A generic framework for Arabic to English machine translation of simplex sentences using the Role and Reference Grammar linguistic model

By

Yasser Salem *B.Sc*

Supervisors:

Dr. Brian Nolan

Mr. Arnold Hensman

Submitted in Fulfillment of the Requirements for the of M.Sc. in Computing in the School of Informatics and Engineering at the Institute of Technology Blanchardstown

Dublin, Ireland

April, 2009

Abstract

The aim of this research is to develop a rule-based lexical framework for Arabic language processing using the Role and Reference Grammar linguistic model. A system, called UniArab is introduced to support the framework. The UniArab system for Modern Standard Arabic (MSA), which takes MSA Arabic as input in the native orthography, parses the sentence(s) into a logical meta-representation, and using this, generates a grammatically correct English output with full agreement and morphological resolution. UniArab utilizes an XML-based implementation of elements of the Role and Reference Grammar theory, and its representations for the universal logical structure of Arabic sentences.

Role and Reference Grammar (RRG) is a functional theory of grammar that posits a direct mapping between the semantic representation of a sentence and its syntactic representation. The theory allows a sentence in a specific language to be described in terms of its logical structure and grammatical procedures. RRG creates a linking relationship between syntax and semantics, and can account for how semantic representations are mapped into syntactic representations. We claim that RRG is highly suitable for machine translation of Arabic via an Interlingua bridge implementation model. RRG is a mono strata–theory, positing only one level of syntactic representation, the actual form of the

sentence and its linking algorithm can work in both directions from syntactic representation to semantic representation, or vice versa. In RRG, semantic decomposition of predicates and their semantic argument structures are represented as logical structures. The lexicon in RRG takes the position that lexical entries for verbs should contain unique information only, with as much information as possible derived from general lexical rules. For this reason and due to the functional nature of our linguistic model, we will create our own lexicon.

We use the RRG theory to motivate the architecture of the lexicon and the RRG bidirectional linking system to design and implement the parse and generate functions between the syntax-semantic interfaces. Through an input process with seven phases, including morphological and syntactic unpacking, UniArab extracts the universal logical structure of an Arabic sentence. Using the XML based metadata representing the RRG logical structure (XRRG), UniArab accurately generates an equivalent grammatical sentence in the target language through four output phases. We outline the conceptual structure of the UniArab System which utilizes the framework and translates the Arabic language into another natural language. We follow the Interlingua design approach for machine translation. We analyse the Arabic sentences to create a universal, abstract logical representation, and from this representation we generate English translations.

We also explore how the characteristics of the Arabic language will affect the development of a Machine Translation (MT) tool. Several characteristics of Arabic pertinent to MT will be explored in detail with reference to some potential difficulties that they present. We will conclude with a proposed model incorporating the Role and Reference Grammar techniques to achieve this end. The UniArab system has been tested by generating equivalent grammatical sentences, in English, via the universal logical structure of Arabic sentences, based on MSA Arabic input with very significant and accurate results.

It provides more accurate translations when compared with automated translators from Google and Microsoft though these systems have a much wider coverage than UniArab at present. The free word order nature of Arabic and the challenges of incorporating transitivity into the logical structure will be outlined in detail. This research demonstrates the capabilities of the Role and Reference Grammar as a base for multilingual translation systems.

Declaration

I herby certify that this material, which I now submit for assessment on the programme of study leading to the award of M.Sc. in Computing in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated, and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfillment of the requirements of that stated above.

. . . . . .

Yasser Salem

Dublin, Ireland

April 2009

Acknowledgements

"In the name of GOD (Allah), Most Gracious, Most Merciful. Praise be to GOD, the Cherisher and Sustainer of the worlds; Most Gracious, Most Merciful; Master of the Day of Judgement. Thee do we worship, and Thine aid we seek. Show us the straight way," [The Quran: Al-Fatiha (The Opening)]. "and my success (in my task) can only come from GOD. In Him I trust and unto Him I turn (repentant)." [The Quran: Hud,88]. I am not able to fulfil the due thanks to GOD, but I seek his forgiveness and that GOD assists me in thanking His Majesty. "Say. Surely my prayer and my sacrifice and my life and my death are all for God, the Lord of the worlds." (The Quran: AL-Anaam, 162)

I am deeply indebted to my mother for accepting that I pursue my M.Sc. away from her. And May God bless my father's soul and join all of us in the paradise. I am grateful to my wife, Qamar, and my children, for their encouragement and understanding. Thanks to my brothers and sisters.

I am deeply indebted to my supervisor, Dr. Brian Nolan, for many insightful conversations during the development of the ideas in this thesis, and for helpful comments on the text. Dr. Nolan has supported me not only by providing research guidelines and

advices, but also academically and emotionally through the rough road to finishing this thesis. And during the most difficult times when writing this thesis, he gave me the moral support. I also extend my sincere gratitude to Mr. Arnold Hensman who has had a significantly positive influence on my development through his role as my research co-supervisor. His patience and willingness to discuss the minutiae of the different obstacles I encountered while working on this project were invaluable.

I also extend my sincere gratitude to all the staff of the Department of Informatics, Institute of Technology Blanchardstown, Dublin, Ireland, where I did my undergraduate and masters studies. I specially thank Mark Cummins, Gareth Curran, Dr. Kevin Farrell, Geraldine Gray, Dr. Anthony Keane, Laura Keyes, Margaret Kinsella, John Massey, Hugh McCabe, Dr. Simon McLoughlin, Orla McMahon, Daniel McSweeney, Frances Murphy, Tom Nolan, Michael O'Donnell, Luke Raeside, Stephen Sheridan and Dr. Matt Smith.

Thanks to all my friends who supported me during my studies, especially, and to name just a few Mansoor Bin Khalifa Al-Thani, Mohammed A. Attia, Khalid Buzakhar, Suhaib Fahmy, Mohamed Faouzi Gahbiche, James Kennedy, Essam Mansour, Eftaima Najjair and Aliye Peksen. I would like to express my gratitude to all those who gave me the possibility to complete this thesis.

. . . . . .

Yasser Salem

Dublin, Ireland

April 2009

Contents

List of Figures

List of Tables

1

Introduction

The following paragraph was translated from Arabic into English using the Google translator (Google 2009).

That rely entirely on machine translation ignores the fact that communication in the language of rights is an integral part of the context, and that the human is capable of understanding the context of the original text in a manner sufficient. Therefore can not be trusted after the machine translation programs, they could not analyze the context of the original version is similar to the human understanding of when listening to the same text.

It is clear that the paragraph cannot be easily understood, and a large amount of the information has been confused or mixed up. This shows the problems facing machine translation, and motivates our work. We believe that statistical machine translation has not achieved what people expected in terms of quality. Hence we wish to look at another method, building from the ground up to achieve higher quality translations.

Machine translation has yet to reach its potential within the translation market as a whole. Figures suggest that MT accounts for a small us $ 100 million portion of a us $ 10 billion translation market (Intelligence 2004). Many have suggested that the reason is the poor quality of results, hence it only makes sense when very large amounts of data need to be processed (Oren 2004). For the MT market to expand, it is necessary to improve the quality of results, which will then make it a viable alternative within the much bigger translation market.

Arabic is acquiring attention in the natural language processing (NLP) community because of its political importance and the linguistic differences between it and European languages. These linguistic characteristics, especially complex morphology, present interesting challenges for NLP researchers. According to Holes (2004) Arabic is the sole or joint official language in twenty independent Middle Eastern and African states: Algeria, Bahrain, Egypt, Iraq, Jordan, Kuwait, Lebanon, Libya, Mauritania, Morocco, Oman, Palestine, Qatar, Saudi Arabia, Somalia, Sudan, Syria, Tunisia, the United Arab Emirates and Yemen. Since the end of the nineteenth century, there have been large communities of Arabic speakers outside the Middle East, particularly in the United States and Europe. Arabic is also the language of Islam's holy book the Qur'an, and as such is the religious language of all Muslims. Arabic has been an official language of the United Nations alongside English, French, Spanish, and Chinese since 1 January 1971'(Holes 2004). There are a number of different Arabic words in languages such as Persian, Turkish, Urdu or Malawian. The words derived from Arabic that exist in Spanish, Portuguese, German, Italian, English or French are also numerous (Bateson 2003).

The aim of this research is to create an Interlingua Machine Translation (MT) system that will accept Arabic source sentences and generate English sentences, and to build a

high-quality translation technology that is adequate for text-to-text translation. In this research we build an Interlingua architecture in MT which translates efficiently. We consider semantic analysis and other disambiguation related to Arabic. This research also represents a starting point for the future implementation of a successful and complete Arabic MT engine. The hypothesis under investigation and main aims are to present an interlingua architecture, which is not only successful in translating simplex Arabic (intransitive, transitive, ditransitive and copula-like nominative) sentences to corresponding English sentences, but also does so in the most optimal way.

This research is the first contribution (not just for Arabic) that uses the Role and Reference Grammar (RRG) model as a basis for machine translation. This contribution shows how RRG can be used to deduce the logical structure of sentences and produce a lexical representation which can then be used as the interlingua bridge. The lexicon in RRG takes the position that lexical entries for verbs should contain unique information only, with as much information as possible derived from general lexical rules. This was the reason for creating our own lexicon since we need an RRG–based lexicon of the unique information of verbs and their logical structure.

UniArab stands for **Uni**versal **Arab**ic machine translator system. The UniArab system is a natural language processing application based on Role and Reference Grammar for translating the Arabic language into any other language, using an RRG based interlingua bridge. The UniArab system can understand the part of speech of a word, agreement features, number, gender and the word type. The syntactic parse unpacks the agreement features between elements of the Arabic sentence into a semantic representation (the logical structure) with the ‘state of affairs’ of the sentence. In the UniArab system we intend to have a strong analysis system that can unpack all information and its attributes. This

allows for a generalized target language to be generated from the logical structures. In this research we translate from Arabic to English only, with a view to translate from Arabic to any other target language in the future.

1.1 Motivation

The motivation for an Arabic-English translation tool is obvious when one considers that Arabic is the lingua franca of the Middle-Eastern world. Presently, 20 countries with a combined population of 450 million consider Standard Arabic as their national language. A simple test case during a study at Abu Dhabi University over three popular Arabic translation tools (Google, Sakhr's Tarjim and Systran) revealed little success in generating the correct meaning (Izwaini 2006). This research demonstrates the capabilities of Role and Reference Grammar as a base for multi-language translation.

1.2 Goals

The goal of our research towards an Arabic-to-English machine translation system is to create a system that translates simplex sentences of Modern Standard Arabic as a source language into English. Our goal is to build a system which can translate a wide variety of simple sentence types. We aim to make this system as scalable as possible by allowing users to add to the lexicon and later, in future research, to include complex sentences. To achieve this goal, it is essential to build a robust and accurate lexical system and machine translator. One of the steps we have to achieve is to generate the universal logical structure from a source sentence. The system should be capable of dealing with free word order which Arabic exhibits. This poses a significant challenge to MT due to the vast number of ways to express the same sentence in Arabic. Also, we must account for verbs that do not exist in Arabic like the copula verb 'to be' and the verb 'to have'. The system should deal with the transitivity of verbs (intransitive, transitive,

ditransitive). The Arabic language is written from right to left and has a unique letter shape. Words are written in horizontal lines from right to left. The letter shape depends on its position in the word; initial (prefix), medial (infix), final (suffix) or (Isolated). In technical linguistic terms, Arabic is a 'pro–drop' or 'pronoun–drop' language. It can define who takes the action by using conjugations. The pro–drop parameter is an aspect of grammar that allows subjects to be optional in some languages. That is, every inflection in a verb paradigm is specified uniquely and does not need to use independent pronouns to differentiate the person, number, and gender of the verb. The system should cover and solve the "pro–drop" challenge in Arabic.

1.3 Technologies

We introduce the main technologies used to support the development of the research presented in this thesis. These technologies are mainly the XML language and Java. The most recent recommendation of the XML language has been presented by Bray et al. (2008). XML has become the default standard for data exchange among heterogeneous data sources (Arciniegas 2000). The UniArab system allows data to be stored in XML format. This data can then be queried, exported and serialized into any format the developer wishes. The Java programming language is used to implement the logical structures. The primary advantage being that Java is platform-independent and thus highly suitable for MT.

Advantages of XML

XML is a generalized way to store data, which is not married to any particular technology. This makes it easy to store something, and then come back and grab it later with some other technology for processing. Using XML to exchange information offers a number of advantages, including the following:

Easily built: A well-formed data element must be enclosed between tags. The XML document can be parsed without prior knowledge of the tags. XML allows you to define all sorts of tags with all sorts of rules, such as tags representing data description or data relationships.

Human readable: Using intelligible tag names will make it possible to read, even by novices.

Machine readable: XML was designed to be easy for computers to process. XML is completely compatible with Java and portable platforms. Any application can process XML on any platform, as it is a platform-independent language.

XML fully supports Arabic: We chose to create our datasource as XML files, for optimum support of different platforms. It was also easier as we used Arabic letters rather than Unicode inside the datasource.

XML search engine: It is easy to extend the search sample to display more information about the search. Search by Java API Document Object Model (DOM) is the ideal tool for searching collections of XML documents.

1.4 Thesis organization

This thesis is organized as follows. Chapter 2 explores how the characteristics of the (Modern Standard) Arabic language will affect the development of an Arabic to English machine translation (MT) tool. Several distinguishing features of Arabic pertinent to MT are explored in detail (Salem et al. 2008b). Chapter 3 reviews the most important features of Role and Reference Grammar (RRG) Theory (Salem and Nolan 2009a). Chapter 4 will discuss some distinguishing features of Machine translation strategies. Chapter 5 presents the design of an Arabic to English machine translation framework based on RRG. It also presents a high-level view of the system framework and defines our evaluation criteria for measuring system performance and effectiveness (Salem and Nolan

2009b). Chapter 6 presents UniArab: a proof-of-concept Arabic to English machine translator system. It covers the technical aspects of UniArab, covering all the phases involved in the machine translation process. We describe the lexical system that underlies UniArab, detailing the attribute information held for each type of word. We discuss the input and generation phase and how the system maps the logical structure to a target English sentence. We then briefly discuss the user interface, and some of the technical challenges encountered during the implementation (Salem et al. 2008a) and (Nolan and Salem 2009). Chapter 7 discusses the evaluating and experimental results of the case study. We present the results of our evaluation of UniArab for a wide variety of simple (Intransitive, Transitive and Ditransitive) sentence types (Salem and Nolan 2009c). The thesis conclusions and future work are discussed in Chapter 8.

2

The Arabic Language

Arabic is a language with a derivational and inflectional rich morphology (Holes 2004). The version of Arabic we consider in this research is Modern Standard Arabic (MSA). When we mention Arabic throughout this research we mean MSA which is distinct from classical Arabic. Modern Standard Arabic (MSA) is a modernized form of Classical Arabic (Alosh 2005). MSA is the literary and standard variety of Arabic used in writing and formal speeches today (Schulz 2005). MSA is the universal language of the Arabic-speaking population. MSA is printed in most books, newspapers, magazines, official documents, and reading primers for children. Most of the oral Arabic spoken today is more divergent than the written Arabic language. Arabic words are often ambiguous in their morphological analysis (Al-Sughaiyer and Al-Kharashi 2004). As a language, Arabic is rich in morphological and syntactic structures. Arabic is also challenging in that it is a derivational or constructional language rather than a concatenative one. Words

like ‘go’ يذهب *ydhb* and ذهب *dhb*[*] can easily be seen as being part of a hierarchy of inheritance from a ‘specific root (in this case ذهب *dhb*). In English and in many other languages this is not usually the case. The Arabic language is written from right to left. It has 28 letters, many language specific grammar rules with a relatively free word order language. Each Arabic letter represents a specific sound so the spelling of words can easily be done phonetically. There is no use of silent letters as in English. Similarly, there is no need to combine letters in Arabic to indicate a certain sound. For example, the ’th’ sound in English as in the word ’Thinking’ is reduced in Arabic to the character ث *t*. In addition to the standard challenges involved in developing an efficient translation tool from Arabic to English, the relatively free word order nature of Arabic creates an obstacle. There is no copula verb ‘to be’ in Arabic, for example, the mere juxtaposition of the subject and predicate indicates the predicational relationship. The absence of the indefinite article, while not unique to Arabic still poses many difficulties within the context of the language structure.

2.1 Characteristics of the Arabic language

The copula verbs ‘to be’ and ‘to have’ do not exist in Arabic. Instead of saying ‘My name is Zaid’, the Arabic equivalent would read like ‘Name mine Zaid’ - إسمي زيد *ismy zyd*. Instead of saying ‘She is a student’, the Arabic equivalent would be ‘She student’; in Arabic هي طالبة *hy ṭālbh*. The copula in Arabic is only realised in the past and future tenses and in negation. Regarding the verb ‘to have’, which in English can also mean ‘to own’. Instead of saying “He has a house”, the Arabic equivalent is ‘To him a house’ - له بيت *lh byt*. Adjectives in Arabic have both a masculine and a feminine form. The singular feminine adjective is just like the masculine adjective but morphologically marked (Ryding 2007).

[*]Arabic examples are written here by using Buckwalter Arabic Transliteration which is converted in latex into the DIN 31635 standard of Arabic transliteration

Table 2.1: Dual: merely add two letters to achieve dual form in Arabic

Arabic	English Translation
باب *bāb*	door
بابان *bābān*	two doors

The Arabic number system includes the dual form, whereas other languages move from the singular to the plural form directly. In Arabic we need only to add two letters to the singular form to express the dual form. An example is given in Table 2.1. The plural form, however, is obtained using a different mechanism.

Plurals are of two types:

(1) The sound plural. The sound plural is one in which the singular form of the word remains intact (sound) with some addition at the end. Examples;

Masculine in the nominative case e.g. engineers مهندسون *mhndswn* in which ون *wn* is added to a singular noun. Masculine in the accusative and genitive cases e.g. engineers مهندسين *mhndsyn* in which ين *yn* is added to the singular noun.

Feminine in the nominative e.g engineers مهندساتٌ *mhndsātun* in which اتٌ *ātun* is added to the singular noun.

Feminine in the accusative and genitive cases engineersمهندساتٍ *mhndsātin* in which اتٍ *ātin* is added to the singular noun.

(2) The broken plural. The broken plural is one in which the form of the singular word is broken, that is, changed. It has no fixed rule for making it. Sometimes letters are added or deleted and sometimes there is merely a change in the vowels. Examples كتب *ktb* books, كتاب *ktāb* a book, رجل *rğl* man, رجال *rğāl* men, سنة *snh* a year سنوات *snwāt* years.

2.2 Characteristics of Arabic words

There is no upper and lower case distinction. Words are written horizontally from right to left. Most letters change form depending on whether they appear at the beginning, middle or end of a word or on their own. Arabic letters that may be joined are always joined in both hand-written and printed form.

An interesting feature of Arabic is its treatment of the demonstrative. Whereas in English one refers to an object that is either near or far as simply *this* (very near the speaker) or *that* (away from the speaker up to any distance), Arabic has a third demonstrative to specify objects that are in between these points on the distance spectrum.

Table 2.2: Grammatical gender

Arabic Masculine	**English Translation**
قمر *qmr*	moon
سيف *syf*	sword
باب *bāb*	door
Arabic Feminine	**English Translation**
شمس *šms*	sun
عصا *ʿṣā*	stick
نافذه *nāfdh*	window

In Arabic, all nouns must be either feminine or masculine, and the gender can be either grammatical or natural. The gender of inanimate objects is grammatical, examples are in Table 2.2. In this case the gender is a built-in lexical property of the word. Animate objects have a natural gender, and this gender can be either non-productive or productive. The non-productive gender is the case of nouns where the feminine and the masculine have different lexical entries, i.e., the feminine is not derived from the masculine, as in Table 2.3. By contrast, in the productive gender, the feminine is derived from the masculine, usually by adding a special suffix ‘ta marbuta’ to the end of the masculine

form, as in Table 2.4. The Arabic definite article is concatenated to nouns and adjectives. The shape of the definite article is shown in Table 2.5.

Table 2.3: Feminine is different than masculine

Arabic	English Translation
دَجَاجَة *dağaāğah*	Chicken
ديك *dyk*	Cock

Table 2.4: Feminine and masculine in Arabic

Arabic	English Translation
مُعلِمٌ *muʕilmun*	teacher(M)
مُعلِمةٌ *muʕlimtun*	teacher(F)
طالب *ṭālb*	student(M)
طالبة *ṭ ālbh*	student(F)

Table 2.5: Definiteness in Arabic

Arabic	English Translation
الـ *āl*	the

The definite article in Arabic is graphically prefixed to an Arabic noun. An example of Arabic definiteness is shown in Table 2.6.

Table 2.6: Definiteness example in Arabic

Arabic	English Translation
رجل *rğl*	a man
الرجل *ālrğl*	the man

2.2.1 Free word order

Arabic has a relatively free word order (Ramsay and Mansour 2006), this poses a significant challenge to MT due to the number of possible ways to express the same sentence in Arabic. For the elements of subject(S), verb(V) and object(O), Arabic's relatively free

word order allows the combinations of SVO, VSO, VOS and OVS. For example, consider the following word orders:

(1) Noun1 Verb Noun2

(2) Noun2 Verb Noun1

(3) Verb Noun1 Noun2

(4) Verb Noun2 Noun1

Table 2.7: Free word order

(a) Noun Verb Noun example.

قيس يحب ليلى *qys yḥb lylā*		
Qays loves Laila		
ليلى *lylā*	يحب *yḥb*	قيس *qys*
Laila	loves	**Qays**
noun	verb	noun

(b) Verb Noun Noun example.

يحب قيس ليلى *yḥb qys lylā*		
Qays loves Laila		
ليلى *lylā*	قيس *qys*	يحب *yḥb*
Laila	**Qays**	loves
noun	noun	verb

(c) Verb Noun Noun example.

يحب ليلى قيس *yḥb lylā qys*		
Qays loves Laila		
قيس *qys*	ليلى *lylā*	يحب *yḥb*
Qays	Laila	loves
noun	noun	verb

This means that we have a challenge to identify exactly which is the subject and the object. Tables 2.7(a), 2.7(b) and 2.7(c) show this challenge. In Arabic the subject agrees with the verb with appropriate morphological marking on the word to differentiate subject from object in these free word order sentences. †

The difference in Tables 2.7(a), 2.7(b) and 2.7(c) is the position of the actor. The sentences in fact have the same meaning. While in English the form of a sentence is subject verb object.

†Note that Arabic sentences should be read from right to left.

2.3 Part of speech inventory of the Arabic language

In the Arabic linguistic tradition there is not a clear-cut, well-defined analysis of the inventory of parts of speech in Arabic. Attia (2008) mentioned that the traditional classification of Arabic parts of speech into nouns, verbs and particles is not sufficient for a complete computational grammar. This categorization, originally proposed by Sibawaih (Owens 2006), remains the standard accepted scheme today. However, we have found it lacking when applied to machine translation, and so, developed our own lexical scheme. Our classification of the parts of speech in Arabic is illustrated in Figure 2.1. We classified parts of speech into nouns, adjectives, adverbs, verbs, demonstratives, and others. Each category will be further explained in the following subsections.

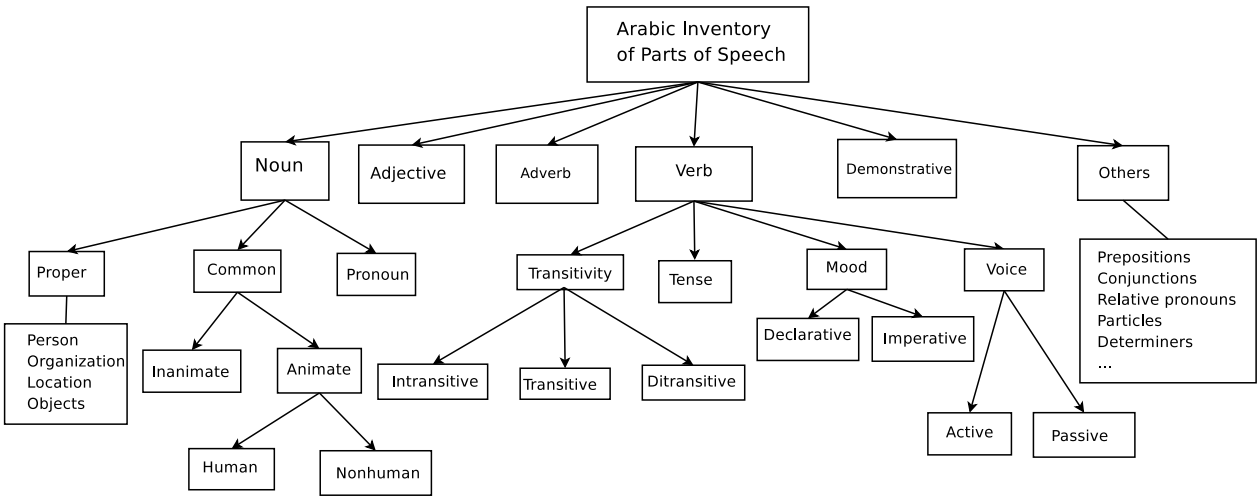

Figure 2.1: A classification for the Arabic language syntax

2.3.1 Noun

A noun denotes either tangible or intangible identities. Nouns are independent of other words in indicating their meaning. What distinguishes nouns from verbs is that nouns refer to entities or things. Nouns are further classified into pronouns, common nouns and proper nouns. Pronouns are classified according to person (first, second, third), number

(singular, dual and plural) and gender (masculine and feminine). They can also be nominative, accusative, or genitive. Examples are أنا *ʾanā* "I", أنت *ʾant* "you", and هو *hw* "he". We make a further classification of common nouns into animate and inanimate. Examples of common noun are in Table 2.8.

Table 2.8: Noun example in Arabic

Arabic	English Translation
رَجُلٌ *raǧulun*	man
شَجرہ *šaǧrh*	tree

Although this seems more like a semantic classification, the Arabic morphology and syntax needs this classification. For example the choice of demonstrative adjective with plural nouns depends on whether the noun is human or non-human. For example

هذه الكلاب *hdh ālklāb*

this.sg.f dog.pl

The proper nouns are further classified into names of persons, such as عمر *ʿmr* "Omar" and خالد *ẖālد* "Khalid"; locations, such as القاهرة *ālqāhrh* "Cairo" and أيرلندا *ʾayrlndā* "Ireland"; organizations الأمم المتحدة *ālʾamm ālmtḥdh* "United Nations"; and objects, such as لينوكس *lynwks* "Linux" Common nouns can either be definite or indefinite.

2.3.1.1 Definite nouns

A noun normally can be considered as definite (in Arabic: معرفه *mʿrfh*) when the speaker and the reader know about the specific object being referred to, for example in Table 2.9.

Table 2.9: Definite example in Arabic

Arabic	الكتاب الذي تبحث عنه فوق الطاوله. *ālktāb ālדy tbḥṯ ʿnh fwq ālṭāwlh.*
English Translation	The book you are looking for is on the table.

In the example, the word 'book' is definite by using the definite article 'the', since both the speaker and the listener know which book they are dealing with. The definite article

in Arabic is used to introduce and talk about a known subject. The Arabic language uses the same article for all nouns, be they male or female, singular or plural. The article is written before the noun it refers to and, graphically, it appears attached to it.

2.3.1.2 Indefinite nouns

Indefinite nouns (in Arabic: نكره *nkrh*) are nouns which are not specified. It is translated as 'a' or 'an' in English, e.g. *a man, an apple, water*. There is no need to translate it everywhere as in the example of *water*. The absence of the indefinite article is, as in Table 2.10, a potential source of problems for Arabic-English machine translation.

Table 2.10: Indefinite example in Arabic

Arabic	وجدت كتابا على الطاوله هل هو لك؟ *wǧdt ktābā ʕlā āl̩tāwlh hl hw lk?*
English	I found (a) book on the table, is it yours?

2.3.2 Adjectives

Adjectives are used to modify nouns. Arabic adjectives agree with nouns in number, gender, definiteness and case. An example is the adjective "useful", in Table 2.11.

Table 2.11: Arabic adjective

Arabic	**English Translation**
قرأت كتابا نافعا *qrʔat ktābā nāfʕā*	I read a useful book

2.3.3 Adverbs

Adverbs are used to modify verbs. They can be adverbs of place, time or manner. An example in Table 2.12.

Table 2.12: Arabic adverb

Arabic	**English Translation**
عقد الاجتماع مساء *ʔqd ālāǧtmāʿ msāʾ*	The meeting was held in the evening

2.3.4 Verbs

A verb describes both an action and tense. There are four ways to classify verbs in Arabic: according to tense, transitivity, mood and voice:

2.3.4.1 Verb tenses

There are mainly two tenses in Arabic: the imperfect and the perfect.

The imperfect tense الفعل المضارع *ālfʕl ālmḍārʿ*, which indicates that an action has not yet been completed but is being done or will be done; something that is happening at the moment. An example is shown in Table 2.13.

Table 2.13: Imperfect tense الفعل المضارع *ālfʕl ālmḍārʿ*

Arabic	English Translation
يَكْتُبُ *yaktubu*	he is writing.

The perfect tense 'ماضي *māḍy*, which indicates that an action has been completed. An example is shown in Table 2.14.

Table 2.14: Perfect tense الفعل الماضي *ālfʕl ālmāḍy*

Arabic	English Translation
كَتَبَ *kataba*	he wrote.

Both the perfect and imperfect tenses can be modified by thirteen inflectional forms which depend on person, mood and number. Table 2.15 shows these forms applied to the imperfect, and Table 2.16 shows the thirteen person markers for the perfect tense.

The word 'سوف *swf*' if it is before the imperfect tense then the verb has a future meaning. Graphically a word like this will look like [*sawfa* + imperfect] or [*s* + imperfect] similar to the example in Table 2.17.

In Arabic, it is possible to combine the verb kaana كان *kān* with the main verb to indicate past progressive. This is where an action took place in the past but happened

Table 2.15: Imperfect inflectional forms of word 'write'

	Singular	Dual	Plural
First Person	نكتبُ *nktbu*		أكتبُ *ʾaktbu*
Second Person (m)	تكتبونَ *tktbwna*	تكتبانِ *tktbāni*	تكتبُ *tktbu*
Second Person (f)	تكتبْنَ *tktbna*	تكتبانِ *tktbāni*	تكتبنَ *tktbna*
Third Person (m)	يكتبونَ *yktbwna*	يكتبانِ *yktbāni*	يكتبُ *yktbu*
Third Person (f)	يكتبْنَ *yktbna*	تكتبانِ *tktbāni*	تكتبُ *tktbu*

Table 2.16: Perfect inflectional forms of word 'wrote'

	Singular	Dual	Plural
First Person	اُكتبْنا *uktbnā*		كتبْت *ktbt*
Second Person (m)	كتبْتُم *ktbtum*	كتبْتُما *ktbtumā*	كتْبتَ *ktbta*
Second Person (f)	كتبْتُنَّ *ktbtuna*	كتبْتُما *ktbtumā*	كتبْتِ *ktbti*
Third Person (m)	كتبوا *ktbwā*	كتبا *ktbā*	كتبَ *ktba*
Third Person (f)	كتبْنَ *ktbna*	كتبَتا *ktbatā*	كتبَتْ *ktbat*

Table 2.17: Future tense in Arabic

Arabic	English Translation
سوف يَكْتُبُ *swf yaktubu*	he will write
سيَكْتُبُ *syaktubu*	he will write

over a long period, or represents a state of being. This construct is used when talking about knowledge of something in the past. In Arabic, the past perfect progressive is actually indicated using the present tense and the particle mundhu منذ *mnḏ*. e.g. أعيش هنا منذ خمس سنوات *ʾayš hnā mnḏ ḫms snwāt* I have been living here for five years. Future perfect in Arabic is indicated using the present tense of kaana with a past tense main verb. e.g. سيكون قد أنهى دراسته *sykwn qd ʾanhā drāsth* he will have finished his studies.

2.3.4.2 Aspect

Tense deals with when an action occurs, aspect determines whether the action has been completed, is ongoing or is yet to occur. In Arabic, tense and aspect are generally blended together, that is why past/present are often switched with perfect/imperfect in discussion. For a larger discussion on the sentax the tense and aspect refer to Ryding (2007).

2.3.4.3 Mood

Mood is reflected in Arabic in word structure, and so analysis is a part of the morphology. The mood can be indicative, subjunctive, imperative or jussive. The indicative are straightforward statements, the subjunctive includes the attitude towards actions, the imperative indicates a command. Mood marking is only done on the present tense. There are no markings for past tense. Examples of the four moods are shown in Tables 2.18, 2.19, 2.20 and 2.21.

Table 2.18: Indicative mood

Arabic	نرحب بزبائننا *nrḥb bzbāʾynnā*
English	**we welcome** our customers.
Arabic	يغادر دبلن اليوم *yġādr dbln ālywm*
English	**He leaves** Dublin today.

Table 2.19: Subjunctive mood

Arabic	يجب أن نقوم بزيارة *yǧb ʾan nqwm bzyārh*
English	It is necessary that **we undertake** a visit.

Table 2.20: Jussive mood

Arabic	لم نأت *lm nʾat*
English	we did not **come**.
Arabic	إصلاحات لم تكتمل منذ عامين *ʾiṣlāḥāt lm tktml mnḏ ʿāmyn*
English	renovations that **have not been completed** for two years

Table 2.21: Imperative mood

Arabic	افتح يا سمسم *āfṭḥ yā smsm*
English	**Open**, Semsame.
Arabic	اسمح لي *āsmḥ ly*
English	**Permit** me.
Arabic	لا تنس *lā tns*
English	Do not **forget**.

2.3.4.4 Voice

The voice in Arabic is indicated by inflection on the verbs and differentiates between active and passive, as shown in the contrast between قال *qāl* "[he] said" and يقال *yqāl* "[it is] said".

2.3.4.5 Transitivity

In Arabic and English, we can classify verbs as either intransitive, transitive or ditransitive.

(1) **Intransitive (**اللاَّزِم *āllāazim***)**

An intransitive verb is unable to take an object; it exists alone. Intransitive verbs include يسبح *ysbḥ* , swim, يأكل *yʾakl* , ate, مات *māt* ,die, نائم *nāʾym* sleep.

Some verbs can be both transitive and intransitive:

أنا فزت *ʾanā fzt* , I won. (Intransitive)

أنا فزت بالجائزة الأولى *ʾanā fzt bālǧāʾyzh ālʾawlā* , I won the first prize. (Transitive)

(2) **Transitive (**المتعدي *ālmtʿdy***)**

A transitive verb takes one or more objects (an object, or undergoer of the verb). For example; إشترى عمر كتاب *ʾištrā ʿmr ktāb* , Omar bought a book. أكتب رساله *ʾaktb rsālh* , I write a letter. أبو بكر وضع الكتاب فوق مكتبه *ʾabw bkr wḍʿ ālktāb fwq mktbh*, Abu Bakr put the book on his disk.

A transitive verb is incomplete without a direct object. For example;

Incomplete: خالد يحمل *ẖālد yḥml* , Khalid holds.

Complete: خالد يحمل ثلاثة كتب و حاسوبه الشخصي و زهور *ẖālد yḥml ṯlāṯh ktb w ḥāswbh ālšẖṣy w zhwr* , Khalid holds three books, his laptop and flowers.

(3) **Ditransitive**

A ditransitive verb takes two objects. This can be through an indirect object construction, عصام أعطى كتاب لعمر *ʿṣām ʿaṭā ktāb lʿmr* , Essam gave a book to Omar

Or double object construction, عصام أعطى عمر كتاب *ʿṣām ʿaṭā ʿmr ktāb* , Essam gave Omar a book.

2.3.5 Demonstratives

The demonstrative pronouns in Arabic include reference for the near هذا *hḏā* "this", the far ذلك *ḏlk* "that" and for the inbetween ذاك *ḏāk*, which has no equivelent in English.

2.3.6 Others

This class includes all other types of words not included in the previous categories. It includes, for example, the prepositions, such as من *mn* "from", على *ʿlā* "on", في *fy* "in", إلى *ʿilā* "till". It also includes conjunctions, such as و *w* "and"; determiners such as الـ *āl* "the"; relative pronouns, such as الذي *ālḏy* "who (masculine)" and التي *ālty* "who (feminine)" and particles, such as لن *ln* "Will not" (Khan 2007).

Table 2.22: Particle 'Lan'

Arabic	Arabic Meaning	English Translation
لَنْ يَذْهَبَ *lan yaḏhaba*	will not 'he' go	he will not go

The particle لن *ln* is used to negate future events. It is used within the imperfect tense (Versteegh 2001). An example is shown in Table 2.22.

2.4 Sentence types in Arabic

A sentence is a string of words that expresses a semantically complete message. There are two main sentence types in Arabic: verbal sentences and equational or copula sentences. The classification of clauses in the Arabic language is illustrated in Figure 2.2.

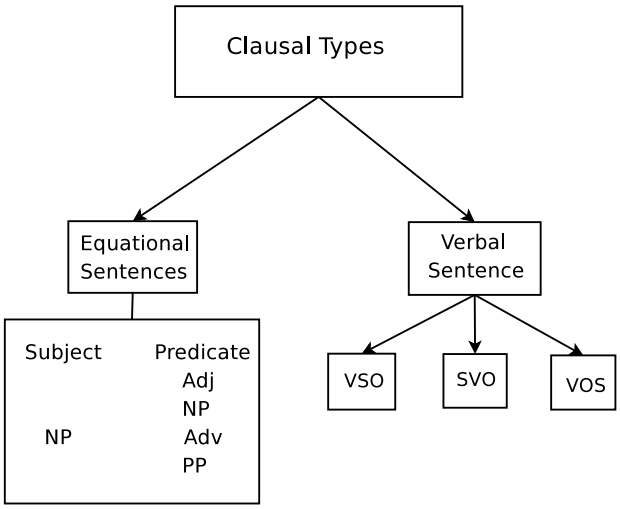

Figure 2.2: A classification of clauses in the Arabic language

2.4.1 Equational sentences

Equational sentences contain two parts (subject and predicate). In Arabic the copula verb 'to be' is not used in the present tense. Both the subject and the predicate have to be in the nominative case if they are not preceded by إن *ʾin* "indeed" or كان *kān* "was" (Abn-Aqeal 2007). In Table 2.23 the predicate in the first example is realized as a noun phrase, in the second example as an adjective, and in the third example as a preposition. The subject and predicate can serve as arguments for other verbs as will be shown in the following subsections.

Table 2.23: Nominal sentence

Arabic	English Translation
زَيْدٌ طَالِبٌ *zaydun ṭaālibun*	Zaid is (a) student.
زَيْدٌ كريم *zaydun krym*	Zaid is generous.
زَيْدٌ في البيت *zaydun fy ālbyt*	Zaid is in the house.

2.4.1.1 Verb and noun

Verb and noun. such as: سأل أحمد *sʾal ʾaḥmd* Ahmad asked.

2.4.1.2 Verb and two nouns

It only occurs in one construct كان و أخواتها *kān w ʾaḫwāthā* kan and its sisters. The verb kna كان *kān* and its sister verbs mark the time or duration of actions, states, and events. Sentences that use these verbs are considered to be a type of nominal sentence according to Arabic grammar, not a type of verbal sentence. The word order resembles Verb Subject Object when there is no other verb in the sentence,

They are كان *kān* was, صار *ṣār* to become, أصبح *ʾaṣbḥ* to become, أضحى *ʾaḍḥā* to become, أمسى *ʾamsā* to become, ظل *ẓl* to remain, بات *bāt* to be, ليس *lys* it is not. English can not express the punctual and telic aspectual differences encoded within the Arabic examples just mentioned.

Table 2.24: Kan and its sisters كان و أخوتها *kān w ʾaḫwthā*

Arabic	English Translation
كان الأكلُ لذيذاً *kān ālʾaklu ldydāan*	The food was delicious.

With these verbs the subject is in the nominative case and the predicate is in the accusative case, an example is shown in Table 2.24.

2.4.1.3 Verb and three nouns

It only occurs in one construct ظن و أخواتها *ẓn w ʾaḫwāthā* anna and its sisters. Both the subject and the predicate of ظن *ẓn* and its sisters are in an equational clause.

They are ظن *ẓn* to guess or to think, حسب *ḥsb* to consider, علم *ʿlm* to learn (about), جعل *ǧʿl* to make, صير *ṣyr* to make. They usually come before the nominal sentences 'subject and a predicate', an example is in Table 2.25. English can not express the semantic and causative 'make' differences encoded within the Arabic examples just mentioned.

Table 2.25: zanna and its sisters ظن و أخوتها *ẓn w ʾaḫwthā*

Arabic	ظن أحمد القياده سهلة. *ẓn ʾaḥmd ālqyādh shlt.*
English Translation	Ahmad thinks leadership is easy.

2.4.1.4 Verb and four nouns

This clausal type is used in classical Arabic, but not in MSA. It is mentioned here only for the sake of completeness. It has one type in أعلم و أري *ʾʿlm w ʾary* informed and showed. They are أعلم *ʾʿlm* informed, أري *ʾary* showed, أنبأ *ʾanbʾa* told. نبّأ *nbʾa* told, أخبر *ʾaḫbr* told, خبّر *ḫbr* told. حدّث *ḥdṯ* talked. أعلم *ʾʿlm* when it has hamza above it can has four nouns (ibn Abd Allah Ibn Malik 1984), such as in Table 2.26.

Table 2.26: Informed and showed

Arabic	أعلمتُ عمراً خَالِداً تلميذاً *ʾʿlmtu ʿmrāan ḫaālidāan tlmyḏāan*
English Translation	I informed Omer that Khalid (is) a student.

2.4.2 The Verbal Sentence

The verbal sentence is the second type of sentence in Arabic. It contains a verb and one or more participants depending on the verb transitivity. The default word order in Arabic is to begin with a verb: verb(V), subject(S) and object(O), such as in Table 2.27

Table 2.27: verb(V), subject(S) and object(O)

Arabic	شرب خالد اللبن *šrb ẖāld āllbn*
Gloss	drank Khalid the-milk
English Translation	Khalid drank the milk.

Another possible word order is to start with the subject, i.e. SVO, such as in Table 2.28

Table 2.28: subject(S), verb(V) and object(O)

Arabic	خالد شرب اللبن *ẖāld šrb āllbn*
Gloss	Khalid drank the-milk
English Translation	Khalid drank the milk.

Another possible, but more restricted, word order is VOS, such as in Table 2.29

Table 2.29: verb(V), object(O) and subject(S)

Arabic	شربه خالد *šrbh ẖāld*
Gloss	drank it Khalid
English Translation	Khalid drank it

The OVS word order is perfectly acceptable in Classical Arabic but no longer occurs in MSA (Attia 2008).

2.4.3 Clause

A clause in Arabic may be simple or complex. A complex clause is formed by conjoining two simple clauses by subordinating conjunction, such as in Table 2.30.

Table 2.30: Two simple clauses by subordinating conjunction

Arabic	شرب خالد اللبن قبل أن يذهب إلى المدرسة *šrb ẖāld āllbn qbl ʾan ydhb ʾilā ālmdrsh*
English	Khalid drank the milk before he went to school.

2.5 Summary

We have shown that Arabic is a language of increasing importance in the modern world. As a language it is fundamentally different from European languages and has many

unique features. Considerations such as its derivational structure, its distinction of gender forms, and its numerous sentence orders present a challenge for automatic machine translation. We discussed an inventory of the language including examples. In order to deal with these challenges it is important that a machine translator understands the structure of the source language. We aim to use this knowledge to build the UniArab translator. In order to provide a standards-based, cross-platform solution, we will make use of XML for data representation and build the system using Java.

3

Role and Reference Grammar (RRG)

This chapter is based largely on material taken from (Van Valin and LaPolla 1997), which explains the theory behind Role and Reference Grammar. Role and Reference Grammar (RRG) is a model of grammar developed by William Foley and Robert Van Valin, Jr. in the 1980s, which incorporates many of the points of view of current functional grammar theories. We have chosen RRG because it has been shown to be flexable and universal in the creation of parsers for English (Van Valin and LaPolla 1997). We wish to apply this success to MT in order to discover its importance and demonstate its viability with accuracy of translation.

In RRG, the description of a sentence in a particular language is formulated in terms of its logical structure and communicative functions, and the grammatical procedures that are available in the language for the expression of these meanings. The main features of RRG are the use of lexical decomposition, based upon predicate semantics, an analysis of clause structure and the use of a set of thematic roles organized into a hier-

archy in which the highest-ranking roles are 'Actor' (for the most active participant) and 'Undergoer' (Van Valin 1993). RRG takes language to be a system of communicative social action, and accordingly, analysing the communicative functions of grammatical structures plays a vital role in grammatical description and theory from this perspective. Role and Reference Grammar posits algorithms to go from syntax to semantics and semantics to syntax. The main contribution is the use of parsing templates and the notion of the core. A core consists of a predicate (generally a verb) and (normally) a number of arguments. It must have a predicate. Everything else is built around one or more cores. Simple sentences contain a single core; complex sentences contain several cores. The fact that RRG focuses on cores, means that the semantics is relatively easy to extract from a parse tree. You just have to look for the (PRED), and (ARG) branches of the core to obtain the predicate (PRED) and the arguments (ARG). Who did what to whom will depend either on the ordering of the ARG branches (in the case of English), or on their cases, or both.

3.1 Role and Reference Grammar linguistic model

Role and Reference Grammar is a model which presupposes a direct mapping between the semantic representation of a sentence and its syntactic representation; there are no intermediate levels of representation (Van Valin 2007). The general view of RRG is presented in Figure 3.1.

RRG creates a relationship between syntax and semantics and can account for how semantic representations are mapped into syntactic representations. RRG also accounts for the very different process of mapping syntactic representations to semantic representations. Before developing the linking algorithms that govern these mappings, it is necessary to first introduce a general principle constraining these algorithms (Van Valin and LaPolla 1997). Of the two directions, syntactic representation to semantic represen-

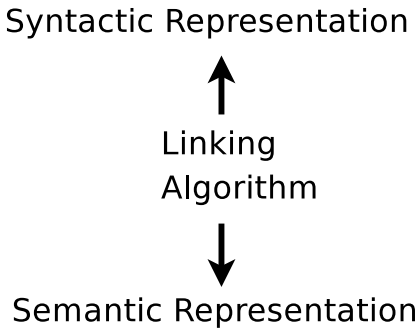

Figure 3.1: Layout of Role and Reference Grammar

tation is the more difficult since it involves interpreting the morphosyntactic form of a sentence and inferring the semantic functions of the sentence from it. Accordingly, the linking rules must refer to the morphosyntactic features of the sentence. The question remains why a grammar should deal with linking from syntax to semantics at all. Simply specifying the possible realizations of a particular semantic representation should suffice. They refute this using the argument that theories of linguistic structure should be directly relatable to testable theories of language production and comprehension (Van Valin and LaPolla 1997). One of our hypotheses it that RRG is very suitable for machine translation of Arabic via an interlingua bridge. It is a mono strata-theory, positing only one level of syntactic representation, the actual form of the sentence. The RRG Linking algorithm can work in the both directions from syntactic representation to semantic representation or vice versa. UniArab will fulfil this role. In RRG, semantic decomposition of predicates and their semantic argument structures are represented as logical structures. The lexicon in RRG takes the position that lexical entries for verbs should contain unique information only, with as much information as possible derived from general lexical rules. We briefly illustrate the active voice linking in (3.1) where (3.1a) is a subject, verb, object (SVO) clause and (3.1b) is the verb, subject, object (VSO) equivalent.

(3.1)

a. زيد رأي عمر *zyd rʕay ʔmr* Zaid saw Omar

زيد *zyd* MsgNOM see.past عمر *ʔmr* - MsgNOM

b. رأى زيد عمر *rʕaā zyd ʔmr* Saw Zaid Omar

see.past زيد *zyd* MsgNOM عمر *ʔmr* MsgNOM

Arabic allows variation in clause word order. The active-voice linkings, those in the sentence in (3.1a)-(3.1b), are illustrated in figure 3.2.

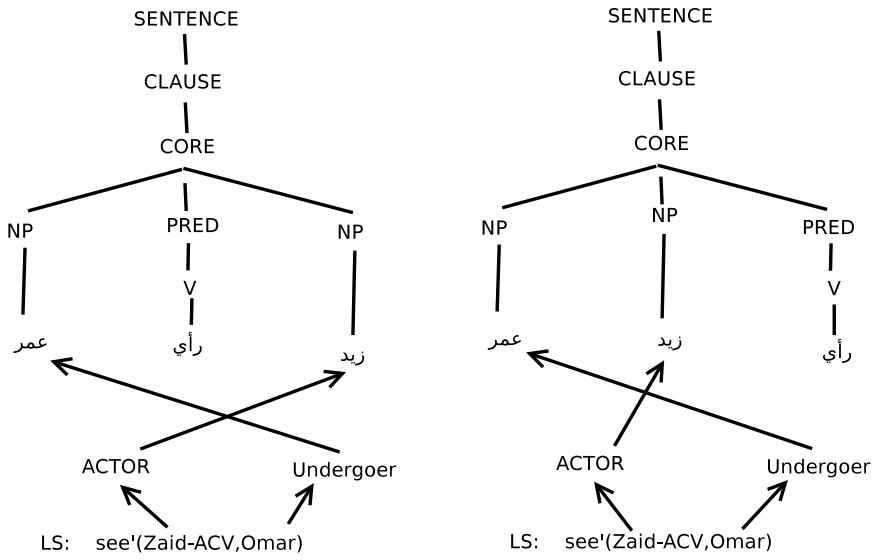

Figure 3.2: Arabic sentence types; verb subject object or subject verb object (for gloss please see example 3.1)

The first (leftmost) argument of '*see*' in the logical structure is the actor, the second the undergoer, following the RRG Actor-Undergoer Hierarchy. Since Arabic is an accusative language and رأى *rʕaā* '*see*' is a regular verb, the actor will receive nominative case and the undergoer accusative case. On the other hand, in Arabic we can start the sentence with verb first as shown in the example in (3.1b). The only changes in the clause are the form of the verb and the form of the actor NP; the arrangement of the arguments has not changed in the logical structure.

3.2 Formal representation of layered structure of the clause

Having introduced the fundamental units of clause structure, we need to have an explicit representation of them. We will present the non-universal features of the layered structure of the clause (LSC).

3.2.1 Representing the universal aspects of the layered structure of the clause

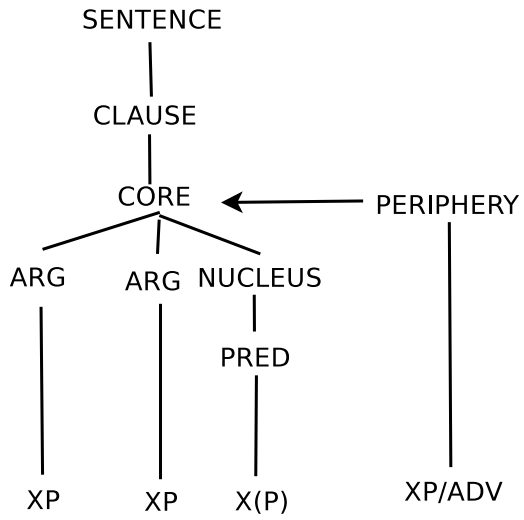

Figure 3.3: Formal representation of the layered structure of the clause

To represent the nucleus, core, periphery and clause, we will use a type of tree diagram which differs substantially from the constituent-structure trees discussed earlier. The abstract schema of the layered structure of the clause can be represented as in Figure 3.3. The clause consists of the core with its arguments, and then the nucleus, which subsumes the predicate. At the very bottom are the actual syntactic categories which realize these units. Notice that there is no VP in the tree, for it is not a concept that plays a direct role in this conception of clause structure. The periphery is represented on the margin, and the arrow there indicates that it is an adjunct; that is, it is an optional modifier of the core (Van Valin and LaPolla 1997).

Constituent structure representations of sentences in free-word-order and head-marking languages are unrevealing, because they fail to capture what is common to clauses in the different language types. The layered approach to clause structure does not suffer form the same shortcomings. For a language like Arabic, the line linking the head nouns with their determiners will be discussed in the section on noun phrase structure 3.3 below.

3.2.2 Layered structure of the clause (LSC)

In the simplex English sentences, *James ate the sandwich in the class*, *James ate the sandwich* is the core (with *ate* the nucleus and *James* and *the sandwich* the core arguments); and *in the class* is in the periphery. The first division in the clause is between a core and a periphery, and within the core a distinction is made between the nucleus (containing the predicating element, normally a verb) and its core arguments (NPs and PPs which are arguments of the predicate in the nucleus). Core arguments are those arguments which are part of the semantic representation of the verb (Van Valin and LaPolla 1997). The relationships between the semantic and syntacts units are summarized in Table 3.1

Table 3.1: Relationships between the semantic and syntactic units

Semantic element (s)	**Syntactic unit**
Predicate	Nucleus
Argument in semantic representation of predicate	Core argument
Non arguments	Periphery
Predicate + arguments	Core
Predicate + arguments + Non- arguments	Clause (= core + periphery)

3.2.3 Non-universal aspects of the layered structure of the clause

An initial phrase cannot be in the precore slot, because there is a WH–word (for example, for English who, where, what etc.) in the precore slot in the sentence; hence the position of the initial phrase is distinct from the precore slot. This position, which will be termed

the left-detached position, is outside of the clause but within the sentence. An example from English with all of these elements is given in Figure 3.4.

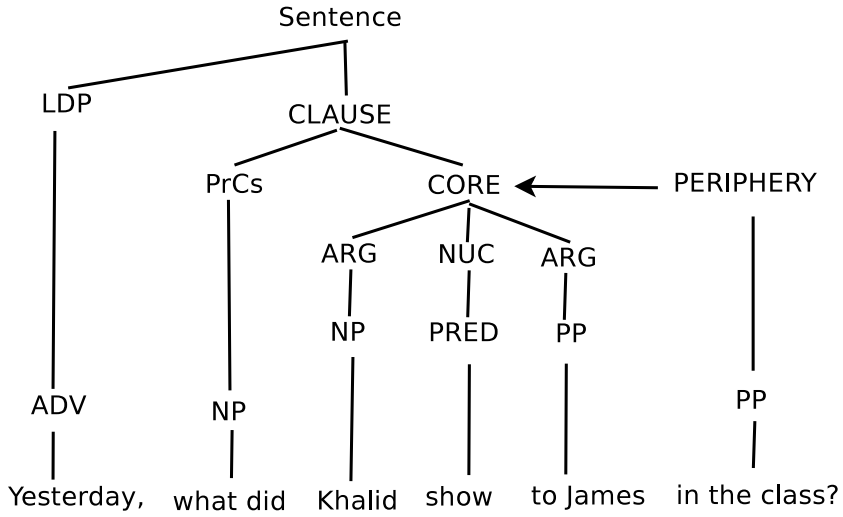

Figure 3.4: English Sentence with precore slot and left-detached position

The operator projection in Figure 3.5 may be combined with what we will call the 'constituent projection' in Figure3.8 to yield a more complete picture of the clause, as in Figure 3.6; the periphery is omitted, since it can occur in a number of different positions. What we have here is two projections of the clause, one of which contains the predicate and its arguments (the constituent projection), while the other contains the operators (the operator projection) (Van Valin and LaPolla 1997).

They are both linked through the predicate, which may be a verb, NP, AdjP or PP, because it is the one crucial element common to both. The operator projection mirrors the constituent projection in terms of layering; hence 'nucleus' in the operator projection corresponds to 'nucleus' in the constituent projection, and so on. The multiple nucleus, core and clause nodes represent each of the individual operators at that level; the number of multiple nodes corresponds to the number of operators at that level present in the sentence. If there are no operators at a given level, a bare node will be given. As the 'bare skeleton' of the layered structure of the clause on the right makes clear, the two

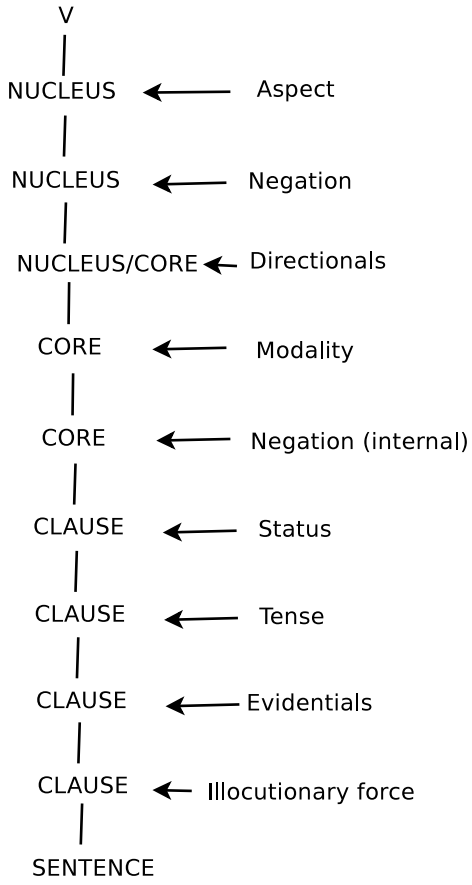

Figure 3.5: Operator projection in LSC

projections are indeed mirror images of each other, and this will become particularly important in representing the structure of complex sentences. A more complete picture of the clause in Arabic, is given in Figure 3.7. Please note that the sentences in Figure 3.7 should be read from right to left.

One of the major motivations for this scheme is that operators virtually always occur in the same linear sequence with respect to the predicating element. When an ordering relationship can be established among operators, they are always ordered in the same way cross-linguistically, such that their linear order reflects their scope. This is a very significant point. Operators are ordered with respect to each other in terms of the scope principle discussed earlier, with the verb or other predicating element in the nucleus

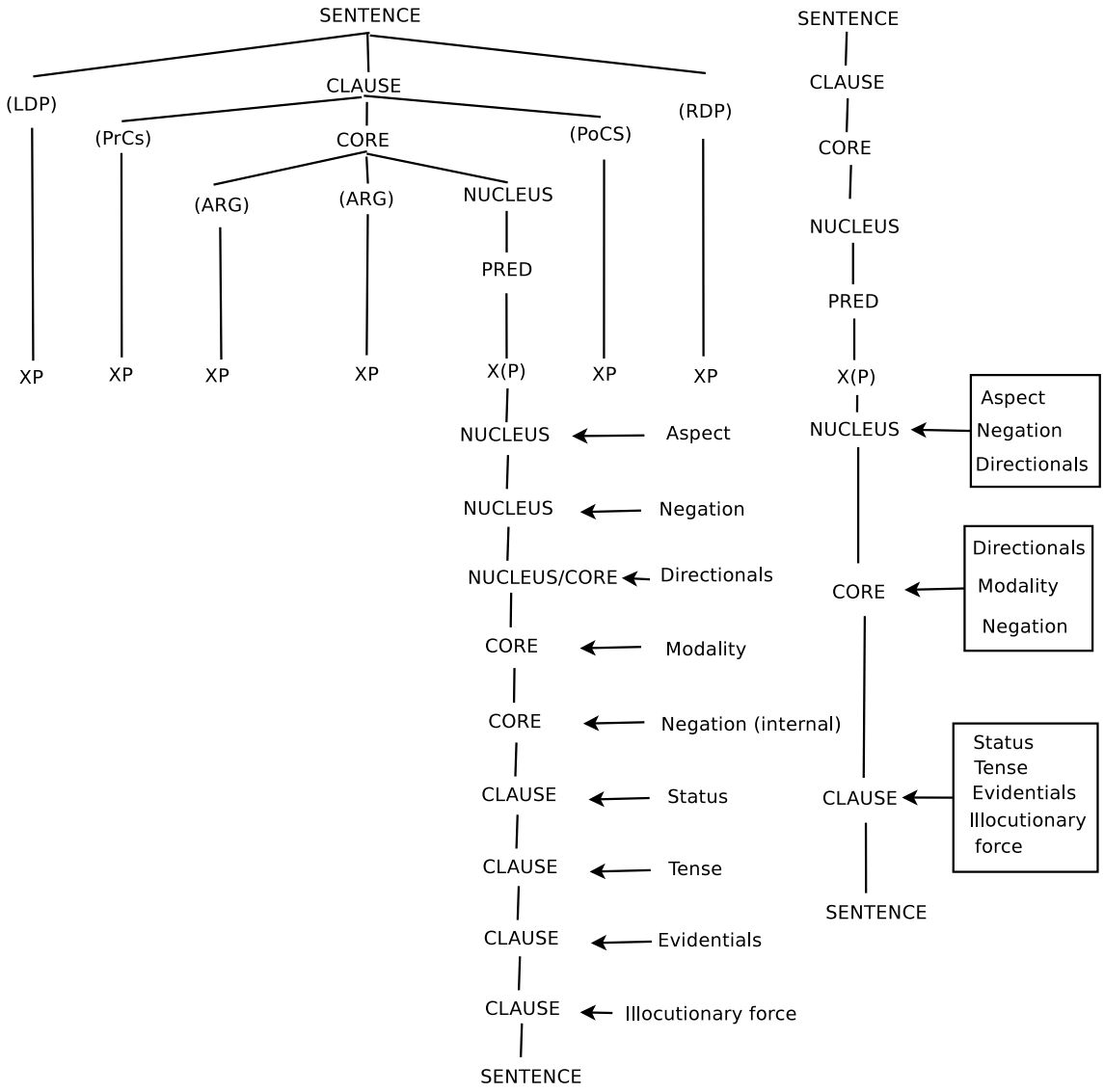

Figure 3.6: LSC with constituent and operator projections

as the anchorpoint, and thus the ordering restrictions on the morphemes expressing the operators are universal. For a technical discussion of the meaning of the various operators in the LSC (Van Valin and LaPolla 1997).

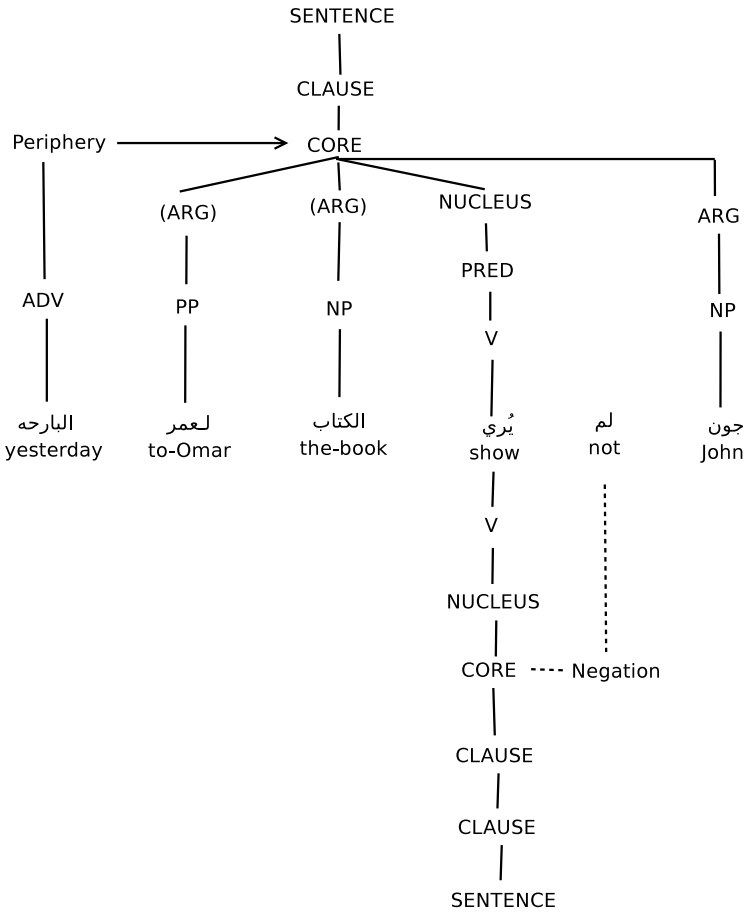

Figure 3.7: Arabic LSC

3.3 Noun phrase structure

Noun phrases refer, while clauses predicate, and yet there are striking parallels between the structure of the two which have long been noted. For example, both can be said to have arguments; while this is obvious in the case of verbs in clauses, it is also clear that relational nouns like father, friend and sister can take what could be analyzed as arguments, e.g. *father of James* / *James's father*, *a friend of Khalid* / *Khalid's friend* and the other sister of Sarah / *Sarah's other sister*. Clauses sometimes have clauses within them as arguments, as in *Zaid believed that pollution isn't a problem*, and the same is true of NPs, e.g. *Zaid's belief that pollution isn't a problem*. Given these parallels, it would be

appropriate to say that at least some nouns take arguments analogous to verbs taking arguments, and therefore it is also appropriate to posit a layered structure for NPs (LSNP) similar but not identical to that for clauses. Relating to the fundamental functional difference between verbs and nouns, is that the nominal nucleus NUC_N dominates a REF (for 'reference') node, indicating that the unit in question refers, in contrast to the PRED (for 'predicate') node which appears in the nucleus of a clause. The word 'of' is non-predicative in this construction, because it does not license the argument; moreover, it is semantically empty, as it can occur with argument NPs having many different semantic functions (Van Valin and LaPolla 1997). Consider the range of semantic functions which the of-NPs have in the following examples.

(2.2)

a.	the attack of the killer bees	Agent
b.	the gift of a new car	Theme
c.	the destruction of the city	Patient
d.	the leg of the table	Possessor
e.	the resupplying of the troops (with ammunition)	Recipient

(Nunes 1993) shows that NPs have only a single direct core argument, and it is marked by of. This is consistent with the point made above that *of* does not mark any particular semantic relation, in much the same way that the direct grammatical functions, subject and direct object, are not restricted to particular semantic functions. Accordingly, the of-marked NP counts as the single direct syntactic argument of the nominal nucleus in the core of the NP. Predicative adpositions, by contrast, have well-defined semantic content, like other predicates.

An important feature of the layered structure of the clause is the differential treatment given to operators like tense, aspect and illocutionary force, and the same contrast is a vital part of the layered structure of the noun phrase. NP operators include determiners (articles, demonstratives, deictics), quantifiers, negation and adjectival and nominal modifiers (Van Valin and LaPolla 1997).

3.3.1 NP headed

Pronouns can be classified into a number of subtypes: personal pronouns, including possessive pronouns (PRO), e.g. I liked her book; relative pronouns (PRO REL), e.g. the book which I bought; demonstrative pronouns (PRO DEM), e.g. *That pleased Mary*; WH-pronouns (PRO wh), e.g. *who did Fred see?*; and expletive pronouns (PRO EXP), e.g. *it rained.*

3.4 Lexical representations for verbs

These distinctions among the four basic Aktionsart types may be represented formally as in Table 3.2. The term Aktionsart refers to the means of a capturing the distinctions between basic states of affairs, or events, of individual verbs. These representations are called logical structures. Following the conventions of formal semantics, constants (which are normally predicates) are presented in boldface followed by a prime, whereas variable elements are presented in normal typeface. The elements in boldface and prime are part of the vocabulary of the semantic metalanguage used in the decomposition; they are not words from any particular human language.

Table 3.2: Lexical representations for the basic Aktionsart classes

Verb class	**Logical structure**
State	**predicate'** (x) or (x, y)
Activity	**do'** (x, [**predicate'**(x) or (x, y)])
Achievement	INGR **predicate'** (x) or (x, y)
Accomplishment	BECOME **predicate'** (x) or (x, y)

Hence the same representations are used for all languages (where appropriate), e.g. the logical structure for Arabic and English 'die' (intransitive) would be `BECOME dead'(x)`. The elements in all capitals, INGR and BECOME, are modifiers of the predicate in the logical structure; their function will be explained below. The variables are filled by lexical items from the language being analysed; for example, the English sentence *The dog died* would have the logical structure `BECOME dead' (dog)`, while the corresponding Arabic sentence الكلب مات *ālklb māt* . "The dog died" would have the logical structure `BECOME dead'` (الكلب) *(ālklb)* should be this sentence start with the verb مات الكلب *māt ālklb*. States are represented as simple predicates, e.g. `broken' (x)`, `be-at'(x,y)`, and `see'(x,y)`. There is no special formal indicator that a predicate is stative.

The logical structure, `be'(x,[pred'] )` is for identificational constructions, e.g. *Omar is a student*, and attributive constructions, such as *The watch is broken* require a different logical structure. In this logical structure the second argument is the attribute or identificational NP, e.g. `be' (Ayesha, [tall']), be'(Omar,[student'])`. The primary criteria for distinguishing between attributive constructions and result state constructions is whether the attribute is inherent, e.g. *Coal is black* `(be' (coal, [black']))`, or whether it is the result of some kind of process, e.g. *The fire blackened the wood* `(... BECOME black'(wood))` (Van Valin and LaPolla 1997).

3.4.1 Agents, effectors, instruments and forces

In 'Zaid is cutting the bread with a knife', an EFFECTOR, typically human, manipulates a knife and brings it into contact with the bread, whereupon the interaction of the knife with the bread brings about the result that the bread becomes cut. This may be represented as in (3.3). (The main CLAUSE in the logical structure is italicized.)

(3.3)

[**do**'(Zaid, [**use**'(Zaid, knife)])] CAUSE

[[**do**'(knife, [**cut**'(knife, bread)])]CAUSE

[BECOME **cut**'(bread)]]

The causing event in (3.3) is complex, and the INSTRUMENT argument appears three times in the logical structure: as the IMPLEMENT of use' and as the EFFECTOR of **do**'(x,[**cut**'(x,y)]). It is possible, if the first argument of the highest **do**' were left unspecified, to say The knife cut the bread, with the INSTRUMENT knife as actor.

3.4.2 change of state verb

A change of state verb may be punctual in one language and non-punctual in another. A good example of this cross-linguistic variation is English 'die' and Arabic. Both have the result that the subject is dead. Accordingly, it is possible to say in English
He died quickly , He died slowly and He died suddenly. In Arabic we can say as (3.4), also, it is possible to say in Arabic Hence the logical structure for English and Arabic 'die' would be `[ BECOME dead'(x)]`, an accomplishment.

(3.4)

(a) مات سريعا *māt sryʿā*

He died quickly.

(b) مات ببطئ *māt bbṭy*

He died slowly.

(c) مات فجاه *māt fǧāh*

He died suddenly.

3.5 Why we use RRG as the linguistic model

A reader might ask the question, why use Role and Reference Grammar as the basis of our machine translator? More than one reason prompts us to choose RRG. The most important one is that RRG is a new linguistic method and there is no research using the Role and Reference Grammar linguistic model as a basis for machine translation until now. We would like to discover this area using the RRG rules and techniques.

What distinguishes the RRG conception is the conviction that grammatical structure can only be understood with reference to its semantic and communicative functions. Syntax is not autonomous. In terms of the abstract paradigmatic and syntagmatic relations that define a structural system, RRG is concerned not only with relations of co-occurrence and combination in strictly formal terms but also with semantic and pragmatic co-occurrence and combinatory relations. According to Van Valin and LaPolla (1997) RRG takes language to be a system of communicative social action, and accordingly, analysing the communicative functions of grammatical structures plays a vital role in grammatical description and theory from this perspective language is a system, and grammar is a system in the traditional structuralist sense.

We claim that RRG is very suitable for machine translation of Arabic via an Interlingua bridge implementation model. RRG is a mono strata-theory, positing only one level of syntactic representation, the actual form of the sentence and its linking algorithm can work in both directions from syntactic representation to semantic representation, or vice versa. In RRG, semantic decomposition of predicates and their semantic argument structures are represented as logical structures. The lexicon in RRG takes the position that lexical entries for verbs should contain unique information only, with as much information as possible derived from general lexical rules. The main features of RRG are the use of lexical decomposition, based upon predicate semantics, an analysis of clause

structure and the use of a set of thematic roles organized into a hierarchy in which the highest-ranking roles are ‘Actor’ (for the most active participant) and ‘Undergoer’.

3.5.1 RRG representing the universal aspects of the layered structure of the clause

A sentence in English is NP VP, but this is not valid in Arabic sentences. There is no copula *(verb to be)* in the Arabic language, this means some types of sentence in Arabic may not contain any verb *(nominal sentence)*. For example خالد طالب *ẖāld ṭālb Khalid (is) a student*; there is no ‘is’ in this sentence in Arabic. In RRG there is no VP in sentence structure. The abstract schema of the RRG layered structure of the clause can be represented as in figure3.8.

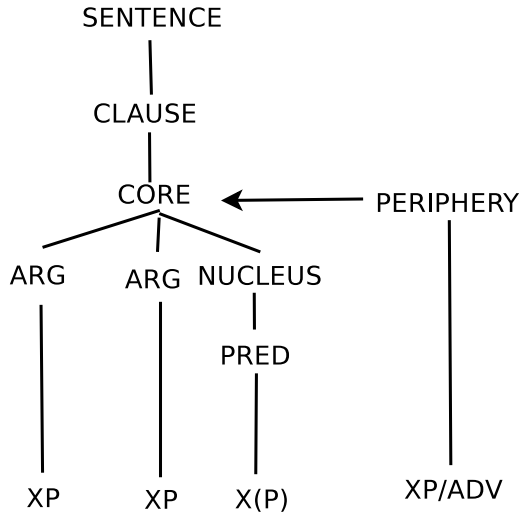

Figure 3.8: The RRG representing the universal aspects of the layered structure of the clause (Van Valin and LaPolla 1997)

The clause consists of the core with its arguments, and then the nucleus, which subsumes the predicate. At the very bottom are the actual syntactic categories which realize these units. Notice that there is no VP in the tree, for it is not a concept that plays a direct role in this conception of clause structure in RRG.

3.5.2 The lexical representation of verbs and their arguments

The approach to the depiction of the lexical meaning of verbs which we will adopt is lexical decomposition, which involves paraphrasing verbs in terms of primitive elements in a well-defined semantic metalanguage. As a simple example of the mechanism of lexical decomposition, 'kill' can be paraphrased into something like 'cause to die', and then 'die' can be broken down into `'become dead'`, Thus the lexical representation of 'kill' would be something like `'x causes [y become dead]'` (Van Valin and LaPolla 1997). A system of lexical representation should include a way of expressing the fact that the subject of 'die' and the object of 'kill' are the same argument semantically. There are many verbs pairs like this, and in many cases the relationship between them is overt. Examples include 'sink', as in 'the boat sank' and 'the torpedo sank the boat', where boat is the subject of intransitive 'sink' and the object of transitive 'sink' (Van Valin and LaPolla 1997). Another example is the predicate 'cool', which can take three forms, one adjectival and two verbal: 'The soup is cool', 'the soup is cooling' and 'the wind cooled the soup'. Thus, there seems to be a pattern of intransitive verbs whose subjects are identical to the objects of their transitive counterparts. There are cases, however, when the intransitive-transitive alternates do not have the same lexical form, as in 'die' and 'kill', or 'receive' and 'give'. An adequate theory of lexical representation should be able to capture these relationships, and lexical decomposition provides a promising method for doing it. There are many theories of lexical decomposition, which differ in terms of how fine-grained they are. It is necessary to find the right level of detail, one which allows the expression of certain important generalizations but which also has representations whose differences have morphosyntactic consequences. Thus, arriving at a decompositional system is a compromise between the demands of semantics (make all necessary distinctions relevant to meaning) and those of syntax (make syntactically

relevant distinctions that permit the expression of significant generalizations) (Van Valin and LaPolla 1997).

3.6 Summary

RRG describes mainly a sentence of a specific language in terms of:

1) logical structure;

2) grammatical procedures.

We use RRG to model Arabic, because there are certain cases where the standard NP VP categorisation does not apply due to the absence of a copula verb in the language. Since RRG does not structure sentences based around a VP, it is more suited to representing such sentences.

The main features of RRG are the use of lexical decomposition, based upon predicate semantics. The RRG model creates a relationship between syntax and semantics and can account for how semantic representations are mapped into syntactic representations. RRG also accounts for the very different process of mapping syntactic representations to semantic representations.

The division in the clause is between a core and a periphery The clause consists of the core with its arguments, and then the nucleus, which subsumes the predicate. The core arguments are those which are part of the semantic representation of the verb. The periphery is represented on the margin, and the arrow there indicates that it is an adjunct; that is, it is an optional modifier of the core.

There are languages in which operators occur on both sides of the nucleus; for example, in Arabic, the imperfect tense الفعل المضارع *ālfʕl ālmḍārʿ* marker is a prefix, while the perfect tense الفعل الماضي *ālfʕl ālmāḍy* marker is a suffix (Ryding 2007). In such cases

there will be more complex language-specific linear precedence rules for operators.

4

Machine translation strategies

In this chapter, we introduce background information about Machine Translation. We discuss the computational techniques, basic strategies, linguistic aspects and the generation problem. Much of the background information is summarised from Hutchins and Somers (1992).

Natural language processing (NLP) can be thought of as a subfield of artificial intelligence. It refers to understanding and automatic generation of natural human languages. Machine translation (MT) is a part of computational linguistics and refers to computerised systems that can translate from one natural language to another. Hence, MT uses many ideas, methods and techniques from these related fields and has also built up a body of techniques which can, in turn, be applied in other areas of computer-based language processing.

Modularity has changed as MT systems have developed. In transfer systems, lexical and structural transfer were sometimes separated. In many direct translation systems, analysis, transfer and generation are often mixed together and were not clearly distinct. As the area has matured, modularity has become an important aspect of MT systems, allowing different aspects to be developed independently.

4.1 Advantages of machine translation

Some of the advantages of machine translations are as follows:

- Machine translation is quicker than human translation.
- It ensures consistency. There is no concern that a translator might take too much creative license with a translation or forget how a particular word was translated in earlier pages. MT will translate a particular word in the same way. However, the downside is that will exhibit the same errors over and over again.
- It gives a neutral approach to translation without introducing bias, which can happen with human translators.
- Machine translation is considerably cheaper. It is a one time cost; the cost of the tool and its installation.

4.2 Computational techniques in MT

Computational processing allows for the analysis and processing of large amounts of data. Before looking at the computational aspects of MT, we introduce some basic concepts. Machine translation can take advantage of one of the basic concepts in computing. Since data and programs are separate, it is possible to build a program that functions with different types of data. In the case of MT, this means that the algorithms for translation,

and the data used for doing the actual translation can be developed separately. In reality this is a little simplistic, but there are certain examples of MT systems that operate in a similar manner for different sets of data like dictionaries and grammar rules (Hutchins and Somers 1992).

4.2.1 System design

As in standard software engineering, recent trends are towards modular and incremental system design. Whereas previously, systems would be built in a monolithic structure, with some debugging access into the system, now we build systems up in stages, completely defining and testing each stage, before incorporating it into the overall system. This method has revolutionarised software engineering and enabled much more effective collaborative design, as well as the integration of other people's work in any design.

4.2.2 Interactive systems

Interactivity is a key aspect of computer systems. MT systems can take advantage of interactivity to achieve higher quality results. It is possible for an MT system to ask the user to select from a set of possible solutions. It is also possible to extend the lexicon through user input at the time of translation. The system might flag unfamiliar words, which the user can then categorise for inclusion in the lexicon. However, intereactivity and relying on user input can have disadvantages. For example, should the user be relied upon to be correct in his input? Is he fully aware of the linguistic properties of the words? Furthermore, as more user input is required, the benefits of MT over human translation become less significant.

4.2.3 Lexical databases

A key component of any rule-based MT system is its lexical resources; the information associated with individual words. The field of computational lexicography is concerned with creating and maintaining computerised dictionaries. In practice, rule-based MT systems can often have different dictionaries, some containing the core entries, and others containing specialised vocabulary. An MT lexicon is different from a standard dictionary, and so is typically concentrated on some linguistically homogeneous set of words, e.g. abstract nouns, intransitive verbs, or the terminology of a specialist field. It is a good investment to develop tools which aid lexicographers to expand the lexicon.

4.2.4 Tokens and tokenization

The term “token” refers to an abstraction for the smallest unit in a text that is considered when describing the syntax of a language. A process of tokenization can be used to split the sentence into word tokens. Although the following example is given as XML there are many ways to represent tokenized input. The sentence *He went to the school.* could be tokenised as follows:

```
<sentence>
 <word>He</word>
 <word>went</word>
 <word>to</word>
 <word>the</word>
 <word>school</word>
</sentence>
```

4.2.5 Syntactic analysis (Parsing)

Syntactic analysis, or parsing, is a major component in a rule-based MT system. It is the process by which a sentence is dissected or analysed into constituent parts, to determine grammatical structure. One of the key challenges in analysis is dealing with ambiguity. One approach is what is called depth-first parsing, in which each possible solution is pursued to its conclusion. Each time a solution is found to be wrong, the system backtracks and takes another route until it eventually finds the correct categorisation of a word. In breadth-first parsing, alternatives are evaluated in parallel, until each alternative is found to be wrong except the right one.

4.3 Basic machine translation strategies

Traditionally three different approaches to MT have been used: direct translation, interlingua translation and transfer based translation. A few new approaches have also been established. In this section we will discuss basic strategies of MT systems.

4.3.1 Multilingual versus bilingual systems

Bilingual systems translate between a single pair of languages; multilingual systems translate between more than two languages. Bilingual systems are uni-directional or bi-directional, they may be designed to translate from one language to another in one direction only, or they are able to translate from both members of a language pair. As a further modification we may differentiate between reversible bilingual systems and non-reversible systems. In a reversible bilingual system the process involved in the analysis of a language can be inverted without change for the generation of output in the same language.

4.3.2 Direct translation

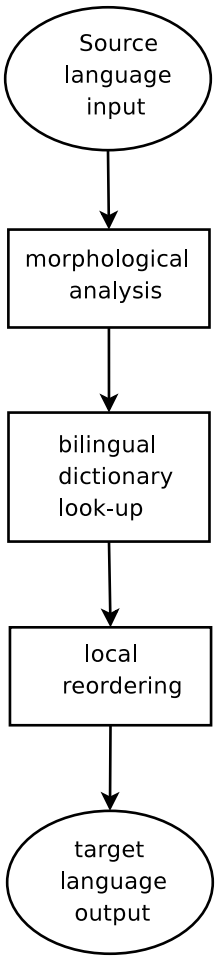

Figure 4.1: Direct MT system

Direct translation is the oldest approach to MT. The direct translation strategy passes each sentence text to be translated through a series of standard stages. If the MT system uses direct translation, this usually means that there is no syntactic analysis after the morphological analysis for the source language. The translation is based on large dictionaries and word-by-word translation with some simple grammatical adjustments e.g. on word order and morphology. A direct translation model is shown in Figure 4.1. This strategy is no longer in significant use.

4.3.3 Interlingua

The Interlingua approach is to develop a universal language-representation for text. In effect, in Interlingua there is no transfer map, and the MT model thus has phases: analysis and generation. In a standard multilingual system with X source languages and Y target languages, the transfer approach will involve XY transfer maps; moreover, we need X analysers and Y generators. In the Interlingua approach, only X parsers and Y generators are needed per language. Interlingua based MT is done via an intermediary (semantic) representation of the source language text. Interlingua is supposed to be a language independent representation from which translations can be generated to different target languages. Translation needs two phases: analysis from the source language to the Interlingua (universal language) and generation from the universal language to the target language. An Interlingua translation model with eight languages is shown in Figure 4.2.

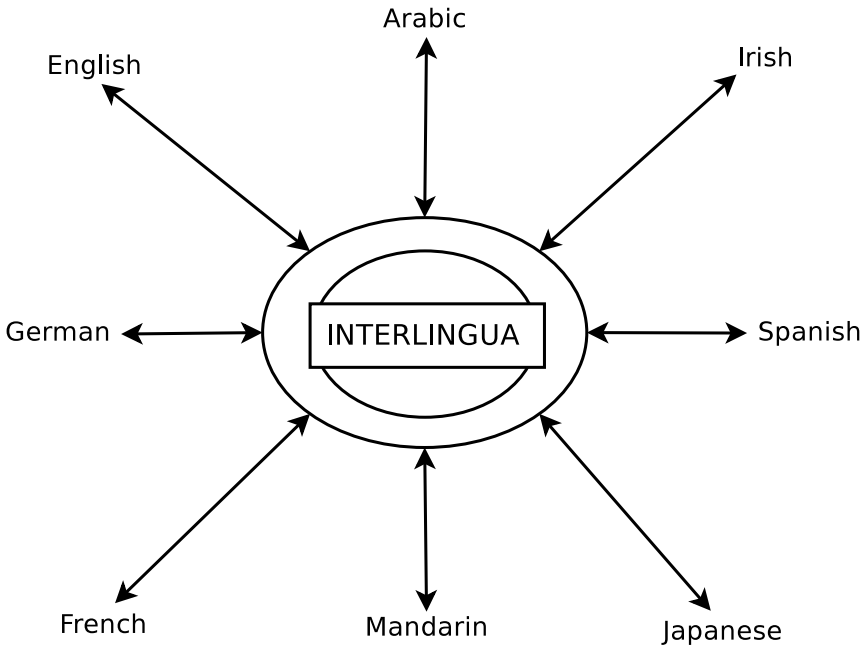

Figure 4.2: Interlingua1 model with eight languages pairs

To apply our framework to other generation languages, we only need to change the generation phases. The intermediate representation is independent of the target language, and this is the benefit of using an Interlingua approach, since analysis and generation are separate tasks which are implemented independently.

4.3.4 Transfer systems

Transfer systems are a middle course between direct and Interlingua MT strategies. Transfer systems divide translation into steps which clearly differentiate source language and target language parts. In the transfer approach there is therefore no language-independent representation: the source language intermediate representation is specific to a particular language, as is the target language intermediate representation. There is no necessary equivalence between the source and target intermediate representations for the same language. In the transfer strategy a source language sentence is first parsed into an internal representation. Thereafter a transfer is made at both lexical and structural levels into equivalent structures of the target language. In the third stage a translation is generated. Whereas the Interlingua approach requires complete resolution of all ambiguities in the source language text so that translation into any other language is possible, in the transfer approach only those ambiguities inherent in the language in question are tackled. This approach is a development over direct translation and this was lexically driven. The level of transfer differs from system to system - the representation varies from only syntactic deep structure to syntactic-semantic interpret trees. A multilingual transfer model with eight languages pairs is presented in Figure 4.3.

In comparison with the Interlingua system there are clear disadvantages in the transfer approach. The addition of a new language involves not only the two modules for analysis and generation, but also the addition of new transfer modules, the number of which

may vary according to the number of languages in the existing system: in the case of a two-language system, a third language would require four new transfer modules. The addition of a fourth language would entail the development of six new transfer modules, and so on as illustrated in Table 4.1.

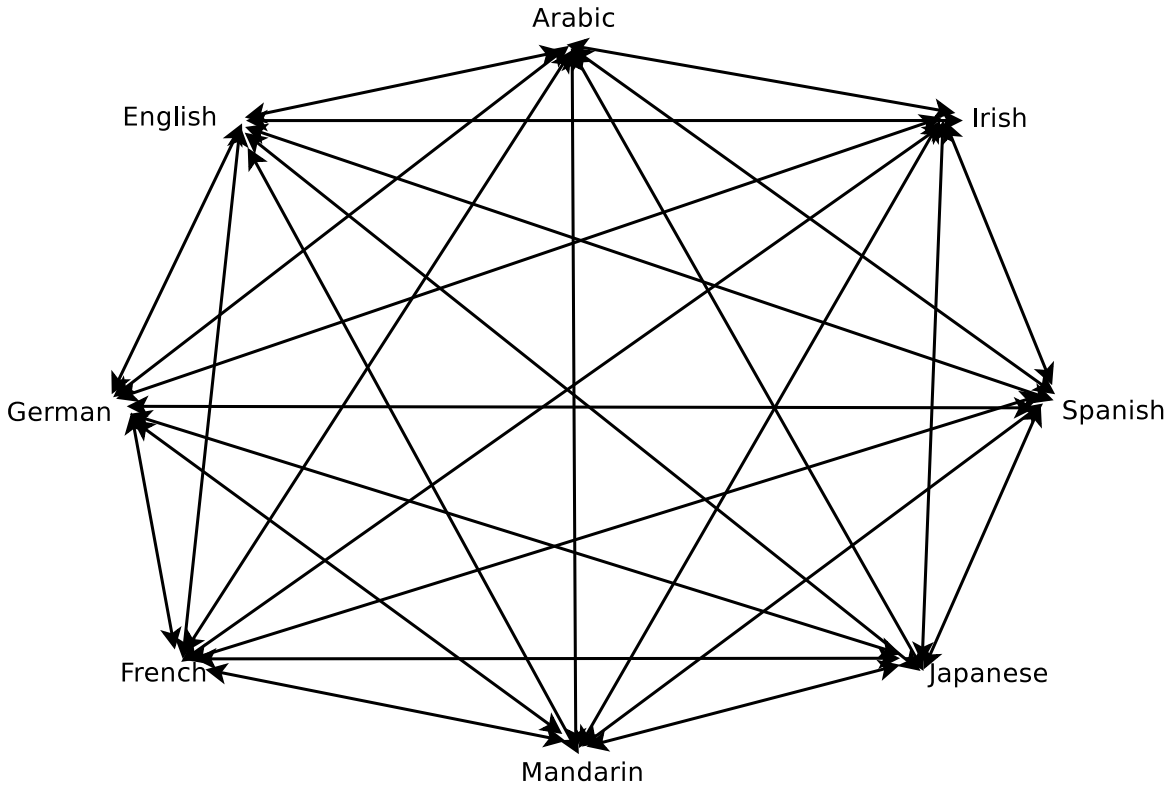

Figure 4.3: Multilinguality transfer model with eight languages pairs

Table 4.1: Modules required in an all-pairs multilingual transfer system

Number of languages	2	3	4	5	...	8	n
Analysis models	2	3	4	5	...	8	n
Generation models	2	3	4	5	...	8	n
Transfer models	2	6	12	20	...	56	$n^2 - n$
Total models	6	12	20	30	...	72	$n^2 + n$

The number of transfer modules in a multilingual transfer system, for all combinations of n languages, is $n^2 - n$. Also needed are n analysis and n generation modules, which

would also be needed for an interlingua system.

As shown in Figure 4.4, the direct method has no modules for source language analysis or target language generation. In the interlingua method the source language is fully analyzed into a language-independent representation from which the target language is generated. The transfer strategy can be viewed as falling between interlingua systems and direct systems.

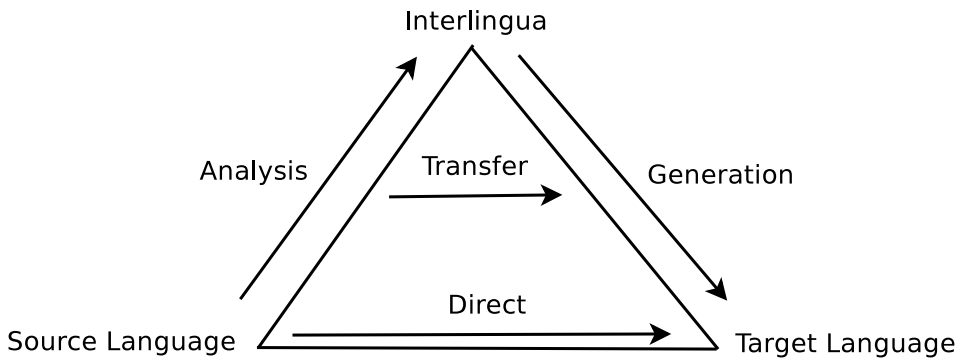

Figure 4.4: Difference between direct, transfer, and interlingua MT models, (Trujillo 1999)

Figure 4.4 shows language analysis up the left-hand side, and target-language generation down the right. The peak of the pyramid represents the theoretical interlingua representation achieved by analysis and suitable for direct use by generation. However, the path to that interlingua is long. By cutting off the monolingual analysis at some point and entering into a bilingual transfer phase, one can avoid the difficulties of a full analysis. The diagram is also intended to suggest that the more the text is analysed, the simpler the transfer will be, as depicted by the length of the line cutting across the pyramid. At the very bottom, where there is smallest amount of analysis, and nearly all the work is done in transfer, as was the case with the early direct method systems.

4.3.5 Statistical machine translation

The ideas behind statistical machine translation come out of information theory. Essentially, the document is translated on the probability $p(e|a)$ that a string e in the target language (for example English) is the translation of a string a in the source language (for example Arabic). As translation systems are not able to store all native strings and their translations, a document is typically translated sentence by sentence, but even this is not enough. We assign to every pair of strings $(e|a)$ a number $P(e|a)$, which we interpret as the probability that a translator, when presented with e, will produce a as its translation. You could imagine another program that takes a sentence a as input, and outputs every conceivable string e along with its $P(e|a)$. This program would take a long time to run, even if you limit English translations to some arbitrary length. They seek the English sentence e that maximizes $P(e|a)$ and minimizes time (Brown et al. 1993). To summarize, we compute $P(e|a)$ by summing the probabilities of all alignments. For each alignment, we make two significant simplifying assumptions: Each English word is generated by exactly one Arabic word; and the generation of each English word is independent of the generation of all other English words in the sentence. This is clearly not true in theory.

4.4 Linguistic aspects of MT

In this section we will look more closely at the kinds of linguistic problems that MT has to face and will discuss ways in which MT programs work around these problems. We will distinguish monolingual problems of morphology, lexical ambiguity, syntactic ambiguity, pragmatic aspects from bilingual problems of language contrast: lexical mismatches, structural divergence, typological differences.

4.4.1 Non-Roman alphabet scripts

Since computer technology developed mostly in English, other languages, particularly those with non-Roman alphabet have historically been seen as a special case and required new code sets to define character representations. Furthermore, not all languages with alphabetic scripts are written left-to-right, e.g. Arabic and Hebrew, so any input/output devices making this assumption will be useless for such languages. Before Unicode was standardised, there were different encoding systems for assigning this problem. Unicode provides a unique code for every character, no matter what the platform, the program and the language are. Appendix A provides the corresponding Unicode for each Arabic letter and describes the letters with their corresponding written shapes.

4.4.2 Lexical ambiguity

Category ambiguities or homographs are examples of lexical ambiguities which arise when there are potentially two or more ways in which a word can be analysed. More complex are lexical ambiguities, where one word can be interpreted in more than one way. Lexical ambiguities are of three basic types: category ambiguities, homographs and transfer (or translational) ambiguities.

4.4.2.1 Category ambiguity

The simplest type of lexical ambiguity is that of category ambiguity: a given word could be assigned to more than one grammatical or syntactic category (e.g. noun, verb or adjective) according to the context. There are several examples of this in English: *light* can be a noun, verb or adjective, also, *control* can be a noun or verb. In Arabic there are some words that can be in more than one category, for example على *ʕalā* could be a preposition with meaning of “on”, or a verb with meaning of “raise”.

4.4.2.2 Homograph

The second type of lexical ambiguity occurs when a word can have two or more different meanings. Linguists distinguish between homographs, homophones and polysemes. Homographs are two (or more) 'words' with quite different meanings which have the same spelling: example, *light* (not dark or not heavy). Many Arabic words can have two or more overlapping meanings examples; إعلان *iʿlān* could be announcement, advertisement, declaration or sign. Also, مركز *mrkz* could be centre, position, rank or status. Moreover, موقع *mwqʿ* could be position, rank, site or status. The direct approach has particular problems with homographs; the usual method of resolving homograph ambiguities is to look at the closest words for clues.

4.4.3 Syntactic ambiguity

Syntactic ambiguity arises when there is more than one way of analysing the underlying structure of a sentence according to the grammar used in the system. Example, *I know a man with a dog who has fleas*, is ambiguous. It could be the man or the dog who has fleas. It is the syntax not the meaning of the words which is unclear. The classical example is *He saw the girl with the telescope*. For the purposes of this discussion, we represent these examples in the notation of a context-tree grammar rather than in RRG notation.

The two trees in Figure 4.5 and Figure 4.6 represent the two different analyses in the sense of recording two different 'parse histories'. In linguistic terms, they correspond to the two readings of the sentence: one in which the PP is part of the NP (i.e. the girl has the telescope), and the other where the PP is the same level as the subject (i.e. the man has the telescope). For convenience, a bracketed notation for trees is sometimes used: the equivalents for the trees in Figure 4.5 and Figure 4.6 are shown in (4.1a) and (4.1b) respectively.

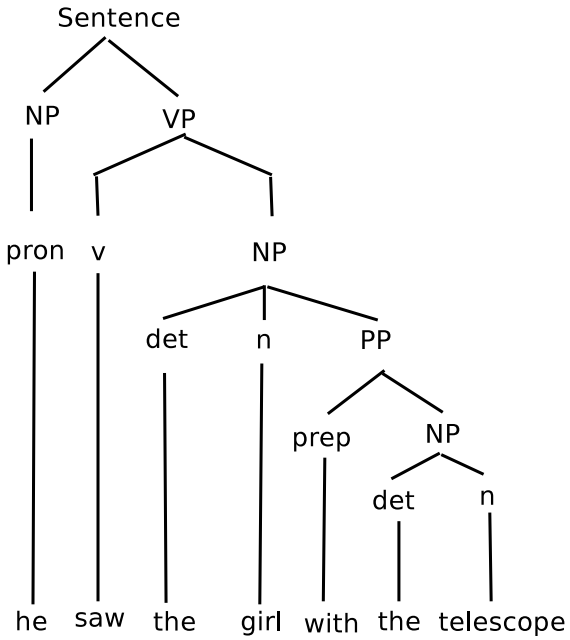

Figure 4.5: NP rule (NP –> det n pp)

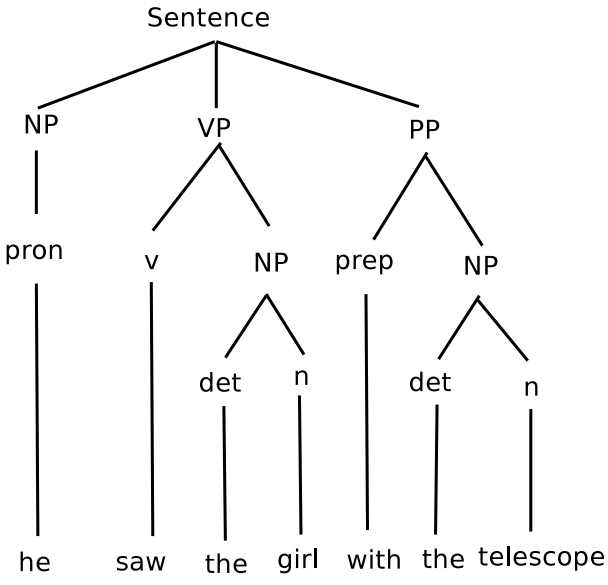

Figure 4.6: PP is attached at a higher level

(4.1a)

```
S(NP(pron(he)), VP( v(saw),NP(det(the), N(girl),
PP(prep(with),NP(det(the),n(telescope))))))
```

(4.1b)

```
S(NP(pron(he)), VP(v(saw),NP(det(the),n(girl)),
PP(prep(with),NP(det(the),n(telescope)))))
```

The tree structures required may of course be much more complex, not only in the sense of having more levels, or more branches at any given level, but also in that the labelling of the nodes (i.e. the ends of the branches) may be more informative.

4.4.4 Structural differences

Many relatively trivial syntactic differences between languages are well known, e.g. in Arabic most adjectives follow nouns but in English adjectives normally precede the nouns they qualify. Also, Arabic sentences have more than one structural type. The sentence which contains a verb, will have order of the form verb(V), subject(S) and object(O) or verb(V), object(O) and subject(S). The only combinations that do not occur in Arabic are OSV and SOV (Attia 2004).

4.5 Challenges of Arabic to English MT

Arabic words can often be ambiguous due to the three-letter root system. These consonant roots interlock with patterns of vowels or consonants to words or word stems. This root system allows the language to evolve to cover a wide range of meanings. In some derivations one or more of the root letters is dropped, resulting in possible ambiguity. Examples of derived words from a three-letter-root are shown in Table 4.2.

Table 4.2: Derived words from a three-letter-root in Arabic

Arabic	Example	POS
كَتَبَ *kataba*	he wrote	verb
كاتَبَ *kātaba*	he corresponded	verb
كُتِبَ *kutiba*	it was written	verb
كِتاب *ktiāb*	book	noun
كُتُب *kutub*	books	noun
كاتِب *kātib*	writer; (adj) writing	noun
كُتّاب *kutāb*	writers	noun
مَكْتَب *maktab*	desk; office	noun
مَكاتِب *makātib*	desks; offices	noun
مَكْتَبَة *maktabah*	library	noun

A root is defined in (Ryding 2007) as "a relatively invariable discontinuous bound morpheme, represented by two to five phonemes, typically three consonants in a certain order, which interlocks with a pattern to form a stem and which has lexical meaning."

There are also two and four letter roots. They are discontinuous because the root letters can be interspersed with other letters in a pattern. However, the order of the root letters must be the same.

A pattern is defined in (Ryding 2007) as "a bound and in many cases discontinuous morpheme consisting of one or more vowels and slots for root phonemes (radicals), which either alone or in combination with one to three derivational affixes, interlocks with a root to form a stem, and which generally has grammatical meaning."

Patterns signify grammatical or language-internal information, distinguishing word types and classes. These patterns can differentiate between nouns, verbs and adjectives, but also give more detailed information about sublasses of these categories. There are fewer patterns than roots.

Arabic has a large set of morphological features (Al-Sughaiyer and Al-Kharashi 2004).

These features are in the form of prefixes, suffixes and also infixes that can completely change the meaning of the word. Also, in Arabic there are some words that hold the meaning of a full sentence for example, سنسافر *snsāfr* , would translate to; *We will travel.* in English. This means any MT system should apply thorough analysis in order to obtain the root or to deduce that in one word there is in fact a full sentence. Arabic has a relatively free word order, this poses a significant challenge to MT due to the vast possibilities to express the same sentence in Arabic.

4.6 Generation

In this section we discuss the generation of target language texts.

4.6.1 Generation in direct systems

In direct systems in Figure 4.7, generation is based as much as possible on source language structures: nothing is changed more than strictly needed for the creation of a suitable target language word order.

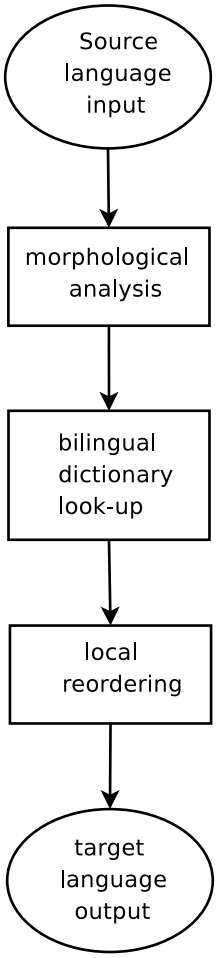

Figure 4.7: Direct MT system

4.6.2 Generation in transfer-based systems

In a transfer system, the generation phase is generally divided into two parts, syntactic generation and morphological generation. Syntactic generation involves creating a deep-tree structure from the output of the analysis, which is then re-ordered by transformational rules. The final tree is labelled with the grammatical functions and features of the target language. This re-ordered surface structure can now be processed by the morphological generator, which creates labelled lexical items which can be easily turned into target sentences.

4.6.3 Generation in interlingua systems

The steps for generating texts in interlingua-based systems are similar to those described for transfer-based systems. Generation includes phases of syntactic and morphological generation. The main difference is that the start point is not a deep-structure syntactic representation, but an interlingua representation, probably based on predicate-argument structures. The syntactic structure must first be generated from the interlingual representation by a phase often known as semantic generation. The process may be described using example in Figure 4.8. The structure to be generated is shown in Figure 4.9.

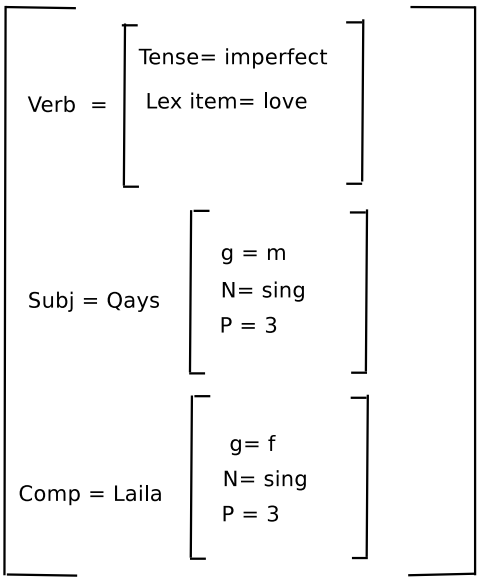

Figure 4.8: Semantic generation

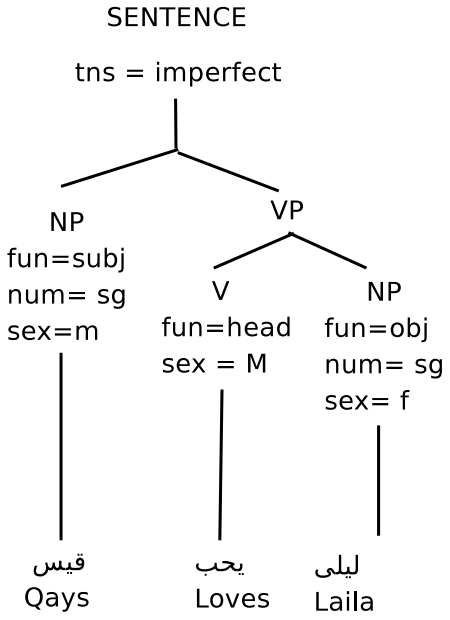

Figure 4.9: Structure to be generated

4.7 Summary

In the stages of analysis and generation, most MT systems contain separated components dealing with different levels of linguistic description: morphology, syntax, semantics. Hence, analysis may be divided into morphological analysis, syntactic analysis and semantic analysis (Hutchins and Somers 1992).

For the purposes of this study, our proposed solution to an Arabic-English translator will be based upon the interlingua model of machine translation. Arabic is unique in many ways but is not immune to the standard challenges faced by other languages such as multiple meanings of words, non-verbalisation and insufficient lexicons.

An Interlingua model that incorporates source language analysis, thereby creating a so called universal logical structure, will facilitate multiple language generation in a more flexible way. An Interlingua model is presented in Figure 4.10.

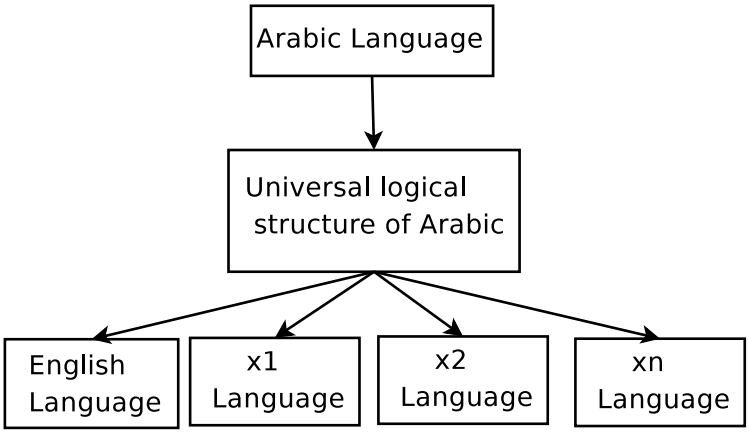

Figure 4.10: Interlingua model of Arabic MT

For the elements of *subject*(S), *verb*(V) and *object*(O), Arabic's relatively free word order allows the combinations of SVO, VSO and VOS. The only combinations that do not occur in Arabic are OSV and SOV. Arabic's flexible word order is discussed later in this research. Our research develops a rule-based and lexical framework for the processing of Arabic using the Role and Reference Grammar (RRG) linguistic model. The framework is to be evaluated using a machine translation system that translates an Arabic text as source language into an English text as target language.

5

Design of Arabic to English machine translation system based on RRG

The UniArab system is a natural language processing application based on Role and Reference Grammar (RRG) for translating the Arabic language into any other language, using an interlingua bridge. An interlingua based MT approach to translation is done via an intermediate semantic representation of the source language (Hutchins 2003). The conceptual architecture of the UniArab system is shown in Figure 5.1. To apply it to any other language, we need only change phases 9, 10, 11 and 12. Figure 5.1 will be discussed in more detail in Chapter 6.

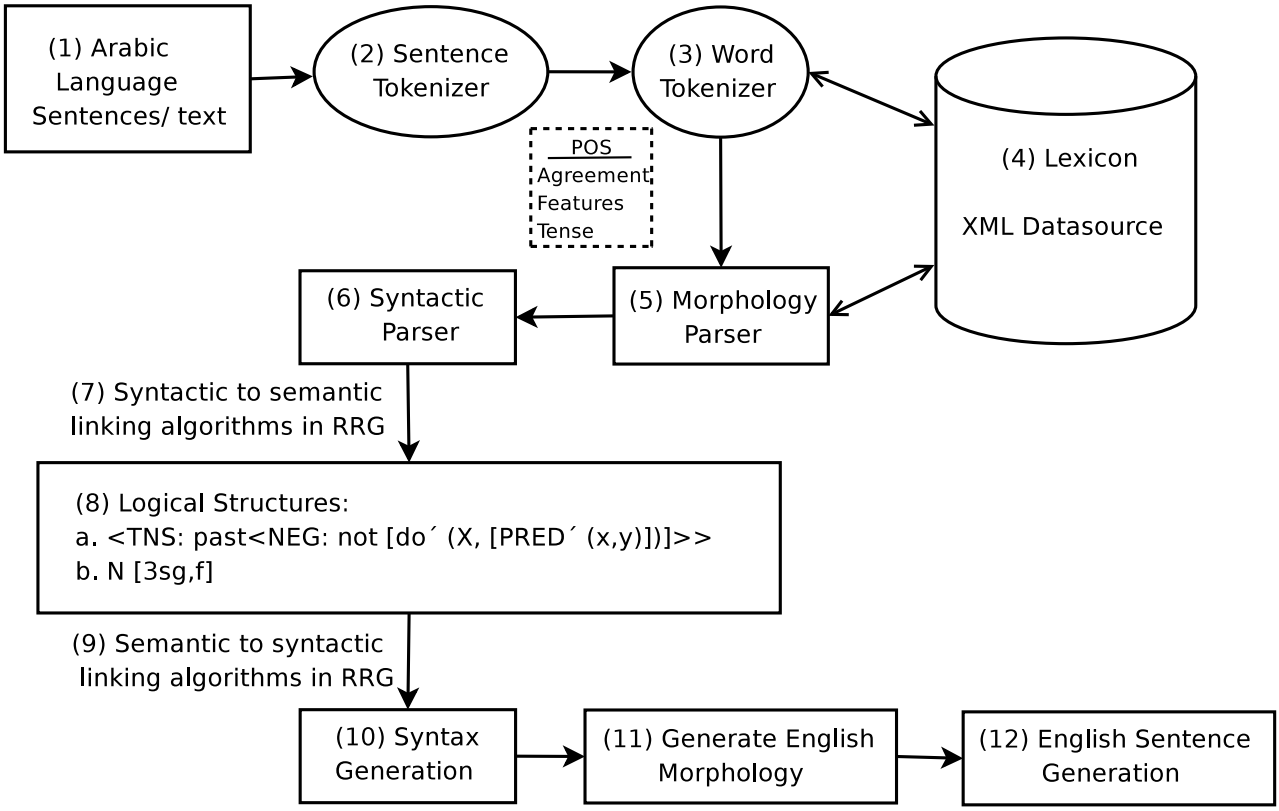

Figure 5.1: The conceptual architecture of the UniArab system

5.1 UniArab: Interlingua-based system

In interlingua MT systems, the result of source language analysis is a language independent representation of the text which is the basis for the generation of the target language text. The advantages of using interlingua for multilingual systems have already have been mentioned in Chapter 4. The challenges start with analysis and generation, they have to be strictly separated; it is not desirable to learn about analysis towards a particular target language and it is not possible, during generation, to refer to the original source language text. Using an RRG based interlingua bridge creates strong analysis methods that incorporate all attributes of a sentence and its words including the logical structure of its verbs. This technique could be very amenable to interlingua. The interlingua representation must include all the information that can possibly be required during

the generation of any target language text or rather more correctly: any target language included in the system from the outset or planned for the future. In effect, this high degree of language-independence and objectivity means that interlinguas must strive towards universality in lexicon and structure: one might almost say, towards representing the meaning of the text. Most interlingua-based systems use representations. The Chomskyan theory of deep structures was thought to be attractive, but it is now agreed they are not sufficiently abstract, being too oriented towards the surface features of individual languages. The implications of neutral structural representations can be illustrated by allowing for differences of word order between languages, and their significance. In English, word order is the primary means of distinguishing grammatical functions like subject and object. The Arabic language has a relatively free word order. The implication for an interlingua is that it is not enough to designate word order on its own: the interlingua must represent the significance in terms of grammatical function (syntactic relations), text function, determination, case role or whatever else the interpretation of the word-order dictates. Structural differences can be treated in transfer-based systems by structural transfer rules. But in interlingua-based systems the representation must be language-neutral.

5.2 Designing an XML lexicon architecture for Arabic MT based on RRG

The lexicon in RRG takes the position that lexical entries for verbs should contain unique information only, with as much information as possible derived from general lexical rules. The lexicon is designed to reflect the word categories in the Arabic language with as much information as possible derived from general lexical rules. The lexicon stores the Arabic words in categories, each category is stored in an XML format datasource

file. In order to be able to analyses Arabic by computer we must first extract the lexical properties of the Arabic words. The UniArab system uses the lexicon to construct a logical structure for Arabic input sentences, also represented in XML, which is then used for generating the target language translation. We show the structure of the UniArab lexicon, discuss how it is used in the system, and show the user interface used for adding to the lexicon. The lexicon is built from individual words at present.

5.2.1 An XML-based lexicon

In order to build this system and represent the data sources, we use the XML language and Java. The most recent recommendation of the XML language has been presented by Bray et al. (2008). XML has become the default standard for data exchange among heterogeneous data sources (Arciniegas 2000). The UniArab system allows data to be stored in XML format. This data can then be queried, exported and serialized into any format the developer wishes.

We choose to create our data source as XML, for optimum support or different platforms. It was also easier as we used Arabic letters not Unicode inside the data source, XML fully supported Arabic. We created our search engine using Java.

5.2.2 Lexical representation in UniArab

Lexical frames represent the language-dependent lexicon. We use an XML data source to represent the UniArab lexicon. The lexicon creates pointers to corresponding conceptual frames or attributes of each word. These frames also have relations which link them to verb class frames, which are organized hierarchically according to the particular language.

In Phase 3 in Figure 5.1, the UniArab system tokenizes a sentence into words, then sends each word to the search engine within the Lexicon to query the category of each word and all attributes for that word. The Lexicon returns the corresponding category and its attributes as detailed below. The Morphology Parser, Phase 5, receives the word meta-data and ensures that the properties of the words are consistent. The verb attributes in particular, are of great importance in correctly extracting sentence logical structure further down the processing chain, helping to answer the basic question 'Who does what?' In free word order sentences, for example, يحب قيس ليلى *yḥb qys lylā*, 'Qays loves Laila' multiple orders are possible including verb-subject-object, verb-object-subject or subject-verb-object. The attributes of the verb agree with the gender of the subject. Given the masculine gender of the verb in this case, the Syntactic Parser will look for a masculine proper noun to make the actor for this sentence. If there is more than one masculine proper noun in such a case, then Modern Standard Arabic defines the first proper noun as the actor. The Morphology Parser will be extended so that it can deal with words that are defined in multiple categories, deciding which should be processed. Meanwhile the Syntactic Parser, so far, has only been implemented for extracting word order, though it will be extended to deal with word ambiguities in future versions.

5.2.3 Lexical properties

Figure 5.2 shows the structure of the Lexicon including the properties stored for each word category. For all categories, an Arabic word is stored along with its English representation. Since word ambiguity has not been dealt with so far, there is a one to one mapping for the simple sentences which UniArab processes up to now. However, word ambiguity is supported in the structure, with each possible case stored as a separate record. All search results will be passed to the Morphology Parser to decide which is taken.

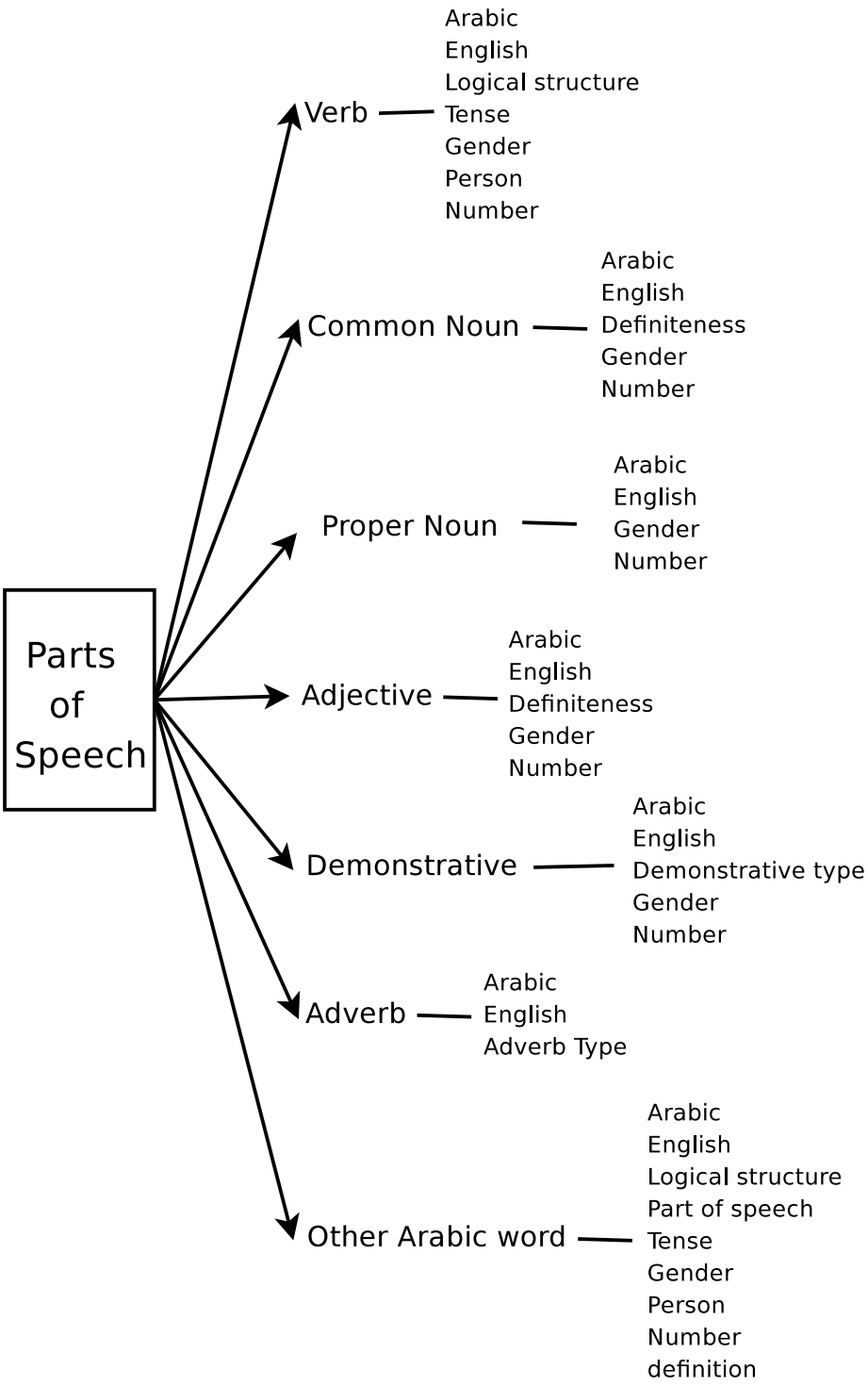

Figure 5.2: Information recorded in the UniArab lexicon

Since the verb is the key component when analysing using RRG, each verb has an associated logical structure, which is later used to determine the logical structure of the full sentence. The tense of the verb is also stored within its metadata along with the person. The verb type also stores the gender, which in Arabic must be either masculine or feminine; there is no neutral gender. The number property in arabic can be singular, dual or plural. These properties help the Syntactic Parser analyse the sentence, since there must be agreement with the subject and verb, among other rules.

Although we adhere to the Interlingua approach, we do not do so with the translation of lexical items. In an ideal Interlingua system lexical entries should be broken down into sets of semantic features. For example the word “man” is broken down into +human +male +adult. While this works in theory, in practice we cannot find enough semantic features to describe every entity in the world. For example “cow”, “computer” and “chair” cannot be described using these sets of semantic features unless we invent a unique semantic feature for every object and this is practically impossible, and of course, beyond the scope of this thises.

Table 5.1: Verb 1

Arabic verb	قرأ *qrʔa*
English translation	read
Logical structure	[do’(x,[read’(x,(y)])]
Tense	past
Gender	m
Person	3rd
Number	singular

In Tables 5.1, 5.2, we show two examples of records for verbs in the Lexicon. The absence of ت *t* ‘t’ suffix signifies m: gender. The English translation of these verbs are ‘read’ and ‘wrote’.

An example of the XML record for a verb in the Lexicon is shown here;

Table 5.2: Verb 2

Arabic verb	كتبت *ktbt*
English translation	wrote
Logical structure	[do'(x,[write'(x,(y)])]
Tense	past
Gender	f
Person	3rd
Number	singular

<قرأ

EnglishTranslate="read"

LogicalStructures= "<TNS:PAST[do'(x,[read'(x,y])]>"

NumberVerb="sg"

P.O.S="Verb"

genderVerb="M"

personVerb="3rd"

tenseVerb="PAST"

/>

5.3 Design of test strategy

We will create variants of Arabic sentences that represent all possible structures of sentences that UniArab can translate. We will evaluate the result of the system output by comparing between human-translated and machine-translated versions. In Tables 5.3 to 5.9 we represent some examples of sentences that are used to test the UniArab system. For actual test examples see Appendix C.

Verb-Subject one argument in deferent tenses:

In Table 5.3, Verb-Subject Agreement with two arguments sentences, are sentences where UniArab should select the correct form of the verb. In particular the verb must agree with the subject.

Table 5.3: Test strategy: verb-subject agreement

Arabic	human-translated	UniArab	other
يشرب عمر الحليب *yšrb ʕmr ālḥlyb*	Omar is drinking the milk.	?	?
شرب عمر الحليب *šrb ʕmr ālḥlyb*	Omar drank the milk.	?	?
مارك قرا الكتاب *mārk qrā ālktāb*	Mark read the book.	?	?
سيشرب مارك اللبن *syšrb mārk āllbn*	Mark will drink the milk	?	?

Demonstrative Adjective-Noun:

The system should place the Demonstrative Adjective-Noun Agreement that agrees in number and gender. The test sentences are shown in Table 5.4.

Table 5.4: Test strategy: demonstrative adjective-noun agreement

Arabic	human-translated	UniArab	best of rest
هذا الرجل *hdā ālrğl*	This man	?	?
ذلك الرجل *dlk ālrğl*	That man	?	?

Gender-Ambiguous proper nouns:

Proper nouns can confuse MT in two different ways. The first, the MT system may not identify that the word is a proper noun and analyse it as a noun, adjective, or any other categories. The second is that it may fail to identify the gender of the noun and thus fail to provide information needed for agreement in Arabic. The test sentences are shown in Table 5.5. The UniArab system should follow the rules for agreement in number and gender. This is due to the fact that Arabic differs greatly from English in the distribution of number and gender in the pronoun system, lexical items as well as the syntactic structure. This difference results in many agreement problems during the translation process.

Table 5.5: Test strategy: gender-ambiguous proper nouns

Arabic	human-translated	UniArab	best of rest
قرأ جاك الكتاب *qrʕa ğāk ālktāb*	Jack read the book.	?	?
قرأت ماري الكتاب *qrʕat māry ālktāb*	Mary read the book.	?	?

Copula verb 'to be':

There are certain cases where the standard NP VP categorisation does not apply due to the absence of a copula verb in the language. In Arabic there is no verb 'to be' (Salem et al. 2008b). UniArab should understand if the sentences contain verb 'to be' and generate them correctly. The test sentences are shown in Table 5.6.

Table 5.6: Test strategy: verb 'to be'

Arabic	human-translated	UniArab	best of rest
أنا المهندس *ʾanā ālmhnds*	I am the engineer	?	?
هو مهندس *hw mhnds*	He is an engineer	?	?

Verb 'to have':

UniArab should understand if the sentences contain 'to have' and generate them correctly. Arabic, like Modern Irish, has no verb of 'to have'. The test sentences are shown in Table 5.7.

Table 5.7: Test strategy: verb 'to have'

Arabic	human-translated	UniArab	best of rest
لقد قمت بالحجز *lqd qmt bālḥǧz*	I have made a reservation.	?	?
لقد فقدت تذكرتي *lqd fqdt tḏkrty*	I have lost my ticket.	?	?

The free word order in Arabic:

Arabic has free word order, this poses a significant challenge to MT due to the vast possibilities to express the same sentence in Arabic (Salem et al. 2008a). The actor in Table 5.8 could be the first, second or third argument. UniArab should analyse who the actor is.

Pro–Drop:

In technical linguistic terms, Arabic is a 'pro–drop' or 'pronoun–drop' language (Ryding 2007). The pro–drop parameter is an aspect of grammar that allows subjects to be optional but understood in some languages. That is, every inflection in a verb paradigm

Table 5.8: Test strategy: free word order (Verb Noun Noun)

Arabic	human-translated	UniArab	best of rest
يحب قيس ليلى *yḥb qys lylā*	Qays loves Laila.	?	?
قيس يحب ليلى *qys yḥb lylā*	Qays loves Laila.	?	?
يحب ليلى قيس *yḥb lylā qys*	Qays loves Laila.	?	?

is specified uniquely and does not need to use independent pronouns to differentiate the person, number, and gender of the verb. The test sentences are shown in Table 5.9.

Table 5.9: Test strategy: pro–drop

Arabic	human-translated	UniArab	best of rest
فاتتني الطائرة *fātny ālṭāyrh*	(I) missed the plane.	?	?
أريد غرفة *ʾaryd ġrfh*	(I) want a room.	?	?
نسيت محفظتي *nsyt mḥfẓty*	(I) forgot my wallet.	?	?
أريد خاتم *ʾaryd ẖātm*	(I) want a ring.	?	?

5.4 Design of evaluation criteria

We will evaluate the result of output by comparing with human-translated and machine-translated versions . Comparisons can be made between two machine translation systems, or between human-translated and machine-translated sentences. UniArab system is compared with translations done by human translators. Then this result is compared with the results of other (Arabic to English) Machine translation systems. We are comparing different levels of human translation with UniArab system output, using human subjects as judges. The human judges were skilled for the purpose of Machine Translation; it is an efficient evaluation for MT research. The evaluation study compared an MT system translating from Arabic into English with human translators. The human translators were a native Arabic speaking L1 adults who had English as their L2. The five point scale for adequacy indicates how much of the meaning expressed in the reference translation is

also expressed in a hypothetical translation:

5 = All

4 = Most

3 = Much

2 = Little

1 = None

The second five point scale indicates how fluent the translation is. When translating into English the values correspond to:

5 = Flawless English

4 = Good English

3 = Non-native English

2 = Bad English

1 = Incomprehensible

5.5 Summary

UniArab is designed as an Interlingua machine translator, which takes Arabic sentences and analyses their structure producing in interlingua representation which can then be used in isolation to generate the English translation. We presented a test strategy in which a wide range of sentence types will be used to test the effectiveness of UniArab. We then set evaluation criteria which can be used to quantify how the system performs for each of these test types.

6

UniArab: a proof-of-concept Arabic to English machine translation system

This chapter presents an Arabic to English machine translator system, called UniArab. UniArab is a proof-of-concept translation system supporting the fundamental aspects of Arabic, such as the parts of speech, agreement and tenses. UniArab stands for **Uni**versal **Arab**ic machine translator system. UniArab is based on the linking algorithm of RRG (syntax to semantics and vice versa) as indicated in Figure 6.1.

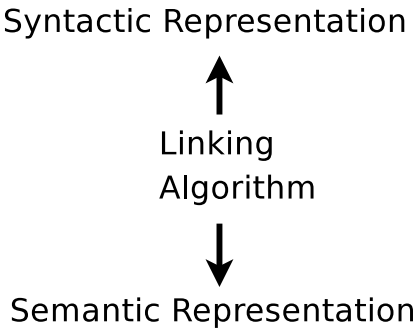

Figure 6.1: Layout of Role and Reference Grammar

6.1 Conceptual structure of the UniArab system

The conceptual structure of the UniArab system is shown in figure 6.2. The system accepts Arabic as its source language. The morphology parser and word tokenizer have a connection to the lexicon which holds all attributes of a word.

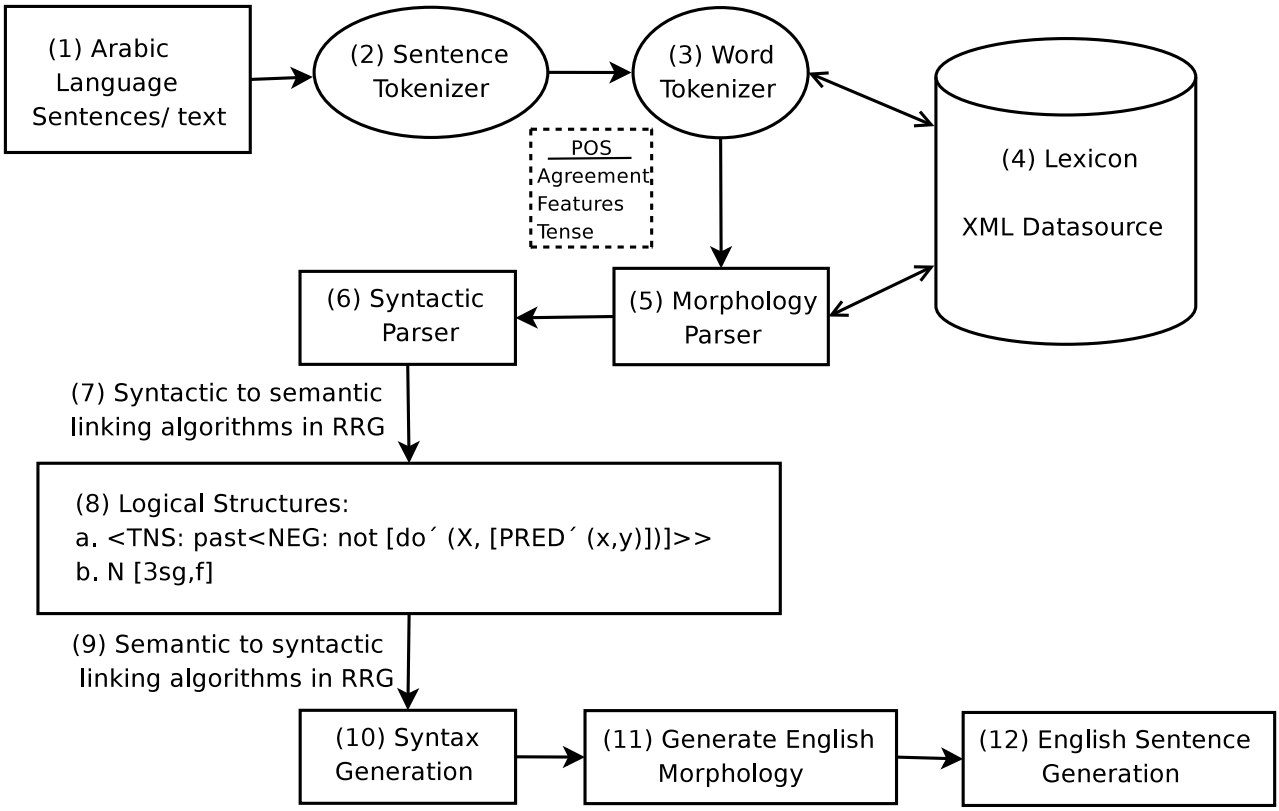

Figure 6.2: The conceptual architecture of the UniArab system

UniArab stores data in XML format. This data can then be queried, exported and serialized into any format the developer wishes. The system can understand the part of speech of a word, agreement features, number, gender and the word type. The syntactic parse unpacks the agreement features between elements of the Arabic sentence into a semantic representation (the logical structure) with the 'state of affairs' of the sentence.

In UniArab we intend to have a strong analysis system that can extract all attributes from the words in a sentence.

6.1.1 Technical architecture of the UniArab system

The structure of the UniArab system in Figure 6.2 breaks down into the following phases:

Phase (1) - Arabic language sentence. The input to the system consists of one or more sentences in Arabic.

Phase (2) - Sentence Tokenizer. Tokenization is the process of demarcating and classifying sections of a string of input characters. In this phase the system splits the text into sentence *tokens*. The resulting tokens are then passed to the word tokenizer phase. For example قرأ خالد الكتاب. خالد تلميذ ذكي. *qr‘a ẖālد ālktāb. ẖālد tlmyد dky.* will be two tokens; قرأ خالد الكتاب *qr‘a ẖālد ālktāb* and خالد تلميذ ذكي *ẖālد tlmyد dky* the translation of these two sentences is *Khalid read the book. Khalid is a clever student*.

Phase (3) Word Tokenizer There, sentences are split into tokens قرأ خالد الكتاب *qr‘a ẖālد ālktāb Khalid read the book*, the output of phase 3 is as follows;

```
<sentence>
<word>قرأ qr‘a</word>
<word>خالد ẖālد</word>
<word>الكتاب ālktāb</word>
</sentence>
```

Phase (4) Lexicon Datasource A set of XML documents for each component category of Arabic.

Phase (5) Morphology Parser Directly works with both the Lexicon and Tokenizer to produce the word order. A connection is made to the datasource of phase 4 which

has been implemented as a set of XML documents. The use of XML has the added advantage of portability. UniArab will effectively work the same regardless of the operating system. To understand the morphology of each word, we first tokenize each sentence and determine the word relationships. Phase 5 of the system holds all attributes specific to each word of the source sentence.

Phase (6) Syntactic Parser Determines the precise phrasal structure and category of the Arabic sentence. At this point, the types and attributes of all words in the sentence are known.

Phase (7) Syntactic linking (RRG) We must first develop the link from syntax to semantics out of the phrasal structure created in Phase 6, if we are to create a logical structure that will generate a target language and also act as the link in the opposite direction from semantics to syntax. The system should answer the main question in this phase, **who does what to whom?** We use the gender of the verb to determine the actor. When the subject and object have different genders, the gender of the verb must match the subject. If they both agree with the verb, then MSA dictates that the first noun is the subject. In this case the actor is *Khalid* and the undergoer is *the book*.

Phase (8) Logical Structure Creation of logical structure is the most crucial phase. An accurate representation of the logical structure of an Arabic sentence is the primary strength of UniArab. Below is a sample output from the UniArab system. The Arabic equivalent of the past tense sentence 'Khalid read the book' قرأ خالد الكتاب *qr‹a ḫālد ālktāb* is input as the source.

الكتاب *ālktāb* book:N خالد *ḫālد* Khalid:MsgN قرأ *qr‹a* read:V

The results of the parse can be seen in the following logical structure:

Verb read

<TNS:PAST[do'(x,[read'(x,(y)])]>

sg 3rd M PAST قرأ *qr‘a*

where the Proper Noun is Khalid sg unspec M خالد *ẖālد*

and the Noun is, the book sg def M الكتاب *ālktāb*

Consider the following example; Omar is a student. be'(Omar,[student']). in Arabic عمر تلميذ *‘mr tlmyد*. This is a challenge since there is no verb 'to be' in Arabic, but this must be inferred for correct translation. Instead of saying *'Omar is a student'*, the Arabic equivalent would be *'Omar student'*. We also face the challenge of inferring the indefinite article, which does not exist in Arabic. All of the unique information for each word can thus be taken from the lexicon to aid in the creation of a logical structure of the target language.

Phase (9) Semantic to Syntax Assuming we have an input and have produced a structured syntactic representation of it, the grammar can map this structure from a semantic representation. In this phase the system uses a linking algorithm provided by RRG to determine actor and undergoer assignments, assign the core arguments and assign the predicate in the nucleus. The system uses semantic arguments of logical structures other than of the main verb.

Phase (10) Syntax Generation This will be unique for each target language. In this phase the system uses the target language rules to generate the syntax. In this case English language rules are used.

Phase (11) Generate English Morphology The system generates English morphology in an innovative way, generating the tenses not existent in Arabic but in English as well as verb 'to be'.

Figure 6.3 shows the technique used to generate the correct verb tenses, and generate verb to be. Verbs in English have a mood; e.g. indicative, subjunctive, imperative

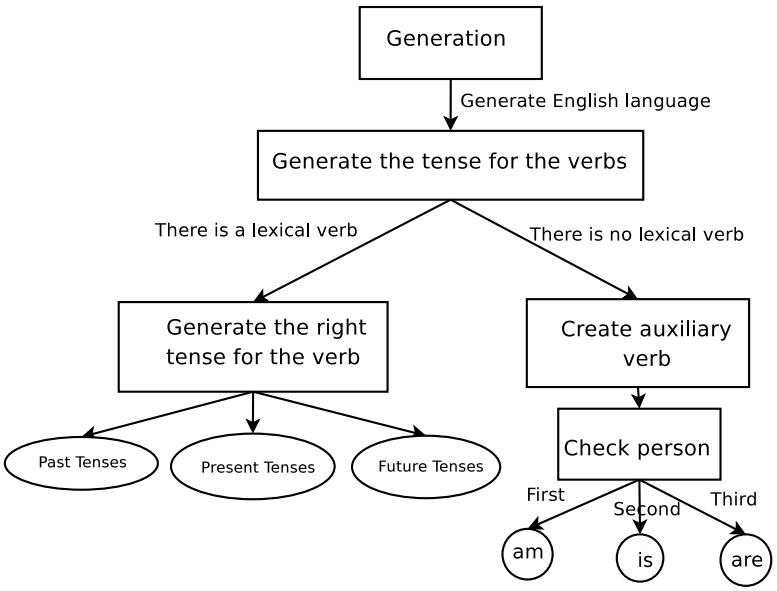

Figure 6.3: Generation the right tense for the verbs

and can be in one of many tenses. We discussed the special situation with reference to the intersection of Arabic tense and aspect in Chapter 2. The solution is to recognize the difference between morphological features and syntactic functional categories. The tense features must be expressed analytically.

Phase (12) English Sentence Generation The process of generating an English sentence can be as simple as keeping a list of rules. These rules can be extended through the life of the MT system. The system will use some operations in English such as vowel change: examples; man men. Sometimes this accompanies affixations: break broke broken `(= broke + en)`.

6.1.2 UniArab: Lexical representation in interlingua system

In transfer-based systems there are no problems if for a particular language pair there are one-to-one equivalents; the problems arise when there is more than one target word for a single source word. But for an interlingua in a multilingual system there are problems even if only one of the languages involved has two or more potential forms for a

single given word in one of the other languages. If an interlingua is to be completely language-neutral, it must represent not the words of one or another of the languages, but language-independent lexical units. Any distinction which is (or can be) expressed lexically in the languages of the system must be represented explicitly in the interlingua representation (Hutchins and Somers 1992). The UniArab system can generate a target language by classifying every Arabic word in the source text. There are six major parts of speech in Arabic. These are Verbs, Nouns, Adjectives, Proper nouns, Demonstratives, Adverbs and we create a seventh, so called 'other' category for Arabic words which do not fit into any of these six categories. The major parts of speech in the Arabic language have their own attributes, and we use these attributes within the UniArab system. For example, verbs in the Arabic language agree with their subjects in gender. Arabic words are masculine and feminine; there is no neutral gender. In the UniArab system we record the gender associated with a verb in the syntax for a particular subject NP. Adjectives and demonstratives also agree with the subject in gender too. In Arabic, words come into three categories with regards to number: They are (1) singular, indicating one, e.g. رجل *rğl* 'one man'. (2) dual, indicating two, e.g. رجلان *rğlān* 'two men' and (3) plural, indicating three or more e.g. رجال *rğāl* 'men'. The UniArab system records these attributes of gender and number. It is important to understand that source language specific features may not be used or may be different in the target language. For example, the Arabic number categ ory of dual is not relevant in English. The UniArab system is based on RRG and uses logical structures for each verb in the lexicon.

6.2 UniArab: Lexical representation in interlingua system based on RRG

Lexical frames represent the language-dependent lexicon. We use an XML data source to represent the UniArab lexicon. The lexicon creates pointers to corresponding conceptual frames or attributes of each word. These frames also have relations which link them to verb class frames, which are organized hierarchically according to the particular language.

Although we adhere to the Interlingua approach, we do not do so with the translation of lexical items. In an ideal Interlingua system lexical entries should be broken down into sets of semantic features. For example the word “man” is broken down into +human +male +adult. While this works in theory, in practice we cannot find enough semantic features to describe every entity in the world. For example “cow”, “computer” and “chair” cannot be described using these sets of semantic features unless we invent a unique semantic feature for every object and this is practically impossible.

6.2.1 Verb

In the UniArab system, we capture the information shown in Figure 6.4 for each verb. The verb information captured consists of *Arabic Verb*, *English Translation, Logical Structure, Tense, Gender, Person* and *Number*. The *Arabic Verb* represents one of the Arabic verbs in a specific tense, for a specific gender, person and number. The *English translation* is the English equivalent of the *Arabic verb*. The *Logical Structure* attribute is the RRG equivalent logical structure or lexical entry representation for the *Arabic Verb*. Arabic inflects verbs for tense and they agree in person, number and gender with the subject. In RRG, *Tense* is a verbal operator in the layer structure of the clause providing

information about the tense of this verb.

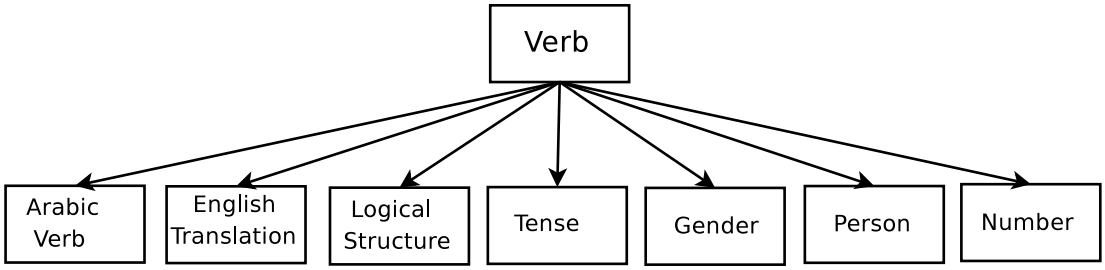

Figure 6.4: Information recorded on the Arabic verb

Table 6.1: Verb 1

Arabic verb	قرأ *qr'a*
English translation	read
Logical structure	[do'(x,[read'(x,(y)])]
Tense	past
Gender	m
Person	3rd
Number	singular

Table 6.2: Verb 2

Arabic verb	كتبت *ktbt*
English translation	wrote
Logical structure	[do'(x,[write'(x,(y)])]
Tense	past
Gender	f
Person	3rd
Number	singular

In the Arabic language, tense can be past or present as the primary distinction. *Gender* is an Arabic attribute of the verb. The verb agrees with the subject in gender. The *Person* attribute could be first, second or third person. The *Number* attribute refers to number of the subject. In Arabic, the number of a verb can be singular, dual or plural. Table 6.1 and Table 6.2 shows an example of one Arabic verb applied to different genders. The absence of ت *t* 't' suffix signifies m: gender. The English translation of these verbs are 'read' and 'wrote'.

6.2.2 Common noun

In the UniArab system, we capture the information shown in Figure 6.5 for each noun. The noun information captured consists of *Arabic Noun*, *English Translation*, *Definiteness*, *Gender* and *Number*. *Arabic Noun* represents a noun in the Arabic language. The *English translation* is the English equivalent of the *Arabic Noun*. *Definiteness* of the nouns can be definite or indefinite. *Gender* is an Arabic attribute of the noun.

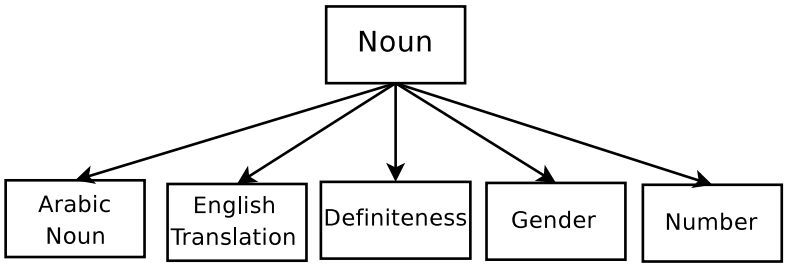

Figure 6.5: Information recorded on the Arabic noun

Table 6.3: Noun

Arabic noun	أشجار *ašğār*	الكتاب *ālktāb*
English translation	trees	the book
Definiteness	indefinite	definite
Gender	f	m
Number	plural	singular

The *Number* attribute refers to number of the noun. In the Arabic language number of nouns can be single, dual or plural. Table 6.3 shows examples of two different Arabic noun words, whose English translations are ‘trees’ and ‘book’. Please note that ‘book’ is def+, meaning ‘definite’.

6.2.3 Proper noun

Proper nouns in Arabic are not capitalized. In the UniArab system we capture the information shown in Figure 6.6. For each proper noun the system captures *Arabic proper noun*, *English translation*, *definiteness*, *gender* and *number*. *Arabic proper nouns* rep-

resents a proper noun in the Arabic language. The *English translation* is the English equivalent of the *Arabic proper noun*. *Gender* is an Arabic attribute of the proper noun. The *Number* attribute refers to the number of the proper noun; single, dual or plural.

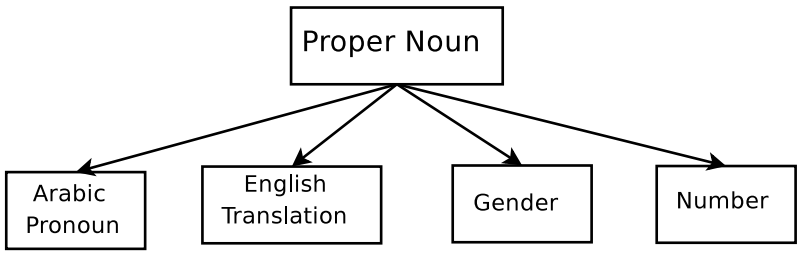

Figure 6.6: Information recorded on the Arabic proper noun

Table 6.4: Proper Noun

Arabic proper noun	عمر *ʔmr*	إيمان *ʔymān*
English translation	Omar	Eman
Gender	m	f
Number	singular	singular

Table 6.4 shows examples of two different Arabic proper noun words, whose English translations are ‘Omar’ and ‘Eman’.

6.2.4 Adjective

In the UniArab system, we capture the information shown in Figure 6.7 for each adjective. This consists of *Arabic Adjective, English Translation, Definiteness, Gender* and *Number*. *Arabic Adjectives* represent adjectives in the Arabic language. The *English translation* is the English equivalent of the *Arabic Adjective*. *Definiteness* can be definite or indefinite. *Gender* is an Arabic attribute of the adjective.

Table 6.5: Adjective

Arabic adjective	قصير *qṣyr*	الطويله *ālṭwylh*
English translation	short	the long
Definiteness	indefinite	definite
Gender	m	f
Number	singular	singular

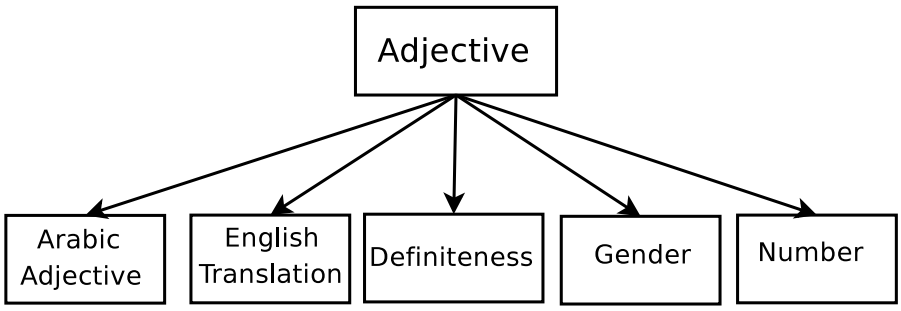

Figure 6.7: Information recorded on the Arabic adjective

The *Number* attribute refers to the number of the adjective. In the Arabic language number agreement for adjectives can be singular, dual or plural. Table 6.5 shows examples of two different Arabic adjective words, whose English translations are 'short' and 'long', please note that 'long' is def+.

6.2.5 Demonstrative

In the UniArab system we capture the information shown in Figure 6.8 for each demonstrative. this consists of *Arabic Demonstrative, English Translation, Demonstrative type, Gender* and *Number*. *Arabic Demonstratives* represents a demonstrative in the Arabic language. The *English translation* is the English equivalent of the *Arabic Demonstrative*. *Demonstrative type* can be, in the Arabic language, near to the speaker, far from the speaker or between near and far from the speaker. *Gender* is an Arabic attribute of the demonstrative. The *Number* attribute refers to number of the demonstrative. Table 6.6 shows examples of two different Arabic demonstratives, whose English translations are 'this' and 'that'.

Table 6.6: Demonstrative representative

Arabic demonstrative	هذا *hdā*	ذلك *dlk*	أولئك *ʾawlʾyk*
English translation	this	that	those
Demonstrative type	close	far	between near and far from the speaker
Gender	m	m	both m and f
Number	singular	singular	plural

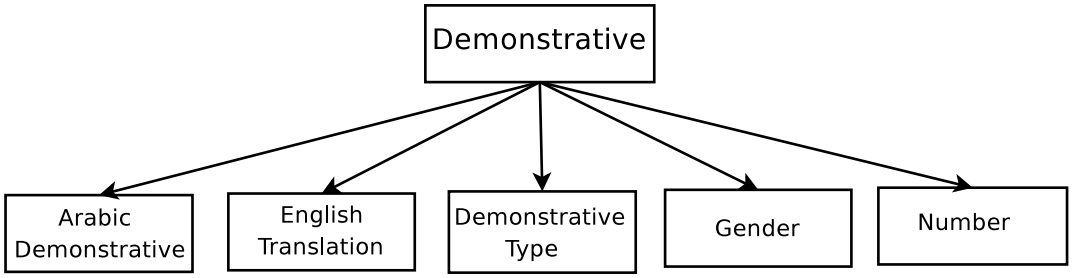

Figure 6.8: Information recorded on the Arabic demonstrative.

6.2.6 Adverb

In the UniArab system we capture the information shown in Figure 6.9 for each adverb. this consists of *Arabic Adverb*, *English Translation* and *Adverb type*. *Arabic Adverbs* represents an adverb in the Arabic language. The *English translation* is the English equivalent of the *Arabic Adverb*. *Adverbs type* refers to time or place (proposition), time such as ‘today’ or ‘tomorrow’ and places like ‘under’, ‘in’, or ‘on’ etc.

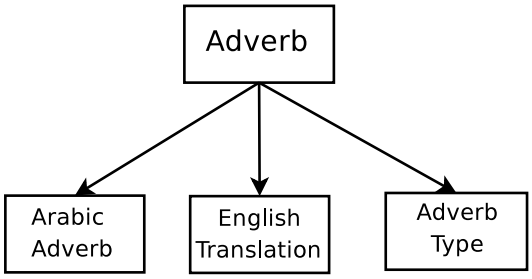

Figure 6.9: Information recorded on the Arabic adverb.

Table 6.7: Adverb

Arabic adverb	بجانب *bğānb*	اليوم *ālywm*
English translation	beside	today
Adverb type	Proposition	time

Table 6.7 shows examples of two different Arabic adverbs, whose English translations are ‘beside’ and ‘today’.

6.2.7 Other Arabic words

In the UniArab system, we capture the information shown in Figure 6.10 for each other Arabic word. This consists of *Arabic Other Word, English Translation, Logical Structure, Part of Speech, Tense, Gender, Person, Number* and *Definiteness*.

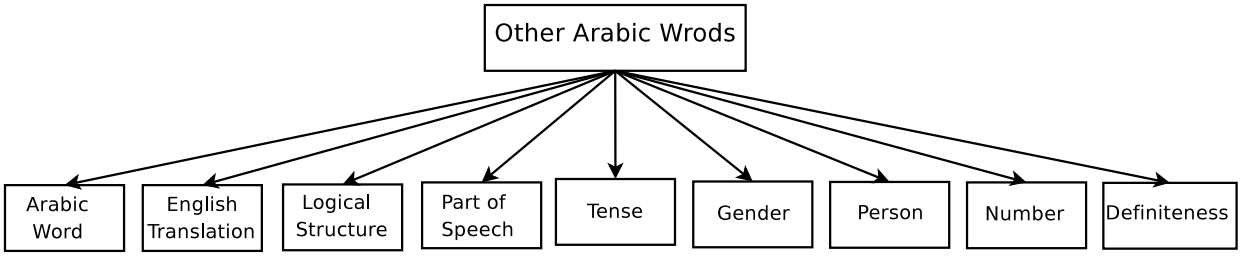

Figure 6.10: Information recorded on the other Arabic words

Table 6.8: Other Arabic words (where 'NON' means not applicable)

Arabic other words	و *w*	هي *hy*
English translation	and	she
Logical structure	NON	NON
Part of speech	conjunction	pronoun
Tense	NON	NON
Gender	NON	f
Person	NON	3rd
Number	NON	singular
Definiteness		

We allow a variety of attribute possibilities for the category 'other' in Arabic words for the moment. Table 6.8 shows examples of two different Arabic Other words, whose English translations are 'and' and 'she'.

6.3 UniArab: Generation

The target language generation phases in the UniArab system follow the syntactic realization model. Generation takes as input, the universal logical structure of the input sentence(s) and produces as output a morphology-syntactic realisation of the sentence in the target language. The UniArab system is designed as a universal machine translator,

which means that it can support translation of the Arabic into any other natural language with the addition of additional language generation bridges. The UniArab system is evaluated using Arabic as source language into English as the target language. In the UniArab system phases 9, 10, 11 and 12 are for generation of the target languages, in our case this is English. For the example given under Phase 8 in Section 6.1.1, قرأ خالد الكتاب *qr'a ẖālد ālktāb, Khalid read the book*, we have the logical structure:

`Verb read [do'(x,[read'(x,(y)])] sg 3rd M PAST` قرأ *qr'a*>

where the Proper Noun is `Khalid sg unspec M` خالد *ẖālد*

and the Noun is `the book sg def M` الكتاب *ālktāb*

Firstly, the *Semantic to Syntactic* phase determines the actor and undergoer assignments, assigns the core arguments and assigns the predicate in the nucleus. In the UniArab system we keep all word attributes whether they are used in the target language or not. In this case, the gender of the noun *the book*, in Arabic is masculine, but in English *book* has neutral gender. In Phase 10, *Syntax Generation*, and Phase 11, *Generate English Morphology*, UniArab uses target language rule to generate the syntax. The verb logical structure gives UniArab a flag indicating how many arguments this verb takes. In this case the logical structure will be `read[do'(x,[read'(x,(y)])]`. Now the UniArab system replaces x with *Khalid*, and y with *the book*. The UniArab system now holds the following for this simple sentence:

`read[do'(Khalid,[read'(Khalid,(the book)])]`.

In the last phase, *English Sentence Generation*, the UniArab system builds the final shape of a sentence: *Khalid read the book*. Moreover, there are some special cases, like the UniArab system adding verb to be or changing the verb tense of the source language to

another tense in the target language. Also, the role of word order in the target language must be considered.

الكتاب *ālktāb* book:N خالد *ẖālد* Khalid:MsgN قرأ *qr'a* read:V

read [**do**'(x,[**read**'(x,(y))] sg 3rd M PAST قرأ *qr'a*

The results of the parse can be seen here with LS as :

`Verb read [do'(x,[read'(x,(y))] sg 3rd M PAST` قرأ *qr'a*>

where the Proper Noun is `Khalid sg unspec M` خالد *ẖālد*

and the Noun is `the book sg def M` الكتاب *ālktāb*

At this point the generation will start; first of all the semantic to syntactic phase determines the actor and undergoer assignments, assign the core arguments and assign the predicate in the nucleus. In the last phase, *English Sentence Generation*, the UniArab system builds the final shape of a sentence: *Khalid read the book.* Moreover, there are some special cases, like the UniArab system adding the verb 'to be' or ensure the verb tense of the source language is reflected as the appropriate tense in the target language. Also, the rules of word order in the target languages must be considered.

6.4 UniArab: Screen design

The graphical user interface (GUI) of UniArab is interactive. Designing the visual composition and temporal behaviour of the GUI is an important aspect of the design of UniArab. We use one text area to allow a user to input source language sentences, two buttons, *Enter* to submit the text to the system, *Clear* to delete all text in the input and output text areas. There is a separate text area for output of the translated text. Also there is a text area for logical structure output of every sentence.

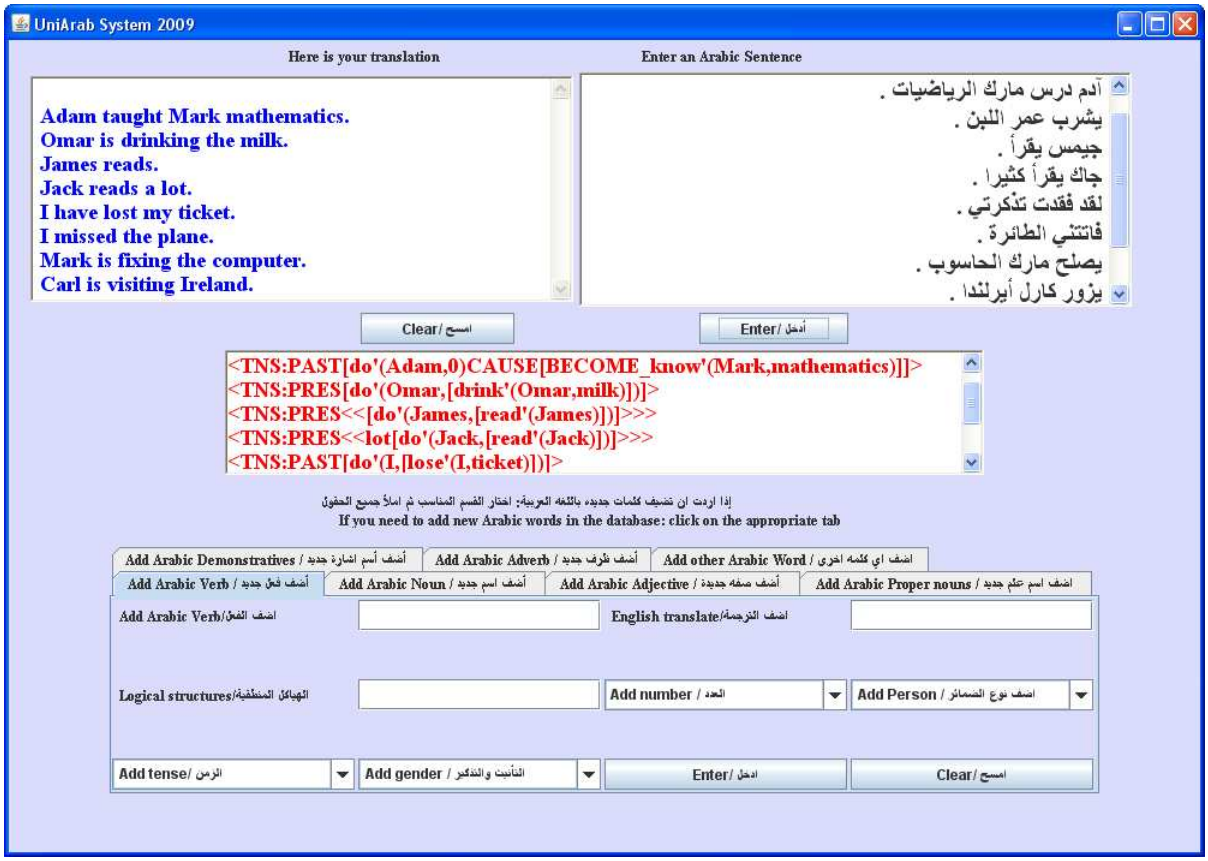

Figure 6.11: UniArab's GUI 1

If a user needs to add a new Arabic word in the UniArab system datasource; he/she can click on the appropriate tab. There are seven different tabs each representing a category of words in the Arabic language; *Add Arabic Verb, Add Arabic Noun, Add Arabic Adjective, Add Arabic Proper nouns, Add Arabic Demonstratives, Add Arabic Adverb and Add other Arabic Word.* In every tab there are a number of combo boxes. A combo box is a combination of a drop-down list or list box, allowing the user to choose from the list of existing options. For example, when a user needs to add a new adjective to the datasource, the user will be presented with a text field to let him/her enter an Arabic adjective. There is another text field for adding the English equivalent of the Arabic adjective. There are a number of combo boxes; number, definition and gender, a user chooses from the list an option. There are two buttons under each tab, *Enter* to submit the information into the

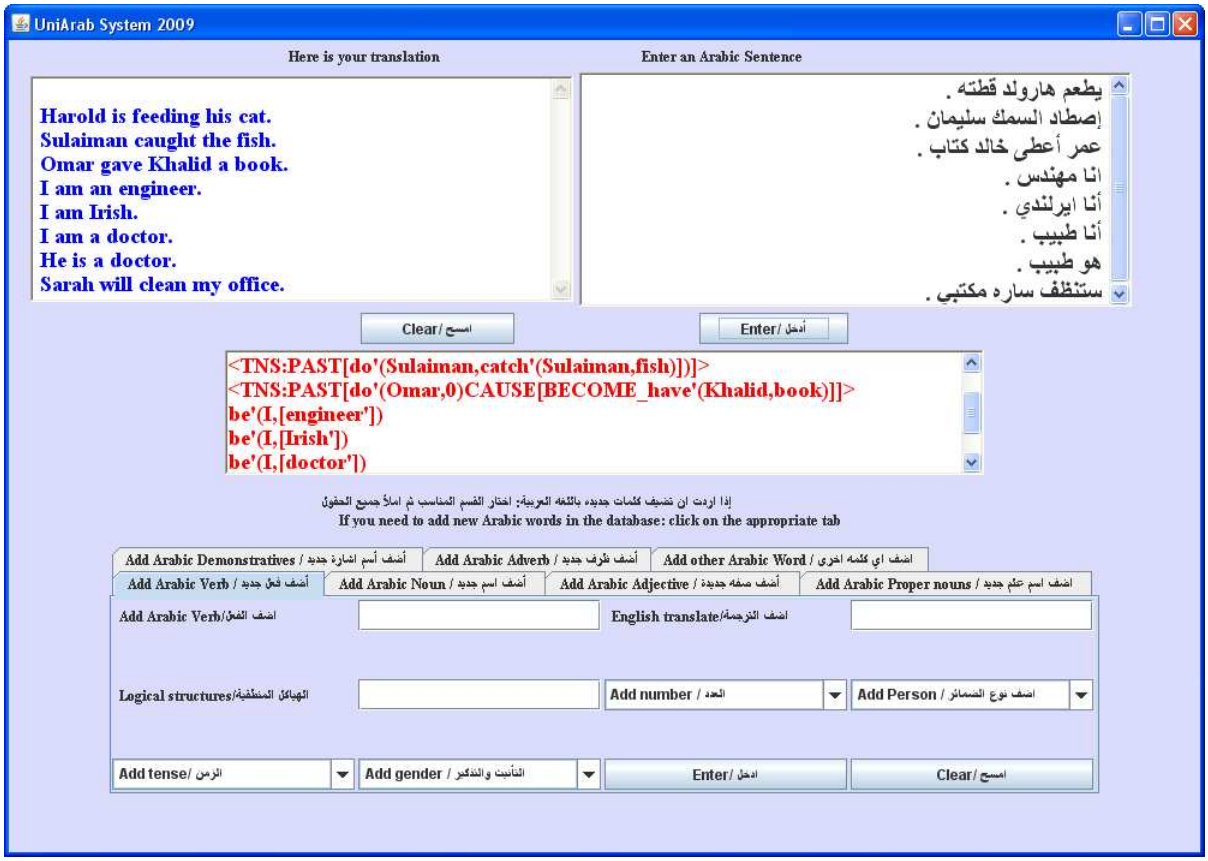

Figure 6.12: UniArab's GUI 2

system, and *Clear* to delete all words in all text fields and return combo boxes to their default state. Figures 6.11,6.12,6.13 shows GUI of the UniArab system.

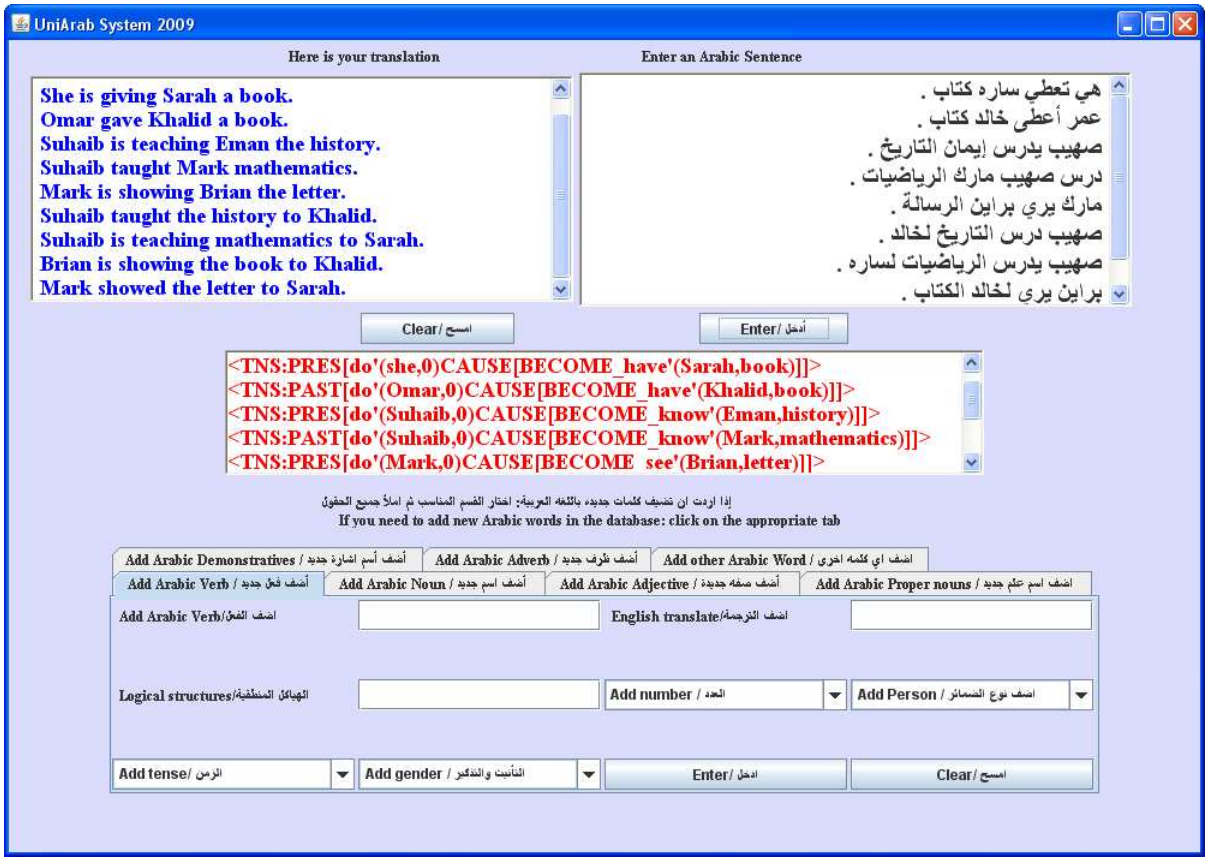

Figure 6.13: UniArab's GUI 3

6.4.1 Lexicon interface

In order to allow for robust user interaction with the lexicon, we use a graphical interface to capture the information for each part of speech. The user selects the part of speech of the word he is adding, and is then presented with only the options relevant to it. The interface also limits the user's selections to acceptable values and ensures that all attributes are filled. With this technique, we minimize the risk of human error, and therefore the information is more accurate. The graphical interface is quicker and easier when a user adds a new word in the lexical (XML data source). When the system displays an information error. Figure 6.14 shows the entry interface that is implemented as part of the UniArab system.

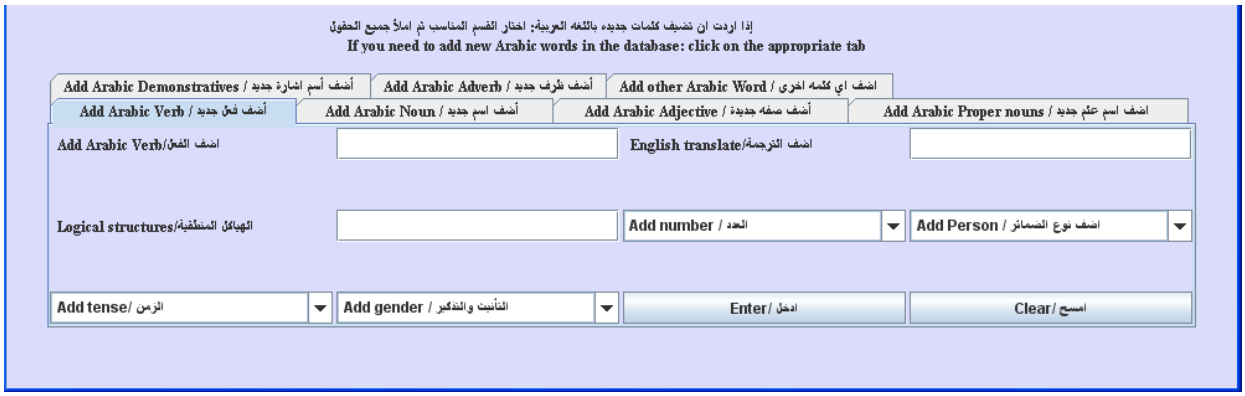

Figure 6.14: UniArab's lexicon interface

6.5 Technical challenges

Arabic letters in the GUI We can not write Arabic letters in UniArab's GUI. We use Unicode to represent them. *Unicode Converter System* allows us to enter Arabic text and click on a button to get the equivalent Unicode of the text.

Arabic letters in Eclipse IDE for Java We used Eclipse IDE for Java development. We can not write Arabic as a string in Eclipse. While Java does support Arabic, the problem lies in the operating system not supporting Arabic letter shapes in IDE. We used Windows XP and Windows 2000 which both have the same problem. To fix this we changed to Ubuntu Linux. Under Linux we can write Arabic text as a string in the Eclipse IDE.

Arabic in data source We choose to create our data source as XML, for optimum support or different platforms. It was also easier as we used Arabic letters not Unicode inside the data source. XML fully supports Arabic. We created our search engine using Java. We used a HashMap to make the keyword in Arabic when we search inside the datasource. We used $verbMap.containsKey(word)$ in order to check the presence of an Arabic word in the data source.

6.6 Summary

We presented the conceptual structure and architecture of the UniArab system. We discussed each of the phases from source language analysis, through the logical representation, then the generation of the target sentences. We detailed the lexical properties of Arabic sentences and the attributes for each type of word. We discussed how generation maps the logic structure to the target language. Finally, we discussed the user interface and some of the problems encountered during development.

7

Testing and evaluation

This chapter presents the results of the evaluation. Evaluation of MT software is necessary in order to improve system performance and analyse potential problems and, of course, its accuracy and effectiveness. In the evaluation session we consider many different aspects of the MT system including quality of translation, time for translation ability to add a new word in the lexicon of the system and resource utilization.

7.1 Evaluation of MT systems

The evaluation of MT systems is a difficult task. This is not only because many different metrics are involved, but also because translation is itself difficult (Laoudi et al. 2004). The first important aspect for a potential test is to determine the translational capability. Therefore, we need to draw up a complete overview of the translational process, in all its different aspects. A good translation has to effectively capture the meaning. This involves establishing the size of the translation task, is it machine legible and if so, according

to which standards? Current general function MT systems can not translate all texts consistently. Output can have very poor quality. It is important to mention that the 'subsequent editing required' increases as translation quality gets poorer (Turian et al. 2003).

Given the limited lexicon implemented in this work so far, we evaluate the effectiveness and accuracy of UniArab by comparison. We create variants of Arabic sentences that represent all possible structures of the sentences that UniArab can translate. We then compare between human-translated and machine-translated versions.

7.2 Sentence tests

We have sentences (for actual test examples see Appendix C) in Arabic and their equivalent translations in English. We have covered a representative broad selection of verbs across intransitive, transitive and ditransitive constructions in simplex sentences in active voice. Complex sentences are beyond the thesis scope. However, we do address copula-like nominative clauses in Arabic. We tested UniArab in more than one way. We tested single sentences and multiple sentences. UniArab easily deals with more than one sentence as input and its output matches. We entered random sentences together in one input or as individual sentences.

7.2.1 Verb-Subject with one argument in different tenses

Table 7.1: Test : Verb-Subject; one argument

Arabic	يشرب عمر اللبن *yšrb ʿmr āllbn*
Human	Omar is drinking the milk.
Google	Omar drink milk
Microsoft	drink milk Omar
UniArab	Omar is drinking the milk .

In Table 7.1, the output of the Google translator (Google 2009) is faulty in tense and verb 'to be'. Microsoft's MT (Microsoft 2009) failed to translate most of the sentence in tense, verb and word order. UniArab successfully translates the sentence in its entirety. Figure 7.1 shows this sentence output in the UniArab system.

Figure 7.1: Verb-Subject with one argument

Table 7.2: Test : Verb-subject; agreement 1

Arabic	شرب عمر اللبن *šrb ʕmr ālbn*
Human	Omar drank the milk
Google	Omar drinking milk
Microsoft	drinking milk Omar
UniArab	Omar drank the milk.

In Table 7.2, the output of the Google translator is faulty in tense and definition. The Microsoft translator failed to translate most of the sentence in tense, definition and word order. UniArab successfully translates the sentence in its entirety. Figure 7.2 shows this sentence output in the UniArab system.

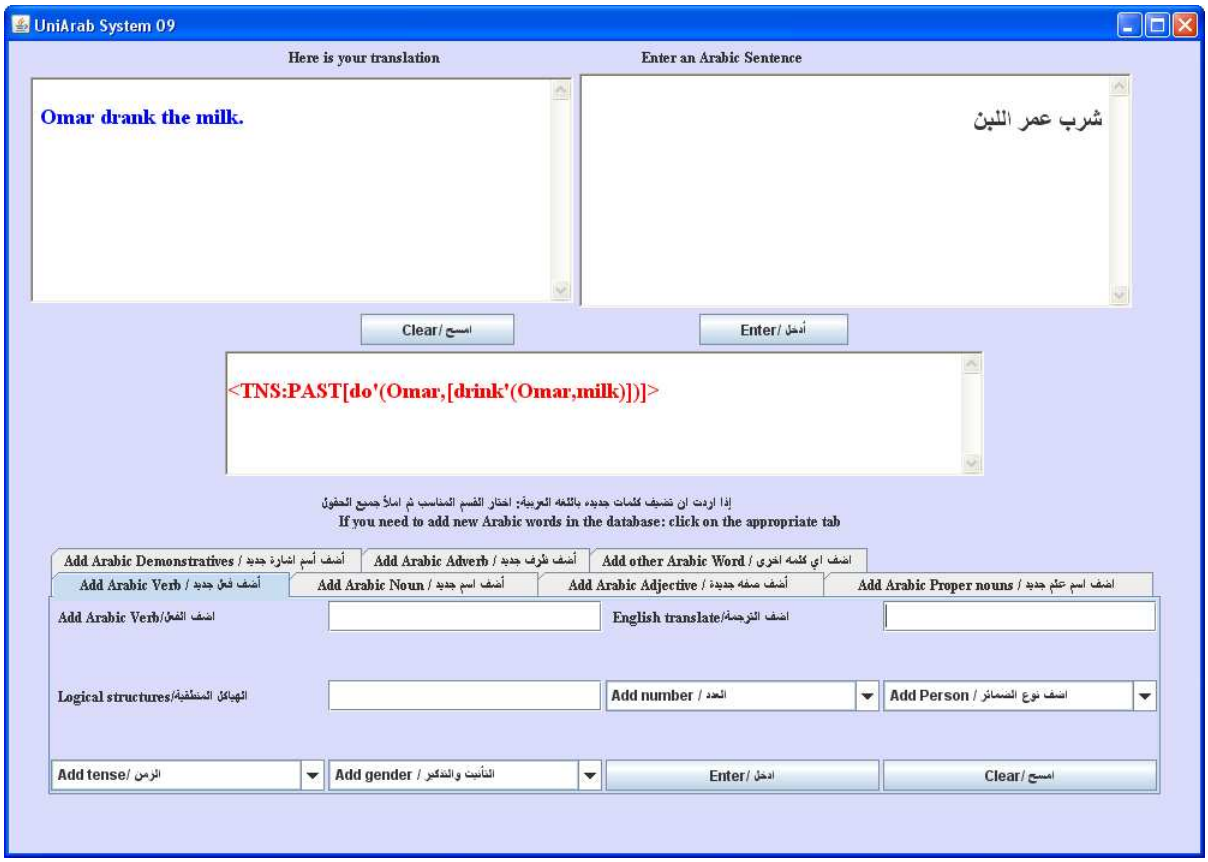

Figure 7.2: Verb-Subject with one argument

Table 7.3: Test : verb-subject; agreement 2

Arabic	خالد قرأ الكتاب *ẖālد qrʿa ālktāb*
human-translated	Khalid read the book
Google	Khalid read the book
Microsoft	Khaled read book
UniArab	Khalid read the book.
Arabic	سيشرب خالد اللبن *syšrb ẖāld āllbn*
human-translated	Khalid will drink the milk
Google	Khalid drink milk.
Microsoft	Khaled drink milk.
UniArab	Khalid will drink the milk.

In Table 7.3, the output of the Google translator is successful. Microsoft's MT failed to translate the definition. UniArab successfully translates the sentence in its entirety. In the output of the second sentence, the Google translator is faulty in tense and definition. Microsoft's MT failed to translate the tense and definition. UniArab successfully translates the sentence in its entirety. This is becouse of the RRG lexicalist approach in the interlingua. Figures 7.3 and 7.4 show this sentence output in the UniArab system.

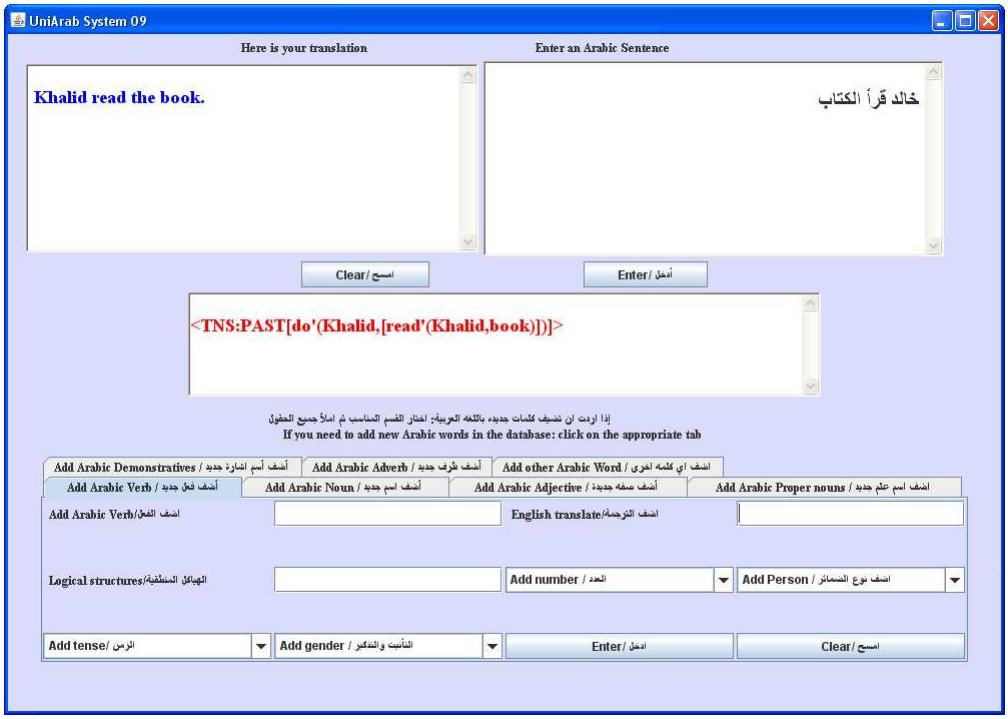

Figure 7.3: Verb-subject agreement 1

Figure 7.4: Verb-subject agreement 2

7.2.2 Gender-ambiguous proper nouns

Table 7.4: Test : Gender-ambiguous proper nouns 1

Arabic	قرأ جاك الكتاب *qrʾa ǧāk ālktāb*
human-translated	Jack read the book
Google	Jack read the book
Microsoft	read Jack book
UniArab	Jack read the book.

In Table 7.4, the output of the Google translator is successful. Microsoft's MT failed to translate the definition. UniArab successfully translates the sentence in its entirety. Figure 7.5 shows this sentence output in the UniArab system.

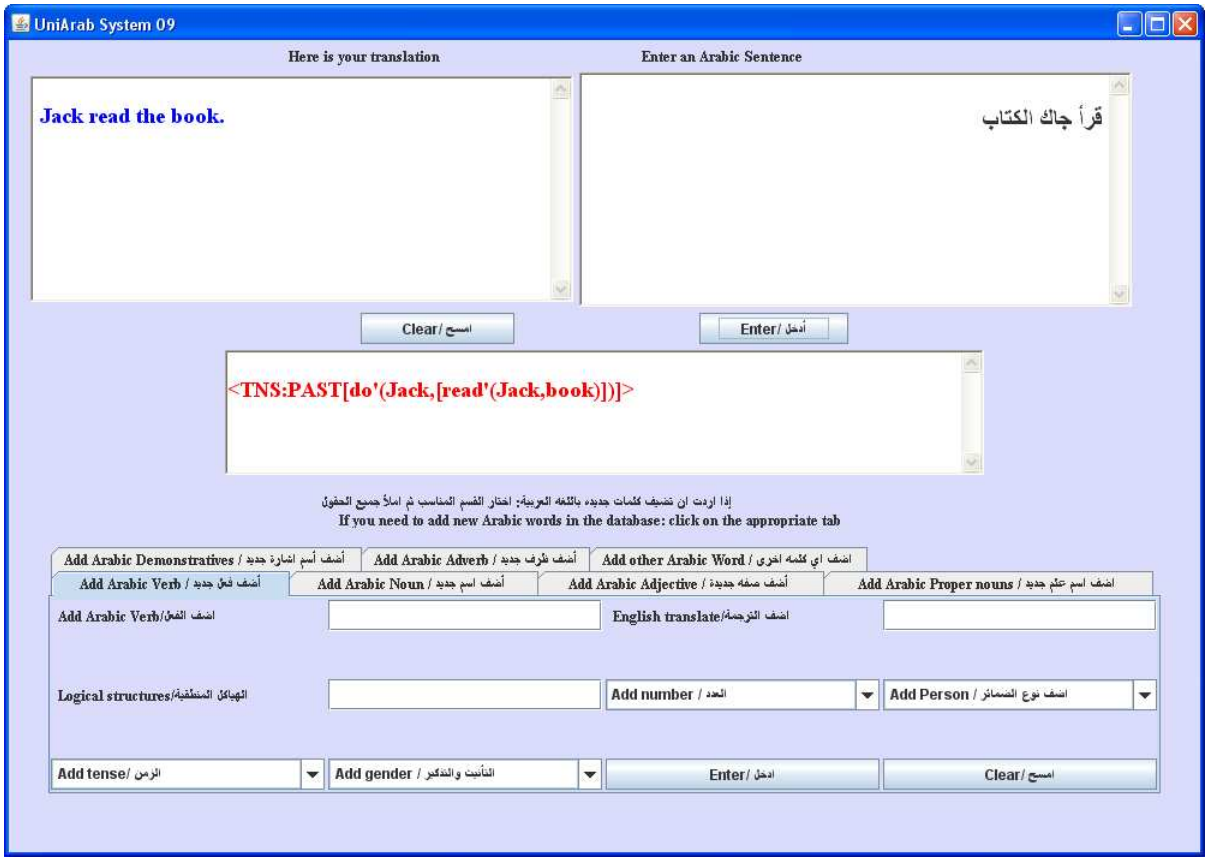

Figure 7.5: Gender-ambiguous proper nouns 1

Table 7.5: Test : gender-ambiguous proper nouns 2

Arabic	قرأت ماري الكتاب *qrʿat māry ālktāb*
human-translated	Mary read the book
Google	Marie read the book
Microsoft	read Marrie book
UniArab	Mary read the book.

In Table 7.5, the output of the Google translator is successful. Microsoft's MT failed to translate the definition and word order. UniArab successfully translates the sentence in its entirety. Figure 7.6 shows this sentence output in the UniArab system.

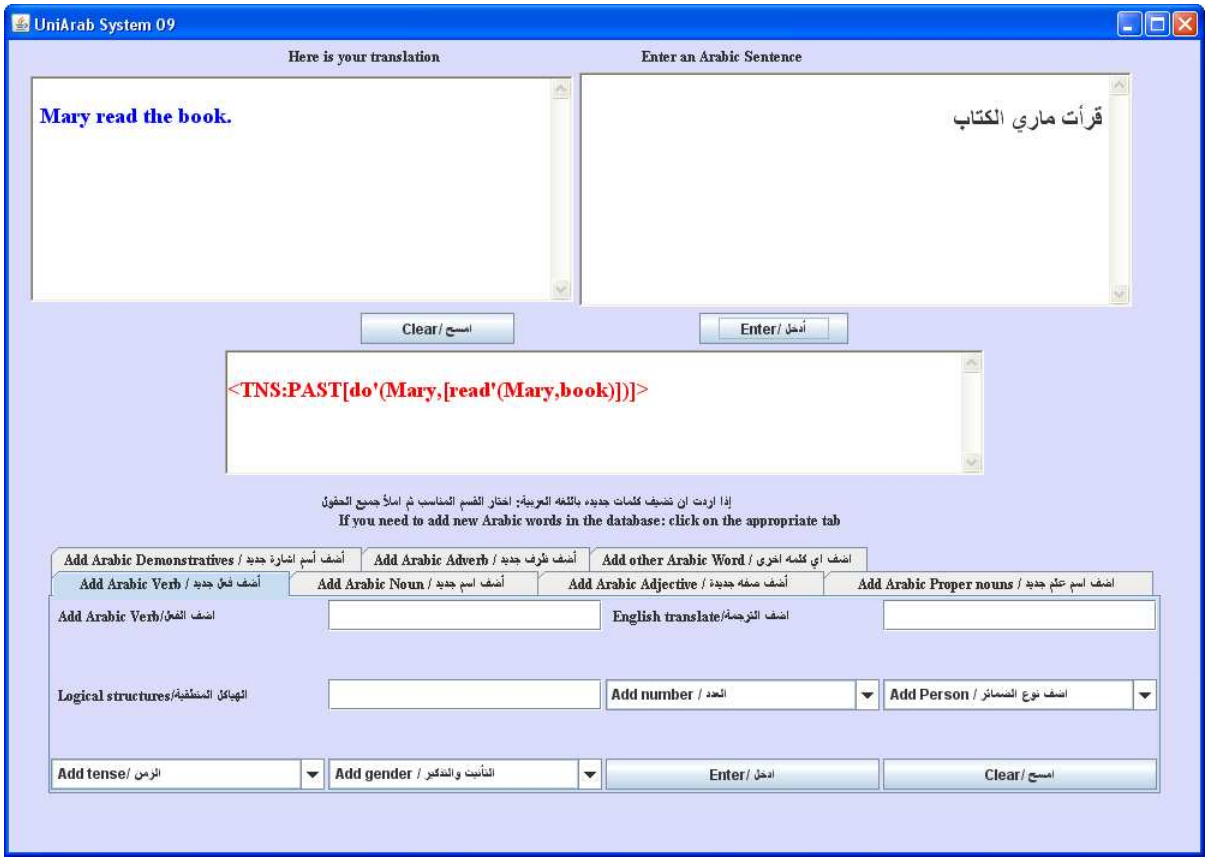

Figure 7.6: Gender-ambiguous proper nouns 2

7.2.3 Verb ‘to be’

Table 7.6: Test : Verb ‘to be’ 1

Arabic	هو مهندس *hw mhnds*
human-translated	He is an engineer.
Google	Is the architect of
Microsoft	is the engineer
UniArab	He is an engineer.

In Table 7.6, the output of the Google translator is faulty. Microsoft’s MT successfully translated the person. UniArab successfully translates the sentence in its entirety. Figure 7.7 shows this sentence output in the UniArab system.

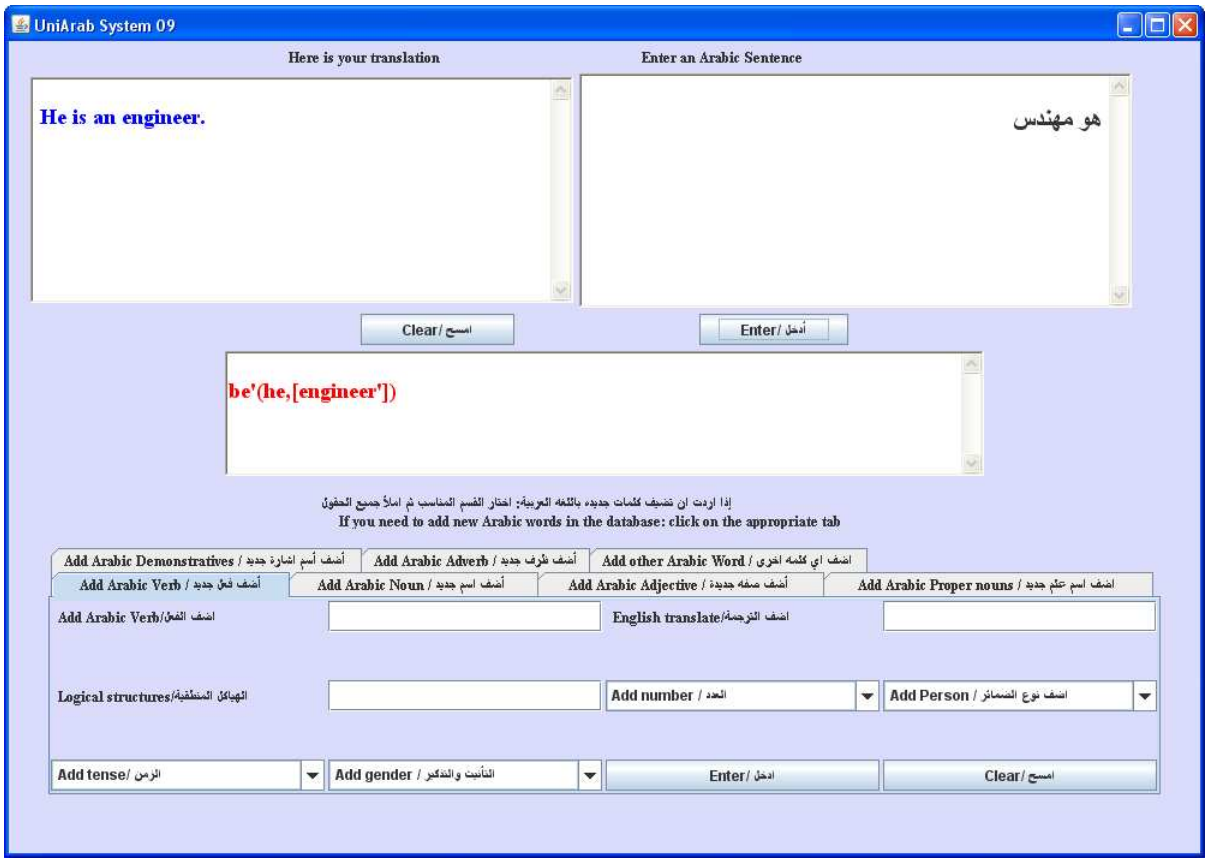

Figure 7.7: Verb ‘to be’ 1

Table 7.7: Test : Verb ‘to be’ 2

Arabic	أنا المهندس ʾanā ālmhnds
human-translated	I am the engineer.
Google	I Engineer
Microsoft	i am engineer
UniArab	I am the engineer.

In Table 7.6, the output of the Google translator is faulty in the verb ‘to be’ and definition. Microsoft’s MT successfully translated the verb ‘to be’, it is faulty in the definition only. UniArab successfully translates the sentence in its entirety. Figure 7.8 shows this sentence output in the UniArab system.

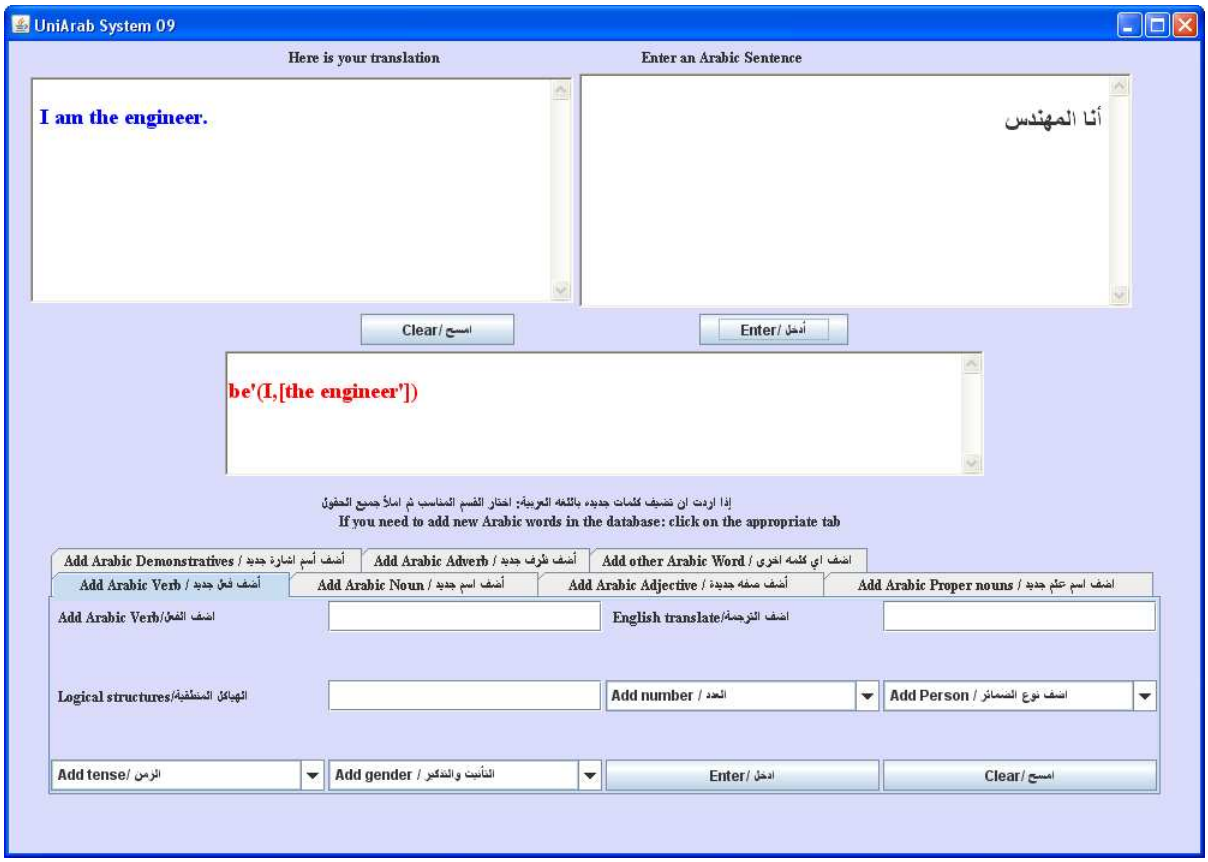

Figure 7.8: Verb ‘to be’ 2

7.2.4 Verb ‘to have’

Table 7.8: Test : Verb ‘to have’ 1

Arabic	لقد قمت بالحجز *lqd qmt bālḥğz*
human-translated	I have made a reservation.
Google	I have made a reservation
Microsoft	You have a booking
UniArab	I have made a reservation.

In Table 7.8, the output of the Google translator is successful. Microsoft’s MT is faulty in person. UniArab successfully translates the sentence in its entirety. Figure 7.9 shows this sentence output in the UniArab system.

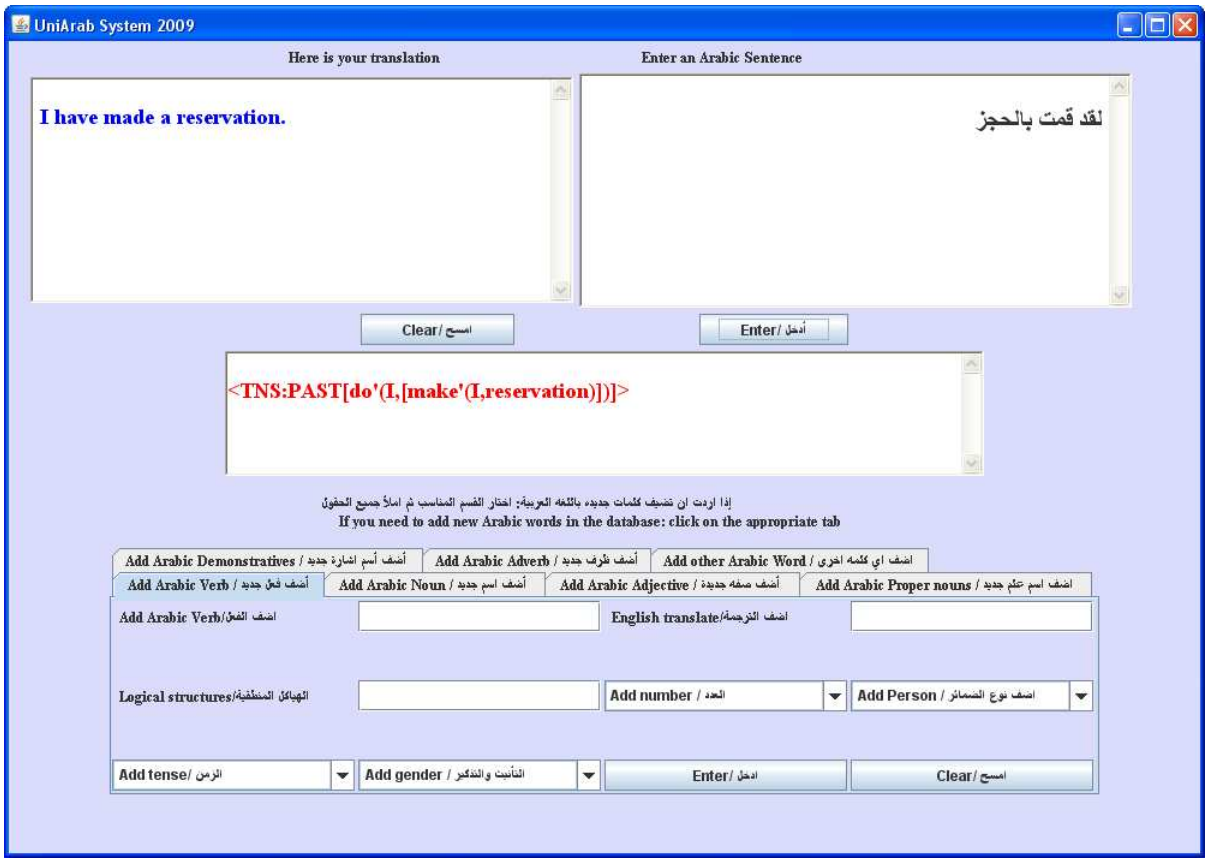

Figure 7.9: Verb ‘to have’ 1

Table 7.9: Test : Verb ‘to have’ 2

Arabic	لقد فقدت تذكرتي *lqd fqdt tdkrty*
human-translated	I have lost my ticket.
Google	I’ve lost my ticket
Microsoft	i have lost تذكرتي *tdkrty*
UniArab	I have lost my ticket.

In Table 7.9, the output of the Google translator is successful. Microsoft’s MT is faulty in the object word. UniArab successfully translates the sentence in its entirety. Figure 7.10 shows this sentence output in the UniArab system.

Figure 7.10: Verb ‘to have’ 2

7.2.5 Free word order

Here we show three Arabic sentences with different word order which translate to the same English output.

Table 7.10: Test : Free word order (Verb Noun Noun scenario one)

Arabic	يحب قيس ليلى *yḥb qys lylā*
human-translated	Qays loves Laila
Google	Qais likes of Laila
Microsoft	Love Qais laili
UniArab	Qays loves Laila

In Table 7.10, the output of the Google translator is faulty in the verb meaning and the system added ‘of’ without any meaning in this sentence. Microsoft’s MT translated each word while ignoring the word order and meaning of the sentence. UniArab successfully translates the sentence in its entirety. Figure 7.11 shows this sentence output in the UniArab system.

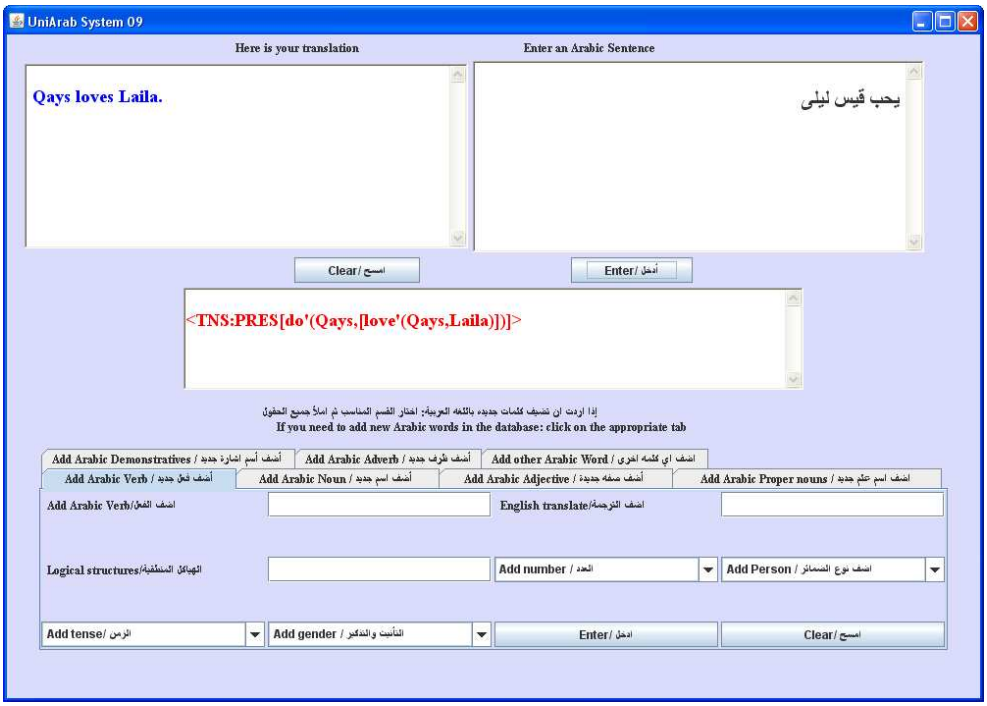

Figure 7.11: Free word order (Verb Noun Noun scenario one)

Table 7.11: Test : Free word order (Verb Noun Noun scenario two)

Arabic	يحب ليلى قيس *yḥb lylā qys*
human-translated	Qays loves Laila
Google	Leila loves measured
Microsoft	Love laili Qais
UniArab	Qays loves Laila

In Table 7.11, the second ordering possibility is shown. The output of the Google translator is faulty in the actor, the system can not analyse 'who does what', the actor is Qais but the output makes the object the subject. Microsoft's translator translates each word while ignores the word order and the meaning of the sentence. It also makes the object the subject. UniArab successfully translates the sentence in its entirety. Figure 7.12 shows this sentence output in the UniArab system.

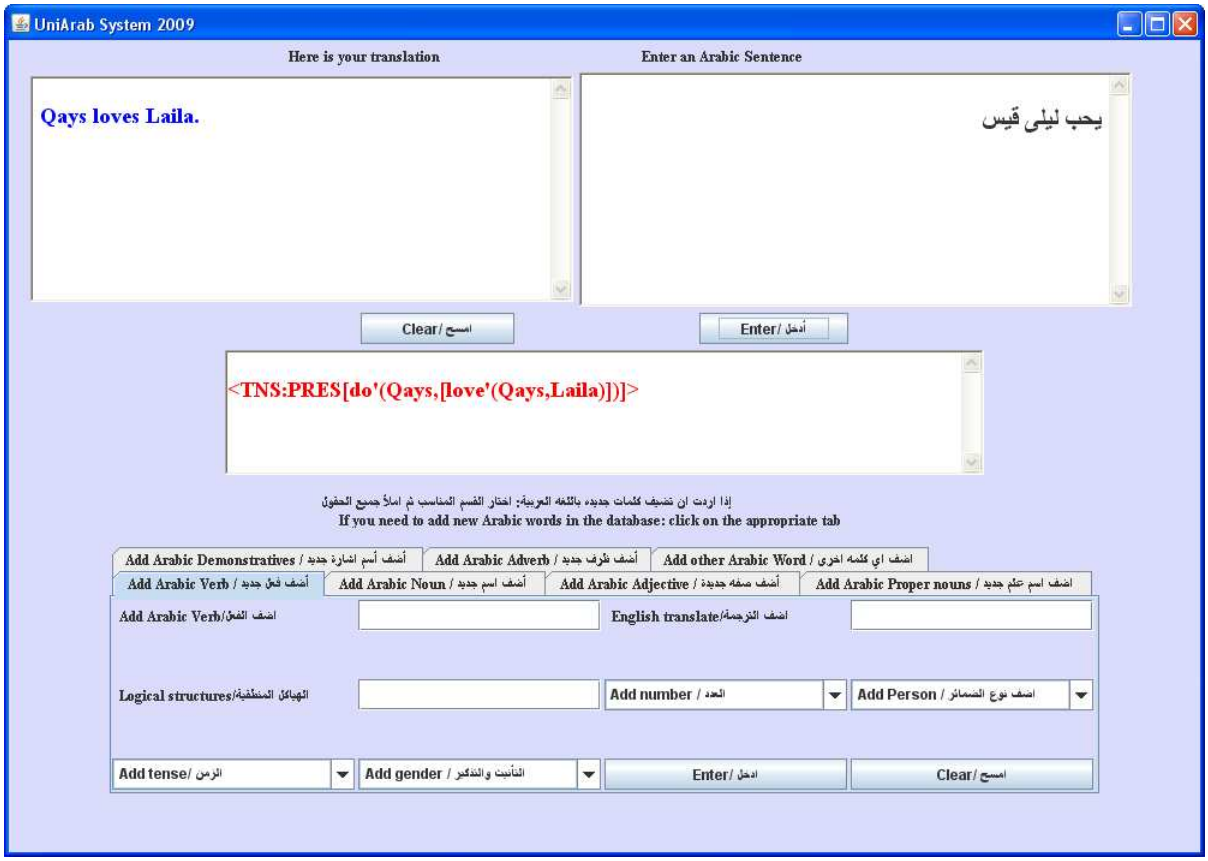

Figure 7.12: Free word order (Verb Noun Noun scenario two)

Table 7.12: Test : Free word order (Verb Noun Noun scenario three)

Arabic	قيس يحب ليلى *qys yḥb lylā*
human-translated	Qays loves Laila
Google	Qais likes of Laila
Microsoft	Qais love laili
UniArab	Qays loves Laila

Table 7.12 shows the third possible sentence order. The output of the Google translator is faulty in verb meaning and adds an extra 'of' which does not carry any meaning. Microsoft's MT translates each word while ignoring the word order, tense and meaning of the sentence. UniArab successfully translates the sentence in its entirety. Figure 7.13 shows this sentence output in the UniArab system.

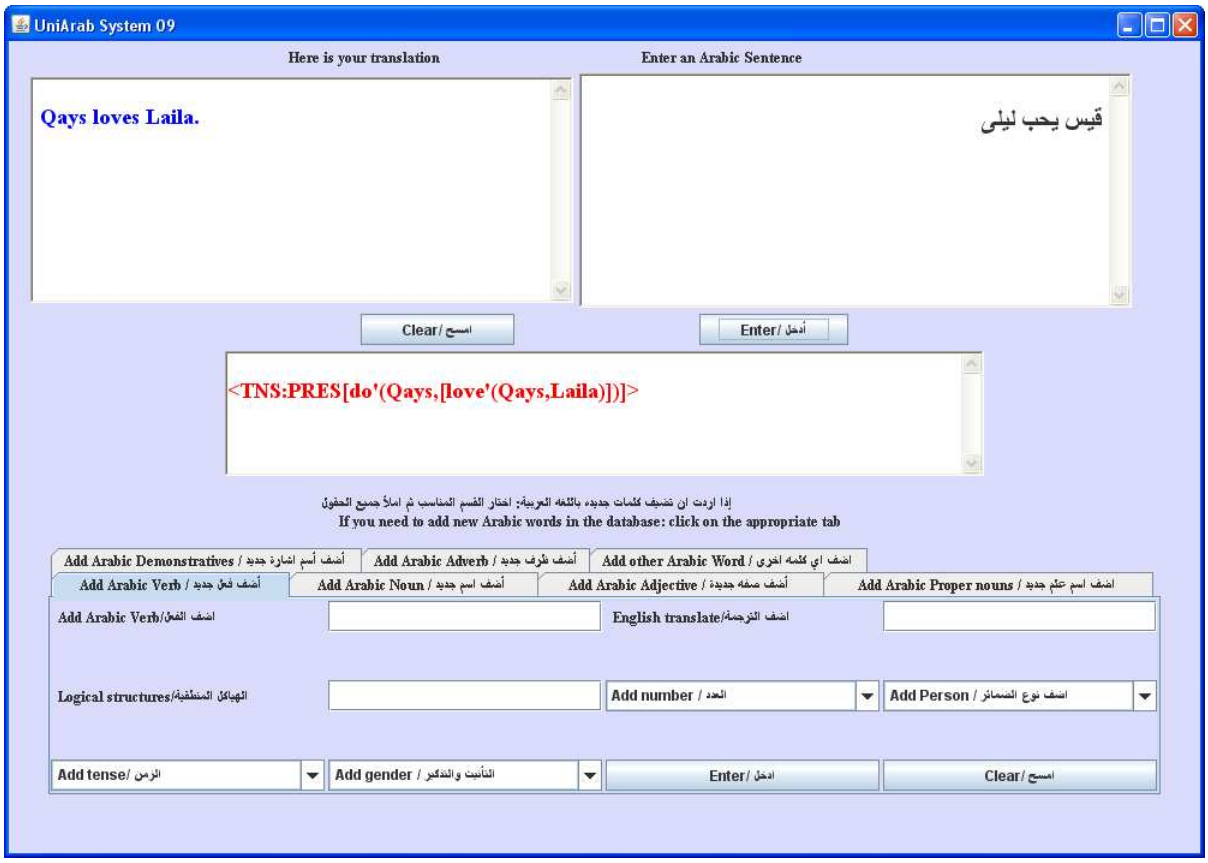

Figure 7.13: Free word order (Verb Noun Noun scenario three)

7.2.6 Pro-drop

Table 7.13: Test: Pro-drop

Arabic	فاتتني الطائرة *fāttny ālṭā-yrh*
human-translated	I missed the plane.
Google	Missed the plane
Microsoft	فاتتني *fāttny* plane
UniArab	I missed the plane.

Table 7.13 shows an example of a pro–drop sentence. The output of the Google translator is faulty in the point of pro-drop; the system did not find the subject. Microsoft's MT did not recognize the important word in the sentence and passed it through to the output. UniArab successfully translates the sentence in its entirety. Figure 7.14 shows this sentence output in the UniArab system.

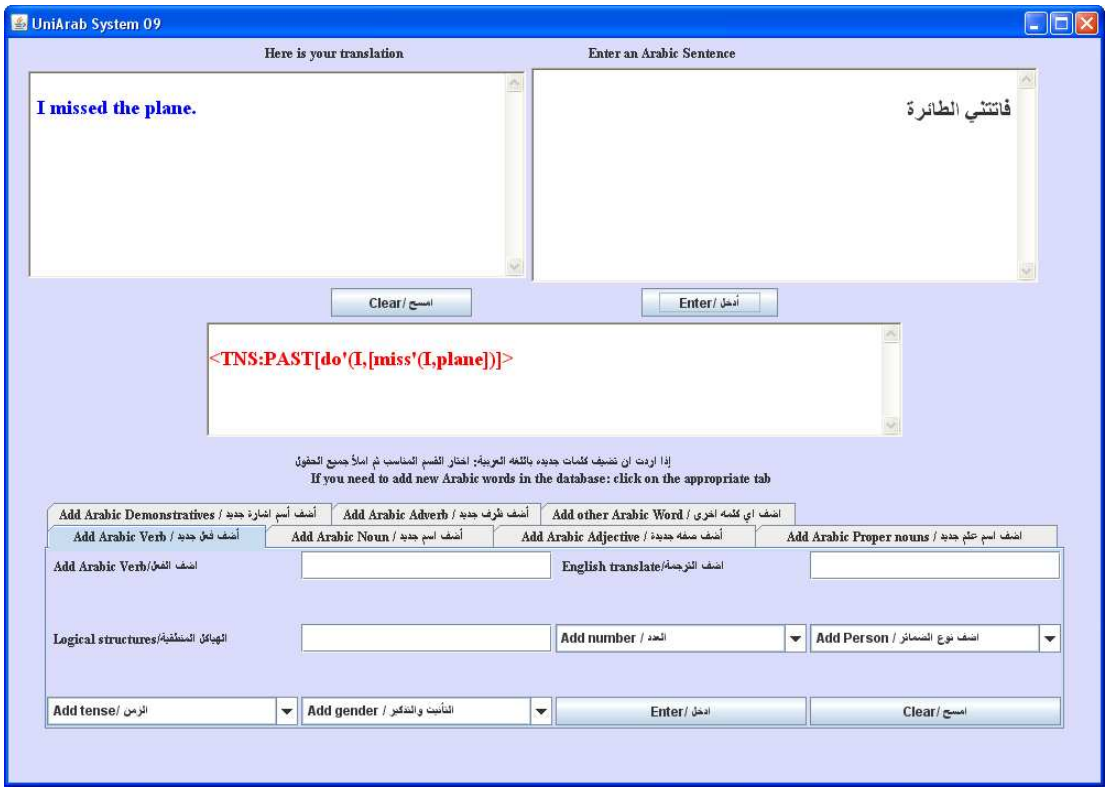

Figure 7.14: Pro-drop

7.2.7 Transitivity of verbs

In Arabic and English, we can classify verbs as both intransitive, transitive and ditransitive.

7.2.7.1 Intransitive

Table 7.14: Test : Intransitive 1

Arabic	صهيب يقرأ *ṣhyb yqrʔa*
human-translated	Suhaib reads.
Google	Suhaib read
Microsoft	suhaib reads
UniArab	Suhaib reads.

Table 7.14 shows an example of an intransitive sentence. The output of the Google translator is faulty in tense. Microsoft's and UniArab translators successfully. Both systems are translate the sentence in its entirety. Figure 7.15 shows this sentence output in the UniArab system.

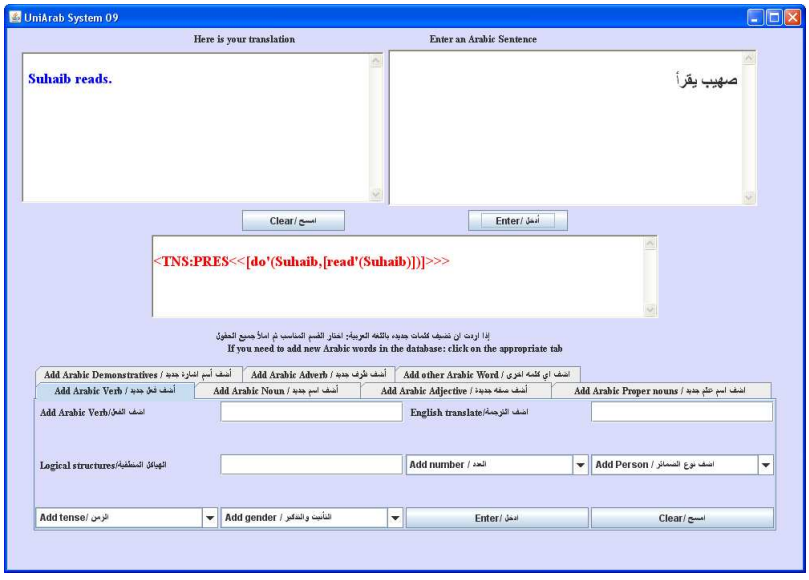

Figure 7.15: Intransitive

Table 7.15: Test : Intransitive 2

Arabic	صهيب يقرأ كثيرا *ṣhyb yqr·a ktyrā*
human-translated	Suhaib reads a lot.
Google	Souhaib read a lot
Microsoft	suhaib reads much
UniArab	Suhaib reads a lot.

Table 7.15 shows an example of an intransitive with an adverb. The output of the Google translator has given the wrong tense. Microsoft's and UniArab translators are both successful. Both systems are translate the sentence in its entirety, though the Microsoft output is more formal. Figure 7.16 shows this sentence output in the UniArab system.

Figure 7.16: Intransitive with an adverb

7.2.7.2 Transitive

Table 7.16: Test : Transitive

Arabic	يصلح مارك الحاسوب *yṣlḥ mārk ālḥāswb*
human-translated	Mark is fixing the computer.
Google	Mark works computer
Microsoft	Marc computer works
UniArab	Mark is fixing the computer.

In Table 7.16, the output of the Google and Microsoft's translators are faulty in the verb 'to be' and the meaning of the verb. UniArab successfully translates the sentence in its entirety. Figure 7.17 shows this sentence output in the UniArab system.

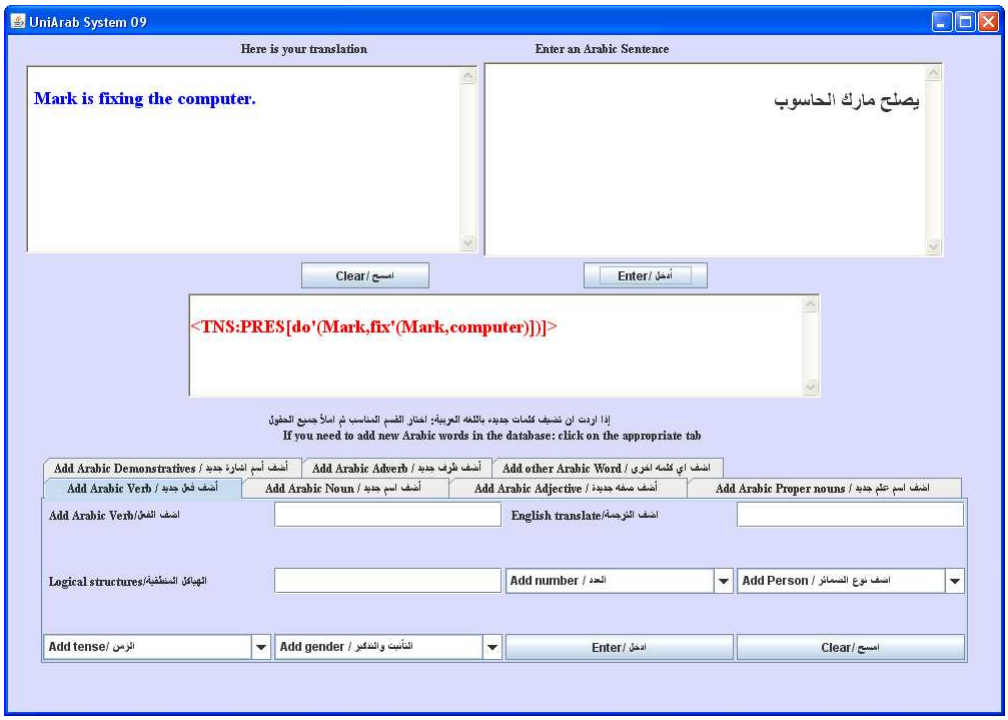

Figure 7.17: Transitive

7.2.7.3 Ditransitive

Table 7.17: Test : Ditransitive 1

Arabic	هو أعطى خالد كتاب *hw ʾaʿṭā ẖālid ktāb*
human-translated	He gave Khalid a book.
Google	Khaled was given a book
Microsoft	is given Khaled book
UniArab	He gave Khalid a book.

Table 7.17 shows an example of a ditransitive. The output of the Google has been given in the passive tense. Microsoft's translator gives an incorrect output. UniArab successfully translates the sentence in its entirety. Figure 7.18 shows this sentence output in the UniArab system.

Figure 7.18: Ditransitive 1

Table 7.18: Test : Ditransitive with 2 NP

Arabic	مارك يري آدم الرسالة *mārk yry ādm ālrsālh*
human-translated	Mark is showing Adam the letter.
Google	Mark Adam see the letter.
Microsoft	Marc finds Adam message.
UniArab	Mark is showing Adam the letter.

Table 7.18 shows an example of another ditransitive. The output of the Google translator is faulty in determining the actor, the system can not analyse who does what. Microsoft's translator is faulty in the meaning of the verb and in sentence meaning. UniArab successfully translates the sentence in its entirety. Figure 7.19 shows this sentence output in the UniArab system.

Figure 7.19: Ditransitive with 2 NP

Table 7.19: Test : Ditransitive with preposition

Arabic	عمر أعطى لخالد كتاب *ʿmr ʾaʿṭā lẖālid ktāb*
human-translated	Omar gave a book to Khalid.
Google	Omar Khaled gave a book
Microsoft	Omar gave Khalid book
UniArab	Omar gave a book to Khalid.

Table 7.19 shows an example of ditransitive. The output of the Google translator is faulty in determining the actor, the system can not analyse who does what. Microsoft's translator is faulty loosing the definite article and sentence meaning. UniArab successfully translates the sentence in its entirety. Figure 7.20 shows this sentence output in the UniArab system.

Figure 7.20: Ditransitive with preposition

7.2.8 Limitation of UniArab

Table 7.20: Test : Limitation of UniArab

Arabic	سنسافر غدا إلى مصر *snsāfr ġdā ʔilā mṣr*
human-translated	We will travel to Egypt tomorrow
Google	Tomorrow he travels to Egypt
Microsoft	سنسافر *snsāfr* on to Egypt
UniArab	

Table 7.20 shows another example of a pro–drop sentence. The output of the Google translator is faulty on the point of pro-drop; the system did not find the subject. We found that Microsoft's MT did not recognize the important word in the sentence and passed it through to the output. UniArab fails to give a translation, because this structure does not exist in the system. Since RRG is built upon the logical structure, when an unknown structure is encountered, it cannot produce an output, even if some of the words are in the lexicon. Figure 7.21 shows this sentence output in the UniArab system.

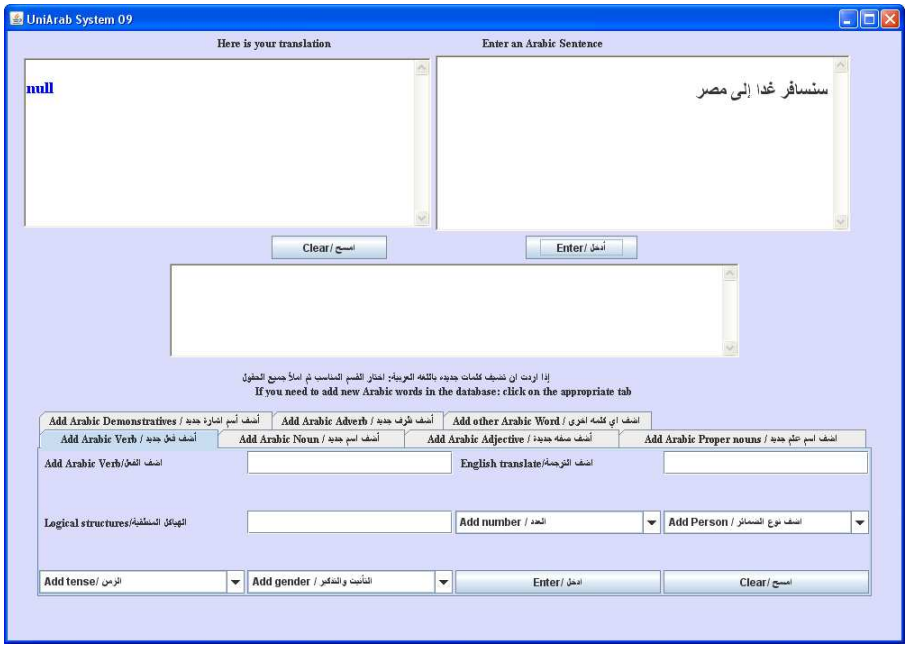

Figure 7.21: Limitation of UniArab 1

In a case where a word is not available in the lexicon, but the logical structure is recognised, UniArab will output a correctly structured translation, but with the unknown Arabic word in its position within the English sentence. This makes the system resilient to slight misspellings which can be recognised and corrected. Figure 7.22 shows this sentence output in the UniArab system.

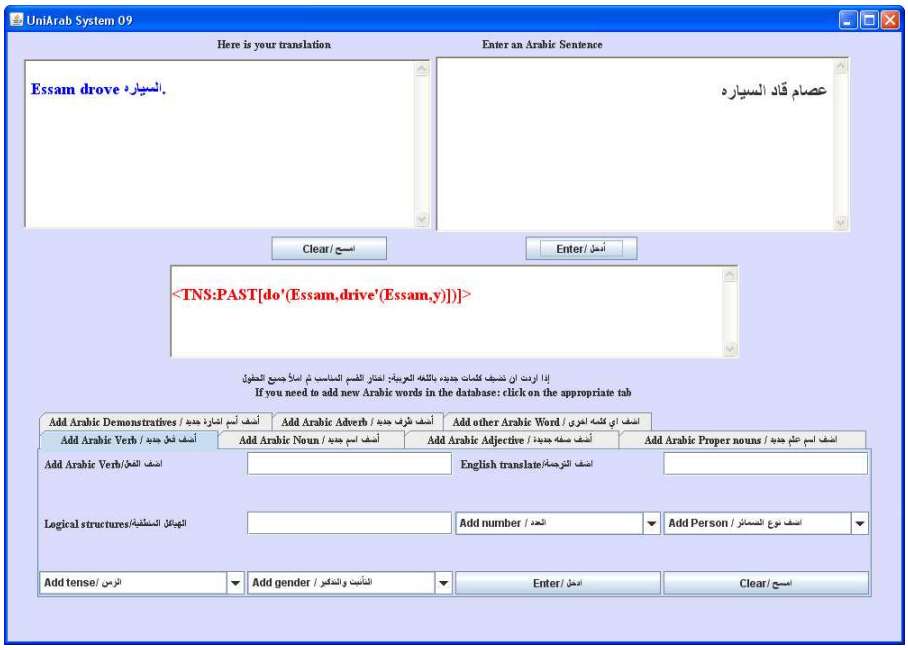

Figure 7.22: Limitation of UniArab 2

Table 7.21: Test : Limitation of UniArab 3 using non existing nonsense word

Arabic	خالد كسر ضصثقفغ *ẖālد ksr ḍṣtqfġ*
human-translated	Khaled broke (ضصثقفغ *ḍṣtqfġ* this word is not an Arabic word).
Google	Khalid break Dsthagafg
Microsoft	Khaled ضصثقفغ *ḍṣtqfġ* break
UniArab	Khaled broke ضصثقفغ *ḍṣtqfġ*

In Table 7.21, we show how the system responds to an unknown word. We have put in a non-word in the Arabic sentence. The output of the Google and Microsoft's translators are faulty in the verb. Microsoft's translator put the unknown word in the wrong position. Google transliterates the word and puts it in the correct position. UniArab successfully translates the verb and puts the unknown word in the correct position. Figure 7.23 shows this sentence output in the UniArab system.

Figure 7.23: Limitation of UniArab 3

7.3 System evaluation

UniArab supports simple sentences with one or two arguments or compounds. We have a number of sentences that were created to aid the grammar in terms of coverage of basic Arabic sentence structures. Further research should be conducted to incorporate more stages into UniArab.

UniArab is based on RRG, and the logical structure of a sentence is the key piece of information for producing a translation. The system is programmed to be capable of dealing with specific structures. Once a structure is enabled within the system, the only limit on translating sentences of that structure is the coverage of the vocabulary. Hence, if a specific sentence structure exists with in UniArab, any sentence of that structure can be translated. This is a strength of being RRG-based, since the structure and vocabulary are dealt with independently, and vocabulary is more straightforward to improve. The structure is also independent of issues of gender and tense, which are only considered once a structure has been assigned, to determine who does what. As we develop UniArab, adding further structures increases the coverage by a considerable amount. However, as the number of structures increases, word ambiguity will become a bigger issue.

UniArab uses an XML-formatted data-source as its lexicon. The key strength is that this data source is open, and can be used under any operating system, and accessed using different tools and languages. The search engine we use to access the data source is able to deal with Arabic words which translate into multiple-word English phrases. For example, أريد *ʿaryd* in Arabic translates to *I want*. However, in its current state, we cannot find single entries that consist of multiple Arabic words. For example شباك الحجز *šbāk ālḥğz* in Arabic translates to *counter* in English; the system cannot deal with this

yet. Another example is عمر يلعب يوم السبت *ʿmr yl*ʿ*b ywm ālsbt* which means *Omar plays on Saturday*. The structure of this sentence works in the system, however, since the two words يوم السبت *ywm ālsbt* translate to a single English word, *Saturday*, and the search engine cannot deal with this now. This also affects idioms and bigrams. We can overcome this issue by modifying the database and search algorithm. For each n-gram phrase, we index it by the first word. When we search the database for this token, we get the single word translation and any n-grams starting with the word. Then we can check the sentence to see if these n-grams are matched.

Another limitation exists because we are not yet dealing with ambiguity. A word like علم *ʿlm* can have many different meanings in Arabic, for example, it can mean flag, taught, knowledge or discovered. At the moment, the Arabic word might exist for different parts of speech. The search extracts all of them, but we only use the first one returned. Once we deal with ambiguity, we will have to analyse the different results, looking at the sentence structures to decide which translation to use.

Since our system is based on RRG, the logical structure of a sentence is the basis of the translation. This was very useful, since it allows the system to deal with issues that can be complicated, like free-word-order, and determining the actor and undergoer. The lexicon used in UniArab can be refined further, and we would like to do this in further research. At the moment, the lexicon contains entries for single Arabic words, which can in some cases translate to clauses in English. For example, قلمي *qlmy* translates to *my pen*. The ي *y* at the end of the Arabic word is the possessive *my* in English. Similarly, بالقلم *bālqlm* translates to *with the pen*, the ب *b* translates to *with* (or *using*), the ال *āl* is the definite article *the*. Finally, بقلمي *bqlmy* translates to *with my pen*. In the future, it makes sense to simplify the lexicon by including only the basic noun, and allowing the

search engine to extract these extra modifiers. Hence, we need only a single entry for the basic noun, rather than an entry for each possible occurrence. This reduces the size of the lexicon, and hence the speed of the search routine.

At the moment, the lexicon is categorised into seven parts of speech. We have designed the GUI so that when adding a specific word to the lexicon, only the related options are presented to the user for that part of speech. This minimises errors when entering data. As our research extends, we may need to modify the categorisation of the lexicon to allow for more complicated word types.

UniArab does not process ambiguous words or complex sentences, so far, in this research. This research focussed first on discovering whether the logical structure of a sentence, based on RRG can be used for translation. Hence, we decided to limit the scope of the project, since this is work in a new area, that has not been investigated before. We fully expect to expand the system to allow it to cope with ambiguity in the future. The system's reliability depends on the data source and fails to handle unknown words. UniArab does not process single words, even if those words are in its lexicon, because UniArab is built on the logical structure of verbs.

In our comparison with other translation systems we have used simplex sentences. While UniArab is limited to simplex sentences and has limited coverage, we believe it is essential to reach high quality translation of these sentences first, in order to be able to expand to high quality translations of more complex sentences. We can see that the existing tools cannot even achieve reasonable translations of simplex sentences, so how can we expect them to give high quality translations of larger text? We have found that small errors in the initial analysis of a sentence can cause huge errors in the final translation, so high quality analysis is very important.

7.4 Summary

In this chapter we subjected the UniArab System to a series of tests in a wide range of sentence categories. For each test we compared the results obtained through UniArab to those obtained when using translation engines from Google and Microsoft. We also presented a human-translated equivalent to each. In contrast, the Google and Microsoft translators gave mixed results. In many cases, sentence meaning was lacking, and even some basic constructs could not be translated. Perhaps this is due to their focus on translating long sentences and paragraphs. We highlighted this by comparing them to UniArab for longer compound sentences and found that they did indeed convey more of the meaning. These results suggest that RRG is a promising candidate for Arabic to English Machine translation, and as the grammar is developed, the system should begin to cope with more complicated sentences. For simplex sentences (intransitive, transitive and ditransitive) it clearly outperforms existing systems.

Now is not the end.

It is not the beginning of the end.

It is perhaps, the end of the beginning.

Winston Churchill

8

Conclusion

In this thesis we have presented an Arabic to English machine translation system called UniArab, which is based on the Role and Reference Grammar model. We detailed the design of the system and how it was built to accommodate specifics of the Arabic language and the generation of English translations.

We started with the goal of designing a machine translation system that could show whether we can extract logical structure from Arabic sentences using RRG, and use this to produce high quality translations into English. We believe the results shown in Chapter 7 show that our system has proved this, and that our method is more robust for these cases, than other MT systems. There are still a number of areas which need to be developed for UniArab to achieve more coverage, and we believe that we can build on the work we have done so far.

Since the logical structure is separate from the vocabulary, when we focus on giving

the system capability to deal with a large number of structure variations, it becomes significantly more powerful since each structure represents all possible sentences of that structure regardless of the specific words included. This is a significant point, since vocabulary is easy to develop, but structure requires much more effort.

The major challenge we faced was to use RRG within a machine translation system. UniArab is the first MT system that uses RRG.In the Arabic linguistic tradition there is not a clear-cut, well-defined analysis of the inventory of parts of speech in Arabic. We found that the existent classifications were not suitable, and so we had to create a classification that made sense for RRG-based translation. We were able to extract logical structures that made sense from natural Arabic. And we were also able to generate English translations from this logical structure.

Some specific challenges included dealing with the absence of the copula verb, 'to be', in Arabic. To solve this, we had to look at some Arabic sentence which do not contain verbs, and correctly deduce how to extract the copula. Free word order was another challenge due to its widespread presence in Arabic, and this was solved by detailed analysis of the source language and incorporating this in the logical structure.

We have discoverd that RRG is a realistic basis for machine translation systems. The use of a sentence's logical structure to create translations is robust and gives high quality translations which can deal with some of the challenges of languages like Arabic.

Our work has contributed the first machine translation system based on RRG, which we have used to prove its effectiveness for MT. This was a major challenge as we had little work to refer to. We have also advanced work on Arabic language classification,

and so we hope our work will be the beginning of more work in this arena. We believe this serves as an excellent foundation for further research in the area.

While statistical machine translation has been promoted by many, we believe that langauges, expecially rich languages like Arabic, are very organised and structural, and such approaches cannot correctly deal with the wide variety of sentence structures. When these systems cannot deal with simplex sentences, as we have shown in Chapter 7, how do we expect them to correctly translate whole paragraphs? In our approach, we expect that high quality translations of simplex sentences are the only basis which can build to good translations of whole paragraphs. The results we have presented are the first step in applying RRG to sophisticated translation. By focussing in this initial stage on the basics, we build a more solid foundation for the next stage.

8.1 Thesis summary

In Chapter 2, we presented a summary overview of the grammatical structure of the Arabic language. We detailed various sentence structures as well as unique word attributes like gender rules applied to all words and duality in number. We discussed how some of these properties could be used to extract information about sentence structure.

In Chapter 3, we presented the Role and Reference Grammar model, and showed how it could be used to deduce the logical structure of sentences and produce a lexical representation which could be used as the interlingua.

In Chapter 4, we presented various approaches to machine translation. We compared direct translation, transfer systems and interlingua systems and showed how interlingua systems require significantly more effort in the analysis and generation stages, but have

a distinct advantage in the simplicity of the translation process. Furthermore, they are more flexible in terms of adding extra languages. We also talked about the challenges of machine translation, with a specific focus on those specific to the Arabic language.

In Chapter 5, we presented a high-level view of the system framework and defined our evaluation criteria for measuring system performance.

In Chapter 6, we detailed the technical aspects of UniArab, covering all the phases involved in the machine translation process. We described the lexical system that underlies UniArab, detailing the attribute information held for each type of word. We discussed the generation phase and how the system maps the logical structure to a target English sentence. We then briefly discussed the user interface, and some of the technical challenges encountered during the implementation.

In Chapter 7, we presented the results of our evaluation of UniArab for a wide variety of sentence types. We compared its results to those of the Google and Microsoft translators as well as human translation. We found that it significantly outperforms the other automated translation systems, matching human translation. We discussed its limits in regards to complex sentence structures.

8.2 Summary of thesis contributions

This thesis contributions are summarised as follows:

- A detailed presentation of the structure of Arabic sentences and a discussion of the language's unique features.
- A detailed system framework for implementing RRG machine translation for Arabic

and proving the suitability of the model.

- A detailed technical implementation of an Arabic to English machine translator based on the RRG model, including user interface and a custom designed, extensible data source.
- An evaluation of the translation system and comparison to existing commercial systems.
- Specifying verb 'to be', free word order, pro-drop and transitivity of verbs.

8.3 Future work

Given the scope of this Masters research project, there are a number of areas where this work could be extended. Firstly, the question of ambiguity is very interesting. We feel that RRG is suited to overcoming word ambiguity by using sentence structure, and would like to explore this. We would also like to incorporate more compound structures allowing UniArab to deal with more complex sentences. We would also like to explore the auto generation of lexicon information from Arabic source verbs as a way to quickly populate the lexical source.

The main topic of investigation is the development of a framework for translating Arabic to English based on RRG. The framework is designed to demonstrate the capabilities of RRG as a base for machine translation of Arabic into English using an interlingua bridge strategy. This thesis showed that RRG facilitates the translation process from a specific language to other languages. Future research should focus on:

(1) Enhancing and extending the UniArab system to support more natural Arabic sentences, and word ambiguity, in particular:

- To understand, process, and translate complex predicates and multi-clause sentences in coordinate, subordinate and cosubordination structures.
- To understand, process and translate voice and valence increasing/decreasing operations in the machine translation of Arabic.
- To design a lexicon architecture to support the morphological templates for Arabic words into their respective consonantal and vowel components with the appropriate word formation rules implemented in software.
- To extend the underlying theory of RRG to encompass more fully the lexicon, syntax and morphology of Arabic.

(2) Evaluating UniArab with respect to other systems based on non-RRG methods.

References

Abn-Aqeal, 2007. *sharah Abn-Aqeal ala'lfat Abn-Malek.* Dar Al'alem llmlaien.

Al-Sughaiyer, I. A. and Al-Kharashi, I. A., 2004. Arabic morphological analysis techniques: A comprehensive survey. *JASIST*, **55**(3):189–213.

Alosh, M., 2005. *Using Arabic: A Guide to Contemporary Usage*. Cambridge University Press.

Arciniegas, F., 2000. *XML Developer's Guide*. McGraw-Hill.

Attia, M., 2004. Report on the introduction of Arabic to ParGram. In *Proceedings of ParGram Fall Meeting, Dublin, Ireland*.

Attia, M. A., 2008. *Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation*. PhD thesis, University of Manchester.

Bateson, M. C., 2003. *Arabic Language Handbook.* Georgetown University Press.

Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., and Yergeau, F., 2008. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*.

Brown, P., Pietra, S. D., Pietra, V. D., and Mer-cer., R., 1993. The mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics*, volume 19 i2, pages 263–311.

Google, 2009. Google translator. http://translate.google.com.

Holes, C., 2004. *Modern Arabic: Structures, Functions, and Varieties*. Georgetown University Press.

Hutchins, J., 2003. *Machine Translation: General Overview*. in : book, Oxford University Press.

Hutchins, J. and Somers, H. L., 1992. *An introduction to machine translation*. Academic Press.

ibn Abd Allah Ibn Malik, M., 1984. *Alfiyat Ibn Malik*. Maktabat al-Adab.

Intelligence, A. B., 2004. Abi research. http://www.abiresearch.com/abiprdisplay2.jsp?pressid=116.

Izwaini, S., 2006. Problems of Arabic machine translation: evaluation of three systems. In *The British Computer Society (BSC), London* ., pages 118–148.

Khan, M. A. S., 2007. *Arabic Tutor - Volume One*, volume one. Madrasah In'aamiyyah.

Laoudi, J., Tate, C., and Voss, C., 2004. Towards an automated evaluation of an embedded mt system. In *roceedings of the European Association for Machine Translation Workshop, Malta*.

Microsoft, 2009. Microsoft translator. http://www.windowslivetranslator.com/Default.aspx.

Nolan, B. and Salem, Y., 2009. UniArab: An RRG Arabic-to-English Machine Translation Software. In *Proceedings of The 2009 International Conference on Role and Reference Grammar, University of California, Berkeley, USA*.

Nunes, M., 1993. *Argument Linking in English Derived Nominals In Van Valin (ed) Advances in Role and Reference Grammar*. John Benjamins.

Oren, T., 2004. Machine translation and the global blogosphere. http://www.windsofchange.net/archives/006011.html. website.

Owens, J., 2006. *A Linguistic History of Arabic*. Oxford University Press.

Ramsay, A. and Mansour, H., 2006. Local constraints on Arabic word order. In *5th International Conference on NLP, FinTAL 2006*, pages 447–457.

Ryding, K. C., 2007. *A Reference Grammar of Modern Standard Arabic*. Cambridge University Press, 3rd edition.

Salem, Y., Hensman, A., and Nolan, B., 2008a. Implementing Arabic-to-English machine translation using the Role and Reference Grammar linguistic model. In *Proceedings of the Eighth Annual International Conference on Information Technology and Telecommunication (IT&T 2008), Galway, Ireland*.

Salem, Y., Hensman, A., and Nolan, B., 2008b. Towards Arabic to English machine translation. *ITB Journal*, **17**:20–31.

Salem, Y. and Nolan, B., 2009a. An Arabic-to-English machine translation system using an XML-based Role and Reference Grammar representation. In *Proceedings of the 23rd Annual Symposium on Arabic Linguistics, University of Wisconsin-Milwaukee*.

Salem, Y. and Nolan, B., 2009b. Designing an XML lexicon architecture for Arabic machine translation based on Role and Reference Grammar. In *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR 2009), Cairo, Egypt*.

Salem, Y. and Nolan, B., 2009c. UNIARAB: An universal machine translator system for Arabic Based on Role and Reference Grammar. In *Proceedings of the 31st Annual Meeting of the Linguistics Association of Germany (DGfS 2009)*.

Schulz, E., 2005. *A Student Grammar of Modern Standard Arabic*. Cambridge University Press.

Trujillo, A., 1999. *Translation Engines: Techniques for Machine Translation*. Springer.

Turian, J. P., Shen, L., and Melamed, I. D., 2003. Evaluation of machine translation and its evaluation. In *Proceedings of the MT Summit IX, New Orleans, USA*, pages 386–393.

Van Valin, R., 1993. *Advances in Role and Reference Grammar*. John Benjamins.

Van Valin, R., 2007. The Role and Reference Grammar analysis of three place predicates. *CEEOL Contemporary Linguistics*, **63**:31–63.

Van Valin, R. and LaPolla, R., 1997. *Syntax: Structure, Meaning, and Function*. Cambridge University Press.

Versteegh, K., 2001. *The Arabic Language*. Edinburgh University Press; New Ed edition.

Appendix

A

The author's publications related to this research

- Brian Nolan and Yasser Salem. 2009. "UniArab: An RRG Arabic-to-English Machine Translation Software", in *Proceedings of The 2009 International Conference on Role and Reference Grammar, University of California, Berkeley, USA*, August 2009.

- Yasser Salem and Brian Nolan. 2009. "Designing an XML Lexicon Architecture for Arabic Machine Translation Based on Role and Reference Grammar", in *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR 2009), Cairo, Egypt*, April 2009.

- Yasser Salem and Brian Nolan, 2009. "An Arabic-to-English Machine translation system using an XMLbased Role and Reference Grammar representation", in *Proceedings of the 23rd Annual Symposium on Arabic Linguistics, University of Wisconsin-Milwaukee, USA*, April 2009.

- Yasser Salem and Brian Nolan. 2009. " UNIARAB: An Universal Machine Translator System For Arabic Based On Role And Reference Grammar", in *Proceedings of the 31st Annual Meeting of the Linguistics Association of Germany (DGfS 2009), University of Osnabruck, Germany*, March 2009.

- Yasser Salem, Arnold Hensman and Brian Nolan. 2008. Implementing Arabic-to-English Machine Translation using the Role and Reference Grammar Linguistic Model, *in Proceedings of the Eighth Annual International Conference on Information Technology and Telecommunication (ITT 2008), Galway, Ireland, October 2008.* (Runner-up for Best Paper Award)

- Yasser Salem, Arnold Hensman and Brian Nolan. 2008. "Towards Arabic to English Machine Translation", *ITB Journal, May 2008, Issue No. 17: 20-31.*

B

Buckwalter Arabic transliteration

	Arabic Letter and Phonetic Value	Letter Name	Unicode
1	ا *ā*	ALEF	u0627
2	ب *b*	BEH	u0628
3	ت *t*	TEH	u062A
4	ث *ṯ*	THEH	u062B
5	ج *ǧ*	JEEM	u062C
6	ح *ḥ*	HAH	u062D
7	خ *ḫ*	KHAH	u062E
8	د *d*	DAL	u062F
9	ذ *ḏ*	THAL	u0630
10	ر *r*	REH	u0631
11	ز *z*	ZAIN	u0632
12	س *s*	SEEN	u0633
13	ش *š*	SHEEN	u0634

	Arabic Letter and Phonetic Value	Letter Name	Unicode
14	ص *ṣ*	SAD	u0635
15	ض *ḍ*	DAD	u0636
16	ط *ṭ*	TAH	u0637
17	ظ *ẓ*	ZAH	u0638
18	ع ʿ	AIN	u0639
19	غ *ġ*	GHAIN	u063A
20	ف *f*	FEH	u0641
21	ق *q*	QAF	u0642
22	ك *k*	KAF	u0643
23	ل *l*	LAM	u0644
24	م *m*	MEEM	u0645
25	ن *n*	NOON	u0646
26	هـ *h*	HEH	ŭ647
27	و *w*	WAW	u0648
28	ي *y*	YEH	u064A

	Arabic Letter and Phonetic Value	Letter Name	Unicode
29	ء ʾ	HAMZA	u0621
30	إِ ʾi	ALEF WITH HAMZA UNDER	u0625
31	أَ ʾa	ALEF WITH HAMZA ABOVE	u0623
32	آ ʾā	ALEF WITH MADDA ABOVE	u0622
33	ى ā	YEH	u0649
34	ئ ʾy	YEH WITH HAMZA ABOVE	u0626
35	ؤ ʾw	WAW WITH HAMZA ABOVE	u0624
36	ة h	TEH MARBUTA	u0629
37	اَ a	FATHA	u064E
38	اُ u	DAMMA	u064F
39	اِ i	KASRA	u0650
40	اً an	TANWIN ALFATH	u064B
41	اٌ un	TANWIN ALDAM	u064C
42	اٍ in	TANWIN ALKASER	u064D
43	اْ	SKOON	u0652
44	اّ	SHADDA	u0651
45	؟ ?	ARABIC QUESTION MARK	u061F

C

List of translatable sentences

I want a ring.	أريد خاتم *ʾaryd ẖātm*
I forgot my wallet.	نسيت محفظتي *nsyt mḥfẓty*
I missed the plane.	فاتتني الطائرة *fātny ālṭāʾyrh*
I want a room.	أريد غرفة *ʾaryd ġrfh*
I am a tourist.	أنا سائح *ʾanā sāʾyḥ*
I am alone.	انا وحدي *ānā wḥdy*
I am Irish.	أنا ايرلندي *ʾanā āyrlndy*
we are students.	نحن تلاميذ *nḥn tlāmyd*
he is an engineer.	هو مهندس *hw mhnds*

I am an engineer.	انا مهندس *ānā mhnds*
I am the engineer.	أنا المهندس *ʾanā ālmhnds*
Sarah hurts Yousuf.	تجرح **ساره** يوسف *tǧrḥ sārh ywsf*
Sarah hurts Yousuf.	**ساره** تجرح يوسف *sārh tǧrḥ ywsf*
Sarah hurts Yousuf.	تجرح يوسف **ساره** *tǧrḥ ywsf sārh*
Omar will drink the milk.	سيشرب اللبن عمر *syšrb āllbn ʿmr*
Omar will drink the milk.	سيشرب عمر اللبن *syšrb ʿmr āllbn*
Khalid is drinking the milk.	يشرب خالد اللبن *yšrb ẖālد āllbn*
Khalid drank the milk.	شرب خالد اللبن *šrb ẖālد āllbn*
Omar is visiting Ireland.	يزور عمر أيرلندا *yzwr ʿmr ʾayrlndā*
Qays loves Laila.	قيس يحب ليلى *qys yḥb lylā*
Qays loves Laila.	يحب قيس ليلى *yḥb qys lylā*
Qays loves Laila.	يحب ليلى قيس *yḥb lylā qys*
Laila loves Qays.	ليلى تحب قيس *lylā tḥb qys*
Laila loves Qays.	تحب ليلى قيس *tḥb lylā qys*
Laila loves Qays.	تحب قيس ليلى *tḥb qys lylā*
Omar read the book.	قرأ عمر الكتاب *qrʾa ʿmr ālktāb*
Brian read the book.	براين قرأ الكتاب *brāyn qrʾa ālktāb*

she is an engineer.	هي مهندسة *hy mhndsh*
Zaid loves Fatima.	زيد يحب فاطمة *zyd yḥb fāṭmh*
Zaid loves Fatima.	يحب زيد فاطمة *yḥb zyd fāṭmh*
Zaid loves Fatima.	يحب فاطمة زيد *yḥb fāṭmh zyd*
Fatima loves Zaid.	فاطمة تحب زيد *fāṭmh tḥb zyd*
Fatima loves Zaid.	تحب فاطمة زيد *tḥb fāṭmh zyd*
Fatima loves Zaid.	تحب زيد فاطمة *tḥb zyd fāṭmh*
Eman drew her school.	رسمت إيمان مدرستها *rsmt ʾiymān mdrsthā*
Louis hit Mark.	ضرب لويس مارك *ḍrb lwys mārk*
Louis hit Mark.	لويس ضرب مارك *lwys ḍrb mārk*
Mark hit Louis.	مارك ضرب لويس *mārk ḍrb lwys*
Mark hit Louis.	ضرب مارك لويس *ḍrb mārk lwys*
Brian wrote the book.	براين كتب الكتاب *brāyn ktb ālktāb*
Ayesha wrote the book.	عائشة كتبت الكتاب *ʿāʾyšh ktbt ālktāb*
Eman wrote the book.	كتبت إيمان الكتاب *ktbt ʾiymān ālktāb*
I have made a reservation.	لقد قمت بالحجز *lqd qmt bālḥğz*
I have lost my ticket.	لقد فقدت تذكرتي *lqd fqdt tdkrty*
I am a doctor.	أنا طبيب *ʾanā ṭbyb*

Abbas is playing with the ball.	يلعب عباس بالكرة *ylʿb ʿbās bālkrh*
Abbas is playing with the ball.	عباس يلعب بالكرة *ʿbās ylʿb bālkrh*
Abbas is playing with the ball.	بالكرة يلعب عباس *bālkrh ylʿb ʿbās*
Yousuf played with the ball.	لعب يوسف بالكرة *lʿb ywsf bālkrh*
Yousuf played with the ball.	لعب بالكرة يوسف *lʿb bālkrh ywsf*
Yousuf played with the ball.	بالكرة لعب يوسف *bālkrh lʿb ywsf*
Yousuf will play with the ball.	سيلعب يوسف بالكرة *sylʿb ywsf bālkrh*
Yousuf will play with the ball.	يوسف سيلعب بالكرة *ywsf sylʿb bālkrh*
Yousuf will play with the ball.	بالكرة سيلعب يوسف *bālkrh sylʿb ywsf*
Essam played with the spoon.	لعب عصام بالملعقة *lʿb ʿṣām bālmlʿqh*
Essam will play with the spoon.	سيلعب عصام بالملعقة *sylʿb ʿṣām bālmlʿqh*
Essam is playing with the spoon.	يلعب عصام بالملعقة *ylʿb ʿṣām bālmlʿqh*
Essam played with the spoons.	لعب عصام بالملاعق *lʿb ʿṣām bālmlāʿq*
Essam will play with the spoons.	سيلعب عصام بالملاعق *sylʿb ʿṣām bālmlāʿq*
Essam is playing with the spoons.	يلعب عصام بالملاعق *ylʿb ʿṣām bālmlāʿq*
Mansour ate with the spoon.	أكل منصور بالملعقة *ʾakl mnṣwr bālmlʿqh*
Mansour ate with the spoon.	منصور أكل بالملعقة *mnṣwr ʾakl bālmlʿqh*
Mansour ate with the spoon.	بالملعقة أكل منصور *bālmlʿqh ʾakl mnṣwr*

Jack is eating with the spoon.	يأكل جاك بالمعقة *yʿakl ǧāk bālmlʿqh*
Jack will eat with the spoon.	سيأكل جاك بالمعقة *syʿakl ǧāk bālmlʿqh*
Jack killed Mary.	قتل جاك ماري *qtl ǧāk māry*
Jack killed Mary.	جاك قتل ماري *ǧāk qtl māry*
Mary killed Jack.	قتلت ماري جاك *qtlt māry ǧāk*
Mary killed Jack.	ماري قتلت جاك *māry qtlt ǧāk*
Jack killed the man.	قتل جاك الرجل *qtl ǧāk ālrǧl*
Jack killed the man.	جاك قتل الرجل *ǧāk qtl ālrǧl*
The man killed Jack.	الرجل قتل جاك *ālrǧl qtl ǧāk*
The man killed Jack.	قتل الرجل جاك *qtl ālrǧl ǧāk*
Suhaib bellowed the fire.	صهيب نفخ النار *ṣhyb nfḫ ālnār*
Suhaib bellowed the fire.	نفخ النار صهيب *nfḫ ālnār ṣhyb*
Suhaib bellowed the fire.	نفخ صهيب النار *nfḫ ṣhyb ālnār*
Sulaiman opened the door.	فتح الباب سليمان *fṭḥ ālbāb slymān*
Sulaiman opened the door.	سليمان فتح الباب *slymān fṭḥ ālbāb*
Sulaiman opened the door.	فتح سليمان الباب *fṭḥ slymān ālbāb*
Zaid took the book.	أخذ زيد الكتاب *ʿaḫd zyd ālktāb*

Zaid took the book.	زيد أخذ الكتاب *zyd ʾaḫḏ ālktāb*
Qays took the files.	أخذ قيس الملفات *ʾaḫḏ qys ālmlfāt*
Qays took the files.	أخذ قيس الملفات *ʾaḫḏ qys ālmlfāt*
Brian rides the bus.	براين يركب الحافلة *brāyn yrkb ālḥāflh*
Brian rides the bus.	يركب براين الحافلة *yrkb brāyn ālḥāflh*
Fahmy rides the car.	فهمي يركب السيارة *fhmy yrkb ālsyārh*
Fahmy rides the car.	يركب فهمي السيارة *yrkb fhmy ālsyārh*
Fahmy rides the car.	يركب السيارة فهمي *yrkb ālsyārh fhmy*
Khalid answered the question.	خالد أجاب السؤال *ḫālḍ ʾağāb ālsʾwāl*
Khalid answered the question.	أجاب السؤال خالد *ʾağāb ālsʾwāl ḫālḍ*
Khalid answered the question.	أجاب خالد السؤال *ʾağāb ḫālḍ ālsʾwāl*
Rashid broke the window.	راشد كسر النافذة *rāšd ksr ālnāfḏh*
Rashid broke the window.	كسر راشد النافذة *ksr rāšd ālnāfḏh*
Rashid broke the window.	كسر النافذة راشد *ksr ālnāfḏh rāšd*
Khalid broke his toy.	كسر خالد لعبته *ksr ḫālḍ lʿbth*
Omar tore the book.	مزق عمر الكتاب *mzq ʿmr ālktāb*
Omar tore the book.	عمر مزق الكتاب *ʿmr mzq ālktāb*
Omar tore the book.	مزق الكتاب عمر *mzq ālktāb ʿmr*

Ayah tore the page.	آيه مزقت الكيس *ʾāyh mzqt ālkys*
Ayah tore the page.	مزقت الكيس آيه *mzqt ālkys ʾāyh*
Ayah tore the page.	مزقت آيه الكيس *mzqt ʾāyh ālkys*
Omar opened his present.	فتح عمر هديته *fṭḥ ʿmr hdyth*
Omar opened the window.	فتح عمر النافذة *fṭḥ ʿmr ālnāfdh*
Omar opened the door.	عمر فتح الباب *ʿmr fṭḥ ālbāb*
James combed his hair.	مشط جيمس شعره *mšṭ ǧyms šʿrh*
James combed his hair.	جيمس مشط شعره *ǧyms mšṭ šʿrh*
Eric cleaned the window.	نظف إيريك النافذة *nẓf ʾiyryk ālnāfdh*
Eric cleaned the plane.	إيريك نظف الطائرة *ʾiyryk nẓf ālṭāʾyrh*
Eric cleaned the house.	نظف المنزل إيريك *nẓf ālmnzl ʾiyryk*
Sarah wiped the table.	مسحت ساره الطاوله *msḥt sārh ālṭāwlh*
Sarah wiped the table.	ساره مسحت الطاوله *sārh msḥt ālṭāwlh*
Roqiah will cook the dinner.	ستطبخ رقية العشاء *stṭbḫ rqyh ālʿšāʾ*
Roqiah will cook the dinner.	رقية ستطبخ العشاء *rqyh stṭbḫ ālʿšāʾ*
Roqiah will cook the dinner.	ستطبخ العشاء رقية *stṭbḫ ālʿšāʾ rqyh*
Harold pinched James.	قرص هارولد جيمس *qrṣ hārwld ǧyms*
Harold pinched James.	هارولد قرص جيمس *hārwld qrṣ ǧyms*

he is a doctor.	هو طبيب *hw ṭbyb*
Ayah is spending her money.	آيه تنفق نقودها *ʾāyh tnfq nqwdhā*
Ayah is spending her money.	تنفق آيه نقودها *tnfq ʾāyh nqwdhā*
Henry lost his money.	هنري فقد نقوده *hnry fqd nqwdh*
Henry lost his money.	فقد هنري نقوده *fqd hnry nqwdh*
Adam punched Philip.	لكم آدم فيليب *lkm ʾādm fylyb*
Zakiah killed Sarah.	قتلت زكيه ساره *qtlt zkyh sārh*
Zakiah killed Sarah.	زكيه قتلت ساره *zkyh qtlt sārh*
Mark slapped Louis.	صفع مارك لويس *ṣfʿ mārk lwys*
Mark slapped Louis.	مارك صفع لويس *mārk ṣfʿ lwys*
Sarah hates Zakiah.	تكره ساره زكيه *tkrh sārh zkyh*
Sarah hates Zakiah.	ساره تكره زكيه *sārh tkrh zkyh*
Ayesha phoned Eman.	هاتفت عائشة إيمان *hātft ʿāʾyšh ʾiymān*
Ayesha phoned Eman.	عائشة هاتفت إيمان *ʿāʾyšh hātft ʾiymān*
Ayah thanked Khalid.	شكرت آيه خالد *škrt ʾāyh ẖāld*
Ayah thanked Khalid.	آيه شكرت خالد *ʾāyh škrt ẖāld*
Sarah called Adam.	نادت ساره آدم *nādt sārh ʾādm*
Sarah called Adam.	ساره نادت آدم *sārh nādt ʾādm*

Eman saw Sarah.	رأت إيمان ساره *rʾat ʾiymān sārh*
Eman saw Carl.	إيمان رأت كارل *ʾiymān rʾat kārl*
Philip caught the ball.	مسك فيليب الكرة *msk fylyb ālkrh*
Philip caught the ball.	فيليب مسك الكرة *fylyb msk ālkrh*
Philip caught the ball.	مسك الكرة فيليب *msk ālkrh fylyb*
Carl bought pens.	إشترى كارل أقلام *ʾištrā kārl ʾaqlām*
Carl bought pens.	كارل إشترى أقلام *kārl ʾištrā ʾaqlām*
Mark drove the plane.	قاد مارك الطائرة *qād mārk ālṭāʾyrh*
Mark drove the bus.	مارك قاد الحافلة *mārk qād ālḥāflh*
Mark drove the car.	قاد السيارة مارك *qād ālsyārh mārk*
Adam is cleaning his toy.	آدم ينظف لعبته *ʾādm ynẓf lʿbth*
Adam is cleaning his room.	ينظف آدم غرفته *ynẓf ʾādm ġrfth*
Adam is cleaning his car.	ينظف آدم سيارته *ynẓf ʾādm syārth*
Mark will clean my kitchen.	مارك سينظف مطبخي *mārk synẓf mṭbẖy*
Sarah will clean my office.	ستنظف ساره مكتبي *stnẓf sārh mktby*
Sarah will clean the car.	ساره ستنظف السيارة *sārh stnẓf ālsyārh*
Eman will clean her room.	ستنظف إيمان غرفتها *stnẓf ʾiymān ġrfthā*
Eman will clean her room.	إيمان ستنظف غرفتها *ʾiymān stnẓf ġrfthā*

Louis is washing the dishes.	يغسل لويس الصحون *yġsl lwys ālṣḥwn*
Louis is washing the dishes.	لويس يغسل الصحون *lwys yġsl ālṣḥwn*
Louis is washing the dishes.	يغسل الصحون لويس *yġsl ālṣḥwn lwys*
Harold is feeding his cat.	يطعم هارولد قطته *yṭʿm hārwld qṭth*
Harold is feeding his cat.	هارولد يطعم قطته *hārwld yṭʿm qṭth*
he is a seller.	هو بائع *hw bāʾyʿ*
Eric is fixing his car.	يصلح إيريك سيارته *yṣlḥ ʾiyryk syārth*
Eric is fixing his car.	إيريك يصلح سيارته *ʾiyryk yṣlḥ syārth*
Fahmy speaks English.	فهمي يتكلم الانكليزية *fhmy ytklm ālānklyzyh*
Fahmy speaks English.	يتكلم فهمي الانكليزية *ytklm fhmy ālānklyzyh*
Ayah is cooking the food.	تطبخ آيه الطعام *tṭbḫ ʾāyh ālṭʿām*
Ayah is cooking the dinner.	آيه تطبخ العشاء *ʾāyh tṭbḫ ālʿšāʾ*
Rashid is helping Mark.	يساعد راشد مارك *ysāʿd rāšd mārk*
Rashid is helping Ayesha.	راشد يساعد عائشة *rāšd ysāʿd ʿāyšh*
Mansour is eating his food.	يأكل منصور طعامه *yʾakl mnṣwr ṭʿāmh*
Mansour is eating his food.	منصور يأكل طعامه *mnṣwr yʾakl ṭʿāmh*

Carl is brushing his hair.	يمشط كارل شعره *ymšṭ kārl šʿrh*
Carl is brushing his hair.	كارل يمشط شعره *kārl ymšṭ šʿrh*
Abbas is brushing his teeth.	يفرش عباس أسنانه *yfrš ʿbās ʾasnānh*
Abbas is brushing his teeth.	عباس يفرش أسنانه *ʿbās yfrš ʾasnānh*
Yousuf is wearing his clothes.	يلبس يوسف ثيابه *ylbs ywsf ṯyābh*
Yousuf is wearing his shoes.	يوسف يلبس حذائه *ywsf ylbs ḥdāʾyh*
Henry is watching the television.	يشاهد هنري التلفاز *yšāhd hnry āltlfāz*
Henry is watching the television.	هنري يشاهد التلفاز *hnry yšāhd āltlfāz*
Henry is watching the television.	يشاهد التلفاز هنري *yšāhd āltlfāz hnry*
Sulaiman caught the fish.	إصطاد سليمان السمك *ʾiṣṭād slymān ālsmk*
Sulaiman caught the fish.	سليمان إصطاد السمك *slymān ʾiṣṭād ālsmk*
Sulaiman caught the fish.	إصطاد السمك سليمان *ʾiṣṭād ālsmk slymān*
Omar is planting the trees.	يزرع الأشجار عمر *yzrʿ ālʾašǧār ʿmr*
Omar is planting the trees.	عمر يزرع الأشجار *ʿmr yzrʿ ālʾašǧār*
James pushed the chairs.	جر جيمس الكراسي *ǧr ǧyms ālkrāsy*
James pushed the chairs.	جيمس جر الكراسي *ǧyms ǧr ālkrāsy*
James pushed the chairs.	جر الكراسي جيمس *ǧr ālkrāsy ǧyms*

Roqih drew trees.	رسمت رقية أشجار *rsmt rqyh ʾašǧār*
Roqih drew Omar.	رقية رسمت عمر *rqyh rsmt ʿmr*
Roqih drew Khalid.	رسمت خالد رقية *rsmt ẖālḍ rqyh*
Ayesha picked the flowers.	عائشة قطفت الزهور *ʿāʾyšh qṭft ālzhwr*
Ayesha picked the flowers.	قطفت عائشة الزهور *qṭft ʿāʾyšh ālzhwr*
Ayesha picked the flowers.	قطفت الزهور عائشة *qṭft ālzhwr ʿāʾyšh*
Omar is fixing the computer.	يصلح عمر الحاسوب *yṣlḥ ʿmr ālḥāswb*
Omar is fixing the computer.	عمر يصلح الحاسوب *ʿmr yṣlḥ ālḥāswb*
Omar is fixing the computer.	يصلح الحاسوب عمر *yṣlḥ ālḥāswb ʿmr*
Omar bought the toys.	إشترى عمر اللعب *ʾištrā ʿmr āllʿb*
Omar bought the fish.	إشترى السمك عمر *ʾištrā ālsmk ʿmr*
Eman ironed the clothes.	كوت إيمان الملابس *kwt ʾiymān ālmlābs*
Eman ironed the clothes.	إيمان كوت الملابس *ʾiymān kwt ālmlābs*
Eman ironed the clothes.	كوت الملابس إيمان *kwt ālmlābs ʾiymān*
Ayah painted the picture.	لونت آيه الصورة *lwnt ʾāyh ālṣwrh*
Ayah painted the picture.	آيه لونت الصورة *ʾāyh lwnt ālṣwrh*
Ayah painted the picture.	لونت الصورة آيه *lwnt ālṣwrh ʾāyh*
I want the book.	أريد الكتاب *ʾaryd ālktāb*

I want a book.	أريد كتاب *ʾaryd ktāb*
I want the food.	أريد الطعام *ʾaryd āl-ṭʿām*
I ate the dinner.	أكلت العشاء *ʾaklt āl-šāʾ*
I ate the food.	أكلت الطعام *ʾaklt āl-ṭʿām*
I ate the fish.	أكلت السمك *ʾaklt āl-smk*
I drank the milk.	شربت اللبن *šrbt āl-lbn*
Eman lost her cat.	فقدت إيمان قطتها *fqdt ʾiymān qṭ-thā*
Eman ran over her cat.	دهست إيمان قطتها *dhst ʾiymān qṭ-thā*
Omar won the race.	فاز عمر بالسباق *fāz ʿmr bāl-sbāq*
Omar is sleeping.	ينام عمر *ynām ʿmr*
The children are crying.	الأطفال يبكون *āl-ʾaṭfāl ybkwn*
the wheel squeaks.	الدولاب يصرصر *āl-dwlāb yṣ-rṣr*
Omar reads.	عمر يقرأ *ʿmr yqr-ʾa*
Omar reads a lot.	عمر يقرأ كثيرا *ʿmr yqr-ʾa kṯyrā*
He hit Khalid.	هو ضرب خالد *hw ḍ-rb ẖ-āld*
He played with the spoon.	هو لعب بالملعقة *hw l-ʿb bāl-ml-ʿqh*
He loves Laila.	هو يحب ليلى *hw yḥb lylā*
He loves Laila.	يحب هو ليلى *yḥb hw lylā*
He loves Laila.	يحب ليلى هو *yḥb lylā hw*

Omar gave Khalid the book.	عمر أعطى خالد الكتاب *ʿmr ʾaṭā ẖālد ālktāb*
Omar gave Sarah the book.	عمر أعطى ساره الكتاب *ʿmr ʾaṭā sārh ālktāb*
Omar is giving Khalid the book.	عمر يعطي خالد الكتاب *ʿmr yṭy ẖālد ālktāb*
Omar is giving Sarah the book.	عمر يعطي ساره الكتاب *ʿmr yṭy sārh ālktāb*
Sarah is giving Khalid the book.	ساره تعطي خالد الكتاب *sārh tṭy ẖālد ālktāb*
Sarah is giving Eman the book.	ساره تعطي إيمان الكتاب *sārh tṭy ʾiymān ālktāb*
Sarah gave Eman the book.	ساره أعطت إيمان الكتاب *sārh ʾaṭt ʾiymān ālktāb*
Sarah gave Khalid the book.	ساره أعطت خالد الكتاب *sārh ʾaṭt ẖālد ālktāb*
He gave Khalid the book.	هو أعطى خالد الكتاب *hw ʾaṭā ẖālد ālktāb*
He gave Khalid the book.	أعطى هو خالد الكتاب *ʾaṭā hw ẖālد ālktāb*
He gave Sarah the book.	هو أعطى ساره الكتاب *hw ʾaṭā sārh ālktāb*
He gave Sarah the book.	أعطى هو ساره الكتاب *ʾaṭā hw sārh ālktāb*
He is giving Khalid the book.	هو يعطي خالد الكتاب *hw yṭy ẖālد ālktāb*
She is giving Sarah the book.	هي تعطي ساره الكتاب *hy tṭy sārh ālktāb*
She is giving Khalid the book.	هي تعطي خالد الكتاب *hy tṭy ẖālد ālktāb*
She gave Sarah the book.	هي أعطت ساره الكتاب *hy ʾaṭt sārh ālktāb*
She gave Sarah the book.	أعطت هي ساره الكتاب *ʾaṭt hy sārh ālktāb*
She gave Khalid the book.	هي أعطت خالد الكتاب *hy ʾaṭt ẖālد ālktāb*
She gave Khalid the book.	أعطت هي خالد الكتاب *ʾaṭt hy ẖālد ālktāb*

Omar gave the book to Khalid.	عمر أعطى لخالد الكتاب *ʿmr ʾaṭā lẖālد ālktāb*
Omar gave the book to Khalid.	عمر أعطى الكتاب لخالد *ʿmr ʾaṭā ālktāb lẖālد*
Omar gave the book to Khalid.	عمر يعطي لخالد الكتاب *ʿmr yṭy lẖālد ālktāb*
Omar is giving the book to Khalid.	عمر يعطي الكتاب لخالد *ʿmr yṭy ālktāb lẖālد*
Omar is giving the book to Sarah.	عمر يعطي الكتاب لساره *ʿmr yṭy ālktāb lsārh*
Omar is giving the book to Sarah.	عمر يعطي لساره الكتاب *ʿmr yṭy lsārh ālktāb*
Omar gave the book to Sarah.	عمر أعطى الكتاب لساره *ʿmr ʾaṭā ālktāb lsārh*
Omar gave the book to Sarah.	عمر أعطى لساره الكتاب *ʿmr ʾaṭā lsārh ālktāb*
Eman is giving the book to Khalid.	إيمان تعطي لخالد الكتاب *ʾiymān tṭy lẖālد ālktāb*
Eman is giving the book to Khalid.	إيمان تعطي الكتاب لخالد *ʾiymān tṭy ālktāb lẖālد*
Eman is giving the book to Sarah.	إيمان تعطي الكتاب لساره *ʾiymān tṭy ālktāb lsārh*
Eman is giving the book to Sarah.	إيمان تعطي لساره الكتاب *ʾiymān tṭy lsārh ālktāb*
He gave the book to Khalid.	هو أعطى لخالد الكتاب *hw ʾaṭā lẖālد ālktāb*
He gave a book to Khalid.	هو أعطى كتاب لخالد *hw ʾaṭā ktāb lẖālد*
He is giving a book to Khalid.	هو يعطي لخالد كتاب *hw yṭy lẖālد ktāb*
He is giving a book to Khalid.	هو يعطي كتاب لخالد *hw yṭy ktāb lẖālد*
He is giving a book to Sarah.	هو يعطي كتاب لساره *hw yṭy ktāb lsārh*

He is giving a book to Sarah.	هو يعطي لساره كتاب *hw yʿṭy lsārh ktāb*
He gave a book to Sarah.	هو أعطى كتاب لساره *hw ʾaʿṭā ktāb lsārh*
He gave a book to Sarah.	هو أعطى لساره كتاب *hw ʾaʿṭā lsārh ktāb*
She is giving a book to Khalid.	هي تعطي لخالد كتاب *hy tʿṭy lẖālد ktāb*
She is giving a book to Khalid.	هي تعطي كتاب لخالد *hy tʿṭy ktāb lẖālد*
She is giving a book to Sarah.	هي تعطي كتاب لساره *hy tʿṭy ktāb lsārh*
She is giving a book to Sarah.	هي تعطي لساره كتاب *hy tʿṭy lsārh ktāb*
Eman is giving a book to Sarah.	إيمان تعطي لساره كتاب *ʾiymān tʿṭy lsārh ktāb*
He gave a book to Khalid.	هو أعطى لخالد كتاب *hw ʾaʿṭā lẖālد ktāb*
She is giving Sarah a book.	هي تعطي ساره كتاب *hy tʿṭy sārh ktāb*
Omar gave Khalid a book.	عمر أعطى خالد كتاب *ʿmr ʾaʿṭā ẖālد ktāb*
Khalid drives.	يسوق خالد *yswq ẖālد*
Khalid drives.	خالد يسوق *ẖālد yswq*
Khalid drives a lot.	خالد يسوق كثيرا *ẖālد yswq ktyrā*

D

Verbs in lexicon

Verbs in Arabic change depending on gender, number and tense of the subject, so there are multiple entries in the lexicon that translate to the same English output. The transliteration makes it clear that these are different words in Arabic.

Arabic	**Example**	**Logical Structure**
أريد *ʿaryd*	I want a ring.	$< TNS : PRES[do'(I, [want'(I, y)])] >$
نسيت *nsyt*	I forgot my wallet.	$< TNS : PAST[do'(I, [forget'(I, y)])] >$
أكلت *ʿaklt*	I ate an apple.	$< TNS : PAST[do'(I, [eat'(I, y)])] >$
شربت *šrbt*	I drank the milk.	$< TNS : PAST[do'(I, [drink'(I, y)])] >$
قرأ *qrʿa*	Omar read the book.	$< TNS : PAST[do'(x, [read'(x, y)])] >$
قرأت *qrʿat*	Eman read the book.	$< TNS : PAST[do'(x, [read'(x, y)])] >$

Arabic	**Example**	**Logical Structure**
NON	I am a tourist.	$be'(I, [tourist'])$
NON	He is an engineer.	$be'(he, [engineer'])$
NON	We are students.	$be'(we, [students'])$
شرب *šrb*	Khalid drank the milk.	$< TNS : PAST[do'(x, [drink'(x, y)])] >$
يشرب *yšrb*	Khalid is drinking the milk.	$< TNS : PRES[do'(x, [drink'(x, y)])] >$
سيشرب *syšrb*	Omar will drink the milk.	$< TNS : FUT[do'(x, [drink'(x, y)])] >$
يلبس *ylbs*	Eric is wearing his clothes.	$< TNS : PRES[do'(x, [wear'(x, y)])] >$
ضرب *ḍrb*	Louis hit Mark.	$< TNS : PAST[do'(x, [hit'(x, y)])] >$
يحب *yḥb*	Qays loves Laila.	$< TNS : PRES[do'(x, [love'(x, y)])] >$
تحب *tḥb*	Fatima loves Zaid.	$< TNS : PRES[do'(x, [love'(x, y)])] >$
قمت *qmt*	I have made a reservation.	$< TNS : PAST[do'(x, [make'(x, y)])] >$
فقدت *fqdt*	Eman lost her cat.	$< TNS : PAST[do'(x, [lose'(x, y)])] >$
قتل *qtl*	The man killed Jack.	$< TNS : PAST[do'(x, [kill'(x, y)])] >$
فتح *fṭḥ*	Sulaiman opened the door.	$< TNS : PAST[do'(x, [open'(x, y)])] >$
أخذ *ʾaẖd*	Zaid took the book.	$< TNS : PAST[do'(x, [take'(x, y)])] >$
يركب *yrkb*	Fahmy rides the car.	$< TNS : PRES[do'(x, [ride'(x, y)])] >$
ستطبخ *stṭbḫ*	Raiqa will cook the dinner.	$< TNS : FUT[do'(x, [cook'(x, y)])] >$

Arabic	Example	Logical Structure
أجاب ʔağāb	Khalid answered the question.	$< TNS : PAST[do'(x, [answer'(x, y)])] >$
يزور yzwr	Omar is visiting Ireland.	$< TNS : PRES[do'(x, [visit'(x, y)])] >$
يلعب ylʕb	Abbas is playing with the ball.	$< TNS : PRES[do'(x, [play'(x, y)])] >$
سيلعب sylʕb	Yousuf will play with the ball.	$< TNS : FUT[do'(x, [play'(x, y)])] >$
أكل ʔakl	Mansour ate with the spoon.	$< TNS : PAST[do'(x, [eat'(x, y)])] >$
يأكل yʔakl	Mansour is eating his food.	$< TNS : PRES[do'(x, [eat'(x, y)])] >$
سيأكل syʔakl	Eric will eat with the spoon.	$< TNS : FUT[do'(x, [eat'(x, y)])] >$
نفخ nfḫ	Suhaib bellowed the fire.	$< TNS : PAST[do'(x, [bellow'(x, y)])] >$
دهست dhst	Eman runned over her cat.	$< TNS : PAST[do'(x, [runnover'(x, y)])] >$
كسر ksr	Rashid broke the window.	$< TNS : PAST[do'(x, [break'(x, y)])] >$
تجرح tğrḥ	Sarah hurts Yousuf.	$< TNS : PRES[do'(x, [hurt'(x, y)])] >$
مزق mzq	Almahdi tore the book.	$< TNS : PAST[do'(x, [tear'(x, y)])] >$
مزقت mzqt	Ayah tore the page.	$< TNS : PAST[do'(x, [tear'(x, y)])] >$
فتح fṭḥ	Almahdi opened the window.	$< TNS : PAST[do'(x, [open'(x, y)])] >$
مشط mšṭ	James combed his hair.	$< TNS : PAST[do'(x, [comb'(x, y)])] >$
نظف nẓf	Eric cleaned the plane.	$< TNS : PAST[do'(x, [clean'(x, y)])] >$

Arabic	**Example**	**Logical Structure**
قرص *qrṣ*	Harold pinched James.	$< TNS : PAST[do'(x, [pinch'(x, y)])] >$
تنفق *tnfq*	Ayah is spending her money.	$< TNS : PRES[do'(x, [spend'(x, y)])] >$
فقد *fqd*	Henry lost his money.	$< TNS : PAST[do'(x, [lose'(x, y)])] >$
لكم *lkm*	Adam punched Philip.	$< TNS : PAST[do'(x, [punch'(x, y)])] >$
تكره *tkrh*	Sarah hates Zakiah.	$< TNS : PRES[do'(x, [hate'(x, y)])] >$
لكمت *lkmt*	Sarah punched Sarah.	$< TNS : PAST[do'(x, [punch'(x, y)])] >$
قتلت *qtlt*	Zakiah killed Sarah.	$< TNS : PAST[do'(x, [kill'(x, y)])] >$
قتل *qtl*	Jack killed Mary.	$< TNS : PAST[do'(x, [kill'(x, y)])] >$
صفع *ṣfʿ*	Mark slapped Louis.	$< TNS : PAST[do'(x, [slap'(x, y)])] >$
هاتفت *hātft*	Ayesha phoned Eman.	$< TNS : PAST[do'(x, [phone'(x, y)])] >$
شكرت *škrt*	Ayah thanked Khalid.	$< TNS : PAST[do'(x, [thank'(x, y)])] >$
نادت *nādt*	Sarah called Adam.	$< TNS : PAST[do'(x, [call'(x, y)])] >$
فاز *fāz*	Omar won the race.	$< TNS : PAST[do'(x, [win'(x, y)])] >$
رأت *rʾat*	Eman saw Sarah.	$< TNS : PAST[do'(x, [see'(x, y)])] >$
مسك *msk*	Philip caught the ball.	$< TNS : PAST[do'(x, [catch'(x, y)])] >$
إشترى *ʾištrā*	Carl bought pens.	$< TNS : PAST[do'(x, [buy'(x, y)])] >$
قاد *qād*	Mark drove the bus.	$< TNS : PAST[do'(x, [drive'(x, y)])] >$
ينظف *ynẓf*	Adam is cleaning his room.	$< TNS : PRES[do'(x, [clean'(x, y)])] >$

Arabic	**Example**	**Logical Structure**
سينظف *synẓf*	Mark will clean my kitchen.	$< TNS : FUT[do'(x, [clean'(x, y)])] >$
ستنظف *stnẓf*	Sarah will clean my office.	$< TNS : FUT[do'(x, [clean'(x, y)])] >$
يغسل *yġsl*	Louis is washing the dishes.	$< TNS : PRES[do'(x, [wash'(x, y)])] >$
يطعم *yṭʿm*	Harold is feeding his cat.	$< TNS : PRES[do'(x, [feed'(x, y)])] >$
يصلح *yṣlḥ*	Eric is fixing his car.	$< TNS : PRES[do'(x, [fix'(x, y)])] >$
يتكلم *ytklm*	Fahmy speaks English.	$< TNS : PRES[do'(x, [speak'(x, y)])] >$
تطبخ *tṭbḥ*	Ayah is cooking the dinner.	$< TNS : PRES[do'(x, [cook'(x, y)])] >$
يساعد *ysāʿd*	Rashid is helping Mark.	$< TNS : PRES[do'(x, [help'(x, y)])] >$
يأكل *yʾakl*	Mansour is eating his food.	$< TNS : PRES[do'(x, [eat'(x, y)])] >$
يمشط *ymšṭ*	Carl is brushing his hair	$< TNS : PRES[do'(x, [brush'(x, y)])] >$
يفرش *yfrš*	Abbas is brushing his teeth.	$< TNS : PRES[do'(x, [brush'(x, y)])] >$
يلبس *ylbs*	Yousuf is wearing his shoes.	$< TNS : PRES[do'(x, [wear'(x, y)])] >$
ينام *ynām*	Omar sleeps early.	$< TNS : PRES[do'(x, [sleep'(x, y)])] >$
يشاهد *yšāhd*	Henry is watching the TV.	$< TNS : PRES[do'(x, [watch'(x, y)])] >$
يزرع *yzrʿ*	Almahdi is planting the trees.	$< TNS : PRES[do'(x, [plant'(x, y)])] >$
إصطاد *ʾiṣṭād*	Sulaiman caught the fishs.	$< TNS : PAST[do'(x, [catch'(x, y)])] >$
جر *ǧr*	James pushed the chairs.	$< TNS : PAST[do'(x, [push'(x, y)])] >$

Arabic	Example	Logical Structure
كتبت *ktbt*	Ayesha wrote the book.	$< TNS : PAST[do'(x, [write'(x, y)])] >$
كتب *ktb*	Brian wrote the book.	$< TNS : PAST[do'(x, [write'(x, y)])] >$
مسحت *msḥt*	Fatima wiped the house.	$< TNS : PAST[do'(x, [wipe'(x, y)])] >$
رسمت *rsmt*	Raiqa drew trees.	$< TNS : PAST[do'(x, [draw'(x, y)])] >$
قطفت *qṭft*	Ayesha picked the flowers.	$< TNS : PAST[do'(x, [pick'(x, y)])] >$
كوت *kwt*	Eman ironed the clothes.	$< TNS : PAST[do'(x, [iron'(x, y)])] >$
إشترى *ʾištrā*	Omar bought the toys.	$< TNS : PAST[do'(x, [buy'(x, y)])] >$
لونت *lwnt*	Ayah painted the picture.	$< TNS : PAST[do'(x, [paint'(x, y)])] >$
أعطاك *ʿaṭāk*	Omar gave you the book.	$< TNS : PAST[do'(x, [give'(x, y)])] >$
يصلح *yṣlḥ*	Omar is fixing the computer.	$< TNS : PAST[do'(x, [fix'(x, y)])] >$
يعمل *yʿml*	Yasser works hard.	$< TNS : PRES[do'(x, [work'(x, y)])] >$
مرر *mrr*	Philip passed the ball.	$< TNS : PAST[do'(x, [pass'(x, y)])] >$
يسوق *yswq*	Khalid drives.	$< TNS : PRES << [do'(x, [drive'(x)])] >>>$
يبكون *ybkwn*	The children are crying.	$< TNS : PRES << [do'(x, [cry'(x)])] >>>$
يقرأ *yqrʿa*	Omar reads.	$< TNS : PRES << [do'(x, [read'(x)])] >>>$

Arabic	يصرصر *yṣrṣr*
Example	the wheel squeaks.
Logical Structure	$< TNS : PRES << [do'(x, [squeak'(x)])] >>>$
Arabic	ينام *ynām*
Example	Omar is sleeping.
Logical Structure	$< TNS : PRES << [do'(x, [sleep'(x)])] >>>$
Arabic	أعطى *ʿaṭā*
Example	Omar gave Khalid the book.
Logical Structure	$< TNS : PAST[do'(x, 0)CAUSE[BECOMEhave'(y, z)]] >$
Arabic	يعطي *yṭy*
Example	Omar is giving Eman a book.
Logical Structure	$< TNS : PRES[do'(x, 0)CAUSE[BECOMEhave'(y, z)]] >$
Arabic	تعطي *tṭy*
Example	Sarah is giving Eman a book.
Logical Structure	$< TNS : PRES[do'(x, 0)CAUSE[BECOMEhave'(y, z)]] >$
Arabic	أعطت *ʿaṭt*
Example	Sarah gave Khalid a book.
Logical Structure	$< TNS : PAST[do'(x, 0)CAUSE[BECOMEhave'(y, z)]] >$

Arabic	أرت *art*
Example	Fatima showed the letter to Khalid.
Logical Structure	$< TNS : PAST[do'(x, 0)CAUSE[BECOMEsee'(y, z)]] >$
Arabic	يري *yry*
Example	Mark is showing Brian the letter.
Logical Structure	$< TNS : PRES[do'(x, 0)CAUSE[BECOMEsee'(y, z)]] >$
Arabic	أرى *arā*
Example	Brian showed the letter to Sarah.
Logical Structure	$< TNS : PAST[do'(x, 0)CAUSE[BECOMEsee'(y, z)]] >$
Arabic	تري *try*
Example	Fatima is showing Adam the letter.
Logical Structure	$< TNS : PRES[do'(x, 0)CAUSE[BECOMEsee'(y, z)]] >$
Arabic	يدرس *ydrs*
Example	Suhaib is teaching Eman the history.
Logical Structure	$< TNS : PRES[do'(x, 0)CAUSE[BECOMEknow'(y, z)]] >$
Arabic	تدرس *tdrs*
Example	Eman is teaching mathematics to Sarah.
Logical Structure	$< TNS : PRES[do'(x, 0)CAUSE[BECOMEknow'(y, z)]] >$

Arabic	أدرس *ʾadrs*
Example	I am teaching mathematics to Sarah.
Logical Structure	$< TNS : PRES[do'(x, 0)CAUSE[BECOMEknow'(y, z)]] >$
Arabic	درس *drs*
Example	Suhaib taught Mark mathematics.
Logical Structure	$< TNS : PAST[do'(x, 0)CAUSE[BECOMEknow'(y, z)]] >$

Arabic	**Example**	**Logical Structure**
فاتتني *fātny*	I missed the plane.	$< TNS : PAST[do'(I, [miss'(I, y])] >$
أريد *ʾaryd*	I want a ring.	$< TNS : PRES[do'(I, [want'(I, ring])] >$
نسيت *nsyt*	I forgot my wallet.	$< TNS : PAST[do'(I, [forget'(I, wallet])] >$
أكلت *ʾaklt*	I ate the food.	$< TNS : PAST[do'(I, [eat'(I, food])] >$
شربت *šrbt*	I drank the milk.	$< TNS : PAST[do'(I, [drink'(I, milk])] >$

E

The UniArab code

Given the large amount of code developed as part of the work presented in this thesis, it is available in the attached CD rather than included here.

Package: Name of Class	**Class Summary**
pkg1: ArabicToEnglishMT	The main class
pkg1: AdjectiveXMLWriter	To write an adjective in the datasource
pkg1: Adjective	To hold adjective attributes from the datasource
pkg1: AdverbXMLWriter	To write an adverb in the datasource
pkg1: Adverb	To hold adverb attributes from the datasource
pkg1: DemonstrativeXMLWriter	To write a demonstrative in the datasource
pkg1: Demonstrative	To hold demonstrative attributes from the datasource
pkg1: Global	This class to add a new word in lexicon
pkg1: NounXMLWriter	To write a noun in the datasource
pkg1: Noun	To hold noun attributes from the datasource

Package: Name of Class	Class Summary
pkg1: OtherWordXMLWriter	To write an OtherWord in the datasource
pkg1: OtherWord	To hold OtherWord attributes from the datasource
pkg1: Preparation	To change the value from lexicon interface's list to be saved in datasource
pkg1: ProperNounXMLWriter	To write a proper noun in the datasource
pkg1: ProperNoun	To hold proper noun attributes from the datasource
pkg1: SearchEngine2	This class to manage search in datasource
pkg1: TempAdjectiveXML	To manage a written an adjective in the XML datasource while the UniArab system is running
pkg1: TempAdverbXML	To manage a written a new adverb in the XML datasource while the UniArab system is running
pkg1: TempDemonstrativeXML	To manage a written a new demonstrative in the XML datasource while the UniArab system is running
pkg1: TempNounXML	To manage a written a new noun in the XML datasource while the UniArab system is running
pkg1: TempOtherWordXML	To manage a written a new OtherWord in the XML datasource while the UniArab system is running
pkg1: TempProperNounXML	To manage a written a new proper noun in the XML datasource while the UniArab system is running
pkg1: TempVerbXML	To manage a written a new verb in the XML datasource while the UniArab system is running
pkg1: Tokenizer	This class to split a sentence into word tokens
pkg1: VerbXMLWriter	To write a verb in the datasource
pkg1: Verb	To hold verb attributes from the datasource

Package: Name of Class	Class Summary
gui: AdjectivePanel	This class to mange the adjective panel in the UniArabs lexicon interface
gui: AdverbPanel	This class to mange the adverb panel in the UniArabs lexicon interface
gui: DemonstrativesPanel	This class to mange the demonstrative panel in the UniArabs lexicon interface
gui: NounPanel	This class to mange the noun panel in the UniArabs lexicon interface
gui: OtherWordPanel	This class to mange the OtherWord panel in the UniArabs lexicon interface
gui: ProperNounPanel	This class to mange the proper noun panel in the UniArabs lexicon interface
gui: VerbPanel	This class to mange the verb panel in the UniArabs lexicon interface
uniArab: PreUniArab	This class to mange the POS of input words
uniArab: UniArab	This class to preparation of syntactic parser
uniArab: GenerationLS	This class to generate the logical structure
syntaxGeneration: syntaxGeneration	This class to mange the syntax generation
generationEnglishMorphology: pressTenseToBe	This class to mange the generation of target language morphology

Package: Name of Class	Class Summary
xml: AdjectiveDB.XML	This is the adjectives stored in the XML datasource
xml: AdverbDB.XML	This is the adverbs stored in the XML datasource
xml: DemonstrativeDB.XML	This is the demonstratives stored in the XML datasource
xml: NounDB.XML	This is the nouns stored in the XML datasource
xml: OtherWordDB.XML	This is the OtherWords stored in the XML datasource
xml: ProperNounDB.XML	This is the proper nouns stored in the XML datasource
xml: VerbDB.XML	This is the verbs stored in the XML datasource