MAGEAD: A Morphological Analyzer and Generator for the Arabic Dialects

Nizar Habash and Owen Rambow
Center for Computational Learning Systems
Columbia University
New York, NY 10115, USA
{habash,rambow}@cs.columbia.edu

Abstract

We present MAGEAD, a morphological analyzer and generator for the Arabic language family. Our work is novel in that it explicitly addresses the need for processing the morphology of the dialects. MAGEAD performs an on-line analysis to or generation from a root+pattern+features representation, it has separate phonological and orthographic representations, and it allows for combining morphemes from different dialects. We present a detailed evaluation of MAGEAD.

1 Introduction

In this paper we present MAGEAD, a morphological analyzer and generator for the Arabic language family, by which we mean both Modern Standard Arabic (MSA) and the spoken dialects.[1] Our work is novel in that it explicitly addresses the need for processing the morphology of the dialects as well.

The principal theoretical contribution of this paper is an organization of morphological knowledge for processing multiple variants of one language family. The principal practical contribution is the first morphological analyzer and generator for an Arabic dialect that includes a root-and-pattern analysis (which is also the first wide-coverage implementation of root-and-pattern morphology for any language using a multitape finite-state machine). We also provide a novel type of detailed evaluation in which we investigate how different sources of lexical information affect performance of morphological analysis.

This paper is organized as follows. In Section 2, we present the relevant facts about morphology in the Arabic language family. Previous work is summarized in Section 3. We present our design goals in Section 4, and then discuss our approach to representing linguistic knowledge for morphological analysis in Section 5. The implementation is sketched in Section 6. We outline the steps involved in creating a Levantine analyzer in Section 7. We evaluate our system in Section 8, and then conclude.

2 Arabic Morphology

2.1 Variants of Arabic

The Arabic-speaking world is characterized by diglossia (Ferguson, 1959). Modern Standard Arabic (MSA) is the shared written language from Morocco to the Gulf, but it is not a native language of anyone. It is spoken only in formal, scripted contexts (news, speeches). In addition, there is a continuum of spoken dialects (varying geographically, but also by social class, gender, etc.) which are native languages, but rarely written (except in very informal contexts: collections of folk tales, newsgroups, email, etc). We will refer to MSA and the dialects as **variants** of Arabic. Variants differ phonologically, lexically, morphologically, and syntactically from one another; many pairs of variants are mutually unintelligible. In unscripted situations where spoken MSA would normally be required (such as talk shows on TV), speakers usually resort to repeated code-switching between their dialect and MSA, as nearly all native speakers of Arabic are unable to produce sustained spontaneous discourse in MSA.

[1] We would like to thank several anonymous reviewers for comments that helped us improve this paper. The work reported in this paper was supported by NSF Award 0329163, with additional work performed under the DARPA GALE program, contract HR0011-06-C-0023. The authors are listed in alphabetical order.

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 681–688,
Sydney, July 2006.

In this paper, we discuss MSA and Levantine, the dialect spoken (roughly) in Syria, Lebanon, Jordan, Palestine, and Israel. Our Levantine data comes from Jordan. The discussion in this section uses only examples from MSA, but all variants show a combination of root-and-pattern and affixational morphology and similar examples could be found for Levantine.

2.2 Roots, Patterns and Vocalism

Arabic morphemes fall into three categories: templatic morphemes, affixational morphemes, and non-templatic word stems (NTWSs). NTWSs are word stems that are not constructed from a root/pattern/vocalism combination. Verbs are never NTWSs.

Templatic morphemes come in three types that are equally needed to create a word stem: roots, patterns and vocalisms. The root morpheme is a sequence of three, four, or five consonants (termed *radicals*) that signifies some abstract meaning shared by all its derivations. For example, the words كتب *katab* 'to write', كاتب *kaAtib* 'writer', and مكتوب *maktuwb* 'written' all share the root morpheme *ktb* (ك ت ب) 'writing-related'. The pattern morpheme is an abstract template in which roots and vocalisms are inserted. The vocalism morpheme specifies which short vowels to use with a pattern. We will represent the pattern as a string made up of numbers to indicate radical position, of the symbol *V* to indicate the position of the vocalism, and of pattern consonants (if needed).

A word stem is constructed by interleaving the three types of templatic morphemes. For example, the word stem كتب *katab* 'to write' is constructed from the root *ktb* (ك ت ب), the pattern *1V2V3* and the vocalism *aa*.

2.3 Affixational Morphemes

Arabic affixes can be prefixes such as *sa+* (+س) 'will/[future]', suffixes such as *+uwna* (+ون) '[masculine plural]' or circumfixes such as *ta++na* (ت++ن) '[imperfective subject 2nd person fem. plural]'. Multiple affixes can appear in a word. For example, the word وسيكتبونها *wasayaktubuwnahA* 'and they will write it' has two prefixes, one circumfix and one suffix:[2]

(1) wasayaktubuwnahA
wa+ sa+ y+ aktub +uwna
and will 3person write masculine-plural
+hA
it

2.4 Morphological Rewrite Rules

An Arabic word is constructed by first creating a word stem from templatic morphemes or by using a NTWS. Affixational morphemes are then added to this stem. The process of combining morphemes involves a number of phonological, morphemic and orthographic rules that modify the form of the created word so it is not a simple interleaving or concatenation of its morphemic components.

An example of a phonological rewrite rule is the voicing of the /t/ of the verbal pattern *V1tV2V3* (Form VIII) when the first root radical is /z/, /d/, or /*/ (ز, د, or ذ): the verbal stem *zhr+V1tV2V3+iaa* is realized phonologically as /izdahar/ (orthographically: ازدهر) 'flourish' not /iztahar/ (orthographically: ازتهر). An example of an orthographic rewrite rule is the deletion of the Alif (ا) of the definite article morpheme *Al+* (+ال) in nouns when preceded by the preposition *l+* (+ل).

3 Previous Work

There has been a considerable amount of work on Arabic morphological analysis; for an overview, see (Al-Sughaiyer and Al-Kharashi, 2004). We summarize some of the most relevant work here.

Kataja and Koskenniemi (1988) present a system for handling Akkadian root-and-pattern morphology by adding an additional lexicon component to Koskenniemi's two-level morphology (1983). The first large scale implementation of Arabic morphology within the constraints of finite-state methods is that of Beesley et al. (1989) with a 'detouring' mechanism for access to multiple lexica, which gives rise to other works by Beesley (Beesley, 1998) and, independently, by Buckwalter (2004).

The approach of McCarthy (1981) to describing root-and-pattern morphology in the framework of autosegmental phonology has given rise to a number of computational proposals. Kay (1987) proposes a framework with which each of the autosegmental tiers is assigned a tape in a multi-tape finite state machine, with an additional tape for the surface form. Kiraz (2000,2001) extends Kay's

[2]We analyze the imperfective word stem as including an initial short vowel, and leave a discussion of this analysis to future publications.

approach and implements a small working multitape system for MSA and Syriac. Other autosegmental approaches (described in more details in Kiraz 2001 (Chapter 4)) include those of Kornai (1995), Bird and Ellison (1994), Pulman and Hepple (1993), whose formalism Kiraz adopts, and others.

4 Design Goals for MAGEAD

This work is aimed at a unified processing architecture for the morphology of all variants of Arabic, including the dialects. Three design goals follow from this overall goal:

• First, we want to be able to use the analyzer when we do not have a lexicon, or only a partial lexicon. This is because, despite the similarities between dialects at the morphological and lexical levels, we do cannot assume we have a complete lexicon for every dialect we wish to morphologically analyze. As a result, we want an **on-line analyzer** which performs full morphological analysis at run time.

• Second, we want to be able to exploit the existing regularities among the variants, in particular systematic sound changes which operate at the level of the radicals, and pattern changes. This requires an **explicit analysis into root and pattern**.

• Third, the dialects are mainly used in spoken communication and in the rare cases when they are written they do not have standard orthographies, and different (inconsistent) orthographies may be used even within a single written text. We thus need a representation of morphology that incorporates **models of both phonology and orthography**.

In addition, we add two general requirements for morphological analyzers. First, we want both a morphological analyzer and a morphological generator. Second, we want to use a representation that is defined in terms of a lexeme and attribute-value pairs for morphological features such as aspect or person. This is because we want our component to be usable in natural language processing (NLP) applications such as natural language generation and machine translation, and the lexeme provides a usable lexicographic abstraction. Note that the second general requirement (an analysis to a lexemic representation) appears to clash with the first design desideratum (we may not have a lexicon).

We tackle these requirements by doing a full analysis of templatic morphology, rather than "precompiling" the templatic morphology into stems and only analyzing affixational morphology on-line (as is done in (Buckwalter, 2004)). Our implementation uses the multitape approach of Kiraz (2000). This is the first large-scale implementation of that approach. We extend it by adding an additional tape for independently modeling phonology and orthography. The use of finite state technology makes MAGEAD usable as a generator as well as an analyzer, unlike some morphological analyzers which cannot be converted to generators in a straightforward manner (Buckwalter, 2004; Habash, 2004).

5 The MAGEAD System: Representation of Linguistic Knowledge

MAGEAD relates (bidirectionally) a lexeme and a set of linguistic features to a surface word form through a sequence of transformations. In a generation perspective, the features are translated to abstract morphemes which are then ordered, and expressed as concrete morphemes. The concrete templatic morphemes are interdigitated and affixes added, and finally morphological and phonological rewrite rules are applied. In this section, we discuss our organization of linguistic knowledge, and give some examples; a more complete discussion of the organization of linguistic knowledge in MAGEAD can be found in (Habash et al., 2006).

5.1 Morphological Behavior Classes

Morphological analyses are represented in terms of a lexeme and features. We define the **lexeme** to be a triple consisting of a root (or an NTWS), a meaning index, and a **morphological behavior class** (MBC). We do not deal with issues relating to word sense here and therefore do not further discuss the meaning index. It is through this view of the lexeme (which incorporates productive derivational morphology without making claims about semantic predictability) that we can both have a lexeme-based representation, and operate without a lexicon. In fact, because lexemes have internal structure, we can hypothesize lexemes on the fly without having to make wild guesses (we know the pattern, it is only the root that we are guessing). We will see in Section 8 that this approach does not wildly overgenerate.

We use as our example the surface form ازدهرت *Aizdaharat* (*Azdhrt* without diacritics)

'she/it flourished'. The lexeme-and-features representation of this word form is as follows:

(2) Root:zhr MBC:verb-VIII POS:V PER:3 GEN:F NUM:SG ASPECT:PERF

An MBC maps sets of linguistic feature-value pairs to sets of abstract morphemes. For example, MBC verb-VIII maps the feature-value pair ASPECT:PERF to the abstract root morpheme [PAT_PV:VIII], which in MSA corresponds to the concrete root morpheme *AV1tV2V3*, while the MBC verb-I maps ASPECT:PERF to the abstract root morpheme [PAT_PV:I], which in MSA corresponds to the concrete root morpheme *1V2V3*. We define MBCs using a hierarchical representation with non-monotonic inheritance. The hierarchy allows us to specify only once those feature-to-morpheme mappings for all MBCs which share them. For example, the root node of our MBC hierarchy is a word, and all Arabic words share certain mappings, such as that from the linguistic feature conj:w to the clitic *w+*. This means that all Arabic words can take a cliticized conjunction. Similarly, the object pronominal clitics are the same for all transitive verbs, no matter what their templatic pattern is. We have developed a specification language for expressing MBC hierarchies in a concise manner. Our hypothesis is that the MBC hierarchy is variant-independent, though as more variants are added, some modifications may be needed. Our current MBC hierarchy specification for both MSA and Levantine, which covers only the verbs, comprises 66 classes, of which 25 are abstract, i.e., only used for organizing the inheritance hierarchy and never instantiated in a lexeme.

5.2 Ordering and Mapping Abstract and Concrete Morphemes

To keep the MBC hierarchy variant-independent, we have also chosen a variant-independent representation of the morphemes that the MBC hierarchy maps to. We refer to these morphemes as **abstract morphemes** (AMs). The AMs are then ordered into the surface order of the corresponding concrete morphemes. The ordering of AMs is specified in a variant-independent context-free grammar. At this point, our example (2) looks like this:

(3) [Root:zhr][PAT_PV:VIII]
[VOC_PV:VIII-act] + [SUBJSUF_PV:3FS]

Note that as the root, pattern, and vocalism are not ordered with respect to each other, they are simply juxtaposed. The '+' sign indicates the ordering of affixational morphemes. Only now are the AMs translated to **concrete morphemes** (CMs), which are concatenated in the specified order. Our example becomes:

(4) <zhr,AV1tV2V3,iaa> +at

The interdigitation of root, pattern and vocalism then yields the form *Aiztahar+at*.

5.3 Morphological, Phonological, and Orthographic Rules

We have two types of rules. **Morphophonemic/phonological rules** map from the morphemic representation to the phonological and orthographic representations. This includes default rules which copy roots and vocalisms to the phonological and orthographic tiers, and specialized rules to handle hollow verbs (verbs with a glide as their middle radical), or more specialized rules for cases such as the pattern consonant change in Form VIII (the /t/ of the pattern changes to a /d/ if the first radical is /z/, /d/, or /*/; this rule operates in our example). For MSA, we have 69 rules of this type.

Orthographic rules rewrite only the orthographic representation. These include, for examples, rules for using the *shadda* (consonant doubling diacritic). For MSA, we have 53 such rules.

For our example, we get /izdaharat/ at the phonological level. Using standard MSA diacritized orthography, our example becomes *Aizdaharat* (in transliteration). Removing the diacritics turns this into the more familiar ازدهرت *Azdhrt*. Note that in analysis mode, we hypothesize all possible diacritics (a finite number, even in combination) and perform the analysis on the resulting multi-path automaton.

6 The MAGEAD System: Implementation

We follow (Kiraz, 2000) in using a multitape representation. We extend the analysis of Kiraz by introducing a fifth tier. The five tiers are used as follows: Tier 1: pattern and affixational morphemes; Tier 2: root; Tier 3: vocalism; Tier 4: phonological representation; Tier 5: orthographic representation. In the generation direction, tiers 1 through 3 are always input tiers. Tier 4 is first an output tier, and subsequently an input tier. Tier 5 is always an output tier.

We have implemented multi-tape finite state automata as a layer on top of the AT&T two-tape finite state transducers (Mohri et al., 1998). We have defined a specification language for the higher multitape level, the new **Morphtools format**. Specification in the Morphtools format of different types of information such as rules or context-free grammars for morpheme ordering are compiled to the appropriate Lextools format (an NLP-oriented extension of the AT&T toolkit for finite-state machines, (Sproat, 1995)). For reasons of space, we omit a further discussion of Morphtools. For details, see (Habash et al., 2005).

7 From MSA to Levantine

We modified MAGEAD so that it accepts Levantine rather than MSA verbs. Our effort concentrated on the orthographic representation; to simplify our task, we used a diacritic-free orthography for Levantine developed at the Linguistic Data Consortium (Maamouri et al., 2006). Changes were done only to the representations of linguistic knowledge at the four levels discussed in Section 5, not to the processing engine.

Morphological Behavior Classes: The MBCs are variant-independent, so in theory no changes needed to be implemented. However, as Levantine is our first dialect, we expand the MBCs to include two AMs not found in MSA: the aspectual particle and the postfix negation marker.

Abstract Morpheme Ordering: The context-free grammar representing the ordering of AMs needed to be extended to order the two new AMs, which was straightforward.

Mapping Abstract to Concrete Morphemes: This step requires four types of changes to a table representing this mapping. In the first category, the new AMs require mapping to CMs. Second, those AMs which do not exist in Levantine need to be mapped to zero (or to an error value). These are dual number, and subjunctive and jussive moods. Third, in Levantine some AMs allow additional CMs in allomorphic variation with the same CMs as seen in MSA. This affects three object clitics; for example, the second person masculine plural, in addition to كم +*kum* (also found in MSA), also can be كو +*kuwA*. Fourth, in five cases, the subject suffix in the imperfective is simply different for Levantine. For example, the second person feminine singular indicative imperfective suffix changes from ين+ +*iyna* in MSA to ي+ +*iy* in Levantine. Note that more changes in CMs would be required were we completely modeling Levantine phonology (i.e., including the short vowels).

Morphological, Phonological, and Orthographic Rules. We needed to change one rule, and add one. In MSA, the vowel between the second and third radical is deleted when they are identical ("gemination") only if the third radical is followed by a suffix starting with a vowel. In Levantine, in contrast, gemination always happens, independently of the suffix. If the suffix starts with a consonant, a long /e/ is inserted *after* the third radical. The new rule deletes the first person singular subject prefix for the imperfective, +ا *A*+, when it is preceded by the aspectual marker +ب *b*+.

We summarize now the expertise required to convert MSA resources to Levantine, and we comment on the amount of work needed for adding a further dialect. We modified the MBC hierarchy, but only minor changes were needed. We expect only one major further change to the MBCs, namely the addition of an indirect object clitic (since the indirect object in some dialects is sometimes represented as an orthographic clitic). The AM ordering can be read off from examples in a fairly straightforward manner; the introduction of an indirect object AM would, for example, require an extension of the ordering specification. The mapping from AMs to CMs, which is variant-specific, can be obtained easily from a linguistically trained (near-)native speaker or from a grammar handbook, and with a little more effort from an informant. Finally, the rules, which again can be variant-specific, require either a good morphophonological treatise for the dialect, a linguistically trained (near-)native speaker, or extensive access to an informant. In our case, the entire conversion from MSA to Levantine was performed by a native speaker linguist in about six hours.

8 Evaluation

The goal of the evaluation is primarily to investigate how reduced lexical resources affect the performance of morphological analysis, as we will not have complete lexicons for the dialects. A second goal is to validate MAGEAD in analysis mode by comparing it to the Buckwalter analyzer (Buckwalter, 2004) when MAGEAD has a full lexicon at its disposal. Because of the lack of resources for the dialects, we use primarily MSA for both goals, but we also discuss a more modest evaluation on a

Levantine corpus.

We first discuss the different sources of lexical knowledge, and then present our evaluation metrics. We then separately evaluate MSA and Levantine morphological analysis.

8.1 Lexical Knowledge Sources

We evaluate the following sources of lexical knowledge on what roots, i.e, combinations of radicals, are possible. Except for **all**, these are lists of attested verbal roots. It is not a trivial task to compile a list of verbal roots for MSA, and we compare different sources for these lists.

- **all**: All radical combinations are allowed, we use no lexical knowledge at all.
- **dar**: List of roots extracted by (Darwish, 2003) from *Lisan Al'arab*, a large Arabic dictionary.
- **bwl**: A list of roots appearing as comments in the Buckwalter lexicon (Buckwalter, 2004).
- **lex**: Roots extracted by us from the list of lexeme citation forms in the Buckwalter lexicon using surfacy heuristics for quick-and-dirty morphological analysis.
- **mbc**: This is the same list as **lex**, except that we pair each root with the MBCs with which it was seen in the Buckwalter lexicon (recall that for us, a lexeme is a root with an MBC). Note that **mbc** represents a full lexicon, though it was converted automatically from the Buckwalter lexicon and it has not been hand-checked.

8.2 Test Corpora and Metrics

For development and testing purposes, we use MSA and Levantine. For MSA, we use the Penn Arabic Treebank (ATB) (Maamouri et al., 2004). The morphological annotation we use is the "before-file", which lists the untokenized words (as they appear in the Arabic original text) and all possible analyses according to the Buckwalter analyzer (Buckwalter, 2004). The analysis which is correct for the given token in its context is marked; sometimes, it is also hand-corrected (or added by hand), while the contextually incorrect analyses are never hand-corrected. For development, we use ATB1 section 20000715, and for testing, Sections 20001015 and 20001115 (13,885 distinct verbal types).

For Levantine, we use a similarly annotated corpus, the Levantine Arabic Treebank (LATB) from the Linguistic Data Consortium. However, there are three major differences: the text is transcribed speech, the corpus is much smaller, and, since, there is no morphological analyzer for Levantine currently, the before-files are the result of running the MSA Buckwalter analyzer on the Levantine token, with many of the analyses incorrect, and only the analysis chosen for the token in context usually hand-corrected. We use LATB files fsa_16* for development, and for testing, files fsa_17*, fsa_18* (14 conversations, 3,175 distinct verbal types).

We evaluate using three different metrics. The token-based metrics are the corresponding type-based metric weighted by the number of occurrences of the type in the test corpus.

- Recall (**TyR** for type recall, **ToR** for token recall): what proportion of the analyses in the gold standard does MAGEAD get?
- Precision (**TyP** for type precision, **ToP** for token precision): what proportion of the analyses that MAGEAD gets are also in the gold standard?
- Context token recall (**CToR**): how often does MAGEAD get the contextually correct analysis for that token?

We do not give context precision figures, as MAGEAD does not determine the contextually correct analysis (this is a tagging problem). Rather, we interpret the context recall figures as a measure of how often MAGEAD gets the most important of the analyses (i.e., the correct one) for each token.

	Roots	TyR	TyP	ToR	ToP	CToR
all	21952	98.5	44.8	98.6	36.9	97.9
dar	10377	98.1	50.5	98.3	43.3	97.7
bwl	6450	96.7	52.2	97.2	42.9	96.7
lex	3658	97.3	55.6	97.3	49.2	97.5
mbc	3658	96.1	63.5	95.8	59.4	96.4

Figure 1: Results comparing MAGEAD to the Buckwalter Analyzer on MSA for different root restrictions, and for different metrics; "Roots" indicates the number of possible roots for that restriction; all numbers are percent figures

8.3 Quantitative Analysis: MSA

The results are summarized in Figure 1. We see that we get a (rough) recall-precision trade-off, both for types and for tokens: the more restrictive we are, the higher our precision, but recall declines. For **all**, we get excellent recall, and an overgeneration by a factor of only 2. This performance, assuming it is roughly indicative of dialect performance, allows us to conclude that we can use MAGEAD as a dialect morphological analyzer without a lexicon.

For the root lists, we see that precision is al-

ways higher than for **all**, as many false analyses are eliminated. At the same time, some correct analyses are also eliminated. Furthermore, **bwl** under performs somewhat. The change from **lex** to **mbc** is interesting, as **mbc** is a true lexicon (since it does not only state which roots are possible, but also what their MBC is). Precision increases substantially, but not as much as we had hoped. We investigate the errors of **mbc** in the next subsection in more detail.

8.4 Qualitative Analysis: MSA

The gold standard we are using has been generated automatically using the Buckwalter analyzer. Only the contextually correct analysis has been hand-checked. As a result, our quantitative analysis in Section 8.3 leaves open the question of how good the gold standard is in the first place. We analyzed all of the 2,536 false positives (types) produced by MAGEAD on our development set (analyses it suggested, but which the Test corpus did not have). In 75% of the errors, the Buckwalter analyzer does not provide a passive voice analysis which differs from the active voice one only in diacritics which are not written. 7% are cases where Buckwalter does not make distinctions that MAGEAD makes (e.g. mood variations that are not phonologically realized); in 4.4% of the errors a correct analysis was created but it was not produced by Buckwalter for various reasons. If we count these cases as true positives rather than as false positives (as in the case in Figure 1) and take type frequency into account, we obtain a token precision rate of 94.9% on the development set.

The remaining cases are MAGEAD errors. 3.3% are missing rules to handle special cases such as jussive mood interaction with weak radicals; 5.4% are incorrect combinations of morphemes such as passive voice and object pronouns; 2.6% of the errors are cases of pragmatic overgeneration such as second person masculine subjects with a second person feminine plural object. 1.5% of the errors are errors of the mbc-root list and 1.2% are other errors. A large number of these errors are fixable errors.

There were 162 false negatives (gold standard analyses MAGEAD did not get). 65.4% of these errors were a result of the use of the **mbc** list restriction. The rest of the errors are all a result of unhandled phenomena in MAGEAD: quadrilateral roots (13.6%), imperatives (8%), and specific missing rules/ rule failures (13%) (e.g., for handling some weak radicals/hamza cases, pattern IX gemination-like behavior, etc.).

We conclude that we can claim that our precision numbers are actually much higher, and that we can further improve them by adding more rules and knowledge to MAGEAD.

8.5 Quantitative and Qualitative Analysis: Levantine

For the Levantine, we do not have a list of all possible analyses for each word in the gold standard: only the contextually appropriate analysis is hand-checked. We therefore only report context recall in Figure 2. As a baseline, we report the MSA MAGEAD with the **all** restriction applied to the same Levantine test corpus. As we can see, the MSA system performs poorly on Levantine input. The Levantine system we use is the one described in Section 7. We use the resulting analyzer with the **all** option as we have no information on roots in Levantine. MAGEAD with Levantine knowledge does well, missing only one in 20 contextually correct analyses. We take this to mean that the architecture of MAGEAD allows us to port MAGEAD fairly rapidly to a new dialect and to perform adequately well on the most important analysis for each token, the contextually relevant one.

System	CTyR	CToR
MSA-all	52.9	60.4
LEV-all	95.4	94.2

Figure 2: Results on Levantine; MSA-all is a baseline

For the Levantine MAGEAD, there were 25 errors, cases of contextually selected analyses that MAGEAD did not get (false negatives). Most of these are related to phenomena that MAGEAD doesn't currently handle: imperatives (48%) (which are much more common in speech corpora) and quadrilateral roots (8%). There were four cases (16%) of an unhandled variant spelling of an object pronoun and 7 cases (28%) of hamza/weak radical rule errors.

9 Outlook

We have described a morphological analyzer for Arabic and its dialects which decomposes word forms into the templatic morphemes and relates

morphemes to strings. We have evaluated the current state of the implementation both for MSA and for Levantine, both quantitatively and in a detailed error analysis, and have shown that we have met our design objectives of having a flexible analyzer which can be used on a new dialect in the absence of a lexicon and with a restrained amount of manual knowledge engineering needed.

In ongoing work, we are populating MAGEAD with more knowledge (morphemes and rules) for MSA nouns and other parts of speech, for more of Levantine, and for more dialects. We intend to include a full phonological representation for Levantine (including short vowels). In future work, we will investigate the derivation of words with morphemes from more than one variant (code switching). We will also investigate ways of using morphologically tagged corpora to assign weights to the arcs in the transducer so that the analyses returned by MAGEAD are ranked.

References

Imad A. Al-Sughaiyer and Ibrahim A. Al-Kharashi. 2004. Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology*, 55(3):189–213.

K. Beesley, T. Buckwalter, and S. Newton. 1989. Two-level finite-state analysis of Arabic morphology. In *Proceedings of the Seminar on Bilingual Computing in Arabic and English*, page n.p.

K. Beesley. 1998. Arabic morphology using only finite-state operations. In M. Rosner, editor, *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, pages 50–7, Montereal.

S. Bird and T. Ellison. 1994. One-level phonology. *Computational Linguistics*, 20(1):55–90.

Tim Buckwalter. 2004. Buckwalter Arabic morphological analyzer version 2.0.

Kareem Darwish. 2003. Building a shallow Arabic morphological analyser in one day. In *ACL02 Workshop on Computational Approaches to Semitic Languages*, Philadelpia, PA. Association for Computational Linguistics.

Charles F Ferguson. 1959. Diglossia. *Word*, 15(2):325–340.

Nizar Habash, Owen Rambow, and Geroge Kiraz. 2005. Morphological analysis and generation for arabic dialects. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Ann Arbor, MI.

Nizar Habash, Owen Rabmow, and Richard Sproat. 2006. The representation of linguistic knowledge in a pan-Arabic morphological analyzer. Paper under preparation, Columbia University and UIUC.

Nizar Habash. 2004. Large scale lexeme based arabic morphological generation. In *Proceedings of Traitement Automatique du Langage Naturel (TALN-04)*. Fez, Morocco.

L. Kataja and K. Koskenniemi. 1988. Finite state description of Semitic morphology. In *COLING-88: Papers Presented to the 12th International Conference on Computational Linguistics*, volume 1, pages 313–15.

Martin Kay. 1987. Nonconcatenative finite-state morphology. In *Proceedings of the Third Conference of the European Chapter of the Association for Computational Linguistics*, pages 2–10.

George Anton Kiraz. 2000. Multi-tiered nonlinear morphology using multi-tape finite automata: A case study on Syriac and Arabic. *Computational Linguistics*, 26(1):77–105.

George Kiraz. 2001. *Computational Nonlinear Morphology: With Emphasis on Semitic Languages*. Cambridge University Press.

A. Kornai. 1995. *Formal Phonology*. Garland Publishing.

K. Koskenniemi. 1983. *Two-Level Morphology*. Ph.D. thesis, University of Helsinki.

Mohamed Maamouri, Ann Bies, and Tim Buckwalter. 2004. The Penn Arabic Treebank: Building a large-scale annotated arabic corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, Mona Diab, Nizar Habash, Owen Rambow, and Dalila Tabessi. 2006. Developing and using a pilot dialectal arabic treebank. In *Proceedings of LREC*, Genoa, Italy.

John McCarthy. 1981. A prosodic theory of nonconcatenative morphology. *Linguistic Inquiry*, 12(3):373–418.

M. Mohri, F. Pereira, and M. Riley. 1998. A rational design for a weighted finite-state transducer library. In D. Wood and S. Yu, editors, *Automata Implementation*, Lecture Notes in Computer Science 1436, pages 144–58. Springer.

S. Pulman and M. Hepple. 1993. A feature-based formalism for two-level phonology: a description and implementation. *Computer Speech and Language*, 7:333–58.

Richard Sproat. 1995. Lextools: Tools for finite-state linguistic analysis. Technical Report 11522-951108-10TM, Bell Laboratories.