ARABIC MORPHOLOGICAL GENERATION FROM INTERLINGUA

A Rule-based Approach

Khaled Shaalan[1&2], Azza Abdel Monem[3], Ahmed Rafea[4]
[1]Institute of Informatics,
The British University in Dubai,
P O Box 502216, Dubai,UAE
[2]Honorary Fellow, School of Informatics, University of Edinburgh
khaled.shaalan@buid.ac.ae
[3]Central Lab. For Agricultural Expert Systems (CLAES),
P O Box: 100 Dokki, Giza, Egypt,
azza@mail.claes.sci.eg
[4]Computer Science Dept., Faculty of Computers and Information,
Cairo Univ., 5 Ahmed Zewel St., Orman, Giza, Egypt,
rafea@mail.claes.sci.eg

Abstract: Arabic is a Semitic language that is rich in its morphology. Arabic has very numerous and complex morphological rules. Arabic morphological analysis has gained the focus of Arabic natural language processing research for a long time in order to achieve the automated understanding of Arabic. With the recent technological advances, Arabic natural language generation has received attentions in order to allow for a room for wider applications such as machine translation. For machine translation systems that support a large number of languages, interlingua-based machine translation approaches are particularly attractive. In this paper, we report our attempt at developing a rule-based Arabic morphological generator for task-oriented interlingua-based spoken dialogues. Examples of morphological generation results from the Arabic morphological generator will be given and will illustrate how the system works. Nevertheless, we will discuss the issues related to the morphological generation of Arabic words from an interlingua representation, and present how we have handled them.

Key words: interlingua-based machine translation, Arabic morphological generation

Please use the following format when citing this chapter:

Shaalan, K., Monem, A.A., Rafea, A., 2006, in IFIP International Federation for Information Processing, Volume 228, Intelligent Information Processing III, eds. Z. Shi, Shimohara K., Feng D., (Boston: Springer), pp. 441–451.

1. Introduction

Arabic is morphologically rich language in which a single inflected word may correspond to a full sentence, (e.g. "سمعتك"—I heard you). Arabic morphological analysis has gained the focus of Arabic natural language processing research for a long time in order to achieve the automated understanding of Arabic [8]. On the other hand, Arabic morphological generation has received little attention in spite of the fact that the types of generation problems can be as complex as those of the analysis [4].

The basic principle of morphological generation is to get inflected forms from a root and a set of features (lexical category and morphological properties). Generally, there are two categories of approaches to developing an Arabic morphological generator: approaches that use finite-state transducers (FSTs) and approaches that use rule-based transformations.

FSTs, such as the one described in [1], are limited to applications that are heavily dependent on morphological generation because the lexical and surface levels are very close. On the contrary, the rule-based transformation approach allows to morphologically generate an Arabic inflected word from the input which is usually a root with a specified feature list. This approach has been used by [3] [4]. The former is a prototype that is restricted in its coverage. The later follows the approach of [2] in that morphotactics and orthographic rules are built directly into the lexicon itself. Our approach is rule-based that uses general transformational rules to address the issue of generating inflected Arabic words in various prefix/suffix contexts. Unlike [4], we use general computational rules that interact to realize the output. The advantages of our approach are that it is easy to incorporate domain knowledge and heuristic rules.

In applications such as interlingua-based machine translation, Arabic morphological generation is an important issue for generating inflected Arabic word forms from semantic representation. This has led us to design and implement an Arabic morphological generator using Prolog. The Arabic word is represented as a feature structure (FS), a Prolog term, which is handled through unification during the morphological generation process. The morphological generator described here has also successfully been used in other natural language processing applications such as Arabic audio indexing [7] and intelligent computer assisted language learning for Arabic [6].

The Arabic generation approach described in this research is developed primarily within the framework of the NESPOLE! (NEgotiating through SPOken Language in E-commerce) multilingual speech-to-speech MT project. The goal of NESPOLE! is to provide speech-translation for common users engaged in real-world e-commerce applications for travel and tourism domain. There are six languages in NESPOLE, English, French, Italian, German, Japanese, and Korean. Arabic will be the seventh

in this family. The Arabic morphological generator has been developed using the interlingua in Carnegie Mellon University (CMU) machine translation [1] and is compatible with the NESPOLE! interlingua specification[2].

The rest of the paper is organized as follows: description of the interlingua representation is summarized in section 2. This is followed by introducing the Arabic morphological generator in Section 3. Next, in Section 4, we discuss the set of important issues that we encountered during the design and implementation of the system. Finally, a conclusion and recommendations for further enhancements are given in section 5.

2. The Description of Interlingua

The NESPOLE! translation system [5] is designed to provide human-to-human speech-to-speech machine translation using an interlingua-based approach similar to that used in the JANUS system [9]. The domain addressed in NESPOLE! is the travel planning, a task-oriented domain. The NESPOLE machine translation project uses an interlingua representation called Interchange Format (IF), which is based on speaker intention rather than literal meaning. The IF defines a shallow semantic representation for task-oriented utterances that abstracts away from language-specific syntax and idiosyncrasies while capturing the meaning of the input. IF is based on a set of domain actions (DA) with parametric arguments. Each DA has up to four components: the *speech act*, the *concepts*, the *arguments*, and a *speaker tag*. Plus sign separate speech acts from the concepts and concepts from each other. In general, each DA has a speaker tag and at least one speech act optionally followed by string of concepts and optionally, a string of arguments. DAs can be roughly characterized as follows:

Speaker: speech act + concept* arguments*

1. a:on the twelfth we have a single and a double available.
   a: give-information+availability+room (provider=we, room-type= (operator=conjunct,[(single_room, quantity=plural),(double_room, quantity=plural)], time=(md=12))
2. a:and we+ll see you on February twelfth.
   a: greeting(conjunction= discourse, greeting=goodbye, to-whom=we, time=(month=2, md=12))

[1] See Carnegie Mellon University (CMU) web site for NESPOLE, http://www.is.cs.cmu.edu/nespole
[2] See interlingua specification for NESPOLE project, http://www.is.cs.cmu.edu/nespole/db/specification.html

3. c:thank you very much
   c: thank

In example (1) the speech act is give-information, the concepts are `availability` and `room` and the arguments are `time` and `room-type`. The possible arguments of DA are determined by inheritance through a hierarchy of speech acts and concepts. In this case `time` is an argument of `availability` and `room-type` is an argument of `room`. Example (2) shows a DA which consists of speech act with no concepts attached to it. The argument time is inherited from the speech act `greeting`. Finally, Example (3) demonstrates a case of DA which contains neither concepts nor arguments.

3. The Arabic Morphological Generator

Arabic morphological generation is an important issue for generating inflected Arabic word forms from semantic representation.

3.1 The Lexicon

An Arabic lexicon was needed to successfully implement the morphological generator. In our approach, we shall consider four basic morphological categories for Arabic—noun, verb, and particle— each with a different set of features. The Arabic word is represented as a FS. We differentiate among three lexical entries: noun, verb, and particle.

The lexicon entry is implemented as a Prolog fact, i.e. lex/2. The first argument is the Arabic stem and the second argument is the Arabic word FS written as a Prolog list. The following describes the forms of the lexicon entry:

1. **Nouns:** A noun has the following form:

lex('Arabic-noun',[Stem,Category,Gender,Number,Sub_Cat,Case,
Irrgular_plural]). Where:

stem:'Arabic-noun'
cat:noun
gender:feminine/masculine/neuter

- number:singular/dual/plural
  sub_cat:demonstrative/proper_noun/common_noun/
  adverb/adjective/question/...,
  definition:yes/no
  case:nomnative/accusative/genitive/...
- irr_pl:'Broken_pl_form'

Examples:

- lex('أجازة',[stem:'أجازة',cat:noun,gender:feminine,number:sg,
  sub_cat:common_noun, definition:no,case:nom,irr_pl:[]]).

- lex('صيف',[stem:'صيف',cat:noun,gender:masculine,number:sg, sub_cat:common_noun, definition:no,case:nom,irr_pl:[]]).

.2 **Verbs:** A verb has the following form:

lex('Arabic-verb',[Stem,Category,Gender,Number,Tense,Case,Sub_Cat,Irregular_past]).
Where

stem:'Arabic-verb'
cat:verb
gender: feminine/masculine
number: singular/dual/plural

- tense:past/present/future
  case:nominative/accusative/...
  sub_cat:intrans/trans/sentence
- irr_past:'Past_Form'

Examples:

- lex('أخطط',[stem:'أخطط ',cat:verb,gender:masculine,number:sg,tense:present,case: nom,sub_cat:intrans, irr_past:[]]).
- lex('أرغب',[stem:'أرغب ',cat:verb,gender:masculine,number:sg,tense:present,case:n om,sub_cat:sentence, irr_past:[]]).

.3 **Particles:** A particle has the following form:

lex('Arabic-word',[Stem,Categor,Sub_Cat]). Where

stem:'Arabicnoun'
cat:particle

- sub_cat:conjunct/preposition...

Examples:

- lex('و',[stem:'و',cat:particle, sub_cat:conjunct]).
  lex('ل',[stem:'ل',cat:particle, sub_cat:preposition]).

3.2 Morphological Features in the Interlingua Representation

In the specification of IF, there are some argument-value pairs which encode the deep semantic of the intended inflected words in the target language. They are morphological features that can be used to generate inflected Arabic word forms from these features. These arguments include:

- `object-spec=pronoun`, In IF representation, this expression is used to indicate a third person pronoun (e.g. 'أحتاجه'—I need it). The generation of this pronoun depends on the gender and number of the Arabic stem.
- `identifiability=yes/no`, indicates the definiteness of a lexeme (e.g. 'الغرفة المزدوجة'—the double room). Other values include `non-distant` demonstrative (this— 'أسم الإشارة للقريب') and `distant` demonstrative (that— 'أسم الإشارة للبعيد').
- `sex=male/female`, indicates the gender of a lexeme (e.g. 'زوجة'—wife).
- `whose=i/we/you/...`, indicates a possessor (e.g. 'زوجتي'—my wife).
- `quantity=all/entire/many/much/some/both`, indicates quantity and quantifier (e.g. 'كل الغرف المزدوجة'—all double rooms).
- `quantity=plural/integer`, indicates quantity (e.g. 'ليلتان'—two nights).

- `object-ref=any`, indicates quantifier (e.g. 'أية مطاعم'—any restaurants).
- `e-time=following/previous`, indicates tense and in some cases it indicates verb "to be" (e.g. 'سنصل'—we will arrive).
- `to-whom=i/we` indicates first person pronoun such as "yeh el Khetab" (e.g. 'أخبرني'—tell me).

3.3 Morphological Generation Rules

A morphological generation rule takes a morphological feature-value pair to be applied and an Arabic word FS, retrieved from the lexicon, as input. Then it applies the required morphological generation action to update the current inflected Arabic word FS being constructed to reflect this change. Morphological generation rules can be classified into rules that are responsible for: synthesis of inflected noun, synthesis of inflected verb, and Synthesis of inflected particle.

Rules for synthesis of inflected noun. Rules for synthesis of inflected Arabic nouns are provided to define noun, feminize noun, pluralize noun, dualize noun, and conjugate a pronoun. Figure 1 shows an example of a noun synthesis rule that defines an Arabic noun. This rule ensures that the noun is not a proper noun, applies the addition of the prefix definition article to the noun, (possibly, a compound noun), then updates the Arabic word FS with then new value of the stem and the definition features.

```
rule: synthesize defined noun
input: Noun or adjective
output: defined noun or adjective
Example: الغرفة المزدوجة - الفنادق – الجيدة

If noun.sub_cat = proper_noun
Then return
else add_prefix(noun.stem,"ال")
```

Figure 1. An example of morphological generation rule for defining a noun

Rules for synthesis of inflected verb. Rules for synthesis of inflected Arabic verbs are provided to conjugate the verb form with regard to tense, number, and affix pronoun. Figure 2 shows an example of a rule for synthesizing a plural form of a verb. This rule conjugates with regard to the tense the number for the verb that may follow the first person pronoun 'we' ('نحن'), or similar expressions, e.g. ('أنا و زوجتي').

```
rule: synthesize plural verb
input: verb
output: pluralized verb
Eample: لاحظنا - نستطيع - سنحتاج

If verb.tense = future
then replace_prefix("سن","سأ")
else if verb.tense = present
    then replace_prefix(verb.stem,"ن","أ")
    else add_suffix(verb.stem,"نا")
```

Figure 2. An example of morphological generation rule for synthesizing a plural form of a verb

Rules for synthesis of inflected particle. Rules for synthesis of inflected Arabic particle are provided to conjugate a particle with suffix pronoun. Figure 3 shows an example of a rule for synthesizing a particle suffix form. This rule conjugates the first person suffix pronoun 'ي' 'yeh' to the particle that connects two verbs.

```
rule: synthesize suffixed pronoun
input: pronoun
output: noun suffixed with connected pronoun
Example: أننا - أنني
        أظن أنني سأصل ...

If Pronoun.number = sg AND
  Pronoun.person = first
then add_suffix(particle.stem,"ني")
else Pronoun.number = pl AND
    Pronoun.person = first
    then add_suffix(particle.stem,"نا")
```

Figure 3. An example of morphological generation rule for synthesizing a suffix pronoun form of a particle

3.4 Morphological Generation Examples

Table 1 shows some examples of the Arabic morphological generation results that are produced form running the morphological generator on the input FSs to get the target inflected Arabic words. These examples illustrate how the inflectional morphology can make use of morphological features.

Table 1. Examples of generating inflected Arabic words from a stem and its morphological features.

Input	Inflected word Output
[...[...'زوج'...],'sex=',female, 'whose=', [... 'أنا'...] ...]	[...[...'زوجتي'...]...]
[...[...'غرفة'...],'whose=',[... 'أنت'...] ...]	[...[...'غرفتك'...]...]
[...[...'أحتاج'...], 'e-time=', following, 'object-spec=',pronoun ...]	[...[...'سأحتاجها'...]...]
[...[...'رؤية'...],'object-spec=', pronoun ...]	[...[...'رؤيتها'...]...]
[...['e-time=',following] ...]	[...[...'سأكون'...]...]
[...[...'أحجز'...],['e-time=', previous]...]	[...[...'حجز'...]...]
[...[...'أعطي'...],'to-whom=',[... 'أنا'...] ...]	[...[...'أعطني'...]...]
[...[...'أسمع'...],'to-whom=',[... 'أنت'...] ...]	[...[...'أسمعك'...]...]
[...['operator=[... 'و'],[[[...'أسرة'...], 'whose=',[... 'أنا'...]],[... 'أنا'...]]]...]	[...[...'أنا'...],[...'و'...] [...أسرتي'...]...]
[...[...'أن'...],'whose=', [... 'أنا'...] ...]	[...[...'أنني'...]...]
[...[...'برامج شتوية'...],'whose=', [... 'نحن'...] ...]	[...[...'برامجنا الشتوية'...]...]

4. Issues and Problems

In this section, we will discuss issues related to the generation of Arabic text from an interlingua representation, and present how we have handled them. These issues have arisen when we were integrating the Arabic morphological generator with the Arabic generator module of the machine translation system.

4.1 Issues Related to Definite Nouns

The argument `identifiability=` has two values that can be used as morphological features (`yes` to indicate definite noun and `no` to indicate indefinite noun). Another value is `non-distant` that is used to indicate a demonstrative noun. They are mutually exclusive such that we cannot get a case where a noun is modified by a demonstrative and is also definite (i.e. substitution form). We found the substitution form is always the case with demonstratives. We solved this problem by assuming that the noun that the demonstrative modifies is definite with the article.

Numbers that are values of the argument `quantity=` indicates a number associated with a counted name—a value of the parent argument. Although we generate definite numbers in Arabic we cannot have this form because the IF specification of the argument `quantity=` does not have `identifiability=`

as a subargument. This problem is so specific to Arabic that we cannot generate definite numbers.

4.2 Issues Related to Numbers and Counted Nouns

Number-counted noun expression is governed by a set of complex rules for determining the gender, definiteness, and case markings. The markings of numbers depend on the occurrence of the number within the sentence.

- The number 'one', agreement is as expected, but there may be a reversal of word order (e.g. 'شخص واحد' *(one person)* and 'ليلة واحدة' *(one night))*.
- The number 'two' is expressed by the dual of the noun (e.g. 'شخصان—شخصين' *(two person)* and 'ليلتان—ليلتين' *(two nights))*.
- Numbers 'three' through 'ten' require the counted noun to be plural and the gender of the number to be the opposite of the gender of the singular noun. For example: خمس (five, masculine) سنوات (plural of سنة 'year', feminine) but خمسة (five, feminine) متاحف (plural of متحف 'museum', masculine).
- Compound numbers 'eleven' and 'twelve' require a singular counted noun in the indefinite accusative and agrees in the gender with the counted noun.
- Compound numbers 'thirteen' through 'nineteen' require a singular counted noun in the indefinite accusative. They also require the gender of the first part of the number to be the opposite of the gender of the counted noun and gender of the second part agrees with counted noun.
- Decades (Numbers '20' through '90', and hundred, thousand, etc.) require a singular counted noun in the indefinite accusative and the number to be sound masculine plural.

  Conjunction of numbers (units, tens, hundreds, thousands, etc.) follows the above rules and the individual numbers are separated by the conjunction particle waw 'و'.

Agreement decisions are made in the generator to synthesis the correct form of the numbers and their counted nouns.

Another issue is the mapping of ordinal numbers (first, second, etc.) and cardinal numbers (one, two, etc.). These depend on argument-value mapping. For example, the value of the argument hours= is mapped to a cardinal number and the value of the argument md= is mapped to an ordinal number.

```
your room will be available at eleven o'clock
a:give-information+feature+room (e-time=following, room-
spec=(room, whose=you), feature=(modifier=available),
time=(start-time=clock=(hours=2))))
   غرفتك ستكون متاحة الساعة الحادية عشر

I and my wife will be arriving February eleventh
```

```
c:give-information+arrival (who=(operator=conjunct,[i,(spouse,
sex=female,whose=I)]), e-time=following, time=(month=2,md=11))
```

أنـا وزوجتي سنصل في الحادي عشر من فبراير

4.3 Issues Related to Arabic script

During inflectional morphology some letters of Arabic changes into other forms for example:

- The Alef letter ('ا') of the definite article 'ال' is dropped when used with preposition 'ل'. For example, using 'ل' with 'الطفل' produces 'للطفل'.
- The Hamza letter is changed to other forms during the morphological generation of the inflected word. For example, the use of Yeh ('ي') El-Khetab pronoun with the broken plural 'زملاء' should produce 'زملائي' instead of 'زملاءي'
- The feminine Teh ('ة') is change into ('ت') when a suffix is attached to it. For example, the dual of 'غرفة' is 'غرفتان-غرفتين'.

5. Conclusions and Future Work

In this paper, we described the development of a novel Arabic morphological generator. The morphological generator is implemented in SICStus Prolog and takes the advantage of Prolog's built-in term-unification. The morphological generator follows the rule-based approach. The advantages of this approach are that it is easy to incorporate domain knowledge and heuristic rules into the linguistic knowledge which provide highly accurate generations that represents a speaker's intention for each semantic segment. We have separately evaluated our Arabic morphological generator and the results were satisfactory. We have discussed the problems encountered in the generation of inflected Arabic words from the interlingua representation used in NESPOLE!. For these problems we have described how we handled them. Our morphological generator has also successfully been used in other natural language processing applications such as Arabic audio indexing and intelligent computer-assisted language learning for Arabic.

Future work will include extending the Arabic morphological generator to colloquial dialects of Arabic. Another interesting challenge would be to introduce diacritics into the lexicon. Text in Arabic is generally written without the diacritics (or vowels) and these are sometimes essential for the disambiguation of words.

References

1. Beesley, K. Arabic finite-state morphological analysis and generation. In Proceedings of the 16th International Conference on Computational Linguistics (COLING-96), Copenhagen, Denmark, (1996) (1): 89–94.
2. Buckwalter, T. Buckwalter Arabic Morphological Analyzer Version 1.0. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Catalog No.: LDC2002L49, 2002.
3. Cavalli-Sforza V., Soudi, A., Mitamura, T. Arabic morphology generation using a concatenative strategy. In Proceedings of the 6th Applied Natural Language Processing Conference (ANLP 2000), Seattle, Washington, USA, (2000) 86–93,.
4. Habash N. Large scale lexeme based Arabic morphological generation. In Proceedings of Traitement Automatique du Lan-gage Naturel (TALN-04). Fez, Morocco, 2004.
5. Lavie, A., Metze, F., Pianesi, F., Burger, S., Gates, D., Levin, L., Langley, C., Peterson, K., Schultz, T., Waibel, A., Wallace, D., McDonough, J., Soltau, H., Cattoni, R., Lazzari, G., Mana, N., Pianta, E., Costantini, E., Besacier, L., Blanchon, H., Vaufreydaz, D., Taddei, L., Enhancing the Usability and Performance of NESPOLE! - a Real-World Speech-to-Speech Translation System. In Proceedings of HLT-2002 Human Language Technology Conference, San Diego, CA, March (2002).
6. Shaalan K. An Intelligent Computer Assisted Language Learning System for Arabic Learners, Computer Assisted Language Learning: An International Journal, Taylor & Francis Group Ltd. (2005) 18(1 & 2): 81-108.
7. Shaalan, K., Talhami H., and Kamel I., A Morphological Generator for the Indexing of Arabic Audio, In the Proceedings of The IASTED International Conference on Artificial Intelligence and Soft Computing (ASC), September 12-14, Benidorm, Spain, (2005) 308-312.
8. Sughaiyer I., Al-Kharashi,I. Arabic Morphological Analysis Techniques: A Comprehensive Survey. Journal of The American Society for Information Science and Technology, (2004) 55(3):189-213.
9. Waibel, A., Jain, A., McNair, A., Tebelskis, J., Osterholtz, L., Saito, H., Schmidbauer, O., Sloboda, T., Woszczyna, M., JANUS: Speech-to-Speech Translation Using Connectionist and Non-Connectionist Techniques. Advances in Neural Information Processing Systems, (1992) (4).