 POS Tagging Results

Table 3 shows the results for POS tagging on the two data sets given the settings and the feature set described above.

AMIRA-SPLIT: Using the feature set described above, we reach an accuracy of 96.61% on known words and 74.46% on unknown words, averaging 96.26% on all words.

MADA-SPLIT: We reach an accuracy of 94.61% on known words and of 86.00% on unknown words, averaging 94.67% on all words. In comparison, the results for unknown words are much higher. This is due to the fact that in the MADA split, we only have 593 unknown words while the AMIRA split has more than twice as many (i.e. 1261 unknown words).

These experiments show that for POS tagging, the MADA split is considerably more challenging than the AMIRA split. This means that even if results reported for MSA are based on the same sub-word analysis, the data splits have to be taken into account in a comparison as well.

Our POS tagging results are not directly comparable to AMIRA, because of the differences in segmentation and because of the different POS tagsets. They are comparable to those obtained with MADA using tokenization by TOKAN. Roth et al. (2008) report 94.7% accuracy on predicting 10 morphological types of features, the closest setting to our tagset. This is very close to the 94.67% we report using the MADA-SPLIT. Roth et al. report a slight improvement for an extended system using diacritic markers as additional input, but as Kübler and Mohamed (2012) have shown, automatic diacritization must be extremely accurate in order to be useful for POS tagging.

6.3 Experimenting with Other Tagsets

We also ran experiments with two other tagsets, the standard RTS tagset, which is composed of 25 tags, and the CATiB tagset (Habash and Roth, 2009), which comprises only 6 tags, in order to investigate the effect of using different levels of morphological and morpho-syntactic information in the tagset. The full tagset, as mentioned above, includes all morphological information, except for case and mood markers. The RTS tagset is a reduced version, resulting in a tagset that is similar to the English Penn Treebank tagset (Santorini, 1990). Using the RTS tagset also allows us to make our results more comparable to AMIRA. The CATiB tagset represents only the major word classes, such as noun or verb. We used CATiB because its tagset corresponds to traditional notions in Arabic grammar and because it was used in the Columbia Arabic Treebank (Habash and Roth, 2009).

For this set of experiments, we use the same parameters and feature settings as described in section 6.2 above. Thus, the results reported on this set of experiments are potentially suboptimal. In the future, we plan to tune the performance of ASMA with each of these tagsets. Table 4 shows the results of these experiments.

6.3.1 RTS

AMIRA-SPLIT: Using RTS, we reach an accuracy of 96.28%. This is very slightly higher than our results for the full POS tagset (96.26%), and it is very close to AMIRA's results when using

System	Split	Acc: known	Acc: unknown	Acc: all
ASMA	AMIRA-SPLIT	96.61	74.46	96.26
	MADA-SPLIT	94.80	86.00	94.67
MADA 3.2				94.70
AMIRA 2.1 - ERTS				96.13

Table 3: POS tagging results

Tagset	System	Split	Acc: known	Acc: unknown	Acc: all
RTS	ASMA	AMIRA-SPLIT	96.56	77.79	96.28
	ASMA	MADA-SPLIT	94.20	84.99	94.06
	AMIRA 2.1				96.60
CATiB	ASMA	AMIRA-SPLIT	97.88	79.27	97.59
	ASMA	MADA-SPLIT	96.04	88.36	95.92

Table 4: POS tagging results with the RTS and CATiB tagsets

the RTS. But note that AMIRA uses tokenization rather than segmentation; thus the results are not directly comparable. We also notice that ASMA's performance on unknown words improves by almost 3 percent points to 77.79%, as opposed to 74.46% using the full tagset. This is to be expected since guessing the morphological information for an unknown word is more difficult than guessing only the main category in RTS.

MADA-SPLIT: Here, ASMA reaches an overall accuracy of 94.06%. This is slightly lower than for the full tagset (94.67%), due to a drop in accuracy on unknown words, from 86.00% to 84.99% and a slight drop in accuracy on known words from 94.80% to 94.20%.

The results for the RTS on both data splits show that ASMA reaches state-of-the-art results, without using morphological analysis and while using a classifier not optimized for sequence handling, but which has access to previous classification decisions. The results also show that, in general, using the reduced tagset does not significantly change the difficulty of the task. In other words, giving up morphological information in the tagset in this specific case does not lead to higher tagging accuracy.

6.3.2 CATiB

AMIRA-SPLIT: With the CATiB tagset, ASMA reaches an overall accuracy of 97.59%, showing that an extreme reduction of the tagset to one completely devoid of morphological information increases tagging accuracy.

MADA-SPLIT: With the CATiB tagset used with this split, ASMA reaches an overall accuracy of 95.92%.

Both sets of experiments show that the amount of morphological and morpho-syntactic information present in the POS tagset has an influence on the difficulty of the POS tagging step, even though the connection is not always a direct one. Thus, if ASMA is used as a preprocessing system for upstream modules, it is necessary to choose the tagset with regard to the upstream task.

Tag	Conf. %	% of Error
NOUN	3.6	1.05
NOUN_PROP	8.16	0.62
ADJ	7.64	0.59
PV	7.76	0.30
PV_PASS	45.54	0.13
IV_PASS	45.23	0.12
ADJ.VN	43.23	0.11
IV	3.38	0.10
PVSUFF_SUBJ:3FS	7.36	0.10
NOUN.VN	31.14	0.09

Table 5: Example results per POS category and their respective confusable modified ATB POS tag

7 Error Analysis

We performed an error analysis to see which types of errors ASMA makes. Table 5 presents a confusion matrix for the ATB tagset we used in section 6.2. We provide results only with the AMIRA split, as the results for the MADA split are similar. The table is sorted based on the contribution

the confusion pair makes towards the overall error rate.

The table shows that because of the high number of POS labels, each confusion case contributes only marginally to the overall error rate. The most likely errors involve nouns (NOUN), proper nouns (NOUN_PROP), and adjectives (ADJ). These errors can be explained via the characteristics of Arabic: Proper nouns in Arabic are generally standard nouns used as names. Thus, the same word can be used as either noun or proper noun, depending on the context. Additionally, unlike English, Arabic proper nouns are not marked by capitalization or other orthographic means. The noun-adjective distinction is not clear in Arabic: Adjectives can be used as nouns, and they share the same morphological patterns as nouns.

The next set concerns the POS tags PV_PASS, IV_PASS, and ADJ.VN. With the lack of diacritics, the classifier is prone to erring with regard to cases where diacritics play a crucial factor in carrying the grammatical function. Since passivization is marked using diacritics in Arabic, passive verbs also suffer from the lack of diacritics, both in the perfective (i.e., PV_PASS) and imperfective (i.e., IV_PASS) cases, and hence the misclassification and high percent of confusion between passive and active verbs in the data. Adjectival verbal nouns – i.e., ADJ.VN as in مُعلِن (*mu'lin*, Eng. 'announcing') – are also confused with adjectives as these two parts of speech have very similar contexts, especially given the lack of diacritic *nunation*[4] characteristic of the adjectival verbal noun.

8 ASMA in Comparison

As described above, ASMA performs both inflectional morpheme segmentation and agglutinative clitic segmentation, as well as fine grained POS tagging of Modern Standard Arabic (MSA). Compared to AMIRA, ASMA performs more fine grained morphological disambiguation due to ASMA's identification of inflectional morpheme boundaries. Compared to MADA, ASMA performs the same tasks, however without using a morphological analyzer. Given that restriction, it still achieves state-of-the-art results, only minimally lower than MADA's. One major advantage of ASMA is the high speed with which it operates: On a PowerPC 970 machine, with a Darwin Kernel Version 8.11.0 and 2GB memory, it takes ASMA about 5 minutes to process 100 000 words. Although we have not had the chance to compare ASMA and MADA in terms of the speed with which each operates, we believe that ASMA is significantly faster than MADA. After all, whereas MADA employs 14 individual SVM classifiers to learn individual features, ASMA employs a single classifier per task, segmentation and morpho-syntactic disambiguation. AMIRA is observably slower than ASMA. In addition, while the MBL framework in ASMA uses virtually no time to train, SVMs (which AMIRA and MADA use) are known for long training times. Its speed makes ASMA valuable especially for real-world tasks, such as information retrieval and extraction, and tasks depending on big data processing.

ASMA is flexible in terms of the granularity of its output as it renders morphological disambiguation with three different tagsets (i.e., the full ATB 139 tagset, the RTS, and the reduced CATiB tagset). As such, ASMA can be customized to different NLP tasks depending on the specific needs of each task. Both AMIRA and MADA also employ different tagsets. In the context of our introduction of ASMA, we have shown how it is that performance varies according to the size of the tagset used. To the best of our knowledge, this is the first report exploiting the CATiB tagset.

9 Conclusion and Future Work

In this paper, we have presented ASMA, a system for automatic segmentation and morpho-syntactic disambiguation of Modern Standard Arabic (MSA). We compared ASMA to the two most popular Arabic processing suites, AMIRA and MADA, and showed ASMA's advantages. ASMA has the advantages of speed as well as non-dependence on an external morphological analyzer (unlike MADA). It also identifies morpheme boundaries at a level more fine grained than AMIRA. Moreover, ASMA performs POS tagging with different degrees of granularity and hence can be customized according to an upstream task if used as a preprocessing system. For the future, we plan to investigate the utility of using a conditional random fields classifier either to complement or replace ASMA's current memory-based classifier. In addition, we will attempt to improve ASMA's performance based on our error analysis.

[4] *Nunation* indicates indefiniteness and refers to word-final diacritics occuring as a short vowel followed by an unwritten /n/ sound.

References

David Aha, Dennis Kibler, and Marc K. Albert. 1991. Instance-based learning algorithms. *Machine Learning*, 6:37–66.

Tim Buckwalter. 2004. Arabic morphological analyzer version 2.0. Linguistic Data Consortium.

Walter Daelemans and Antal van den Bosch. 2005. *Memory Based Language Processing*. Cambridge University Press, Cambridge, UK.

Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. MBT: A memory-based part of speech tagger-generator. In *Proceedings of the 4th Workshop on Very Large Corpora*, pages 14–27, Copenhagen, Denmark.

Walter Daelemans, Antal van den Bosch, and Jakub Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34(1-3):11–43. Special Issue on Natural Language Learning.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2004. Automatic tagging of Arabic text: From raw text to base phrase chunks. In *Proceedings of the 5th Meeting of the North American Chapter of the Association for Computational Linguistics and Human Language Technologies Conference (HLT-NAACL)*, pages 149–152, Boston, MA.

Mona Diab, Kadri Hacioglu, and Dan Jurafsky. 2007. Automatic processing of Modern Standard Arabic text. *Arabic Computational Morphology*, pages 159–179.

Mona Diab. 2007. Towards an optimal POS tag set for Modern Standard Arabic processing. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*.

Mona Diab. 2009. Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and base phrase chunking. In *2nd International Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Nizar Habash and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 573—580, Ann Arbor, MI.

Nizar Habash and Ryan Roth. 2009. CATiB: The Columbia Arabic Treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 221–224, Suntec, Singapore.

Nizar Habash, Owen Rambow, and Ryan Roth. 2009. MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Jan Hajič. 2000. Morphological tagging: Data vs. dictionaries. In *Proceedings of the1st conference of the North American Chapter of the Association for Computational Linguistics*, pages 94–101.

Sandra Kübler and Emad Mohamed. 2012. Part of speech tagging for Arabic. *Natural Language Engineering*, 18(4):521–548.

Mohamed Maamouri, Anne Bies, Tim Buckwalter, and W. Mekki. 2004. The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109.

Emad Mohamed and Sandra Kübler. 2010. Is arabic part of speech tagging feasible without word segmentation? In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 705–708. Association for Computational Linguistics.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the ACL 3rd Workshop on Very Large Corpora*, pages 82–94, Cambridge, MA.

Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 2008. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of ACL-08: HLT, Short Papers*, pages 117–120, Columbus, Ohio.

Beatrice Santorini. 1990. Part-of-speech tagging guidelines for the Penn Treebank Project. Department of Computer and Information Science, University of Pennsylvania, 3rd Revision, 2nd Printing.

Reut Tsarfaty, Djamé Seddah, Yoav Goldberg, Sandra Kübler, Marie Candito, Jennifer Foster, Yannick Versley, Ines Rehbein, and Lamia Tounsi. 2010. Statistical parsing of morphologically rich languages (SPMRL): What, how and whither. In *Proceedings of the NAACL Workshop on Statistical Parsing of Morphologically Rich Languages*, Los Angeles, CA.