A Unification Based Approach to the Morphological Analysis and Generation of Arabic

Selçuk Köprü
AppTek, Inc.
METU Technopolis
06531, Ankara, TR
skopru@apptek.com

Jude Miller
AppTek, Inc.
6867 Elm Street
McLean, 22101, VA, USA
jude@apptek.com

Abstract

In this paper, we present a powerful Arabic morphological analyzer and generator. The approach employs finite state machines enriched with unification capability. The presented system is used as a component in both statistical and rule based machine translation systems. We give detailed illustrations on how we handle nominal and verbal morphology in Arabic. Issues regarding derivational morphology and morphological generation are also addressed. Stimulating problems particular to Arabic and our solutions to these problems are explored meanwhile. An evaluation of the system is presented at the end.

1 Introduction

The continuous increase in the demand for processing Arabic faces many challenges. One important challenge at this point is the requirement to analyze Arabic morphology with high quality. Morphological analysis and generation systems can be used in many Natural Language Processing (NLP) applications, such as Machine Translation (MT) or Information Retrieval (IR). The success of the morphological analysis component plays an important role in the overall quality of the entire system.

In any MT system, whether it is a statistical machine translation (SMT) system or a rule-based machine translation (RBMT) system, the morphological analyzer is an indispensable component. In SMT systems, training data is annotated with morphological information for the purpose of factored translation models. Koehn and Hoang (2007) have shown that factored translation models containing morphological information lead to better translation performance. Morphological analysis and generation becomes more important when translating to or from morphologically rich languages such as Arabic.

In RBMT systems, morphological processing plays a key role in overall analysis and generation. Morphological analysis is the first step before syntactic analysis. In this paper, we present a standalone morphological analyzer and generator for the Modern Standard Arabic (MSA). The same system is used both in RBMT and SMT successfully. Scientific contribution of this work is that it presents an attempt to improve the results of state-of-the-art Arabic morphological analyzers employing a unification based finite-state approach. The presented work covers the whole Arabic morphology by giving fragments of morphological processing related to main lexical category groups.

Arabic, as other Semitic languages, has a non-concatenative morphology which requires more complex morphotactic transformations compared to concatenative morphology systems. Morpheme characters can be discontinuous so that they are interleaved among the root characters. The most stimulating problem in Arabic Morphology is the excess number of different transformation patterns for specific morphemes (e.g. number morpheme in nominals).

The rest of this paper is organized as follows. In the next section we give a brief overview of related work. In section 3, the approach and the system that is used in this study is introduced. Next, the implementation details of Arabic morphological analysis and generation are presented. In section 5, we explain the experiment conducted for the evaluation of the system. Finally, we conclude the paper.

2 Related Work

In the past decade, various morphological analyzers for Arabic have appeared from different academic and commercial sources. In Al-Sughaiyer and Al-Kharashi (2004), a detailed summary is given about published literature work. Al-Sughaiyer and Al-Kharashi classify Arabic morphological analyzers according to the employed approach: Finite State Machine (FSM), two-level morphology or a pattern based approach. As pointed out in the same survey, very few of the available systems are evaluated using systematic and scientific procedures.

Beesley opposes the belief that finite-state power is insufficient to handle Semitic morphology by illustrating an Arabic morphological analyzer using only finite-state operations (Beesley, 1998). In Cavalli-Sforza et al. (2000), the authors claim to handle Arabic morphology using a concatenative strategy by separating infixal variations from the other ones. They separated concatenative and infixal operations into two steps because of practical concerns. A semi-statistical approach which learns from word-root pairs complemented with affix lists and transformation templates is presented in Darwish (2002). In Habash et al. (2005) and Habash and Rambow (2006), a morphological analyzer both for MSA and spoken dialects are presented. It is evaluated and compared to the Buckwalter analyzer (Buckwalter, 2004).

A recent and comprehensive book about the processing of Arabic morphology is Soudi et al. (2007). The editors categorized the computational approaches into two main paradigms: knowledge-based and empirical approaches. The collection of works included in the book is grouped according to these two main approaches. Studies highlighting the integration of Arabic morphology in larger applications are also included in the same book.

In Cahill (2007), a Syllable-Based Morphology (SBM) approach is applied to Arabic verbs. Cahill concludes that SBM analysis is possible for Semitic languages and it is not significantly different from European language analyses such as English and German. Cavalli-Sforza and Soudi present an approach to Arabic morphology that draws from Lexeme-Based Morphology (LBM) in (Cavalli-Sforza and Soudi, 2007).

Our work is novel in the respect that a complete full-fledged morphological analyzer and a generator is described thoroughly.

3 The System

A finite state transducer (FST) augmented with unification based feature structures (FS) stays in the heart of the system. The FST is responsible for both analysis and generation tasks. The FST is not visible to the grammar developer who writes the morphological analysis and generation rules in a high-level formalism. Manually crafted rules are later compiled into finite state machines. The main difference between two-level systems and our approach lies in the definition and organization of fundamental information. In two-level morphology rules, the finite state machine reflects the relation between lexical forms and surface forms. In our approach, the fundamental unit of definition is the morpheme category which is the abstraction of morphological operations.

```
Rule Definition ::
  stem
  ( mc1 )
  { mc2 mc3 }
  mc4 *
  < mc5 mc6 >
->
  word-cat
```

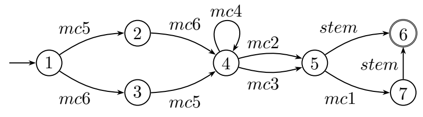


Figure 1: A sample morphology rule definition and the corresponding FSM.

Morphological rules are used to derive the uninflected form of an inflected word during analysis. The rule definition is composed of morpheme categories bundled with regular expression formalism. A sample rule definition and the corresponding FSM is shown in Figure 1. Each morpheme category in the rule definition corresponds to one or more arcs in the compiled FSM. In the formalism, parentheses represent optionality; asterisk and plus are the signs for *kleene* star and *kleene* plus, respectively. Disjunction is represented by curly brackets and morpheme categories listed inside angle brackets represent free ordering. The regular expression formalism is recursive: disjunctive or optional elements can themselves contain disjunctive or optional elements.

The sample rule in Figure 1 declares that a new category '`word-cat`' is built if all essential elements in the definition are matched successfully. It is important to notice that the ordering doesn't imply concatenative morphotactics, instead, it refers to the order of the application of the morpheme operations (which might or might not be concatenative) on the word.

The exploited formalism allows declaring and formulating linguistic knowledge in a concise manner. The primary data structure to represent the features and values is a directed acyclic graph (dag). The system also includes an expressive Boolean formalism, used to represent functional equations to access, inspect or modify features or feature sets in the dag. Complex feature structures (e.g. lists, sets, strings, and conglomerate lists) can be associated with lexical entries and grammatical categories using inheritance operations. Unification is used as the fundamental mechanism to integrate information from lexical entries into larger grammatical constituents. The same framework has been used successfully for concatenative as well as non-concatenative languages. English, Korean, Turkish, and Arabic are a few of the many languages implemented within the same system.

A morpheme category appearing in the rule is defined with a FS and an allomorph table. The FS contains the lexical information and the allomorph table contains the set of string operations. A morpheme is matched if at least one of the string operations in the allomorph table succeeds. Moreover, the FSs of all matching morphemes and the FS that comes from the lexical stem should unify. A morpheme with a non-unifying FS is not matched. A sample allomorph table is shown below.

Successful match of a morpheme indicates that at least one of the string operations in the corresponding allomorph table is performed successfully. String operators are unary functions that allow prefixal, suffixal, circumfixal and infixal modification of the word. String modifications applied during analysis and generation employ exactly reverse operations (cut vs. add, geminate vs. de-geminate

```
mCOMP-SUPER ::
   [   -ا أ  - أ     %+ ائ   ] COMPAR-HAZ'
   [   -ا أ  - أ     + يء    ] COMPAR-BTY'
   [   -ا أ  - ى     %+ اي   ] COMPAR-RAQY
   [   -ا أ  - ى     + ي     ] COMPAR-WFY
   [   -ا أ  %- أ.   %+ ائ.  ] COMPAR-YA'S
   [   -ا أ  %- و.   %+ ائ.  ] COMPAR-RA'J
   ...
```

Figure 2: A sample allomorph table containing morphological operations for the Arabic comperative.

etc.) In analysis, feature structures associated with the matching arc are unified in order to build the final feature structure.

During analysis, processing starts from the last morpheme category in the rule definition towards the stem. The morphemes used in the rule definition are associated with allomorph tables and typed feature structures. Each row in the allomorph table contains an allomorph described by a set of morphological string operations as in Figure 2.

4 Implementation

In this part, we explore the essential parts of the implementation. First, we look into the details of the lexicon. Next, we examine verbal and nominal categories which constitute the significant part of Arabic morphology. A small fragment of derivational morphology is also shown. Finally, we study the generation process.

There exists only one morphology rule for a group of lexical categories. For example, nouns, adjectives, quantifiers and pronouns are handled in the same rule. Each rule contains the related morpheme categories and the associated allomorph table - feature structure pairs.

4.1 Lexicon

As in any morphological analyzer, the quality is directly related with the enhancement level of the lexicon. There are two aspects that contribute to this enhancement level. First is the number of lexical entries contained in the lexicon. Second is the richness in linguistic information contained by the lexical entries.

To conform to these quality requirements, a lexicon of 75K entries is populated covering all lexical categories. Each entry contains feature value pairs compulsory for morphological and syntactic analysis and generation. Verbs and nouns in the lexicon contain appropriate inflection patterns besides other linguistic data. A word with multiple part-of-speech contains a list of FS's. A sample lexical entry with minimal information is given below:

$$\text{رقم}\left[\begin{bmatrix} \text{CAT} & \textit{vstem} \\ \text{PRED} & \#\,\textit{obj}\,\# \\ \text{VPATTERN} & \textit{f'l} \\ \text{FORM} & \text{رقم} \\ \ldots & \end{bmatrix}, \begin{bmatrix} \text{CAT} & \textit{nstem} \\ \text{NPATTERN} & \textit{rqm} \\ \text{GENDER} & \textit{msc} \\ \text{FORM} & \text{رقم} \\ \text{NUMBER} & \textit{sg} \\ \ldots & \end{bmatrix}\right]$$

In Buckwalter (2007), it is differentiated between two types of morphological analysis lexicons: one with entries based on root and pattern morphemes and another whit entries that make use of word stems. Our lexicon conforms more to the second approach because word stems constitute the key for the entries. However, pattern information used in analysis and generation is also included in the same dictionary.

The stem-grounded lexicon in our implementation allows us to associate entries with grammar rules and linguistic features. In Dichy and Farghaly (2007), it is claimed that this approach is the most appropriate organization for the storage of pertinent information. Compared to the lexicon used in Buckwalter (2004) analyzer, our entries contain a larger set of grammatical features. The rational motive for this design decision was because the lexicon is used for different tasks besides morphological processing (e.g. syntactic analysis in MT or entity detection).

In general, it is aimed to exclude any kind of duplicate information in the lexicon. For example, feminine nouns do not exist in the lexicon as separate entries if they are not intrinsically feminine. معلمة '*teacher*-FEM' is not in the lexicon as it can be deduced from معلم '*teacher*'. The entry under the key معلم '*teacher*' does not contain any gender information in the lexicon because it is completed at the end of morphological analysis.

4.2 Verbal Categories

We have classified verbs according to their syntactic characteristics into the following groups: Ordinary verbs, auxiliaries, modals and inna-kana verbs. All of these verbs are handled with the same morphology rule. Ordinary verbs in Arabic inflect for tense (perfective: PERF, imperfective: IMPERF), number (singular: SG, dual: DL, plural: PL), person (1, 2, 3), gender (masculine: MSC, feminine: FEM), mood (indicative: IND, subjunctive: SUB, imperative: IMP, jussive: JUS), voice (active: ACT, passive: PAS) and modality (future: FUT). Personal pronouns can be attached to the end as suffixes. Additionally, there exist coordination and subordination prefixes attached to verbs: و /wa/ '*and*', ف /fa/ '*then*', and ل /la/ '*so*'. Verbal stems in the lexicon are 3-SG-MSC-PERF.

Figure 3 depicts the simplified verb rule and the FS associated to the morpheme categories. Final FS, which is built via unification at the end of analysis, is associated with VERB syntactic category. The FS belonging to the matching morpheme contributes to the final FS. How these feature structures are combined is defined with projection operators and functional roles. In the example, the FS belonging to the attached pronoun morpheme is projected as OBJ functional role in the final FS. Other components are combined with ordinary unification mechanism. After all morphemes and the lexical stem are matched successfully, a new complete edge in the chart is created with VERB category. The FS can be further modified with functional expressions. The caret symbol (^) is a path operator and points to the newly built FS.

Table 1 lists the processing steps for ستسمعونهم /satasomaEonahum/ '*you will hear them*' during analysis according to the rule illustrated in Figure 3. 'String' column in Table 1 shows the resulting string at the end of each step. '−' and '−|' operators cut the specified argument from the end and from the beginning of the input string, respectively. Processing starts with prefix morpheme category MPREFIX and س /sa/ is cut from the beginning of the input

```
verb-rule ::
  vstem
  ( mpngt )
  ( mpron : OBJ )
  ( mprefix )
->
  verb
  &&
  ^ = [ mood indic
        voice active ]
```

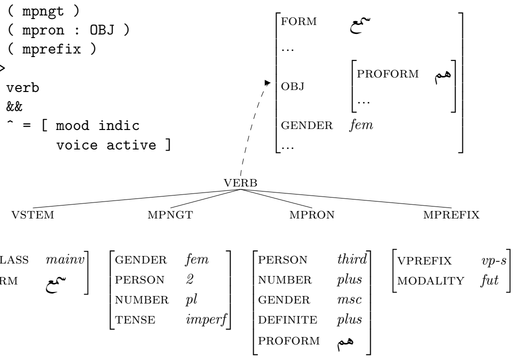


Figure 3: Verb morphology rule and the unifying feature structures.

string. FS of MPREFIX contains the VPREFIX label and a value. So, the VPREFIX label is present also in the final FS. Next, the attached pronoun suffix هم is stripped from the end of the word. Since the pronoun is defined to be projected as OBJ in the rule, the associated FS is put under OBJ label in the final FS. At step 3, morpheme category MPNGT, representing person-number-gender-tense information, is cut from the end. Finally, the remaining string سمع is matched successfully in the lexicon with the VSTEM lexical category. In order to conclude the analysis with success, the unification of feature structures associated to morphemes should also succeed. Finally, the final FS is built and MOOD and VOICE labels are inserted using additional functional expressions.

	Category	Explanation	Operation	Result
1	MPREFIX	FUT	−| س	تسمعنهم
2	MPRON	*them*	− هم	تسمعن
3	MPNGT	2-PL-FEM -IMPERF	− ن −| ت	سمع
4	VSTEM	*do*	*match*	سمع

Table 1: Processing steps for ستسمعنهم according to the rule in Figure 3.

The strong verb in Table 1 belongs to a regular paradigm because it doesn't require any additional operations once the standard suffixes and prefixes are stripped from the stem. Table 2 illustrates the processing steps of ليؤويانهم /liyuw>owiyAnahum/ '*so they shelter them*'. This weak verb requires additional operations, which we call defective operations, after standard affix stripping in order to match the verb stem in the lexicon. Row 4 in the new table shows the defective operations that are performed inside the string. The MDEFECT morpheme category is inserted as an optional category in the verb morphology rule right before MPNGT.

	Category	Explanation	Operation	Result
1	MPREFIX	*so*	−| ل	يؤويانهم
2	MPRON	*them*	− هم	يؤويان
3	MPNGT	3-DL-MSC -IMPERF	−| ي − ان	ؤوي
4	MDEFECT	*defective*	− ؤ.ي	.و.
		operations	+ آ.ى	آوى
5	VSTEM	*shelter*	*match*	آوى

Table 2: Processing steps for ليؤويانهم with defective operations.

In our lexicon, there are about 80 different inflection patterns for verbs. Each of these patterns requires a different string operation. The strong verb pattern KTB does not require any defective operation in any MPNGT inflection. About 10% of the verbs in our lexicon belong to this paradigm. Some paradigms require a defective operation in only one inflection instance. For instance, verbs belonging to the FAAL paradigm (e.g. طالب /tAlaba/ '*request*') do not demand any additional operations as KTB verbs except in passive voice. There exist some patterns which have only one verb instance belonging to it. For example, the ASTAA paradigm is created specifically to handle one verb, استاء /AstA'a/ '*resent*'. This means, no other verb does undergo the same operations in all MPNGT transformations as استاء /AstA'a/. In the lexicon, the appropriate pattern for each verb is assigned manually.

4.3 Nominal Categories

Nouns, adjectives, pronouns (relative, personal and demonstrative), and quantifiers fall into this category. In general, nouns in Arabic inflect for number (singular: SG, dual: DL, plural: PL), definiteness (definite: DEF, indefinite: INDEF), case (nominative: NOM, accusative: ACC, genitive: GEN), person (1, 2, 3), gender (masculine: MSC, feminine: FEM). Simi-

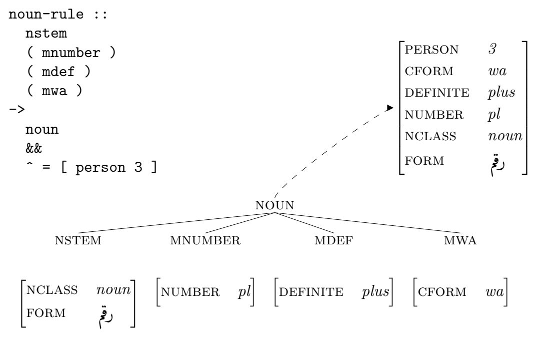
	Category	Explanation	Operation	Result
1	MWA	AND	–| و	الأرقام
2	MDEF	*the*	–| ال	أرقام
3	MNUMBER	PL	–| أ – ا	رقم
4	NSTEM	*number*	*match*	رقم

Table 3: Processing steps for والأرقام /wAlAroqam/ '*and the numbers*'.

lar to verbs, personal pronouns can be attached to the end as suffixes but with a different functional role this time. و /wa/ '*and*' and ف /fa/ '*then*' can be attached to nouns as a prefix.

The simplified noun rule and the feature structures associated with morpheme categories are illustrated in Figure 4 for the word والأرقام /wAlAroqam/ '*and the numbers*'. All morpheme categories listed in the rule are optional, only the stem is compulsory. So, the absence of any of the morpheme categories does not make the rule fail. Table 3 lists the processing steps for the same input. After removing the attached coordinator و /wa/ '*and*' and the definite marker ال /Al/ '*the*', at step 3, an infixal operation is executed by the MNUMBER morpheme.

There are about 350 different patterns for broken plural forms of nouns, each one requiring a different string modification in the MNUMBER morpheme. Similar to verbs, the patterns are inserted manually in the lexicon.



```
noun-rule ::
  nstem
  ( mnumber )
  ( mdef )
  ( mwa )
->
  noun
  &&
  ^ = [ person 3 ]
```

Figure 4: Noun morphology rule and the unifying feature structures.

One important point that needs further clarification is the question on how the same rule can be used for different lexical categories. For example, pronouns do not undergo number inflection, but they are handled in the same rule with nouns. The solution comes from the functional expression formalism. Each category element in the rule definition can be assigned a conditional expression. Figure 5 depicts the sample solution to this problem. The expression assigned to NSTEM will prevent a pronoun to be analyzed with MNUMBER morpheme.

```
noun-rule ::
  nstem ? if [ ^ nclass ] = pronoun then
              no mnumber
            end if
  ( mnumber )
  ( mdef )
  ( mwa )
->
  noun
  ...
```

Figure 5: Functional expression formalism.

4.4 Derivational Morphology

Derivational morphology is usually discarded in morphological analysis but there are certain advantages in syntactic analysis if it is handled properly. For example, gerunds undergo nominal transformation in morphology but the constituent structure they enforce in the sentence is the same as their verbal stem. Locating the verb is important in order to make use of this constituent information which is defined by the PRED label. Putting the gerund into the lexicon and duplicating this information is an alternative approach but it causes additional lexicon work.

```
gerund-rule ::
  vstem
  ( mderive )
  ( mnumber )
  ( mdef )
->
  verb
```

Figure 6: Simplified gerund rule.

Besides utilizing the noun-stem analysis, we also capture the verb-root-derivational forms in Arabic

	Category	Explanation	Operation	Result
1	MWA	AND	$-\vert$ و	الأرقام
2	MDEF	*the*	$-\vert$ ال	أرقام
3	MNUMBER	PL	$-\vert$ أ $-$ ا	رقم
4	NSTEM	*number*	*match*	رقم

Table 4: Processing steps for الإمضاء /Al<imDAY/ '*spending*'.

nouns. For example, الإمضاء /Al<imDAY/ means literally '*signature*', but it can also be the gerund form of the verb '*spend*'. It is important that all possible analyses are produced in the word context. The right analysis can be picked in a wider context.

Figure 6 illustrates the simplified gerund derivation rule and the feature structures. Table 4 lists the processing steps for الإمضاء /Al<imDAY/ '*spending*'. Noun morphology transformations are applied to the input gerund. At step 2, the verbal stem is found by applying the derivational transformation. There are about 30 different gerund derivation patterns that can be applied to verbs. The appropriate derivation pattern is specified in the verb lexical entry using the DPATTERN label. As it can be seen in Figure 6, final FS that is built after analysis contains the constituent information for the gerund under the PRED feature.

4.5 Generation

The aim in morphological generation is to produce the inflected form of a word according to the features and values in the FS. It is important to reuse the linguistic resources created for analysis purpose. From a practical point of view, morphological generation is the inverted form of analysis. Same analysis rule definition can be used to generate the desired word form. The only difference will be the direction of execution order of the elements in the rule definition. This can be achieved by inverting the corresponding FSM. Starting state and final states have to be switched and the directions of the arcs have to be changed. Another modification that has to be performed is in the string manipulation operations. Reverse actions have to be carried out in inverted order during generation. Table 5 lists the reverted and inverted operations to build ليؤويانهم /liyuw>owiyAnahum/ '*so they shelter them*'. Notice the differences in this listing compared to Table 2. In this case, processing starts with the stem and the lexicon is looked up to load necessary linguistic data, e.g. PATTERN, into the FS. Morpheme operations are applied in the reverse order in contrast to analysis. Cut operation ($-$ or $-|$) in analysis is replaced with add operation ($+$ or $+|$) and vice versa.

	Category	Explanation	Operation	Result
1	VSTEM	*shelter*	*match*	آوى
2	MDEFECT	*defective*	$-$ آ.ى	.و.
		operations	$+$ ؤ.ي	ؤوي
3	MPNGT	3-DL-MSC -IMPERF	$+\vert$ ي $+$ ان	يؤويان
4	MPRON	*them*	$+$ هم	يؤويانهم
5	MPREFIX	*so*	$+\vert$ ل	ليؤويانهم

Table 5: Processing steps for the generation of ليؤويانهم.

In general, morphological generation can be described as an easy task compared to morphological analysis. The search space in generation is reduced much because of the linguistic features available. The morphology rule and the morphotactic transformation to be applied can be deduced from the available FS. In morphological analysis, on the contrary, all rules and transformations in the grammar have to be tested whether they succeed or not.

5 Evaluation

We evaluated the proposed system to find out the accuracy of the morphological analyzer. In the experiments, we use data from the LDC Penn Arabic Treebank (ATB) (Maamouri et al., 2004). Data from section 20000915 of the ATB corpus is used in the experiments. Tokens in the ATB corpus contain multiple morphological analysis labels that are produced by the Buckwalter Arabic morphological analyzer (Buckwalter, 2004). Human annotators picked the correct solution among the available ones. Thus, the output of our morphological analyzer was compared with the gold standard test set. Automatic evaluation was not possible because of the format differences between our analyzer and the ATB morphological labels. Therefore we limited the evaluation with the first 500 tokens from the section 20000915. Table 6 lists the category counts in the evaluation data and

noun	171	88%	number	12	92%
adj.	61	90%	pron.	11	100%
proper n.	55	85%	abbrev.	8	100%
punc.	55	100%	part.	5	100%
prep.	50	98%	other	5	100%
verb	49	90%	adv.	3	66%
conj.	15	100%			

Table 6: Word category counts and the percent correct rates in the evaluation data.

the scores in percent correct rates. Overall accuracy rate is 91.4%. 72% of the errors are originated from words that do not exist in the lexicon and there is no analysis for them. In the remaining 28% of the errors, a wrong analysis is produced for the input. The Buckwalter analyzer produced 5,286 different solutions for 500 tokens. Our analyzer produced only 689 solutions. The huge difference is because of our lexicon-driven approach.

6 Conclusions

In this paper, we present a powerful and complete Arabic morphological analyzer employing a unification based approach. We give implementation details on the lexicon, nominal morphology and verbal morphology analysis. Derivational morphology, particularly gerunds, is also explained. Our approach to the generation process is also addressed. Stimulating problems particular to Arabic and our solutions to these problems are explored meanwhile.

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*, 55:189–213.

Kenneth R. Beesley. 1998. Arabic morphology using only finite-state operations. In *Proceedings of the Workshop on Computational Approaches to Semitic languages*, pages 50–57, Montreal, Quebec.

Timothy Buckwalter. 2004. Buckwalter arabic morphological analyzer, version 2.0.

Timothy Buckwalter. 2007. Issues in arabic morphological analysis. In Abdelhadi Soudi, Günter Neuman, and Antal Bosch, editors, *Arabic Computational Morphology*, pages 23–41. Springer.

Lynne Cahill. 2007. A syllable-based account of arabic morphology. In Abdelhadi Soudi, Günter Neuman, and Antal Bosch, editors, *Arabic Computational Morphology*, pages 45–66. Springer.

Violetta Cavalli-Sforza and Abdelhadi Soudi. 2007. Arabic computational morphology: A trade-off between multiple operations and multiple stems. In Abdelhadi Soudi, Günter Neuman, and Antal Bosch, editors, *Arabic Computational Morphology*, pages 89–114. Springer.

Violetta Cavalli-Sforza, Abdelhadi Soudi, and Teruko Mitamura. 2000. Arabic morphology generation using a concatenative strategy. In *Proceedings of the 1st Conference on North American Chapter of the Association for Computational Linguistics: NAACL 00*, pages 86–93, Seattle, Washington.

Kareem Darwish. 2002. Building a shallow arabic morphological analyzer in one day. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic languages*, pages 1–8, Philadelphia, Pennsylvania. Association for Computational Linguistics.

Joseph Dichy and Ali Farghaly. 2007. Grammar-lexis relations in the computational morphology of arabic. In Abdelhadi Soudi, Günter Neuman, and Antal Bosch, editors, *Arabic Computational Morphology*, pages 115–140. Springer.

Nizar Habash and Owen Rambow. 2006. Magead: A morphological analyzer and generator for the arabic dialects. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 681–688, Sydney, Australia, July. Association for Computational Linguistics.

Nizar Habash, Owen Rambow, and George Kiraz. 2005. Morphological analysis and generation for Arabic dialects. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic, June. Association for Computational Linguistics.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The penn arabic treebank: Building a large-scale annotated arabic corpus. In *Proceedings of the NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Abdelhadi Soudi, Günter Neuman, and Antal Bosch, editors. 2007. *Arabic Computational Morphology*. Springer.