Arabic Morphological Representations for Machine Translation

Nizar Habash
Center for Computational Learning Systems
Columbia University
habash@cs.columbia.edu

1 Introduction

There has been extensive work on Arabic morphology, lexicography and syntax resulting in many resources (morphological analyzers, dictionaries, treebanks, etc.). These resources often adopt various representations that are not necessarily compatible with each other. For example, dictionaries use the notion of a *lexeme* that is different from the root/pattern/vocalism and stem/affix representations used by many morphological analyzers. Statistical approaches, such as statistical parsing or statistical machine translation, can be content with an inflected undiacritized word stem as the proper level of representation for Arabic. The result is that for researchers working on machine translation (MT), there is a need to relate multiple representations used by different resources (e.g., parser or dictionary) to each other within a single system. This chapter describes the different morphological representations used by MT-relevant natural language processing (NLP) resources and tools and their usability in different MT approaches for Arabic. With a special focus on symbolic MT, we motivate the lexeme-and-feature level of representation and describe and evaluate ALMORGEANA, a large-scale system for analysis and generation from/to that level. ALMORGEANA's wide-range coverage in terms of representations and its bidirectionality makes it a desirable tool for relating different resources available to MT researchers/developers who work with Arabic as a source or target language.

Section 2 introduces different representations in Arabic morphology. Section 3 discusses approaches to MT and how they interact with the different representa-

tions. Section 4 and Section 5 describe ALMORGEANA and how it can be used for navigating among different representations, respectively.

2 Representations of Arabic Morphology

In discussing representations of Arabic morphology, it is important to separate two different aspects of morphemes: *type* versus *function*. Morpheme *type* refers to the different kinds of morphemes and their interactions with each other. A distinguishing feature of Arabic (in fact, Semitic) morphology is the presence of templatic morphemes in addition to affixational morphemes. Morpheme *function* refers to the distinction between derivational morphology and inflectional morphology. These two aspects, type and function, are independent, i.e., a morpheme type does not determine its function and vice versa. This independence complicates the task of deciding on the proper representation of morphology in different NLP resources and tools. This section introduces these two aspects and their interactions in more detail.

2.1 Morpheme Type: Templatic vs. Affixational

Arabic has seven types of morphemes that fall into three categories: templatic morphemes, affixational morphemes, and non-templatic word stems (NTWS). Templatic morphemes come in three types that are equally needed to create a templatic word stem: roots, patterns and vocalisms. Affixes can be classified into prefixes, suffixes and circumfixes, which either precede, follow or surround the word stem, respectively. Finally NTWS are word stems that are not constructed from a root/pattern/vocalism combination. The following three subsections discuss each of the morpheme categories. This is followed by a brief discussion of some phonological, morphological, and orthographic adjustment phenomena that occur when combining morphemes to form words.

2.1.1 Roots, Patterns and Vocalism

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 (ك ت ب) *ktb* ’writing-related’.

The pattern morpheme is an abstract template in which roots and vocalisms are inserted. In this chapter, the pattern is represented as a string of letters including special symbols to mark where root radicals and vocalisms are inserted. Numbers, (i.e. 1, 2, 3, 4, or 5), are used to indicate radical position[1] and the symbol *V* is used to indicate vocalism position. For example, the verbal pattern *1V22V3* (Form II) indicates that the second root radical is doubled. A pattern can have additional consonants and vowels, e.g., the verbal pattern *Ai1tV2V3* (Form VIII).

The vocalism morpheme specifies which vowels to use with a pattern.[2] A word stem is constructed by interleaving a root, a pattern and a vocalism. For example, the word stem كتب *katab* 'to write' is constructed from the root كتب *ktb*, the pattern *1V2V3* and the vocalism *aa*. Another example, is the word stem استعمل *Aistuçmil* 'to be used', which is constructed from the root عمل *çml* 'work-related', the pattern *AistV12V3* and the vocalism *ui*.

2.1.2 Affixational Morphemes

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

(1) wa+ sa+ y+ aktub +uwna +haA
and+ will+ 3rd+ write +plural +it

'And they will write it'

Some of the affixes can be thought of as orthographic clitics, such as the conjunction +و *wa+* 'and', the prepositions (+ل *li+* 'to/for', +ب *bi+* 'in/with' and +ك *ka+* 'like') and the pronominal object/possessive clitics (e.g. +ها *+haA* 'her/it/its'). Others are bound morphemes.

2.1.3 Non-Templatic Word Stem

NTWS are word stems that are not derivable from templatic morphemes. They tend to be foreign names (e.g., واشنطن *waAšinTun* 'Washington') or borrowed

[1] Often in the literature, radical position is indicated with *C*.

[2] Traditional accounts of Arabic morphology collapse vocalism and pattern [18]. The separation of vocalisms was introduced with the emergence of more sophisticated models [28].

terms (e.g., ديموقراطية *diymuqraATiy~aħ* ‘democracy’). NTWS can still take affixational morphemes, e.g., والواشنطنيون *waAlwaAšinTuniyuwn* ‘and the Washingtonians’. Some borrowed word stems can be forced into templatic morphology and as a result create new root and pattern combinations. For example, the word stem ديموقراطية *diymuqraATiy~aħ* ‘democracy’ has brought to existence the root دمقرط *dmqrT* (an odd 5-radical root) that is used to create the noun دمقرطة *damaqraTaħ* ‘democratization’ by combining with the already existing noun pattern *1V2V34V5aħ* and vocalism *aaa*.

2.1.4 Arabic Phonological, Morphological and Orthographic Phenomena

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

An example of a phonological adjustment rule is the voicing of the *t* of the verbal pattern *Ai1**t**V2V3* (Form VIII perfective) when the first root radical is ز ,د, or ذ (*z*, *d* or *ð*): zhr+Ai1tV2V3+aa is realized as ازدهر *Aizdahar* ‘flourish’ ***not as*** ازتهر *Aiztahar*. An example of a morphological rule is the feminine morpheme, ة *+ħ (ta marbuta)*, which can only be word final[3]. In medial position, it is turned into ت *t*. For example, كتبة+هم *katabaħu+hum* is realized as كتبتهم *katabatuhum* ‘their writers’.

Finally, an example of an orthographic rule is the deletion of the Alif (ا) of the definite article +ال *Al+* in nouns when preceded by the preposition +ل *l+* ‘to/for’ but not with any other prefixing preposition (in either case, the Alif is silent):

(2) للبيت *lilbayti* /lilbayti/ ‘to the house’

li+ Al+ bayt +i
to+ the+ house +[genitive]

(3) بالبيت *biAlbayti* /bilbayti/ ‘in the house’

bi+ Al+ bayt +i
in+ the+ house +[genitive]

[3]Only diacritics can follow a ta marbuta at the end of a word.

Another common orthographic rule that can create added ambiguity is the usage of the shadda diacritic (ّ ~), which indicates a doubled consonant. For example, in undiacritized orthography, في *fy* can be analyzed as the preposition في *fiy* 'in' or the prepositional phrase في+ي *fiy+ya* 'in me'.

2.2 Morpheme Function: Derivational vs. Inflectional

The distinction between derivational and inflectional morphology in Arabic is similar to that in other languages. Derivational morphology is concerned with creating words from other words/stems/roots where the core meaning is modified. For example, the Arabic كاتب *kaAtib* 'writer' can be seen as derived from the root كتب *ktb* the same way the English *writer* can be seen as a derivation from *write*. Although compositional aspects of derivations do exist, the derived meaning is often idiosyncratic. For example, the masculine noun مكتب *maktab* 'office/bureau/agency' and the feminine noun مكتبة *maktabaħ* 'library/bookstore' are derived from the root كتب *ktb* 'writing-related' with the pattern+vocalism *ma12a3*, which indicates location. The exact type of the location is thus idiosyncratic, and it is not clear how the gender variation can account for the semantic difference.

On the other hand, in inflectional morphology, the core meaning of the word remains intact and the extensions are always predictable. For example, the semantic relationship between كاتب *kaAtib* 'writer' and كتاب *kut~aAb* 'writers' maintains the sense of the kind of person described, but only varies the number. The change in number in this example is accomplished using templatic morphemes (pattern and vocalism change). This form of plural construction in Arabic is often called "broken plural" to distinguish it from the strictly affixational "sound plural" (e.g. كاتب+ات *kaAtib+aAt* 'writers [fem]').

A common perception of Arabic morphology is that templatic morphemes are used for deriving new lexical items (derivational morphology), while affixational morphemes are used for inflecting lexical items into word forms (inflectional morphology). However, this is not true: morpheme type and morpheme function are independent. Templatic morphemes can be derivational or inflectional, with the exception of the roots, which are always derivational. Affixational morphemes can also be derivational or inflectional. Table 1 exemplifies this relationship. Templatic morphemes distinguish between the words كتاب *kitaAb* 'book' and كتب *katab* 'wrote', both of which are derived from the root كتب *ktb* 'writing-related'. Templatic morphemes also distinguish between inflected forms of each of these

Table 1: Arabic Morpheme Type and Function

Type	**Function**	**Example**
Template	Derivation	*Root, Pattern, Vocalism* ktb+1V2V:3+ia = كتاب *kitaAb* ‘book’ ktb+1V2V3+aa = كتب *katab* ‘wrote’
	Inflection	*Pattern, Vocalism* ktb+1V2V3+uu = كتب *kutub* ‘books’ ktb+1V2V3+ui = كتب *kutib* ‘was written’
Affix	Derivation	*Suffixes* kutub+iy~ = كتبي *kutubiy~* ‘book-related’
	Inflection	*Prefixes, Suffixes,Circumfixes* Al+kitaAb = الكتاب *AlkitaAb* ‘the book’ kitaAb+ayn = كتابين *kitaAbayn* ‘two books’ y+aktub+uwna = يكتبون *yaktubuwna* ‘they write’

two words. كتب *kutub* ‘books’ is the plural form of كتاب *kitaAb* ‘book’. And كتب *kutib* ‘was written’ is the passive voice of كتب *katab* ‘wrote’. The majority of affixational morphemes are inflectional but there are some affixational derivational morphemes: the adjective كتبي *kutubiy~* ‘book-related’ is derived from the word كتب *kutub* ’books’ using the affixational morpheme ي+ *+iy~*.

2.3 Arabic Morphological Representations

Given the variability in the relationship between morpheme type and function in addition to the presence of phonological, morphological, and orthographic adjustment phenomena, there are many ways to represent Arabic words in terms of their morphological units. Table 2 illustrates some of these possible representations using the example ؟ولكتبتهم *walikatabatihim?* ‘and for their writers?’.

There are many variations among these different representations: (a.) whether they address inflectional/derivational phenomena or templatic/affixational phenomena, (b.) whether they preserve or resolve ambiguity,[4] and (c.) which degree

[4]This discussion does not address the issue of morphological disambiguation, which is outside the scope of this chapter [15, 31].

Table 2: Morphological Representations of Arabic Words

Representation	**Example**	**Found where?**
Natural Token	wlktbthm?	naturally occurring text
Simple Token	wlktbthm ?	common preprocessing for NLP
Segmentation	wl+ ktb +thm ? w+ l+ ktbt +hm ? w+ l+ ktb +t +hm ?	[7] [26]
Normalized Segmentation	w+ l+ ktbħ +hm ?	Penn Arab Treebank [27],[17]
Templatic Segmentation	w+ l+ ktb+1V2V3aħ+aa +hm ?	[21]
Morphemes and Features	w+/CONJ l+/PREP kataba*ħ* +hm/P:3MP ? ktb&CaCaCa*ħ* w+ l+ +P:3MP ? ktb +PL w+ l+ +GEN +P:3MP ?	[7], [17] [3]
Lexeme and Features	[kAtib w+ l+ PL P:3MP] [?]	ALMORGEANA dictionaries (lexeme only)

of abstraction from allomorphs (actual form of morpheme after applying various adjustment rules) they use. These variations put the various representations on a continuum of possibilities. For instance, any subset or all of the orthographic, morphological and phonological adjustment phenomena can be normalized thus creating multiple different possibilities.

The *natural token* refers to the way Arabic words appear in actual text where they are undiacritized and segmented only using white space. Punctuations, for example, could be attached to the word string in this representation. All naturally occurring Arabic text is in this representation. *Simple tokenization* separates punctuation but maintains the morphological complexity of the Arabic word tokens. There is no change in ambiguity compared to the natural token.

Segmentation is the simplest way to dissect an Arabic word. It is strictly defined here to exclude any form of orthographic, morphological or phonological normalization. Segmentation splits up the letters into segments that correspond to clusters of a stem plus one or more affixational morphemes. There are many ways to segment an Arabic word as Table 2 shows. Segmentation can select a subset of analyses of a word. For example, segmenting للجنة *lljnħ* into *l+l+jnħ* (*li+l+jan~aħ* ‘to Paradise’ or *li+l+jin~aħ* ‘to insanity/mania’) is selecting a sub-

set of analyses excluding *l+ljnħ* (*li+lajnaħ* ‘to a committee’ or *li+l~ajnaħ* ‘to the committee’).

Normalized segmentation abstracts away from some of the adjustment phenomena discussed in section 2.1.4. In the example in Table 2, the form of the segmented word stem is *ktbħ* not *ktbt*. Normalization disambiguates the unnormalized segmented form *ktbt* (‘he/she/you[sg.] wrote’ or ‘writers’). The Penn Arabic Treebank [27] uses a normalized segmentation that breaks up a word into four regions: conjunction, particle, normalized word stem and pronominal clitic.

Templatic segmentation is a deeper level of segmentation that involves normalization by definition. Here, the root, pattern and vocalism are separated. Up to this level of representation, the tokens are driven by a templatic/affixational view of morphology rather than a derivational/inflectional view. The introduction of features at the next level of representation, *morphemes and features*, abstracts away from different morphemes that at an underlying level signify the same feature. For example, The affixational morphemes ون++ي *y++uwna*, وا++ي *y++uwA* and وا+ *+uwA* all realize the third person masculine plural subject for different verb aspect/mood combinations. There are many different degrees to the transition from morphemes to features. A combination of both is often used.

The final representation is *lexeme and features*. The lexeme can be defined as an abstraction over a set of word forms differing only in inflectional morphology. The lexeme itself captures a specific derived meaning that does not change with inflectional variations. The traditional citation form of a lexeme used in dictionaries is the perfective third person masculine singular for verbs and the singular masculine form for nouns and adjectives. If there is no masculine form, the feminine singular is used. As such, the Lexeme [كاتب] [kaAtib] ‘writer’ normalizes over all the different inflectional forms of كاتب *kaAtib* such as كاتبان *kaAtibaAn* ‘two writers’, كتبة *katabaħ* ’writers’, and كاتبة *kaAtibaħ* ‘female writer’. Lexemes as opposed to stems provide a desirable level of abstraction that is to a certain degree language independent for applications such as MT. Lexemes are also less abstract than roots and patterns which tend to be too vague semantically and derivationally unpredictable, making them less useful in practice for MT.

The next section discusses how these different levels of representation interact with different MT approaches.

3 Machine Translation

3.1 Statistical MT approaches

In statistical approaches to MT, a translation model is trained on word-aligned parallel text of source and target languages [6, 5, 22, 23]. The translation model is then used to generate multiple target language hypotheses from the source language input. The target hypotheses are typically ranked using a log-linear combination of a variety of features [30]. Statistical MT has been quite successful in producing good quality MT on the genre it is trained on in much faster time than symbolic approaches. For statistical MT, in principle, it doesn't matter what level of morphological representation is used so long that the input is on the same level as the data used in training. Practically however there are certain concerns with issues such as sparsity, ambiguity, language pair, and training data size. Shallower representations such as simple tokenization tend to maintain distinctions among morphological forms that might not be relevant for translation, thus increasing the sparsity of the data. This point interacts with the MT language pair: for example, normalizing subject inflections of Arabic verbs when translating to a morphologically poor language like English might be desirable since it reduces sparsity without potentially affecting translation quality. If the target language is morphologically rich, such as French, that would not be the case. This, of course, may not be a problem when large amounts of training data are available. Additionally, transforming the training text to deeper representations comes at a cost since selecting a deeper representation involves some degree of morphological disambiguation, a task that is typically neither cheap nor foolproof [15]. There has been some work on exploring the effect of different kinds of preprocessing on statistical MT for Arabic [25, 17].

3.2 Symbolic MT approaches

In symbolic approaches to MT, such as transfer-based or interlingual MT, linguistically motivated rules (morphological, syntactic and/or semantic) are manually or semi-automatically constructed to create a system that translates the source language into the target language [13]. Symbolic MT approaches tend to capture more abstract generalities about the languages they translate between compared to statistical MT. This comes at a cost of being more complex than statistical MT, involving more human effort, and depending on already existing resources for morphological analysis and parsing. This dependence on already existing re-

sources highlights the problem of variation in morphological representations for Arabic. In a typical situation, the input/output text of an MT system is in natural or simplified tokenization. But, a statistical parser (such as [10] or [4]) trained out-of-the-box on the Penn Arabic Treebank assumes the same kind of tokenization (4-way normalized segments) used by the treebank. This means, a separate tokenizer is needed to convert input text to this representation [12, 15]. Moreover, the output of such a parser, being in normalized segmentation, will not contain morphological information such as features or lexemes that are important for translation: Arabic-English dictionaries use lexemes and proper translation of features, such as number and tense, requires access to these features in both source and target languages. As a result, additional conversion is needed to relate the normalized segmentation to the lexeme and feature levels. Of course, in principle, the treebank and parser could be modified to be at the desired level of representation (i.e. lexeme and features). But this can be a rather involved task for researchers interested in MT.

The next section describes ALMORGEANA (Arabic Lexeme-base MORphological GEnerator/ANAlyzer). ALMORGEANA is a morphological analysis and generation system built on top of the Buckwalter analyzer databases, which are at a different level of representation (3-way segmentation). Being an analysis and generation system, it can be used with MT systems analyzing or generating Arabic. ALMORGEANA relates the deepest level of representation (lexeme and features) to the shallowest (simple tokenization).[5] This wide range together with bidirectionality (analysis/generation) allows using ALMORGEANA to navigate between different levels of representations as will be discussed in section 5. Morphological disambiguation, or the selection of an analysis from a list of possible analyses, is a different task that is out of the scope of this chapter although it is quite relevant to MT [15, 31].

4 ALMORGEANA

ALMORGEANA is a large-scale lexeme-based Arabic morphological analysis and generation system.[6] ALMORGEANA uses the databases of the Buckwalter Arabic morphological analysis system with a different engine focused on generation from

[5] Going to natural tokenization is a trivial step where, for example, punctuation marks are attached to preceding words.

[6] A previous publication about ALMORGEANA focused on the generation component of the system which was named Aragen [14].

and analysis to the lexeme-and-feature level of representation. The building of ALMORGEANA didn't just involve the reversal of the Buckwalter analyzer engine, which only focuses on analysis, but also extending it and its databases to be used in a lexeme-and-feature level of representation for both analysis and generation.

The next section reviews other efforts on morphological analysis and generation in Arabic. Section 4.2 introduces the Buckwalter analyzer's database and engine. Section 4.3 describes the different components of ALMORGEANA. An evaluation of ALMORGEANA is discussed in Section 4.4.

4.1 Morphological Analysis and Generation

Arabic morphological analysis has been the focus of researchers in natural language processing for a long time. This is due to features of Arabic semitic morphology such as optional diacritization and templatic morphology. Numerous forms of morphological analyzers have been built for a wide range of application areas from information retrieval (IR) to MT in a variety of linguistic theoretical contexts [2, 1, 3, 7, 11, 21].

Arabic morphological generation, by comparison, has received little attention although the types of problems in generation can be as complex as in analysis. Finite-state transducer (FST) approaches to morphology [24] and their extensions for Arabic such as the Xerox Arabic analyzer [3] are attractive for being generative models. However, a major hurdle to their usability is that lexical and surface levels are very close [20]. Thus, generation from the lexical level is not useful to many applications such as symbolic MT where the input to a generation component is typically a lexeme with a feature list. A solution to this problem was proposed by [20] which involved composition of multiple FSTs that convert input from a deep level of representation to the lexical level. However, there are still many restrictions on the order of elements presented as input and their compatibility.[7] The MAGEAD (Morphological Analysis and Generation for Arabic and its Dialects) system attempts to design an end-to-end lexeme-and-features to surface FST-based system for Arabic [16]. As of the time of the writing of this chapter, MAGEAD's coverage is limited to verbs in Modern Standard Arabic and Levantine Arabic. The only work on Arabic morphological generation that focuses on generation issues within a lexeme-based approach is done by [9, 32]. Their work uses transformational rules to address the issue of stem change in various prefix/suffix contexts. Their system is a prototype that lacks in large-scale

[7] Other work on using FSTs designed for analysis in generation is discussed in [29].

coverage.

There are certain desiderata that are expected from a morphological analysis/generation system for any language. These include (1) coverage of the language of interest in terms of both lexical coverage (large scale) and coverage of morphological and orthographic phenomena (robustness); (2) the surface forms are mapped to/from a deep level of representation that abstracts over language-specific morphological and orthographic features; (3) full reversibility of the system so it can be used as an analyzer or a generator; (4) usability in a wide range of natural language processing applications such as MT or IR; and finally, (5) availability for the research community. These issues are essential in the design of ALMORGEANA for Arabic morphological analysis and generation. ALMORGEANA[8] is a lexeme-based system built on top of a publicly available large-scale database, Buckwalter's lexicon for morphological analysis.

4.2 Buckwalter Morphological Analyzer

The Buckwalter morphological analyzer uses a concatenative lexicon-driven approach where morphotactics and orthographic rules are built directly into the lexicon itself instead of being specified in terms of general rules that interact to realize the output [7, 8]. The system has three components: the lexicon, the compatibility tables and the analysis engine. An Arabic word is viewed as a concatenation of three regions, a prefix region, a stem region and a suffix region. The prefix and suffix regions can be null. Prefix and suffix lexicon entries cover all possible concatenations of Arabic prefixes and suffixes, respectively. For every lexicon entry, a morphological compatibility category, an English gloss and occasional part-of-speech (POS) data are specified. Stem lexicon entries are clustered around their specific lexeme, which is not used in the analysis process. Figure 1[9] shows sample entries: the first six in the left column are prefixes; the rest in that column are suffixes; the right column contains seven stems belonging to three lexemes. The stem entries also include English glosses which allows the lexicon to function as a dictionary. However, the presence of inflected forms, such as passives and plurals among these glosses makes them less usable as lexemic translations.

[8]The ALMORGEANA engine can be freely downloaded under an OpenSource license for research purposes from http://www.ccls.columbia.edu/ cadim/ resources.html. The lexical databases need to be acquired independently from the Linguistic Data Consortium (LDC) as part of the Buckwalter Arabic Morphological Analyzer [7, 8].

[9]The Buckwalter transliteration is preserved in examples of Buckwalter lexicon entries (see Chapter 2).

و/wa	Pref-Wa	*and*
ب/bi	NPref-Bi	*by/with*
وب/wabi	NPref-Bi	*and + by/with*
ال/Al	NPref-Al	*the*
بال/biAl	NPref-BiAl	*with/by + the*
وبال/wabiAl	NPref-BiAl	*and + with/by the*
ة/ap	NSuff-ap	*[fem.sg.]*
تان/atAni	NSuff-atAn	*two*
تين/atayoni	NSuff-tayn	*two*
تاه/atAhu	NSuff-atAh	*his/its two*
ات/At	NSuff-At	*[fem.pl.]*

;; 1_كَتَب /katab-u_1		
كَتَب /katab	PV	*write*
كُتُب /kotub	IV	*write*
كُتِب /kutib	PV_Pass	*be written*
كُتَب /kotab	IV_Pass_yu	*be written*
;; 1_كِتاب /kitAb_1		
كِتاب /kitAb	Ndu	*book*
كُتُب /kutub	N	*books*
;; 1_كِتابة /kitAbap_1		
كِتاب /kitAb	Nap	*writing*

Figure 1: Some Buckwalter Lexical Entries

Compatibility tables specify which morphological categories are allowed to co-occur. For example, the morphological category for the prefix conjunction و/wa *wa+* 'and', `Pref-Wa`, is compatible with all noun stem categories and perfect verb stem categories. However, `Pref-Wa` is not compatible with imperfective verb stems because they must contain a subject prefix. Similarly, the stem كتاب/`kitAb` *kitaAb* of the the lexeme 1_كتاب/`kitAb_1` *kitaAb* 'book' has the category (`Ndu`), which is not compatible with the category of the feminine marker ة/`ap` *aħ*: `NSuff-ap`. The same stem, كتاب/`kitAb` *kitaAb*, appears as one of the stems of the lexeme 1_كتابة/`kitAbap_1` *kitaAbaħ* 'writing' with a category that *requires* a suffix with the feminine marker. Cases such as these are quite common and pose a challenge to the use of stems as tokens since they add unnecessary ambiguity.

The analysis algorithm is rather simple since all of the hard decisions are coded in the lexicon and the compatibility tables: Arabic words are segmented into all possible sets of prefix, stem and suffix strings. In a valid segmentation, the three strings exist in the lexicon and are three-way compatible (prefix-stem, stem-suffix and prefix-suffix).

4.3 ALMORGEANA Components

4.3.1 Input/Output

In generation mode, the input to ALMORGEANA is a *feature-set*, a set of lexeme and features from a closed class of inflectional phenomena. The output of generation is one or more word strings in simple tokenization. In analysis mode, the input is the string and the output a set of possible feature-sets. The features in a feature-set include number, gender and case inflections, which do appear in other lan-

guages, but also prefix conjunctions and prepositions that are written as part of the word in Arabic orthography. Table 3 lists the different features and their possible values. The first column includes the names of the features. The second and third column list the possible values they can have and their definitions, respectively. The last column lists the default value assigned during generation in case a feature is unspecified based on its type. There are two types of features: obligatory and optional. Obligatory features, such as verb subject or noun number, require a value to be specified. Therefore, in case of under-specification, all possible values are generated. Optional features, such as conjunction, preposition or pronominal object/possessive clitics, on the other hand can be absent. The pronominal features, subject, object and possessive, are defined in terms of sub-features specifying person, gender and number. In case any of these sub-features is under-specified, they are expanded to all their possible values. For example, the subject feature `S:2`, as in the case of the English pronoun 'you' (which is under-specified for gender and number), is expanded to `(S:2MS S:2FS S:2D S:2MP S:2FP)`. If no POS is specified, it is automatically determined by the lexeme and/or features. For example, the presence of a definite article implies the lexeme is a noun or an adjective; whereas a verbal particle or a subject/object implies the lexeme is a verb.[10]

The following is an example of an Arabic word and its lexeme-and-feature representation in ALMORGEANA.

(4) `[kitAb_1 POS:N PL Al+ l+]`
للكتب *lilkutubi*
'for the books'

The feature-set in this example consists of the nominal lexeme `kitAb_1` 'book' with the feature `PL` '*plural*', the definite article `Al+` 'the' and the prefix preposition `l+` 'to/for'.

4.3.2 Preprocessing Buckwalter Lexicons

ALMORGEANA uses the Buckwalter lexicon described in section 4.2 *as is*. The lexicon is processed in ALMORGEANA to index entries based on inferred sets of features values (or *feature-keys*) that are used to map features in the input feature-sets to proper lexicon entries. This task is trivial for cases where the lexicon entry

[10]Other POS not included in Table 3 are D *Determiner*, C *Conjunction*, NEG *Negative particle*, NUM *Number*, AB *Abbreviation*, IJ *Interjection*, and PX *Punctuation*.

Table 3: ALMORGEANA Features

Feature	**Value**	**Definition**	**Default**
Part-of-Speech	POS:N	*Noun*	automatically determined
	POS:PN	*Proper Noun*	
	POS:V	*Verb*	
	POS:AJ	*Adjective*	
	POS:AV	*Adverb*	
	POS:PRO	*Pronoun*	
	POS:P *and others*	*Preposition*	
Conjunction	w+	*'and'*	none
	f+	*'and, so'*	
Preposition	b+	*'by, with'*	none
	k+	*'like'*	
	l+	*'for, to'*	
Verbal Particle	s+	*'will'*	none
	l+	*so as to*	
Definite Article	Al+	*the*	none
Verb Aspect	PV	*Perfective*	all
	IV	*Imperfective*	
	CV	*Imperative*	
Voice	PASS	*Passive*	all
Gender	FEM	*Feminine*	all
	MASC	*Masculine*	
Subject	S:PerGenNum	**Per***son = {1,2,3}*	all
Object	O:PerGenNum	**Gen***der = {M,F}*	none
Possessive	P:PerGenNum	**Num***ber = {S,D,P}*	none
Mood	MOOD:I	*Indicative*	all
	MOOD:S	*Subjunctive*	
	MOOD:J	*Jussive*	
Number	SG	*Singular*	all
	DU	*Dual*	
	PL	*Plural*	
Case	NOM	*Nominative*	all
	ACC	*Accusative*	
	GEN	*Genetive*	
Definiteness	INDEF	Indefinite	all
Possession	POSS	Possessed	all

provides all necessary information. For example, verb voice and aspect are always part of the stem: the feature-key for *kutib*, the stem of the passive perfective form of the verb كتب/`katab` is `katab+PV+PASS`.

Many lexicon entries, however, lack feature specifications. One example is broken plurals, which appear under their lexeme cluster, but are not marked in any way for plurality (see the entry for كتب/`kutub` in Figure 1). Detecting when a stem is plural is necessary to include the feature *plural* in the feature-key for that stem. Using the English gloss to detect the presence of a broken plural is a possible solution. However, it fails for adjectival entries since English adjectives do not inflect for plurality, e.g. كبير/`kabiyr` (SG) and كبار/`kibAr` (PL) are both glossed as '*big*'. Additionally, some sound plural stems in the lexicon are glossed as plurals. The Buckwalter categories are not helpful on their own for this task. For example, the presence of a stem with morphological category `N` is ambiguous as to being a broken plural or a singular nominalization of a form I verb [7]. The solution for this problem stems from the observation that a singular verbal nominalization is its own *lexeme*, whereas a broken plural is always listed under a lexeme that is in a singular base form. A broken plural is by definition a major change in the form of the lexeme. Therefore, if a stem under a lexeme has the morphological category `N`, `Ndip`, or `Nap` (all of which can mark a broken plural) AND it is **not** a subset string of the lexeme, it is considered a broken plural. This technique works for entries considered part of the same lexeme in the Buckwalter lexicon. Entries that treat a broken plural as a separate lexeme will not be processed correctly, e.g. the lexeme إخوة *Âixwaħ* 'brothers'.

4.3.3 Analysis and Generation

Analysis in ALMORGEANA is similar to Buckwalter's analyzer (Section 4.2). The difference lies in an extra step that uses feature-keys associated with stem, prefix and suffix to construct a feature-set for the lexeme-and-feature output. In the case of failed analysis, a back-off step is explored where prefix and suffix substrings are sought. If a compatible pair is found, the stem is used as a degenerate lexeme and the features are constructed from the feature-keys associated with the prefix and suffix.

The process of generating from feature-sets is also similar to Buckwalter analysis except that feature-keys are used instead of string sequences. First, the feature-set is expanded to include all forms of underspecified obligatory features, such as case, gender, number, etc. Next, all feature-keys in the ALMORGEANA lexicon that fully match any subset of the expanded feature-set are selected. All combina-

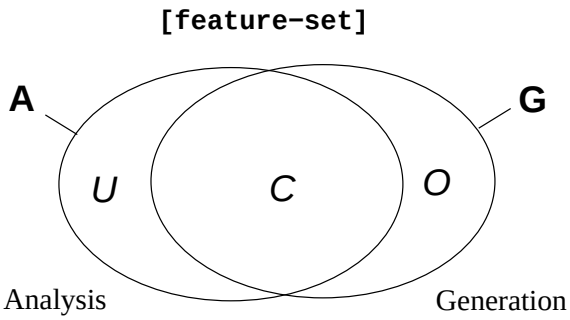

Figure 2: ALMORGEANA Evaluation

tions of feature-keys that completely cover the features in the expanded feature-set are matched up in prefix-stem-suffix triples. Then, each feature-key is converted to its corresponding prefix, stem or suffix. The same compatibility tables used in Buckwalter analysis are used to accept or reject prefix-stem-suffix triples. Finally, all unique accepted triples are concatenated and output. In the case that no surface form is found, a back-off solution that attempts to regenerate after discarding one of the input features is explored. If the back-off fails, typically due to a missing lexical entry, a baseline Arabic morphological generator is used.

The baseline generator uses a simple concatenative word structure rule and a small lexicon. The lexicon contains 70 entries that map all features to most common surface realizations. For example, FEM maps to (ة/ap *aħ*, ت/at, and ϕ) and PL maps to (ات/At, ين/iyna, ي/iy, ون/uwna and و/uw). Subtleties of feature interaction are generally ignored except for the case of subject and verb aspect since the circumfix realization of subjects in the imperfective/imperative form is rather complex to model concatenatively. The only word structure rule used in the baseline generator is the following:

```
<WORD> ::= (w|f) (s|l|b|k) Al <SubjectAspect>
           <Lexeme>
           <AspectSubject> <Gender> <Number> <Object> <Possessive>
```

4.4 Evaluation

ALMORGEANA uses the databases of the Buckwalter analyzer; therefore, its coverage is equivalent to the coverage of these lexicons. In this section, we evaluate ALMORGEANA engine for analysis and generation only[11].

[11]The evaluation described here was run over the Buckwalter lexicons (version 1) [7].

Table 4: Evaluation Results

System	UnderErr	OverErr	CombErr	Time (secs)
ALMORGEANA *diacritized*	0.39%	12.22%	0.76%	1,769
ALMORGEANA *undiacritized*	0.38%	12.42%	0.74%	1,745
Baseline *diacritized*	43.90%	60.99%	51.05%	281
Baseline *undiacritized*	32.84%	47.93 %	38.98%	293

A sample text of over one million Arabic words from the UN Arabic-English corpus [19] was used in this evaluation. For each unique word in the text, ALMORGEANA is used in analysis mode to produce feature-sets. The resulting feature-sets are then input to two systems: the complete ALMORGEANA as described earlier *and* the baseline generator used as back-off to ALMORGEANA generation. For each feature-set, there are two sets of words: (a) words that analyze into the feature-set (A words) and (b) words that are generated from the feature-set (G words) (see Figure 2). The bigger the intersection between the two sets (C words), the better the performance of a system. Generated words that are not part of the intersection (C words) are Overgenerated words (O words). Words that analyze into the feature-set but are not generated are Undergenerated words (U words). In principle, U words are definite signs of problems in the generation system; whereas, O words can be correct but unseen in the analyzed text.

A system's Undergeneration Error (UnderErr) is defined as the ratio of U words to A words. Overgeneration Error (OverErr) is defined as the ratio of O words to G words. These two measure are similar to (1 - Precision) and (1- Recall) respectively, if the set of A words paired with a feature-set are considered a gold standard to be replicated in reverse by a generation system. The Combined Undergeneration and Overgeneration Error (CombErr) is calculated as their harmonic mean (in a manner similar to calculating the F-scores for Precision and Recall)

$$\text{UnderErr} = \frac{U}{A} = \frac{A-C}{A} \qquad \text{OverErr} = \frac{O}{G} = \frac{G-C}{G} \qquad \text{CombErr} = \frac{2\times UnderErr\times OverErr}{UnderErr+OverErr}$$

The evaluation text contained 63,066 undiacritized unique words, which were analyzed into 118,835 unique feature-sets corresponding to 14,883 unique lexemes. The number of unique diacritized words corresponding to the text words is 104,117. The evaluation was run in two modes controlling for the type of matching between A words and G words: diacritized (or diacritization-sensitive) and undiacritized. Evaluation results comparing ALMORGEANA to the baseline are

presented in Table 4. The baseline system is almost six times faster than ALMORGEANA[12], but it had high undergeneration and overgeneration error rates. Both were reduced in the undiacritized mode, where some erroneous output became ambiguous with correct output. ALMORGEANA, by comparison, reduced the error rate from the baseline by more than one order of magnitude overall.

Many of the overgeneration errors are false alarms. They include cases of overgeneration of broken plurals, some of which are archaic or genre-specific but correct. For example, the word for 'sheik', شيخ *šayx*, has three uncommon broken plurals in addition to the common شيوخ *šuyuwx*: اشياخ *ÂšyaAx*, مشايخ *mašaAyix*, and مشائخ *mašaAŷix*. Another very common overgeneration error resulted from the underspecification of some mood-specific vocalic verbal suffixes in the Buckwalter lexicon. Arabic hollow verbs, for example, undergo a stem change in the jussive mood (from يقول *yaquwl* to يقل *yaqul*), which is indistinguishable in the analysis.

Undergeneration errors stem exclusively from lexicon errors. These are not many and they can be expected in a manually created database. One example is caused by a missing lexeme comment in the Buckwalter lexicon which resulted in pairing all the forms of the verb رأى *raÂaý* 'to see' to the lexeme that appears just before it, راوند *raAwand* 'rhubarb'. Such cases suggest a valuable use of ALMORGEANA as a debugging tool for the Buckwalter lexicon.[13]

5 Interoperability of Morphological Representations

This section describes how ALMORGEANA can be used to navigate between different levels of morphological representation. An Arabic word in simple tokenization can be analyzed using ALMORGEANA to multiple possible lexeme-and-feature analyses. This automatically gives us access to the lexeme-and-feature level and also the three-way segmentation used by Buckwalter's lexicons. To generate an intermediate representation such as the normalized segmentation used by the Penn Arabic Treebank [27], the features for conjunction, preposition and pronominal object/possessive can be stripped from the lexeme-and-feature analyses. The remaining features and lexeme are then used to generate the word stem using ALMORGEANA to guarantee a normalized form. The stripped features are also trivially generated and positioned relative to the word stem: [conjunction] [preposition] [word-stem] [pronoun]. Table 5 shows the different analyses for

[12]The experiments were run on a Dell Inspiron machine with Pentium 4 CPU and 2.66 GHz.

[13]All of the errors described here are for version 1 of the Buckwalter analyzer only [7]. We did not conduct a similar study on version 2 of the Buckwalter analyzer [8].

Table 5: Normalized Segmentation Example

word	Analysis	Segments
wqd	[qad~_1 POS:N w+ +SG +MASC gloss:size/physique] [qad_2 POS:F w+ gloss:may/might] [qad_1 POS:F w+ gloss:has/have] [qid~_1 POS:N w+ +SG +MASC gloss:thong/strap]	w qd
	[waq~ad_1 POS:V +PV +S:3MS gloss:kindle/ignite] [waqod_1 POS:N +SG +MASC gloss:fuel/burning] [waqadi_1 POS:V +PV +S:3MS gloss:ignite/burn]	wqd
kAtbth	[kAtib_1 POS:N +FEM +SG +P:3MS gloss:author/writer/clerk] [kAtib_2 POS:AJ +FEM +SG +P:3MS gloss:writing]	kAtbħ h
	[kAtab_1 POS:V +PV +S:3FS +O:3MS gloss:correspond_with] [kAtab_1 POS:V +PV +S:1S +O:3MS gloss:correspond_with] [kAtab_1 POS:V +PV +S:2FS +O:3MS gloss:correspond_with] [kAtab_1 POS:V +PV +S:2MS +O:3MS gloss:correspond_with]	kAtbt h
ftHyħ	[taHiy~ap_1 POS:N +FEM +SG f+ gloss:greeting/salute]	f tHyħ
	[fatHiy~ap_1 POS:PN gloss:Fathia]	ftHyħ
lmdħ	[mud~ap_1 POS:N +FEM +SG l+ gloss:interval/period]	l mdħ
sntyn	[sinot_1 POS:N +MASC +DU +ACCGEN gloss:cent] [sanap_1 POS:N +FEM +DU +ACCGEN gloss:year]	sntyn
.	[. POS:PX gloss:.]	.

each word in the sentence . وقد كاتبته فتحية لمدة سنتين *wqd kAtbth ftHyħ lmdħ sntyn. 'and Fathia continued to correspond with him for two years'.* The correct Penn Arabic Treebank tokenization for this example is *w qd kAtbt h ftHyħ l mdħ sntyn.*

The ambiguity inherent in both the analysis and generation processes results in multiple possibilities (column 3 in Table 5). To select a specific segmentation, any of a set of possible techniques can be used such as rule-based heuristics or language models trained on text in the correct tokenization. For example, in the case of the Penn Arabic Treebank, the already tokenized text of the treebank can be used to build a language model for ranking/selecting among options produced by this technique (similar to [26]). Alternatively, machine learning over the features of the annotated words in the Penn Arabic Treebank can be used to select among the different analyses (similar to [15, 31, 17]).[14]

[14]The Morphological Analysis and Disambiguation for Arabic (MADA) tool [15] is a disambiguation system fully integrated with ALMORGEANA. More information on MADA is available

5.1 TOKAN

The general tokenization tool, TOKAN, is an implementation of this disambiguate-and-regenerate approach to tokenization. TOKAN is built on top of ALMORGEANA. TOKAN takes as input (a.) disambiguated ALMORGEANA analyses and (b.) a token definition sequence that specifies which features are to be extracted from the word and where they should be placed. For example, the token definition for splitting off the conjunction *w+* only is `"w+ REST"`. This token definition specifies that the conjunction *w+* is split from the word and whatever is left (REST) is regenerated after the conjunction *w+*. Similarly, the token definition for the Penn Arab Treebank tokenization is `"w+ f+ l+ k+ b+ REST +O: +P:"`.[15]

6 Conclusions

This chapter described obstacles present for MT researchers when using Arabic resources in differing morphological representations. The lexeme-and-feature level of representation has been motivated and, ALMORGEANA, a large-scale system for analysis and generation from/to that level has been described and evaluated. The wide-range coverage in terms of representations and its bidirectionality makes ALMORGEANA a desirable tool for relating different resources available to the MT researcher/developer working with Arabic as a source or target language.

Acknowledgments

This work has been supported, in part, by Army Research Lab Cooperative Agreement DAAD190320020, NSF CISE Research Infrastructure Award EIA0130422, Office of Naval Research MURI Contract FCPO.810548265, NSF Award #0329163 and Defense Advanced Research Projects Agency Contract No. HR0011-06-C-0023. I would like to thank Owen Rambow, Mona Diab, Bonnie Dorr, Tim Buckwalter and Michael Subotin for helpful discussions.

at http://www.ccls.columbia.edu/ cadim/ resources.html.

[15] More information on TOKAN is available at http://www.ccls.columbia.edu/ cadim/ resources.html.

References

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

[2] M. Aljlayl and O. Frieder. On arabic search: Improving the retrieval effectiveness via a light stemming approach. In *Proceedings ofACM Eleventh Conference on Information and Knowledge Management, Mclean, VA*, 2002.

[3] K. Beesley. Arabic finite-state morphological analysis and generation. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, pages volume 1, 89–94, Copenhagen, Denmark, 1996.

[4] Daniel M. Bikel. Design of a multi-lingual, parallel-processing statistical parsing engine. In *Proceedings of International Conference on Human Language Technology Research (HLT)*, 2002.

[5] P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer. The Mathematics of Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, 1993.

[6] Peter F. Brown, John Cocke, Stephen Della-Pietra, Vincent J. Della-Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16:79–85, June 1990.

[7] Tim Buckwalter. Buckwalter Arabic Morphological Analyzer Version 1.0, 2002. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Catalog No.: LDC2002L49.

[8] Tim Buckwalter. Buckwalter Arabic Morphological Analyzer Version 2.0, 2004. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Cat alog No.: LDC2004L02.

[9] Violetta Cavalli-Sforza, Abdelhadi Soudi, and Teruko Mitamura. Arabic Morphology Generation Using a Concatenative Strategy. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP 2000)*, pages 86–93, Seattle, Washington, USA, 2000.

[10] Michael Collins. Three Generative, Lexicalised Models for Statistical Parsing. In *Proceedings of the 35th Annual Meeting of the ACL (jointly with the 8th Conference of the EACL)*, Madrid, Spain, 1997.

[11] Kareem Darwish. Building a Shallow Morphological Analyzer in One Day. In *Proceedings of the workshop on Computational Approaches to Semitic Languages in the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, Philadelphia, PA, USA, 2002.

[12] Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. Automatic tagging of arabic text: From raw text to base phrase chunks. In *Proceedings of the 5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL04)*, Boston, MA, 2004.

[13] Bonnie J. Dorr, Pamela W. Jordan, and John W. Benoit. A Survey of Current Research in Machine Translation. In M. Zelkowitz, editor, *Advances in Computers, Vol. 49*, pages 1–68. Academic Press, London, 1999.

[14] Nizar Habash. Large scale lexeme based arabic morphological generation. In *Proceedings of Traitement Automatique des Langues Naturelles (TALN-04)*, 2004. Fez, Morocco.

[15] Nizar Habash and Owen Rambow. Tokenization, Morphological Analysis, and Part-of-Speech Tagging for Arabic in One Fell Swoop. In *Proceedings of the 43rd Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, 2005.

[16] Nizar Habash, Owen Rambow, and George Kiraz. Morphological Analysis and Generation for Arabic Dialects. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages at 43rd Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, 2005.

[17] Nizar Habash and Fatiha 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)*, Brooklyn, New York, 2006.

[18] Z. Harris. Linguistic structure of Hebrew. *Journal of the American Oriental Society*, 62:143–67, 1941.

[19] Xu Jinxi. UN Parallel Text (Arabic-English), LDC Catalog No.: LDC2002E15, 2002. Linguistic Data Consortium, University of Pennsylvania.

[20] L. Karttunen, R. Kaplan, and A. Zaenen. Two-level morphology with composition. In *Proceedings of Fourteenth International Conference on Computational Linguistics (COLING-92)*, pages 141–148, Nantes, France, July 20–28 1992.

[21] George Kiraz. Multi-tape Two-level Morphology: A Case study in Semitic Non-Linear Morphology. In *Proceedings of Fifteenth International Conference on Computational Linguistics (COLING-94)*, pages 180–186, Kyoto, Japan, 1994.

[22] Kevin Knight. A Statistical MT Tutorial Workbook, 1999. http://www.clsp.jhu.edu/ws99/projects/mt/mt-workbook.htm.

[23] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Edomonton, Canada, May 27-June 1 2003.

[24] K. Koskenniemi. Two-Level Model for Morphological Analysis. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 683–685, 1983.

[25] Young-Suk Lee. Morphological Analysis for Statistical Machine Translation. In *Proceedings of the 5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL04)*, Boston, MA, 2004.

[26] Young-Suk Lee, Kishore Papineni, Salim Roukos, Ossama Emam, and Hany Hassan. Language Model Based Arabic Word Segmentation. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL'03)*, Sapporo, Japan, 2003.

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

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

[29] Guido Minnen, John Carroll, and Darren Pearce. Robust, Applied Morphological Generation. In *Proceedings of the 1st International Conference on Natural Language Generation (INLG 2000)*, Mitzpe Ramon, Israel, 2000.

[30] Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. A Smorgasbord of Features for Statistical Machine Translation. In *Proceedings of the Human Language Technology / North American Association of Computational Linguistics Conference*, Boston, Massachusetts, May 2004.

[31] Noah Smith, David Smith, and Roy Tromble. Context-Based Morphological Disambiguation with Random Fields. In *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP05)*, Vancouver, Canada, 2005.

[32] A. Soudi, V. Cavalli-Sforza, and A. Jamari. A Computational Lexeme-Based Treatment of Arabic Morphology. In *Proceedings of the Arabic Natural Language Processing Workshop, Conference of the Association for Computational Linguistics (ACL 2001)*, Toulouse, France, 2001.