Arabic Generation in the Framework of the Universal Networking Language

Daoud Maher Daoud

ALzaytoonah University, Amman, Jordan
daoud_m@yahoo.com , daoud@maherinfo.com
&
GETA, CLIPS, IMAG
daoud.daoud@imag.fr

Abstract. This paper describes the work done on the developing of Arabic Deconversion within the framework of the Universal Networking Language (UNL). In this paper, the architecture of the system is explained along with the strategy used for the development. We also discuss issues and problems related to the UNL representation that affect the quality of generation. Additionally, the lingware engineering is introduced as a technique to enhance the quality and increase the development efficiency.

1 Introduction

Arabic is one of the world's main languages. It is the official language for over 289 million people. It is also the sacred language of nearly 1.48 billion Muslims throughout the world.

The alphabet consists of twenty-eight consonants but three of these are used as long vowels. Arabic also contains short vowel signs being indicated by marks above or below the letters. Like other Semitic languages, Arabic is written from right to left. It is a language characterized by rich morphology: most of the words are built from consonantal roots in which inflections and derivations are generated by vowel changes, insertions, and deletions.

The Universal Networking Language is a specification for the exchange of information. It is a formal language for symbolizing the sense of natural language sentences.

Currently, the UNL includes 16 languages. These include the six official languages of the United Nations (Arabic, Chinese, English, French, Russian and Spanish), in addition to ten other widely spoken languages (German, Hindi, Italian, Indonesian, Japanese, Latvian, Mongol, Portuguese, Swahili and Thai). In its second phase (1999–2005) the project will seek to further extend UNL access.

This paper presents the work completed on the generation of Arabic from UNL during the author's employment with Royal Scientific Society (RSS) in Jordan and his work on the UNL project. It described the work done on the generation of Arabic from UNL between 1996 till 1999. Since then, we think that the generation system maintained its main architecture.

Universal Network Language: Advances in Theory and Applications.
Research on Computing Science 12, 2005, pp. 195–209.

Section 2 gives a description of the system that generates Arabic sentences from UNL representations. In section 3, we show how the system in implemented. In this perspective, issues such as mapping of relations, word ordering, and morphological generation are addressed. Results, future works, and conclusions are presented in Sections 4, 5 and 6 respectively.

2 The Architecture of Arabic Generation (Deconversion) System

Universal networking language (UNL) is a semantic, language independent representation of a sentence that mediates between the enconversion (analysis) and deconversion (generation). It is a computer language aiming at removing language barriers from the Internet. The pivot paradigm is used: the representation of an utterance in the UNL interlingua is a hypergraph where normal nodes bear UWs ("Universal Words", or interlingual acceptions) with semantic attributes, and arcs bear semantic relations [13].

The sentence "Khaled bought a new car" can be expressed in UNL as:

agt*(buy(icl>do(obj>thing),icl>purchase).@past.@entry, Khaled)*
obj*(buy(icl>do(obj>thing),icl>purchase).@past.@entry, car(icl>automobile))*
mod(car(icl>automobile),new)

Figure 1 shows the graph representation of this UNL expression. The node represents the Universal Word (UW). Arcs represent binary relations such as "agt", "obj" and "mod". Attributes are attached to UW to include information about time, aspect, number, modality, etc. In the previous sentence, the attribute "@past" was attached to the event "buy" to indicate that the event happened in the past. The "@entry" attribute is used to indicate the entry point or main node for the whole expression.

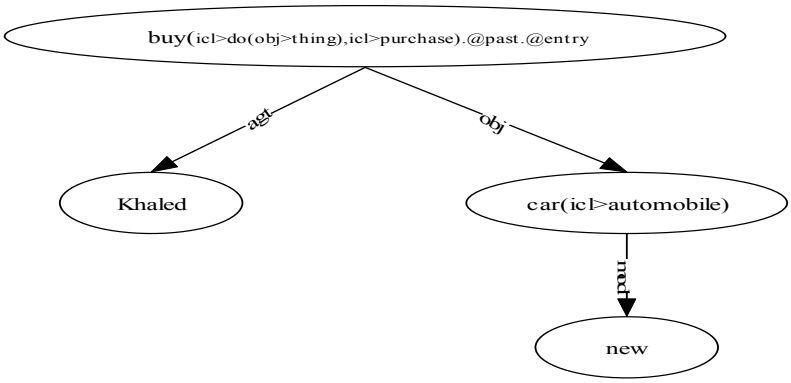

Fig. 1. The graph of the UNL expression

Generation of the target sentence is the process of converting the UNL expression or the hyper-graph to one dimensional character string. We use the DeCo tool, which is provided by UNDL foundation to work the Arabic Deconversion. On the other hand, enconversion is the process of generating UNL from a natural language text. A soft-

ware for enconversion called "EnCo" which constitutes an enconverter together with a word dictionary, UNL knowledge base, and conversion rules for a language [4].

2.1 Deconverter

DeCo is a language independent system capable of traversing any UNL graph and constructing morphemes based on each node visited. As shown on Figure 2 the main inputs of the Deco tool are a UNL expression, a dictionary, and generation rules. First, DeCor transforms the sentence represented by an UNL expression - that is, a set of binary relations - into the directed hyper graph structure called **Node-net**. Then, it applies generation rules to every node in the node-net respectively, and generates the word list in the target language (Node-list) [3].

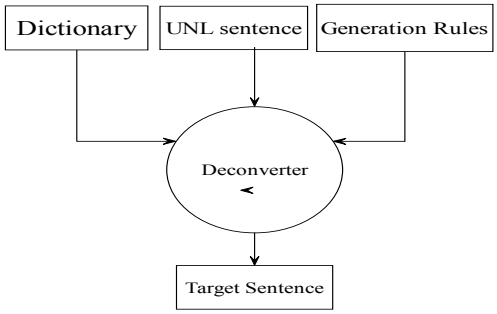

Fig. 2. The deconversion Process

The order of traversal is specified by the generation rules, which also systematize the preference of the target lexis.

The DeCo engine employs the generation rules to map the UNL expression into the appropriate syntactic and morphological structure of the target sentence.

The DeCo tool implements an abstract transducer model with multi-heads (or windows). The DeCo tool uses two types of windows: Generation Window (GW) and Condition Window (CW). There are 2 GWs bordered from both sides by several CWs (Figure 3).

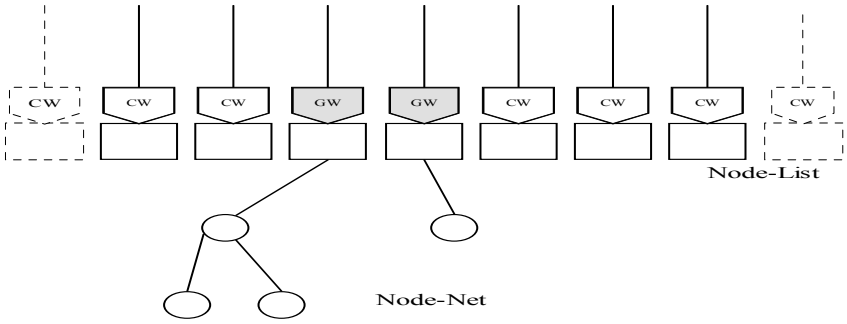

Fig. 3. The conceptual model of the DeCo tool

The Condition Windows are used to test out the conditions in the neighboring nodes. Alternatively, the Generation Window is able to test a condition, modify attributes, and insert nodes from the graph to the Node-List.

2.2 The Generation Rules

A generation rule is a finite collection of instructions, each calling for a certain operation to be performed if certain conditions are met.
Every rule is of the form:

<TYPE>
["("<PRE>")" ["*"]]...
"{" | """ [<COND1>] ":" [<ACTION1>] ":" [<RELATION1>] ":" [<ROLE1>]
"}" | """
["("<MID>")" ["*"]]...
"{" | """ [<COND2>] ":" [<ACTION2>] ":" [<RELATION2>] ":" [<ROLE2>]
"}" | """
["("<SUF>")" ["*"]]...
"P(" <PRIORITY> ");"

As an example of inserting a new node to the right, the following rule layout is used:

:{<COND1>:<ACTION1>:<RELATION1>:<ROLE1>}
"<COND2>:<ACTION2>:<RELATION2>:<ROLE2>"

When a node in the node-list satisfies the conditions expressed in <COND1>, AND a node in the node-net which is linked to by the relation of <RELATION1> or <RELATION2> is found, AND IF the node satisfies the conditions expressed in <COND2>, THEN the system inserts a new node to the right of node in the node-list, and executes <ACTION1> AND <ACTION2> [3].

Structuring the corresponding target sentence (node-list) is directed by the rules (Figure 4).

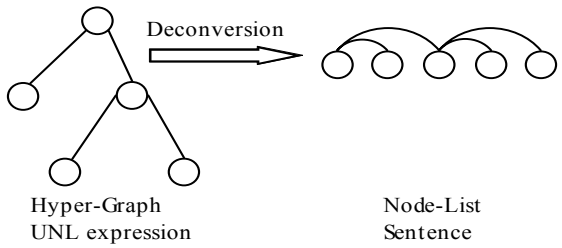

Fig. 4. Building Node-list.

Rules specify conditions and possibly semantic relations needed to trigger actions. Conditions concern the lexical, semantic, and morphological attributes of the node under processing which are specified in the dictionary and/or through the conversion process. Semantic relations are the relations linking two nodes such as agt, obj, etc.

Actions (such as insertion of a node in the node-list) are executed when conditions are met.

For example the following rule:
:{V, >obj: ->obj, +obj_ad, +RBL::} " N,^#N, <obj, ^@pl, ^PLUR:-<obj, +is_obj,+ACC:obj:"P110;

Inserts a node from the UNL graph into the node-list right of the verb node, which is already in the node list. The inserted node is an object of the verb. Besides the insertion action, the rule modifies certain attributes to be used by other rules to add morphological features to the generated word.

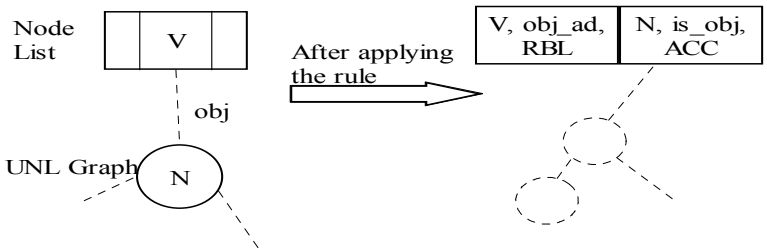

Fig. 5. Applying the rule

Figure 5 shows this process. The insertion of the node is followed by changing attributes: obj_ad (object is added to the verb) and RBL (to add a blank right of the verb) are added to the verb node, is_obj (to mark that this node is an object) and ACC (to mark that the case of this noun is accusative according to Arabic grammar) are added to the object node.

2.3 The Dictionary

Dictionary stores word entities for each language. The data format of each entity consists of three main components: Head Word (HW) of each local language, Universal Word (UW) and Grammatical Attributes of HW.

Attributes are used by the generation rules to control the selection of the target word in addition to the surface form of the target sentence.

Although Arabic has many inflectional and derivational forms which increase the need to do morphological synthesis rather than full-form listing during the generation process, we preferred full-form listing. This comes from the fact that the DeCo tool lacks the functions to perform infixation.

The generation of lexical entries, (Head word) is based on syntactic and morphological features of each lemma. As a result, each UW can be mapped to different forms that share the same meaning.

For example the UW "sell(ant>buy, icl>event)" is mapped to the HW "باع" [baa'] which is weak middle radical (hollow) verb. As shown in figure 6, different forms are added to dictionary with inflection for gender (masculine, feminine), number (singular, plural, dual), person (first, second, third) , tense (past, present, future), mood (indicative, subjunctive, jussive) and voice.(passive, active) .

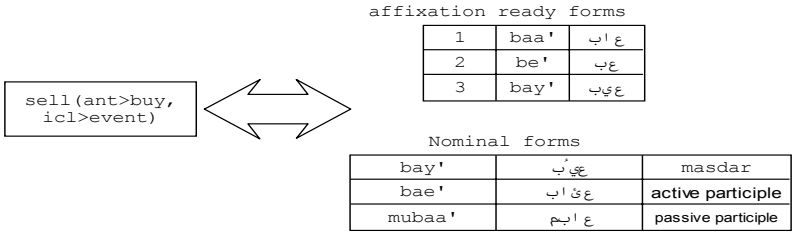

Fig. 6. Entries for verb baa'

In addition to the "affixation ready forms", “nominal forms" are also linked to the same UW which is derived directly from the verb. The nominal inflection of verbs is used to generate accurate sentences. Grammatically these forms act as nouns or adjectives.

inflections	Selected form	Prefix	suffix
masculine, singular, first, perfect, indicative, active	2	-	ت
masculine, plural, third, imperfect, indicative, active	3	ي	ون
feminine,plural, third, imperfect, indicative, passive	1	ت	-

Fig. 7. Examples of inflection and selected forms

The generation rules should select the right form and add the necessary prefixes and suffixes (figure 7).

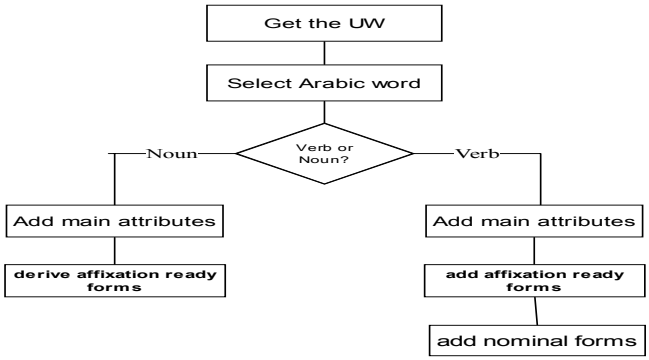

Fig. 8. adding entries for verb and nouns

This approach proved to be feasible also for handling nouns. The linguistic attributes of nouns that have been used in the dictionary are basically: gender, number, case and definiteness.

The issue of broken plural is solved by linking this form to the same UW. Finally the variations in the written forms (such as hamza and nouns ending with long vowel) of Arabic is also handled by making entry for each of these forms in the dictionary.

A database system has been developed for the classification and features adding for each entry in the dictionary. As shown in figure 8 the system gets the UW and tries to get the equivalent Arabic word from an English-Arabic dictionary. The selected Arabic word is then classified to Noun or Verb or Particle. As an example : If the word is denoted as having a broken plural, the system will ask the user to add this entry and both forms are linked to the same UW.

3 The Deconversion of Arabic

Deconversion is the process of producing a grammatically correct sentence from the UNL graph. This process involves mapping of relations, Lexical transfer, word ordering, and morphological generations.

3.1 Mapping of relations

Each UNL relation has been mapped to the corresponding Arabic grammar structure or syntactic relation. It is not a one-to-one mapping as one relation can be mapped to different target grammatical relations. As an example, the "obj" relation can acquire the syntactic role of subject or object or to Idafa construct depending in the UWs involved and the adjacent relations in the UNL graph.

As an example the sentence:

1- The mouse died

Could have the following UNL expression:

obj(die(icl>event).@past.@entry, mouse)

In this sentence, the mouse is not responsible for the event and it undergoes a change of state, so semantically it is the object of the verb die. However, when the sentence is deconverted into Arabic the mouse is the grammatical subject and should have its inflections (nominative).

2- The flour becomes bread.
obj(become(icl>event).@entry, flour)
gol(become(icl>event).@entry, bread)

Since flour experiences a change of state, it is the semantic object of the verb become. Flour is the final state then it is linked with become by the gol relation. Syntactically flour is the nominative actor and bread is the accusative object.

3- He told me a joke.
agt(tell(icl>event).@past.@entry, he)
ben(tell(icl>event).@past.@entry, me)
obj(tell(icl>event).@past.@entry, joke)

In this sentence: *he* is the nominative actor, *me* is the first accusative object and *joke* is the second accusative object.

4- Khaled was killed.
obj(kill(icl>event).@past.@entry, Khaled)

The verb kill is transitive and the agent is deleted from the verb argument. Khaled who is the object of the killing takes the role of "substitute of the doer of the verb" in the generated Arabic sentence. The verb kill is inflected by the passive voice form.

5- Khaled appreciated Ali's learning of French.
agt(appreciate (icl<event).@past.@entry, Khaled)
obj(appreciate(icl<event).@past.@entry, learn(icl<event))
agt(learn(icl<event), Ali)
obj(learn(icl<event), French)

The IDAFA construction is an important grammatical structure in Arabic. It is a genitive construction in which two nouns are linked up in such a way that the second (second particle of the construction) qualifies or specifies the application of the first (first particle of the construction). The usage that concerns us here involves the nominalization of processes, in which the first prticle is typically a *masdar* representing a nominalized process, and the second particle represents either the 'agent' or 'object' of that process. When the above UNL expression is deconverted to Arabic, "learn" becomes the accusative object of the verb "appreciate" which is converted to the *masdar* form. The *masdar* form of the verb "learns" is also the first particle of the IDAFA construction. The second particle is "Ali" who is the actual actor of the verb "learn" but in the genitive case. "French" is the accusative object of the *masdar* as shown in Figure 9.

The relation mapping is implemented in the deconversion rules. The following rule shows how the relation "obj" is mapped. The inserted node becomes the object and marked by "is_obj" attribute and takes the accusative case "ACC".
:{V,>obj:->obj, +obj_ad, +RBL::} "NDE,^MASDAR,^#N,<obj, ^@pl, ^PLUR: -<obj,+is_obj,+ACC:obj:" P110;

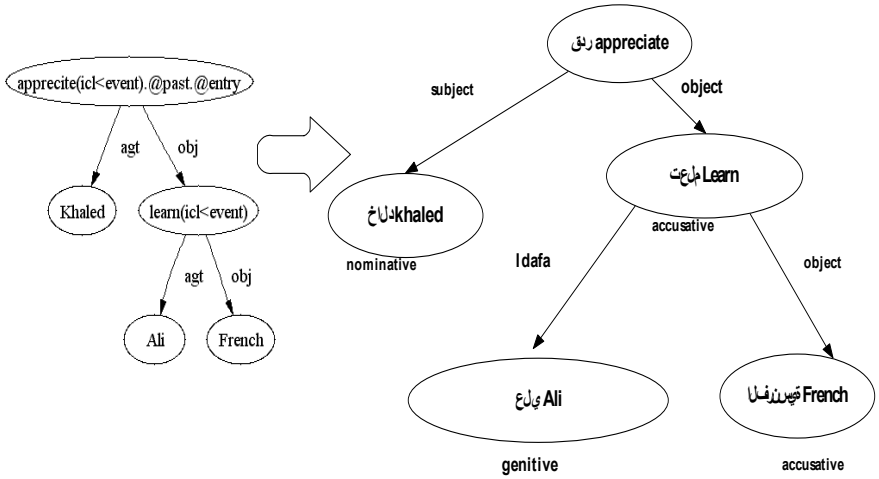

Fig. 9. Mapping of relations

3.2 Word Ordering

Although Arabic shows a flexible word order. It can be said that the dominant or preferred ordering is VSO, the subject and object follow the verb. We also find that specifiers, adjectives, genitives, and relative clauses usually follow the nouns they modify, that adverbs and adjectival arguments usually follow the adjectives they modify, and that noun phrases usually follow the prepositions that govern them. In other words, with very few exceptions, modifiers and arguments usually follow the words they modify or what govern them.

Figure 10 shows the process of insertion and the direction of the UNL graph (5). This is governed by the deconversion rules during the insertion of a new node from the graph to the node-list.

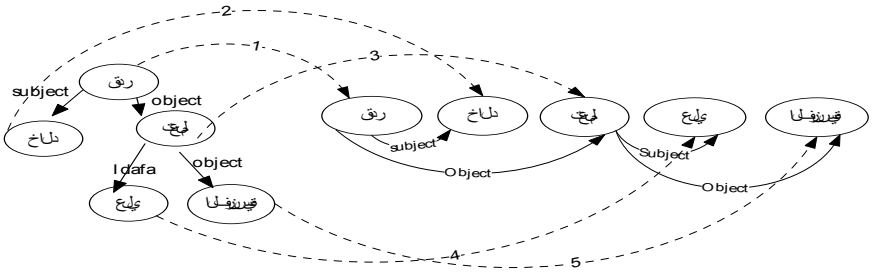

Fig. 10. Insertion sequence and direction

3.3 Morphological Generation

Arabic is a Semitic language, and its basic characteristic is the rich morphology in which most of its words are derived from roots. Inflections and derivations are generated by changing vowels and insertion of consonants.

Arabic sentences are characterized by a strong tendency for agreement between its constituents, between verb and noun, noun and objective, in matters of numbers, gender, definitiveness, case, person etc. These properties are expressed by a comprehensive system of affixation. To satisfy these grammatical properties, generation rules are expected to be complex, to handle the processing of generating grammatically correct Arabic sentences from UNL expression and structure.

In our system, we managed to handle this rich and complicated morphology by implementing a modular approach to coding the rules (figure11).

Our implemented process of morphological generation starts by choosing the right stem which is set to accept prefixes or suffixes depending on its position and role in the sentence.

As an example, both rules below insert a plural subject to a verb already in the Node-list. If the corresponding Arabic word has a regular plural form (by adding the right suffix), then the first rule is executed. Otherwise, the system looks for the other form (broken plural) in the dictionary, and the second rule must be triggered.

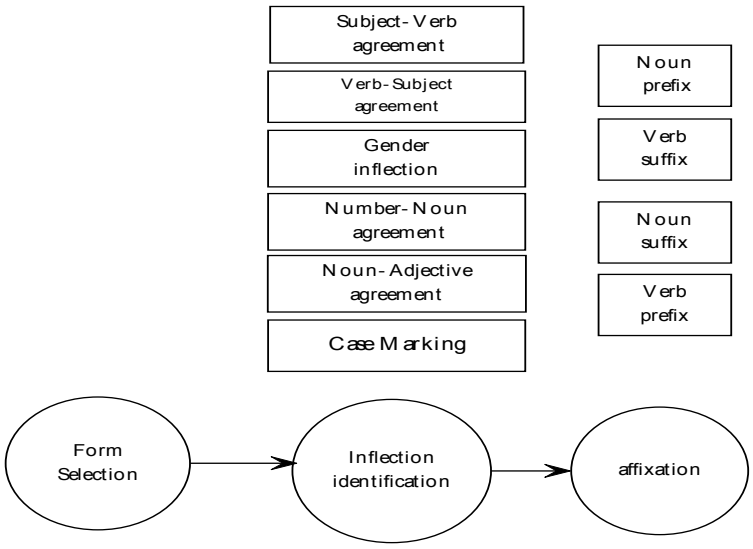

Fig. 11. Morhological generation

Rule 1
:{V,>agt:->agt,RBL,has_subj, V_subj}"^#N,N,@pl,^IRGPL,<agt:-
<agt,is_subj,NOM:agt"P100;

Rule 2
:{V,>agt:->agt,RBL,has_subj, V_subj}"^#N,@pl,PLUR,<agt:-
<agt,is_subj,NOM:agt"P100;

This approach of word selection is dependent on the syntactic conditions at the time of insertion. However if new facts or conditions become true later, which the selected form does not comply with, the system should backtrack and select a new form. A good example for this observable fact is the implementation of the number-noun agreement that is controlled by complicated set of rules in Arabic. As an illustration, in numbers above ten the noun must be singular, indefinite and accusative and the number takes the grammatical role of the noun. The difficulty becomes very apparent if this noun is attributed by "@pl" in the original UNL expression.

The second phase is marked by identifying types of inflections required to generate quality Arabic sentence such as agreements.

All types of agreements are implemented in our system. For example, Arabic has incomplete agreement in verb-subject sentences. In this case, the agreement will be in the gender but not in the number. Rule 3 (figure 12) shows the implementation of verb-subject agreement. When this rule is executed the verb is marked by the attribute "male_infl". This information is then passed to other rules to add the necessary suffix depending on the type of verb as shown in the rules (4-5) listed below.

The last phase of morphological generation is implemented by prefixing and suffixing rules to mark the inflections identified in the previous process.

```
Rule 3
:{V,has_subj:male_infl,V_SUB_GenderAgr}{is_subj,male,^V_SUB_GenderAgr:V_SUB_GenderAgr} P200;

Rule4 :{V,V_subj,QDAA,3person_infl,^add_past,^@not,^suff,male_infl:suff} "[ي]"P2;

Rule 5
:{V,V_subj,SAA,3person_infl,^add_past,^@not,^suff,male_infl:suff} "[ا]"P2;

rule 6
:{V,V_subj,DAA,3person_infl,^add_past,^@not,^suff,male_infl:suff} "[ا]"P2;

Rule 7
:{V,V_subj,QDAA,3person_infl,add_past,^suff,male_infl:suff} "[ا]"P2;
```

Fig. 12. Verb-subject agreement rules

In our system, three main groups of rule are designed: insertion rules, inflection identification rules and affixation rules.

4 Results

During the development period of Arabic Module the number of lexical items added to UNL-Arabic dictionary reached 120,000 entries. This covers the UWs provided by UNL center and the most frequent Arabic lexicon. More sophisticated features are added to each entry to cover morphological, syntactic and semantics aspects. In designing those features, we took into consideration the analysis and generation processes.

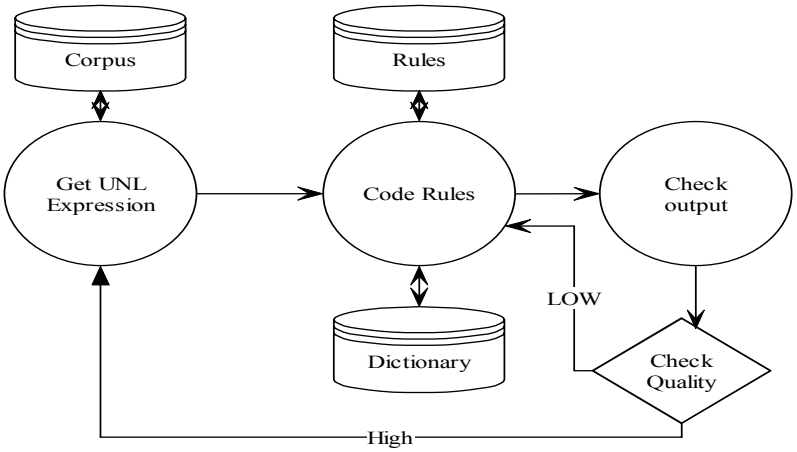

Fig. 13. Current methodology of coding generation rule

Arabic Corpus has been built for Soccer and other topics in order to specify accurately the word usage and to extract the most frequent Arabic words. Functional words are also added to the dictionary along with all prefixes and suffixes needed for Arabic morphology.

The Arabic Deconversion system managed to handle the following situation and sentences:

- Agreement and Morphological generation
- Scope
- All type of relations and attributes
- Loop structure
- Embedded and relative sentences
- Nominal and verbal sentences

However, a variety of problems emerged during the implementation of our system. Some of these problems are related to the nature of UNL others are due to wrong UNL representations of the source language. While Other problems are caused by the dictionary and the limitation of the DeCo tool.

In reality, UNL expressions are not always language-independent. They are influenced by the source language. In the same context, using a separate UWs and relations rather than using attributes to describe subjectivity of the sentences is also a demonstration of the source language influence. For example:

A possible UNL expression of "People no longer have to go." is:
agt(go(icl<event).@obligation.@entry, people)
man(go(icl<event).@obligation.@entry, no longer)
The generated Arabic sentence is not acceptable from the above UNL expression. In contrast, the following UNL expression has produced a good result:
agt(go(icl<event). @obligation-not.@entry, people)

Additionally, multiple identical relations connected to the same UW are problematic since the DeCo tool is incapable of the right word ordering. As examples: "honest (aoj) Jordanian (aoj) citizen" or "I prefer orange (obj) over apple (obj)".

As for dictionary related problems, they are mainly caused by using unrestricted UWs. Leading to imprecise selection of Arabic corresponding words. Besides, the Arabic compound words that correspond to UWs in the dictionary are also a significant cause of low quality generated sentence.

In most cases, the quality of Arabic is highly dependent on the UNL expressions. The need for common consensus and standards among the producers of UNL expressions is important. A grammatically appropriate input sentence is a prerequisite for parsing. Likewise, generation requires correct UNL expressions to produce satisfactory results.

More than 2000 rules have been written to generate the Arabic language.

Figure 13 shows the current methodology of coding the generation rules. It is an evolutionary process, which demands many activities: writing rules, testing, and validation of rules, maintenance of rules, updating, and maintenance of the dictionary. Controlling these activities requires many resources, is very time consuming and the results is not always accurate.

5 Future Work

DeCo is a true SLLPS (specialized languages for linguistic programming), but still of quite "low level" in the hierarchy of programming models. The iterative methodology of writing rules shown in Figure 13 proved to be inadequate. In this methodology of development, the rules are written for each sentence in the corpus and there is no guarantee that any modification will not have harmful side effects. Therefore, a systematic development methodology is necessary to transform the natural language representation of a sentence into rules based representation. The main function of this methodology is to specify diagrammatically the language grammar using language components, which are entities that embed syntactic and semantic information that can be identified in the source language from its unique function in the sentence. These diagrams can be integrated into one development environment enabling systematic development of rules and ease of maintenance.

A Computer Aided rules Engineering (CARE) is needed at this point. It is an integrated environment, which provides a set of tools for the production and maintenance of the rules and dictionary.

Basically the proposed system consists of the following main components:

- Language Modeling
- Repository
- Rules Generator
- Dictionary Maintenance

Language modeling module facilitates the description and representation of linguistic knowledge using language components. This module is capable to describe natural language structure. This description should specify the words ordering, relationships, and dependencies among the constituents of the sentence. Additionally, it provides the proper description of UNL and the structure for mapping it to Arabic.

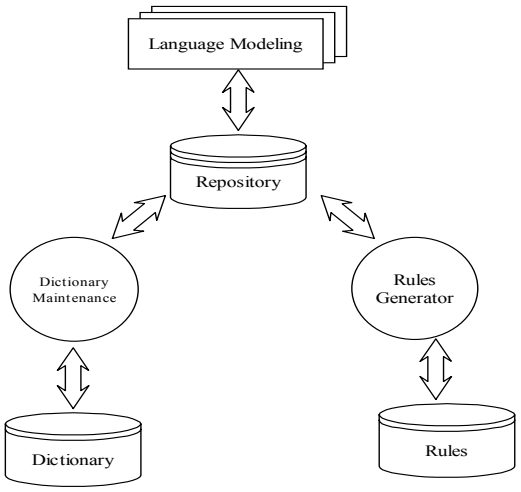

Fig. 14. Basic structure of CARE

As shown in figure 14, linguistics knowledge, UNL, and mapping rules are stored in the repository.

The repository is then interfaced with rules generation component that will facilitate the automatic production of the DeCo rules.

The repository is also interfaced with the dictionary to enable handling and maintenance of the dictionary.

6 Conclusion

In this paper, we have described the development of our first version of an UNL-Arabic Deconversion system. All information for the generation of Arabic from UNL has been addressed in all levels (i.e. morphological and syntactic). We also presented some complexities and issues related to the generation of Arabic. We have tried to introduce some systematicity in using the available DeCo tool, in order to compensate for its lack of high level programming constructs and modularity features. Our future work will concentrate on the development of an adequate CARE environment.

References

1. Uchida, H. (1989). “ATLAS-II: A machine translation system using conceptual structure as an Interlingua”. Proceedings of the Second Machine Translation Summit. Tokyo, Japan.
2. UNL center UNDL Foundation (2003). “The Universal Networking Language Specifications”. http://www.undl.org.
3. UNU/IAS (1999). "DeConverter Specifications. UNU/IAS UNL Center". www.undl. org.
4. Uchida H., Zhu M. (2001), "The Universal Networking Language Beyond Machine Translation". http://www.undl.org.
5. Soudi, A., Cavalli-Sforza, V., & Jamari, A., "Prototype English-to-Arabic Interlingua-based MT System," Proceedings of the Workshop on Arabic Language Resources and Evaluation - Status and Prospects, 3rd International Conference on Language Resources and Evaluation (LREC 2002), Jun 1, 2002, Las Palmas de Canaria, Spain.
6. Abuleil, S. and Evens M. (1998). Discovering Lexical Information by Tagging Arabic Newspaper Text., Workshop on Semitic Language Processing. COLING-ACL.98,
7. G. Sérasset and C. Boitet (1999), “Unl-french deconversion as transfer and generation from an interlingua with possible quality enhancement through offline human interaction,” in MT Summit VII, J.-I. Tsujii, Ed. Singapore: Asia Pacific Ass. For MT, pp. 220–228.
8. G. Sérasset and E. Blanc (2003) , “Remaining issues that could prevent UNL to be accepted as a standard,” in Convergences 3, Library of Alexandria, Egypt.
9. Boguslavsky I. (2001a). UNL from the linguistic point of view. Proceedings of the First International Workshop on MultiMedia Annotation. Electrotechnical Laboratory SigMatics, Tokyo, 1-6.
10. Mel’čuk I. (1988), Dependency Syntax: Theory and Practice, State University of New York Press.

11. Beesley, K.R. (2001). Finite-State Morphological Analysis and Generation of Arabic at Xerox Re-search: Status and Plans in 2001, in Arabic Language Processing: Status and Prospects -39th Annual Meeting of the Association for Computational Linguistics, pp. 1–8.
12. Dichy, J. (2001),"On Lemmatization of the Arabic Entries of Multilingual Lexical Databases, in Arabic Language Processing: Status and Prospects - 39th Annual Meeting of the Association for Computational Linguistics, pp. 23–30.
13. G. Sérasset and C. Boitet (2000), "On UNL as the future "html of the linguistic content" & the reuse of existing NLP components in UNL-related applications with the example of a UNL-French Deconverter", COLING 2000.
14. Uchida H., Zhu M. (1993), "Interlingua for Multilingual Machine Translation", CICC.