Rule-based Approach in Arabic Natural Language Processing

Khaled Shaalan

Abstract— **The rule-based approach has successfully been used in developing many natural language processing systems. Systems that use rule-based transformations are based on a core of solid linguistic knowledge. The linguistic knowledge acquired for one natural language processing system may be reused to build knowledge required for a similar task in another system. The advantage of the rule-based approach over the corpus-based approach is clear for: 1) less-resourced languages, for which large corpora, possibly parallel or bilingual, with representative structures and entities are neither available nor easily affordable, and 2) for morphologically rich languages, which even with the availability of corpora suffer from data sparseness. These have motivated many researchers to fully or partially follow the rule-based approach in developing their Arabic natural processing tools and systems. In this paper we address our successful efforts that involved rule-based approach for different Arabic natural language processing tasks.**

Index Terms— **Arabic morphology, Arabic natural language processing, Arabic Parsing, Arabic syntax, Intelligent computer assisted language learning, machine translation, named entity recognition, rule-based approach, rule-based natural language processing tools, rule-based natural language processing systems..**

I. Introduction

ARABIC is a Semitic language spoken by more than 330 million people as a native language, in an area extending from the Arabian/Persian Gulf in the East to the Atlantic Ocean in the West. Moreover, it is the language in which 1.4 billion Muslims around the world perform their daily prayers. Arabic is a highly structured and derivational language where morphology plays a very important role [3], [5], [6], [8], [28].

Over the last few years, Arabic natural language processing (ANLP) has gained increasing importance, and several state of the art systems have been developed for a wide range of applications. These applications had to deal with several complex problems pertinent to the nature and structure of the Arabic language [9]. The lack of available resources and their limitations have motivated many scholars to follow the rule-based approach and rely on hand-constructed linguistic rules in developing their tools, systems, and resources. ANLP tools based on this approach generally include morphological analyzers/generators and syntactic analyzers/generators. The approach is also used for some specific tasks. For example, the work reported on in [24] transfers Egyptian Arabic texts to Modern Standard Arabic (MSA) using a lexical transfer approach in addition to changing the SVO Egyptian order into the MSA VSO order. Moreover, rule-based ANLP systems include machine translators, named entity recognizers, and intelligent computer assisted language learning systems.

Systems and tools that use rule-based transformations are based on a core of solid linguistic knowledge [1]. The characteristics of a rule-based approach are:

1. It has a strict sense of well-formedness in mind,
2. It imposes linguistic constraints to satisfy well-formedness,
3. It allows the use of heuristics (such as a verb cannot be preceded by a preposition), and
4. It relies on hand-constructed rules that are to be acquired from language specialists rather than automatically trained from data.

The advantages of this approach are that it is easy to incorporate domain knowledge into the linguistic knowledge which provides highly accurate results. Domain rules have been used in generating Arabic sentences [27] and analysis of ill-formed learner input [11]. Furthermore, the linguistic knowledge acquired for one natural language processing system may be reused to build knowledge required for a similar task in another system.

Because time was of the essence, and in the absence of complete computationally viable grammars of Arabic, statistical approaches that rely primarily on training data and parallel texts gained momentum. Machine learning approach usually gives good results when the training set and the testing data are similar. There is also a point at which more training data does not make significant improvement. Moreover, there may be some structures or entities that are sparse. In this case the machine learning component does not have enough data to make the right generalization. Regardless of sparseness of the data, the statistical-based or machine learning approaches have some difficulties with specific natural language processing tasks such as distinguishing between well-formed and ill-formed input, whereas the rule-based approaches have the advantage of providing detailed analyses of the Arabic learner's answer using linguistic (morphological and syntactic) knowledge which, in applications such intelligent tutoring

Manuscript received November 12, 2009.

Khaled Shaalan is with the The British University in Dubai, Dubai, PO Box 502216 Dubai, UAE; (Fellow) School of Informatics University of Edinburgh, UK (phone: 971-4-3671963; fax: 971-4-3664698; E-mail: khaled.shaalan@buid.ac.ae).

systems, enables feedback elaboration that helps learners to understand better their knowledge gab.

In this paper we discuss our experiences in developing successful rule-based Arabic language processing tools and systems.

The rest of this paper is structured as follows. In section II, we briefly highlight some important background aspects of the Arabic language. Section III presents some rule-based Arabic natural language processing tools including morphological and syntactic analyzers/generators. Section IV describes some rule-based Arabic natural language processing systems including machine translation, name-entity recognition, and intelligent tutoring systems. Finally, in Section V, we draw some concluding remarks.

II. ASPECTS OF THE ARABIC LANGUAGE

Arabic is rooted in the Classical or Quranic Arabic, but over the centuries, the language has developed to what is now accepted as MSA. MSA is a simplified form of Classical Arabic, and follows its grammar [1]. The main differences between Classical Arabic and MSA are that MSA has a larger (more modern) vocabulary, and does not use some of the more complicated forms of grammar found in Classical Arabic. For example, short vowels are omitted in MSA such that letters of the Arabic text are written without diacritic signs.

The Arabic language is written from right to left. It has 28 letters, some of which have one form (like "د"), while others have two forms ("س" ;"سـ"), three forms ("ه" ;"ـهـ" ;"هـ") or four forms ("14] ("ع" ;"ـع" ;"ـعـ" ;"عـ"]. Arabic words are generally classified into three main categories [19]: noun, verb, and particle.

Arabic is a language of rich and complex morphology, both derivational and inflectional [9]. Word derivation in Arabic involves three concepts: root, pattern, and form. Word forms (e.g. verbs, verbal nouns, agent nouns, etc.) are obtained from roots by applying derivational rules to obtain corresponding patterns. Generally, each pattern carries a meaning which, when combined with the meaning inherent in the root, gives the target meaning of the lexical form. For example, the meaning of the word form "كاتب" (writer) is the combination of the meaning inherent in the root "كتب" (write) and the meaning carried by the pattern (or 'template') "ف-ا-ع-ل" (fa'il) which is the pattern of the doer of the root. Arabic inflectional morphology involves adding morphosyntactic features such as tense, number, person, case, etc.

Arabic also has some more morphological peculiarities. For example, an indefinite word can be made definite by attaching the prefix definite article "الـ" (the) to it, but there is no indefinite article. As another example, a verb can take affix pronouns such as "سأعطيكما" (will-I-give-you); this also shows that the verb is conjugated with the dual suffix pronoun "كما" (you). An Arabic inflected verb can form a complete sentence, e.g. the verb "سمعتك" (heard-I-you) contains a complete syntactic structure in just a one-word sentence. Moreover, the rich morphology of Arabic allows the dropping of the subject pronoun ('pro-drop'), i.e. to have a null subject when the inflected verb includes subject affixes.

There are two types of Arabic sentences [15]: nominal and verbal sentence. An Arabic compound sentence is formed from a simple sentence followed by a complementary sentence [10], such as a conjunction form (عطف), e.g. " نحن نرغب في تأجير سيارة وسنحتاج لساحة انتظار قريبة من الفندق" (We want to rent a car and we-will-need to park near the-hotel), or a quasi-sentence (شبه جمله), e.g. "بالفندق" (in-the-hotel).

Agreement is a major syntactic principle that affects the generation of an Arabic sentence. Agreement in Arabic is full or partial and is determined by word order [15]. An adjective in Arabic usually follows the noun it modifies ("الموصوف") and fully agrees with it with respect to number, gender, case, and definiteness. The verb in Verb-Subject-Object order agrees with the subject in gender, e.g. "جاء الولد / الأولاد" (came the-boy/the-boys) versus "جاءت البنت / البنات" (came the-girl/the-girls). In Subject-Verb-Object (SVO) order, the verb agrees with the subject with respect to number and gender, e.g. " الولد جاء / الأولاد جاءوا" (came the-boy/the-boys) versus " البنت جاءت / البنات جئن" (came the-girl/the-girls). For more details regarding aspects of the Arabic language, including agreements in Arabic, we refer the reader to [4]. .

III. RULE-BASED ARABIC NATURAL LANGUAGE PROCESSING TOOLS

In the rule-based approach, two components can usually be distinguished in an analyzer/generator [1]: a declarative component corresponding to linguistic knowledge and a procedural component which represent the analysis/generation strategy. Linguistic knowledge includes the grammar and the lexicon of the language while analysis/generation strategy is an algorithm which specifies in detail each of the operations involved in the process of analysis/generation. In the following subsections, we will discuss our experiences in developing tools that have successfully been used in different ANLP applications.

A. *Arabic Morphological Analyzers*

As Arabic is a morphologically rich language, morphological processing plays a key role in developing Arabic NLP systems and applications. For a comprehensive survey of the subject we refer the reader to [2], [28]. The basic principle of morphological analysis is to breakdown an inflected form into a root and a set of features (lexical category and morphosyntactic properties). Arabic is considered to be a non-concatenative language because, for some forms, it alters the radical letters within the stem according to syntactic context [7]. This can be best clarified by an example showing the inflections of Arabic weak verbs, which include one or more weak letter. Weak letters can be deleted or substituted by other letters because of Arabic linguistic theory. For example, the replacement of the letter (و) /w/ by (ا) /A/ can be explained by taking the past (perfect) tense of the trilateral root ق-و-ل /q-w-l/ as an example. In its abstract form, using regular rules would erroneously generate قَوَلَ* /qawala/ but as it is a hollow[1] verb it should be generated according to special weak rules and thus it should appear in written texts as قَالَ /qAla/ (said).

In order to analyze an inflected Arabic word, we need a sophisticated morphological analyzer that is capable of transforming the inflected form into its origin. To achieve this function we developed a rule-based morphological analyzer for inflected Arabic words [14] which has been used as a core module for the analysis of the words and sentences in a number of systems; some of them described here. An Augmented Transition Network (ATN) [30] technique was successfully used to represent the context-sensitive knowledge about the relation between a stem and its inflectional additions, see Figure 1. The ATN consists of arcs. Each of which is a rule that links a departure node to a destination node, called states. More than one rule may be associated with one arc which allows actions, such as omit affix or convert radical letter, to be associated with each arc. Figure 2 shows an example of a verb analysis rule that removes a doubled letter; see the transition from S3 to S4 in Figure 1.

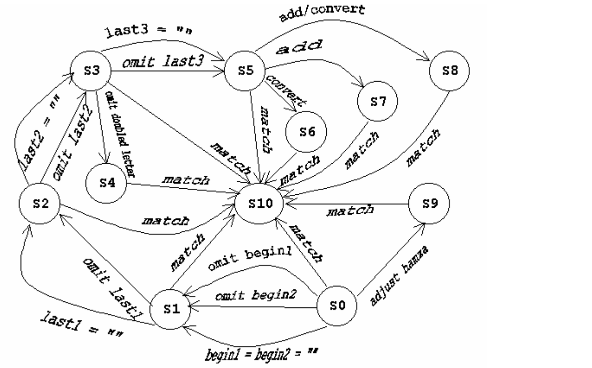

Fig. 1. ATN representing the relation between the additions and root of an inflected Arabic word.

For example, this rule analyzes the verb "مددنا" into the stem 'مد' and suffix 'نا'. An exhaustive-search to traverse the ATN generates all the possible interpretations of an inflected Arabic word. The algorithm starts from the node S0 and nondeterministically generates analyses of the input when it reaches the final node S10.

Another set of rules will apply on the word segments and features of the lexical entry of the root to determine the morpho-syntactic features of the input inflected word. For example the morpho-syntactic features of the root 'مد' and suffix 'نا' are, *category:verb, sub_category:doubled_verb, transitivity:transitive, tense:perfect*, and *number:dual,* respectively.

```
Input: Inflected doubled verb
Output: stem and suffix
Omit doubled letter rule:
 If Suffix = "ن" ,"نـا", or "ت"
 then omit doubled letter from verb.stem
End Rule
```

Fig. 2. An example of morphological analysis rule for a doubled verb

There are a number of issues concerning the morphological analysis of Arabic words. An important is issue is the handling of ill-formed input in order to give response to the user, especially in education systems. Usually morphological analyzers assume the input is a well-formed inflected Arabic word. However, morphological analyzers, in some cases, might handle ill-formed input in three ways [17]. One way is by applying compatibility check rules to prevent ill-formedness, such as checking that the prefix "ف" cannot come after the prefix "ب" or "ك". A second is by augmenting the linguistic rules with heuristic rules, so-called buggy rules, which are capable of spell checking ill-formed input and apply only if the morphological rules fail. A third way is by normalizing the input in order to deal with the different variations in MSA script, such as replacing the initial *alif* with a hamza above or below, or *alif madda* with simply an *alif*, (i.e. *bare alif).*

Another issue, which concerns the morphological analysis of Arabic words, is the overgeneration where too many output are generated. In this case developer should provide mechanisms to filter out irrelevant or rarely used constructions. For example, an expected verb input should exclude all analyses that refer to a noun. So, known features of the input can be useful in producing less ambiguous output.

B. Arabic Morphological Generators

We have developed a rule-based Arabic morphological generator that is able to generate an inflected Arabic word from a stem and morpho-syntactic features using an Arabic monolingual lexicon and Arabic morphological rules [21]. This lexicon includes vocabulary from the travel and tourism domain which is specified in the Interligua specification of the multilingual machine translation system [1]. Each rule is responsible for applying a single feature on a given stem to yield an inflected form. The input stem is represented as a feature structure (FS), and the feature—value pair corresponding to the inflectional operation to be applied. rules have conditions (or constraints) and actions. When the condition is met, the action is applied which results in the FS being updated to reflect this change. Generating an inflected form using multiple features (e.g. to derive a definite plural noun) is applied one rule at a time. Figure 3 shows a rule for synthesizing a first person plural form of a hollow ('middle weak') verb in the active voice with respect to the verb tense. In case of the perfect form of the verb, the rule should remove the middle weak letter before attaching the suffix pronoun. The position of this letter is recognized by matching the stem of the

[1] A Hollow verb has a weak middle radical. it falls into four categories:

- Verbs where middle radical is (و) /w/ of the pattern فَعَلَ /faEala/ such as صَوَمَ /Sawama/ 'fasted' or فَعُلَ /faEula/ such as طَوُلَ/Tawula/ 'he/it was long',
- Verbs where middle radical is (و) /w/ of the pattern فَعِلَ /faEila/ such as خَوِفَ/xawifa/ 'was afraid or frightened',
- Verbs where middle radical is (ي) /y/ of the pattern فَعَلَ /faEala/ such as سَيَرَ /sayara/ 'walked', and
- • Verbs where middle radical is (ي) /y/ of the pattern فَعِلَ /faEila/ such as حَيِرَ/Hayira/ 'was confused or hesitated'.

hollow verb with its pattern. For example, for the input "استطاع" (could) in the perfect tense, this rule generates "استطعنا" (could-we).

There are a number of issues concerning the morphological generation of Arabic words. One of them is the generation of a literal number as well as the agreement with its counted noun expression which is governed by a set of complex rules for determining the *gender*, *definiteness*, and *case markings*. For instance, synthesizing a number between 3 and 9 should agree with its counted noun in *gender*. In this case, the gender of the literal number is the opposite of the gender of the singular form of the counted noun, e.g. “خمس سيارات” “five.masc.sg car.fem.pl”. Another issue is the Arabic script changes as a result of the application of morphological rules, e.g. the prefix “ل” Lam (for) when attached to a definite noun it removes the first letter of the definite article, i.e. both prefixes become “للـ”. In the same manner, attaching the possessive suffix pronoun “ي” (my) to a noun that ends with “ء’, an isolate hamza, such as “زملاء” (colleagues) it changes the last letter to “ئ”, i.e. “زملائي” (colleagues-my).

```
Input: first person singular verb
Output: first person plural verb
Synthesize a plural form of a hollow verb Rule:
If verb.tense = future
then replace the prefix ‘سأ’ with ‘سن’
else if verb.tense = present
     then replace the prefix ‘أ’ with ‘ن’
     else
     begin
       get Position of weak letter in verb.pattern,
       use Position to remove middle weak letter
         from verb.stem,
       attach suffix "نـا" to verb.stem
     end
End Rule
```

Fig. 3. A morphological generation rule for synthesizing a first person plural form of a hollow verb

C. Arabic Syntactic Analyzers

We have two successful attempts to build an Arabic parser. In the first one we developed an Arabic parser for modern scientific text [16]. Figure 4 shows a rule for parsing a nominal sentence that agrees in *number* and *gender* but disagree in *definiteness* such as “الطالبة مجتهدة” (the-student.definite.sg.fem [is] diligent.indefinite.sg.fem).

```
Input: simple nominal sentence
Output: success or failure of parsing
Nominal sentence Rule:
If   inchoative.defniniteness=definite,
     inchoative.number=enunciative.number,
     inchoative.gender=enunciative.gender,
     enunciative.defniniteness=indefinite,
then accept nominal sentence
End Rule
```

Fig. 4. A rule that include constraints for analyzing a simple nominal sentence

There are a number of issues concerning the syntactic analysis of Arabic sentences. A major issue is the parsing ambiguity. We developed another efficient Chart parser [12], which is able to satisfy syntactic and semantic constraints in order to reduce parsing ambiguity (cf. [13]). Figure 5 shows a grammar rule of a definite noun object with three constraints that should be met in order to apply this rule: 1) *semantic constraints*: the object should be neither a demonstrative noun nor a connected pronoun, and 2) *syntactic constraint*: the object should be neither in a nominative nor genitive case.

```
Input: definite noun object
Output: success or failure of parsing
Disambiguation of definite noun object Rule:
If   Noun.definitness = definite,
     Noun.category ≠ demonstrative noun,
     Noun.category ≠ connected pronoun,
     Noun.end_case ≠ nominative,
     Noun.end_case ≠ genitive
then definite noun object
```

Fig. 5 A grammar rule with simple disambiguation constrains

Another issue, which concerns the syntactic analysis of Arabic sentences, is the grammatical checking of Arabic sentences. We developed an Arabic grammatical checker [19] as an extension of the aforementioned Chart parser. This grammatical checker is aimed at helping the average user by checking his/her writing for certain common grammatical errors, describing the problem for him/her, and offering suggestions for improvement. In our implementation, the error detection is embedded within the grammar rule and is based on the unification of the FSs to determine the source of the grammar error. Figure 6 shows an example rule. This rule says that in order to precede a verb with a particle (accusative or apocopative) some constraints must be satisfied:

1. The verb must not be in the past tense,
2. The verb must not be in the nominative case, and
3. The particle must not be a preposition.

If any of the above constraints is not satisfied, then the whole rule will fail and an error message reporting which type of error has occurred will be issued.

```
Input: a particle followed by a verb
Output: success or failure of parsing. A feedback
message in case of failure
Grammar checking of a particle followed by a verb
Rule:
feedback_message=‘’
if  verb.tense = past
then append feedback_message with ‘ لا يأتي الفعل
الماضي بعد أداة النصب أو الجزم’,
if  verb.end_case = nominative
then append feedback_message with ‘ لا يكون الفعل
مرفوع بعد أداة النصب أو الجزم’,
if particle.category = preposition
then append feedback_message with ‘ لا يسبق الفعل حرف
جر’,
if feedback_message=‘’
then accept particle followed by a verb
else issue feedback_message and fail
End Rule
```

Fig. 6 A grammar checking rule

D. Arabic Syntactic Generators

There are two major interdependent syntactic issues concerning the syntactic generation of Arabic sentences: Word order and agreement. In Arabic, agreement is full or partial and is determined by word order, i.e. VSO, SVO, etc.

We have developed a ruled-based Arabic syntactic generator that consists of two steps: 1) Determining the syntactic structure of the Arabic sentence [22], and 2) generating the surface Arabic sentence [27]. Structural mapping rules are used to determine the syntactic structure of the Arabic sentence by first recognizing a set of constituents from word senses. Figure 7 illustrates a structural mapping rule that transforms the constituents into a feature structure that conforms to the syntactic rule that consists of a coordinator followed by SVO order.

```
Input: Set of Arabic words
Output: syntactic structure of an Arabic Sentence
Structural mapping of constituents to Cord SVO Rule:
If  Recognize a coordination from the input Words
    Recognize a verb from the input Words,
    Recognize a subject from the input Words,
    Recognize a complement from the rest of
      input words,
Then Sentence=[coordination,subject,verb,complement]
End Rule
```

Fig. 7 A structural mapping rule

Once the syntactic structure of a sentence is determined, another set of rules are used to ensure the agreement relations between various elements in the sentence. Arabic is rich in agreement. We have implemented rules for different types of agreement relationships, such as subject–verb, noun–adjective, demonstrative pronoun–noun, and number–counted noun [27].

In Arabic, an adjective agrees with the noun it modifies with respect to *number*, *gender*, and *definiteness*, except in the case of an irregular (broken) plural, where it partially agrees in *gender* and *definiteness*. Figure 8 shows a rule for synthesizing an adjective that agrees with the noun it modifies. For example, consider the sentence: “الأولاد زاروا المتاحف قديم” (the-boys visited-they the-museum.fem.pl old.masc.sg). In this case, the adjective, "قديم" (old [masc.sg]) and the (broken plural) noun it modifies "المتاحف" (the-museums [the-museum.fem.pl]) should agree in *gender* and *definiteness*. The generated Arabic sentence would therefore be as follows: “الأولاد زاروا المتاحف القديمة” (the-boys visited-they the-museum.fem.pl the-old.fem.sg).

IV. RULE-BASED ARABIC NATURAL LANGUAGE PROCESSING SYSTEMS

In this section we describe how the rule-based Arabic language processing tools are used to build state-of-the-art natural language processing systems.

A. Machine translation

Machine translation (MT) is the area of information technology and applied linguistics dealing with the translation of human languages such as English and Arabic. There are three different approaches of rule-based translation systems [29]: direct, transfer, and interlingual. The simplest approach is the direct translation approach where a word-by-word translation (lexical transfer) from the source language to the target language is performed. From a linguistic point of view, what is missing in this approach is any analysis of the internal structure of the source text, particularly the grammatical relationships between the constituents of the sentences. Such systems gave the kind of translation that was characterized by frequent mistranslations at the lexical level and largely inappropriate syntactic structures which mirrored too closely those of the source language.

```
Input: an adjective and the noun it modifies
Output: inflect adjective agreed with the
noun it modifies
Noun-adjective agreement Rule:
If Noun.number = broken_plural
Then
  begin
    synthesize_noun(Noun.gender,Adj.stem)
    synthesize_noun(Noun.definiteness,
    Adj.stem)
  end
else
  begin
    synthesize_noun(Noun.number,Adj.stem)
    synthesize_noun(Noun.gender,Adj.stem)
    synthesize_noun(Noun.definiteness,
     Adj.stem)
  end
End Rule
```

Fig. 8. A rule for noun–adjective agreement

In the transfer approach, the translation process is decomposed into three steps: analysis, transfer, and generation. In the analysis step, the input sentence is analyzed syntactically (and in some cases semantically) to produce an abstract representation of the source sentence, usually an annotated parse tree. In the transfer step, this representation is transferred into a corresponding representation in the target language; a collection of tree-to-tree transformations is applied recursively to the analysis tree of the source language in order to construct a target-language analysis tree. In the generation step, the target-language output is produced. The (morphological and syntactic) generator is responsible for polishing and producing the surface structure of the target sentence.

We developed a transfer-based machine translation system of English noun phrase to Arabic. The architecture is shown in Figure 9 [18]. The analysis and generation components are similar to the ones described in Section III. In our noun phrase translator, the actual translation occurs in the transfer step in which one side of the tree-to-tree transfer rules is matched against the input structure, resulting in the structure on the right-hand-side. Figure 10 illustrates the translation of the noun phrase (NP) “networks performance evaluation” into “تقييم أداء شبكة”, which shows the switching of words that is indicated by the following transfer rule:

$$\begin{array}{ll} [w_i:\$1,\ w_{i+1}:\$2,\ \ldots,\ w_k:\$k] \Leftrightarrow & (1 \le i \le k) \\ [w_k:\$k,\ w_{k-1}:\$k-1,\ \ldots,\ w_i:\$i] & (1 \le i \le k) \end{array}$$

This rule says that the translation of the word at level `i` is switched with the word at level `k-i+1`. Where k is the number of NPs equivalent to maximum (sub)tree level.

Transfer-based approach in some cases cannot give a correct transfer of distance relationships. For example, consider the translation of “intelligent tutoring systems” into “نظـم التعلـيم الذكيـة” (intelligent.adj.fem tutoring.noun.masc systems.noun.fem) as the adjective “intelligent” refers to the noun “systems” which is feminine and not to the noun “tutoring” which is masculine. The noun phrase translator translated it as “نظـم التعلـيم الـذكي” (intelligent.adj.masc tutoring.noun.masc systems.noun.fem) and incorrectly takes the noun directly preceding the adjective as the noun it modifies.

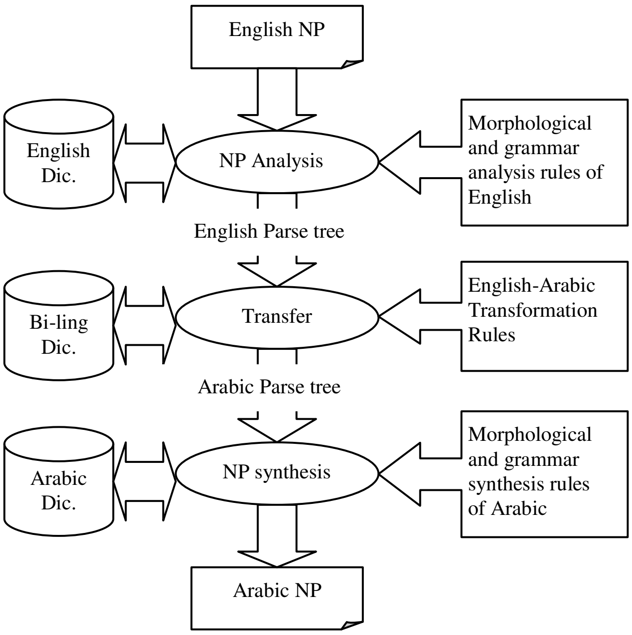
English NP
English Dic.
NP Analysis
Morphological and grammar analysis rules of English
English Parse tree
Bi-ling Dic.
Transfer
English-Arabic Transformation Rules
Arabic Parse tree
Arabic Dic.
NP synthesis
Morphological and grammar synthesis rules of Arabic
Arabic NP

Fig. 9. Overall structure of English-Arabic noun phrase translator

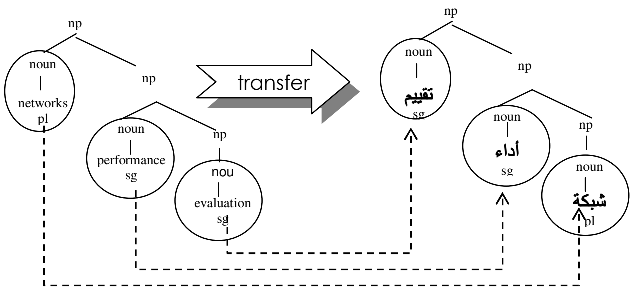

Fig. 10. Simple transfer

The Interlingual approach MT is used successfully in multilingual translation. It aims to achieve the translation task in two independent steps. First, meanings of the source-language sentences are represented in an intermediate language-independent (Interlingua) representation. Then, sentences of the target language are generated from those meaning representations. We developed a rule-based Interlingua-to-Arabic generator using the morphological and syntactic generators described in Section III [1]. The Interlingua representation plays an important role in the accuracy of the translation as it should be a language-neutral representation and captures the intended meaning of the source sentence.

B. *Named Entity Recognition*

A Named Entity Recognition (NER) system is a significant tool in natural language processing research since it allows identification of proper nouns in open-domain (i.e., unstructured) text. We developed the system, Named Entity Recognition for Arabic (NERA), using a rule-based approach (see [23], [25], and [26]). The system covers the ten most important categories of named entities in Arabic script; the person name, location, company, date, time, price, measurement, phone number, ISBN and file name. The recognition and extraction of Arabic named entities from the input text is based on derived rules obtained from deep contextual analysis of a large amount of diverse data. The rules include indicator patterns of named entities, such as job title and person title, and morphological information to strip off various prefixes and suffixes from inflected forms. Figure 11 shows an example of a NERA's grammar rule (in regular expression representation) for Person name recognition.

```
((honorfic+ws(location_name( ي|يـة)+ws)?)+
first_name (ws+last_name)?ws+(number)?)
```

Fig. 11. A grammar rule for person name recognition

The rule is interpreted as follows:

- The names should be verified against their respective dictionaries (i.e. first, middle, and last names).
- The indicator pattern is composed of an honorific such as "الملك" [The king] followed by an optional *Nisba* derived from a location name such as "الأردني" [Jordanian]. These act as trigger words to recognize the person name and should be verified against their respective dictionaries of honorific and locations.
- The rule also matches an optional ordinal number appearing at the end of some names such as "الثاني" [II].
- The Arabic suffix letters "ية" and "ي" used in the above pattern parses the inflections attached to *Nisba* derived from locations that are commonly found in Arabic text.

This rule recognizes a person name composed of a first name followed by optional last name based on a preceding person indicator pattern. The following name entities would be recognized by the above rule:

[The king Abdullah]الملك عبد الله
[The Jordanian king Abdullah]الملك الأردني عبد الله
[The Jordanian king Abdullah II]الملك الأردني عبد الله الثاني
[The Jordanian queen Rania]الملكة الأردنية رانيا

Apart from contextual cues, the typical Arabic naming elements were used to formulate rules such as *nasab*, *kunya*, etc. Thereby the rules resulted in a good control over critical instances by recognizing complex entities.

In NERA, we addressed major challenges posed by NER in the Arabic language [26], arising due to: the complexity of the morphological system, peculiarities in the Arabic orthographic system, non-standardization of the written text, ambiguity, and lack of resources.

C. Intelligent Computer Assisted-Language Learning

Computer-assisted language learning addresses the use of computers for language teaching and learning. Intelligent computer-assisted language learning started as a separate research field, when Artificial Intelligence (AI), in particular natural language processing, technologies were mature enough to be included in language learning systems. We developed an intelligent computer assisted language learning system for Arabic learners [20]. This system could be used for learning Arabic by students at primary schools or by learners of Arabic as a second or foreign language. The system has two natural language processing components: sentence analysis and error analysis.

The sentence analysis includes two types of rules. The first type includes knowledge required to analyze a grammatically correct sentence. The sentence analysis rules are similar to the grammar rules described in Section III.

The second type of rules is used to parse the learner's answer in response to a question about the linguistic analysis (إعراب) of a given Arabic sentence. The free text learner's answer is converted into a quadruple abstract representation of the canonical form:

السبب	Reason
علامة الإعراب	◌ٍ◌ِ◌ِA◌ِnalytic sign
الإعراب	End case
الموقع الإعرابي	analytic location

This abstract representation is unique and unambiguous such that it facilitates comparing the learner's answer with the correct answer generated by the system.

For example, the following is the linguistic analysis of all possible learner's answer that indicates “an object that takes fat-hah in the accusative case because it is broken plural.”

- ○ **مفعول به** **منصوب** وعلامة نصبه **الفتحة** لأنه **جمع تكثير**
- ○ **مفعول به** **منصوب** وعلامة نصبه **الفتحة** وهو **جمع تكثير**
- ○ **مفعول به** **منصوب** **بالفتحة** لأنه **جمع تكثير**.

This will be parsed into the abstract representation

Reason	'جمع تكثير' broken-plural
◌ٍ◌ِ◌ِA◌ِnalytic sign	'فتحة' fat-hah
End case	'منصوب' accusative
analytic location	'مفعول به' object

To show how the linguistic analysis is produced, consider the following question as an example.

أعرب ما بين القوسين في الجملة الآتية:

- ▪ ألف العقاد (كتبا كثيرة).

Give the linguistic analysis of the words between brackets in the following sentence:

- ▪ Al-aakad authored (many books).

This sentence is a simple verbal sentence having the following structure:

```
simple_verbal_sentence ::= verb   subject
object adjective
```

The following is the abstract representation of the linguistic analysis of the words between brackets:

Books ‘كتبا’	Reason	'جمع تكثير' broken-plural
	◌ٍ◌ِ◌ِA◌ِnalytic sign	'فتحة' fat-hah
	End case	'منصوب' accusative
	analytic location	'مفعول به' object
Many ‘كثيرة’	Reason	'مفرد' singular
	◌ٍ◌ِ◌ِA◌ِnalytic sign	'فتحة' fat-hah
	End case	'منصوب' accusative
	analytic location	'نعت' adjective

The system takes the learner’s answer, applies the respective linguistic rules, and produces the feedback according to the following steps.

- Parse the words between brackets:
  - o Morphologically analyze the object and get its features
  - o Morphologically analyze the adjective and get its features
  - o Check agreement between object and adjective
- Generate the linguistic analysis of the words between brackets:
  - o Generate the quadruple abstract representation form
- Generate response to the learner:
  - o Convert the learner’s answer into a quadruple abstract representation form
  - o Compare the generated representation with the learner representation
  - o Send feedback to the learner

The features that result from the morphological analysis of the words between brackets in the above example are as follows:

- كتبا (books): category:noun, gender:feminine, number:broken-plural, definiteness:indefinite,…
- كثيرة (many): category:noun, gender:feminine, number:singular, definiteness:indefinite, sub_category:adjective,…

The analytic location and reason of the words between brackets that results from the parsing the words between brackets are as follows:

- كتبا (books): analytic location: 'مفعول به ' (object) and reason: 'جمع تكثير' (broken plural)
- كثيرة (many): analytic location: 'نعت' (adjective) and reason: 'مفرد' (singular)

The end cases of the words between brackets that results from applying the rules shown in Figure 12 are as follows:

- كتبا (books): end case: 'منصوب' (accusative)

- كثيرة (many): end case: 'منصوب' (accusative)

The analytic signs of the words between brackets that results from applying the rules shown in Figure 13 are as follows:

- كتبا (books): analytic sign: 'فتحة' (fah-hah)
- كثيرة (many): analytic sign: 'فتحة'(fah-hah)

```
Input: syntactic category of a word
Output: end case of the word
Determine end case of a word Rule:
If object
Then end case = accusative
If inchoative
Then end case = nominative
If adjective
Then end case = end case of the noun it modifies
```

Fig. 12. A subset of rules for determining the end case of a word

```
Input: number and end case of a word
Output: analytic sign of the word
Determine analytic sign of a word Rule:
If number = dual and end case = nominative
Then analytic sign = Alef
If number = broken plural and end case = accusative
Then analytic sign = fat-hah
If number = singular and end case = accusative
Then analytic sign = fat-hah
```

Fig. 13. A subset of rules for determining the analytic sign of a word

The error analysis component of our intelligent computer assisted language learning system includes rules which are capable of parsing ill-formed input and which apply if the grammatical rules fail. As an example, consider the following question to complete a sentence with a suitable unrestricted object "المفعول المطلق":

أكمل بمفعول مطلق مناسب: أبر أبى_________.

Complete the following with the correct unrestricted object: I am kind to my father _________.

The grammar rule of the unrestricted object construct is augmented by an error analysis rule that checks the learner's answer against every possible ill-formed construction and produces the appropriate feedback, see Figure 14.

```
Input: unrestricted object
Output: success or failure of parsing. A feedback
message in case of failure
Error analysis of unrestricted object Rule:
ﻩf object.category ≠ noun
then feedback_message = ‘ المفعول المطلق يجب أن يكون
اسم’,
if object.origin ≠ infinitive
then feedback_message = ‘ المفعول المطلق يجب أن يكون
من مصدر الفعل’
if definiteness = definite
then feedback_message = ‘ المفعول المطلق يجب أن يكون
نكرة’
if need_alaf_tanween(object)
then feedback_message = ‘ المفعول المطلق يحتاج الي
ألف تنوين’
```

Fig. 14. An error analysis rule

The error analysis rule says that in order to accept an unrestricted object some constraints must be satisfied:

1. The input word must be a noun.
2. The input word must be originated from the infinitive verb.
3. The input word must be indefinite.
4. In some cases, the input word must take the end case "Alef Tanween".

If any of the above constraints is not satisfied, then the whole rule will fail and an error message reporting which type of error has occurred will be issued.

In Arabic ICALL [20], we addressed the challenges posed by ICALL in the Arabic language. We presented an architecture that consists of the following components: user interface, course material, sentence analysis, and feedback. The NLP components are the sentence analysis and feedback. There are major issues that we currently addressing and would appear in our future publications. One issue is to employ techniques, such as edit distance and constraint relaxation techniques, to identify the source of errors and generate error-specific feedback suited to the student expertise. We are concentrating on listening and writing skills of second language learners who usually commit errors that are difficult to be caught by spell checkers. Another issue is the developments of a complete computationally erroneous tagged Arabic corpus. This corpus would consist of representative amount of wrong learner's data that can be used to evaluate our proposed model instead of manually collect a test set from the real teaching environment. This data is considered a bottleneck in the development of Arabic ICALL systems.

V. CONCLUSION

We have presented the rule-based approach in Arabic natural language processing. One possible criticism of the rule-based approach is that it is a traditional and widely studied topic especially when it comes to European languages. However, given the status of the Arabic language technology nowadays, current research still marks a step towards helping Arabic language technology catch up with more mature language technology such as English.

Our experience shows that rapid development of rule-based systems is feasible, especially in the absence of linguistic resources and the difficulties faced in adapting tools from other languages due to peculiarities an the nature of Arabic language. Finally, our experience also points to the necessity of adopting general solutions as much as possible, as this increases the chances that the linguistic knowledge and tools can be used in other domains and systems as well.

REFERENCES

[1] A. Abdel Monem, K. Shaalan, A. Rafea, H. Baraka. Generating Arabic Text in Multilingual Speech-to-Speech Machine Translation Framework, *Machine Translation, Springer, Netherlands*, 20(4): 205-258, December 2008.

[2] I. Al-Sughaiyer, I. Al-Kharashi. Arabic morphological analysis techniques: A comprehensive survey. *Journal of The American Society for Information Science and Technology (JASIST)*, John Wiley & Sons, Inc., NJ, USA,55, 3, 189-213, 2004.

[3] M. Attia, *A large scale computational processor of Arabic morphology and applications*. MS Dissertation, Computer Engineering, Cairo University, Egypt, 1999.

[4] M. Attia. *Handling Arabic Morphological and Syntactic Ambiguities within the LFG Framework with a View to Machine Translation*. PhD Thesis, University of Manchester, Manchester, UK, 2008.

[5] K. Beesley. Finite-state morphological analysis of Arabic at Xerox Research: Status and plans in 2001. *In the proceedings of the workshop on Arabic natural language processing*, 39th Annual Meeting of the Association for Computational Linguistics (ACL), Toulouse, France, 1-8, 2001.

[6] Buckwalter, T. Issues in Arabic Orthography and Morphology Analysis. In the Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages (CAASL), COLING 2004, Geneva, Switzerland, August 28, 31-34, 2004.

[7] V. Cavalli-sforza, A. Soudi. Arabic Morphology Generation Using a Concatenative Strategy*, In Proceedings of the First Conference of the North-American Chapter of the Association for Computational Linguistics (NACCL)*, Seattle, WA, USA, 86-93,2000.

[8] Farghaly, A. Three Level Morphology for Arabic, presented at the Arabic Morphology Workshop, Linguistics Summer Institute, Stanford, CA, 1987.

[9] A. Farghaly, K. Shaalan. Arabic Natural Language Processing: Challenges and Solutions, *ACM Transactions on Asian Language Information Processing (TALIP)*, the Association for Computing Machinery (ACM). TALIP Vol 8, Issue 4, December 2009.

[10] J. Mace. *Arabic Grammar: A Reference Guide. Edinburgh University Press*, Edinburgh, UK, 1998.

[11] M. Magdy, K. Shaalan, A. Fahmy. Lexical Error Diagnosis for Second Language Learners of Arabic. *In the Proceedings of The Seventh Conference on Language Engineering, Egyptian Society of Language Engineering (ELSE)*, Cairo, Egypt, 5-6 December 2007.

[12] E. Othman, K. Shaalan, and A. Rafea. A Chart Parser for Analyzing Modern Standard Arabic Sentence, *In proceedings of the MT Summit IX Workshop on Machine Translation for Semitic Languages: Issues and Approaches*, New Orleans, Louisiana, USA., September, 2003.

[13] E. Othman, K. Shaalan, and A. Rafea, Towards Resolving Ambiguity in Understanding Arabic Sentence, *In the Proceedings of the International Conference on Arabic Language Resources and Tools, NEMLAR,* PP. 118-122, 22nd–23rd Sept., Egypt, 2004.

[14] A. Rafea, K. Shaalan K. Lexical Analysis of Inflected Arabic words using Exhaustive Search of an Augmented Transition Network, *Software Practice and Experience*, John Wiley & sons Ltd., UK,23(6):567-588, June 1993.

[15] K. Ryding. *Reference Grammar of Modern Standard Arabic*, Cambridge University Press, Cambridge, UK, 2005.

[16] K. Shaalan, A. Farouk, A. Rafea. Towards An Arabic Parser for Modern Scientific Text, *In Proceeding of the International Conference on Artificial Intelligence and computational Intelligence for Decision, Control and Automation in Engineering and Industrial Applications (ACIDCA'2000)*, PP. 228-235, 22-24 March, Monastir, Tunisia, 2000.

[17] K. Shaalan, A. Allam, A. Gomah. Towards Automatic Spell Checking for Arabic, *In Proceedings of the 4th Conference on Language Engineering, Egyptian Society of Language Engineering (ELSE)*, PP. 240-247, Egypt, Oct. 21-22, 2003.

[18] Shaalan K., Rafea, A., Abdel Monem, A., Baraka, H., Machine Translation of English Noun Phrases into Arabic, *The International Journal of Computer Processing of Oriental Languages (IJCPOL)*, World Scientific Publishing Company, 17(2):121-134, 2004.

[19] K. Shaalan. Arabic GramCheck: A grammar checker for Arabic. *Software Practice and Experience*, 35(7):643—665, 2005.

[20] K. Shaalan. An Intelligent Computer Assisted Language Learning System for Arabic Learners*, Computer Assisted Language Learning: An International Journal*, Taylor & Francis Group Ltd., 18(1 & 2): 81-108, February 2005.

[21] K. Shaalan, A. Monem, A. Rafea., Arabic Morphological Generation from Interlingua: A Rule-based Approach, *in IFIP International Federation for Information Processing*, *Intelligent Information Processing III,* Vol. 228, eds. Z. Shi, Shimohara K., Feng D., (Boston:Springer), PP. 441-451, 2006. (Also appeared in the Proceedings of the 4th International Conference on Intelligent Information Processing, Adelaide, Australia, September 20-23, 2006).

[22] K. Shaalan, A. Abdel Monem, A. Rafea, H. Baraka. Mapping Interlingua representations to feature structures of Arabic sentences, *The challenge of Arabic for NLP/MT international conference*, the British Computer Society (BCS), London, UK pp. 149-159, 23 October, 2006.

[23] K. Shaalan, H. Raza. Person Name Entity Recognition for Arabic, *In the Proceedings of the ACL 2007 Workshop on Computational Approaches to Semitic Languages*: Common Issues and Resources, Association for Computational Linguistics, PP. 17–24, Prague, Czech Republic, 28-29th June, 2007.

[24] K. Shaalan, H. Abo Bakr, I. Ziedan. Transferring Egyptian Colloquial into Modern Standard Arabic. *In the proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP – 2007b)*, Borovets, Bulgaria, 525-529, September 27-29, 2007.

[25] K. Shaalan, H. Raza H., Arabic Named Entity Recognition from Diverse Text Types, In Eds. Nordström, B., and Ranta, A., (Eds.) GoTAL 2008: 6th International Conference on Natural Language Processing, Gothenburg, Sweden, August 25-27, *Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence (LNCS/LNAI): Advances in Natural Language Proceedings*, Vol. 5221, PP. 440-451, Springer-Verlag, Berlin, Germany, 2008.

[26] K. Shaalan, H. Raza. NERA: Named Entity Recognition for Arabic, *The Journal of the American Society for Information Science and Technology (JASIST)*, John Wiley & Sons, Inc., NJ, USA, 60(8): 1652–1663, July 2009.

[27] K. Shaalan, A. Abdel Monem, A. Rafea, H. Baraka. Syntactic Generation of Arabic in Interlingua-based Machine Translation Framework, *Third workshop on Computational Approaches to Arabic Script-based Languages (CAASL3), Machine Translation Summit XII,* Ottawa, Ontario, Canada, August 26, 2009.

[28] Soudi A., Bosch, A., Neumann, G. *Arabic Computational Morphology: Knowledge-based and Empirical Methods, Text and Language Technology*, Vol 38. Springer, New York, 2007.

[29] Trujillo A., *Translation Engines: Techniques for Machine Translation*, Springer Verlag, 1999.

[30] Woods, W. Transition network grammar for natural language analysis, *Communication of the ACM*, Vol. 10, pp. 591-66, 1970.

Khaled Shaalan is a tenured Professor in Computer Science at the Faculty of Computers & Information (FCI), Cairo University, Egypt. He is on secondment to The British University in Dubai, the first post-graduate, research-based university in the UAE. He is also an Honorary Fellow at the School of Informatics at the University of Edinburgh, UK. He holds a PhD from Cairo University jointly with the Swedish Institute of Computer Science.

His major research interests include computational linguistics; in particular, Arabic natural language processing. This field has become increasingly important, as more and more textual information is now available at homes and businesses through the Web, Internet and Intranet services. Khaled has long standing expertise in Arabic natural language processing tools. He has been involved in developing a set of tools at different analysis and generation levels: morphology, syntax, and (shallow) semantics. Some of these tools address, to some extent, another dimension of the three Arabic language varieties: Classical, Modern Standard, and (Egyptian) dialect. Whereas, some others address the ill-formed learner's input and give detailed analysis of the erroneous input that detects, diagnoses, and generates remedial feedback explaining the source of learner's errors.