The Development of English – Arabic Machine Translation System Modules as an Advanced Software Engineering Course

Ahmed Guessoum
Department of Computer Science
The University of Sharjah
P.O. Box 27272, Sharjah, UAE
e-mail: guessoum@sharjah.ac.ae

Abstract

Developing a Machine Translation (MT) system is well known to be a time and effort consuming task. It requires the development of monolingual and bi-lingual dictionaries, lexical analyzers and generators, and parsers and generators of the source and target languages, in addition to a transfer module in the case of transfer-based MT, or some equivalent module. This activity is usually undertaken by dedicated research centers and software development companies. This paper reports on an attempt to develop various modules of an English-to-Arabic MT system as part of a course. The course was taught as an advanced software engineering course to students working on their graduation projects. We explain the adopted work methodology, the various modules that were worked on, as well as the achievements and difficulties.

Keywords: Machine translation; language engineering; Arabic, lexical analyzer, morphological analyzer, parser, Arabic lexical generator, Arabic sentence generator.

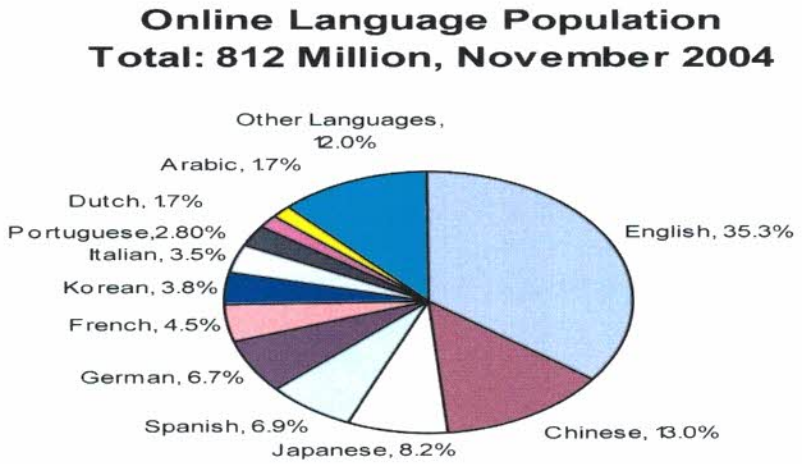
I. Introduction

There is no doubt that the information explosion that we have been witnessing since the inception of the Internet in the early 90s has revolutionized human societies. Never before have people been able to access information with such ease and speed. According to an MIT study, 6.5 Exabytes (10^{18} bytes) of information were produced in 2003 (i.e. 100 million billion documents of 10 KB each) [1].

This Massive production of information occurs in the industry, trade, financial sectors, political arenas, media, education, research, etc. It is indeed an information revolution/explosion which is characterized by the ever decreasing costs of (personal) computers, major advances in networking and wireless communication, the Internet/www as a central pillar of modern societies and economies. This internet growth has been exponential. According to Internet Software Consortium (www.isc.org) we have moved from about 50 sites in 1994 to an estimated 300 million sites in December 2004! In fact, even a look at the daily growth of the Internet gives a stunning picture of how fast it occurs. Indeed, the following table, reproduced from http://www.whois.sc/internet-statistics/ , shows that under the **.COM** domain alone, the number of new sites on 20/12/2004 was 17,951. The table shows the daily growth of the most popular internet domains.

Domain Counts						
			Daily Changes (last 24hrs)			
Active	Deleted	On-Hold	New	Deleted	Transfered	TLD
32,797,004	17,465,006	346,746	23,164	17,969	17,951	.COM
5,245,718	3,392,406	63,376	3,462	3,102	3,008	.NET
3,262,971	2,051,575	33,401	1,840	1,733	1,201	.ORG
2,949,431	954,993	1,274	4,136	832	1,519	.INFO
1,068,916	419,201	1,021	752	1,572	561	.BIZ
889,522	503,853	490	383	994	355	.US
46,213,562	24,639,819	446,308	33,737	26,202	24,595	Total
Last Updated 12/20/2004						

In terms of internet users, these were estimated in November 2004 by the Internet World Statistics (*www.InternetWorldStats.com*) as 812 million distributed by language use as follows:



This major information revolution has become both a challenge and an opportunity to language engineers. Indeed, it is primordial to develop the necessary tools to automatically process this overwhelming flow of information. This is even more of a challenge for languages like Arabic that have not had as much attention in terms of language engineering as their counterparts like English, Latin languages, German, Japanese, Russian, Chinese, etc. Only recently, and for obvious geo-strategic reasons, has there been a revival of interest for the Arabic language by western institutions and R&D funding agencies. In the Arab world, although a number of institutions and people have been working in the area, the efforts have most of the time lacked the proper funding, pan-Arab cooperation, etc.[2].

With this as a background, we have decided to sail in the deep seas of Arabic Language Engineering. In particular, we have taken up the task of challenging graduating students who registered in an Advanced Software Engineering course to work on the development of various modules of an English-to-Arabic MT system. The aims were manifold:

- Gaining experience with the inner intricacies of machine translation through a prototypical exploration
- Developing any modules that could later be used in a more solid implementation
- Gaining experience with teaching an advanced software engineering course in the form of an emulation of actual, interacting software development teams.

In Section 2, we present the setting for the development of the MT system modules. These will be followed in the same section by the system architecture with explanations of the various modules. In section 3, we will present the implementation of the various modules and what has been achieved on each one that was tackled. In section 4, we will draw some conclusions about the benefits of the experiment as well as the limitations that we have faced and what can be reused from the developed modules.

II. MT System Development Setting and Architecture

II.1. Course Setting

The development of the MT system modules was done as part of a course with the following description. The course was organized as an Advanced Software Engineering course. Ten (10) students registered in the course. The course description given to them stated that the aim was the attempt to develop a Prototype English-to-Arabic Machine Translation System. The description also stated that the system architecture would be given to the students and every couple of students would be asked to take care of one or more of its components. The various teams would first write the system and component specifications and then move through the various stages of software development: design, implementation, testing, and debugging. Wherever it would judged more appropriate, a different software development model could be used by any of the teams.

Four teams of students were formed: a 1-student team to take care of the English lexical and morphological analyzers, a 3-student team for the English parser (and generator) and the Arabic generator (and parser), a 3-student team for the transfer module, and a 3-student team for the Arabic morphological generator. I formed these teams and distributed the tasks to them based on (1) my knowledge of the students' backgrounds, so as to make the work as efficient as possible, and (2) relationships to one another, to avoid conflicts of personalities that may arise during times of pressure.

II.2. MT System Architecture

As presented in [3], MT systems can follow different approaches: direct approach, transfer approach, Interlingua approach. They can also be classified as knowledge-based, translation memory-based, or statistical MT systems. Each of these has its own characteristics. In our case, and to make it closer to the background of the students, a number of which had taken an Artificial Intelligence course and used Prolog for their programming assignments, I decided to direct the students towards a knowledge-based transfer-based MT system.

This technical decision having been taken, the system architecture was quickly reached. It is shown in the diagram given below.

The Arabic MT system was divided into six components as shown in Figure 1. These components are:

- English Language Morphological & Lexical Analyzer: Each term of the English text is analyzed based on its stem and formation. This module produces the word analysis and sends it to the English Parser.

- English Language Parser: This module consists of a grammar for the English language. The English terms of the sentence along with their lexical features are passed through the English grammar rules to find their match so that the grammatical categories get added to the terms. All this is used to update the feature structures.

- Transfer Module: To avoid the tremendous problem of having to develop transfer rules for Arabic, a task almost impossible for novice students working on a 4-month project in an area they are not familiar with. As such, 3 students were asked to use a novel approach to develop the transfer module. Indeed, they were to develop a Neural Network-based transfer rule. Its basic task was to produce the feature structure, in the target text (in Arabic), which corresponds to the source text (in English) feature structure. Thus, this module would take as input a feature structure produced by the English Parser module and would produce as output another feature structure, this time in Arabic. During the training of the Neural Network

however, the NN is meant to be trained by the provision of training data manually prepared. Once the implementation of the English and the Arabic Parsers gets completed, they can be used to train the Neural Network automatically.

- English Language to Arabic Language Dictionary: This module looks up the bilingual dictionary to get the Arabic translation of English words.

- Arabic Language Parser & Generator:
This module consists of two sub modules, the Arabic Parser and the Arabic Generator. The Arabic Parser consists of a grammar for the Arabic language. The Arabic terms of the sentence along with their lexical features are passed through the Arabic grammar rules to find their match so that the grammatical categories get added to the terms. All this is used to update the feature structures. This module is also used for training the Neural Network.

The Arabic sentence generator works in a reverse order. It takes the analyzed Arabic sentence along with all its features and passes it through the Arabic grammar rules to find a match that ends in a complete translated sentence, which is produced to the user.

- Arabic Language Morphological Generator:
This component receives the feature structure of a term or a set of terms, and forms (generates) the structure of the Arabic word and then returns it to the Arabic sentence Generator which will form the proper translated sentence.

In terms of assignments, one student was assigned the English lexical and morphological analysis modules. Three students were assigned the tasks of developing parsers-generators for English and for Arabic. Three other students were assigned the task of developing the neural network-based transfer module. Another three students were given the task of implementing the Arabic morphological generator.

III. System Modules development and Implementation

III.1. English Lexical and Morphological Analyzers

Lexical analysis is the process of breaking up a language stream into tokens of different types (e.g. words, numbers, punctuations etc.). Morphology is the study of the structure and formation of words. Its most important unit is the morpheme, which is defined as the "minimal unit of meaning" [4].

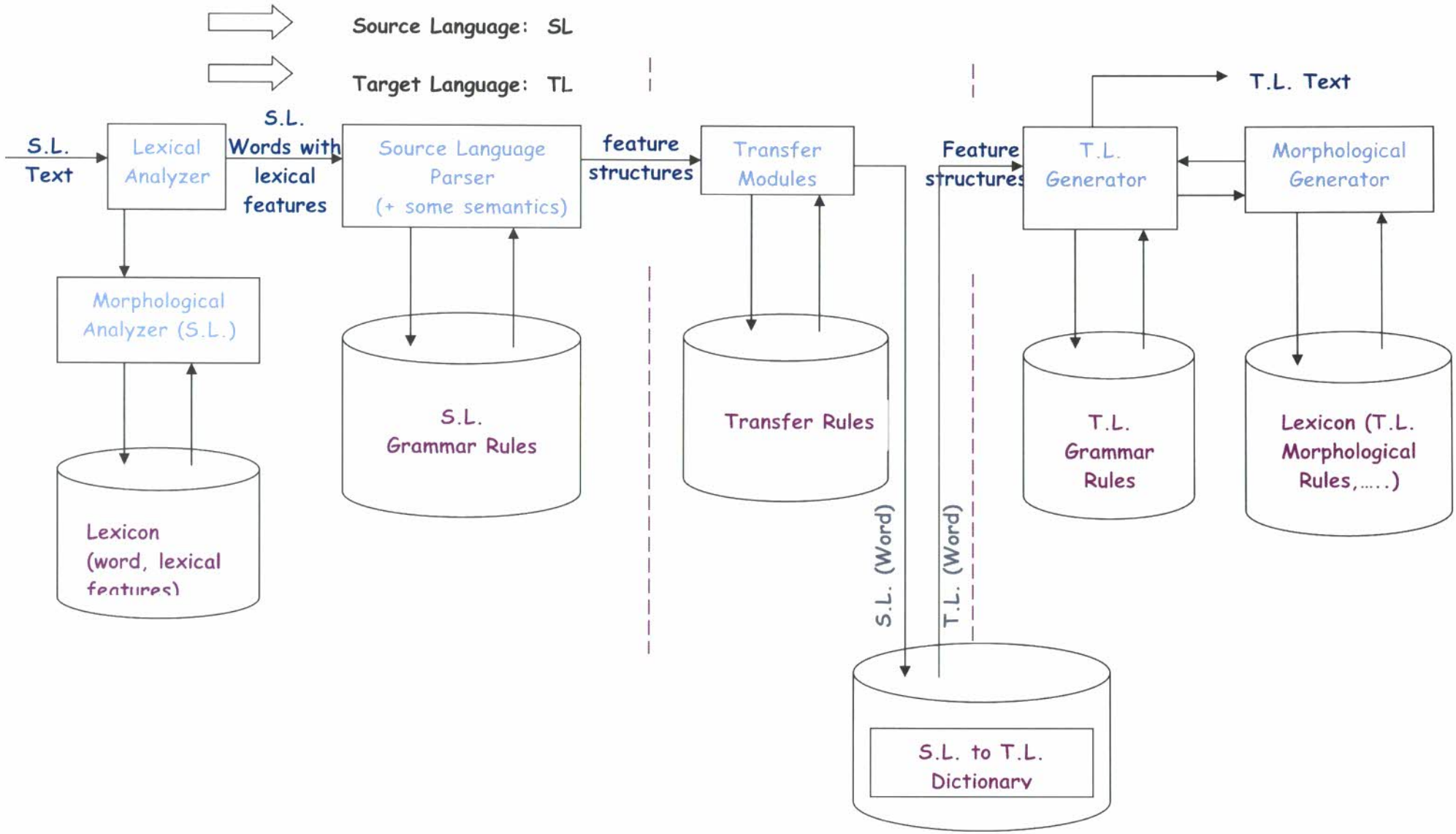
These modules were implemented using Java. The results were very encouraging as the modules were very stable and able to correctly analyze about 97% of all the tested cases.

Here are some sample analyses produced by these modules. The input sentences are:

The road to success is not straight. There is a curve called failure. A loop called confusion.

The English lexical and morphological analyzers produce:
[the +article +sp] [road +n] [to +pp , to +cj] [success +n] [be +verbtobe +sg] [not +av , not +det] [straight +aj , straight +av] [there +av , there +pr] [be +verbtobe +sg] [a +article +sg] [curve +n , curve +v] [called +v +past , called +v +past +part , called +aj] [failure +n] [a +article +sg] [loop +n , loop +v] [called +v +past , called +v +past +part , called +aj] [confusion +n]

Figure 1: System Architecture



III.2. English Parser

The English Parser reads the sentence produced by the English Lexical and Morphological Analyzers as a list of lists (to be easily useable in Prolog). It then tries to match this sentence with some grammar rules to check if the structure of this sentence is correct. If it finds a match, it creates a feature structure and sends it to the Neural Network; else it sends the sentence to a default case which builds blindly a feature structure for the entered sentence. This is to ensure robustness of the MT system.

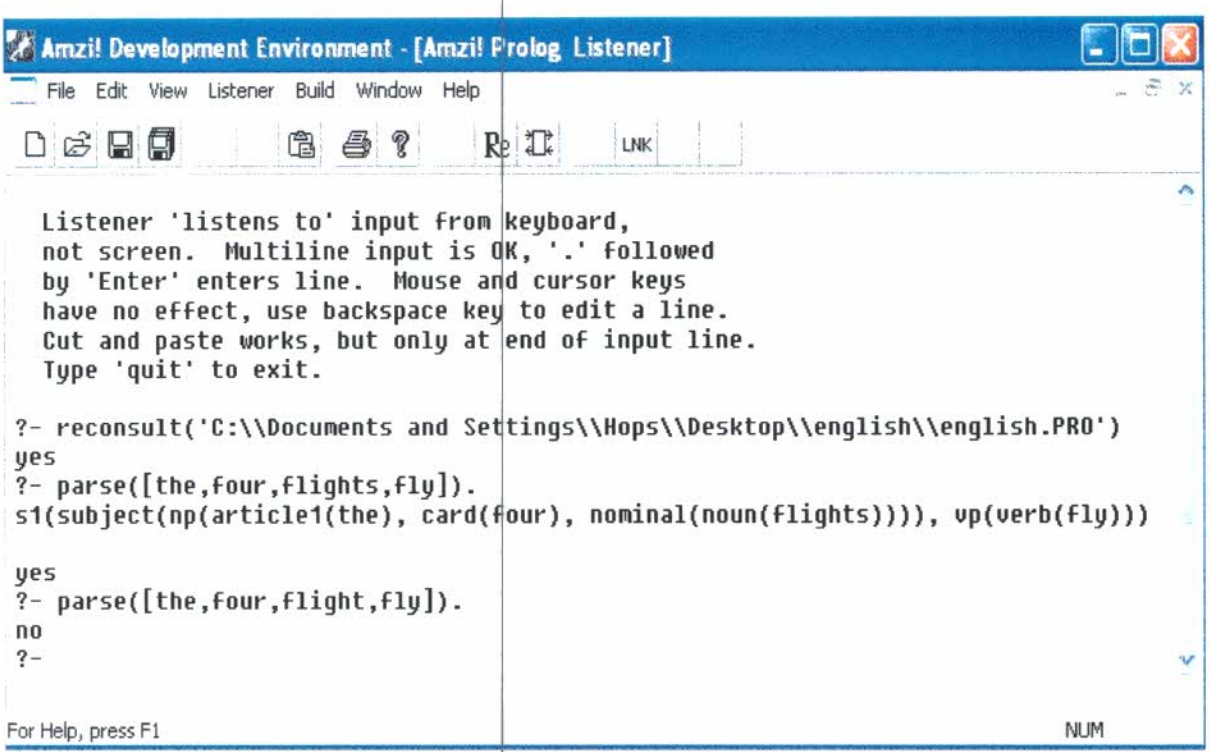
A grammar of English was developed and refined and a parser based on it was implemented using AMZI Prolog [5]. It produced parse trees with feature structures. A feature structure is a set of feature-value pairs, where features are simple symbols that cannot be further analyzed. Each feature in the feature structure has a different name, associated with one or more values.

Consider for example the sentence *the boy ate*. The feature structure of each word will look like the following:

$$\begin{bmatrix} value : the \\ type : \text{det} \end{bmatrix} \qquad \begin{bmatrix} value : boy \\ type : noun \\ number : sg \end{bmatrix} \qquad \begin{bmatrix} value : ate \\ type : verb \\ tense : past \\ firstPerson : 1p \end{bmatrix}$$

The English parser covered an extensive part of English. This includes various types of sentences, questions, auxiliary clauses, adjective clauses, conditional statements, cardinals, ordinals, determiners, adverbs, pronouns, prepositions, negation, conjunction, as well as email addresses, URLs, etc.

The following screen shot shows the parsing of a correct sentence of an incorrect sentence.



III.3. Arabic Parser/Generator

The Arabic Parser reads the structures of a sentence, and then tries to match these with the available Arabic grammars in order to generate a correct Arabic sentence. If it cannot match it, it produces a default match so that the MT system can carry on.

A major effort was put here to develop an extensive grammar of Arabic. This was particularly difficult as one does not find grammar books of Arabic where the grammar is presented in some form clause enough to BNF notation or similar forms. The students here had to double efforts to collect all the bits and pieces of Arabic grammar they could get or think of.

The Arabic grammatical constructs covered here are verbal sentences, nominal sentences, questions, possessives, adjectives, conjunctions, etc. The following shows an example of a correct sentence which gets parsed correctly and an example of an incorrect input.

```
?- reconsult('C:\\Documents and Settings\\winxp\\Desktop\\New Folder\\arabic.pro')
yes
?  parse([حتى,يأكل,الولد,التفاحة]).
حرف_قرر(حرف_يدخل_على_فعل(حرف_نصب(حتى)), فعل_مضارع(يأكل), الفاعل(اسم(الولد)), مفعول_به(اسم(التفا
ح)))))

yes
?- parse([حتى,أكل,الولد,التفاحة]).
no
?-
```

Both the Arabic parser and the English parser gave very good results. We need to do a systematic corpus-based testing of these parsers to see whether our initial tests are confirmed that the parsers have a very high coverage of the languages.

III.4. The Transfer Module

Unfortunately, the weakest link in the whole MT system development process was the transfer module. The students had problems understanding the concept of neural networks, designing the appropriate architecture, and going through the training and testing of the network. This has unfortunately not given the other students the joy of testing the overall system. Nevertheless a Master's student has recently taken up the task and is actively working on the learning of a transfer module from bilingual English-to-Arabic corpora.

III.5. Arabic Morphological Generator

This component receives the feature structure(s) of a term or a set of terms and produces (generates) the structure of the Arabic word and then returns it to the Arabic Generator to form the proper translated sentence. The features expected as input from the Arabic parser are as follows:

Data Type: noun **Translated word:** the translated word **Gender:** masculine , feminine **Definite:** define , not define **Quantity:** 0,1,2,3, etc. (any positive integer) **Number:** singular, dual, plural **Particle:** one of the particles that come with nouns **State:** accusative, indicative, non-vowelled **Pronoun:** the actual pronoun used	**التصنيف:** إسم **الترجمة:** "ترجمة الكلمة" **الجنس:** مذكر , مؤنث **التعريف:** معرف , لا يعرف **الكمية:** 0,1,2,.....أو أي عدد صحيح موجب **العدد:** مفرد , مثنى , جمع **الأداة:** أداة من أدوات الجر:{ من , إلى , عن , على, في , ب, ك, ل, و, ت, رب, خلا, عدا, حاشا,مذ , منذ} أو أحد الحروف الناسخة(الناصبة):{إن, أن, كأن, لكن, ليت, لعل}. **الحالة:** مرفوع , منصوب , مجرور **الضمير:** ها , ه

Data Type: verb **Translated word:** the translated word **Tense:** present , past, imperative **Gender:** masculine , feminine **Number:** singular, dual, plural **Pronoun:** he , she , they, etc **Particle:** one of the particles that can be used with verbs. **State:** accusative, indicative, non-**vowelled**	**التصنيف:** فعل **الترجمة:** "ترجمة الكلمة" **الزمن:** مضارع , ماضي , أمر **الجنس:** مذكر , مؤنث **العدد:** مفرد , مثنى , جمع **الضمير:** ضمير من ضمائر الغائب:{ هو, هي, هما, هم, هن } أو المتكلم:{أنا , نحن } أو المخاطب:{أنت, أنت, أنتما, أنتم, أنتن } **الأداة:** أداة من الأدوات الناصبة: {أن , لن, كي, لا, ل } أو الجازمة: {لم, لما, ل } أو أي أداة من الأدوات التي تدخل على الأفعال. **الحالة:** مرفوع , منصوب , مجزوم

The module uses a data base which was automatically created from files produced by researchers at KACST led by Prof. Mohammed El-Hannache. The latter gave the students a little less than 4000 entries. Each entry gives the morphological details for some Arabic word as shown in Figure 2.

The Arabic Morphological Generator takes a set of features in the input and produces an Arabic word that would be passed back to the Arabic sentence (syntactic) generator. Here are some examples.

The following examples show the generation of nouns.

Input	Output
التصنيف: إسم الترجمة : ضؤضؤ الجنس: مذكر التعريف: غير معرف الكمية : العدد : مفرد الأداة : الحالة : الضمير :	ضُؤْضُؤ

Input	Output
التصنيف: إسم الترجمة : ضؤضؤ الجنس: مذكر التعريف: غير معرف الكمية : العدد : مثنى الأداة : الحالة : الضمير: ها	ضُؤْضُؤ انها

Figure 3.2: **Sample of an text file that contains verb data (which has been converted into a database)**

جدول تصريف الفعل (بَعَثَ - يَبْعَثُ) - 3/ت -

مُجَرَّد ثلاثي / فَعَلَ - يَفْعَلُ / صحيح سالم .

الضمائر	المبني المعلوم								المبني للمجهول					
	الماضي	المضارع			المضارع المؤكد		الأمر	الأمر المؤكد		الماضي	المضارع			المضارع المؤكد	
		المرفوع	المنصوب	المجزوم	بالثقيل	بالخفيف		بالثقيل	بالخفيف		المرفوع	المنصوب	المجزوم	بالثقيل	بالخفيف
أَنَا	بَعَثْتُ	أَبْعَثُ	أَبْعَثَ	أَبْعَثْ	أَبْعَثَنَّ	أَبْعَثَنْ				بُعِثْتُ	أُبْعَثُ	أُبْعَثَ	أُبْعَثْ	أُبْعَثَنَّ	أُبْعَثَنْ
نَحْنُ	بَعَثْنَا	نَبْعَثُ	نَبْعَثَ	نَبْعَثْ	نَبْعَثَنَّ	نَبْعَثَنْ				بُعِثْنَا	نُبْعَثُ	نُبْعَثَ	نُبْعَثْ	نُبْعَثَنَّ	نُبْعَثَنْ
أَنْتَ	بَعَثْتَ	تَبْعَثُ	تَبْعَثَ	تَبْعَثْ	تَبْعَثَنَّ	تَبْعَثَنْ	ابْعَثْ	ابْعَثَنَّ	ابْعَثَنْ	بُعِثْتَ	تُبْعَثُ	تُبْعَثَ	تُبْعَثْ	تُبْعَثَنَّ	تُبْعَثَنْ
أَنْتِ	بَعَثْتِ	تَبْعَثِينَ	تَبْعَثِي		تَبْعَثَنَّ	تَبْعَثَنْ	ابْعَثِي	ابْعَثَنَّ	ابْعَثَنْ	بُعِثْتِ	تُبْعَثِينَ	تُبْعَثِي		تُبْعَثَنَّ	تُبْعَثَنْ
أَنْتُمَا	بَعَثْتُمَا	تَبْعَثَانِ	تَبْعَثَا		تَبْعَثَانِّ	*	ابْعَثَا	ابْعَثَانِّ	*	بُعِثْتُمَا	تُبْعَثَانِ	تُبْعَثَا		تُبْعَثَانِّ	*
أَنْتُم	بَعَثْتُمْ	تَبْعَثُونَ	تَبْعَثُوا		تَبْعَثُنَّ	تَبْعَثُنْ	ابْعَثُوا	ابْعَثُنَّ	ابْعَثُنْ	بُعِثْتُم	تُبْعَثُونَ	تُبْعَثُوا		تُبْعَثُنَّ	تُبْعَثُنْ
أَنْتُنَّ	بَعَثْتُنَّ		تَبْعَثْنَ		تَبْعَثْنَانِّ	*	ابْعَثْنَ	ابْعَثْنَانِّ	*	بُعِثْتُنَّ	تُبْعَثْنَ			تُبْعَثْنَانِّ	*
هُوَ	بَعَثَ	يَبْعَثُ	يَبْعَثَ	يَبْعَثْ	يَبْعَثَنَّ	يَبْعَثَنْ				بُعِثَ	يُبْعَثُ	يُبْعَثَ	يُبْعَثْ	يُبْعَثَنَّ	يُبْعَثَنْ
هِيَ	بَعَثَتْ	تَبْعَثُ	تَبْعَثَ	تَبْعَثْ	تَبْعَثَنَّ	تَبْعَثَنْ				بُعِثَتْ	تُبْعَثُ	تُبْعَثَ	تُبْعَثْ	تُبْعَثَنَّ	تُبْعَثَنْ
هُمَا	بَعَثَا	يَبْعَثَانِ	يَبْعَثَا		يَبْعَثَانِّ	*				بُعِثَا	يُبْعَثَانِ	يُبْعَثَا		يُبْعَثَانِّ	*

Input	Output
التصنيف: إسم الترجمة : ضباب الجنس: مذكر التعريف: غير معرف الكمية : العدد : جمع الأداة : الحالة : الضمير :	ضَبَابون

Input	Output
التصنيف: إسم الترجمة : ضوضؤ الجنس: مذكر التعريف: معرف الكمية : 2 العدد : مثنى الأداة : أن الحالة : منصوب الضمير :	أن الضَوْضَوْين

The following examples show the generation of verbs.

Input	Output
التصنيف: فعل الترجمة : بعث الزمن: مضارع الجنس: مذكر العدد : مثنى الضمير: هما الأداة : الحالة : مرفوع	يَبْعَثَان

Input	Output
التصنيف: فعل الترجمة : بعث الزمن: ماضي الجنس: مؤنث العدد : مفرد الضمير: أنت الأداة : الحالة : منصوب	بَعَثْتِ

V. Conclusion

The experiment we have tried of teaching Machine Translation system development as an advanced software engineering course has been very fruitful. While we were initially skeptical whether the students would be able to develop the various modules that were assigned to them, the results were overall very satisfactory. Except for the transfer module which could not be developed by the team it was assigned to, all the other teams produced very good modules for English lexical and morphological analysis, English and Arabic parsing/generation, and Arabic morphological generation. These modules are being refined and reused in a more serious attempt to build an English-to-Arabic machine translation system.

The neural network-based transfer module which could not be developed by one team of students, has been taken over by an MSc student who intends to study the problem more deeply.

We do hope that in the near future we will be able to test a first prototype of our Arabic MT system.

In terms of the software engineering aspect of the experiment, the major problem faced was to explain all the needed background to the students. This included giving them the right background about machine translation as well as more technical details about lexical and morphological analyses, parsing and generation, neural networks, etc. This had to be done quick enough so that the students would not waste any of the short time span of the course (less than 4 months). This, of course, put a lot of pressure on the students as well as the instructor since I had to run common meetings for all the teams as well as meetings with each team to probe deeper in their problems and findings. It has nonetheless been a very enriching teaching and research and development experience.

Acknowledgements

My sincerest thanks go to the 10 students in the course: Amel Abdalrahman, Hiba Al-Azzam, Manal Alkhulaqi, Sumaya Ba Omran, Ola El-Gharabli, Riham El-Hadari, Tamara Issa, Amani Kaafarani, Shahad Sanaseeri, and Alia Shihab. All the students were very enthusiastic about the goal of building some tools that would ultimately serve the Arabic language. They spent a lot of time and effort to take up the challenge. They deserve all the admiration.

References

[1] Tae Yoo, Corporate Vice President, CISCO Systems, lecture at the conf. on ICT for Sustainable Development, Abu Dhabi, 12-13 Dec. 2004.

[2] Guessoum, A. and R. Zantout: 2000, 'Arabic Machine Translation: A Strategic Choice for the Arab World', KSU Computer and Info. Sciences Journal, Volume 12, pages 117 – 144.

[3] Hutchins, John and Harold L. Somers: 1992, ' An Introduction to Machine Translation', Academic Press.

[4] Jurafsky, D. and J.H. Martin: 2000, '*Speech Processing and Language Processing*', Prentice Hall.

[5] Amzi Prolog is available as a free version at www.amzi.com .