Generating Arabic text: The Decoding Component in an Interlingual System for Man-Machine Communication in Natural Language[1].

Sameh Alansary[†‡] Sameh.alansary@bibalex.org
Magdy Nagi[††‡] magdy.nagi@bibalex.org
Noha Adly [††‡] noha.adly@bibalex.org

[‡] Bibliotheca Alexandrina, P.O. Box 138, 21526, El Shatby, Alexandria, Egypt.

[†] Department of Phonetics and Linguistics
Faculty of Arts
Alexandria University
El Shatby, Alexandria, Egypt.

[††] Computer and System Engineering Dept.
Faculty of Engineering
Alexandria University,
Egypt.

1. Abstract.

This paper describes the decoding part in an interlingual system for man-machine communication in natural language. It is based on the Universal Networking Language (UNL) framework. Given a semantic network that represents a relation between a number of concepts, this network can be decoded (or 'DeConverted' in UNL technical terms) back to any natural language. This depends on the existence of a dictionary and a grammar for the language to which this network is to be decoded. The role of the dictionary is to find the word in which a given concept is to be expressed. The role of the grammar is to arrange the nodes or the concepts of the network in a way that produces a syntactically well-formed sentence in the target language. The paper addresses the overall technical structure of both the grammar and the dictionary used in the DeConvesion process of the UNL network to generate Arabic text. In addition it will describe different challenges faced in making this mission feasible.

2. Introduction.

Natural language generation (NLG) is a subfield of Computational Linguistics that focuses on the generation of texts (spoken or written) in natural languages from some underlying non-linguistic representation of information, generally from databases or knowledge sources, i.e. from computer-internal representations of information. NLG is viewed as being amore difficult area to work in than language analysis. For language analysis, the linguist is given a set of data (strings of the language) to work with, while for language generation, the linguist has ideas and plans that need to be turned into language. So it is not accurate to view language generation as the reverse of the language analysis (Klavans 1997).

While everyone speaks a language, not everyone speaks it equally; there are substantial difference concerning its speed of learning, and its ease and success of use. How language works in our mind is still a mystery, and some researchers consider the construction of NLG system as a methodology for helping in unravel that mystery Zock et al (1998). Others consider NLG as an approach to envision a number of different purposes, including standardized and/or multi-lingual reports, summaries, machine translation, dialogue applications, and embedding in multi-media and hypertext environments (Paris (1991)). Consequently, the automated production of language is associated with a large number of highly diverse tasks whose appropriate orchestration in high quality poses a variety of theoretical and practical problems. Relevant issues include content selection, text organization, production of referring expressions, aggregation, lexicalization, and surface realization, as well as coordination with other media.

In NLG, the system needs to take decisions about how to put a concept into words, and of course this is different from the language understanding where as in natural language understanding the system needs to disambiguate the input sentence to produce the machine representation language, for Natural language generation (NLG) only concepts and ideas are available to work with, choices of words (lexical items) and syntactic structures are apart of decisions to be made in building a text. NLG requires many kinds of expertise: Knowledge of the domains (what to say relevant to the situation), knowledge of the language (lexicon, grammar, semantics) and strategic rhetorical knowledge (i.e. how to achieve communicative goals, text types, style). Moreover NLG requires engineering system to decompose, represent and coordinate the processing of all this information.

[1] This work was carried out within Ibrahim Shihata Arabic-UNL Center (ISAUC) hosted by Bibliotecha Alexandrian (BA) and managed by its affiliated research center, the International School of Information Science (ISIS). The project is funded by the Arab Fund for Economic and Social Development, Kuwait. ISAUC is playing a major role in designing and implementing the Arabic language tools to act as an active language center for Arabic.

Architecture of NLG system needs to include levels of planning and merging of information to enable the generation of text that looks natural and does not become repetitive. Typical types of information are (Reiter and Dale (2000)):

- **Content determination**: Determination of the salient features that are worth being said. Methods used in this stage are related to data mining.
- **Discourse planning**: Overall organization of the information to convey.
- **Sentence aggregation**: Merging of similar sentences to improve readability and naturalness. For example, the sentences "The next train is the Caledonian Express" and "The next train leaves Aberdeen at 10 am" can be aggregated to form "The next train, which leaves at 10 am, is the Caledonian express".
- **Lexicalization**: Putting words to the concepts.
- **Referring expression generation**: Linking words in the sentences by introducing pronouns and other types of means of reference.
- **Syntactic and morphological realization**: This stage is the inverse of parsing: given all the information collected above, syntactic and morphological rules are applied to produce the surface string.
- **Orthographic realization**: Matters like casing, punctuation, and formatting are resolved.

Different types of generation techniques can be classified into four main categories (Konrad (2004), Cahill et al. (1999), and Cole et al (1996)):

- **Canned text systems** constitute the simplest approach for single-sentence and multi-sentence text generation. They are trivial to create, but very inflexible and are very wasteful. This approach is used in the majority of software: the system simply prints a string of words without any change (error messages, warnings, letters, etc.).
- **Template systems,** the next level of sophistication, rely on the application of pre-defined templates or schemas and are able to support flexible alterations. The template approach is used mainly for multi-sentence generation, particularly in applications whose texts are fairly regular in structure.
- **Phrase-based systems** employ what can be seen as generalized templates. In such systems, a phrasal pattern is first selected to match the top level of the input, and then each part of the pattern is recursively expanded into a more specific phrasal pattern that matches some subportion of the input. At the sentence level, the phrases resemble phrase structure grammar rules and at the discourse level they play the role of text plans.
- **Feature-based systems,** which are as yet restricted to single-sentence generation, represent each possible minimal alternative of expression by a single feature. Accordingly, each sentence is specified by a unique set of features.

This paper describes the decoding (generation) module in a man-machine communication system that generates Arabic text from a hyper semantic network that is encoded within the Universal Networking Language (UNL) framework (the encoding module is presented in Alansary et al (2006) in this volume). It is organized as follows: Section 3 summarizes what the Universal Networking Language is. Section 4 describes the structure of the Arabic dictionary used in the DeConversion process. Section 5 traces the application of DeConversion rules till the Arabic text has been generated. Section 6 presents a conclusion and future work.

3. The Universal Networking Language.

Universal Networking Language (UNL) developed at UNU (United Nations University) is a formal language for representing the meaning of natural language sentences. This language is assumed to express meanings in the same standardized way as HTML presents its layout. A UNL expression is a (possibly) cyclic graph composed of nodes connected by semantic relations. Nodes, or Universal Words (UWs) are words loaned from English and disambiguated by their positioning in a knowledge base (KB) of conceptual hierarchies. Function words, such as determiners and auxiliaries are represented in the form of attributes to UWs, provided that these function words contribute to the meaning and are not syntactically motivated. Each relation is labeled with one of the possible label descriptors. Relations that link UWs are labeled with semantic roles of the type such as agent, object, experiencer, time, place, cause, which characterize the relationships between the concepts participating in the events or states a natural language sentence may denote. A simplified example of a UNL expression, as shown graphically in Figure (1), shows the different components of a UNL expression of the sentence “I hear a dog parking outside”.

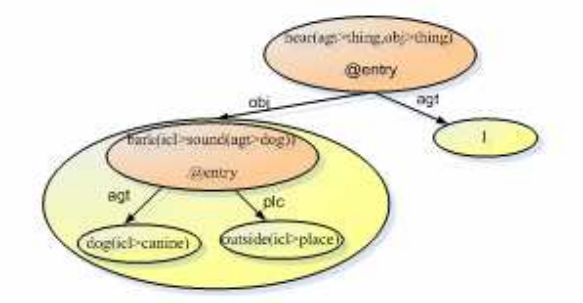
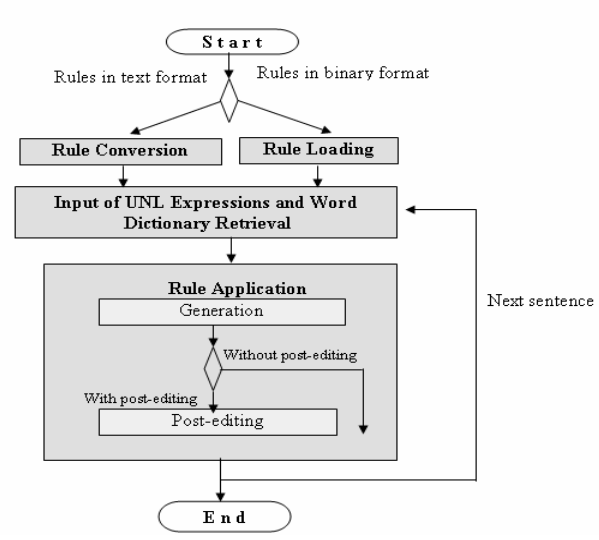


Figure (1): An example of a UNL graph.

As a result of this standardized meaning representation, documents no longer need to be multiplied in order to represent the content in different natural languages (Uchida (1996), Uchida et al (2002a), Uchida et al (2005)). The meaning representation is directly available to retrieval and indexing mechanisms and tools for automatic summarizing and knowledge extraction, and it will be converted to a natural language only when communicating with a human user. The task of the presentation of a UNL web-page to a web user will be taken over by a UNL-viewer. In one commercially oriented scenario, the UNL-viewer represents a new generation of web-browser which, in addition to their capacities to handle Java and Java-script, are equipped with one or more national UNL-DeConverter in order to display the meaning content in a national language.

3.1 The UNL DeConverter

The whole UNL System was described in Alansary et al (2006) of this volume. As the current paper is mainly concerned with the DeConversion process from UNL to Arabic, this section provides more detailed information about the 'UNL DeConverter' as a language independent generator (more technical information can be found in Uchida (1996 and 2002b)). This will help us to follow the discussions given through this paper. DeConverter generates target sentences of a native language from UNL expressions by applying DeConversion rules. Figure (2) shows how DeConverter works.



Figure(2): Flowchart of DeConversion process

Firstly, DeConverter converts DeConversion rules from text format into binary format, or loads the binary format DeConversion rules directly if they are already in binary format. Secondly, it inputs a sentence of UNL expressions and converts it into Node-net, when the word entries are also retrieved from the Word Dictionary using the UW of each node. Thirdly, it starts to apply rules to the Node-list from the initial state (See figure (3)).

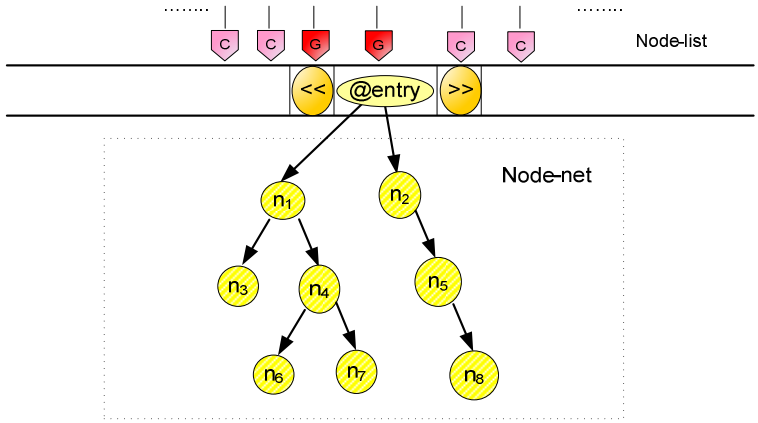


Figure (3): Initial state of the Generation Windows and the Node-net in the DeConversion.

DeConverter applies DeConversion rules to the Node-list and inserts nodes from the Node-net. This process will end when either the Sentence Tail node of the Node-list appears in the left Generation Window or the Sentence Head node appears in the right Generation Window (See figure (4)).

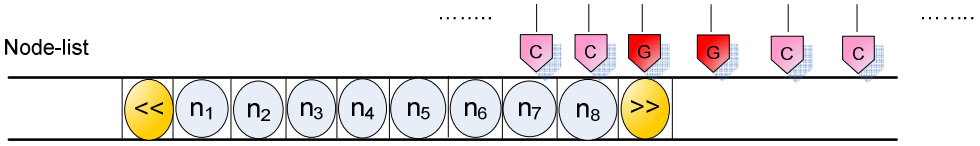


Figure (4): Final state of the Generation Windows and the Node-list in the DeConversion

4. Building the Arabic Dictionary

A UNL dictionary stores information for a language. It stores information concerning what kinds of UWs (concepts) the language expresses and where those words can be used. A word dictionary stores the following items:

1) Universal words for identifying concepts
2) Word headings for universal words that can express concepts
3) Information on the linguistic behavior of words

A word dictionary provides information for computers to understand natural language, and express information in natural language. A dictionary entry consists of a correspondence between a concept and a word, and information concerning morphological and syntactic properties of a word when that correspondence was established.

Each entry in the dictionary has the following format:

[HW] {ID} "UW" (ATTR,…) <FLG,FRE,PRI>;

For example: [ولد] {1} “boy(icl>person)” (CommonNoun,Sing,Masc….etc.) <A,0,0>;

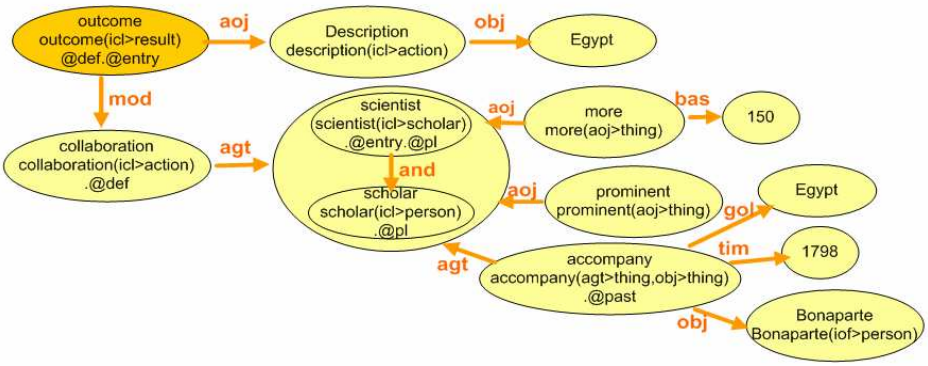
In building this dictionary, we considered that the head word will be stem based because this makes the derivation of plural nouns, for example, easier, without the need of another entry in our dictionary to express the plural noun. In fact, the design of the Arabic dictionary depends entirely on the approach by which the Arabic words have been dealt with. According to our design, the focus of attention is given to the form of the head word of the entry needed to fulfill language analysis and generation tasks adequately. Doing this is twofold: first, it will make it possible to avoid adding all possible inflectional and derivational paradigms of each lexical item to the dictionary (e.g. instead of storing حكومة, حكومتنا , حكومات etc., only حكوم will be stored) (cf. Al-Ansary (2003)). Second, to minimize the number of entries in the dictionary which will give more efficiency in the analysis and generation tasks and minimize the processing time. To reach this target a detailed computational linguistic analysis was conducted on the Arabic word form keeping an eye on both analysis and generation of word forms at the same time, given the fact that the same dictionary should be used in

both analysis and generation. Based on this computational linguistic study the best form of the lexeme to be stored to represent all its paradigms has been reached.

5. DeConverting UNL networks to Arabic

The DeConverter's generation ability is similar to that of a tuning machine. It is capable of generating all types of sentences applicable to all languages. Co-occurrence relations between words contribute to a better word selection. This means it is possible to generate more natural sentences by using co-occurrence relations.

In DeConverting hyper networks to natural language, the DeConverter transforms the sentence represented by a UNL expression – i.e., a set of binary relations – into a directed hyper-graph structure called Node-net. The root node of a Node-net is called Entry Node and represents the main predicate of the sentence. The DeConverter applies generation rules to every node in the Node-net and generates the word list in the target language. In this process, the syntactic structure is determined by applying syntactic rules. Morphemes are similarly generated by applying morphological rules. Given the hyper semantic network:



the DeConverter outputs the following Arabic sentence given Arabic DeConversion rules and Arabic dictionary:

وصف مصر محصلة تعاون أكثر من 150 باحث وعالم مرموق الذين صاحبوا بونابرت في 1798 إلى مصر.

5.1 Implementing Arabic Generation Rules

According to our technical design of the Arabic DeConverter, it is divided into two stages, namely the syntactic stage and the morphological stage. The syntactic stage deals with order of words in the node list, while morphological stage specifies how to form words and deals with agreement gender, number, person and definiteness. The following subsections will deal with the syntactic stage and the morphological stage respectively.

5.1.1 Syntactic rules

Syntactic rules can be divided into two sub-stages, namely, determining the main predicate of the sentence together with its modifiers representing the main skeleton of the sentence; and the modifiers representing relations with each element composing the main structure. The following subsections will discuss each sub-stage in more detail.

5.1.1.1 Determining the main sentence structure

Syntactic rules can be divided into two phases. The first phase deals with the main sentence structure that can be determined from the input itself, the UNL expression. The starting node in the UNL network is the 'entry' node that refers to the main predicate of the sentence which is marked "@entry". The first phase deals with identifying the modifiers of the main predicate that share it in forming the main sentence structure. Note that both of the nodes representing the main sentence structure and the nodes representing modifiers have to be arranged in a way that conveys the correct intended meaning of the original sentence from which the UNL expression has been EnConverted. Therefore, in organizing our grammar, different relations inside UNL networks have been studied to prepare for different possible structures that can be generated in Arabic. Accordingly, syntactic patterns have been determined to be targets for UNL networks (see figure (5)).

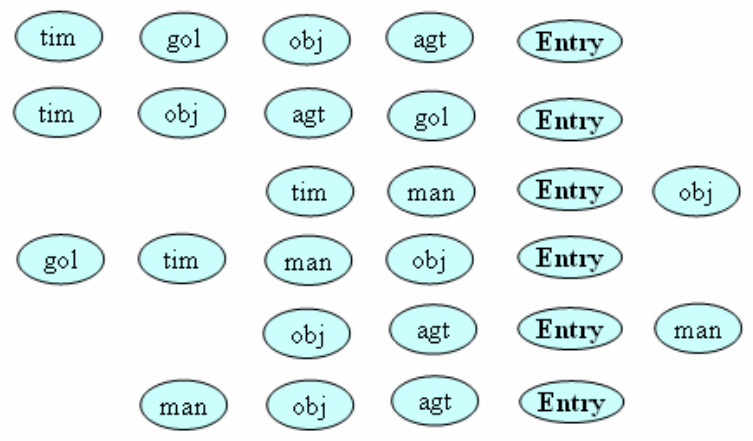


Figure (5)

However, deciding which syntactic pattern that should be selected to generate the main structure of UNL expressions is not an easy task. For example, consider the following UNL expression:

{org}
Oe became increasingly interested in rural folk legends.
{/org}
{unl}
tim(become(gol>thing,obj>thing,src>thing):0N.@past.@entry, cry(icl>voice):0G.@def)
obj(become(gol>thing,obj>thing,src>thing):0N.@past.@entry, Oe(iof>person):0K.@topic)
gol(become(gol>thing,obj>thing,src>thing):0N.@past.@entry, interested in(aoj>thing,obj>thing):17)
man(become(gol>thing,obj>thing,src>thing):0N.@past.@entry, increasingly:0U)
aoj(silent(aoj>thing):09, cry(icl>voice):0G.@def)
mod(legend(icl>tale):1W.@pl, folk(mod<thing):1R)
aoj(rural(aoj>thing):1L, legend(icl>tale):1W.@pl)
obj(interested in(aoj>thing,obj>thing):17, legend(icl>tale):1W.@pl)
{/unl}

To generate this UNL in Arabic. The following structure should be selected from the list of structures listed above in figure (5):

This structure can be selected to DeConvert the UNL expression mentioned above to Arabic, therefore the following Arabic sentence can be considered as a possible output:

أصبح أويه بشكل متزايد منذ الصرخة الصامتة مهتما بالأساطير.

What is important here is that the object 'أويه ' is inserted after the 'entry node' أصبح. This situation maybe undesirable with another UNL expression that has the same type of relations. Consider the following example:

{org}
nothing similar had ever been attempted before.
{/org}
{unl}
obj(attempt(agt>thing,obj>thing):0U.@complete.@past.@entry, nothing:00.@topic)
man(attempt(agt>thing,obj>thing):0U.@complete.@past.@entry, ever:09)
tim(attempt(agt>thing,obj>thing):0U.@complete.@past.@entry, before(icl>time):14)
aoj(similar(aoj>thing):08, nothing:00.@topic)
{/unl}

As the 'Entry node' has the same type of relation i.e. 'obj', the node representing the obj relation should inserted after the entry as happened with the previous situation. The result will be the following out put:

أعد لاشيء مماثل أبدًا من قبل.

However, as it might be noticed, the position of the 'obj' in this output makes the sentence seem odd, and it would be better if the 'obj' would have been inserted before the 'Entry'. In this case the structure

should be used to generate the following output:

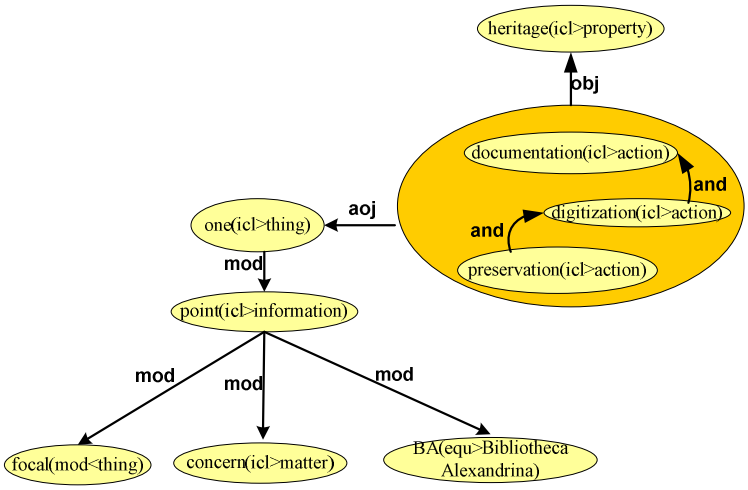
لاشيء مماثل أعد أبدًا من قبل.

In its current state, the grammar is capable of dealing with situations like these mentioned above. This has been achieved by the sub-classifiaction of concepts and the detailed description of the environment in which nodes are going to be inserted.

5.1.1.2 Generating Modifiers:

In this section we will give an overview of the second sub-stage of the grammar that deals with the insertion of modifiers after the main sentence structure has been generated. One of the challenges faced in this stage is when we have a given node in the semantic network that has more than one modifier of the same type. For example, consider the following UNL expression and its graphical representation that follows:

{org}
one of the BA's focal points of concern is the <c>documentation, digitization, and preservation</c> of heritage.
{/org}
{unl}
aoj(:01.@entry, one(icl>thing):00.@topic)
obj(:01.@entry, heritage(icl>property):2R)
and:01(preservation(icl>action):2B.@entry.@def, digitization(icl>action):1T.@def)
and:01(digitization(icl>action):1T.@def, documentation(icl>action):1E.@def)
mod(one(icl>thing):00.@topic, point(icl>information):0M.@pl)
mod(point(icl>information):0M.@pl, concern(icl>matter):0W)
mod(point(icl>information):0M.@pl, BA(equ>Bibliotheca Alexandrina):0B.@def)
mod(point(icl>information):0M.@pl, focal(mod<thing):0G)
{/unl}



As seen the graph above, the node 'point(icl>information)' has three simultaneous ‘mod’ relations with other nodes. Therefore, the question now is: *Which node of the three modifiers should be inserted first? And which node should*

follow it? Of course we need a priority in the insertion of the three modifiers. According to the organization of our grammar the node "focal(mod<thing)" 'أساسي' is inserted first, then "BA(equ>Bibliotheca Alexandria)" ' مكتبة الإسكندرية ' and finally "concern(icl>matter)" 'اهتمام' which has the lowest priority in the insertion. This will result in generating the phrase:

محاور(اهتمام) (مكتبة الإسكندرية) (أساسي).

Accordingly, the syntactic component of the grammar can generate the whole UNL expression above as :

أحد محاور اهتمام مكتبة الإسكندرية أساسي التوثيق والتسجيل في شكل رقمي والتخزين للتراث[2].

5.1.2 Morphological Rules

The Morphological stage is the final stage in the DeConversion process that is concerned with three axes. First, inserting affixes to the node list to generate the final form of the entries according to the linguistic features attached to each entry in the dictionary, or according to attributes attached to the nodes because of semantic relations. Second, inserting prepositions, attributes, and pronouns that are needed because of the Arabic syntactic structure under generation. Third, inserting punctuations and spaces whenever needed. In the subsequent sections every axis will be dealt with independently in some more detail.

5.1.2.1 Inserting Affixes:

As insertion of this type of affix is based on the features stored in the dictionary, the features in turn are based on the form of the dictionary entries selected to represent different paradigms representing lexemes. For example, the form of the augmented defective verb "أعطى" 'give' changes according to subject pronouns. Therefore, three forms for this verb have been selected:

[عطى] {}"give(agt>thing,obj>action)"(P1.1.2,3V,V1)<A,0,0>;
[عطي] {}"give(agt>thing,obj>action)"(P1.1.2,3V,V2)<A,0,0>;
[عط] {}"give(agt>thing,obj>action)"(P1.1.2,3V,V3)<A,0,0>;

Each of the entries has given a different code, to be used in selecting the form requires to represent the concept "give(agt>thing,obj>action)". In addition, based on the subject a given affix will be added to the head word to generate the realized form. Two types of rules work in complementary with different forms representing the same lexeme; namely, Backtrack rules and message transfer rules.

Back track rules are responsible for rejecting incompatible morphemes. For example, in the UNL: **agt(live(agt>thing):09.@entry.@progress, I:00.@topic)**, there is an agent relation between **live(agt>thing)** and **I** . When DeConverting this UNL, the engine has to choose the correct form of the progressive verb that goes with the first person singular pronoun. As there are three forms stored in the dictionary for the verb 'live', therefore, if the form "عاش" has been selected, the back track rule:

?R{1.1AG<15,15,C1,N1,R2.1,P1:::}{1.1AG>15,21,@progress,2V,V1:-V1::}P220;   **أنا عاش**

will be applied to reject the right node. If the form "عش" is selected, the back track rule:

?R{1.1AG<15,15,C1,N1,R2.1,P1:::}{1.1AG>15,21,@progress,2V,V3:-V3::}P220;   **أنا عش**

will be applied to reject the right node as it is still morphologically inappropriate to the left node. So, in case of incompatible nodes, backtrack rules redirect the engine to select "**عيش**" with"**أنا**". After the correct form of the verb has been selected, a 'message transfer' rule will be applied to transfer a certain feature to the verb by which the prefix "أ" will be added. Finally, the output will be "**أنا أعيش**".

[2] Note that the sentence still needs agreement (in gender and definiteness) which will be dealt in the third stage of the grammar (morphological stage)

5.1.2.2 Inserting Particles:

This stage takes care of inserting nodes that are not included in the UNL expression itself but they are needed for a syntactic necessity for the generated language. Many relations need a specific preposition to be generated in the target language. For example, "plc" relation (place) needs preposition "في" and "tim" relation (time) needs "أثناء". In addition, some concepts in the dictionary imply insertion of a specific preposition with a specific relation, for example: "**ناتج**" implies the preposition "**عن**" with "**aoj**" relation, and "أرسل"implies the preposition "**إلى**" with "**gol**" relation .

5.1.2.3 Inserting punctuations and spaces:

Punctuation marks are usually expressed in the UNL expression by attributes. These attributes will be generated by inserting punctuation marks in the morphological phase. Spaces will be added at the end of the morphological phase after inserting all nodes from the node net. Spaces separate all nodes except nodes that represent affixes.

5.1.3 A corpus-based example of the generation process:

In this section a concrete example of the whole generation process to DeConvert a given UNL to Arabic will be presented. The example will trace rule applications on every node till the final output has been obtained. Consider the following UNL expression[3]:

{unl}
obj(express(agt>thing,obj>abstract thing):0G.@entry.@past, sense(icl>feeling):0U)
agt(express(agt>thing,obj>abstract thing):0G.@entry.@past, work(icl>book):0A.@pl)
mod(work(icl>book):0A.@pl, he:00)
mod(work(icl>book):0A.@pl, early(mod<thing):04)
mod(sense(icl>feeling):0U, :01)
mod(sense(icl>feeling):0U, he:0Q)
obj(cause(aoj>thing,obj>thing):29.@past, :01)
and:01(disorientation(icl>state):1Q.@entry.@def, degradation(icl>phenomenon):1A.@def)
aoj(cause(aoj>thing,obj>thing):29.@past, surrender(icl>phenomenon):2R)
tim(cause(aoj>thing,obj>thing):29.@past, end(icl>time):38.@def)
mod(surrender(icl>phenomenon):2R, Japan:2J)
mod(end(icl>time):38.@def, World War II:3F)
{/unl}

which can be represented graphically as:

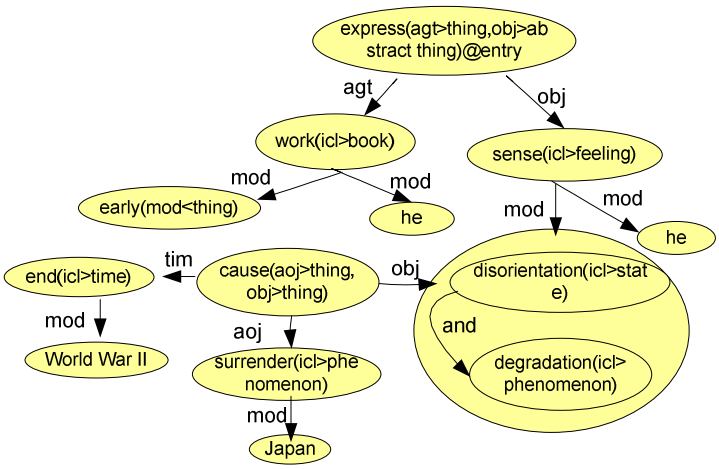


Figure (6)

[3] This UNL is taken from the C. V. of Kenzaboro Oe, a famous Japanese writer who has been awarded the Nobel Prize for literature.

The DeConversion process initiates automatically from the node "express(agt>thing,obj>abstract thing)" as it is marked as the 'entry' of the network; it represents the main predicate of the sentence. As we see in figure (6) the node “express(agt>thing,object>abstract thing)” has 'agt' relation with work(icl>book) and 'obj' with "sense(icl>feeling)". Therefore, the main sentence structure is determined according to the following the order:

Therefore the first rule applies is to tag the entry as representing part of the main skeleton of the sentence.

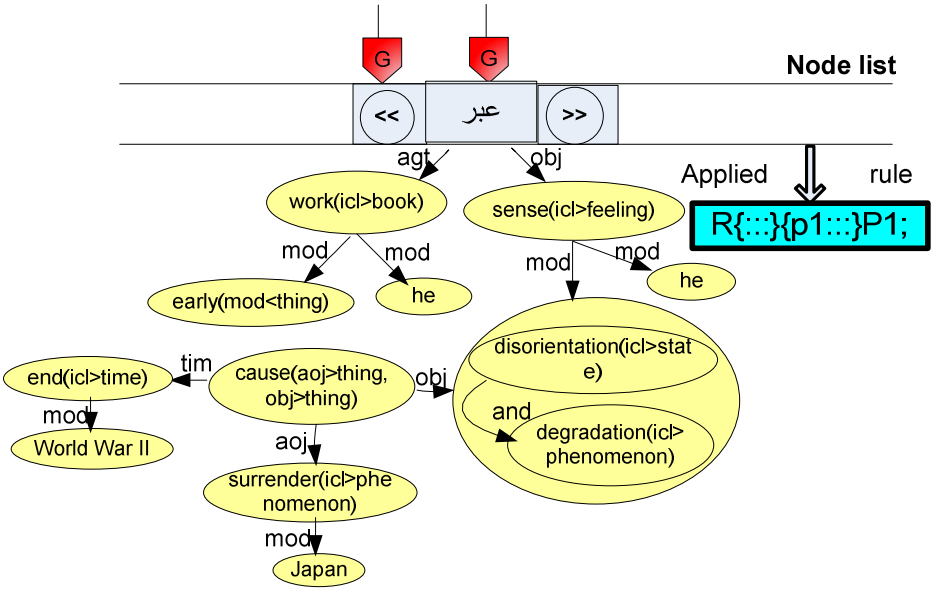


Figure (7)

In figure (7) the left Generation Window is locate on the sentence Head node and the right Generation Window is located on the entry node. As the processing is still in the beginning, the applied rule moves the Generation Windows to the right to be located on the entry and the sentence tail as appears in figure (8).

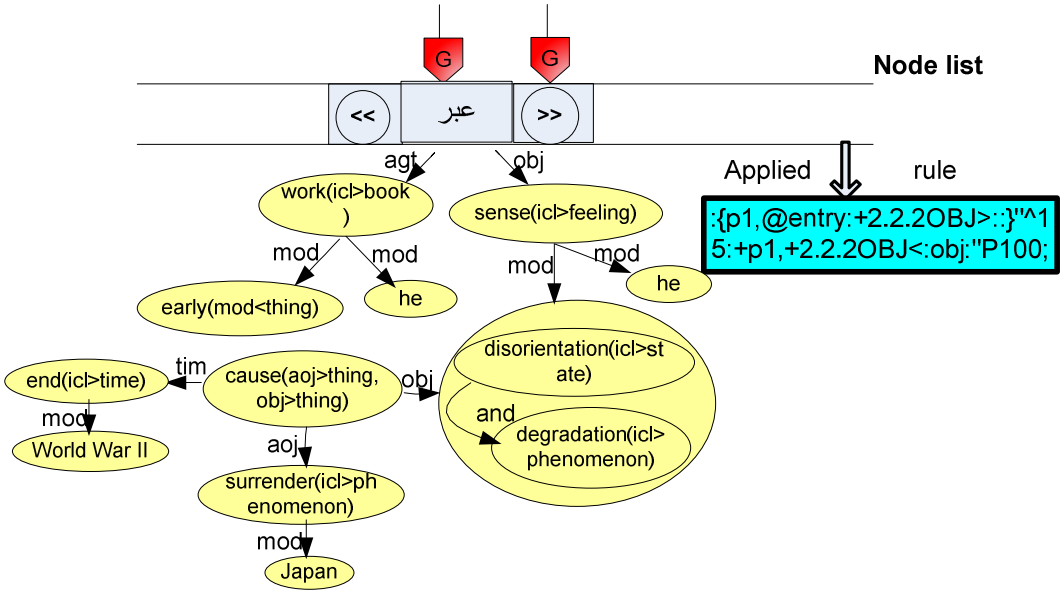


Figure (8)

In figure (8), the DeConverter applies a right insertion rule to insert the node “sense(icl>feeling)” which is related to the entry by an 'obj' relation. It can be noted that although both of the node “sense(icl>feeling)” and “work(icl>book)” have simultaneous relations with the entry but the DeConverter intends to insert “sense(icl>feeling)” before

“work(icl>book)”. This occurs because, according to the design of the grammar, the priority of the 'obj' rule is higher than that of the 'agt' rule.

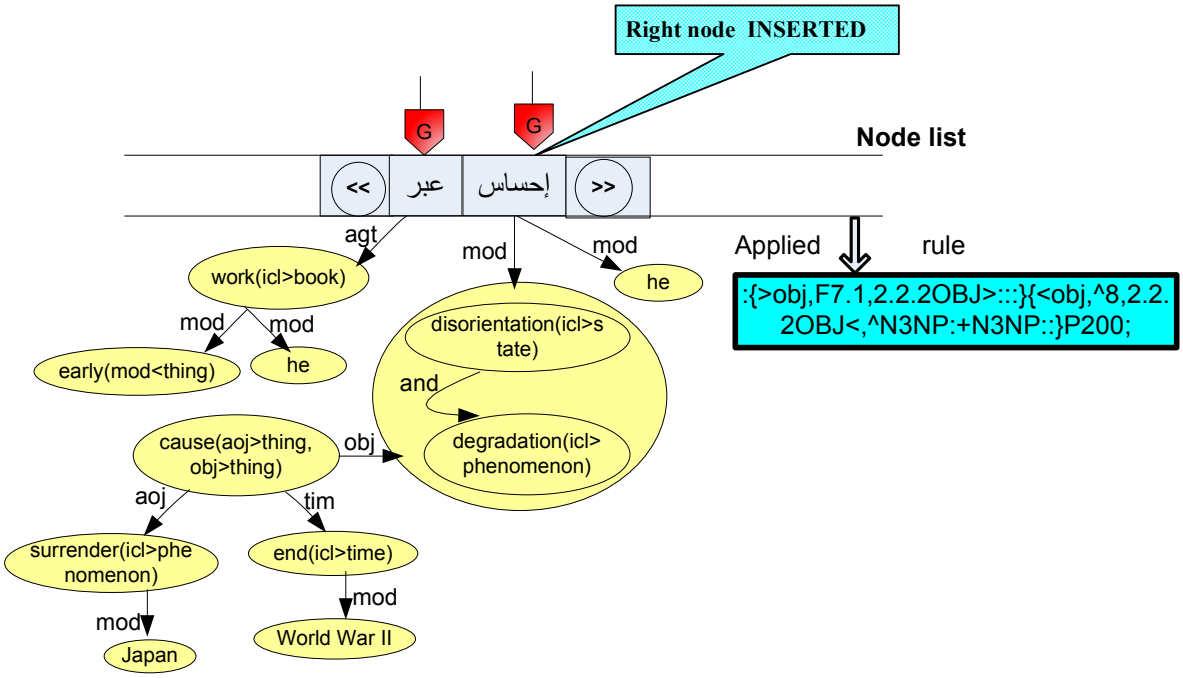


Figure (9)

As a result of rule application, the node “إحساس” is inserted in the node list, as shown in figure (9) In addition, the node “إحساس” is marked to prepare for “عن” to be inserted in a later stage in the grammar.

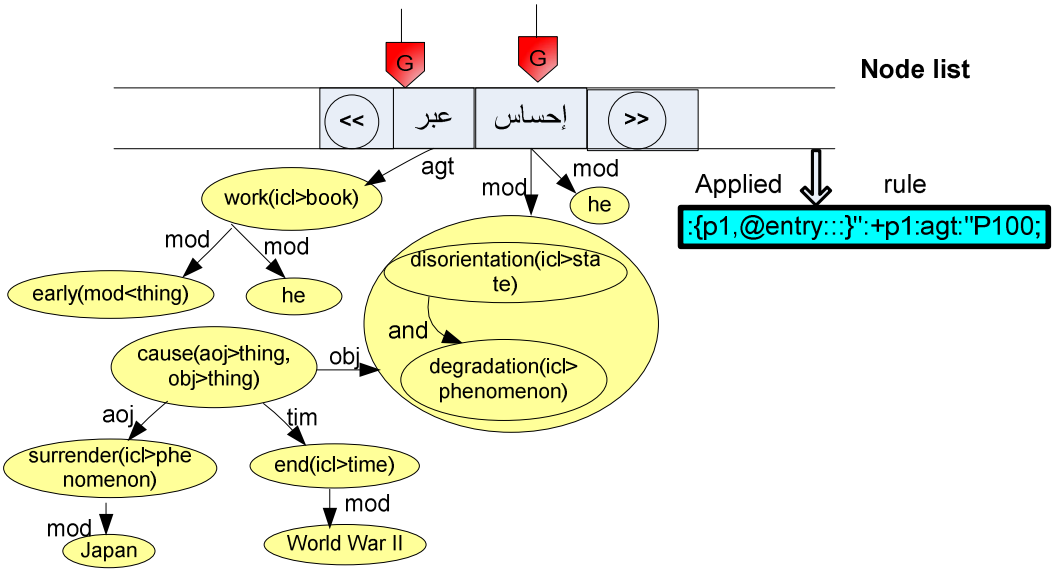


Figure (10)

In figure (10), after the 'obj' has been inserted, the left Generation Window is placed on the entry node “عبر” and the right Generation Window placed on the “إحساس”. The priority is given the 'agt' rule to apply.

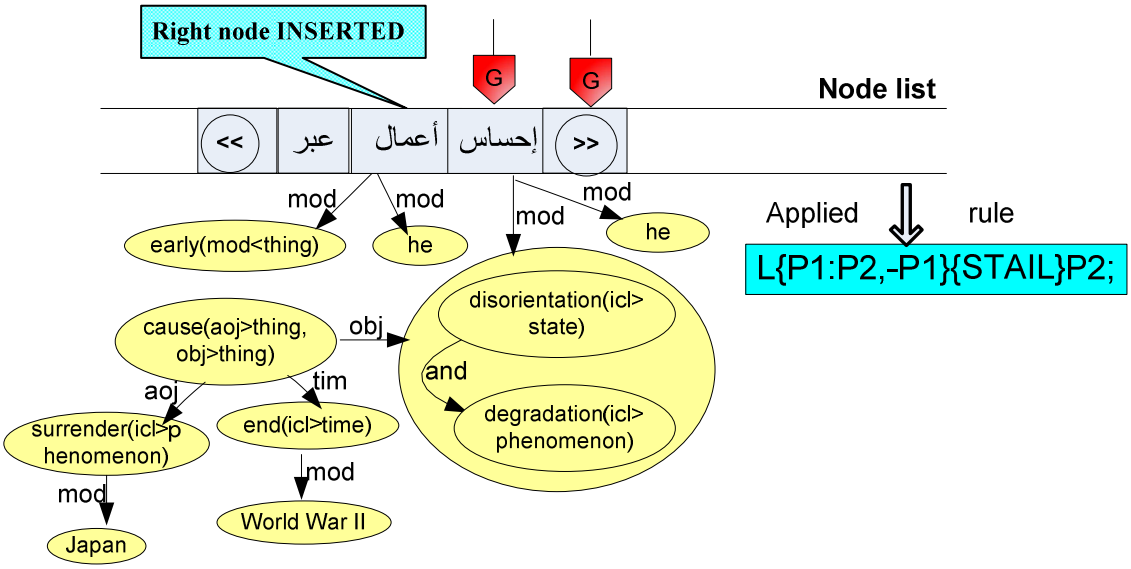


Figure (11)

In figure (11) the word "أعمال" is moved from the node net to the node list. As there is no other node linked with the entry, the DeConverted realizes that the main sentence structure has been completed, therefore Generation Windows move right to STAIL to start a new phase for inserting modifiers. The left shift rule in figure (11) moves Generation Windows to the beginning of the node list to generate modifiers, if exist.

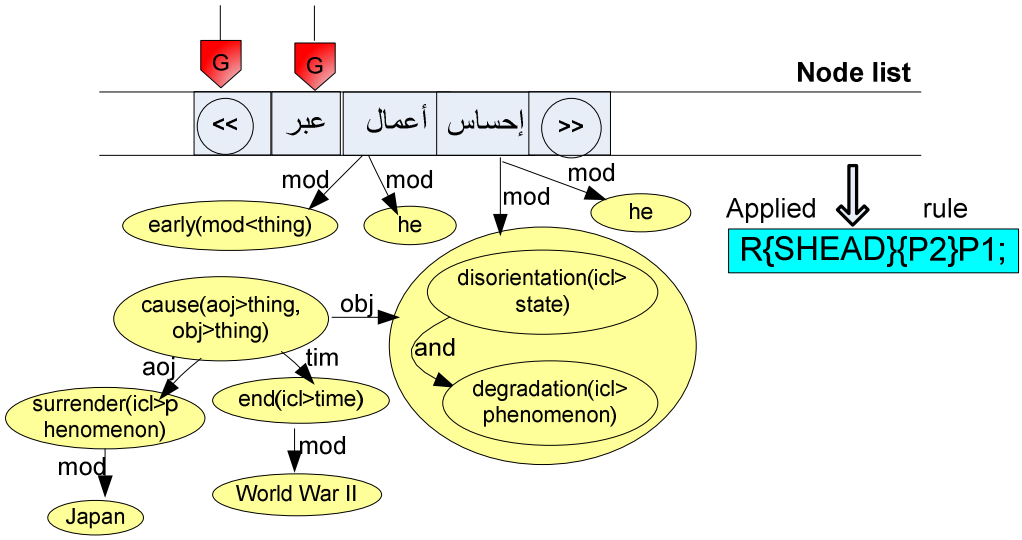


Figure (12)

In figure (12) the left Generation Window is on the SHEAD and the right one is on "عبر" where no relation exists. The right shift rule in figure (12) moves the flow of processing till the first node that has modifiers has been reached as seen in figure (13).

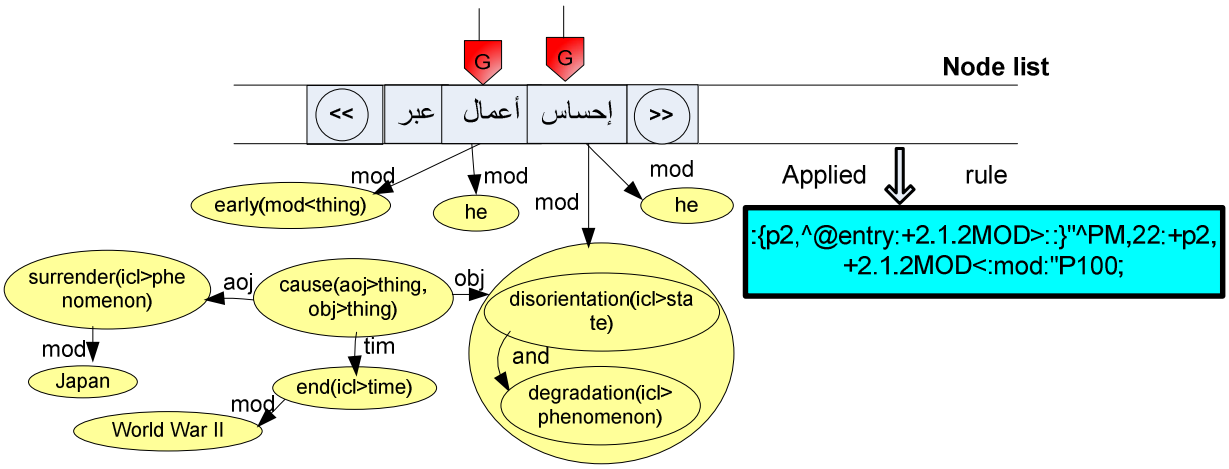


Figure (13)

The rule in figure (13), will insert the modifiers of the left node. It maybe noted that both modifiers are linked to the node "أعمال" with the same relation. To disambiguate which node should be inserted first, the priority is given to the node that is represented by an adjectival concept than the node that is represented by a nominal concept (pronoun). Therefore, the rule in figure (13) will be applied to insert the node 'early(mod<thing)' "مبكر" as shown in figure (14).

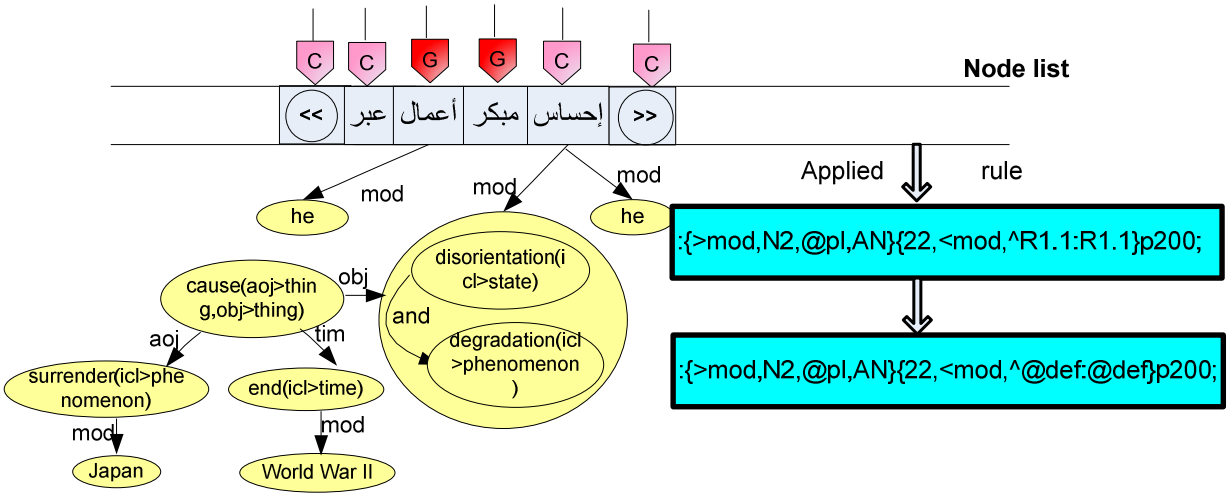


Figure (14)

In this figure the adjective "مبكر" has been inserted, the Generation Windows move right to "أعمال" and "إحساس". In this situation, a massage transfer rule will give a feature to the adjective which will enable the morphological generation rules, later on, from inserting the suffix which refers to the noun-adjective agreement. Also another message transfer will apply to give a feature to the adjective to achieve the agreement indefiniteness between "أعمال", "مبكر".

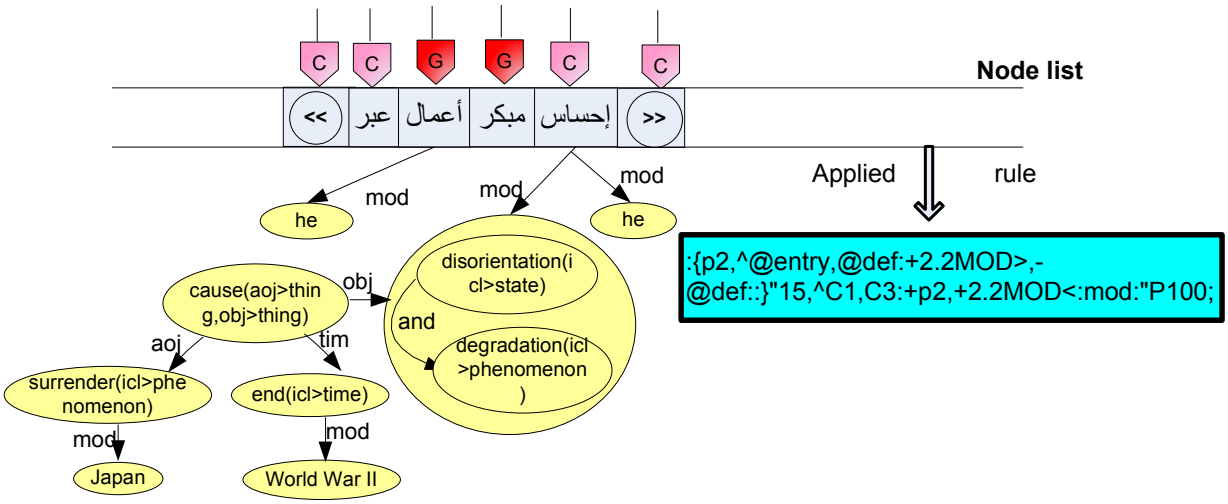


Figure (15)

The node “he” remains in the node net in figure (15) which will be inserted afterwards by applying the rule given above.

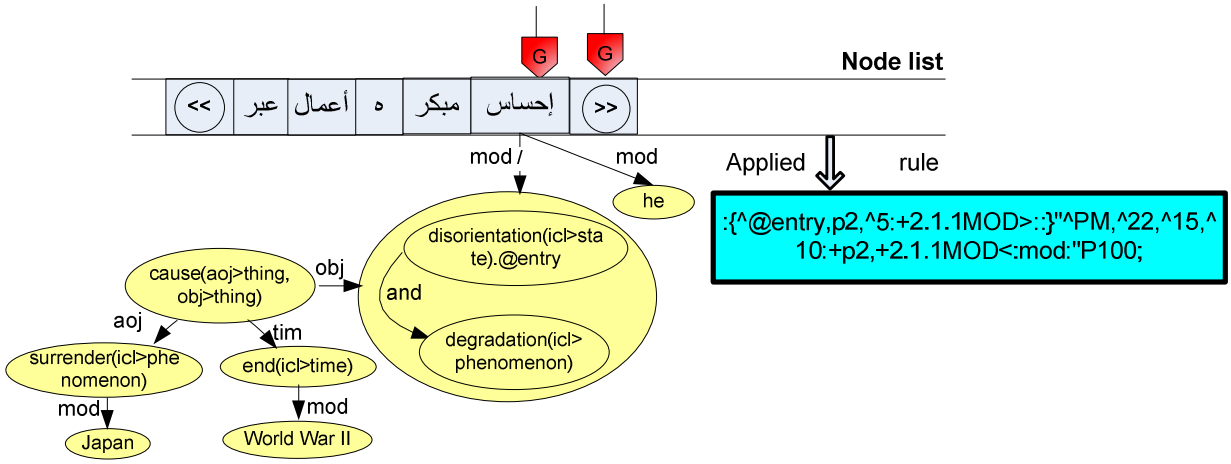


Figure (16)

In figure (16) the pronoun “ه” has been inserted. As the node “أعمال” is no longer connected with any other modifier, Generation Windows move right to “إحساس” and the STAIL. In this situation, the two modifiers have been detected with the node “إحساس”; the first one is a scope and the second is the node “he”. Insertion of the scope has given the priority over the insertion of the pronoun. Therefore, the DeConverter begins generating the scope starting from the entry of the scope (disorientation(icl>state)).

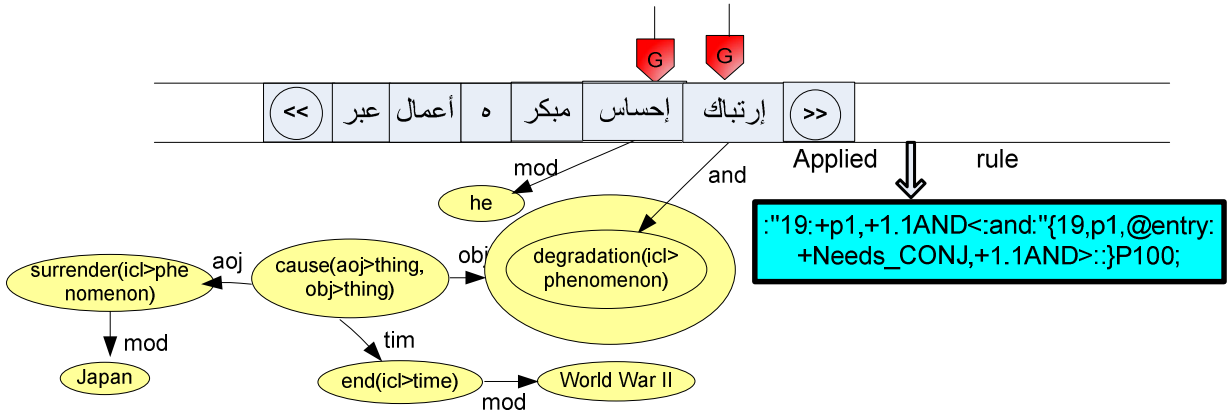


Figure (17)

In figure (17), the entry of the Scope is moved from the node net to the node list. Yet, the entry of the scope still has other relations with other nodes. The rest of the relations inside the scope is generated by inserting "استسلام" in the node list as seen in figure (18).

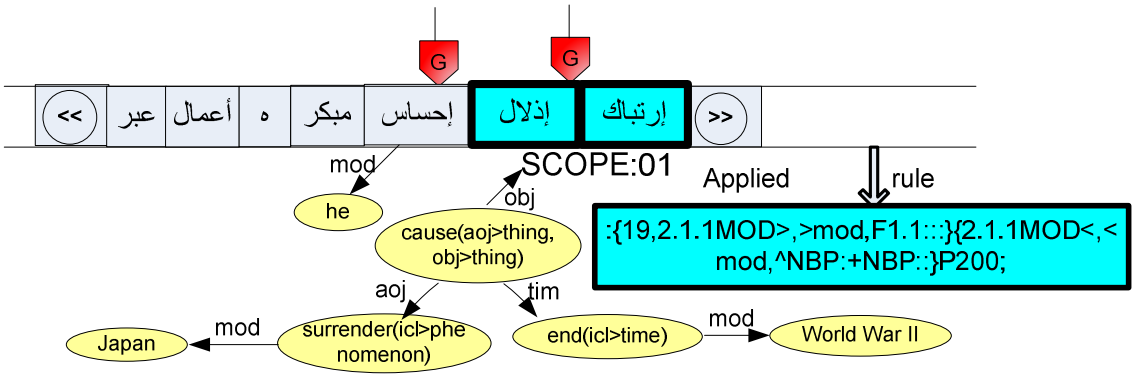


Figure (18)

As the relation between "إحساس" and the scope was 'mod', the scope is marked that it will need a preposition. After the scope has been generated, the DeConverter returns back to generate the node "he" that is linked to "إحساس" by a 'mod' relation and that has been left before generating the scope. The rule in figure (98) starts to apply to generate the 'mod' relation between "he" and "إحساس".

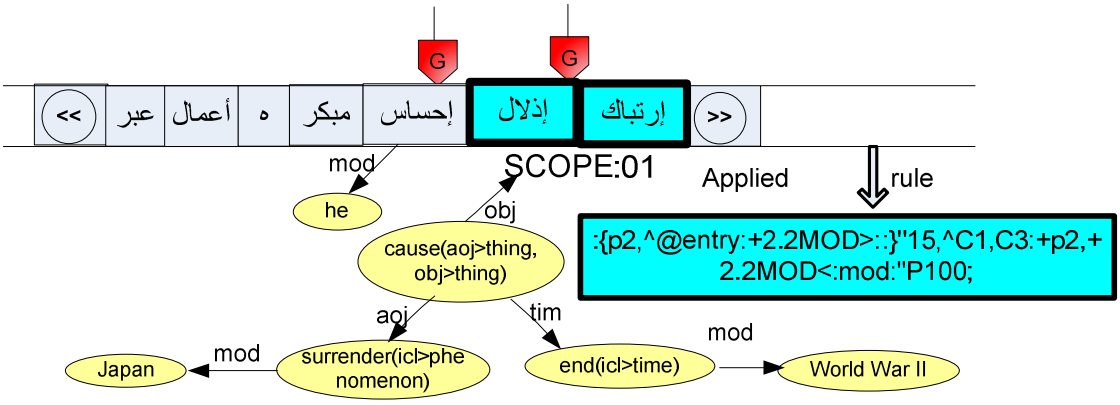


Figure (19)

After the application of the rule in figure (19), the pronoun "ه" appears in the node list as shown in figure (20).

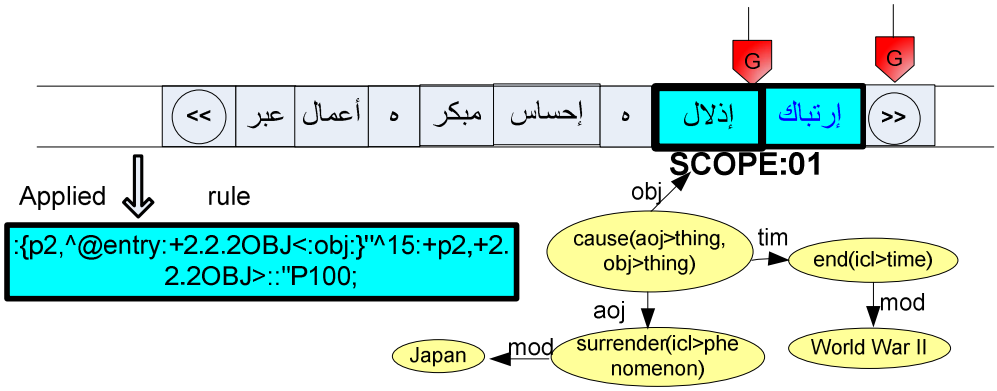


Figure (20)

Afterwards, the Generation Windows moves right to the scope and the STAIL where there is a modifier linked with the scope by an ‘obj’ relation, therefore the rule in figure (20) applies to generate the node “cause(aoj>thing,obj>thing).

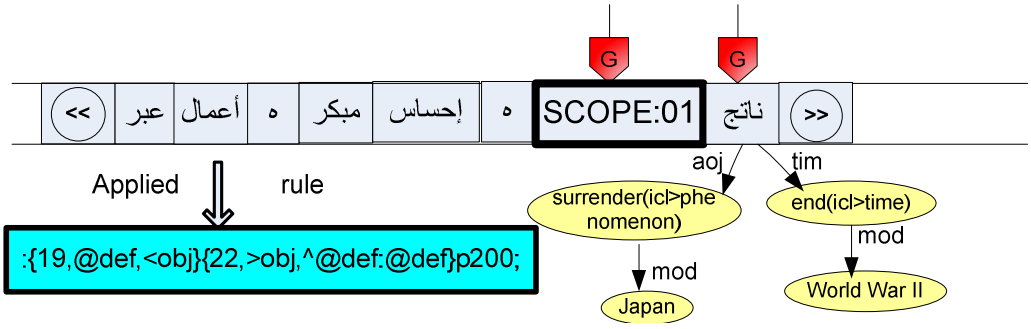


Figure (21)

After the insertion of the node “ناتج”, the attribute “@def” has been transferred to it from the scope that precedes as it is definite (figure (21)).

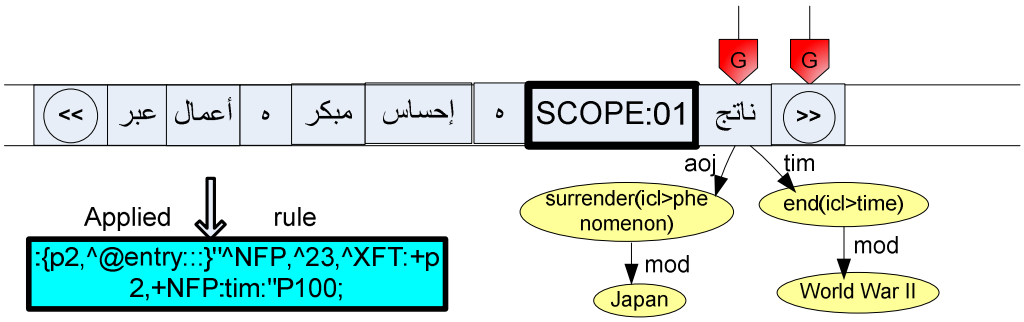


Figure (22)

It is clear in figure (22) that the node “ناتج” has two modifiers; the first is linked by ‘aoj’ relation while the second is linked by a ‘tim’ relation. The ‘tim’ relation has the priority over the ‘aoj’, therefore the rule in figure (22) applied to DeConvert the ‘tim’ relation first.

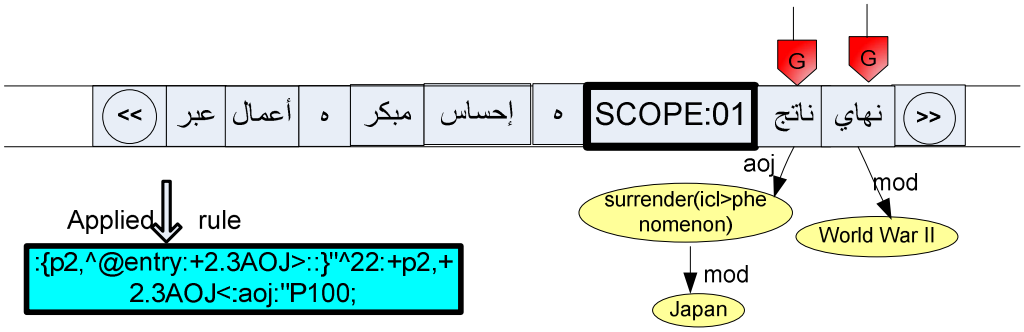


Figure (23)

The DeConversion rule in figure (22) moved the node ‘end(icl>time)’ from the node net and inserted it as “نهاي” in the node list as seen in figure(23). Another rule that takes care of the ‘aoj’ relation can be seen in figure (24) in which the node “استسلام” is inserted into the node list.

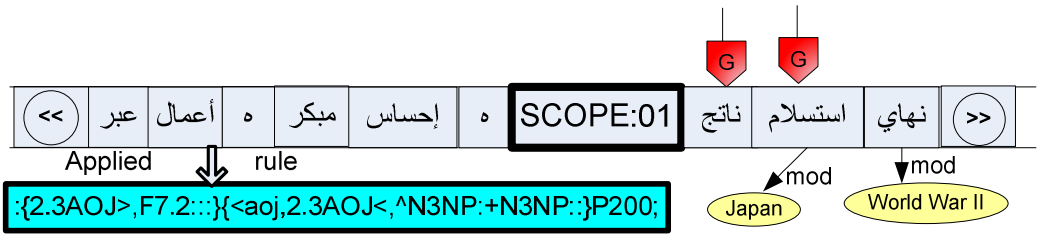


Figure (24)

As "ناتج" is subcategorized in the dictionary as permitting a following preposition "عن", a message transfer rule is applied to transfer an attribute from the node "ناتج" to the node "استسلام" which will enable the morphological generation rules, later on, from inserting the preposition "عن" before "استسلام" (figure (24). Then, the two Generation Windows will move to the node word "استسلام" and "نهاي".

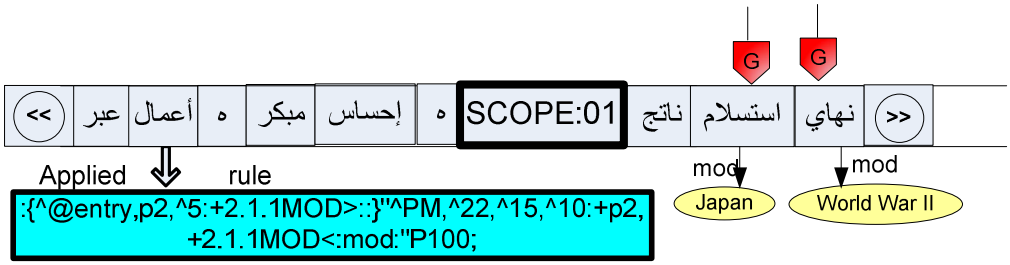


Figure (25)

Figure (25) shows that the left Generation Window is placed on the node "استسلام" which still has one modifier and the right Generation Window is placed on the node "نهاي" which also has one modifier. The rule in this figure will generate the modifier of the left node as shown in figure (26).

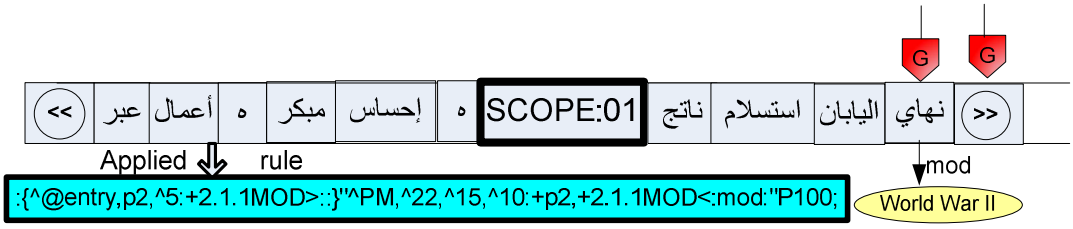


Figure (26)

The applied rule in figure (26) resulted in inserting the node "اليابان" after the node "استسلام". In addition, the Generation Windows shift right focusing on the node "نهاي" and the STAIL. Another insertion rule will start to insert the modifier of the last node in the node list, "الحرب العالمية الثانية".

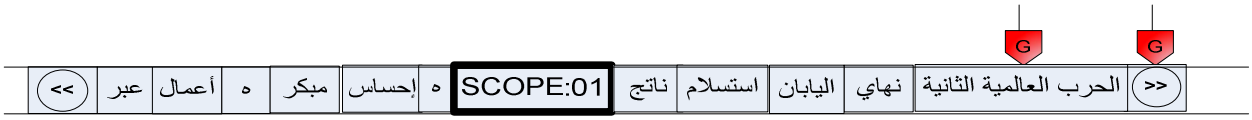


Figure (27)

Figure (27) shows that all the modifiers of the nodes that share in the main sentence structure are inserted in the node list. In this situation where the left Generation Window is placed on "الحرب العالمية الثانية" and the right Generation Window is on STAIL the two Generation Windows move left direction starting to apply morphological rules.

Figure (28)

Figure (28) shows that while the right Generation Window is on the node "نهاي", the preposition "في" is inserted before the word "نهاي" which during its insertion in the node list, it has given a feature to allow for inserting this preposition in the morphological phase.

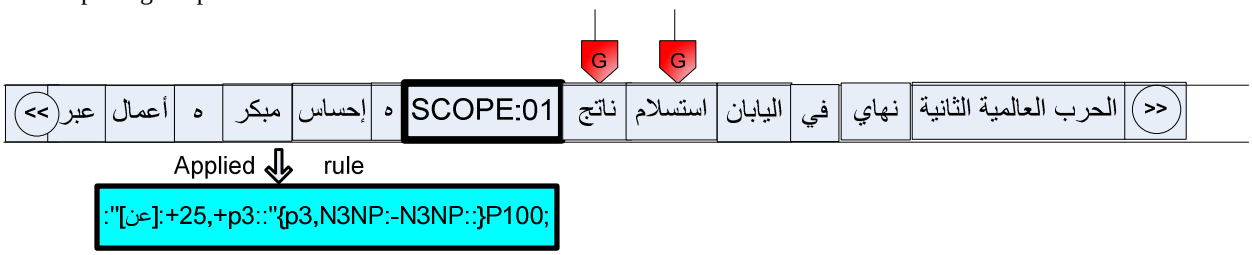


Figure (29)

The preposition "في" has been inserted in figure (29), and while the left and right Generation Windows are moving left are placed on the "ناتج" and "استسلام" respectively, the preposition "عن" is being inserted before the node "استسلام" which has been marked before to allow for inserting "عن" in the morphological stage.

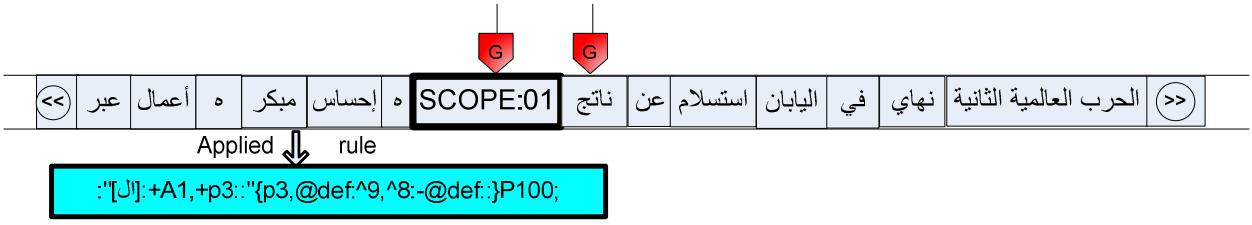


Figure (30)

Figure (30) shows that the preposition "عن" is inserted in the node list while another rule inserts the "ال " because of the attribute "@def" the node "ناتج" has been previously marked for in figure (21).

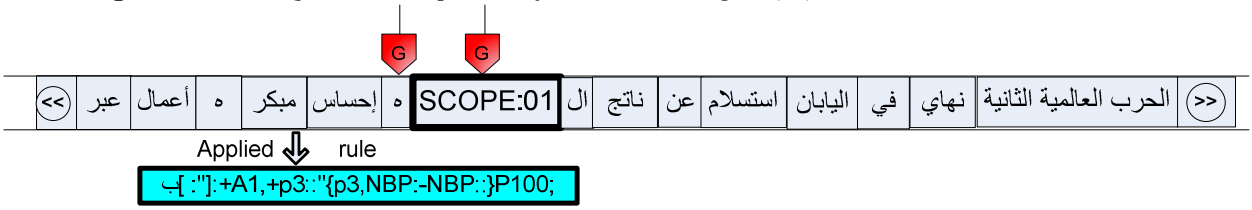


Figure (31)

Figure (31) shows that the preposition "عن" is inserted in the node list. Generation Windows continue moving left till reaching the Scope and the pronoun "ه". An insertion rule will apply aiming at inserting the preposition "ب" before the Scope.

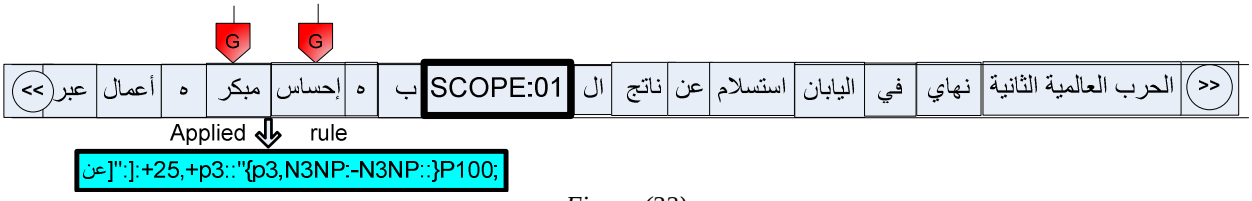


Figure (32)

After the preposition "ب" has been inserted, Generation Windows continue moving left stopping again by the nodes "مبكر" and "إحساس". As the node "إحساس" at an early stage in the grammar has received a message from "عبر" that allows it for inserting the preposition "عن", the rule in figure (32) inserts this preposition before the right node.

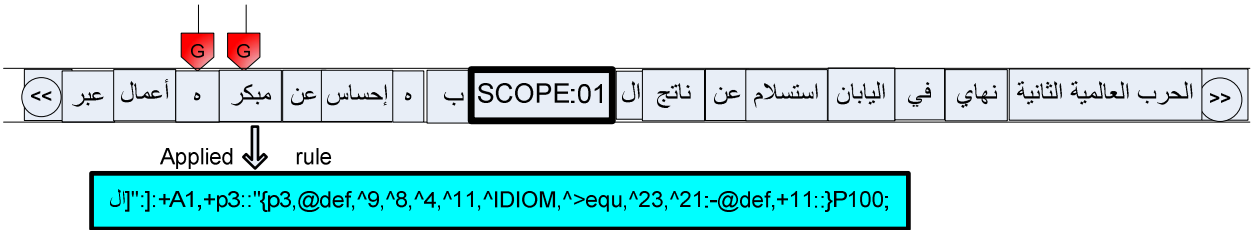


Figure (33)

In figure (33), the Generation Windows move left until another morphological rule applies. As the node "مبكر" carries the UNL attribute '.@def', the rule in figure (33) inserts the definite article "ال" before the right node.

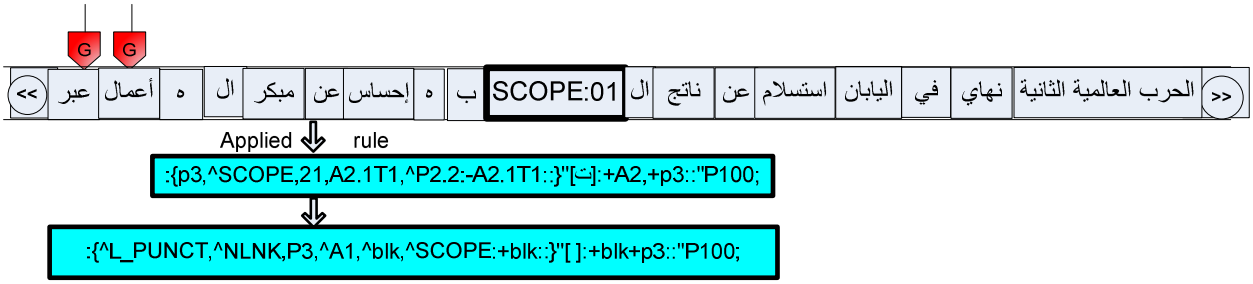


Figure (34)

Afterwards, Generation Windows focus on the verb "عبر" and its agent "أعمال", where agreement is needed. Therefore, first the rule in figure (34) is designed to insert the feminine "ت" before the agent. Then the second above rule starts to add spaces to separate the words of the sentence.

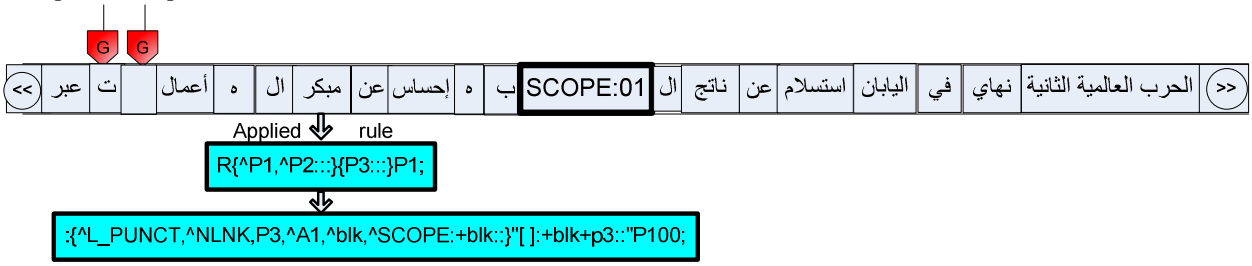


Figure (35)

After agreement has been established between the verb and the subject (figure 35), the grammar tries to find out other situations where agreement is needed.

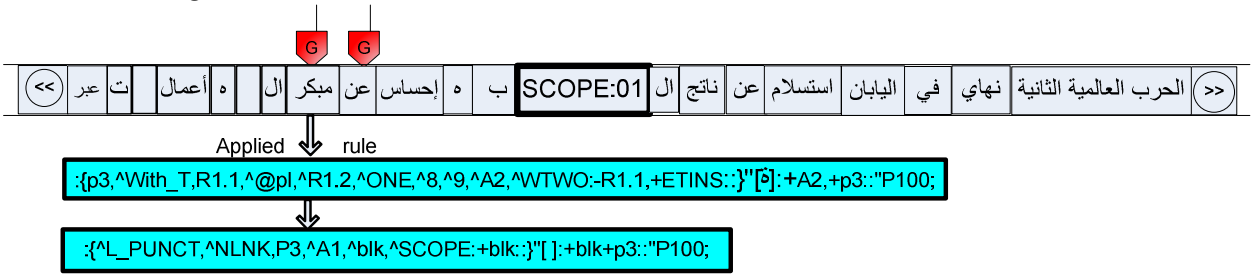


Figure (36)

Generation Windows continue moving right until the left Generation Window is on "مبكر". The rule in figure (36) inserts a third person singular feminine suffix to capture noun-adjective agreement.

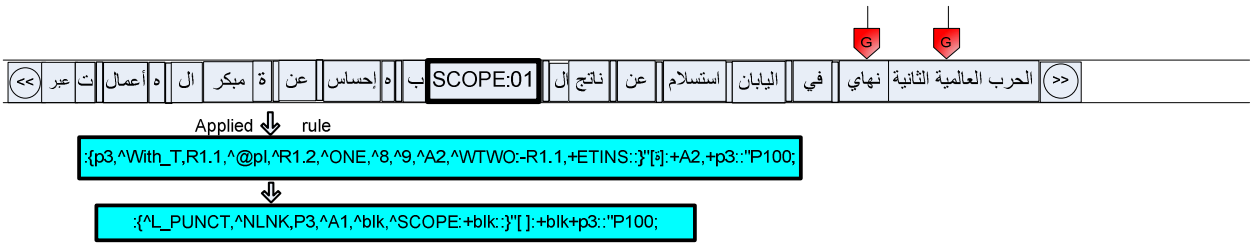


Figure (37)

Figure (37) shows that the suffix "ة" is inserted after the adjective "مبكر" as the previous noun is a broken plural. Generation Windows continue moving right tracing insertion of other suffixes. A final morphological rule is applied to insert the suffix "ة" after the node "نهاي" as it is marked in the dictionary that it needs this suffix in case of singular. Another rule applies to insert a blank space after the "ة" has been inserted.

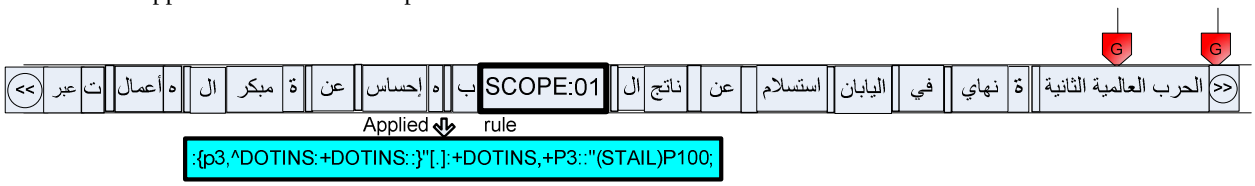


Figure (38)

In figure (38), the suffix "ة" is actually inserted after the node "نهاي". As the Generation Windows have reached the STAIL, only punctuation marks need to be inserted in the correct places to make the sentence more readable.

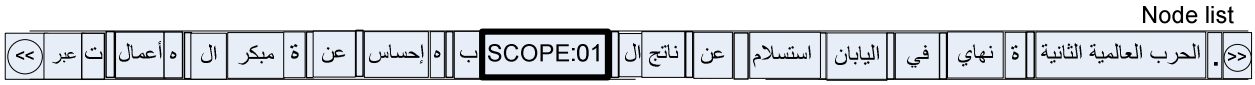


Figure (39)

After inserting a full stop by the end of the sentence in figure (39), the UNL expression is now fully generated to Arabic. At this moment, given the following input, the DeConverter gives the output that follows:

Input:

```
[S:15]
{unl}
obj(express(agt>thing,obj>abstract thing):0G.@entry.@past,   sense(icl>feeling):0U)
agt(express(agt>thing,obj>abstract thing):0G.@entry.@past,   work(icl>book):0A.@pl)
mod(work(icl>book):0A.@pl,        he:00)
mod(work(icl>book):0A.@pl,        early(mod<thing):04)
mod(sense(icl>feeling):0U,        :01)
mod(sense(icl>feeling):0U,        he:0Q)
obj(cause(aoj>thing,obj>thing):29.@past,    :01)
and:01(disorientation(icl>state):1Q.@entry.@def,     degradation(icl>phenomenon):1A.@def)
aoj(cause(aoj>thing,obj>thing):29.@past,    surrender(icl>phenomenon):2R)
tim(cause(aoj>thing,obj>thing):29.@past,    end(icl>time):38.@def)
mod(surrender(icl>phenomenon):2R,        Japan:2J)
mod(end(icl>time):38.@def,        World War II:3F)
{/unl}
[/S]
```

Output:

```
[S:15]
                    عبرت أعماله المبكرة عن إحساسه بالإذلال والارتباك الناتج عن استسلام اليابان في نهاية الحرب العالمية الثانية.
;;Time  0.2        Sec
```

To evaluate this output, the original sentence from which the UNL expression above is enconverted can be considered. The original sentence is: *"His early works expressed his sense of the degradation and disorientation caused by Japan's surrender at the end of World War II."* Comparing the original with the generated sentence above reflects that the generated sentence does not deviate, in neither the meaning nor the focus, from the original sentence. This underlies the high-quality output of the generation grammar.

6. Conclusion

Using the UNL system with its language components it has been proved to be a powerful environment for man machine communication. It enables natural language phenomena to be expressed in formal semantic framework which enables computers to understand natural language. If the UNL is added to the network platforms, the communication status will be changed. UNL will make the communication among people through different Natural Languages possible, which will share information and provide a common educational environment as language is an essential part of the communication process. Communication between different nations will be easier since language barriers will be broken. Breaking language barriers, in turn, will result in, for example, a) encouraging mutual understanding among different cultures which is one of the ultimate goals of UNL. Sure, using foreign languages will make nations go through the risk of loosing a big part of their culture; consequently, as time goes, their roots will be lost as well. With the existence of UNL this risk will not exist; b) communication through UNL will make the mission of international organizations, like United Nations and UNESCO, easier as they are concerned about all people with different mother tongues; one of the main problems faced in the exchange of information between the organizations and different nations is the existence of language barriers.

On the other hand, other machine translation systems will not be able to provide such environment for education and exchange of information as they are away from universality. They will never be inter-lingual. This will make their value limited to only the one or two languages involved in the translation. Consequently, communication and the distribution of information will be negatively affected. However one drawback can be that UNL has not yet conceived as a fully automatic machine translation system.

The Arabic language could successfully be generated from UNL hyper semantic networks with a high degree of accuracy. The main skeleton of Arabic sentence structure has been handled however many problems remain unsolved such as generating passive structures, correct ordering of modifiers of the same type, selecting the correct word representing universal words which represents the main challenges of the future work.

7. References

Al-Ansary, S. (2003). Building a Computational Lexicon for Arabic, presented in the in the *17th* **ALS Annual Symposium on Arabic Linguistics.** 9-10 March 2003, Alexandria, Egypt.

Al-Ansary, S. (2004a). Arabic - English Machine Translation Systems: Discrepancies and Implications. **JEP/TALN International Conference, Special session on Arabic text and speech language processing.** Fez, Morocco, 19-22 April 2004.

Al-Ansary, S. (2004b). A Morphological Analyzer and Generator for Arabic: Covering the Derivational Part. **NEMLAR International Conference on Arabic Language Resources and Tools**, Cairo, Egypt.

Al-Ansary, S., Nagi, M. and Adly, N. (2006). Processing Arabic Content: The Encoding Component of an Interlingual System for Man-Machine Communication in Natural Language, **the 6th International Conference on Language Engineering**, 6-7 December, Cairo, Egypt.

Arnold, D. (1994). **Machine Translation: An Introductory Guide.** Manchester, NCC Blackwell.

Auh T. (2001). Language Divide and Knowledge Gap in Cyberspace: Beyond Digital Divide**. International Symposium on Language in Cyberspace**, 26 - 27 September 2001, Seoul, Korea.

Beesley, K. R. and L. Karttunen (2003). **Finite State Morphology**. Stanford, Calif., CSLI; [Bristol : University Presses Marketing, distributor].

Dorr, B. J. (1993). **Machine Translation: A View from the Lexicon**. Cambridge, Mass., MIT Press.

Eldakar Y., Adly N., and Nagi M. (2006). **A Framework for the Encoding of Multilayered Documents**, accepted for publication at First International Conference on Digital Information Management (ICDIM 2006), Bangalore, December 6-8, 2006.

Eldakar Y., El-Gazzar K., Adly N., and Nagi M.(2005). The Million Book Project at Bibliotheca Alexandrina, **Journal of Zhejiang University SCIENCE**, vol. 6A, no. 11, pp. 1327-1340, Nov. 2005. and in Proceedings of International Conference on Digital Libraries 2005 (ICUDL05), Hangzhou, China, pp. Nov. 2005. Available: www.zju.edu.cn/jzus/2005/A0511/A051122.pdf

Galinski C. (2001). Dialogue among Civilizations in the Cyberspace**. International Symposium on Language in Cyberspace**, 26 - 27 September 2001, Seoul, Korea.

Hausser, R. R. (1999). **Foundations of Computational Linguistics: Man-machine Communication in Natural Language**. Berlin ; New York, Springer.

Hausser R. (2001). Human-Computer Communication in Natural Language. **International Symposium on Language in Cyberspace**, 26 - 27 September 2001, Seoul, Korea.

Kiraz, G. A. (1996). **Computational Approach to Non-linear Morphology,** University of Cambridge.

Kiraz, G. A. (2001). **Computational Nonlinear Morphology : With Emphasis on Semitic Languages.** Cambridge, Cambridge University Press.

Klavans (1997). Computational Linguistics. In O' Grady W., Dobrovolsky M. and Katmba F. (eds), **Contemporary Linguistics: An Introduction.** Longman.

Koskenniemi, K. (1983). **Two-level Morphology: A General Computational Model for Word-form Recognition and Production**. Helsinki, University of Helsinki, Department of General Linguistics.

Montviloff V.(2001**).** Meeting the Challenges of Languae Diversity in the Information Soceity. **International Symposium on Language in Cyberspace**, 26 - 27 September 2001, Seoul, Korea.

Nirenburg, S., H. L. Somers, et al. (2003). **Readings in Machine Translation.** Cambridge, Mass. London, MIT.

Saleh I. Adly N. and Nagi M. (2005). DAR:A Digital Assets Repository for Library Collections, **9th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2005)**, Vienna, pp. 116-127, Sep. 2005.

Uchida H. (1996).**UNL: Universal Networking Language – An Electronic Language for Communication, Understanding, and Collaboration**. UNU/IAS/UNL Center. Tokyo, Japan.

Uchida H. (2001). The Universal Networking Language Beyond Machine Translation**. International Symposium on Language in Cyberspace**, 26 - 27 September 2001, Seoul, Korea.

Uchida H., Zhu M. (2002a). Universal Word and UNL Knowledge Base**, International Conference on Universal Knowledge and Language (ICUKL)**, Goa, India.

Uchida H. (2002b). How to Build Universal Knowledge, **International Conference on Universal Knowledge and Language (ICUKL)**, Goa, India.

Uchida H.(2003). Knowledge Description Language, **Semantic Computing workshop,** Tokyo, Japan.

Uchida H., Zhu M. (2005). UNL2005 for Providing Knowledge Infrastructure, **SeC2005 Workshop**, Chiba, Japan.

Adorni G., M. Zock (eds.)(1996). **Trends in Natural Language Generation: An Artificial Intelligence Perspective,** Berlin, Heidelberg: Springer.

McDonald D. D. (1992): Natural-Language Generation. In: S. C. Shapiro (ed.), **Encyclopedia of Artificial Intelligence,** 2nd edition, New York: Wiley, pp. 983-997.

Zock, M. and G. r. Sabah (1988). **Advances in natural language generation : an interdisciplinary perspectiv**e. London, Pinter.

Konrad, K. (2004). **Model generation for natural language interpretation and analysis**. Berlin ; London, Springer.

Reiter, E. and R. Dale (2000). **Building natural language generation systems.** Cambridge, Cambridge University Press.

Cahill, L., C. Doran, et al. (1999). **Towards a reference architecture for natural language generation systems**, Human Communication Research Centre.

Adorni, G. E. and M. E. Zock (1996). **Trends in natural language generation : an artificial intelligence perspective**, Springer.

Paris, C. L., W. R. Swartout, et al. (1991). Natural language generation in artificial intelligence and computational linguistics : **4th International workshop on natural language generation** : Papers, Kluwer Academic Publishers.

Dale, R., C. S. Mellish, et al. (1990). **Current research in natural language generation.** London, Academic.

Cole R. A., Mariani J., Uszkoreit H., Zaenen A., Zue V. (eds.) (1996): **Survey of the State of the Art in Human Language Technology**. Kluwer, Dordrecht.