MADA+TOKAN:
A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization

Nizar Habash, Owen Rambow and Ryan Roth

Center for Computational Learning Systems
Columbia University
New York, NY, USA
{habash,rambow,ryanr}@ccls.columbia.edu

Abstract

We describe the MADA+TOKAN toolkit, a versatile and freely available system that can derive extensive morphological and contextual information from raw Arabic text, and then use this information for a multitude of crucial NLP tasks. Applications include high-accuracy part-of-speech tagging, diacritization, lemmatization, disambiguation, stemming, and glossing. MADA operates by examining a list of all possible analyses for each word, and then selecting the analysis that matches the current context best by means of support vector machine models classifying for 19 distinct, weighted morphological features. The selected analyses carry complete diacritic, lexemic, glossary and morphological information; thus all disambiguation decisions are made in one step. TOKAN takes the information provided by MADA to generate tokenized output in a wide variety of customizable formats. MADA, TOKAN and their support utilities are highly configurable, allowing users to extract and manipulate the exact information that they require. In this paper we describe the features and capabilities of MADA+TOKAN, detail recent improvements, and provide examples of the toolkit's use.

Introduction

MADA+TOKAN is a versatile, highly customizable and freely available toolkit for Arabic NLP applications. It consists of two components. MADA is a utility that, given raw Arabic text, adds as much lexical and morphological information as possible by disambiguating in one operation part-of-speech tags, lexemes, diacritizations and full morphological analyses. TOKAN is a utility that, given the information MADA produces, can generate a tokenization (sometimes also called a "segmentation") formatted exactly to user specifications. This tokenization also identifies the stem of the word. Together, these two programs provide an excellent preprocessing tool for major NLP applications such as machine translation (MT), automatic speech recognition (ASR), named entity recognition (NER) sand many others.

This paper is organized as follows. First, some difficulties of processing Arabic are explained. We describe the general strategy used by MADA+TOKAN for overcoming these challenges. In the MADA+TOKAN Toolkit section, we detail the operational aspects of each component of the toolkit. We then describe how MADA+TOKAN can be used in a variety of NLP applications, and present a case-study of an ASR/MT pipeline that utilizes the toolkit in several ways.

Challenges of Arabic Processing

The Arabic language raises many challenges for natural language processing (NLP). First, Arabic is a morphologically complex language. The morphological analysis of a word consists of determining the values of a large number of (partially orthogonal) features, such as basic part-of-speech (i.e., noun, verb, and so on), voice, gender, number, information about the clitics, and so on. For Arabic, this gives us about 333,000 theoretically possible completely specified morphological analyses. In contrast, English morphological tagsets usually have about 50 tags, which cover all morphological variations. Second, Arabic orthographic rules cause some parts of words to be deleted or modified when cliticization occurs. For example, the Ta-Marbuta (ة *ħ*)[1] appears as a regular Ta (ت *t*) when followed by a pronominal clitic. Simple segmentation of the pronominal clitic without recovering the Ta-Marbuta could cause unnecessary ambiguity or add to the sparsity problem. Third, Arabic is written with optional diacritics that specify short vowels and gemination (consonant doubling); they are usually absent in written Arabic, which contributes to ambiguity. Finally, the writing system also shows different levels of specificity in spelling some letters, e.g. Alef-Hamza-Above (أ *Â*) can be spelled without the Hamza (ء) as Alef (ا *A*), and Ya (ي *y*) can be spelled without the dots as Alef-Maqsura (ى *ý*). The complexity of the morphology together with the underspecification of the orthography creates a high degree of ambiguity. On average, a word form in the Penn Arabic Treebank (PATB; Maamouri et al, 2004)

[1] Our system internally uses the Buckwalter Arabic transliteration scheme (Buckwalter, 2004); however, examples of Arabic text in this document are presented in the Habash-Soudi-Buckwalter (HSB) transliteration scheme (Habash et al., 2007). This scheme extends Buckwalter's scheme to increase its readability while maintaining the 1-to-1 correspondence with Arabic orthography as represented in standard encodings of Arabic, such as Unicode. System-internal examples will be presented in the Buckwalter scheme. The following is the HSB transliteration map with different Buckwalter scheme values indicated in parentheses: آ Ā (|), أ Â (>), ؤ ŵ (&), إ Ǎ (<), ئ ŷ (}),ة ħ (p), ث θ (v), ذ ð (*),ش š ($), ظ Ď (Z), ع ς (E), غ γ (g), ى ý (Y), ي y, ـً ã (F), ـٌ ũ (N), ـٍ ĩ (K), ـْ . (o).

has about 12 morphological analyses. For example, the word والى *wAlý* can be analyzed as والي *wAly* 'ruler', و+الى+ي *w+Alý+y* 'and to me', و+ألي *w+Âly* 'and I follow', و+آل+ي *w+Āl+y* 'and my clan' or و+آلي *w+Āly* 'and automatic'. Each of these cases has a different diacritization.

Our Approach

Much work has been done on addressing different specific natural language processing tasks for Arabic, such as tokenization, diacritization, morphological disambiguation, part-of-speech (POS) tagging, stemming and lemmatization. The toolkit we present here provides one solution to all of these different problems. Our approach distinguishes between the problems of morphological analysis (*what are all the different readings of a word without regard to context*) and morphological disambiguation (*what is the correct reading in a specific context*). Once a morphological analysis is determined in context, we can determine its full POS tag, lemma and diacritization. Morphological analysis and disambiguation is handled in the MADA component of our toolkit. Knowing the morphological analysis also allows us to tokenize and stem deterministically. Since there are many different ways to tokenize Arabic (the tokenization scheme is determined by the needs of the application being developed and/or the linguistic theory being used), the TOKAN component is used to systematically define tokenization schemes that can be generated from disambiguated analyses. This gives our system a high degree of versatility and makes it very easy to use to evaluate different ways of processing Arabic to discover the optimal tokenization for a particular application.

Comparison to Related Work

Much work has been done in the area of Arabic morphological analysis and part-of-speech tagging (Al-Sughaiyer, and Al-Kharashi, 2004). A lot of the created systems tend to target a specific application or a POS tagset that is not general enough for different applications (e.g., Khoja, 2001, Darwish, 2002, Diab et al., 2007a). MADA+TOKAN, in contrast, does not assume that there is a single, small POS tagset, or that there is a single correct tokenization. In fact, Habash and Sadat (2006) and Diab (2007) demonstrate that different representations of morphology, whether tokenization schemes or POS tagsets perform differently on the same task. The MADA+TOKAN system allows researchers to quickly and easily explore a large space of possible tags and annotations. We can do this because we target the finest possible POS tagset for Arabic: it expresses *all* morphological differences. Mapping to coarse sets is rather simple and can be done in TOKAN. This obviously comes at an added computational cost compared to other approaches that target a specific tokenization/POS tagset. But note that MADA needs to be run only once to produce disambiguated morphological tags. TOKAN, which handles tokenization and mapping into coarser tagsets is quite quick to run. Another subtle but important distinction of how MADA approaches Arabic POS tagging is that many different tasks are solved in one fell swoop: tokenization, diacritization, full morphological disambiguation and thus POS tagging (to a number of tagsets), and even lemmatization. This contrasts with what most Arabic taggers do, which is to separate tokenization/stemming from POS tagging in two different steps (Khoja, 2001, Diab et al., 2007), which may pass on, and thus amplify, errors.

The type of solution for POS tagging explored in MADA is inspired by the seminal work of Hajic (2000) on morphological tagging for morphologically rich European languages using loglinear exponential models. Hajic later extended his work to Arabic in (Hajic 2005). Smith et al. (2005) presented a very similar setup to MADA's first version (Habash and Rambow, 2005), except for using Context Random Fields instead of Support Vector Machines. Note that Habash and Rambow (2005) did not separate the toolkit into MADA proper and TOKAN.

MADA+TOKAN Toolkit

MADA and TOKAN are packaged and continuously updated. The toolkit is freely available for research purposes. For details, documentation and a quick manual, see the MADA website (Habash et al., 2009).

MADA

MADA (Morphological Analysis and Disambiguation for Arabic) makes use of up to 19 orthogonal features to select, for each word, a proper analysis from a list of potential analyses provided by the Buckwalter Arabic Morphological Analyzer (BAMA; Buckwalter 2004). The BAMA analysis that most closely matches the collection of weighted, predicted features wins. The 19 features (shown in Table 1) include 14 morphological features that MADA predicts using 14 distinct Support Vector Machines (SVMs) trained on the PATB. In addition, MADA uses five features that capture information such as spelling variations and n-gram statistics.

Each analysis that MADA considers consists of the diacritized form of the word, its lexeme, its morphological features, and an English glossary entry:

```
<diac-form>=[<lexeme> <features>]=<gloss>
```

Once MADA ranks the possible analyses, it appends a numerical score to each analysis, and flags the top-scoring analysis for each word in each context. The user of MADA can extract any or all of the analysis information. This is why MADA can be used for a multitude of tasks, including part-of-speech and morphological feature tagging, lemmatization, predicting full diacritization, glossing, stemming and others.

Since MADA selects a complete analysis from BAMA, all decisions regarding morphological ambiguity, lexical ambiguity, tokenization, diacritization and POS tagging in any possible POS tagset are made in one fell swoop (Habash and Rambow, 2005; Habash and Rambow 2007; Roth et al, 2008). The choices are ranked in terms of their score. MADA has over 96% accuracy on basic morphological choice (including tokenization but excluding case, mood, and nunation) and on lemmatization. MADA has over 86% accuracy in predicting full diacritization (including case and mood). Detailed comparative evaluations are provided in the following publications: (Habash and Rambow, 2005; Habash and Rambow 2007; Roth et al, 2008).

The operation of MADA is versatile and highly configurable. Starting with version 2.0, MADA applies weights to each of the 19 features it uses for better accuracy; these weights were determined on a tuning set and are optimized for different purposes, such as tokenization, diacritization, or POS tagging. These weight sets are included with the package and should be chosen by the user depending on how MADA will be used. However, users can also choose to set these weights directly themselves. By default, MADA attempts to rank complete analyses in terms of overall correctness; in this weight set, MADA does not use the gender or idafa features (which, in the presence of the other 17 features, were found to be redundant). By choosing an alternative feature and weight set, it is possible to have MADA focus more specifically on getting a particular analysis aspect correct. For example, users can achieve a 0.4% absolute improvement in POS tagging accuracy if they use the weight set that was tuned for POS tagging, as opposed to the default set. However, the accuracy of the other MADA outputs (the lexeme prediction, for example) may suffer.

Starting with version 2.1, MADA also includes a morphological backoff procedure, which can be turned on or off by the user. In this procedure, MADA uses the ARAGEN version of the BAMA analyzer (Habash 2007) which can generate a morphological analysis even for words not covered by the lexicon. It does this by using the prefixes and suffixes in the BAMA databases, and by hypothesizing a stem (which is unattested). This allows MADA to generate its own morphological analyses to augment the ones produced by BAMA, and provide a selection for words that BAMA does not recognize. A full description can be found in (Habash and Rambow 2005).

Table 2 shows an example of an Arabic sentence, its English translation and its MADA output. For space reasons, only the three highest scoring analyses for each word is shown.

TOKAN

TOKAN, a general tokenizer for Arabic, provides an easy-to-use resource for tokenizing MADA disambiguated Arabic text into a large set of possible tokenization schemes (Habash 2007). For instance, the decision on whether an Arabic word has a conjunction or preposition clitic is made in MADA; TOKAN determines if and how such clitics are separated (accounting for various morphotactics and normalizations) before using them in an application. The different types of tokenizations can be used as machine learning features for a variety of applications, such as machine translation, or named-entity recognition.

TOKAN takes as input a MADA-disambiguated file and a tokenization scheme description that specifies how the tokenization is done. For instance, the scheme

"w+ f+ b+ k+ l+ s+ Al+ REST + / + POS +P: +O: -DIAC"

separates conjunctions, prepositions, verbal particles, the definite article and pronominal clitics and it adds the basic POS tag to the form of the word. The scheme also specifies that diacritics are generated. An analysis of the word وسيكاتبها *wasayukAtibuhA* `and he will correspond with her' would be tokenized as "wa+ sa+ yukAtibu/V +hA." A simpler scheme such as "w+ f+ REST" would simply produce "w+ sykAtbhA." See Sadat and Habash (2006) for a detailed description of several schemes that have become commonly followed since that work was published. TOKAN has a large number of other features that allow the user to perform different kinds of orthographic normalization or control how the output is presented as it may fit different needs of different systems.

Table 3 shows several example tokenization schemes associated with the MADA choices in Table 2.

Internally, TOKAN uses morphological generation to recreate the word once different clitics are split off. We do this to guarantee the form of the generated word is normalized and consistent with other occurrences of that word. For example, simply splitting the pronominal clitic off a word with Ta-Marbuta (ة *ħ*) would keep the Ta-Marbuta in its word-internal form (regular letter Ta, ت *t*). With TOKAN, the Ta-Marbuta is generated as appropriate (see the D3 tokenization shown in Table 3 which converts جولته *jwlth* 'his-visit' into جولة+ه *jwlħ +h* 'visit +his').

Useful Tools and Resources

Included with the MADA package are a number of small utilities that users have found to be useful. Scripts that perform simple conversion between Buckwalter and UTF-8 encoding are provided. In addition, there is a stem orthographic normalization utility that can be configured to run immediately after MADA completes.

MADA utilizes a set of Perl libraries to maintain and manipulate its internal data structures. Full documentation of these libraries is provided so that users can use them to readily develop their own scripts that process the information MADA produces.

Internally, MADA depends on a three resources that must be downloaded and installed separately. The first of these is the **Buckwalter Arabic Morphological Analyzer** (Buckwalter, 2004). The second is the **SRILM** toolkit, specifically the *disambig* utility (Stolcke, 2002). MADA uses this utility to construct lexeme n-grams. Finally, MADA currently uses the **SVMTools** package to operate its SVMs (Gimenez, 2004).

Examples of MADA+TOKAN Usability

MADA and TOKAN have been used by numerous academic and commercial research institutes around the world, including University of Washington, Cambridge University, SRI, BBN, Fair Isaac Inc., MIT, RWTH Aachen, Polytechnic University of Catalunya (UPC), Copenhagen Business School, and the National Research Center of Canada. The tools have been cited in numerous publications and have been shown to improve performance in a variety of NLP applications.

MADA+TOKAN for NLP applications

In the context of machine translation (MT) from Arabic to English, Habash and Sadat (2006) and Sadat and Habash (2006) explored the use of different preprocessing schemes and their combination. Their results have been followed by different groups of researchers working on Arabic-English MT, such as (Costa-Jussa, et al., 2006; Crego et al., 2006; Vilar et al., 2008). Diab et al. (2007b) explored the use of MADA-generated diacritizations for MT. Elming and Habash (2007) and Elming et al. (2008) improved automatic word alignment for Arabic-English MT using combinations of different tokenization schemes generated by MADA+TOKAN. See Habash (2007) for more details on different representations of Arabic morphology for MT. Badr et al. (2008) used MADA in the context of English-to-Arabic MT. MADA has also been used to produce features for Named Entity Recognition (NER) by Farber et al. (2008) and Benajiba et al (2008).

MADA is a useful resource not only for NLP applications but could also be used for language learning as its output can be used as a study/reading aid that provides contextual disambiguation.

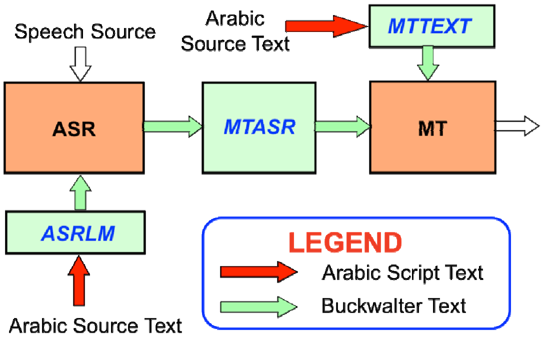

Figure 1: Example of an ASR/MT pipeline. MADA+TOKAN is used in the ASRLM, MTASR and MTTEXT subsystems.

Case Study of a Complex ASR+MT system

In the following example we show how MADA+TOKAN can be incorporated into an MT project that is based loosely on the SRI GALE project “Nightingale”.[2]. Figure 1 shows the overall architecture of the project. The MT process needs to make use of data from both text sources and audio sources via automatic speech recognition (ASR). The ASR process will process audio files, but needs to build reliable models from text sources first. The subsystems ASRLM and MTTEXT are meant to process raw Arabic script before passing important information to the ASR and MT components, respectively. The MTASR subsystem is meant to process ASR output for use in MT. All three subsystems use MADA+TOKAN components.

The ASRLM subsystem cleans the raw data and converts the UTF-8 encoding into Buckwalter transliteration. A separate utility is used to convert numeric digits to words (Habash and Roth, 2008), as is required for ASR. The subsystem then runs MADA and uses the toolkit’s stem orthographic normalization tool to remove spelling variations. The subsystem consequently runs TOKAN to produce an output suitable for ASR; here, TOKAN uses the READOFF scheme shown in Table 3 with Alef/Ya normalization. Finally, punctuation is removed. This provides the ASR system with nicely formatted, fully diacritized data, which is what the acoustic component of the ASR component produces.

The MTASR subsystem takes the output of ASR (which originally came form the audio files), cleans it, and runs MADA+TOKAN, using the D2 scheme shown in Table 3. Stem orthographic normalization is also used. The same numerical utility used in ASRLM is also used here to tag numerical expressions (which may be digital or expressed as words). The Buckwalter-transliterated data is converted back to UTF-8 prior to sending the data to the MT system.

The MTTEXT subsystem processes text for MT. It cleans the raw data and converts UTF-8 to Buckwalter transliteration. MADA+TOKAN (with stem orthographic normalization, number tagging and UTF-8 conversion) are used here to produce the same tokenization (D2) as the MTASR subsystem, making the output of MTTEXT and MTASR identical. The result is that the MT system can draw on similarly formatted ASR-derived and text-derived data for training and development.

Acknowledgements

The authors would like to thank Kristen Parton for her contributions. This work has been supported, in part, by NSF Award 0329163, Defense Advanced Research Projects Agency Contract No. HR0011-06-C-0023, and Defense Advanced Research Projects Agency Contract No. HR0011-08-C-0110.

[2] To our knowledge, there is, at present, no single publication summarizing the entire project. The system shown in Figure 1 was the product of a large team effort led by SRI International (http://www.speech.sri.com/projects/GALE/).

Bibliographical References

Al Sughaiyer, I. and I. Al Kharashi. 2004. Arabic Morphological Analysis Techniques: A Comprehensive Survey. Journal of the American Society for Information Science and Technology. Volume 55, Issue 3.

Badr, I., R. Zbib and J. Glass. 2008. Segmentation for English-to-Arabic Statistical Machine Translation. In *Proceedings of the Conference of Association for Computational Linguistics (ACL).*

Benajiba, Y., M. Diab and P. Rosso. 2008. Arabic Named Entity Recognition using Optimized Feature Sets. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*.

Buckwalter, T. 2004. Buckwalter Arabic Morphological Analyzer Version 2.0. Linguistic Data Consortium (LDC) catalogue number LDC2004L02, ISBN 1-58563-324-0.

Costa-Jussa, M. R., J. M. Crego, A. de Gispert, P. Lambert, M. Khalilov, J. A. R. Fonollosa, J. B. Marino and R. Banchs. 2006. TALP Phrase-Based System and TALP System Combination for IWSLT 2006. In *Proceedings of the International Workshop on Spoken Language Translation*.

Crego, J. M., A. de Gispert, P. Lambert, M. Khalilov, M. R. Costa-jussa, J. B. Marino, R. Banchs and J. A. R. Fonollosa. The TALP Ngram-based SMT System for IWSLT 2006. In *Proceedings of the International Workshop on Spoken Language Translation*.

Darwish, K. 2002. Building a Shallow Morphological Analyzer in One Day. In *Proceedings of ACL workshop on Computational Approaches to Semitic Languages.*

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

Diab, M., M. Ghoneim and N. Habash. 2007b. Arabic Diacritization in the Context of Statistical Machine Translation, In *Proceedings of the Machine Translation Summit (MT-Summit).*

Diab, M., K. Hacioglu and D. Jurafsky. 2007a. “Automated Methods for Processing Arabic Text: From Tokenization to Base Phrase Chunking.” Book Chapter. In Arabic Computational Morphology: Knowledge-based and Empirical Methods. Editors A. van den Bosch and A. Soudi.

Elming, J. and N. Habash. 2007. Combination of Statistical Word Alignments Based on Multiple Preprocessing Schemes, In *Proceedings of the North American chapter of the Association for Computational Linguistics (NAACL).*

Elming, J., N. Habash and J. Crego. 2008. “Combination of Statistical Word Alignments Based on Multiple Preprocessing Schemes.” Book Chapter. In Learning for Machine Translation. Editors C. Goutte, N. Cancedda, M. Dymetman, and G. Foster.

Farber, B., D. Freitag, N. Habash and O. Rambow. 2008. Improving NER in Arabic Using a Morphological Tagger. In *Proceedings of the Language Resources and Evaluation Conference (LREC).*

Gimenez, J. and L. Marquez. 2004. SVMTool: A general POS tagger generator based on Support Vector Machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC).*

Habash, N. 2007. "Arabic Morphological Representations for Machine Translation." Book Chapter. In Arabic Computational Morphology: Knowledge-based and Empirical Methods. Editors A. van den Bosch and A. Soudi.

Habash, N. and O. Rambow. 2005. Arabic Tokenization, Morphological Analysis, and Part-of-Speech Tagging in One Fell Swoop. In *Proceedings of the Conference of American Association for Computational Linguistics (ACL05).*

Habash, N. and O. Rambow. 2007. Arabic Diacritization through Full Morphological Tagging. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT).*

Habash, N., O. Rambow and R. Roth. 2009. “MADA+TOKAN” Website: http://www1.ccls.columbia.edu/~cadim/MADA.

Habash, N. and R. Roth. 2008. Identification of naturally occurring numerical expressions in Arabic. In *Proceeedings of the Language Resources and Evaluation Conference (LREC).*

Habash, N. and F. Sadat. 2006. Arabic Preprocessing Schemes for Statistical Machine Translation, In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL).*

Habash, N., A. Soudi, and T. Buckwalter. 2007. “On Arabic Transliteration.” Book Chapter. In *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Editors A. van den Bosch and A. Soudi.

Hajič, J. 2000. Morphological tagging: Data vs. dictionaries. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL’00).*

Hajič, J., O. Smrž, T. Buckwalter, and H. Jin. 2005. Feature-based tagger of approximations of functional Arabic morphology. In *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT).*

Khoja, S. 2001. APT: Arabic Part-of-Speech Tagger. In *Proceedings of NAACL Student Research Workshop*.

Maamouri, M., A. Bies, T. Buckwalter, W. Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *NEMLAR Conference on Arabic Language Resources and Tools.*

Roth, R., O. Rambow, N. Habash, M. Diab, and C. Rudin. 2008. Arabic Morphological Tagging, Diacritization, and Lemmatization Using Lexeme Models and Feature Ranking. In *Proceedings of Association for Computational Linguistics (ACL).*

Sadat, F. and N. Habash. 2006. Combination of Preprocessing Schemes for Statistical MT. In *Proceedings of COLING-ACL.*

Smith, N., D. Smith, R. Tromble. 2005. Context-based morphological disambiguation with random fields. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP).*

Stolcke, A. 2002. Srilm – an extensible language toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP).*

Vilar, D., D. Stein, Y. Zhang, E. Matusov, A. Mauser, O. Bender, S. Mansour and H. Ney. 2008. The RWTH Machine Translation System for IWSLT 2008. In *Proceedings of the International Workshop on Spoken Language Translation.*

APPENDIX: TABLES

Feature	AKA	Description	Predicted With
pos	POS	Part-of-Speech (e.g., N, AJ, V, PRO, etc.)	SVM
conj	CNJ	Presence of a conjunction (w+ or f+)	SVM
part	PRT	Presence of a particle clitic (b+, k+, l+)	SVM
clitic	PRO	Presence of a pronominal clitic (object or possessive)	SVM
art	DET	Presence of definite article (Al+)	SVM
gen	GEN	Gender (FEM or MASC)	SVM
num	NUM	Number (SG, DU, PL)	SVM
per	PER	Person (1,2,3)	SVM
voice	VOX	Voice (PASS or ACT)	SVM
aspect	ASP	Aspect (CV, IV, PV)	SVM
mood	MOD	Mood (I, S, J, SJ)	SVM
def	NUN	Presence of nunation (DEF or INDEF)	SVM
idafa	CON	Construct state (POSS or NOPOSS)	SVM
case	CAS	Case (ACC, GEN, NOM)	SVM
unigramlex		Lexeme predicted by a unigram model of lexemes	N-gram
unigramdiac		Diacritic form predicted by a unigram model of diacritic forms	N-gram
ngramlex		Lexeme predicted by an N-gram model of lexemes	N-gram
isdefault		Boolean: Whether the analysis a default BAMA output	Deterministic
spellmatch		Boolean: Whether the diacritic form is a valid spelling match	Deterministic

Table 1: Features used in MADA. The first column shows the name of the feature as it appears in the MADA documentation, output and configuration files; the second shows an alternative label used in some publications. These features roughly correspond to the features represented in the BAMA analysis format. The first ten are easier to predict than the next four since they rely on more visible inflectional morphology; the final five are supplementary features that are not predicted with SVMs.

INPUT	wsynhY	Alr}rys	jwlth	bzyArp	AlY	trkyA	.
GLOSS	and will finish	the president	tour his	with visit	to	Turkey	.
ENGLISH	The president will finish his tour with a visit to Turkey.						

```
;;; SENTENCE wsynhY Alr}ys jwlth bzyArp AlY trkyA .
;;WORD wsynhY
;;MADA: wsynhY art-NA aspect-IV case-NA clitic-NO conj-YES def-NA mood-I num-SG part-NO per-3 pos-V
       voice-ACT
*0.930061 wasayunohiy=[>anohaY_1 POS:V +IV s+ MOOD:I +S:3MS w+
       BW:wa/CONJ+sa/FUT+yu/IV3MS+nohiy/IV+(null)/IVSUFF_MOOD:I]=complete/finish/communicate
^0.780654 wasayanohaY=[nahaY-i_1 POS:V +IV s+ MOOD:I +S:3MS w+
       BW:wa/CONJ+sa/FUT+ya/IV3MS+nohaY/IV+(null)/IVSUFF_MOOD:I]=forbid/restrain
_0.739338 wasayunohaY=[>anohaY_1 POS:V +IV s+ +PASS MOOD:I +S:3MS w+
       BW:wa/CONJ+sa/FUT+yu/IV3MS+nohaY/IV_PASS+(null)/IVSUFF_MOOD:I]=be_completed/be_communicated
[ ... 7 additional options omitted ...]
;;WORD Alr}ys
;;MADA: Alr}ys art-YES aspect-NA case-NOM clitic-NO conj-NO def-DEF mood-NA num-SG part-NO per-3
       pos-N voice-NA
*0.823716 Alr~a}iysu=[ra}iys_1 POS:N Al+ +NOM +DEF
       BW:Al/DET+ra}iys/NOUN+u/CASE_DEF_NOM]=head/chairman/president
_0.798404 Alr~a}iysi=[ra}iys_1 POS:N Al+ +GEN +DEF
       BW:Al/DET+ra}iys/NOUN+i/CASE_DEF_GEN]=head/chairman/president
_0.768109 Alr~a}iysa=[ra}iys_1 POS:N Al+ +ACC +DEF
       BW:Al/DET+ra}iys/NOUN+a/CASE_DEF_ACC]=head/chairman/president
[ ... 1 additional option omitted ... ]
;;WORD jwlth
;;MADA: jwlth art-NO aspect-NA case-ACC clitic-YES conj-NO def-DEF mood-NA num-SG part-NO per-3 pos-N
       voice-NA
*0.887482 jawolatahu=[jawolap_1 POS:N +FEM +SG +ACC +DEF +P:3MS
       BW:+jawol/NOUN+at/NSUFF_FEM_SG+a/CASE_DEF_ACC+hu/POSS_PRON_3MS]=round/patrol/tour
_0.862169 jawolatihi=[jawolap_1 POS:N +FEM +SG +GEN +DEF +P:3MS
       BW:+jawol/NOUN+at/NSUFF_FEM_SG+i/CASE_DEF_GEN+hi/POSS_PRON_3MS]=round/patrol/tour
_0.831874 jawolatuhu=[jawolap_1 POS:N +FEM +SG +NOM +DEF +P:3MS
       BW:+jawol/NOUN+at/NSUFF_FEM_SG+u/CASE_DEF_NOM+hu/POSS_PRON_3MS]=round/patrol/tour
[ ... 4 additional options omitted ... ]
;;WORD bzyArp
;;MADA: bzyArp art-NO aspect-NA case-GEN clitic-NO conj-NO def-INDEF mood-NA num-SG part-YES per-3
       pos-N voice-NA
*0.930061 biziyArapK=[ziyArap_2 POS:N +FEM +SG +GEN +INDEF b+
       BW:bi/PREP+ziyAr/NOUN+ap/NSUFF_FEM_SG+K/CASE_INDEF_GEN]=visit
^0.780654 biziyArapK=[ziyArap_1 POS:N +FEM +SG +GEN +INDEF b+
       BW:bi/PREP+ziyAr/NOUN+ap/NSUFF_FEM_SG+K/CASE_INDEF_GEN]=visiting
_0.704130 biziyArapi=[ziyArap_2 POS:N +FEM +SG +GEN +DEF b+
       BW:bi/PREP+ziyAr/NOUN+ap/NSUFF_FEM_SG+i/CASE_DEF_GEN]=visit
[ ... 3 additional options omitted ... ]
;;WORD AlY
;;MADA: AlY art-NA aspect-NA case-NA clitic-NO conj-NO def-NA mood-NA num-NA part-NO per-NA pos-P
       voice-NA
*0.794149 <ilaY=[<ilaY_1 POS:P BW:+<ilaY/PREP+]=to/towards
_0.704073 <ilay~a=[<ilaY_1 POS:P +O:1S BW:+<ilay/PREP+~a/PRON_1S]=to/towards
_0.412087 |liy~a=[|liy~_1 POS:AJ +ACC +DEF BW:+|liy~/ADJ+a/CASE_DEF_ACC]=automatic/mechanical
[ ... 9 additional options omitted ... ]
;;WORD trkyA
;;MADA: trkyA art-NO aspect-NA case-NOCASE clitic-NO conj-NO def-DEF mood-NA num-SG part-NO per-3
       pos-PN voice-NA
*0.794340 turokiyA=[turokiyA_1 POS:PN BW:+turokiyA/NOUN_PROP+]=Turkey
_0.582330 turokiy~AF=[turokiy~_1 POS:N +ACC +INDEF BW:+turokiy~/NOUN+AF/CASE_INDEF_ACC]=Turk
_0.582330 turokiy~A=[turokiy~_1 POS:N +MASC +DU +NOM +POSS
       BW:+turokiy~/NOUN+A/NSUFF_MASC_DU_NOM_POSS]=Turk
[ .. 2 additional options omitted ... ]
;;WORD .
;;MADA: . art-NA aspect-NA case-NA clitic-NA conj-NO def-NA mood-NA num-NA part-NO per-NA pos-PX
       voice-NA
*0.999541 .=[. POS:PX]=.
SENTENCE BREAK
```

Table 2: Example of MADA output for a single sentence. The ";;MADA" line for each word indicates the predictions of the SVM classifiers. Each analysis is preceded by its score; the chosen analysis is marked with a '*'. For space reasons, only the 3 top scoring analyses for each word are shown.

ARABIC	وسينهى الرئيس جولته بزيارة الى تركيا						
ORIGINAL	wsynhý	Alrŷys	jwlth	bzyArħ	Alý	trkyA	.
GLOSS	and will finish	the president	tour his	with visit	to	Turkey	.
ENGLISH	The president will finish his tour with a visit to Turkey.						
SCHEME	BASELINE						
D1	w+ synhy	Alrŷys	jwlth	bzyArħ	Ǎlý	trkyA	.
D2	w+ s+ ynhy	Alrŷys	jwlth	b+ zyArħ	Ǎlý	trkyA	.
TBold	w+ synhy	Alrŷys	jwlħ + h	b+ zyArħ	Ǎlý	trkyA	.
TB	w+ s+ ynhy	Alrŷys	jwlħ + h	b+ zyArħ	Ǎlý	trkyA	.
D3	w+ s+ ynhy	Al+ rŷys	jwlħ + h	b+ zyArħ	Ǎlý	trkyA	.
EN	w+ s+ Ânhý/VBP +S:3MS	Al+ rŷys/NN	jwlħ/NN + h	b+ zyArħ/NN	Ǎlý /IN	trkyA/NNP	.
READOFF	wasayun.hiy	Alr~aŷiy.su	jaw.latahu	biziyAraħĩ	Ǎilaý	tur.kiyA	.

Table 3: Examples of TOKAN output for several tokenization schemes. D1 only splits off conjunction clitics (w+ and f+). D2 splits conjunctions and particles (l+, k+, b+, s+). TB is the Penn Arabic Treebank tokenization (TBold is the pre-2009 version of this tokenization). D3 does the same as D2, plus the definite article Al+ and all pronominal enclitics. EN is an English-like scheme that extends on D3 and replaces words with their lexeme and Bies POS tag. The READOFF scheme instructs TOKAN to simply output the fully-diacritized forms of the words without any segmentation.