Robust Part-of-Speech Tagging of Arabic Text

Hanan Aldarmaki and Mona Diab
Department of Computer Science
The George Washington University
{aldarmaki;mtdiab}@gwu.edu

Abstract

We present a new and improved part of speech tagger for Arabic text that incorporates a set of novel features and constraints. This framework is presented within the MADAMIRA software suite, a state-of-the-art toolkit for Arabic language processing. Starting from a linear SVM model with basic lexical features, we add a range of features derived from morphological analysis and clustering methods. We show that using these features significantly improves part-of-speech tagging accuracy, especially for unseen words, which results in better generalization across genres. The final model, embedded in a sequential tagging framework, achieved 97.15% accuracy on the main test set of newswire data, which is higher than the current MADAMIRA accuracy of 96.91% while being 30% faster.

1 Introduction

Part-of-speech (POS) tagging is an essential enabling technology and a precursor for most Natural Language Processing (NLP) tasks such as syntactic parsing, semantic role labeling, machine translation, and information extraction. POS tagging ranges in its complexity depending on the morphological richness of the targeted language. For morphologically rich languages, POS tagging poses a significant challenge, especially when moving away from formal textual genres to more informal genres. In this paper, we present a suite of linear supervised learning methods and features used to enhance POS tagging for Modern Standard Arabic (MSA) text using a relatively complex POS tag set. The novel POS tagger is presented within the context of the MADAMIRA suite framework (Pasha et. al., 2014). MADAMIRA is a combination of two well established approaches: AMIRA (Diab, 2009) and MADA (Roth et al, 2008). AMIRA is a relatively simple cascaded system that performs clitic segmentation, segmentation correction or normalization, and POS tagging as three separate steps performed sequentially, while MADA performs all three steps in one fell swoop. Both systems are based on supervised learning. However, the MADA approach relies on optimizing the results of a morphological analyzer while AMIRA does not rely on external resources. While the current MADAMIRA release only includes the MADA system, the goal is to combine both systems to improve MADAMIRA further. For the remainder of this paper, we will refer to the two systems as MD-MADA and MD-AMIRA, as both are presented within the MADAMIRA software suite.

In MD-AMIRA, POS tagging is implemented for MSA using linear classification with lexical features, and it results in reasonable accuracy within familiar contexts. However, the performance degrades when the model encounters words unseen in training. In this paper, we attempt to enhance the performance of this model, especially for unseen words, by including features from various external resources while maintaining the simplicity of the linear model. We refer to this enhanced model as MD-AMIRA+EX. Using a morphological analyzer, we extract morphological features as well as valid part-of-speech tags for input tokens. We also use these tags to impose soft constraints on the output. In addition, we include word clusters using two clustering methods applied to a large unlabeled data set. The basic POS tagging model and the additional features and constraints are described in Section 4.

MSA exhibits affixival and agglutinative morphology, where various forms of prepositions, articles, and pronouns are merged with words as clitics. A surface space-delimited word such as

Proceedings of the Second Workshop on Arabic Natural Language Processing, pages 173–182,
Beijing, China, July 26-31, 2015. ©2014 Association for Computational Linguistics

"wsyktbwnhA",[1] 'and they will write it', packs what would be considered several words in a language such as English. The different words are expressed as agglutinated morphemes or clitics broken up as follows: *"w+ s+ yktbwn +hA"*, 'and+ will+ write_plural +it_fem'. Due to limited amounts of labeled data, separating clitics from words, i.e. tokenization, is essential in reducing sparsity and enhancing the accuracy of POS tagging. In this paper, we address MSA tokenization as a precursor to POS tagging.

In MD-AMIRA, MSA tokenization is split into two steps: clitic segmentation and segmentation normalization. In the segmentation step, described in Section 3.1.1, we break each input word into segments corresponding to the clitics and the stem that make up that word. In MSA, some morphemes change in form as a result of affixation, and we render them to their original underlying forms in the segmentation normalization step, which is described in Section 3.1.2.

We evaluate the performance of the separate steps and the whole pipeline from tokenization to part-of-speech tagging in Sections 6 to 8. We show that this approach is robust and efficient as it compares to state of the art accuracy while exhibiting robust performance on unseen words.

2 Related Work

The sequential NLP process presented in this paper is adapted from the AMIRA toolkit, which is described in (Diab, 2009) and (Diab et al., 2004), and is publicly available. Another data-driven part-of-speech tagger for Arabic was presented in (Kopru, 2011), which uses an HMM to learn an efficient classifier using surface features.

The alternative approach is MADA, which relies on deep morphological analysis and disambiguation, as described in (Habah et al., 2005) and (Roth et al, 2008). Several SVM classifiers are trained to predict morphological features in the first stage. These features are then used to rank the morphological analyses retrieved from a dictionary, and the analysis with the highest score is taken as the final analysis for the given word. This deep analysis results in accurate and detailed tagging albeit slower than simple SVM methods. Finally, the problem of classification and incorporating structural constraints on the output is studied in (Punyakanok et al, 2005). This is related to the constrained POS tagging attempted here, where external inference is used to maintain consistency after learning. A related example of incorporating external resources to constrain the learned classifiers is presented in (Do and Roth, 2010)

3 Approach

We adapt the AMIRA tagger approach by using linear support vector machines (SVM) as our basic classification machinery for both MD-AMIRA and MD-AMIRA+EX. We approach both Tokenization and POS tagging as classification problems. The basic models directly follow the implementation details described in (Diab, 2009).

3.1 Tokenization

3.1.1 Clitic Segmentation

Clitics are independent meaning-bearing units that are phonologically and orthographically merged with words, either as prefixes (proclitics) or suffixes (enclitics). Clitics are different from derivational or inflectional affixes, which either change the meaning or the syntactic role of their stems and are not segmented here. A word, in this context, refers to a stem and its inflectional and derivational affixes, and clitic segmentation is the process of separating clitics from words. Since clitics have their own meaning and part-of-speech tags, separating them reduces sparsity in the input space.

In MSA, a word can have up to three proclitics and one enclitic. Table 1 shows some of the word classes that serve as clitics in MSA.

Type	Category	Examples
Proclitic	Definite article	Al
Proclitic	Prepositions	b, l, k
Proclitic	Conjunctions	w, f
Proclitic	Future marker	s
Enclitics	Pronouns	h, hm, hmA, etc.

Table 1: Examples of clitics in MSA

Set up: Segmentation is modeled as a classification problem at the character level, where each character is given a tag. Similar to the AMIRA framework, we adopt an IOB chunk/segment tagging scheme. The tag set is defined as follows:

[1] We transliterate Arabic text using Buckwalter romanization scheme: http://www.qamus.org

Tag set: {B-PRE, I-PRE, B-WORD, I-WORD, B-SUFF, I-SUFF, O}

WORD:	stem+inflectional affixes
PRE:	enclitic
SUFF:	proclitic
B- :	beginning of segment
I- :	Inside segment
O:	outside of segment (word boundaries)

The input consists of Buckwalter (BW) transliterated Arabic characters (Habah et al., 2007) with word boundary markers, preprocessed by digit normalization (converting all digits into '8') and removal of diacritics, if any. The features are as follows: (1) contextual features: [-5,+5] characters in context, the previous [-5, -1] tag decisions, and (2) lexical features: the whole space-delimited word, and character N-grams, N≤4, within that word.

3.1.2 Segmentation Normalization

Segmentation normalization is a correction step that attempts to restore citation forms of some words that have been transformed as a result of the morphotactics of clitic affixation. This task aims to reduce sparsity in the input space, and is inspired by the AMIRA tokenization lemmatization step (Diab et al., 2004), but we include additional forms of normalization. More details on Arabic orthographic and morphological adjustment rules can be found in (El Kholy and Habash, 2010).

Some forms of correction are deterministic, such as restoring the definite article *"Al"* ('the') from its reduced form *"l"* when it's preceded by the preposition *"l"* ('for'). Another example is the restoration of the trailing *"n"* in prepositions such as *"mn"* (from') and *"En"* ('about') when followed by the suffix *"mA"* ('what'), as in *"mmA"* → *"mn+mA"*. These are cases of clitic lemmatization, and since they are observed on a closed class of tokens, they are easily addressed as a post-tokenization processing step.

On the other hand, segmentation of open-class word forms cannot be restored deterministically. In MSA, words that end with the feminine marker character Taa Marbuta *"p"*, are transformed into *"t"* when followed by suffixes, as in: *"klmp"* ('word') → *"klmt+h"* ('his word'). A stem that ends with a *"t"* could either be a transformed *"p"* or a word that originally ends with *"t"*, as in *"byt"* (house). The other type of word ending that is transformed in affixation is the character Alef Maqsura *"Y"*, which is transformed into *"y"* or *"A"* if followed by suffixes (e.g.: *"ElY"* ('on'), → *"Ely+h"* ('on him')). The problem is more complex for words that can correspond to multiple lemmas such as *"ESAhm"*, which could correspond to the verb *"ESY+hm"* ('disobeyed+them') or the noun *"ESA+hm"* ('stick +their').

The restoration of Taa Marbuta and Alef Maqsura is not a deterministic process and it requires both contextual information and/or deeper knowledge of the language. The segmentation normalization step attempts to achieve this type of correction by learning to distinguish these cases from contextual data as follows.

Set up: The problem of segmentation normalization is addressed as a classification problem on the token level cascaded from the prior segmentation step. The input consists of a list of tokens, with proclitic and enclitic markers–the '+' marker indicating a segmentation point. The feature vector consists of [-2,+2] tokens in context, character N-grams, N≤4, for the current token, and the previous 2 tag decisions. Each token is assigned one of the following tags:

CIP:	Change trailing t to p
CIY:	Change trailing y or A to Y
NA:	Do nothing

4 Part of Speech Tagging

POS tagging is performed on the resulting tokenized text; that is, after performing clitic segmentation and segmentation normalization. The tag set used is a modified version of ERTS (Diab, 2007), which explicitly encodes several morphological features like determiner definiteness, gender, and number for nominals. We extend the tagset to include person, gender, number, and voice for verbs, and we refer to the new tagset as ERTS2. These fine-grained tags can be easily reduced to broad part-of-speech classes after prediction, which makes them suitable for a range of applications. They encode the full part-of-speech tag information provided in the LDC Arabic Treebank, excluding syntactic mood, syntactic case, and construct state definiteness. The following are some examples of ERTS2 tags, which illustrate the level of encoded details—refer to the appendix for a full listing of possible tags:

PV+3_MASC_SG: Third-person singular masculine perfective verb
IV_PASS+3_MASC_SG: Passive-voice masculine singular imperfective verb
ADJ+FEM_PL: Feminine plural adjective

The input to this classification problem consists of a list of digit-normalized tokens with explicit enclitic and proclitic affixes marked with '+' at segmentation points. The feature vector consists of [-2,+2] tokens in context, character N-grams, N$\leq$4, for the current token, the type of the current token {alpha, numeric}, and the previous 2 tag decisions.

We add two more components (constraints and features) over the MD-AMIRA POS tagging pipeline as follows.

4.1 ALMOR for Constrained Tagging

ALMOR (Habah, 2007) is a morphological analysis and generation system for MSA and dialectal Arabic. Given a word, ALMOR retrieves all possible analyses for that word and a list of characteristics, including part-of-speech tags, for each analysis. ALMOR constructs the analyses by generating all possible segmentations and verifying the validity and compatibility of the segments on an underlying database of valid stems and affixes.

In our POS tagging model, ALMOR is used as a source of external knowledge to constrain the statistical SVM tagger: the retrieved ALMOR part-of-speech tags are used as constraints on the SVM decision function by penalizing tags that do not appear in ALMOR analyses set. Given k tags, the POS tag y_i for a word w_i, as given by the original SVM decision function, is the tag with the maximum SVM score

$$\underset{y_i, i\in\{1,..,k\}}{\text{argmax}}\ (\text{score}(y_i))$$

Using ALMOR, we retrieve the set of possible part-of-speech tags, S_i, and penalize the tags that are not found in this set by reducing their SVM score. Accordingly, the final tag is given by the modified decision function:

$$\underset{y_i, i\in\{1,..,k\}}{\text{argmax}}\ (\text{score}(y_i) - \rho\, \mathrm{I}_{S_i^C}(y_i))$$

where I is the indicator function of the complement of S_i, and ρ is the penalty parameter. This modification is implemented only in the prediction step, so the experiment doesn't require re-training of the models.

4.2 Additional Features

The following sets of features were extracted from external resources and tested separately as well as in combination.

Morphological features: The top m part-of-speech tags from the set of analyses S_i, as described in the previous section, are used as features. The optimal number of tags, m, to include is tuned from the data. Additional morphological features extracted from ALMOR are voice, gender, person, and number.

Clustering features: We add cluster IDs retrieved from a large unlabeled dataset using two clustering methods: Brown clustering (Brown et al., 1992), and word2vec K-means clustering (Mikolov et. al., 2013).

Named-entity-related features: To support proper noun identification, we add binary features for exact and partial match in a gazetteer, and capitalization in the English gloss in any one of ALMOR analyses.

5 Experimental Set Up

5.1 Data set

The data sets used for training the models are LDC's Arabic Treebank (ATB) parts 1,2, and 3 (Maamouri et. al., 2004), which consist of MSA newswire data. The data is split as follows: 10% development set, 80% training set, and 10% test set. For cross-genre evaluation, we use the test set from ATB parts 5,6,7 and 10, which consist of MSA broadcast news and a small portion from the Weblog genres. The data sets are pre-processed using the approach described in (Habah et al., 2005) to correct annotation inconsistencies.

5.2 SVM Classification

Linear SVM classification is implemented using Cogent (Pasha et. al., 2014), a java utility and a wrapper around Liblinear (Fan et. al., 2008). Cogent pre-processes the input and converts text features into binary feature vectors for linear classification. In these experiments, Cogent is configured to keep a maximum of 100,000 features, so features are filtered to keep the maximum value within that range by removing the least frequent feature-value pairs. This limitation is imposed to keep the models more manageable during training and prediction.

6 Evaluation

The performance of our systems, MD-AMIRA and MD-AMIRA+EX, are evaluated and compared against the performance of MD-MADA (Pasha et. al., 2014), on the same tasks. MD-MADA produces highly sophisticated and accurate analysis of raw text, which includes a large number of morphological features reflecting the full spectrum of part-of-speech tags used in ATB, which is more specified than the ERTS2 tag set used in this work. Moreover, MD-MADA produces lemmas and their corresponding diacritization forms. We report comparative results on Tokenization and POS tagging using a subset of MD_-MADA outputs that correspond directly to our output specifications.

6.1 Clitic Segmentation

We evaluate the performance of MD-AMIRA as described in Section 3.1.1. Table 2 shows the overall performance of MD-AMIRA segmentation model compared with MD-MADA using the harmonic mean F-score metric. We perform clitic segmentation at the most detailed segmentation level, D3, which is ATB tokenization in addition to segmenting out the definite article Al (Habah et al., 2006). The overall F score of our linear segmentation is over 99 on ATB1-2-3 test set, comparable to the F score achieved by MD-MADA. Both models perform worse on cross-genre data, i.e. ATB5-6-7-10 test set, and MD-AMIRA performs worse on this set.

Model	**F score on test set**	
	ATB1-2-3	**ATB5-6-7-10**
MD-MADA	99.20	98.54
MD-AMIRA	99.24	97.76

Table 2: Overall segmentation performance on held-out test data

We report precision and recall results at the chunk level, PRE, WORD, SUFF in Figure 1. On ATB1-2-3 test set, MD-AMIRA has higher precision and lower recall rates over all segment types. On cross-genre data, MD-AMIRA precision drops for all types, with a notable drop in suffix segmentation. This set consists primarily of broadcast news transcriptions, and it includes filled pauses transcribed as ”>*h*” ('uh'), which are not encountered in the formal newswire training data. In MD-AMIRA, the ”*h*” in this interjection is incorrectly segmented as a possessive pronoun ”+*h*”, ('his'), and this is responsible for about 60% of the drop in suffix precision.

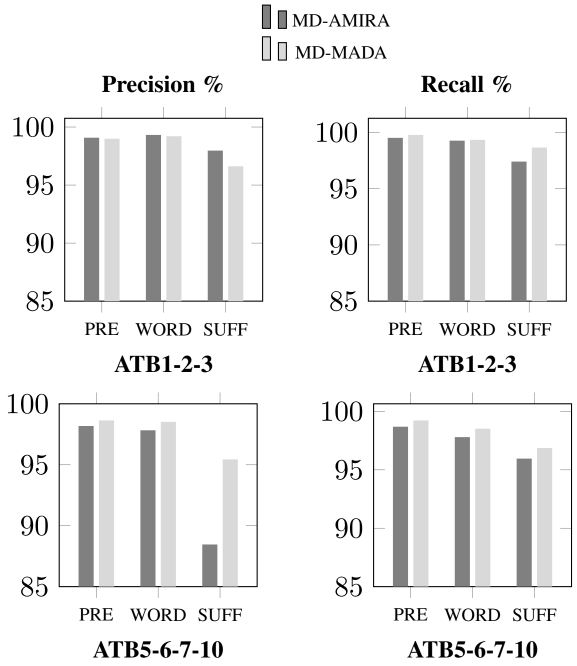

Figure 1: Chunk-level segmentation performance on held-out test data

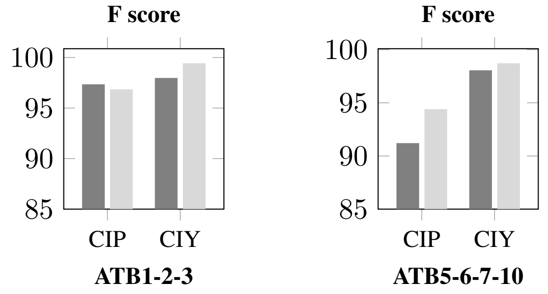

Figure 2: Segmentation Normalization performance on held-out test data

6.2 Segmentation Normalization

Figure 2 shows the performance of normalization conditions CIP and CIY using both systems on each test set. On ATB1-2-3 test set, the performance MD-AMIRA is comparable to MD-MADA. On cross-genre data, the performance of MD-AMIRA in CIP normalization is significantly lower than MD-MADA. Around half the errors in CIP identification are caused by words unseen in training since MD-AMIRA does not use any external resources in this step. Note that these results are evaluated after performing automatic segmentation with each system, so some errors are propagated from the clitic segmentation step.

7 Part of Speech Tagging

We first analyze the performance of the POS tagging module on the development set independently using gold tokenization. The purpose of this analysis is to tune the model without the effect of errors cascaded from automatic tokenization. In Section 8, we evaluate the performance of the finalized POS tagging model within the pipelined MD-AMIRA system and compare it with MD-MADA.

7.1 ALMOR Constrained Tagging

As discussed in Section 4.1, we modify the SVM scores to prioritize the tags retrieved by ALMOR. Figure 3 shows the performance as a function of ρ on the development set (the y-axis is divided and scaled for clarity). Without a constraint, the overall accuracy is 97.3%. Adding the constraints initially improves the overall accuracy, which peaks around $\rho = 1$, then drops considerably. Breaking up the accuracy on seen versus unseen words in training, the accuracy of unseen words increases generally, and is maximized around $\rho = 2$. For seen words, where the accuracy is close to 98% to start with, adding the constraints degrades performance.

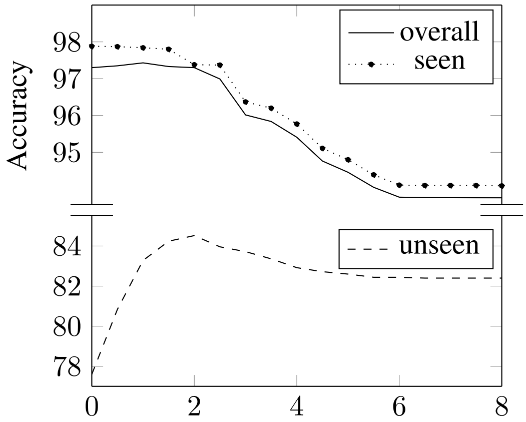

Figure 3: Accuracy as a function of ρ

Accordingly, we impose the constraints only on words that are unseen in training. This achieves an overall accuracy of 97.5%, which is a statistically significant improvement.[2]

7.2 ALMOR Tags as Features

An alternative use for the part-of-speech tags retrieved from ALMOR is to include the top m tags as features. Figure 4 shows the accuracy with m tags. Adding a single tag significantly improves the overall accuracy, which continues to improve up to $m = 4$. While adding more tags as features slightly improves the accuracy, limiting the number of retrieved tags improves the speed of the prediction model. Since the improvement beyond $m = 2$ is not statistically significant, we keep the number of tags at $m = 2$, which achieves an accuracy of 97.64%.

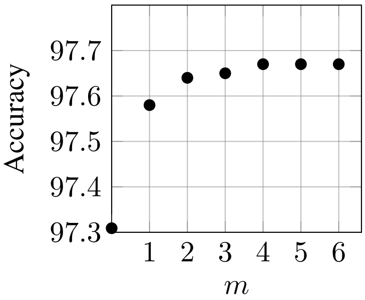

Figure 4: Accuracy as a function of m

7.3 Impact of additional features

In addition to POS tags, additional morphological features can be extracted from ALMOR analyses. We extracted the following set of features: number, gender, person, and voice, from the top two analyses, and included them as features on top of the basic set of lexical features.

We also experimented with a set of named-entity-related features: a binary feature for having a match in a set of gazetteers, and a binary feature for capitalization in the English gloss in one of ALMOR analyses (which is equivalent to having proper noun as one of the analyses). These features are added to help identify proper nouns. We tested these two sets of features separately, and the overall accuracy as well as the accuracy for unseen words are shown in Table 3.

Feature Set	Overall	Unseen
MD-AMIRA	96.58%	77.60
Morph. Features	97.50%	84.36%
NE Features	97.31%	77.76%

Table 3: Performance with Additional Features

[2] We test statistical significance using an exact test for one sample binomial proportions, at the 0.05 significance level.

Adding morphological features results in a statistically significant improvement in accuracy and around 7% reduction in error rate for unseen words. Named-entity features, on the other hand, do not improve performance. Both gazetteer matches and capitalization are features that could be triggered by adjectives, nouns, and proper nouns as they have similar word forms in MSA. The neutral result suggests that these features add noise which offsets any improvement from proper noun identification.

7.4 Clustering

We performed Brown clustering as well as Google's word2vec K-means clustering using an automatically tokenized version of LDC's Arabic Gigaword dataset (Graff, 2003). The number of clusters, k, is empirically set to 500. Table 4 shows the effect of adding these cluster IDs as features on top of the basic model. Both clustering methods result in a statistically significant improvement in accuracy, especially for unseen words. Combining both clustering methods as features achieves additional gains in performance, suggesting that the two clustering methods provide complementary information.

Clustering Method	Accuracy	
	Overall	Unseen
MD-AMIRA	97.31%	77.60%
+Brown Clustering	97.49%	83.96%
+Word2Vec	97.44%	82.36%
+Brown & Word2Vec	97.52%	84.76%

Table 4: POS Tagging Accuracy with Clustering Features

7.5 Combining Features

We now evaluate the models with a combination of these features. Table 5 shows the performance of the different models as evaluated on the development set. Starting from the basic model, MD-AMIRA, with only lexical features, we add the feature sets one at a time and compare the accuracy.

Each set of features incrementally improves the performance, and the highest improvement is achieved by adding two tags from ALMOR. Adding more features can improve the performance, but the improvements are less evident when combined with the existing features. Adding morphological features in M_2, for example, does not help since the morphological features are implied in the POS tags already included in M_1, and in this case the accuracy drops slightly. In M_3, where we combine POS tags, morphological features, and cluster IDs, the accuracy improves for unseen words, and it performs better than M_3b where we exclude morphological features.

In M_4, we re-tune the penalty parameter ρ over M_3. Adding this penalty does not significantly improve the performance as it is outweighed by the improvements from the other features. Moreover, adding these soft constraints reduces the speed of the prediction model. Thus, we choose M_3 as our final model–note that the difference between M_3 and M_3b is not statistically significant and both have the same prediction speed. Using M_3, the accuracy of tagging words unseen in training is around 90%, a considerable gain over the baseline. We use M_3 as the POS tagging model in MD-AMIRA+EX.

Model	Accuracy	
	Overall	Unseen Words
MD-AMIRA	97.309%	77.60%
M_1	97.637%	88.96%
M_2	97.628%	88.72%
M_3	97.682%	90.64%
M_3b	97.677%	90.40%
M_4	97.686%	90.76%

Table 5: Performance of POS tagging models on ATB1-2-3 development set. **MD-AMIRA**: baseline model with surface features. **M**$_1$: basic features + top two tags from ALMOR. **M**$_2$:The features in M_1 + morphological features. **M**$_3$: The features in M_2 + clustering. **M**$_3$b: The features in M_1 + clustering. **M**$_4$: The features in M_3 + penalty $\rho = 0.45$.

8 Overall Performance

We evaluate the performance of the system as a whole process from tokenization to part-of-speech tagging. The performance of our final system MD-AMIRA+EX on ATB1-2-3 held out test set is compared against two systems: the baseline of basic lexical features, MD-AMIRA, and the state-of-the-art system, MD-MADA. In order to compare MD-MADA to our system, we reduce the MD-MADA tag set to the ERTS2 tag set.

Table 6 shows the overall accuracy of these systems in addition to the tagging speed in tokens per

Model	ERTS Accuracy		Broad Tags [3]	Speed
	Overall	Unseen	Accuracy	tokens\sec
MD-AMIRA	96.78%	73.38%	98.01%	~2415
MD-AMIRA+EX	97.15%	89.22%	98.24%	~1395
MD-MADA	96.91%	85.28%	98.19%	~1050

Table 6: Performance on ATB1-2-3 held out test set

second, which is evaluated on the same hardware.

The fastest system is MD-AMIRA, which can tag at least 70% more tokens per second than the other models, but results in lower accuracy. The performance on unseen words, which make up about 2.5% of tokens in this set, is particularly bad. MD-AMIRA+EX processes about 30% more tokens per second than MD-MADA while achieving a higher accuracy in this set, which is statistically significant. The large improvement on unseen tokens reflects the generalization power of this model compared to the baseline. Note that in each model, some errors are due to segmentation, but since MD-AMIRA, MD-AMIRA+EX, and MD-MADA systems achieved high segmentation accuracy on this set, the effect is minimal. As a demonstration of this effect, MD-AMIRA+EX achieved 97.64% accuracy on this set using gold tokenization; segmentation errors reduced the overall accuracy by about 0.5%. For unseen words, the accuracy using gold tokenization is 91.78%, more than 3% relative increase in accuracy compared to automatic segmentation. This indicates that we have a relatively robust and efficient POS tagging model.

Most of the errors in POS tagging are due to confusion between the main classes: nouns, adjectives, and verbs. Interestingly, MD-MADA have lower recall for proper nouns than MD-AMIRA+EX. Table 7 shows the number of proper noun misclassifications using both systems. We only show the count of proper nouns that are incorrectly classified as either adjective or verb, which account for the majority of errors related to proper nouns. The table illustrate one category in which MD-AMIRA+EX outperform MD-MADA in POS tagging.

Model	ADJ	VERB
MD-AMIRA+EX	69	50
MD-MADA	125	107

Table 7: Proper noun misclassifications

Table 8 shows the accuracy on cross-genre data. MD-AMIRA+EX achieves a significantly higher accuracy than MD-AMIRA with a large improvement in accuracy for unseen words, which make up about 5% of tokens in this set. Compared to MD-MADA, MD-AMIRA+EX performed worse on this set. This decline in performance is mostly attributed to the segmentation errors from MD-AMIRA+EX tokenization, which is worse than MD-MADA tokenization in this set as shown in Section 6.1. Using gold tokenization, MD-AMIRA+EX resulted in 95.4% POS tagging accuracy on this set; segmentation errors reduced the overall accuracy by more than 1%. For unseen words, the accuracy using gold tokenization is 80.16%, an increase of more than 5% over the accuracy using automatic segmentation. This is further evidence that MD-AMIRA+EX POS tagging model is robust as it achieves close to state-of-the-art accuracy in spite of having more segmentation errors.

Model	ERTS Accuracy	
	Overall	Unseen Words
MD-AMIRA	93.35%	55.92%
MD-AMIRA+EX	94.38%	75.53%
MD-MADA	94.71%	77.19%

Table 8: Accuracy on cross-genre data

9 Conclusions

We experimented with various feature sets to improve the performance and generalization power of linear part-of-speech tagging. Adding a couple of part-of-speech tags from a morphological analyzer as features greatly reduced the error rate and achieved the largest gain in performance in our final model. Adding morphological features from the same analyzer, while it achieved significant improvements when tested separately, did not achieve large gains in the final model's accuracy

[3] broad part-of-speech classes, such as noun, verb, etc.

since these features are mostly redundant given the POS tags. Similarly, using the POS tags as soft constraints on the SVM decision function did not achieve significant gains on the model that already incorporates these tags as features. Adding cluster IDs, on the other hand, reduced the error rate, particularly for unseen words and genres, even when combined with the other features.

The clustering methods we experimented with were implemented using a large dataset of newswire data: the same genre used for training. To achieve better generalization over different genre, clustering data from various genre would be an interesting experiment for future work. Furthermore, part-of-speech tagging performance depends on the accuracy of segmentation. Our final model achieved lower accuracy on cross-genre data due to segmentation errors. Improving the performance of tokenization can be another way to improve the final model. Overall, the model achieved close to state-of-the-art performance and good generalization over unseen words while being reasonably fast.

References

P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. Computational Linguistics, 18(4):467-479, 1992.

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 , pages 149152. Association for Computational Linguistics, 2004.

M. Diab. Improved Arabic base phrase chunking with a new enriched pos tag set. In Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages.

M. Diab. Second generation amira tools for arabic processing: Fast and robust tokenization, pos tagging, and base phrase chunking. In Proceedings of the Second International Conference on Arabic Language Resources and Tools , pages 285288, 2009.

Q.X. Do and D. Roth. Constraints based taxonomic relation classification. In Proceedings of EMNLP2010 , pages 10991109. Association for Computational Linguistics, 2010.

A. El Kholy and N. Habash. Techniques for Arabic morphological detokenization and orthographic denormalization. In LREC 2010 Workshop on Language Resources and Human Language Technology for Semitic Languages.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A Library for Large Linear Classification. Journal of Machine Learning Research, 9:18711874. 2008.

David Graff. Arabic Gigaword, LDC Catalog No.: LDC2003T12. Linguistic Data Consortium, University of Pennsylvania. 2003.

N. Habash and O. Rambow. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In Proceedings of the ACL-05 , pages 573580. Association for Computational Linguistics, 2005.

N. Habash, and F. Sadat. Arabic Preprocessing Schemes for Statistical Machine Translation. In Proceedings of the 7th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL06), p. 4952, New York, NY. 2006.

N, Habash, A. Soudi, and T. Buckwalter. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, Arabic Computational Mor- phology: Knowledge-based and Empirical Methods. Springer (2007).

N. Habash. Arabic morphological representations for machine translation. In A. van den Bosch, A. Soudi (Eds.), Arabic Computational Morphology: Knowledge-based and Empirical Methods, Springer (2007).

M. Maamouri, A. Bies, and T. Buckwalter. The penn arabic treebank : Building a large- scale annotated arabic corpus. In NEMLAR Confer- ence on Arabic Language Resources and Tools, Cairo, Egypt. 2004.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.

S. Kopru. An efficient part-of-speech tagger for arabic. In Computational Linguistics and Intelligent Text Processing , pages 202213, 2011.

A. Pasha, M. Al-Badrashiny, M. Diab, A. El Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow, and R. Roth. MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014).

V. Punyakanok, D. Roth, W. Yih, and D. Zimak. Learning and inference over constrained output. In IJCAI , pages 11241129, 2005.

R. Roth, O. Rambow, N. Habash, M. Diab, and C. Rudin. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In Proceedings of ACL-08: HLT, Short Papers , pages 117120, 2008.

Appendix: ERTS2 Tagset

- ABBREV
- ADJ
- ADJ+FEM_DU
- ADJ+FEM_PL
- ADJ+FEM_SG
- ADJ+MASC_DU
- ADJ+MASC_PL
- ADJ_COMP
- ADJ_COMP+FEM_SG
- ADJ_COMP+MASC_PL
- ADJ_NUM
- ADJ_NUM+FEM_DU
- ADJ_NUM+FEM_PL
- ADJ_NUM+FEM_SG
- ADJ_NUM+MASC_DU
- ADJ_NUM+MASC_PL
- ADJ_VN
- ADJ_VN+FEM_DU
- ADJ_VN+FEM_PL
- ADJ_VN+FEM_SG
- ADJ_VN+MASC_DU
- ADJ_VN+MASC_PL
- ADV
- ADV_INTERROG
- ADV_REL
- CONJ
- CV
- CV+2_FEM_SG
- CV+2_MASC_PL
- CV+2_MASC_SG
- DET
- INTERJ
- IV
- IV+1_PL
- IV+1_SG
- IV+2_DU
- IV+2_FEM_PL
- IV+2_FEM_SG
- IV+2_MASC_PL
- IV+2_MASC_SG
- IV+3_FEM_DU
- IV+3_FEM_PL
- IV+3_FEM_SG
- IV+3_MASC_DU
- IV+3_MASC_PL
- IV+3_MASC_SG
- IV_PASS
- IV_PASS+1_PL
- IV_PASS+1_SG
- IV_PASS+2_FEM_SG
- IV_PASS+2_MASC_SG
- IV_PASS+3_FEM_SG
- IV_PASS+3_MASC_DU
- IV_PASS+3_MASC_PL
- IV_PASS+3_MASC_SG
- NOUN
- NOUN+FEM_DU
- NOUN+FEM_PL
- NOUN+FEM_SG
- NOUN+MASC_DU
- NOUN+MASC_PL
- NOUN+PRN+1_SG
- NOUN+PRN+1_SG+FEM_DU
- NOUN+PRN+1_SG+MASC_DU
- NOUN+PRN+1_SG+MASC_PL
- NOUN_NUM
- NOUN_NUM+FEM_DU
- NOUN_NUM+FEM_PL
- NOUN_NUM+FEM_SG
- NOUN_NUM+MASC_DU
- NOUN_NUM+MASC_PL
- NOUN_PROP
- NOUN_PROP+FEM_DU
- NOUN_PROP+FEM_PL
- NOUN_PROP+FEM_SG
- NOUN_PROP+MASC_DU
- NOUN_PROP+MASC_PL
- NOUN_QUANT
- NOUN_QUANT+FEM_SG
- NOUN_QUANT+MASC_DU
- NOUN_VN
- NOUN_VN+FEM_DU
- NOUN_VN+FEM_PL
- NOUN_VN+FEM_SG
- NOUN_VN+MASC_DU
- NOUN_VN+MASC_PL
- OTH
- PART
- PART_FOC
- PART_FUT
- PART_INTERROG
- PART_NEG
- PART_VERB
- PART_VOC
- PREP
- PREP+NOUN
- PREP+PRN+1_SG
- PRN
- PRN+1_PL
- PRN+1_SG
- PRN+2_DU
- PRN+2_FEM_PL
- PRN+2_FEM_SG
- PRN+2_MASC_PL
- PRN+2_MASC_SG
- PRN+3_DU
- PRN+3_FEM_PL
- PRN+3_FEM_SG
- PRN+3_MASC_PL
- PRN+3_MASC_SG
- PRN_DEM
- PRN_DEM+FEM
- PRN_DEM+FEM_DU
- PRN_DEM+FEM_SG
- PRN_DEM+MASC_DU
- PRN_DEM+MASC_PL
- PRN_DEM+MASC_SG
- PRN_DEM+PL
- PRN_DO+1_PL
- PRN_DO+1_SG
- PRN_DO+2_FEM_SG
- PRN_DO+2_MASC_PL
- PRN_DO+2_MASC_SG
- PRN_DO+3_DU
- PRN_DO+3_FEM_SG
- PRN_DO+3_MASC_PL
- PRN_DO+3_MASC_SG
- PRN_INTERROG
- PRN_INTERROG+FEM_SG
- PRN_REL
- PRN_REL+FEM_SG
- PUNC
- PV
- PV+1_PL
- PV+1_SG
- PV+2_FEM_SG
- PV+2_MASC_PL
- PV+2_MASC_SG
- PV+3_FEM_DU
- PV+3_FEM_PL
- PV+3_FEM_SG
- PV+3_MASC_DU
- PV+3_MASC_PL
- PV+3_MASC_SG
- PV_PASS
- PV_PASS+1_PL
- PV_PASS+1_SG
- PV_PASS+3_FEM_DU
- PV_PASS+3_FEM_PL
- PV_PASS+3_FEM_SG
- PV_PASS+3_MASC_DU
- PV_PASS+3_MASC_PL
- PV_PASS+3_MASC_SG
- SUB_CONJ