Computational Support for Learners of Arabic

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCE

2012

MAJDA MAJEED AL-LIABI

School of Computer Science
Contents

List of Figures .................................................................................................................. 6
List of Tables .................................................................................................................. 8
Abstract ......................................................................................................................... 9
Declaration ...................................................................................................................... 10
Copyright Statement ...................................................................................................... 11
Acknowledgments .......................................................................................................... 12
Transliteration Table ...................................................................................................... 13
Abbreviations and Acronyms list .................................................................................. 14

Chapter 1: Introduction ................................................................................................. 16
  1.1 An Overview of Problems for Language Learners ................................................. 16
  1.2 Problems with Analysing Arabic ............................................................................. 18
  1.3 Background on CALL ............................................................................................... 18
  1.4 Research Questions, Hypotheses and Contributions ............................................. 21
  1.5 Structure of the System ............................................................................................ 23
    1.5.1 System Outline ................................................................................................ 26
  1.6 Thesis Outline .......................................................................................................... 27

Chapter 2 : Language Learning and CALL ................................................................. 29
  2.1 Introduction .............................................................................................................. 29
  2.2 Language Learning and Teaching ............................................................................ 29
    2.2.1 Language Learning ............................................................................................. 30
    2.2.2 Language Teaching ............................................................................................. 30
    2.2.3 Assessment Methods for Learning a Language .................................................. 32
      2.2.3.1 Formative Assessment ................................................................................ 32
      2.2.3.2 Summative Assessment .............................................................................. 32
      2.2.3.2.1 Norm-referenced Assessment .................................................................. 33
      2.2.3.2.2 Criterion-referenced Assessment ............................................................ 33
  2.3 Computers in Language Learning and Teaching .................................................... 33
  2.4 Computational Tools for Teachers and Learners ................................................... 35
    2.4.1 Teacher’s Role .................................................................................................... 36
    2.4.2 Student’s Role .................................................................................................... 37
  2.5 The role of CALL in Learning Pedagogy ................................................................. 39
  2.6 The Role of ICALL in Grammar Learning ............................................................... 42
  2.7 CALL Overview ...................................................................................................... 44
  2.8 The Concept of ICALL ............................................................................................. 45
    2.8.1 Intelligent Tutoring Systems .............................................................................. 45
    2.8.2 Expert Systems .................................................................................................. 47
    2.8.3 Automated Speech Recognition ....................................................................... 48
    2.8.4 Machine Translation .......................................................................................... 49
    2.8.5 User Modelling: Adapting Interaction ............................................................... 50
    2.8.6 Natural Language Processing .......................................................................... 50
  2.9 Applications of CALL .............................................................................................. 51
Chapter 3: Characteristic with Arabic ................................................................. 61
  3.1 Introduction ................................................................................................. 61
  3.2 Properties of Arabic .................................................................................... 61
  3.3 Arabic Morphology: A descriptive Analysis ............................................... 62
    3.3.1 Introduction .......................................................................................... 62
    3.3.2 Arabic Word Pattern ............................................................................. 63
    3.3.3 Arabic Word Structure ........................................................................ 65
    3.3.4 Non-Linear Morphology ....................................................................... 67
      3.3.4.1 McCarthy’s Approach ................................................................. 68
    3.3.5 Arabic Lexical Categories ...................................................................... 70
      3.3.5.1 Nouns .......................................................................................... 71
      3.3.5.2 Pronouns ...................................................................................... 71
      3.3.5.3 Adjectives ..................................................................................... 75
      3.3.5.4 Verbs .......................................................................................... 76
    3.3.5.5 Particles ........................................................................................... 80
  3.4 Arabic Syntax: A Descriptive Analysis ....................................................... 82
    3.4.1 Introduction .......................................................................................... 82
    3.4.2 Arabic Sentence Structure ..................................................................... 82
      3.4.2.1 Equational Sentences ................................................................ 83
      3.4.2.2 Verbal Sentences ......................................................................... 85
    3.4.3 Word Order and Agreement ................................................................. 86
    3.4.4 External Governors .............................................................................. 88
  3.5 Case study and Error Treatment ................................................................. 90
  3.6 Summary ...................................................................................................... 91

Chapter 4: Classification of Learner Errors ....................................................... 92
  4.1 Introduction ................................................................................................ 92
  4.2 A Historical Perspective of Error Analysis ................................................. 93
  4.3 Error Diagnosis ......................................................................................... 93
  4.4 Error Classification ................................................................................... 95
    4.4.1 Error Description and Taylor’s Classification .................................... 95
    4.4.2 Classification by Tokuda and Chen .................................................... 97
    4.4.3 Dulay et al. classification ..................................................................... 98
      4.4.3.1 Linguistic Category Classification .............................................. 99
      4.4.3.2 The Surface Structure Taxonomy .............................................. 100
Chapter 5: Arabic Morphosyntax within Parasite Framework

5.1 Introduction .................................................................................112
5.2 Parasite Overview ..................................................................................112
5.3 Survey of Arabic Morphological Analysers ..........................................................115
5.3.1 Two-level Morphology ...........................................................................115
  5.3.1.1 Beesley’s Approach ...........................................................................116
5.3.2 Finite-state Morphology ........................................................................117
  5.3.2.1 Kay’s Approach ................................................................................117
5.3.3 Multi Two-level FST Morphology ..............................................................117
  5.3.3.1 Kiraz’s Approach .................................................................................117
5.4 Treatment of Arabic Morphology in Parasite ......................................................119
5.4.1 Arabic Morphological Analyser .................................................................120
5.4.2 Root and Lexical Entry ..............................................................................120
  5.4.2.1 Lexical Trie and Slot-filler ..................................................................121
5.4.2.2 Categorial Grammar - CG ..................................................................123
  5.4.2.3 Affix Entry ........................................................................................127
5.5 Morphological Analysis: Roots and Affixes ......................................................128
5.5.1 Verbal Morphology ................................................................................128
5.5.2 Nominal Morphology ..............................................................................131
  5.5.2.1 Case Study on Broken Noun ................................................................132
5.6 Syntax ..............................................................................................133
5.6.1 Introduction ........................................................................................133
5.6.2 Head-Driven Phrase Structure Grammar- HPSG .........................................134
  5.6.2.1 Background Study on HPSG ..............................................................134
5.6.3 Subcategorisation in HPSG ....................................................................135
5.6.4 HPSG Schemata ....................................................................................139
5.7 Parasite Framework ..............................................................................143
5.7.1 Constraints on Arguments ......................................................................144
5.7.2 Canonical Order and Theta Roles ..............................................................145
5.7.3 Non-Canonical Word Order .....................................................................146
  5.7.3.1 Constraints on Out-of-Position Items ..................................................148
5.7.4 Internal / External Syntax ......................................................................149
  5.7.4.1 Constraints on Verbless Sentences ......................................................150
  5.7.4.2 Relative Clauses ...............................................................................152
5.8 Summary ..........................................................................................153
List of Figures

Figure 1.1: The block diagram of the current system ................................................................. 24
Figure 1.2: Flowchart of the Parasite system .............................................................................. 25
Figure 2.1: A diagram of the role of an ITS in an integrated CALL system ......................... 46
Figure 3.1: Arabic word formation ............................................................................................. 67
Figure 3.2: CV derivational processing approach ..................................................................... 69
Figure 5.1: Multi-level template Arabic morphology scheme ............................................... 118
Figure 5.2: Root trie representation ......................................................................................... 122
Figure 6.1: aRaBCALL login screen ......................................................................................... 162
Figure 6.2: aRaBCALL input screen ......................................................................................... 162
Figure 6.3: aRaBCALL output screen: with no errors ............................................................. 163
Figure 6.4: aRaBCALL output screen - first level feedback ................................................... 164
Figure 6.5: aRaBCALL output screen – second level feedback ............................................. 164
Figure 6.6: aRaBCALL output screen – third level feedback ................................................ 165
Figure 6.7: aRaBCALL output: no errors ................................................................................. 165
Figure 6.8: General diagram of the current system ................................................................. 170
Figure 6.9: aRaBCALL output screen - first level of feedback for the word أكتبون Ouikotibunuwa ................................................................. 181
Figure 6.10: aRaBCALL output screen - second level of feedback for the word أكتبون Ouikotibunuwa ................................................................. 182
Figure 6.11: aRaBCALL output screen - third level of feedback for the word أكتبون Ouikotibunuwa ................................................................. 183
Figure 7.1: Scores for verb exercise ......................................................................................... 204
Figure 7.2: Network for all legal options included equally ..................................................... 207
Figure 7.3: The pattern of the best user interaction path ......................................................... 209
Figure 7.4: The pattern of the worst user interaction path ....................................................... 209
Figure 7.5: The pattern of the ideal user of the system ................................................................. 210

Figure 7.6: The path of sensible use of the system ................................................................. 211

Figure 7.7: The path of a poor use of the system ................................................................. 211

Figure B.1: aRaBCALL displaying first level feedback on misused word order constraints .... 226

Figure B.2: aRaBCALL displaying second level of feedback on misuse of word order constraints .............................................................................................................................. 227

Figure B.3: aRaBCALL displaying correct use of word order constraints ......................... 227

Figure B.4: aRaBCALL displaying first level feedback on misused tense ......................... 228

Figure B.5: aRaBCALL displaying second level feedback on subject verb agreement .... 228

Figure B.6: aRaBCALL displaying first level feedback on misused negation complementiser patterns ُلا لا َّلا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا ل
## List of Tables

Table 2.1: Part of Kayes and Burnetts’ view on the assessment of learner’s Key skills ............ 31
Table 3.1: Example of a derived form for the root خَرَمُون ders ............................................. 65
Table 3.2: CV non-linear morphology approach ........................................................................ 69
Table 3.3: Personal nominative pronouns ..................................................................................... 72
Table 3.4: Arabic attached pronouns ............................................................................................. 73
Table 3.5: Arabic demonstrative pronouns ..................................................................................... 74
Table 3.6: Pattern of adjectival noun طويل Twyl ‘tall’ .............................................................. 75
Table 3.7: All Arabic adjectival colours ......................................................................................... 76
Table 3.8: Arabic derived colours .................................................................................................. 76
Table 3.9: Representing a perfective verb mood .......................................................................... 77
Table 3.10: Representing an imperfective verb mood ................................................................. 78
Table 3.11: Surface forms for the present tense verb pattern كتب ktb ............................................. 79
Table 3.12: Kinds of Arabic prepositions ....................................................................................... 80
Table 3.13: Arabic complementiser with its property ..................................................................... 81
Table 3.14: Arabic Particles with their features ............................................................................ 90
Table 4.1: Types of word formation errors ................................................................................... 107
Table 4.2: Types of local feature mismatches .............................................................................. 108
Table 4.3: Types of word order errors ........................................................................................ 110
Table 5.1: Solid nouns diacritic patterns ...................................................................................... 132
Table 6.1: The expected verb morphological errors ................................................................. 178
Table 7.1: Student’s score for the question about making mistakes ............................................. 202
Table 7.2: A detailed group point of view on subjective assessments .......................................... 203
Table 7.3: A representation of each exercise rank ...................................................................... 204
Table 7.4: The representation of the activities’ path ................................................................. 207
Abstract

This thesis documents the use of Natural Language Processing (NLP) in Computer Assisted Language Learning (CALL) and its contribution to the learning experience of students studying Arabic as a foreign language. The goal of this project is to build an Intelligent Computer Assisted Language Learning (ICALL) system that provides computational assistance to learners of Arabic by teaching grammar, producing homework and issuing students with immediate feedback. To produce this system we use the Parasite system, which produces morphological, syntactic and semantic analysis of textual input, and extend it to provide error detection and diagnosis. The methodology we adopt involves relaxing constraints on unification so that correct information contained in a badly formed sentence may still be used to obtain a coherent overall analysis. We look at a range of errors, drawn from experience with learners at various levels, covering word internal problems (addition of inappropriate affixes, failure to apply morphotactic rules properly) and problems with relations between words (local constraints on features, and word order problems). As feedback is an important factor in learning, we look into different types of feedback that can be used to evaluate which is the most appropriate for the aim of our system.
Declaration

I hereby declare that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.
Copyright Statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://www.campus.manchester.ac.uk/medialibrary/policies/intellectual-property.pdf), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University’s policy on presentation of Theses.
Acknowledgments

First and Foremost, I would like to express my heartfelt gratitude to my supervisor Professor Allan Ramsay for his support during my PhD. I would like to thank him for his patience, encouragement, and immense knowledge. His guidance helped me throughout the research and the writing of this thesis. I could not have imagined having a better advisor for my PhD study. I truly will miss our weekly meetings.

I would like to dedicate this thesis to the souls of my Father and Mother for their love, endless support and encouragement.

I miss them always, may Allah bless them.

I must thank my beloved husband, Nizar Adnan, who spent days and nights with our grown up children and taking care of all the tasks that should have been shared by me as a mother and a wife. I wish to thank him for helping me get through the difficult times, and for all the emotional support, encouragement, and care he provided. I am really in debt to you and appreciate everything you have done sincerely. Your love, understanding and patience sustained me during all stages of this PhD.

I would like to offer my regards and blessings to my sons Mustafa and Ahmed and my daughter Abir for all their respect during the completion of the thesis. I would also like to especially thank my son in law, Rizwan, for his assistance in the writing style of this thesis.

I must show my gratitude to my colleagues Maytham Alabbas and Sohel Vhora, who gave me kind assistance when I required their opinion. My appreciation also goes to my brothers and sisters for their constant love and support.

O Allah! Make useful for me what you have taught me and teach me knowledge that will be useful to me. O Allah I ask you for the understanding of the prophets and the memory of the messengers and those nearest to you.
## Transliteration Table

<table>
<thead>
<tr>
<th>Arabic Letter</th>
<th>Trans.</th>
<th>Name of letter</th>
<th>Arabic Letter</th>
<th>Trans.</th>
<th>Name of letter</th>
</tr>
</thead>
<tbody>
<tr>
<td>ء</td>
<td>HAMZA</td>
<td>ع</td>
<td>ء</td>
<td>ZAH</td>
<td>ء</td>
</tr>
<tr>
<td>أ</td>
<td>ALEF WITH MADDA ABOVE</td>
<td>أ</td>
<td>غ</td>
<td>E</td>
<td>AIN</td>
</tr>
<tr>
<td>أ</td>
<td>ALEF WITH HAMZA ABOVE</td>
<td>أ</td>
<td>غ</td>
<td>E</td>
<td>AIN</td>
</tr>
<tr>
<td>و</td>
<td>WAW WITH HAMZA ABOVE</td>
<td>و</td>
<td>ع</td>
<td>E</td>
<td>AIN</td>
</tr>
<tr>
<td>I</td>
<td>ALEF WITH HAMZA BELOW</td>
<td>I</td>
<td>ف</td>
<td>f</td>
<td>FEH</td>
</tr>
<tr>
<td>ئ</td>
<td>YEH WITH HAMZA ABOVE</td>
<td>ئ</td>
<td>ق</td>
<td>q</td>
<td>QAF</td>
</tr>
<tr>
<td>أ</td>
<td>ALEF</td>
<td>أ</td>
<td>ك</td>
<td>k</td>
<td>KAF</td>
</tr>
<tr>
<td>ب</td>
<td>BEH</td>
<td>ب</td>
<td>ل</td>
<td>l</td>
<td>LAM</td>
</tr>
<tr>
<td>ة</td>
<td>TEH MARBUTA</td>
<td>ة</td>
<td>م</td>
<td>m</td>
<td>MEEM</td>
</tr>
<tr>
<td>ت</td>
<td>THE</td>
<td>ت</td>
<td>ن</td>
<td>n</td>
<td>NOON</td>
</tr>
<tr>
<td>ث</td>
<td>THEH</td>
<td>ث</td>
<td>ه</td>
<td>h</td>
<td>HEH</td>
</tr>
<tr>
<td>ج</td>
<td>JEEM</td>
<td>ج</td>
<td>و</td>
<td>w</td>
<td>WAW</td>
</tr>
<tr>
<td>ح</td>
<td>HAH</td>
<td>ح</td>
<td>ي</td>
<td>y</td>
<td>YEH</td>
</tr>
<tr>
<td>خ</td>
<td>KHAH</td>
<td>خ</td>
<td>ح</td>
<td>f</td>
<td>ALEF MAQSURA</td>
</tr>
<tr>
<td>د</td>
<td>DAL</td>
<td>د</td>
<td>ق</td>
<td>q</td>
<td>QAF</td>
</tr>
<tr>
<td>ذ</td>
<td>THAL</td>
<td>ذ</td>
<td>ط</td>
<td>k</td>
<td>KASRATAN</td>
</tr>
<tr>
<td>ئ</td>
<td>ZAIN</td>
<td>ئ</td>
<td>a</td>
<td>a</td>
<td>FATHA</td>
</tr>
<tr>
<td>س</td>
<td>SEEN</td>
<td>س</td>
<td>u</td>
<td>u</td>
<td>DAMMA</td>
</tr>
<tr>
<td>ش</td>
<td>SHEEN</td>
<td>ش</td>
<td>i</td>
<td>i</td>
<td>KASRA</td>
</tr>
<tr>
<td>ص</td>
<td>SAD</td>
<td>ص</td>
<td>~</td>
<td>~</td>
<td>SHADDA</td>
</tr>
<tr>
<td>ض</td>
<td>DAD</td>
<td>ض</td>
<td>o</td>
<td>o</td>
<td>SUKUN</td>
</tr>
<tr>
<td>ط</td>
<td>TAH</td>
<td>ط</td>
<td>ت</td>
<td>t</td>
<td>TAH</td>
</tr>
</tbody>
</table>

1 The standard Buckwalter transliteration is used for converting Arabic script to the Roman alphabet (Buckwalter, 2002). The transliteration scheme is available at: http://www.qamus.org/transliteration.htm
## Abbreviations and Acronyms List

<table>
<thead>
<tr>
<th>Abbreviation and Acronyms</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>1&lt;sup&gt;st&lt;/sup&gt;</td>
<td>First</td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt;</td>
<td>Second</td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt;</td>
<td>Third</td>
</tr>
<tr>
<td>ACTPAST</td>
<td>Active past</td>
</tr>
<tr>
<td>ACTPRES</td>
<td>Active present</td>
</tr>
<tr>
<td>ADJ</td>
<td>Adjective</td>
</tr>
<tr>
<td>AI</td>
<td>Artificial Intelligent</td>
</tr>
<tr>
<td>ASR</td>
<td>Automated Speech Recognition</td>
</tr>
<tr>
<td>CG</td>
<td>Categorial Grammar</td>
</tr>
<tr>
<td>CA</td>
<td>Classical Arabic</td>
</tr>
<tr>
<td>CAA</td>
<td>Computer Aided Assessment</td>
</tr>
<tr>
<td>CALL</td>
<td>Computer Assisted Language Learning</td>
</tr>
<tr>
<td>CL</td>
<td>Computational Linguistics</td>
</tr>
<tr>
<td>CM</td>
<td>Categorial Morphology</td>
</tr>
<tr>
<td>CRT</td>
<td>Criterion-Referenced Testing</td>
</tr>
<tr>
<td>CSR</td>
<td>Continuous Speech Recognition</td>
</tr>
<tr>
<td>DSR</td>
<td>Discrete Speech Recognition</td>
</tr>
<tr>
<td>EH</td>
<td>Error Heurism</td>
</tr>
<tr>
<td>EM</td>
<td>Expert Module</td>
</tr>
<tr>
<td>Fem</td>
<td>Feminine</td>
</tr>
<tr>
<td>FL</td>
<td>Foreign language</td>
</tr>
<tr>
<td>GPSG</td>
<td>Generalized Phrase Structure Grammar</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphic User Interface</td>
</tr>
<tr>
<td>HPSG</td>
<td>Head-driven Phrase Structure Grammar</td>
</tr>
<tr>
<td>ICALL</td>
<td>Intelligent Computer Assisted Language Learning</td>
</tr>
<tr>
<td>ID</td>
<td>Immediate Dominance</td>
</tr>
<tr>
<td>IDP</td>
<td>Immediate Dominance Principle</td>
</tr>
<tr>
<td>IM</td>
<td>Interface module</td>
</tr>
<tr>
<td>ITS</td>
<td>Intelligent Tutoring Systems</td>
</tr>
<tr>
<td>LE</td>
<td>Linguistic Engine</td>
</tr>
<tr>
<td>LFG</td>
<td>Lexical Functional Grammar</td>
</tr>
<tr>
<td>LM</td>
<td>Learner model</td>
</tr>
<tr>
<td>LP</td>
<td>Linear Precedence</td>
</tr>
<tr>
<td>Mas</td>
<td>Masculine</td>
</tr>
<tr>
<td>MSA</td>
<td>Modern Standard Arabic</td>
</tr>
<tr>
<td>MT</td>
<td>Machine Translation</td>
</tr>
<tr>
<td>N</td>
<td>Noun</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>Nom</td>
<td>Nominative</td>
</tr>
<tr>
<td>NRT</td>
<td>Norm-Referenced Test</td>
</tr>
<tr>
<td>NP</td>
<td>Noun phrase</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>OBJ</td>
<td>Object</td>
</tr>
<tr>
<td>OVS</td>
<td>Object-Verb-Subject</td>
</tr>
<tr>
<td>Per</td>
<td>Person</td>
</tr>
<tr>
<td>Pl</td>
<td>Plural</td>
</tr>
<tr>
<td>POS</td>
<td>Part Of Speech</td>
</tr>
<tr>
<td>POSS</td>
<td>Possessive</td>
</tr>
<tr>
<td>PP</td>
<td>Prepositional phrase</td>
</tr>
<tr>
<td>PREP</td>
<td>Preposition</td>
</tr>
<tr>
<td>Pres</td>
<td>Present</td>
</tr>
<tr>
<td>Pro</td>
<td>Pronoun</td>
</tr>
<tr>
<td>PSG</td>
<td>Phrase Structure Grammar</td>
</tr>
<tr>
<td>PSV Past</td>
<td>Passive past</td>
</tr>
<tr>
<td>PSVPRES</td>
<td>Passive present</td>
</tr>
<tr>
<td>S</td>
<td>Sentence</td>
</tr>
<tr>
<td>SLA</td>
<td>Second Language Acquisition</td>
</tr>
<tr>
<td>Sing</td>
<td>Singular</td>
</tr>
<tr>
<td>SL</td>
<td>Source language</td>
</tr>
<tr>
<td>SMT</td>
<td>Statistical Machine Translation</td>
</tr>
<tr>
<td>ST</td>
<td>Source text</td>
</tr>
<tr>
<td>SUBJ</td>
<td>Subject</td>
</tr>
<tr>
<td>SVO</td>
<td>Subject-Verb-Object</td>
</tr>
<tr>
<td>TL</td>
<td>Target language</td>
</tr>
<tr>
<td>TM</td>
<td>Tutor Module</td>
</tr>
<tr>
<td>TT</td>
<td>Target text</td>
</tr>
<tr>
<td>V</td>
<td>Verb</td>
</tr>
<tr>
<td>VOS</td>
<td>Verb-Object-Subject</td>
</tr>
<tr>
<td>VP</td>
<td>Verb phrase</td>
</tr>
<tr>
<td>VSO</td>
<td>Verb-Subject-Object</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 An Overview of Problems for Language Learners

Teaching Arabic as a foreign language in the UK has enabled us to recognise problems that learners have with various aspects of learning a new language. We have always wondered whether or not we could provide automated assistance for students, using some form of computational support tool.

In order to achieve our goal, first of all, it had to be decided what type of support students require. From many years of personal experience as a tutor of Arabic in English colleges, I am aware that many students find Arabic hard to learn across all aspects of the language, including pronunciation, vocabulary, writing and grammar. One problematic area of study for a learner of Arabic is that of grammar skills, with many problems arising from confusion between the first and second languages (Dawood, 2009). In the next chapter, we provide a detailed specification of Arabic in order to bring out those aspects of the language that make it difficult to learn.

Exploiting CALL tools can relatively extend effective learning of new languages (Hubbard and Levy, 2006), by an amount which depends on how much emphasis is placed on the role of this technology. The use of CALL in language teaching and learning has become a standard and expected part of the curriculum (ibid), ranging from an aid to both teacher and learner by complementing the teacher’s instruction, to programs that can completely replace the teacher.

In the last thirty years, there have been different projects using of a large variety of complex techniques within ICALL (Intelligent CALL), such as NLP (Natural language processing) and student modelling in CALL (Heift and Schulze, 2007). The aim of NLP
is to let computers extract information from human language. The ultimate goal of NLP is to design and build computers that are able to manipulate language by analysing, understanding and generating it, as humans do naturally, though there are also useful but less ambitious NLP systems.

Our research focuses on the design of an ICALL tool which, as one of its stages, acts as an analyser which is able to identify errors in student input. In theory, this means that students receive feedback immediately and the teacher does not have to spend as much time carrying out the hard task of marking. One of the outputs of this project has been a parser that detects errors identified as potential problems for students. Although the parser itself is not directly suitable for use by students, it could be used within a fully developed CALL tool. It can locate items that cannot be parsed successfully and diagnose the specific error, and error type, that has been made. It also generates appropriate error messages that integrate relevant information into a feedback message.

The proposed model consists of two main aspects: a constraint-based approach to Arabic morphosyntax and a mechanism for relaxing constraints. Our tool tests a predefined set of exercises, containing grammar questions based on specific elements of the syllabus for all levels of learning, from beginners to advanced. The student is presented with a sentence which may contain errors. If it is not correct the student has to enter the correct answer whilst also receiving hints from the system if needed. The system does not generate the correct answer, however, but determines whether it is correct or not, with an explanation.

In an edit box, the student can also to write their own sentence in order to test if it is correct or not. The system checks the answer and gives appropriate feedback. Furthermore, the system can also be used for transliteration for learners who have difficulty in writing the Arabic letters, to help them recognise and learn these letters.
1.2 Problems with Analysing Arabic

In order to provide an ICALL tool, we need a program that can analyse Arabic morphosyntax. This is more difficult to construct than for English. There is, indeed, no agreement among researchers on the basic sentence structures in Arabic (see for instance the discussion of verbal and nominal sentences in section 3.4.2). There is a wide gap between Arab and Western grammarians in their attempts to describe Arabic syntactic structures, with each applying a different set of criteria to characterise the same phenomena, and many aspects of Arabic are not investigated satisfactorily, such as topicalisation, agreement and long-distance dependencies. Arabic has 60 characters for letters, diacritics, numbers and punctuation marks. In addition to this, Arabic letters need to be connected together in a cursive manner, which sometimes leads to changes in appearance depending on the context in which they arise.

It is generally agreed that Arabic has two standard word orders: verb initial sentences and sentences without a verb at all. However, a wide variety of other orders can occur, and it is in the analysis of these that disagreements among linguists arise.

Arabic words are built from roots and not from stems. In written Modern Standard Arabic diacritics, which help in making the pronunciation of words with the same form, are usually omitted.

Arabic uses clitics, which are morphemes with the syntactic characteristics of words (Crystal, 1980). In Arabic, clitics can attach to either the start or end of words, so that fairly complex sentences can be composed of a single word. Clitics in Arabic include many prepositions and particles, coordinating conjunctions, a class of pronouns, and the definite article.

1.3 Background on CALL

Researchers need to gain an understanding of various academic aspects of a project and to seek and read information based on this understanding. In a continuously evolving field, it is important to take current developments into consideration to ensure research
remains relevant. To assist in this, a number of research papers detailing similar studies and explaining the mechanisms of dealing with CALL, were considered, gaining beneficial information.

Over the last two decades, the use of computer programs to support students in their learning has been the subject of much research and is not a new concept in the area of language learning (Barr, 2004). The potential of using technology in order to teach and provide explicit guidance has been well documented in mechanical areas of language learning, such as vocabulary acquisition and grammar learning (ibid).

The development of user-adaptive systems has seen a clear increase in recent years, particularly for online learning environments. However, there has been very little work done on student modelling in any area of CALL, not just for parser-based systems (Heift and Schulze, 2003). This is due to the limited understanding of foreign language learning processes (Chapelle, 1997). With the huge amount of multilingual technical documentation produced by companies nowadays, grammar and controlled language checking is becoming an application that is high in demand (Crysman et al., 2008).

Davies (2006) discusses a number of categories. These do not add up to a classification as they address several different dimensions of CALL, with a number of partially overlapping classes. Some of his categories, for instance, address the delivery of CALL whereas some underline the typology and the role of what they give to us. Davies distinguishes CALL systems into the following groups:

(i) Traditional CALL
(ii) Communicative CALL
(iii) Multimedia CALL
(iv) Web-based CALL
(v) CALL authoring programs
(vi) CAA (Computer Aided Assessment)
(vii) ICALL
Chapter 1: Introduction

(viii) Whole class teaching and CALL

Most CALL packages combine features making it possible to assign them to more than one group. With traditional CALL, programs are based more on the stimulus response theory which has been the dominant theory in regards to how people learn. Communicative CALL “has always been a degree of controversy about the teacher-centered, drill-based approach to CALL” and has been affected by the newer theories of learning (ibid: 461). This approach is also characterized by the use of concordance programs in the language classrooms. This type of CALL has become more popular over recent years due to the increased number of people being able to access the Internet. Multimedia CALL was a breakthrough as the system would now be able to consist of a combination of any text, still images, sound and motion video. The learner would be able to respond to these new developments, rather than solely typing at the keyboard, and be able to point and click or drag and drop with a mouse and make use of a microphone.

Due to the massive use of the Internet, we have noticed the tendency of CALL practitioners to use the Web both as a source of information as well as a means to deliver CALL materials to learners. CALL authoring programs enable the speedy development of CALL materials but are too complex for most language teachers. However, there are some specifically designed CALL authoring programs that are well suited to the capabilities of the average language teacher who has no knowledge in computer programming. Computer-aided assessment (CAA) covers a range of assessment procedures and is a rapidly developing subset of CALL. CAA refers to any instance in which some aspect of computer technology is used as part of the assessment process such as tests that take a similar form to CALL exercises (multiple choice, gap-filling format). It can also be used for marking of students’ word-processed writing and using a database or spreadsheet to keep a record of student’s grades. Whole-class teaching was first introduced when it was usual for a teacher to focus the attention of the class on a TV monitor, when they were asked to respond to what appeared on the screen. It is now
undergoing a revival through the use of the white-board, a touch sensitive projection screen, with specific CALL software designed for this type of use.

Recently, some commercial CALL tools for learning Arabic have been available. RosettaStone®, for instance, is an electronic language software tool for learning Arabic as a foreign language, and is suitable for beginners. Free language material is an important type of online teaching material that contains all four of the skills (reading, writing speaking and listening) needed in teaching strategies.

1.4 Research Questions, Hypotheses and Contributions

In this thesis, we aim to answer to the following questions:

i. What types of error can be diagnosed by softening the constraints in the rule-based description of Arabic morphosyntax?

ii. What types of feedback do students find useful?

iii. Can student behaviour be characterised in ways that allow guidance to be provided?

In order to answer these questions and ensure that the issues are addressed, the following are carried out in the course of this thesis:

1. Extension of the existing fine-grained computational description of Arabic morphology and syntax. The work in this thesis is based on Ramsay and Mansour's (2001) treatment of Arabic morphosyntax. However, Ramsay and Mansour concentrated on the commoner and more regular aspects of Arabic morphosyntax, while most problems that learners encounter relate to irregular constructions. To deal with these constructions we will need to extend Ramsay and Mansour’s original treatment.

2. An investigation into the use of soft parsing techniques to provide students with feedback on morphological and syntactic errors. Soft parsing is a well-known technique, but has not yet been extensively applied to Arabic. Soft parsing is a
means of differentiating between input that is very nearly acceptable and input that cannot go unchallenged. As soft parsing involves use of a constraint-based grammar, a very wide range of phenomena, from obvious things like agreement failure to complex constraints on word order, can be dealt with using this single mechanism.

3. Development of a diagnostic tool for morpho-syntactic errors that uses constraint relaxation techniques. Soft parsing allows you to detect and diagnose errors, but for such diagnoses to be useful to a learner this information has to be presented to them in some suitable format. We therefore need to map 'error codes' to meaningful diagnostic messages which contain general information about the kind of error that has been made, and which also show exactly how this diagnosis maps onto the current example.

4. Provision of multilevel feedback and feedback messages that describe the problem and the underlying linguistic generalisation.

5. Evaluation of student work on using our system, aRaBCALL.

These five contributions can be generalised into two major hypotheses that we will aim to prove throughout the rest of this thesis, which are:

- Providing students with multi-level guidance about grammatical errors would be useful as a support to independent learning

- Relaxing constraints in a constraint-based grammar would enable us to diagnose a wide-variety of errors

When it comes to evaluation of the research questions, we need to look at them specifically. To evaluate question i), we must find out which errors students are typically reported to make and how they can be handled simply by softening constraints. Section 6.5 looks into this. For the evaluation of question ii), we must look at the various levels of feedback there are and investigate when each student needs to use each level of feedback, which is shown in section 6.3. The use of questionnaires and interviews helps to identify which feedback they use, and which specific topics they require feedback for in order to overcome their difficulties. This subjective method gives us a clear view
about the types of feedback they find useful. In order to evaluate question iii), we use the log file, which is an activity network that records the students’ activity, to look at students’ behaviour. This method gives us an idea about how students use the system and helps us to determine how guidance can be provided.

1.5 Structure of the System

This work is to build a morpho-syntactic representation which is part of the Parasite system\(^2\) and to introduce it into a classroom situation (Ramsay and Mansour, 2001). The methodology we use is to consider firstly morphology, as the Arabic language has a rich and difficult morphology, and secondly which syntactic formalism to adopt in the implementation of our tool. Students use the tool by correcting a sentence which has been displayed to them and which may contain errors. If the version that the student submits still contains errors, then the system flags the words that are seen to be problematic and invites the student to try again. The system outline is shown in the diagram below:

\(^2\) Standing for PrAgmatics ReAsoning about the Speaker’s InTEnsions; the manual for the system is available at (http://www.cs.manchester.ac.uk/~ramsay/SYSTEMS/manual.pdf)
This research will make use of Parasite, which is an existing NLP system. The current system is used to analyse the input text by carrying out lexical look-up and spelling rules and then using HPSG-like (Head-driven Phrase Structure Grammar) rules of combination for morpho-syntactic analysis, including constraint techniques applied to soft parsing.

Parasite makes use of the notion of a filter, which is a type of constraint that controls fine-grained constraints on word order. It is a part of the treatment that is responsible for allowing arguments of the head to move from canonical positions through a set of filters. The dynamic constraints are the case of marked orders governed by a set of constraints.
Internal and external components are key parts in the Parasite system framework. In chapter five, a detailed representation of the other Parasite components is discussed, such as argument, SUBCAT list, Arabic canonical order and the filter. Soft parsing with diagnostic labels is presented using a web based GUI to help receive the final feedback.

The Parasite system currently has dictionaries and morpho-syntactic rules for a number of languages such as English, Arabic, Persian, German and Somali. Parasite consists of lexical look up (including morphology), parsing, semantic analysis and reasoning about the speaker’s goals. Our research focuses on the structural analysis and employs it to develop our ICALL tool for Arabic. Figure 1.2 shows the general outline of the Parasite system.

Figure 1.2: Flowchart of the Parasite system

The use of the Parasite system and other Computational Linguistic (CL) applications are advantageous to teaching and learning organisations, as they realise benefits of
harnessing the potential of the Internet and computers. The purpose of the current work is to employ CL techniques to investigate certain linguistic features of standard Arabic in order to help learners to study both the morphological and syntactic aspects of the language. Parasite has been designed and implemented in order to become a computational morphological tool for the use of Arabic language learners.

As mentioned in the previous section, Arabic has a number of characteristics that make it different from other languages. Two features in particular make Arabic difficult to process automatically. Firstly, Arabic morphosyntax is very complicated, a problem which is made more complex by the fact that the short vowels are not written. Secondly, Arabic allows free-word order. The parsing presentation of a word, for new learners of the language, or even for the native speaker, can alter the whole meaning or context, for example, if the subject is positioned at the far end of the sentence.

### 1.5.1 System Outline

Our work aims to allow teachers to set different levels of exercises for checking the learning of grammar and have them marked automatically. This benefits teachers, by relieving them of the burden of marking their students’ work; and it benefits students, by providing immediate and detailed comments on their work.

aRaBCALL provides a feedback technique for morphosyntactic and structural errors in students’ work, which requires some features. Firstly, we provide a tool to engage the student with grammar exercises, such as the use of adjectives and verbs. Secondly, we diagnose syntactic and morphological errors automatically so that the teacher does not need to correct them.

Parasite is the NLP toolkit which is used to support our tool. The program is written in such a way that it can give learners feedback on morphosyntactic and structural errors in their work. We aim to produce a tool to diagnose morphological and syntactic errors in student texts.

As the aRaBCALL system uses an NLP based parser, it contains two major components: a Linguistic Engine (LE) for carrying out morpho-syntactic analysis, and a
Web-based Front-End (Ramsay and Mansour, 2001). LE consists of a learning exercise model which should be controlled by the tutor. The text can be written either in the Roman alphabet using the Buckwalter transliteration or in the Arabic alphabet using either an Arabic keyboard or a virtual keyboard provided in the Front-End. To employ the parser as ICALL and run it in an Internet browser, we need a Web-based Front-End which allows input using either Arabic characters or transliteration, and which provides feedback in a clear and easy to read form.

1.6 Thesis Outline

Based on what is covered above, this dissertation is comprised of the following chapters.

**Chapter 2** is devoted to an overview of problems of language learners with an emphasis on learning Arabic as a foreign language. Since language learning and teaching is the main focus of this research it is explained in much detail, as are computational learning techniques. The concept of CALL and ICALL, and commercial CALL applications such as RosettaStone® and ArabicPod, are also explained in this chapter.

**Chapter 3** explains the characteristics of Arabic. This chapter describes Arabic from a linguistic perspective and deals with its morphology and syntax. Nouns, verbs and particles, as well as ambiguity are also discussed. We explain the errors that can occur in morphology and syntax, discuss problems faced by English learners of Arabic.

**Chapter 4** is a detailed study of types of error, concentrating on morphosyntactic errors, and deals with the pedagogical view of the different classification of learner errors. We investigate the historical perspective on learner errors and focus on English speaking learners of Arabic for error data. We also discuss error diagnosis, classification of learner errors with an emphasis on morphological and syntactic errors, error taxonomies and the classifications by James (1998), Tokuda and Chen (2004); Taylor (1986; 1998).

**Chapter 5** provides a computational analysis of Arabic by discussing in detail theoretical aspects of the lexical and syntactic description of Arabic for computational implementation from the perspective of Categorial Grammar (CG) and Head-Driven
Phrase Structure Grammar (HPSG). Following this, we explain Parasite’s components and move on to discuss how Parasite analyses the morphosyntax of natural language.

Chapter 6 discusses our observations regard to our learners’ errors. This involves common learning methods, such as homework tasks that we usually use when teaching the language. The different behaviours of NLP-based parsers are explained in detail. The chapter begins by examining how the system treats the syntactic properties of verbs, adjectives, nouns and their complements. This leads to discussion of the architecture of the system, in turn leading to discussion of the interface engine, which that uses soft parsing techniques adapted to support to the use of a parser as a CALL tool.

Chapter 7 describes evaluation and feedback. Look into different evaluation techniques, the goals we want to achieve and the feedback received from the evaluation.

Chapter 8 concludes and presents information with regard to future research.
Chapter 2

Language Learning and CALL

2.1 Introduction

This chapter is aimed at presenting an overview of the principles of language learning and language teaching. Identifying problems on acquiring a second language will also be explained. The specific characteristic of the difficulties when Arabic is the target language for learning and teaching will be discussed. Also, this chapter will address the concept and the notion of the ICALL and computational tools for language learning and language teaching which make use of NLP techniques. Furthermore, there will be a particular explanation on Parser-based CALL for Arabic.

2.2 Language Learning and Teaching

Arabic is one of the foreign languages that have been taught in the UK for decades. Whilst teaching Arabic as a foreign language for more than eighteen years in the UK, I have become aware of the many problems that students and teachers have with various aspects of the language that they are dealing with.

English learners of the Arabic language have always had specific difficulties in different areas when learning the language. A major problem area is the use of the verb tense, followed by issues related to the gender marker and finally, issues related to grammatical problems (Khalil, 2002; Soudi et al., 2007). These difficulties will be investigated through all linguistic elements that commonly lead to student error. In this section, we will consider the specific reasons for these difficulties. The following
sections will distinguish between the meaning of language learning and the teaching in the field of language acquisition.

2.2.1 Language Learning

We can define learning strategies as the actions that individuals are able to take to accomplish learning goals. Language learning is a part of cognitive science and the definition of it is the process by which humans acquire the capacity to recognise, produce and use words to understand and communicate (Tomasello, 2008). The process involves the picking up of diverse capacities including syntax, phonetics, and an extensive vocabulary (Chomsky, 1957; 1965). Also, learning a language is something that can be achieved by anyone successfully, but requires a matter of a few years with the need for formal lessons. Another way of learning a language, as Fodor (1983) theorises, is Modularity. Fodor explains that modularity arises from the need to develop specialised modules for perception, which are then integrated into a collection of communicating elements. The full theory proposed that the output of these modules is monitored by a central processor, which then chooses an action, causing further modules to generate expressed behaviour from these decisions (Bryson, 2005).

2.2.2 Language Teaching

In recent years, the phrase ‘teaching for learning’ has become increasingly dominant. This is a result of the growing awareness that teaching and learning must be considered as complementary activities. The aim of teachers always is to be sure that the learner is placed at the core of the structure, content and delivery of the teaching lesson.

The principle motivation for teaching languages or studying on a language course is to achieve a development of skills and to integrate independent learning, which will develop the language learning process.

The teacher has two major responsibilities of equal importance determining the success of a learning experience for individuals and groups. One responsibility is the
planning of well-structured, stimulating and effectively taught lessons, with coherent, appropriate aims and learning outcomes and relevant assessment strategies. The other is the management of the learning environment, so that learners can achieve (Armitage et al., 2007).

In contrast, it is important to assess learners’ key skills development and independent study skills. The following are important key skills that the learner has to be encouraged to be aware of, mentioned by Kayes and Burnett (2006):

| Self-analysis of needs                      |
| Awareness and development of personal and interpersonal learning strategies |
| Awareness of motivational type             |
| Awareness of strength and weakness         |
| Familiarity with range of available resources and materials |

**Table 2.1: Part of Kayes and Burnetts’ view on the assessment of learner’s Key skills**

All the above entities are the issues that the teacher and the learner should bear in mind. The teaching and learning strategy is not a one-way process; it is a multi-way process (Kannan, 2009).

There are wide ranges of the standard language skills that are important for learners to be aware of when learning a foreign language (Wenden, 1991; O’Malley and Chamot, 1990).

- Speaking skills:- which include pronunciation
- Listening skills: Listening for gist, and listening for detail
- Reading skills: reading for gist, reading for detail
Chapter 2: Language Learning and CALL

- Writing skills: spelling, punctuation, planning style, and form
- Strategies for learning vocabulary
- Strategies for learning grammar

2.2.3 Assessment Methods for Learning a Language

As teachers, we always focus on the outcome of the learning process which leads to the development of learning skills in the student causing it to be a more effective and efficient learning process. A way to focus on the learning process is by assessments or tests. They are used for different purposes in the learning process at different points. There are four main ways to conduct an assessment or a test. Before looking at them, it is worth considering the contexts in which we intend to deploy them. “Assessment is variously formal or informal: at one extreme a degree finals paper, at the other, very generalized judgments made by a teacher as he or she observes an individual or group. It is formative or summative, its prime purpose being either to support student learning or, on the other hand, to gather information about it” (Armitage et al., 2007: 149). The following are the brief explanation of the types of tests or assessments.

2.2.3.1 Formative Assessment

This is a method designed to establish how much progress a student is making during their learning journey with a view to giving feedback to them (Atkins et al., 1993). For example essays, can be used for formative assessment, with comments being provided by the teacher to help the student learn from mistakes to improve their work. Formative assessment is often referred to as an assessment of the process of independent learning.

2.2.3.2 Summative Assessment

This kind of assessment method is designed to test what a student has achieved at the end of a unit or course. The grade or the final mark is awarded (ibid) contrasting with
formative assessment i.e. the assessment of process. Summative assessment aims to determine learning up to the point of the test (Knight and Yorke, 2003).

2.2.3.2.1 Norm-referenced Assessment

A norm-referenced test (NRT) refers to a test in which the score of an examiner reported is compared to a distribution of scores of other examiners in a reference group. The standards for comparisons are called norms, and the group that the norms are obtained in is called a norm group. Norm-referenced testing varies from criterion-referenced testing (CRT). The main purpose of CRTs is to determine how students perform compared to a predetermined performance level or outcome. NRT information is provided on how well a student is performing in comparison to other students, whereas CRTs provide information on what they know and what they can do (Kanjee, 2010).

2.2.3.2.2 Criterion-referenced Assessment

A Criterion Reference Assessment is when a criterion for passing a test has been predetermined. The most popular Criterion Reference Assessment is the driving test where the criteria for passing have been determined prior to the test being taken. This causes you to know whether or not you have passed or failed immediately without having to find out later (Reddy, 2004). The achievement of the student performance in this kind of assessment is graded relating to pre-specified standards. Theoretically, all students could achieve the maximum possible grade or may fail to meet the standards set (Atkins et al., 1993).

2.3 Computers in Language Learning and Teaching

Computers have been used in education since the early 60s. They have raised the curiosity and the interest of many instructors, language instructors being among them. Indeed, as we have seen, computers, with their characteristics and inherent advantages,
had something to offer that could potentially benefit both the learner and the teacher as well, and could also contribute to educational innovations.

McCarty (1993) describes the computer within the language learning and teaching environment as no longer a machine but a specific educational tool, devoted to the learning that models the learning/teaching situation which will often consist of simulation of the learning/teaching reality. McCarty’s four key concepts (language, program, model and simulation) have helped us to define the use of the computer in context, a language learning/teaching context which is referred to as CALL (ibid).

Armitage et al. (2007) stated that teaching and learning resources should be used to promote student learning. One of the key advantages of using a computer is that the presentation can be made completely systematic, so that users know exactly what to expect, files are easily accessible and can be adapted and amended if and when necessary.

Ahmad et al. (1985: 4) describes how the computer “can repeat an activity with none of the errors which easily arise from repetition by humans”, as well as this “can work at the speed best suited to the individual” (ibid: 6). The computer is also known for its “ability to control lesson delivery, to scrupulously collect data on its use and to analyze the results” (Stevens, 1992: 24). It often, for example, acts as a language trainer and a resource manager whereas the teacher acts as the language expert, being the native speaker.

The computer in language learning and teaching performs well at dedicated tasks. It has been pointed out by Hubbard (1992: 47) that it is “effective as a text manipulator, as an answer judger, as an animator, as a timer, as a record keeper, as a controller of audio and video devices”. Since a computer is good at repetitive tasks, it will perform effectively in practice types of scenarios and activities. Considering all the dedicated tasks the computer can do, the advantages will be numerous. Although these advantages would probably apply mostly in learning and teaching situations, some may be more relevant to the specific aspects of the language learning and teaching situation, as we will observe in the final part of the following discussion. The advantages of computers
in language learning and teaching that benefit both the language learner and the language instructor by supporting the teaching processing include flexibility, versatility, economy, control, variety, popularity and viability (Stevens, 1992; Ahmad et al., 1985).

There are many benefits in language learning of using computers such as autonomy, motivation and individuality (Ahmad et al., 1985; Borraccino, 1994). The advantages are, firstly, individualization, which gives rise to a ‘potential for personalized instruction’ since the “branching capacity [discrimination ability] can be made sensible to the learner pace, pattern of responses” (Ahmad et al., 1985: 6). Secondly, the computer promotes learner autonomy as it offers an opportunity for teacher-independent feedback and learning (Borraccino, 1994). Additionally, there are also many more advantages for the use of computers in supporting language teaching, such as interaction, patience, neutrality, rapidity, effectiveness and realism (Ahmad et al., 1985; Mitterer et al., 1990).

### 2.4 Computational Tools for Teachers and Learners

The computer as a teacher and the computer as a student are two approaches which can be used to classify research into CALL for foreign language learning. The computer as a teacher approach is when some computer programs are able to be used as a tool to teach without necessarily providing any other language learning materials. This approach empowers the learner, who is usually a non-native speaker, to understand or learn the language. The computer as a learner approach involves the process of finding the right answer. Both approaches have familiar features, for example, they take the focus away from the teacher being a portrayer of knowledge to giving students a learning experience that is as realistic as possible, where the student plays a pivotal role. In addition, the approaches are inclined to emphasise fluency over accuracy, thus, allowing students to take risks in using activities that are aimed more at students and will also allow them to work together, rather than compete against each other. The computer provides an atmosphere of less dependency of students for teachers and gives them more freedom to experiment on their own with natural languages in a natural or semi-natural setting.
2.4.1 Teacher’s Role

These days, the presentation of new material is the first stage in a typical language learning environment. The practice stage follows, which in the majority of cases takes the form of the student recycling the target language in a modified context. The possession of the material is finalised, perhaps at a later date, through less guided, summative type activities. This teaching approach is just as appropriate in the traditional classroom as it is in the CALL environment. It is important to emphasise that a CALL environment may include the presence of the teacher. Occasionally, it is assumed that CALL equates to what is known as ‘open or self-access’ mode, this is far from the truth. In fact, students regularly carry out these activities in the presence of a teacher allowing the teacher to identify and set appropriate remedial work using the automatically generated feedback. The incorporation of CALL into a foreign language programme can lead to great concern among language teachers (Thelmadatter, 1995). In contrast, CALL changes the role of the teacher, as claimed regularly by researchers, but it does not abandon the need for a teacher altogether. Instead of forwarding knowledge to students and being the main theme of students’ attention, teachers become guides as they create the activities students are to do and help the students complete their tasks. In other words, instead of being directly involved in students’ performance, the teacher interacts with students mostly to help them with difficulties they may have in using the target language, such as the vocabulary and grammar they use to interact with the computer or other students (Stepp-Greany, 2002; Domingo, 2007). Removal of a strong teacher presence has been shown to lead to better quality and larger quantity of communication such as more fluidity, increased use of complex sentences and a greater sharing of students’ personal experiences (Stepp-Greany, 2002). Nonetheless, teacher presence is still very essential to students when carrying out CALL activities. Teachers should be adept enough with the resources to be used so they can anticipate technical problems and limitations (Domingo, 2007). Students need the teacher motivation and supportive presence in CALL environments. Review sessions are required from students in order to emphasize and reinforce what the students have learned. To ensure this is incorporated, students are encouraged to participate and praise is often offered to those who perform.
In conclusion, “the computer can best assist teachers in its role as a creator, an informant, an environment” (Ahmad et al., 1985: 8). Moreover, “whether it is used as a tool or as a partner, the computer consists of an attempt to diversify the initial inputs in the teaching and learning process. It allows the teacher to get rid of partitioning and to introduce multi or trans-disciplinary initiatives while giving a larger initiative to the learner”. (Demaiziere et al, 1992: 111)

2.4.2 Student’s Role

The most important goal of the computer in language learning is certainly to help the learner enhance their linguistic skills, as is the goal of other supports he could receive in their language learning environment; such as support provided by the tutor or a peer, support provided by use of traditional tools such as books and audio-visual materials. What makes the computer such a unique support lies in the fact that it can occupy in turns and/or combine the role of one of:

- A tutor (to provide the learner with explanation, feedback and remediation);
- A trainer (for the learner to practise their language skills in repetitive/drill-and-practice types of activities);
- A resource (for the learner to consult);
- A tool (to support the learner in performing the language, such as speaking and writing);
- A model (for the learner to imitate via automatic provision of spoken/written examples, of correct answers);

The computer will typically support language learning by proposing to the learner an individualized, interactive and resourceful exchange (Hamel, 2008). Indeed, as Stevens (1992: 21) suggests, its role is to “provide learners with a rich environment of functional, communicative and interactive material in a given language”. “The computer will do so while also giving the learner the opportunity to work at a learning task in simple, achievable stages”. (Hudson, 1984: 1)
The responsibilities of the computer are either to present information, to guide learners or to encourage discovery (McCarty, 1993), as a result of which, another of its important roles is the ‘promotion of student autonomy’ in learning (Stevens, 1992). Indeed, the computer’s role in language learning is “not only as a remedial tool, as a sophisticated device for nothing more than programmed learning, but it also is to afford students the opportunity to have greater input to all aspects of curriculum development and management. It should also provide learners with increased opportunities to take responsibility for their own learning”. (Naiman, 1993: 99)

With regards to Levy’s (1997) tutor-tool framework for the computer’s roles in language teaching and learning, the dichotomy between the tutor and the tool is clear cut. The tutor evaluates while the tool does not; the tutor judges while the tool remains neutral; the tutor is directive while the tool is not. In this framework, the double-assumption is that the tutor-tool is used when the teacher is temporarily absent and is always in self-access mode. On the other hand, with the teacher, the role is as guidance is not available via the computer program therefore will have to be provided by the teacher through the use of computer. The computer belongs to the tutor’s role since in both cases, the teacher is absent and the learning is done by access.

In order to use CALL effectively, students are required to amend their expectations of their participation in the class. Rather than absorbing information passively, students must take in new information and negotiate meaning through collaboration and interaction with someone other than the teacher, such as a colleague. Students must also learn to understand new information and experiences on their own terms. However, the reorganization of students and teachers’ considerations due to the use of technology have enabled students with less ability to become more active participants in the class because the class interaction is not limited to that directed by the teacher (Stepp-Greany, 2002). Furthermore, shy students can feel more comfortable in their own students'-centred environment. This will increase their self-esteem and at the same time their knowledge will be improving. If students perform a collaborative project they will endeavour to perform it within a set time frame.
2.5 The role of CALL in Learning Pedagogy

Chamot (2004) defined learning strategies as the actions and thoughts that individuals use to accomplish a learning goal. The history of the language teaching profession in the last century confirms the impression that improvements in language teaching will come about as a result of improvements in the quality of methods and that ultimately the language teaching method will be developed. Nunan (1991: 87) supports this view: “For many years it was believed that linguistic or psycholinguistic theory would uncover the secrets of second language acquisition and that then the problem of how to teach a second language would be solved once and for all”. As this has not always proved to be the case, we will briefly outline in the next section, some of the theories that informed current teaching practice.

Over the years a number of methods which will be considered have emerged, such as the audio-lingual approach, the humanistic, cognitive code learning and the second language traditions. According to (ibid), the most enduring method is the audio-lingual approach, by borrowing behaviorist principles from contemporary psychology to explain the mechanism by which language acquisition takes place. The reasoning was that successful learning occurred when the subject’s response (to an initial stimulus) was reinforced by some other event, known as a ‘reinforce’. This suggested that (language) learning was predominantly a question of habit formation, which required instant reward, i.e. approval. Observation had shown that knowledge of the meta-language was not a requirement of speaking the language itself. In addition, the concept of a general grammar was rejected of language specific, self-referential analysis of structure. Certain teaching methods came to the fore, as a result of this theoretical framework. The use of the target language (preferably authentic material) as used by natives became more important than what was contained in grammar books.

According to McDonough (1986; 2006), at the same time cognitive psychologists were identifying different kinds of learning which were not based on habit formation. They observed that the subject did not apparently require the presence of reinforces in order to learn, but there were several kinds of behavior that might easily be described as
Chomsky's work on transformational grammar attempted to further undermine the theoretical base that underpinned the audiolingual approach, transformational grammar raised serious doubts about the Structuralists’ model of language (Chomsky, 1965). It was no longer justifiable that a language was learnt by experiencing and reinforcing a determined collection of structures, as this view did not account for the possibility of humans to produce novel and innovative sentences that they have never heard before by applying certain rules. An example which is often quoted in this respect is that of children who become aware of grammar rules (a long time after they have begun to speak) and attempt to use them inaccurately or incorrectly although they had previously produced correct forms through imitation. Cognitive code learning proposed a method for learners to manipulate the material they were presented with to solve problems or to be creative with it. Learners were required to be pro-active if they were going to recognize the rules of the language and then, more importantly, be capable of generating their own sentences.

As a result the audio-lingual was replaced by a more communicative approach, which is much more functional and group-based than its predecessor. The main route to acquisition, in terms of classroom practice, was not seen as reinforcement but generation based on a set of grammar rules. Consequently, it became necessary to talk about grammar, as learners were encouraged to discover for themselves and then apply the rules. One consequence of this approach was the positive view taken of mistakes. Whereas previously the audio-lingual approach viewed incorrect responses as being deviant, cognitive code learning believed that they were a natural part of the learning process. The incorrect application of a valid grammatical rule could be interpreted as an error. At this stage the detection and diagnosis of student errors began to assume a central role.

Psychologists have turned their attention to the study of the impact of affective and emotional stimuli on subjects. Language teachers who adopted the Humanistic Method began to examine the attitudinal state of learners to attempt to explain the differing degrees of success. Factors included points such as the students' level of interest and motivation as well as the type of environment in which the learning was taking place.
Shift in power was taking place, away from the teacher and towards the student. Roles were redefined as courses became learner-centered due to the learner becoming the main focus. The teacher, on the other hand, became the provider and the guide, no longer teaching but facilitating.

Krashen (2003) attempted to incorporate these changes in attitude to language teaching and learning into the five hypotheses that underpin the Second Language Acquisition (SLA). Hypothesis one distinguished between language acquisition, which is based on the process that enables (monolingual) children to become successful communicators and language learning, which is a general description of the various traditional ways in which foreign languages had been studied. Learning highlighted the extensive knowledge and use of grammar, but during acquisition it was assumed that the grammar rules were being used unconsciously. The two are related symbiotically for in order for learners to speak a foreign language, acquisition and learning must both take place.

Adopting a natural order of acquisition is suggested in hypothesis two. In Krashen’s third hypothesis, the introduction of the monitor principle explains the process of learning in much more detail. The process of learning builds up a conscious understanding of the rules governing the language, the rules act as a monitor ensuring adherence to the learnt principles. Therefore, the model of foreign language output proposes the spontaneous generation of the TL as a result of the acquisition process and the subsequent interaction of the monitor. Hypothesis four emphasises very clearly the importance of using comprehensible input with learners. The fifth and final hypothesis, the active filter, explains the differing levels of success achieved by different students. Similarly to the ideas put forward in the Humanistic approach, the active filter proposes that the learner’s attitude and other emotional factors have a direct impact on levels of achievement; the higher the filter, the greater the difficulty in achieving success. In terms of activities, problem-solving games, physical response activities and multimedia role-play in the target language were all encouraged. At the micro level, the aim was to engage students’ emotive side through interviews and questionnaires, etc., in order to lower the filter described in hypothesis five.
To summarise the methodological overview, the audio-lingual theory had a central belief in a simplistic, behaviourist approach to language learning. This resulted in the idea that repetition was good, as was immediate reinforcement. Subsequently, it was superseded by a more communicative view of the learning process. Consequently, reinforcement was seen in terms of praise or support and not necessarily immediately after the event. The view of error diagnosis as an inherent part of the actual learning process also established itself. Subsequently the Humanistic approach and eventually SLA methodology came to the front. The former emphasized the need to provide a more learner-friendly environment in the classroom. The latter, while adopting much of what had gone, also established the link between acquisition and learning. It also stressed the intelligibility of instruction and the importance of logical ordering in the presentation of new material (Krashen, 2008). Today, many language teachers adopt a general SLA approach, due to the eclectic nature of Krashen's approach.

2.6 The Role of ICALL in Grammar Learning

The notion of grammar in the context of second language learning is complex and is best broken down into inter-related grammar systems (Schulze, 2001). Over the past few decades, a number of researchers have concentrated on teaching grammar while teaching a foreign language (Nassaji and Fotos, 2010).

Barr’s conclusion regarding the link between technology and grammar found that “technology made a difference as it changed students’ attitudes towards learning grammar” (Barr, 2008: 112). From this, we are able to determine that the use of a computer will play a vital role for the students, as it will help to provide them with a positive outlook when learning grammar. However, the integration of CALL tools into the Natural Language learning is one of the most effective ways of using computers for learning grammar.

Many studies in the past decade have examined different approaches to teaching grammar (Long and Doughty, 2011; Ellis, 1994; 2005; Nassaji and Fotos, 2010; Ferris, 2004; Tschichold, 2003; Tschichold et al., 1997). There are many effective ways of
teaching second language grammar that these studies have shown; these can be from the
teacher role directed constructivist tutorials (Hauck et al., 1994) to cooperative group
work (Nassaji and Fotos, 2010) and individual study with textbooks (Ellis, 2005; 2006).
As these studies indicate that alternative effective ways to learn grammar do exist, there
is an ever increasing array of options teachers have in order to meet the needs of the
students.

Computer-based grammar instruction offers many potential benefits; however, it has
not received the same amount of attention as communicative CALL. Although it is
currently impossible for the computer to engage learners in peer-to-peer collaboration, it
is possible for CALL to provide rich input in the form of integrated multimedia
programs and provide explicit grammar explanations that can be viewed and reviewed at
the learner’s own pace (Jones, 2006). Reagan et al. (1994) analysis of research on the
use of multimedia or software instruction to teach a variety of subjects found out that, in
most cases, electronic instruction reduces learning time by 30% compared to the
traditional learning (Reagan et al., 1994). It was also demonstrated that some features
such as learner interactivity and learner control improved outcomes in achievement.

Natural language parsers take written language as their input and produce a formal
representation of the syntactic and sometimes semantic structure of this input (Schulze,
2003).

The main difference between parser based CALL and conventional CALL, is that in
parser-based CALL the student has relatively free way and can write a potentially huge
variety of sentences whereas conventional CALL cannot. Adding knowledge to CALL is
to move from CALL to ICALL. ICALL thus, allows the practice of production skills,
which require recalling and constructing, not just recognizing words and structures
(Holland et al, 1993). However, simultaneously, parsing imposes certain limitations.

parsers tend to concentrate on the syntax of the textual input, therefore, ICALL may
actually subvert a principal goal of language pedagogy, which is of communicating
meaning rather than producing the right forms (ibid). Therefore, a parser-based CALL
application can play a useful role in detecting many of the morpho-syntactic errors that
constitute a high percentage of learner errors in freely produced texts. Nevertheless, the
fact remains that “a second limitation of ICALL is that parsers are not foolproof. Because no parser today can accurately analyse all the syntax of a language, false acceptance and false alarms are inevitable”. (ibid, 1993: 33)

The advantage of parser based CALL is certainly the potential to avoid a scenario in which the language learner is just flooded with unnecessary stimuli, but also to provide the learner with information, help and support that is really wanted at a given point in the learning process, such as relevant, contextualised and informative error feedback.

### 2.7 CALL Overview

The development of CALL can be seen in three stages that correspond to the technological changes over a range of approximately 40 years. The three stages are defined as communicative, behaviourist, and most recently, integrative CALL. Integrative CALL emphasises authentic contexts and genuine language use in meaningful ways. It also aims to integrate the four language skills: reading, writing, listening and speaking (Levy, 1997). CALL integrates computer technology with the language teaching field as a support to the presentation, reinforcement and assessment of learning materials including a significant interactive and effective component (Davies, 2006).

A variety of techniques in CALL have been formed from artificial intelligent (AI), resulting with the classification of ICALL. The ICALL area has a range of different tasks and many roles which make the computer behave as a teacher who teaches, gives instructions and assesses, and also as a learner, who studies, understands and then substitutes the language teacher. NLP systems as well as techniques of AI are used by many CALL programs to achieve interactivity. Our system attempts to benefit student learning when students complete an exercise outside the classroom. This is done by providing feedback immediately, meaning that teachers do not have to spend as much time carrying out the laborious task of marking.
2.8 The Concept of ICALL

The field of AI deals with understanding of human intelligence such as behaviour, learning and adaptation by trying to model it in machines. Matthews defines ICALL as a mix of AI techniques and CALL (Matthews, 1992). Schulze believes that ICALL should be called parser-based CALL rather than ICALL. This is because it is said the ‘intelligence’ lies in the use of parsing (Schulze, 2001). The following are some AI techniques that have been used in the field of CALL to add to the quality of such programs. Since the main focus of this research is the application of NLP in CALL, other techniques will be mentioned briefly; however, when we come to NLP, we will provide more detailed information.

2.8.1 Intelligent Tutoring Systems

Intelligent Tutoring Systems (ITS) are computer-assisted instructional systems carrying out dialogues with students and use student responses to assess the learning process (Lehman et al., 2008). The systems usually consist of many modules. In the domain of language learning, ITS will contain an expert system to store the domain knowledge, a learner model describing the learner’s knowledge, a tutor module representing the tutoring strategies and learning goals, and a graphical user interface acting as a bridge between the user and the system (Mayo et al., 2001). In figure 2.1 below, an example of an ITS diagram is shown.
The following are the many specific features of ITS:

- ITS are dynamic systems, which means that an ITS’s domain knowledge of the recipient’s knowledge will be adapted as learning progresses.

- One of the main features of ITS, for the teacher, is that they are competent in the subject it teaches (Lehman et al., 2008). In this case, the system will be able to resolve the problems which were caused by the learner.
• ITS are modular systems (Swartz and Yazdani, 1992) consisting of the Expert Module (EM), the Learner Module (LM), which describes the learner’s knowledge, the Tutor Module (TM) which represents the tutor strategies, learning goals, and the Interface Module (IM), which acts as a bridge between the user and the system (Mayo et al., 2001). These four modules are separated from each other but on the other hand they can communicate with each other. In other words, the representation of the information contained in each module depends on the description of each other module.

2.8.2 Expert Systems

Expert systems constitute a subfield of Artificial Intelligence technology which attempts to clarify uncertainties where normally one or more human experts would need to be consulted or provide an answer to a problem. An important embodiment within a computer of knowledge based components is the expert system, as it is skilled in such a way that it can provide intelligent advice and act intelligently when deciding a processing function. This knowledge is used for the analysis of the student’s interaction with the system including the submitted answers and allows providing a more detailed feedback than in traditional systems (Bos and Plassche, 1994; Harrington, 1994; Soria, 1997). For example, the Spengels system (Bos and Plassche, 1994) is equipped with knowledge about the spelling of verb forms. The student can either learn the spelling and conjugation rules from scratch or practice them by filling in gapped sentences. Error specific feedback can be given at each stage of the learning process. “Expert systems for language learning store a large body of knowledge about language learning such as typical mistakes, learning strategies, questions and answers etc” (Gamper and Knapp, 2002).
2.8.3 Automated Speech Recognition

Automated Speech Recognition (ASR) is a technology that allows users to use speech when entering information instead of punching the numbers keys or letters on a keypad. ASR can also be defined as a software or hardware system that is capable of translating the spoken word or a conversation into another usable representation. Since only a limited number of answers have to be expected from the learner, therefore types of ASR in a CALL context is quite fast and reliable (Myers, 2000). There are two distinguished ASR in the field of ICALL: Discrete Speech Recognition (DSR) and Continuous Speech Recognition (CSR).

- **DSR** system requires one word to be spoken at a time, meaning that there will be a brief pause between each word by the user/speaker. In relation to ICALL, DSR is used in fluency training (Holland et al., 1999) and is also used when teaching pronunciation. This kind of recognition also allows analysis of single patterns which are known to the system (Levinson et al., 1993; Murray, 1995; Myers, 2000; Eskenazi 1999).

- **CSR** depends on the assumption that speaking is a naturally acquired skill which does not require much practice from users (Ortmanns et al., 1997). The use of CSR allows speech to be very fast (ibid). It is used predominantly with natural phrases and sentences without breaks between words (Paliwal, 1996). Speech recognition systems provide a natural means of access to information (Mostow and Aist, 1999), but the system works on the expectation of a certain amount of input (Menzel et al., 2001). The Conversim’s system is a CSR system which allows users to have face to face dialogues in real time with virtual characters. The system prompts the user to ask three relevant questions continuously. A video instructor helps to ensure correct pronunciation. The user must rely on their language skills, intuition and experience to evaluate whether the character lies or not (Gamper and Knapp, 2002).
2.8.4 Machine Translation

Translation through the computer is known as Machine Translation (MT). MT is the application of language sciences and the computer to the development of systems answering practical needs. It can be used for the production of translations with or without human assistance. MT is one of the CALL systems because of the use of traditional translation strategy in language learning. MT is used to translate text from one natural language to the other in CALL. It excludes computer-based translation tools which support translators by accessing on-line dictionaries, transmission and reception of texts. MT is divided into Rule-Based Machine Translation (RBMT), and statistical Machine Translation (SMT), which have both seen vast progress over the last decades but which have developed very much in isolation. The reason for this is that different communities have played a role in each of the technology’s development (Koehn, 2010).

A translation system can be used to produce an initial text version of the target language, on which the student should continue to work (Anderson, 1995; Anderson and Elloumi, 2008; Torre, 1999). There are some MT systems which have been developed from ICALL systems. These systems are easier to integrate into the learning process (Tokuda and Chen, 2001; Shei and Pain, 2001). MT systems show the learner texts with typical key language patterns, which have to be translated by the learner. The system compares these translations with correct and erroneous model sentences and provides error corrections, and comment. There is a quite unusual approach which has been used in a study which aims at investigating whether a handheld translation machine used in daily life could be helpful for language teaching (Myers, 2007). The evaluation result shows that the student vocabulary has been increased and pronunciation has been improved. Generally speaking, MT is a non-CALL system, however, in such ways it has been deployed as a CALL system.
2.8.5 User Modelling: Adapting Interaction

As Murphy and McTear (1997) point out, many packages of CALL systems would be lacking when it comes to support the learner with individualised teaching and flexible and sufficient feedback.

User modelling is one of the AI techniques used in CALL to support student learning. It makes use of a structured model which represents on a student’s mental and behaviour activities. Gamper and Knapp (2002:333) define user modelling in language learning as a “...viewable user model is included recording the user’s steps and mistakes”. The teacher and the student after the learning session have to test the learning procedure systematically (Burton and Brown, 1976; Jones and Virvou, 1991; Bos and Plassche, 1994; Harrington, 1994; Cook and Kay, 1994; DeSmedt and Kommissar, 1995; Frederiksen et al., 1995).

The main feature of the student modelling component is the processing of a diagnosis error. The capability of individualised error diagnosis is important for students, who can benefit from modified advice to their problems. Typically, not all learners can learn from traditional teaching strategies. However, new media in combination with adaptive technologies allow the development of teaching aids. For example, remedial exercises are offered depending on the error frequency (Menzel et al., 2001; Tokuda and Chen, 2001). Adaptive sequencing is also used to (Murphy and McTear, 1997; Dalby and Kewley-Port, 2007; Holland, 2005; Cooley, 2001; Shei and Pain, 2001) produce a language lesson adapted to the specific needs of the learner and adaptive annotations are provided in Gamper and Knapp (2002).

2.8.6 Natural Language Processing

Natural Language Processing (NLP), is one of the central themes of language engineering which allows a computer to interact with the human language. To achieve this, a computer requires a program normally referred to as a natural language parser. “A natural language system must use a considerable amount of knowledge about the structure of the language itself, including what the words are, how words combine to
form sentences, what the words mean, how word meanings contribute to sentence meanings, and so on” (Allen, 1995: 9). Allen mentioned different forms of knowledge related to NLP such as: morphological knowledge, syntactic knowledge, semantic knowledge, pragmatic knowledge, discourse knowledge, and world knowledge. Most of the previous types of knowledge are very important to the main field of NLP which is important in application of CALL.

2.9 Applications of CALL

As mentioned previously, CALL is an approach to language teaching that involves the provision of automated tools for the support of learners. The scope of this part is on the uses of CALL in four skills and in both scientific and commercial applications.

2.9.1 Use of Non Commercial CALL for the Four Skills

A number of studies (Levy, 1997; Schulze, 2003; Kayes and Burnett, 2006) concerned with the effectiveness of CALL in developing the four skills of language learning (listening, speaking, reading and writing). This is mostly because of the high demand of computer technology in the current state of learning. Most reading and listening software is based on drills (Domingo, 2007). Gains in writing skills have not been as impressive as computers cannot assess this well (Stepp-Greany, 2002). Using limited CALL technology for the development of speaking and writing abilities has gained much attention from researchers. There are some successes in using CALL, in particular computer-mediated communication, to help with speaking skills closely linked to ‘communicative competence’ and provide controlled interactive speaking practice outside the classroom (Ehsani and Eva, 1998).

The use of computer-based methods for learning language skills is to assist students in the development of their linguistic skills by providing more language-learning opportunities than human teaching methods. Machine learning, a branch of AI, allows computers to evolve behaviours based on data as it is scientifically disciplined with the
design and development of algorithms. A learner can take advantage of the data to capture characteristics for the interest of their unknown underlying probability distribution. Data can be understood as examples that portray relations between observed variables. The machine learning concept is the study and computer modelling of the learning process in which they are represented (Carbonell et al, 1983). There is a large amount research on tools or computer-based methods to teach English as a foreign language. Wagner et al. (2009) have worked with a classifier for English language learners, dealing mostly with closed-class words such as prepositions and determiners. Chodorow et al. (2007) present an approach to preposition error detection which is a model based on a maximum entropy classifier trained on a set of contextual features, together with a rule-based filter. Gamon et al. (2009) uses a complex system including a decision tree and a language model for both preposition and determiner errors, while Yi et al. (2008) use a Web count-based system to correct determiner errors.

2.9.2 Commercial Applications
There are a few existing commercial software tools on how to learn Arabic without the help of a teacher online. However, these software tools do not cover the grammatical features required by the EdExcel Arabic syllabus, which is the most widely used in British schools. Examples of such commercial applications are RosettaStone® Language Software and ArabicPod Software.

2.9.2.1 RosettaStone® Language Software
RosettaStone® language software is a good product to learn Arabic as a foreign language for beginners as it covers the day to day aspects of the curriculum. It has many different products for each type of level of the learners of Arabic. It covers the basic grammar and vocabulary to basic sentence structure. However, RosettaStone® is not
appropriate to our learners as it does not cover most of the syllabus for teaching Arabic as a foreign language within the British curriculum\(^3\).

### 2.9.2.2 ArabicPod Software

ArabicPod is a supportive commercial software tool for learning Arabic. It contains the lesson transcripts and learning materials. Lessons are well-organised and tackle selected difficult aspects of learning the Arabic language, such as the letters with difficulty in pronunciation and cultural colloquial expressions. The dialect taught on this podcast is ‘common colloquial’ Arabic, common within all Arabic dialects, as well as Standard Arabic. However, this software is very expensive and does not fulfil our teaching objectives as its topics are not related to the EdExcel exam board syllabus which is what we have tried to cover most of in our project.

### 2.9.2.3 Arabic Tutor

It is the first commercial multimedia Arabic teaching software for beginners, which works with any Microsoft Windows™ operating system (Levy, 1997). It uses an easy to use point and click characteristic sound. It allows the user to listen to pronunciations, record the voice and compare it to native pronunciations. The users are also able to watch animated illustrations of how letters are written. It teaches the user how to join letters into words and how to read words and sentences. There are basic grammars included for a variety of reading comprehensions tasks.

### 2.10 NLP and CALL

In the historical overview of CALL, there were two key elements that were incorporated: to allow students to work at their own speed and to receive immediate

\(^3\)Evidence of British syllabus available at [http://www.dhahranbritish.com/Syllabi_2012/edexcel%20syllabuses/Arabic2AR01.pdf](http://www.dhahranbritish.com/Syllabi_2012/edexcel%20syllabuses/Arabic2AR01.pdf)
feedback from their work. The majority of feedback and analysis was by means of text matching and comparison with model answers. However, a number of researchers came to the conclusion that more intelligence, such as the ability to process lexicon, syntax, semantics and pragmatics, was needed in CALL systems (Pusack, 1983; Brierley and Kemble, 1991). As a result, ICALL was established which consisted of language-processing software and student modelling tools which at present is the point of interest.

This led to the relationship between CALL and NLP, in which NLP is currently integrated in most of the CALL software engineering processes. This integration is usually in the form of a parser and dictionary used alongside a text manipulation CALL tool. Students normally enter data in the target language and the parser is used to analyse the morpho-syntax of the input using soft parsing techniques. In the following sections, I will discuss some of the intelligence and NLP techniques used in CALL and examine a number of systems which have used the parser-based approach to CALL.

NLP should be implemented in the programs that the majority of students will come into contact with and not just programs designed for specialists. Computers are most successful at parsing the kinds of language, short simple sentences, that the majority of learners are most likely to encounter. It is thus, important to design the tasks that students perform so that they involve fairly short sentences even when exploring advanced grammatical concepts.

2.11 Parser-Based CALL

CL and CALL are interdependent fields of study. An important aim of CL is to program computers to understand input sentences and generate appropriate output sentences in response, for translation and direct communication with computers in which the computer understands and generates natural language. CL shares the values of theoretical linguistics, but with the additional aim of characterizing a language with the computational applications in mind (Levy, 1997). “Natural language parsers take written language as their input and produce a formal representation of the syntactic and sometimes semantic structure of this input”. (Schulze, 2001: 14) Schulze (ibid) admits
that there are few CALL applications incorporating NLP techniques. “...this is due to the fact that the parsing techniques and our knowledge about linguistics have only fairly recently reached a level which permits the development of fully functional parsers. Also the little use of ... [parser]...in CALL is due to the fact that the development of a parser, a computational grammar and their integration in a CALL package is still a very complicated and time-consuming endeavour” (ibid: 11). Whilst robust wide-coverage parsers have been developed for a number of languages since Schulze’s remark, it remains the case that parsing for Arabic is very challenging, with reported accuracy of below 80% for data-driven dependency parsers.

Holland et al. (1999) argue that the benefit of parsers or programs capable of analysing human language in CALL is the opportunity it opens for students to practise productive and not only receptive language skills. While traditional CALL supports the receptive skills of reading and listening, intelligent CALL would in theory allow students to practice the productive skills of writing and speaking. It would act as human tutors do: point out obvious linguistic errors and provide individualised feedback, correction and remediation.

Heift and Nicholson (2000) implemented a parser–based CALL system for German which uses HPSG as its underlying grammatical formalism. This work differed from the traditional approach by permitting the grammar to freely generate as many parses as it could and using separate pedagogic principles to select the appropriate interpretation and response.

Kordoni and Nue (2005) explain that the Greek system used a similar architecture and language processing approach as it relied on rules that describe the ill-formed grammar of the input.

Chen and Tokuda (2002) developed an effective learner model for a template-structured ICALL system for Japanese-English writing skills which was reinforced by an efficient matching algorithm and a part of speech (POS) tagger.
Chapter 2: Language Learning and CALL

At UMIST\(^4\), between 1996 and 2001, a few projects on developing Parasite were carried out. Schulze (2001) reported that, in 1998, Brocklebank worked with Ramsay, the developer of the system, to explore the use of a parser for robust parsing of erroneous texts produced by Japanese learners of English. Parasite has been adapted for a number of languages such as Persian, Somalian, French and German. Schulze (Ramsay and Schulze, 2001; Ramsay and Mirzaiean, 2005) developed the Textana system, an experimental prototype of a grammar checker for English-speaking learners of German. It is meant to support and facilitate language learning by supporting the production of comprehensible output. In that sense, the program can be described as a writing tool (Williams and Holt, 1989). A writing tool is suitable to one or more of the three areas of writing; firstly, the pre-writing stage, secondly the composition and the post-writing stage and finally the lists of grammar checking as a post-writing stage. Williams (1991) states that “[s]tylecheckers are, perhaps surprisingly, more common than grammar checkers. Linguists know that the analysis of style is more complex than that of grammar, because grammatical variation is subsumed within stylistic variation. As with spelling checkers, most style checkers do not incorporate full syntactic analysis” (ibid: 256). It may not be as surprising, as statistical techniques are often easy to implement and are therefore used to analyse style. Our conclusion following this review for all these examples of Parser-based CALL applications is to make use of the important contributions in the further development of aRaBCALL tool for Arabic.

Mirzaeian (2003) has developed a system for Persian learners of English by reporting on semantic errors and then mismatches in translations into English. This was done by extending the Parasite parser to reduce overgeneration and limitations in the existing Persian computational grammar for Parasite.

The component of our project described here concentrates on the morpho-syntactic analysis. The aim is to provide support during the online learning strategies by setting different kinds of questions related to the grammar area such as pronouns, negation, verb, adjective, complementiser and weak verb.

\(^4\) University of Manchester Institute of Science and Technology
2.11.1 Advantages and Disadvantages for Parser-Based CALL

Over the years, researchers have raised the advantages and disadvantages of parser-based CALL. Holland compares parser-based CALL with conventional CALL and decided that “in parser-based CALL the student has relatively free reign and can write a potentially huge variety of sentences. [Parser-Based CALL] thus, permits practice of production skills, which require recalling and constructing, not just recognizing [as in conventional CALL] words and structures”. (Holland et al., 1993: 31)

In contrast, parsing imposes a few limitations as it entirely concentrates on the syntax of the textual input, therefore Parser-Based CALL “may actually subvert a principal goal of language pedagogy that of communicating meanings rather than producing the right forms”. (ibid: 32)

Schulze proposes that the disadvantage can be avoided by “a ‘focus on form’ which is mainly achieved by putting the parser/grammar checker to use within a relevant, authentic communicative task”. (Schulze 2001: 15)

Holland et al. (1993) believe that parser-based CALL is good for form-focused instruction: it benefits certain students; the whole sentence is tracked by the parser, errors are detected, classified and recorded by the system making it useful for research; and it plays a role in communicative practice.

Heift (2002) reports that the majority of students correct errors on their own which is another advantage of parser-based CALL systems as it provides students with the freedom of correcting their own mistakes.

It should be asked whether these are differences between efficiency of parser-based CALL and non-parser based. Nagata (1996) confesses that there is not a great difference between the outcome of non-CALL and CALL instructions. There is no clear evidence that this state of affairs by Nagata has changed since 1996. However, only CALL programs make good use of the potential by providing immediate and appropriate feedback (Blake, 2007).
2.12 Feedback and Soft Parsing

In this chapter, the specific questions of dealing with student error and the way in which feedback is given to users should also be considered. Heift and Schulze (2003) have raised an issue about the feedback techniques. He pointed out that theoretical issues relating to feedback techniques have not received the degree of attention they deserved: “The advent of the World Wide Web and to some extent of CD-ROMs has resulted in an overwhelming amount of technologically innovative, multi-media CALL materials. At the same time it appears that this development has also led to a reduced emphasis on pedagogical issues, one of which is learner feedback” Heift and Schulze (ibid: 433).

Demaziere and Blanvillain (1992) made a point that when generating feedback, students should not be overwhelmed with masses of linguistic information. The approach employed by many classroom teachers is to separate out and then select specific errors in the student work to which to respond. Rather than adopting a blanket response to all errors, teachers categorise them and may correct only when it is deemed appropriate. Heift and Schulze (2003) conclude that feedback provided to students must be accurate and made known if they are to read it. Feedback is available for a student but it is their choice whether to choose to read or to ignore it.

Heift and Schulze (2003) also recognise the lack of feedback in Web-based learning materials and propose the use of a parser to address the problem. Like Matthews (1998) and Tschichold (1997; 2003), she suggests that parsers for language learners cannot use the same tolerances as grammar checkers aimed at native speakers. She develops a CALL system with a parser for language learners at its heart, in which the approach has been to relax some of the morpho-syntactic rules. This notion of relaxing or removing some of the morpho-syntactic constraints to allow incorrectly formed sentences to parse is known as soft parsing.

The use of soft parsing is to provide a means of differentiating between input that is very nearly acceptable and input that cannot be allowed to go unchallenged. For example, input that displays inaccuracies that are considered to be minor, i.e., the swapping of frequently interchanged suffixes, can be dealt with by a request for
clarification. Therefore, the use of soft parsing opens up a number of possibilities as has just been seen. It allows more flexibility and targeting of feedback, as well as the ability to moderate the feedback according to perceived degree of incorrectness. However, this level of functionality is dependent on the existence of a mechanism for choosing whether to hard or soft parse. CALL systems should provide teachers with the means of deciding when to enforce and when not to enforce constraints and on exactly which elements, as well as the scope to moderate and even suppress error messages relating to those particular errors. For Heift and Schulze (2003), it is also important that against this background of multiple readings of student input there is scope to organise and even prioritise.

Morphological mistakes can be separated from syntactic errors. In addition, mistakes are sorted according to the respective values assigned to various word types and features. Elements with a higher rank position will be the ones for which an error message will be generated, thereby determining the precedence and length of the student feedback.

### 2.13 Overview on Our Arabic CALL System

The nature of ICALL applications makes it quite difficult to classify developed applications in the field. Gamper and Knapp (2002) base classification of such systems on the involvement of the target language, the use of AI system techniques, the training on language skills and language elements, and finally to know if the system is Web-based or if they will purchase access to the system. The following two points from the above classifications are of interest:

- There has not previously been any such classified ICALL system for Arabic learners.
- Specific morpho-syntactic problems for learners of Arabic have not been addressed directly by such programs.

Therefore, the current thesis is the first attempt of this type to address non-native speakers as learners of Arabic by focusing on the structured errors they commit while
learning the target language. Based on this aim, it is essential to extend the existing Arabic parser as well as an English parser which will be needed due to the system being for English speaking Arabic learners. The English and Arabic parsers are already available (Ramsay and Mansour, 2004), however, in order to be able to use them, a formal linguistic description of the language is required which should be provided in the dictionary of Arabic which is expressed in HPSG-based formalism for use within the current parser. Therefore, the next two chapters will be dedicated to the linguistic and computational description as well as implementation of Arabic morphology, lexical and syntax (chapters 3 and 5).
Chapter 3

Characteristics of Arabic

3.1 Introduction

This chapter is devoted to the linguistic and computational description of Arabic morphosyntax. It is important to study the issues that make Arabic hard for English native speakers to learn to be able to overcome them. A detailed survey of the characteristics of Arabic, with particular emphasis on areas where it differs from English, will be presented in this chapter. It is expedient to go through some fundamental principles of morphology that show Arabic morphological complexity. Thus, firstly, a detailed explanation of the lexicon morphology and syntax of Arabic will be provided, and later other morphological issues pertaining to verbs, nouns, adjectives and adverbs will be explored respectively.

3.2 Properties of Arabic

Arabic is one of the oldest languages of the ancient world. Arabs from different parts of the Arab world speak various forms of Arabic (Mahir, 2003). All these various forms are different from Classic Arabic (CA). There are three kinds of Arabic: Classical, Modern Standard and Colloquial. Classical Arabic is the language of the Quran and prophetic traditions. Modern Standard Arabic (MSA) is used in newspapers, text books, academic books and other writing (Parkinson, 1990). With colloquial Arabic, the dialects differ from one country to another and even differ inside a particular country.
Arabic has many complexities which can be challenging to theoretical and computational linguistics (Chalabi, 2000; Daimi, 2001; Fehri, 1993). The following is a list of some of the major issues involved in Arabic linguistic analysis:

- Arabic is a syntactically flexible language. Arabic has a fairly free-word order: the orders VSO, VOS, SVO are all acceptable sentence structures. Daimi (2001) emphasised that Arabic shows high syntactical flexibility, such as the omission of some arguments associated with verbs, and in many cases a noun can occupy the place of the verb.

- Morphological analysis of Arabic is complicated due to it being a highly inflectional language and also with Arabic written forms omitting short vowels and a number of other marks. The words in Arabic are built from roots rather than stems. Furthermore, diacritics that help in the pronunciation of words are usually omitted in modern writing. Chalabi (2000) even claims that the absence of diacritization in Arabic causes computational complexity ‘one order of magnitude bigger than handling Latin-based language counterparts’.

- The main Arabic sentence structure is VSO. However, Arabic also has a different structure of a subject phrase and a predicate phrase, without a verb.

### 3.3 Arabic Morphology: A descriptive Analysis

#### 3.3.1 Introduction

Arabic is a highly inflected natural language with a complex morphological system, due to the clearly marked features for tense, gender, number, case, and other features (Diab, 2007).

The main characteristic feature of Arabic is that it is counted as a non-concatenative language because it involves more than just adding affixes to the root of the word to generate a certain word (Khoja, 2001).
The main characteristic feature of Arabic morphology that causes it to be counted as a non-concatenative type is because it depends on manipulating the letters of the root in a non-concatenative manner to generate a certain word (Soudi et al., 2001; 2007). Therefore the main feature of an Arabic word is that it is constructed from adding affixes or infixes and also by following a certain pattern (Khoja, 2001).

The study of the rules and categories that underlie sentence formation is known as syntax. We use the term morphosyntax as many linguists use the word to capture the fact that morphemes often have sentence-level functions. A morpheme can indicate whether a form functions as subject or object in a given sentence.

Arabic morphology is very rich and based on root-pattern structure. Mainly, Arabic words are generated out of a finite set of roots. The hypothesis of nominal patterns is that a single Arabic root can generate hundreds of words such as a noun or verb. Arabic words may exist in hundreds of shapes in normal text by adding certain suffixes and prefixes (Kiraz, 2000).

We will consider in the next section some morphological aspects of the Arabic language. We will start with discussing the notion of root and pattern in Arabic, then throw light on the basic lexical categories: nouns, verbs, particles and prepositions.

3.3.2 Arabic Word Pattern

The root and the pattern are the main standards of Arabic derivational morphology. However, the root of most words in Arabic is the semantic concept. Most roots consist of three radicals, whilst a smaller number have two or four radicals (Fischer, 2002). فعل الفعل $fEl$ ‘to do’ is the symbolized form from which words are derived. Arabic grammarians use the abstract pattern فعل الفعل $fEl$ to illustrate the various patterns families of verbs display. For example, the verb كتاب $ktb$ ‘to write’ can be substituted from the real radicals $k$, $t$, $b$ to the theoretical radicals $f$, $E$, $l$ (Holes, 2004).

In fact, early grammarians introduced a notation for describing root-and-pattern morphology. The three radical consonants [$f$], [$E$], [$l$] represent the root. The pattern is
represented by inserting a vocalic morpheme in the slot within the root consonant. This is similar to other Semitic languages where stem pattern morphemes determine the distribution of vowels in a word stem (Khalil, 2002). The Arabic verb دَرَسَ daras ‘studied’ consists of two elements: the root دَرَسَ dars ‘he studied’ and the diacritic patterns [---,a,---,a]. Chomsky (1951) developed a formal description of Semitic morphology, where the verb دَرَسَ dar-as ‘he taught’, for example, is generated by a transformational rule as follow:

\[ C_1C_2C_3 \rightarrow C_1V_1C_2C_2V_1C_3 \]

The above rule states the root \([C_1C_2C_3]\) generates the pattern \([C_1V_1C_2C_2V_2C_3]\) through duplicating the middle C2 and inserting the same short vowels in the slots joining the first and the last two consonants (Chomsky, 1957).

There is a set of verbal patterns in Arabic, which consist of root consonants and vowels slotted between these consonants. There are 15 derived forms of the triliteral verb but the four of these are very rare (Moutouakil, 2004). Each pattern is associated with a semantic unit. Roots do not take all the 15 verbal patterns. Every root has idiosyncratic lexical meanings, which determine the patterns.

Nouns and verbs are similar, as they are derived according to patterns. Arabic has case markers, which are marked by suffixes. The morphemes of verbal forms are less numerous than the nominal pattern. There are four categories which are classified as the nominal pattern. The entire number of patterns is more than 100 morpheme patterns. According to Badawi et al. (2004), both verbs and nouns are derived according to patterns. Arabic roots are divided into several classes such as biliteral, triliteral, quadriliteral and quinquiliteral patterns and each of these patterns has a set of patterns. All these patterns consist of root consonants and root vowels. Any derived forms are changing the meaning of the basic stem to various semantic root meanings (ibid).

Holes (2004) defined patterns which are interdigitated with the root in two areas: Morphosemantics and Morphosyntax. Thus, the morpheme pattern مفعول mafEwl is a
noun. Semantically, the derived patterns often share the same meaning. The root and its related morphological patterns often associate with particular meanings. The following table represents different forms derived from the same root.

<table>
<thead>
<tr>
<th>Root</th>
<th>Derived Patterns</th>
<th>Radical</th>
<th>Derived Word</th>
<th>Syntactic Category</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>درس</td>
<td>$faEala$</td>
<td>عَلَ</td>
<td>$darasa$</td>
<td>verb</td>
<td>He studied</td>
</tr>
<tr>
<td></td>
<td>$faEa-l$</td>
<td>عَلَ</td>
<td>$darrasa$</td>
<td>verb</td>
<td>He taught</td>
</tr>
<tr>
<td></td>
<td>$faEil$</td>
<td>عَلَ</td>
<td>$daris$</td>
<td>noun</td>
<td>Learner</td>
</tr>
<tr>
<td></td>
<td>$muEafil$</td>
<td>عَلَ</td>
<td>$mudaris$</td>
<td>noun</td>
<td>Teacher</td>
</tr>
<tr>
<td></td>
<td>$mafEalatp$</td>
<td>عَلَ</td>
<td>$madorasp$</td>
<td>noun</td>
<td>School</td>
</tr>
</tbody>
</table>

Table 3.1: Example of a derived form for the root $درس$ $dars$

We notice from the table that a number of nominal and verbal forms have been derived from same root. Moreover, in most cases the attached affixes indicate certain diacritic patterns for the stem, although, due to the missing diacritics in MSA text, the same surface form has more than one pronunciation or diacritic pattern. The context will assign only one reading. For example, درس $dars$ ‘to study’ may be pronounced in several ways shown in the following:

$دارِس$ $darasa$ → past verb (Pattern I)
$دار-اسا$ $dar-asa$ → past verb (Pattern II)
$دْرس$ $dursa$ → passive past voice
$دْرِس$ $dur-is$ → passive present voice

3.3.3 Arabic Word Structure

Morphotactics is the study of the techniques by which morphemes combine together to form words (Beesley, 1998; Beesley and Karttunen, 2003). In general, morphotactics
can be either concatenative, with morphemes either prefixed or suffixed to stems, or non-concatenative, with stems undergoing internal alternations to convey morphosyntactic information (Attia et al., 2010).

Forms of words are complex units which include the following:

- **Prefixes.** This category includes, for example, the prefixes of the imperfective, e.g. \( ya+ \), prefixed morpheme of the 3rd person.

- **Proclitics,** which come at the beginning of a word. These include monononsonantal conjunctions (e.g. \( wa+ \) ‘and’ and \( li+ \) ‘in order to’) and prepositions (e.g. \( bi+ \) ‘with’ and \( fy+ \) ‘in’ or ‘at’and \( li+ \) ‘for’).

- **Stems,** which are represented in terms of a root and a pattern. The root is an ordered triple or quadruple of consonants. In most cases roots are either trilateral such as \( zar \) ‘to visit’ or quadriliteral such as \( Hawl \) ‘to try’. The pattern is a template of syllables, the consonants of which are that of the trilateral or quadriliteral root. Therefore, stems are formed by a derivational combination of a root morpheme and a vowel melody; the two are arranged according to canonical patterns (Abbès et al., 2004; Smrž, 2007).

- **Suffixes,** such as nominal cases, verb endings, and the nominal feminine ending with the letter \( taA\` marbuwTap \) or the letter \( AlmaqSuwrap \).

- **Enclitics,** which are added at the end of a word. In Arabic, enclitics are generally complement pronouns. The following figure illustrates the Arabic word structure.
The word structure will be discussed in more detail in chapter 5.

3.3.4 Non-Linear Morphology

Non-linear morphology (or non-concatenative) covers a wide range of morphological phenomena. A number of languages have extensive non-concatenative morphology, which combines the morphemes in many complex ways.

McCarthy (1981) and McCarthy and Prince (1990) presented the non-concatenative Arabic morphology by separating the vowels and consonants of the word form onto single tiers. Therefore, a form like دُرِسَ durisa ‘it has been studied’ is represented as in the following scheme:

(3.1)

| Vocalic tier: u i ‘perfective passive’ |
| Skeletal tier C V C V C |
| Consonant: d r s ‘to study’ |
| Lexical form: d u r i s |
McCarthy’s scheme uses three different tiers; each tier contributes to the word form. They are the root consonants (drṣ in 3.1), the pattern vowels or the vowel melody (ui in 3.1), which indicates the passive, and the skeletal pattern CVVCV, which indicates how to gather the parts. These three morphological components link together in a non-linear process to make (3.1) the complete lexical form دُرِسُ durṣ ‘it was been studied’. If one pattern is changed, the resulting word changes. For example, we can change the vowels tier to a, a, resulting in daras ‘to study’ and we can change the CV’s skeleton to CVVCVC, resulting in دارِس dAris ‘a learner’.

Habash (2007) explained non-linear Arabic morphology by distinguishing between morpheme type and morpheme function. Morpheme type includes morphemes with their interactions if they are templatic or affixational. Thus, morpheme function means the nature of the morphology is derivational or inflectional morphology. For example, the word دراس daras ‘to study’ is constructed by creating a word stem from templatic morphemes or using non-templatic word stems, to which affixational morphemes are then added. For example, the word is a combination of root, pattern and vocalism. An Arabic word can also be created by using a non-templatic word stem, and adding to it an affixational morpheme. For example, the word سَيَدرِسونه sayadrswnhu is a non-templated word with two prefixes سَ and يَ and two suffixes وَ and هِ. The detailed analysis of McCarthy’s approach will be explained in the following section.

### 3.3.4.1 McCarthy’s Approach

McCarthy and Prince (1990) presented a study on description of Arabic morphology that has become the basis in the field (Kiraz, 1994a; 2001).

The CV approach was described by McCarthy and Prince (1990) as a good introduction to Arabic non-linear morphology. The CV approach argues for a pattern morpheme that consists of sequences of C and V. McCarthy (1981) developed a linguistic model for describing the stem as represented in table 3.2 below.
• Root morphemes (Cs).
• Vocalism morphemes (vowel melody).
• Pattern morphemes consist of Cs and Vs.

<table>
<thead>
<tr>
<th>Root</th>
<th>Vocalism</th>
<th>Pattern</th>
<th>Stem</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>drs</td>
<td>a,a</td>
<td>d,a,r,a,s</td>
<td>dras</td>
<td>He studied</td>
</tr>
<tr>
<td>lEib</td>
<td>a,i</td>
<td>l,a,E,i,b</td>
<td>lAEib</td>
<td>He played</td>
</tr>
</tbody>
</table>

Table 3.2: CV non-linear morphology approach

Each morpheme can be represented on the autonomous tiers as shown in figure (3.1). Every consonant slot should be associated with at least one consonant. Each vowel melody slot must be represented by at least one vowel. The morphemes are coordinated with association lines according to the principles of autosegmental phonology, as shown in figure (3.2).

(3.2a)

(3.2b)

Figure 3.2: CV derivational processing approach
Chapter 4: Classification of Learner Errors

Figure (3.2) states the derivational processing of pattern I for the root درس *drs* ‘to study’ and لعب *lEb* ‘to play’. The first tier is for the vocalism morpheme. The second tier shows the morpheme pattern and the third one represents the vocalic pattern. The morphemes are coordinated under two principles: Well-Formedness Condition and Association Convention (McCarthy, 1981; McCarthy and Prince, 1990; Kiraz, 1994b).

- **Well-Formedness Condition**

  In this state, every CV skeletal slot has to be associated with at least one melody element, every consonant must be associated with at least one C slot and every vocalic melody element must be associated with at least one V slot. Also, the association lines must not cross.

- **Association Convention**

  Only the rightmost member could spread to more than one member of the other tier. These principles account for the insertion of vowels between the consonants that make up the root. In a number of cases one of the consonants also needs to be duplicated: this reveres further special rule.

  There are a number of other approaches to Arabic morphology, e.g. the moraic approach (McCarthy and Prince, 1990), but the templatic approach is the most widely used, and is the one that we use in Parasite.

### 3.3.5 Arabic Lexical Categories

Arabic is considered as a typical example of a language that occupies non-concatenative morphotactics. Usually, the Arabic grammarians classify Arabic words into three common lexical categories, which are verbs, nouns and particles. The traditional category of nouns includes nouns that are used as adjectives and words that in other grammatical traditions would be called adjectives. These categories can be classified also into further sub-classes which cover the whole of the Arabic language.
3.3.5.1 Nouns

In Arabic, a noun is a word that describes a person, thing, or idea or it is a name to refer to a person, thing or idea. Usually, the noun category in Arabic is subdivided into Derivatives (i.e., noun derived from verb, noun or from particle) or Primitives (noun not so derived). This noun may be further sub-categorised by number, gender and case. The noun category also includes items which in European grammatical theory would be classified as Gerunds, Pronouns, Demonstratives, Relatives and Interrogatives.

All Arabic nouns and adjectives have singular, dual and plural forms. Noun in Arabic is either masculine or feminine. There are many patterns for nouns and adjectives, although almost all nouns belong to at most two patterns. The difference between patterns is the suffix (the letter sounds \( \text{ة} \) \( ha \)) representing the feminine gender appearing at the end of the noun.

Arab grammarians have traditionally distinguished between two types of plurals usually called sound and broken (Levy, 1997) (also called regular and broken plural). A regular plural is formed by adding the feminine plural suffix \( \text{ات} \) \( At \), or the masculine plural suffix \( \text{ون} \) \( wn \) to the singular noun. Broken plurals are formed differently by a number of processes including prefixation and mainly changing the diacritic patterns. Most Arabic broken plurals are feminine (Ryding, 2005). Arabic has 30 diacritic patterns for broken plural (Cowan, 1958).

3.3.5.2 Pronouns

There are two types of pronouns: personal pronouns (such as \( \text{أنا} \) \( OnA \) ‘I’) and demonstrative pronouns (such as \( 
\text{هذا} \) \( h*A \) ‘this’). The surface form of the personal pronoun varies in Arabic depending on its function within the sentence (i.e., nominative, accusative, possessive or genitive). The pronoun \( \text{أنا} \) \( OnA \) ‘I’ is nominative, however, the clitic pronoun \( 
\text{ي} \) \( y \) ‘my’ or ‘me’ is accusative and genitive. The nominative personal pronoun only comes at the beginning of the sentence. Pronouns in Arabic belong to the category of ‘nouns’. Therefore, anything that applies to nouns will apply to them.
Pronouns are always definite. Pronouns in Arabic have numbers, genders and grammatical case. All the Arabic nominative pronouns are shown in the table below and also the table shows that the dual second and third person pronouns are the same for masculine and feminine such as the pronoun أنا *OnA* ‘I’, or attached subject pronouns يُ + ي in *kitAby* ‘my book’.

<table>
<thead>
<tr>
<th>Number</th>
<th>Gender</th>
<th>1st Person</th>
<th>Gloss</th>
<th>2nd Person</th>
<th>Gloss</th>
<th>3rd Person</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Singular</td>
<td>Mas</td>
<td>أنا <em>OnA</em></td>
<td>I</td>
<td>أنتَ <em>Onta</em></td>
<td>You (mas/sing.)</td>
<td>هُوَ <em>huwa</em></td>
<td>He</td>
</tr>
<tr>
<td></td>
<td>Fem</td>
<td>أنا <em>OnA</em></td>
<td>I</td>
<td>أنتِ <em>Onti</em></td>
<td>You (fem/sing)</td>
<td>هِيَ <em>hiya</em></td>
<td>She</td>
</tr>
<tr>
<td>Dual</td>
<td>Mas</td>
<td>نَحنُ <em>naHnu</em></td>
<td>We</td>
<td>أنتُمَا <em>OnturnA</em></td>
<td>You (mas/dual)</td>
<td>هُمَا <em>humA</em></td>
<td>They (mas/dual)</td>
</tr>
<tr>
<td></td>
<td>Fem</td>
<td>نَحنُ <em>naHnu</em></td>
<td>We</td>
<td>أنتُمَا <em>OnturnA</em></td>
<td>You (fem/dual)</td>
<td>هُمَا <em>humA</em></td>
<td>They (fem/dual)</td>
</tr>
<tr>
<td>Plural</td>
<td>Mas</td>
<td>نَحنُ <em>naHnu</em></td>
<td>We</td>
<td>أنتُمَا <em>AntumA</em></td>
<td>You (mas/pl)</td>
<td>هُم  <em>hum</em></td>
<td>They (mas/pl)</td>
</tr>
<tr>
<td></td>
<td>Fem</td>
<td>نَحنُ <em>naHnu</em></td>
<td>We</td>
<td>أنتُمَا <em>AntumA</em></td>
<td>You (fem/pl)</td>
<td>هُنَّ <em>hunna</em></td>
<td>They (fem/pl)</td>
</tr>
</tbody>
</table>

Table 3.3: Personal nominative pronouns

The coming table 3.4 demonstrates the attached pronouns.
<table>
<thead>
<tr>
<th>Person</th>
<th>Singular</th>
<th>Dual</th>
<th>Plural</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pro</td>
<td>Example</td>
<td>Pro</td>
</tr>
<tr>
<td>1st</td>
<td>Pro</td>
<td>ی</td>
<td>我的书</td>
</tr>
<tr>
<td></td>
<td>ny</td>
<td>ني</td>
<td>我的书</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>你</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>你们</td>
</tr>
<tr>
<td></td>
<td>Fem</td>
<td>她</td>
<td>她的书</td>
</tr>
<tr>
<td>2nd</td>
<td>Mas</td>
<td>他</td>
<td>他的书</td>
</tr>
<tr>
<td></td>
<td>Fem</td>
<td>她</td>
<td>她的书</td>
</tr>
<tr>
<td>3rd</td>
<td>Mas</td>
<td>他</td>
<td>他的书</td>
</tr>
<tr>
<td></td>
<td>Fem</td>
<td>她</td>
<td>她的书</td>
</tr>
</tbody>
</table>

Table 3.4: Arabic attached pronouns
In the table above, we notice that, in the first person singular, يٌ is added to a noun to show possession, such as my book, but after a verb, نيٌ نيٌ+ny is used as a direct object such as درُسنيٌ ‘he taught me’. The يٌ +y ending remains the same whether the word is in the nominative, accusative or genitive case. For example, كتابي كَبير kitAby kabyr ‘my book is big’ is nominative, هوُ أَخْذُ كتابيٌ hw Ox* ktAby ‘he took my book’ is accusative, or القَلمُ تحتُ كتابيٌ Olqalam tHt kitAby ‘the pen is under my book’ is genitive.

Table 3.5 shows the Arabic demonstrative pronouns. Arabic has about thirteen demonstrative pronouns which can also be used as a determiner, in which case they agree with the noun in number and gender.

<table>
<thead>
<tr>
<th>Number</th>
<th>Surface form</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Singular</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>هذا</td>
<td>This (mas)</td>
</tr>
<tr>
<td></td>
<td>هذه</td>
<td>This (fem)</td>
</tr>
<tr>
<td>Dual</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>هذان</td>
<td>These (mas)</td>
</tr>
<tr>
<td></td>
<td>هتان</td>
<td>These (fem)</td>
</tr>
<tr>
<td>Plural</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>هؤلاء</td>
<td>These (mas)</td>
</tr>
<tr>
<td></td>
<td>هؤلاء</td>
<td>These (fem)</td>
</tr>
<tr>
<td>Singular</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ذلك</td>
<td>That (mas)</td>
</tr>
<tr>
<td></td>
<td>تلك</td>
<td>That (fem)</td>
</tr>
<tr>
<td>Dual &amp; Plural</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>أولئك</td>
<td>Those (mas)</td>
</tr>
<tr>
<td></td>
<td>أولئك</td>
<td>Those (fem)</td>
</tr>
</tbody>
</table>

Table 3.5: Arabic demonstrative pronouns

The table above shows that the dual and the plural demonstrative pronouns أولئك OlA\jika ‘those’ are the same for plural feminine and masculine.
3.3.5.3 Adjectives

Adjectives take practically all the morphological forms and share the same templatic structures with nouns. Adjectives, for example, can be definite, and are modified for case, number and gender.

Table 3.6 below shows the pattern for the adjectival noun طويل Twyl ‘tall’ and the derived forms regarding the gender and number.

<table>
<thead>
<tr>
<th>Gender</th>
<th>Singular</th>
<th>Dual</th>
<th>Plural</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mas</td>
<td>طويل Twyl</td>
<td>طويلان TwylAn</td>
<td>طويل TwAl</td>
<td>(irregular plural from a pattern fiEAl)</td>
</tr>
<tr>
<td>Fem</td>
<td>طويلة Twylp</td>
<td>طويلتان TwyltAn</td>
<td>طويلات TwylAt</td>
<td>(regular fem. plural)</td>
</tr>
</tbody>
</table>

Table 3.6: Pattern of adjectival noun طويل Twyl ‘tall’

As the table above shows, the adjectival feminine plural has the regular feminine plural form, however, the masculine plural adjectival noun will either be in an irregular plural form or in a regular feminine plural (Wightwick and Gaafar, 2007).

There are two types of colours in Arabic: the nominal name of colours and the adjectival colours. The nominal colours are the pure names of the colours such as green or black, however, the adjectival colour is the derived colour which is driven from a colour of a substance or a noun.
Table 3.7: All Arabic adjectival colours

Table 3.7 presents all the pure Arabic colour names and as shown, the pure colours are different in gender only. However, table 3.8 below illustrates the derived colour names. The colour برتقالي brtqAl means the colour is ‘orange’, derived from the fruit name orange and it means the orange colour.

Table 3.8: Arabic derived colours

As clarified in the previous table 3.8, all colours derived or adjectival has a dual and plural pattern and it goes with all numbers and both genders.

3.3.5.4 Verbs

The verb in Arabic and the verb in English are different. This is because all aspects of tenses are different. The verb class can be sub-categorised into perfect and imperfect
tenses (which may be subjunctive, nominative, jussive and imperative) which are separate dimensions as shown in tables 3.9 and 3.10.

<table>
<thead>
<tr>
<th>Person</th>
<th>Number</th>
<th>Masculine</th>
<th>Feminine</th>
<th>Example</th>
<th>Mood</th>
</tr>
</thead>
<tbody>
<tr>
<td>1&lt;sup&gt;st&lt;/sup&gt;</td>
<td>Sing</td>
<td>تَّ tu…</td>
<td>تَّ tu…</td>
<td>كتبتُ katabtu</td>
<td></td>
</tr>
<tr>
<td>1&lt;sup&gt;st&lt;/sup&gt;</td>
<td>dual &amp; pl</td>
<td>نا naA…</td>
<td>نا naA…</td>
<td>كتبناا katabnaA</td>
<td></td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt;</td>
<td>sing/mas</td>
<td>تَّ ta…</td>
<td>-</td>
<td>كتبتُ katabta</td>
<td></td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt;</td>
<td>sing/fem</td>
<td>-</td>
<td>تَّ ti…</td>
<td>كتبتي katabti</td>
<td></td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt;</td>
<td>dual/mas and dual/fem</td>
<td>تَّ tumaA…</td>
<td>تَّ tumaA…</td>
<td>كتبناا katabtumaA</td>
<td></td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt;</td>
<td>pl/mas</td>
<td>تَّ tum…</td>
<td>-</td>
<td>كتبهم katabtum</td>
<td></td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt;</td>
<td>pl/fem</td>
<td>-</td>
<td>تَّ tun~a…</td>
<td>كتبناا katabtn~a</td>
<td></td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt;</td>
<td>sing/mas</td>
<td>-</td>
<td>-</td>
<td>كتبُ kataba</td>
<td></td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt;</td>
<td>sing/fem</td>
<td>-</td>
<td>تَّ at…</td>
<td>كتبَتُ katabat</td>
<td></td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt;</td>
<td>dual/mas</td>
<td>تَّ aA…</td>
<td>-</td>
<td>كتبَا katabaA</td>
<td></td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt;</td>
<td>dual/fem</td>
<td>-</td>
<td>تَّ atA…</td>
<td>كتبناا katabatA</td>
<td></td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt;</td>
<td>pl/mas</td>
<td>وَا uwA…</td>
<td>-</td>
<td>كتبوا katabuwA</td>
<td></td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt;</td>
<td>pl/fem</td>
<td>نَ na…</td>
<td>كتبنا katabna</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.9: Representing a perfective verb mood
<table>
<thead>
<tr>
<th>Person</th>
<th>Number</th>
<th>Masculine</th>
<th>Feminine</th>
<th>Example</th>
<th>Mood</th>
</tr>
</thead>
<tbody>
<tr>
<td>1&lt;sup&gt;st&lt;/sup&gt;</td>
<td>Sing</td>
<td>A……أ</td>
<td>A……أ</td>
<td>اكتبُ Akotubu</td>
<td></td>
</tr>
<tr>
<td>1&lt;sup&gt;st&lt;/sup&gt;</td>
<td>dual pl/mas &amp; dual &amp; pl/fem</td>
<td>na……ن</td>
<td>na……ن</td>
<td>تكتبُ Nakobubu</td>
<td>Nominative</td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt;</td>
<td>sing/mas</td>
<td>تا……ث</td>
<td>تا……ث</td>
<td>تكتبُ Takotuba</td>
<td></td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt;</td>
<td>sing/fem</td>
<td>تا……ث</td>
<td>تا……ث</td>
<td>تكتبُ تكتبُ Takotubona</td>
<td></td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt;</td>
<td>dual/mas &amp; dual/fem</td>
<td>تا……ث</td>
<td>تا…..ث</td>
<td>تكتبُ takotubAna</td>
<td>Nominative, Subjunctive &amp; Jussive</td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt;</td>
<td>pl/mas</td>
<td>تا…..ث</td>
<td>تا…..ث</td>
<td>تكتبُ takotubwna</td>
<td></td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt;</td>
<td>pl/fem</td>
<td>تا…..ث</td>
<td>تا…..ث</td>
<td>تكتبُ takotubon~a</td>
<td></td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt;</td>
<td>sing/mas</td>
<td>ي…</td>
<td>ي…</td>
<td>يكتبُ Yakotubu</td>
<td>Nominative</td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt;</td>
<td>sing/fem</td>
<td>ت…</td>
<td>ت…</td>
<td>تكتبُ Takotubu</td>
<td></td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt;</td>
<td>dual/mas</td>
<td>ي…</td>
<td>ي…</td>
<td>يكتبُ yakotubAn</td>
<td>Nominative, Subjunctive &amp; Jussive</td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt;</td>
<td>dual/fem</td>
<td>ت…</td>
<td>ت…</td>
<td>تكتبُ takotubAn</td>
<td></td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt;</td>
<td>pl/mas</td>
<td>ي..wn</td>
<td>ي..wn</td>
<td>يكتبون Yakotubwn</td>
<td></td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt;</td>
<td>pl/fem</td>
<td>ي…ن</td>
<td>ي…ن~</td>
<td>يكتبون yakotubn~</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.10: Representing an imperfective verb mood
Moreover, the sub-categorisations of the verb class are likely using number, gender and person. Arabic verbs have at least three letters. Sometimes the second and third letters are combined together with a sign which is called (ـ، َّ، shadda), literally it means to emphasize the letter when it is pronounced. Therefore, the verb looks like it has two letters but in reality it has three letters because of the shadda. In Arabic, there are never two letter verbs. Surface forms for the present verb tenses كتب ktb ‘to write’ pattern are given as follows.

<table>
<thead>
<tr>
<th>Gender</th>
<th>1st Person</th>
<th>2nd Person</th>
<th>3rd Person</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sing</td>
<td>كتب ً</td>
<td>َكتب</td>
<td>يكتب</td>
<td>ktb</td>
</tr>
<tr>
<td>Fem</td>
<td>كتب ً</td>
<td>َكتب</td>
<td>يكتب</td>
<td>ktb</td>
</tr>
<tr>
<td>Dual</td>
<td>نكتب ً</td>
<td>َكتبان</td>
<td>يكتبان</td>
<td>ktbAn</td>
</tr>
<tr>
<td>Fem</td>
<td>نكتب ً</td>
<td>َكتبان</td>
<td>يكتبان</td>
<td>ktbAn</td>
</tr>
<tr>
<td>Pl</td>
<td>نكتب ً</td>
<td>َكتبون</td>
<td>يكتبون</td>
<td>ktbwn</td>
</tr>
<tr>
<td>Fem</td>
<td>نكتب ً</td>
<td>َكتبين</td>
<td>يكتبين</td>
<td>ktb</td>
</tr>
</tbody>
</table>

Table 3.11: Surface forms for the present tense verb pattern كتب ktb

Table 3.11 above shows the stem of the present tense of the word كتب ktb which contains no diacritic signs for the normal MSA word writing. It elaborates the present tense word يَكتب yakotub ‘he is writing’ and other tenses with underlying forms.
3.3.5.5 Particles

The Particles such as prepositions, adverbs, conjunctions and interrogative particles are the particle classes in the Arabic language (Al-Khuli 1979; Khoja 2001; Ahmed, 2005). Recently, and based on the traditional Arabic lexical categories, different parts of speech are used for describing MSA words (Khalil, 2002). We use the grammarian’s traditional classification with a small modification, namely considering the prepositions an individual class, so we have four main categories: verbs, nouns, prepositions and particles. There are two kinds of prepositions in Arabic, attached and detached preposition as shown in table 3.12 below.

<table>
<thead>
<tr>
<th>Attached Prepositions</th>
<th>Gloss</th>
<th>Detached prepositions</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>ب b</td>
<td>With</td>
<td>في fy</td>
<td>In</td>
</tr>
<tr>
<td>ك k</td>
<td>Similar</td>
<td>على ElA</td>
<td>On</td>
</tr>
<tr>
<td>ف f</td>
<td>Then</td>
<td>من mn</td>
<td>From</td>
</tr>
<tr>
<td></td>
<td></td>
<td>إلى IlA</td>
<td>To</td>
</tr>
<tr>
<td></td>
<td></td>
<td>عن En</td>
<td>About</td>
</tr>
<tr>
<td></td>
<td></td>
<td>فوق fwq</td>
<td>Above</td>
</tr>
<tr>
<td></td>
<td></td>
<td>تحت iHt</td>
<td>Under</td>
</tr>
</tbody>
</table>

Table 3.12: Kinds of Arabic prepositions

The complementiser table 3.13 below shows the type of the properties for the rest of the sentence.
<table>
<thead>
<tr>
<th>Complementiser</th>
<th>Example</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>On</em></td>
<td>أنَّ أعرف أنَّ الطفل نائم AErф On AlTfl nA ’m</td>
<td>I know the child is sleeping</td>
</tr>
<tr>
<td><em>In</em></td>
<td>إنْ المدرس يعرف إنَّ الطالب يكتب الدرس Almdrs yErf In~ ALTAlb yktb Aldrs</td>
<td>The teacher knew that the student wrote a lesson</td>
</tr>
<tr>
<td><em>On</em></td>
<td>أنْ المدرس يعتقد أنَّ الطالب يكتب الدرس Almdrs yEtqd On~ ALTAlb yktb Aldrs</td>
<td>The teacher thinks that the student wrote a lesson</td>
</tr>
<tr>
<td><em>Iа</em></td>
<td>لا يدخن الرجل ِIA ydXn Alrjl</td>
<td>He does not smoke</td>
</tr>
<tr>
<td><em>mA</em></td>
<td>ما درس الطالب ِmA drs ALTAlb</td>
<td>The student did not study</td>
</tr>
<tr>
<td><em>lm</em></td>
<td>لم يلعب الطفل في الحديقة lm ylEb AlTfl fy Alhdyqp</td>
<td>The child is not played in the garden</td>
</tr>
<tr>
<td><em>In</em></td>
<td>لن يذهب الطالب إلى المدرسة حتى يشفى ln y*hb ALTAlb IIY Aldrsp HtY y§fY</td>
<td>The student will not go to school until he gets well</td>
</tr>
</tbody>
</table>

Table 3.13: Arabic complementiser with its property
Chapter 4: Classification of Learner Errors

Table 3.13 above shows the property of the sentence when there is a complementiser.

3.4 Arabic Syntax: A Descriptive Analysis

3.4.1 Introduction

In this section, we will consider the syntactic phenomena which we will explore in our CALL framework. Arabic syntax is less complicated than morphology but it is rather ambiguous in surface form. Arabic is a free-word order language so the first computational challenge issue is the word order.

There are two kinds of Arabic sentences making both nominal statement and verbal statement (Hoffiz and Mace, 2003). A written Arabic sentence consists of subject and predicate. The subject agrees in different aspects with the predicate in number, gender and person but under some considerations (Holes, 2004). As for the predicate, it is classified in limited sorts for nominal and verbal statement, as we will see later in this chapter. Noun phrases in Arabic are classified into a variety of types. The varieties of noun phrases (NPs) come from the different sorts of heads of NPs and case markers of the NP head.

3.4.2 Arabic Sentence Structure

Arabic has complex and multiple aspects of syntactic structure; however, in Arabic grammatical tradition, there are two distinguishing classifications of sentences: verbal and nominal. Verbal sentences are those that begin with a verb, whereas, a nominal sentence begins with a noun. It should be noted that under this classification a nominal sentence may contain a verb: a sentence in SVO order will be classified as nominal (Wright, 1969).

The alternative classification, which we will follow, divides sentences into equational sentences, which do not contain a verb, and verbal sentences, which do (Badawi et al. 2004; Ryding, 2005). According to this classification, SVO sentences are verbal. The
confusion of terminology is unfortunate, but is widespread in the literature. As noted, we will use ‘verbal’ to mean ‘contains a verb’.

In other words, there are three main types of Arabic basic sentences that Badawi et al. (2004) classify. The first type is equational sentences, which consist of subject and predicate only, with no verbal elements. An example of this type is the sentence طُرِيق الطَّرِيق طُويل A\textit{r}iyyu Tawiy\textit{l} ‘the road is long’. The second type mentioned is the topic + comment structure which also does not contain any verbal copula. In this type, the topic is a (NP) in the initial position and the comment is an entire verb phrase (VP) anaphorically linked to the topic. These first two types are traditionally labeled as a جُملة اسمية \textit{jumlap Asomiy} ‘nominal sentence’, due to them beginning with nouns. The last type is verbal sentences, which consists of a verb always in the first position accompanied by the agent, which is usually in the second position and the other complements usually in the third position. Wright (1969) mentions that a nominal sentence is one where if the predicate is a noun, a prepositional phrase or a verbal predicate, it will begin with a subject.

### 3.4.2.1 Equational Sentences

An equational sentence consists of two main parts: a subject and a predicate phrase. The subject is an equational phrase NP and the predicate could be noun phrase (NP), adverbial phrase (ADVP), adjectival phrase (ADJP), prepositional phrase (PP) or Complement Phrase (CP), as shown in the examples below. The subject is usually definite and the predicate is indefinite which it may shift from definiteness to indefiniteness that marks the transition from subject to predicate. When the predicate is a noun or an adjective it has to agree with the subject in number and gender.

The following examples show all the Arabic equational structures.

\begin{equation}
\text{الجُوَّار } Aljw HAr \quad \text{‘The weather is hot’}
\end{equation}

NP AdjNP (Verbless nominal sentence)
Chapter 4: Classification of Learner Errors

(3.4) 
\[
\text{‘on the bookshelf is a book’}
\]
\[
\text{ElY Almktbp ktAb}
\]

Prepositional phrase

(3.5) 
\[
\text{‘he is clever’}
\]
\[
\text{hw vky}
\]

AdjNP

(3.6) 
\[
\text{‘he is a clever boy’}
\]
\[
\text{hw wld vky}
\]

ProNP NP AdjNP

(3.7) 
\[
\text{‘The man is in the office’}
\]
\[
\text{Alrjl fy Almktb}
\]

NP PP NP

(3.8) 
\[
\text{‘It’s weather is hot’}
\]
\[
\text{HAr jwh}
\]

AdjNP NP

The examples above show that the predicate phrase does not always have to follow
the subject phrase, as in (3.4) \text{ElY Almktbp} represent the prepositional phrase. There are many constrained or forced instances where the predicate phrase can precede
the subject. Thus, nominal phrases come in two basic types of NP in Arabic, nominal
and pronominal. Nominal NPs have a variety of head as a noun like (3.5), a modified
noun like a construct phrase as in (3.6), or both for the pronoun as an adjective phrase as
its head which is shown in example (3.8) (Abdul-Raof, 1998).

In general, the above structure of NPs can be rewritten by phrase structure rules as
shown in the example predicate:

(3.9)
\[
\text{NP } \rightarrow \text{ Pro}
\]
\[
\text{NP } \rightarrow \text{ N(Mod)}
\]
\[
\text{Mod } \rightarrow \text{ NP(+gen)}
\]
\[
\text{Mod } \rightarrow \text{ ADJ}
\]
\[
\text{Mod } \rightarrow \text{ PP}
\]
3.4.2.2 Verbal Sentences

Verbal sentences or non-equational sentences are sentences where a non-copula verb functions as the main predicator in the construction. Thus, Arabic verbal sentences are those sentences that start with a verb. The following structures start with a verbal constituent and are thus, classified as verbal sentences:

\[(3.10)\]  
\(n\text{A}m\ AlTfl\)  
\(\text{The child slept’}\)  
\(V\ NP\)

\[(3.11)\]  
\(d\text{r} AlTAlb\ Ald\)  
\(\text{The student studies the lesson’}\)  
\(V\ NP1\ NP2\)

\[(3.12)\]  
\(d\text{r}~s\ Al\text{md}~s\ AlTAlb\ Ald\)  
\(\text{The teacher taught the student the lesson’}\)  
\(V\ NP\ NP\ NP\)

\[(3.13)\]  
\(lE\ AlTfl\ fy\ AlHdyqp\)  
\(\text{The child played in the garden’}\)  
\(V\ NP\ PP\)

\[(3.14)\]  
\(r\text{O}y\ Alwld\ Alrjl\ fy\ AlHdyqp\)  
\(\text{The boy saw the man in the garden’}\)  
\(V\ NP1\ NP\ PP\)

\[(3.15)\]  
\(I\text{tqd} Alrjl\ On~ AlTfl\ d\text{Xl\ Albyt}\)  
\(\text{The man thought that the child was entering the house’}\)  
\(V\ NP\ (\text{COMP\ NP\ V\ NP})\)

\[(3.16)\]  
\(O\text{X} AlTAlb\ y\text{dr} s\ Ald\)  
\(\text{The student started to study the lesson’}\)  
\(V\ NP1\ (V\ NP)\)
The examples mentioned above have shown different verbal constructions, which differ according to the subcategorisation frame of a given verb. Some verbs can be intransitive that require only a subject and transitive requiring one object or ditransitive requiring two objects. A third type of verb may subcategorize a whole sentence.

There are generally four accepted word orders: VSO, SVO, SOV, OVS and VOS,

\[(3.17)\]

\[\text{الرجل دخل البيت} \quad \text{Alrjl dXl Albyt} \quad \text{the man entered the house} \]

\[
\text{NP V NP} \quad \text{Verbal sentence S O V}
\]

\[(3.18)\]

\[\text{البيت دخل الرجل} \quad \text{Albyt dXl Alrjl} \quad \text{the house entered the man} \]

\[
\text{NP NP V} \quad \text{Verbal sentence S O V}
\]

\[(3.19)\]

\[\text{دخل البيت الرجل} \quad \text{dXl Albyt Alrjl} \quad \text{entered a man} \]

\[
\text{V NP NP} \quad \text{Verbal sentences V O S}
\]

\[(3.20)\]

\[\text{البيت دخل الرجل} \quad \text{ح Albyt dXl Alrjl} \quad \text{the house entered a man} \]

\[
\text{NP V NP} \quad \text{Verbal sentence O V S}
\]

\[(3.21)\]

\[\text{دخل الرجل البيت} \quad \text{dXl Alrjl Albyt} \quad \text{entered the man the house} \]

\[
\text{V NP NP} \quad \text{Verbal sentence V S O}
\]

3.4.3 Word Order and Agreement

Since Arabic has rich agreement morphology, it allows showing agreement relations between various elements in the sentence. There are five morpho-syntactic features involved in agreement in Arabic: number (singular, dual and plural), person (1st, 2nd and 3rd), gender (feminine and masculine), definiteness (definite and indefinite) and case (nominative, accusative and genitive). The strongest relation of agreement is that between the nouns and the adjectives where four of the five agreement features are involved. These features are number, case, gender and definiteness. Examples (3.17)-(3.21) show the differences of word order. However, when the subjects appear in the
sentence after the verb, the verb shows partial agreement. In this case, verbs agree with their subjects in gender and person only, as in (3.22) - (3.26). Verbs take the default singular form whether subjects are singular, dual or plural as in example (3.27).

الطالبان اللذان ذهب (3.22) AlTAlbAn All*An *hbA  
(noun – relative pronoun: number, gender, case) ‘The two students who went’

رأيت الرجلين الطويلين (3.23) rOyt Alrjlyn AlTwylyn  
(noun-adjective: number, gender, case, definiteness) ‘I saw the two taller men’

البنات درسنَن دروسهن (3.24) AlbnAt drsn drwshn  
(noun – pronoun: person, number, gender) ‘The girls have studied their lessons’

ذهبت الطالبات الى المدرسة (3.25) *hbt AlTAlbat IIY Almdrsp  
(verb, gender, number, preposition, noun, case) ‘The students went to the school’

البنات ذهبن الى الجامعة (3.26) AlbnAt *hbn IIY AljAmEp  
(noun – pronoun: person, number) ‘The girls went to the University’

ذهبت البنات الى الجامعة (3.27) *hbt AlbnAt IIY AljAmEp  
(verb, gender, subject, number, case) ‘The girls went to the university’

As shown in the examples below that, if the subjects are in the post-verbal position, the verbs show poor agreement. In this case, the verbs agree with their subjects in gender and person only. Verbs take the default singular form whether subjects are singular, dual or plural.

In Arabic the feature of humanness shows an important role in agreement. With non-human plural nouns, verbs are always in the singular and feminine, as shown in example (3.28) below.
الطيور تأكل الحبوب (3.28) 

الطيور تأكل الحبوب (noun, non-human, verb, gender, case, number) ‘The birds eat the seeds’

Ryding (2005) clarifies the agreement by defining it as ‘the feature compatibility between words in a phrase’. Corbett (2001) defines agreement as systematic covariance between a formal or a semantic property of one element and a formal property of another element.

3.4.4 External Governors

External governors in Arabic are the particles that can precede the subject and change its case to the accusative while the predicate remains in the nominative case. External governors are seven particles, إن In~ ‘surely’, أن On~ ‘that’, لكن kOn~ ‘as if’, ليت lyt ‘if only’, لعل lEal~ ‘perhaps’ and لا lA ‘no’ (Alsharif and Sadler, 2009; Attia, 2008; Holes, 2004; Hoyt, 2007; Ryding 2005; Hassan, 2007).

We decided to add study of this for our learners as it is one of the complicated issues for the learners’ grammar skills. Table (3.14) shows the criteria of the differences and the similarities between the negation and other components. These items set three kinds of constraint: they specify the word order of the governed clause as illustrated in the examples below:

إعتقد المدرس أن الطالب يكتب الدرس (3.29) 

إعتقد المدرس أن الطالب يكتب الدرس (IEtqd Almdrs On AlTAlb yktb Aldrs) ‘The teacher thought that the student wrote a lesson’

أَنَّ الطالب يكتب الدرس إعتقد المدرس (3.30) 

أَنَّ الطالب يكتب الدرس إعتقد المدرس (On~ AlTAlb yktb Aldrs IEtqd Almdrs) ‘The teacher thought that the lesson is written by the student’

As seen in the examples above, the complementisers are selected by the main verb of the governing clause, such as the verbs إعتقد IEtqd ‘to think’ and أَعْرِف أَعْرِف ‘to know’; and they in turn specify the word order, mood and case of the subject of the governed clause, as shown in examples 3.31 and 3.32 below:
Negation particles in Arabic use five different particles to express the negation: the particle ما $mA$, لا $lA$ (and its tensed equivalents لَم $lm$, ليس $lys$ and لن $ln$). ليس $lys$ is marked only for subject agreement. Arabic sentences always treat all these particles in the same manner, as negation heads undergoing head movement (Shlonsky, 1997; Benmamoun, 2008). However, there is a different analysis between لا $lA$, لَم $lm$ and لن $ln$ and for forms of ليس $lys$. لا $lA$, لَم $lm$ and لن $ln$ show tense distinctions that occur only with imperfective forms of the verb (excluding the perfective). The individual negation markers are associated with different tenses, but they all require the subjunctive mood.

Table 3.14 below shows the particles $mA$ and $lA$ are able to move freely in the sentence and that will change the meaning of negation by focusing on the item that is placed next to it.
Table 3.14: Arabic Particles with their features

Table 3.14 above illustrates how different particles can be in a canonical order or not with specific tenses. Our learners find it very difficult to understand the rules that govern behaviour of these particles. Therefore, exercises have been set to the aRaBCALL system to treat these particles after inserting their grammatical rules in Parasite.

3.5 Case study and Error Treatment

This section has defined some of the key syntactic points surrounding verbs, nouns and particles that pose difficulties for students. We are going to outline the specification of a set of exercises with the grammatical aspects. The exercises are divided into three levels basic, intermediate and advanced. The following types of errors were diagnosed in students’ work which will be treated with using the aRaBCALL system:
1. Misused Agreement marker
2. Misused Gender Marker
3. Misused Tense Agreement
4. Misused Word Order
5. Misused Negation
6. Misused Complementisers

We will show how to detect and diagnose these errors in chapter 6.

3.6 Summary

In this chapter, we first presented some theoretical background of the considered Arabic morphology, lexical and syntax. We presented case studies covering most of the important Arabic syntactic structures such as the phrases, sentences and especially free-word order and agreement. We discussed the sources of syntactic ambiguity of the language. Finally, we devoted a separate section to discuss the problems that learners of Arabic face when learning the language. In the next chapter, we will review various approaches to classifying learner errors: many of these approaches consider a wide range of error types, but as noted above the major problems for the students we have worked with relate to issues in morphosyntax. We will therefore return to the error list given above when we present our own classification at the end of Chapter 4.
Chapter 4

Classification of Learner Errors

4.1 Introduction

This chapter will discuss the present problems in language learning and teaching. The main objective of this research is to develop an ICALL tool for learners of Arabic as a foreign language that supports their learning by providing online exercises. The exercises consist of sentences that may or may not contain errors, which the student must identify. These types of exercises are produced to consolidate the students’ learning mainly in the grammatical field. A detailed study on types of errors and the pedagogical view will signify the different classification of learner errors, concentrating on morpho-syntactic errors. The chapter will contain the following points:

- We will represent the historical perspective on learner errors.
- We will talk about error diagnosis, classification of learner errors with an emphasis on morphological and syntactic errors, error taxonomies, and the classification by James (1998); Tokuda and Chen (2004); Taylor (1998).
- We will focus on English speaking learners of Arabic for error data.
- Then, we will create the relevant picture of the types of errors that will be the subject of our implementation in chapter 6.
4.2 A Historical Perspective of Error Analysis

James (1998) explains that the study of learners’ errors was at its pinnacle in the 1970s which did not last very long (Richards 1974; Corder 1981). A special sort of dialect, special in some rules, is held in common with speakers of the other language (Heift and Schulze, 2007; Izumi et al., 2005). Error analysis was strictly devoted to the determination of incidence, causes, nature and the consequences of error which helped create a decrease in learner’s errors. However, in the field of CALL there has been a recent increase of attention towards error description and classification as well as error correction and feedback (Chen and Tokuda, 2002; Heift and Schulze 2007; Izumi et al., 2005; Tschichold, 2003; Tokuda, 2002). Here, we try to briefly mention major classifications of learners’ errors in the field.

4.3 Error Diagnosis

Identifying second language errors is crucial for any error studies for theoretical and/or pedagogical purposes. However, the very act of detecting such errors seems to be intuitive and hence inexplicable. The only systematic and imitable method available is probably the error identification procedure suggested by Corder (1973).

Corder (1974) differentiates between mistakes, lapses and errors. Out of these, errors solely result from a lack of competence on part of the second-language learner. Lapses and mistakes are performance-based, under certain adverse conditions like stress or fatigue causing their number to increase. As far as computer-aided text production is concerned, typographical mistakes should also be included in the latter two categories (Hamid, 2007). Lapses, mistakes and errors are all deviations from the norm and as such will be detected by the parser, and the learner will be notified about them. The difference ought to be in the feedback the learner receives. An error caused by the lack of competence calls for some sort of support to build competence up in this area. On the other hand, a mistake can easily by corrected by the learner without the need of an explanation in the feedback.
Analysing learner errors is important for teachers, researchers and learners themselves in the following way (Izumi et al., 2005; Corder, 1981). First, for teachers, errors can give them hints about the extent to which the learners have acquired the language system by that time and what they still have to learn. In fact, for learners themselves, making errors is one of the most important learning strategies for testing the inter-language hypothesis that learners have established about the (Izumi et al., 2005).

It has been pointed out by a number of error analysts that the notion of error is highly subjective (Meunier and Granger, 2008). The subjectivity of a parser is indirect as it cooperates with the notion of the parser grammar to be the grammar of an informed idiolect, on which the judgment of well-formedness is based. The fact that the parser will only look at grammatical errors causes it to be the problem. This causes errors of other classes to be overlooked. For example, Corder (1974) separates errors of well-formedness from errors of appropriateness which he distinguishes into four major error classes.

- Social error (addressing a teacher as ‘old man’)
- Textual errors (not selecting the structurally correct form to express the intended relation)
- Referential errors (calling a hat a cap)
- Register errors (referring to a ship as a boat in a naval context)

A useful way to understand the limitations of a morpho-syntactic parser is through Corder’s distinction of superficially ill-formed sentences and errors of appropriateness.

Recognition, description and explanation are the three stages of error analysis (Corder, 1973). A parser that is attached with a feedback module can carry out both the first two stages of error analysis, recognition and description. However, explanation is hard for any analyser and entirely impossible for a parser that only uses morpho-syntactic knowledge, but does not have sufficient information about
other linguistic levels, the situational context and the world in general. This information is necessarily used to determine a plausible and well-formed statement on the basis of a corresponding ill-formed one. The ill-formed sentence can be related to a well-formed counterpart by a parser only on the basis of the violated constraint. Therefore, the violated constraint is the underlying reason for the error.

### 4.4 Error Classification

There have been various approaches to the classification of learners’ errors. Here, we will mention some briefly. All approaches are common in trying to classify errors based on their similarities and differences. However, since their approaches are different, they end up with different classifications, with some degree of overlap. Some of the dimensions which these authors classify errors are outside the scope of the current work (e.g. semantic errors of various kinds), but we will draw on their insights in our own classification in section 4.5.

#### 4.4.1 Error Description and Taylor’s Classification

Mistakes are caused by a lack of attention or carelessness, whilst errors are the use of linguistic items in a manner that a learner of the language regards them as showing incomplete or faulty learning (Abushihab, 2011). This occurs because the learner is not aware of their mistake or is not knowledgeable enough to acknowledge what is correct, and thus, errors and mistakes made are unable to be self-corrected. Chaudron and Richard (1986: 95) described errors as: “the use of a linguistic item in a way which a fluent or native speaker of the language regards as showing faulty or incomplete action”. Gass and Mackey (2006) stated that errors are systematic; they occur repeatedly and are not recognized by the learner. Hence, only the researcher or teacher was capable of locating them, the learner could not.
Chapter 4: Classification of Learner Errors

An appropriate error classification is the important condition for adequate description of errors and the generation of useful feedback (Schulze, 2001). The use of an error classification has the following advantages:

1. Error description and feedback can be established on both the individual error and the error class, which will result in a consistency of feedback and description as well as computational efficiency, because a proportion of description and feedback is identical for all errors of one class.

2. Errors that belong to error classes do not have to be treated as though they were idiosyncratic, they can be described and feedback can be forwarded on the basis of how frequently the learner has made errors that belong to the same error class.

Taylor’s error classification consists of four axes (Taylor, 1998):

1. **Presystematic or systematic** axis is based on a distinction made by Corder (1974). The learner is unaware of a particular rule in the target language, so the system uses the presystematic axis for errors that occur in an unsystematic way. This is normally the case in instructed second language acquisition before the rule has been introduced in the course of the curriculum. When the learner relies systematically on an interlanguage rule, which is not part of the target language rule system, it is categorized as the systematic stage.

2. **Transfer or simplification** axis relies on the classification of errors by inter-language processes (Selinker, 1992). The problems related with this kind of classification are well known. One reason for the discrepancy in research findings has been the issue of error classification. However, "The assignment of a particular error to such categories as 'transfer', 'overgeneralization', 'ambiguous' has been largely an arbitrary matter, subject to the individual biases and point of view of the researcher" (Ellis, 1994: 61).

   The main issue regarding this axis for the classification is that the class-determining features are built on the explanation processes as part of the error
analysis and that these processes are, by definition, reliant on the researcher who performs the analysis. However, this method of explanation is useful due to its predictive power; it predicts the underlying cause for the error and delivers beneficial information for feedback.

3. **Surface-level** linguistically based axis relies on a classification of errors by Part of Speech (POS) as well as Error Heurisms (ER) (insertion, omission and invention).

4. **Systematicity** is the fourth axis stating that all types of errors are initially classed as postsystematic and limited feedback is provided. For the latter, a category of inappropriateness is assumed by Taylor (1998). Here, invention and appropriateness are going to be incorporated under invention, essentially because a differentiation between inappropriateness and invention does not appear to be very useful. Taylor uses 'inappropriateness' if a word form was used that exists, but renders the current grammatical structure as shown in the example أنا أكلت تفاحة 'I eat an apple', and 'invention' if the word form does not exist at all such as هو رمي الكرة ‘he threw the ball’. However, both display a very similar kind of error.

### 4.4.2 Classification by Tokuda and Chen

Tokuda and Chen (2004) studied detailed error classification and error-specific messages. They indicated the errors carried out by students that are vital for curative purposes for achieving an effective CALL system.

Tokuda and Chen gave out an invitation to 200 students, who were native bilingual language teachers, to compose English translations of given Japanese sentences by using assigned key patterns in the text which was prepared for them to read before doing the translation exercises. A reasonable number of well-formed model translations were prepared of the given source sentence by reviewing the study participants’ responses. Then they selected one model translation and followed this by analysing the output. The
participants' responses were then classified into a possible combination of types of errors. Finally, the assigning of ill-formed translations or and the completion of a transition network of nodes comprising a model translation and possible types of errors was carried out.

As a result of the analysis Tokuda and Chen classified the errors into three categories:

- **Grammatical errors**, which are grouped into nine subgroups having a similar hierarchical structure ending up with a total of 164 errors at the leaf level: determiner, punctuation, noun, subject-verb, tense, modifier, preposition, negation and conjunction.

- **Word and usage errors** which are also classified into nine groups: wrong word, awkward word, wrong meaning, misspelling, technically wrong singular form, plural form, sexist and colloquial.

- **Non-preclassified errors** are all of the possible errors made by students that cannot be dealt with.

Tokuda and Chen’s (2004) investigation shows how accurately errors can be diagnosed and the error contingent messages can be presented to the learners.

### 4.4.3 Dulay et al. classification

In addition to the previous classifications there exists another type of error collection called error taxonomy (Dulay et al, 1982). The following features which are typical of error taxonomies distinguish them from other types of classifications i.e., the dictionary:

- Taxonomies are organized according to certain constitutive criteria such as the sex, age or nationality of the learner who has made the error.

- The criteria are not necessarily mutually exclusive, i.e., it is possible to classify errors simultaneously according to a number of criteria. So, it will be possible to specify an error in terms of a large number of features.
Chapter 4: Classification of Learner Errors

- The criteria selected may be binary or may allow more than two grades of the attribute. For instance, a learner may be female or male (binary), but might have any of a score of nationalities.

As a result, a set of criteria can lead to feature taxonomy. The categories used for specifying the types of linguistic deviance are:

- Ambivalence
- Dissonance
- Ellipsis
- Idiom
- Coherence
- Incongruence
- Redundancy

Below, we will briefly mention the two kinds of error taxonomies proposed by Dulay et al. (1982). They mainly deal with error causes and error gravities respectively.

4.4.3.1 Linguistic Category Classification

This type of taxonomy is based on “the linguistic item that is affected by the error” (ibid: 146). The following issues should be taken into account in this classification:

- Level: indicates on what level the language error is located. Possible levels are phonology, graphology, grammar, lexis, text or discourse.
- Class: if the error is decided to be a grammatical one, it should be determined whether it relates to the class of nouns or adjectives, etc.
- System: finally, we should specify the grammatical system that the error affects.

The point stated in which linguistic category classification is based on the item affected by error will be illustrated using the following sentence:
‘She go to school every day.’ The error is related to the misuse of goذهب instead of goesذهب. Thus, the above error can be classified as a grammar level error involving the word class verb, and the system of tense. The above classification can be useful; however, there is no clear-cut borderline between the levels.

4.4.3.2 The Surface Structure Taxonomy

As stated by the authors, it is a taxonomy based on “the ways surface structures are altered” (Dulay et al., 1982: 150). They follow by identifying four principle ways in which learners modify target forms; in other words, four ways in which inter-language and target language forms diverge “in specific and systematic ways” (ibid: 150). In addition to the four ways, there exist four additional subtypes, which is a total of eight. We will present these main ways briefly since they are not the main focus of our investigation:

- **Omission**, which is when learners omit elements that should not be deleted. This should not be confused with ellipsis and zero elements which are totally grammatical. Based on Dulay et al. (1982) it tends to affect function words rather than content words.

- **Addition**, which is the result of all-too-faithful use of certain rules. It is suggested that there are three subtypes in which they can be categorized. Firstly, regularization, which involves overlooking exceptions and spreading, rules to domains where they do not apply. Secondly, double marking which is defined as “failure to delete certain items which are required in some linguistic constructions but not in others” (ibid: 156). Lastly, simple addition which includes all the additions which cannot be classified as double markings or regularizations (Burt et al, 1982).
• **Misformation**, which refers to the use of the wrong form of a structure or morpheme. In other words, this kind of error produces words that are non-existent in foreign language (FL). Again, they can originate either in mother tongue or be created by the learner from the resources of the TL itself. The examples below represent these kinds of errors:

‘هو ذهبا الى المدرسة’ or *these book has been read by her*.

There are, like in addition, subtypes in which they can be categorized, an example of which is alternating forms that are defined as regularization and fairly free alternation of different members of a class with each other.

• **Misordering** which is related to word order in the language and how the word order in two languages differs due to how strict the word order in the language is. For example, Persian has freer word order compared to English; as a result, students may use word orders which are not allowed in English.

### 4.4.4 Levels of Errors and James Classification

James (1998) provided the classification of errors that were made by learners. The study was a reference to three criteria: modality, medium and level. Modality refers to whether the learner’s behaviour was productive or receptive. Medium indicates whether the language produced or received was spoken or written. James defines his method as follows:

He recognized three levels of language: the levels of substance, text and discourse. If the learner was operating the phonological or the graphological substance systems, that is spelling or pronouncing, we say he or she has produced an encoding or decoding error. If this was operating on the discourse level, we label the errors occurring as misformulation or misprocessing errors. These six types are then differentiated further according to the medium of each. This results in a classificatory system of errors, which has **twelve** categories (James, 1998).
4.4.4.1 Substance Errors - Misspelling

James (1998) defines a misspelling as “a substance level production error”. There are various different types of substance errors but they are referred to as ‘mechanical errors’. There are four types of misspelling errors listed as follows:

- **Punctuation errors** - characteristic errors include overuse of the exclamation; misordering of closing inverted commas; underuse of apostrophe, under or overuse of capitals, etc.

- **Typographic errors** – These are spelling errors that occurred while using computer keyboards. Psychologists (MacNeilage, 1964) have been attracted more by these errors than other types of researchers.

- **Dyslexic errors** – Misordering and misselection of letters that produce the same sound are among the common errors made by dyslexic people.

- **Confusables** – Words that are mistakenly used by people due to their similarities. Typical examples include advice and advise or accept and except.

4.4.4.2 Text Errors

James provides the definition of the term ‘text’ as “any instance (or instantiation) of language that results from applying the rules of encoding and of lexico-grammar” and also states that “text is usage.” (James, 1998: 142). He makes a distinction between lexical and grammatical system and discusses them separately.

4.4.4.3 Lexical Errors

James (1998) believes that due to Chomsky’s influence upon linguistics, the attention given to grammar and vocabulary has been largely neglected by linguists. However, he
claims that lexis has begun to take a central role in language study. He states the following reasons for such a claim:

- There is no clear-cut boundary between vocabulary and syntax as was believed by people in the past.
- Language learners put a lot of emphasis on vocabulary learning, comparing it with the language itself.
- The majority of errors are related to vocabulary for some learner groups.
- Lexical errors are regarded as more disruptive and irritating by native speakers.
- Vocabulary has a higher functional capacity especially in the early stages of language learning.

4.4.4.3.1 Classifying Lexical Errors

James divides lexical knowledge into formal and semantic features and justifies it into such a dichotomy.

**Formal** errors of lexis are divided into three different subsections, which are stated as follows:

1. **Formal misselection** - This category includes pairs of words that look and sound similar: parricide/patricide.
2. **Misformations** - These errors are non-existent in the target language.
3. **Distortions** - These are the errors due to the misapplication of one or more of the processing operations discussed in the previous section namely omission, misselection and blending.

When James (1998) applies the term **semantics**, he mentions two main types of such errors confusion of sense relations and collocational errors, which are both discussed briefly below:
Chapter 4: Classification of Learner Errors

- **Confusion of sense relations** - Lexicologists describe vocabulary items in terms of lexical systems by reflecting the meaning relations existing between words. The clustering devised by them is often referred to as lexical fields.

- **Collocational errors** – These can be classified into being semantically determined, statistically weighted preference and arbitrary combinations.
  
  - **Semantically determined**: for instance, it is correct to say crooked stick but not crooked year since in the real world, years cannot be crooked.
  
  - **Statistically weighted preference**: for instance, it is fine to say big losses and heavy losses for an army; however, statistically the latter is preferred to the former (although it may not have been in 1998).
  
  - **Arbitrary combinations**: for instance, it is feasible to say make an attempt and have a try but not have an attempt and make a try.

4.4.4.4 Grammar Errors

James (1998) divided grammatical errors into either morphological or syntactic errors.

4.4.4.4.1 Morphology Errors

There are five lexical word types that James specifies in English: noun, verb, adjective, adverb and preposition. Therefore, he defines a morphology error as “one which involves a failure to comply with the norm in supplying any part of any instance of these word classes...” (James, 1998: 154). Examples of the different types of morphology errors are as follows: four **book** is a noun morphology error; **bringed** is a verb morphology error; **come see me soonly** is an adverb morphology error; and a **colourfuller sky** is an adjective morphology error. The only one which does not contain morphology is prepositions.
4.4.4.4.2 Syntax Errors

James (1998) explained that there are five syntactic classes of error due to several problems due to there not being as many phrase types as there are lexical word types. Errors that affect texts larger than the word namely phrase, clause, sentence and paragraphs are considered as syntax errors.

1. **Phrase structure errors** – Firstly, these are not discrete entities since one can find NPs inside PPs inside further NPs. Secondly, the problem is that every phrase contains an eponymous nucleus or head, a noun heads a NP, a verb heads a VP and an adjective heads an ADJ. If there was an error on this head and the heads are words, a head-located error would be morphological by definition. Lastly, there is an issue with modern syntactic theory and the tendency to multiply the number of phrase types by the lexical word types (ibid).

2. **Clause errors** – James (1998: 157) explains that “While phrase errors involve violations in the internal (or external) relations between parts of phrases, clause errors involve whole phrases entering into the structure of clauses”.

3. **Sentence errors** – These involve the selection and combination of clauses to create large units. All clauses can be blended, as in an essay’s sentence for example, ‘that it is only …that we can …’ and ‘that only …can we’ (ibid), which is the type of error that argues the difference between both complement clauses. There is another kind of sentence error called consolidation errors, which comprise the two subtypes, coordination, and subordination errors. A coordination error has a specific rule stating that only syntactic equals can be joined, and attempts to conjoin unequal constituents lead to broken coordination.

4. **Receptive errors** – These come from the listener’s misunderstanding of what the speaker’s intentions are. To investigate receptive errors, one would need to investigate people’s reactions to requests, orders, etc. An example of a receptive error would be mistakenly hearing the word ‘lounge’ when the word that was said was ‘lunge’
From the wide range of errors discussed above, we have decided to concentrate most closely on morphological and syntactic aspects of the language. This is partly because diagnosing semantic errors requires a huge knowledge base, including lexical knowledge and wide-ranging background knowledge. More importantly, however, from what we know of the British syllabus, the key field that the student should be competent in is grammar. This is backed up by our teaching experience from which we realise that students make the most mistakes in grammar.

In this section, we focus on our learners’ errors. We found errors based on our own intuition, as well as experience gained from years of teaching Arabic as a second language, resulting in our assumption that grammar is the main field of concern when learning Arabic as a foreign language. Furthermore, native speakers also come across a number of grammatical problems while learning the Arabic language (Abushihab, 2011; Dawood, 2009). Therefore, learners of Arabic need to be provided with more support to reduce the errors that they make, especially within the speaking and writing aspects of Arabic. In order to provide support for our learners, we firstly need to classify our learners’ errors at all levels of study.

The aRaBCALL error classification groups errors into three main classes, as follows:

1. **Word formation errors** – As explained in section 3.3.2, word formation occurs when Arabic word forms are joined with various vowels and consonants. In other words, Arabic words are built from the roots of the word rather than the stem due to the complex morphology of Arabic. Furthermore, diacritics that help in the pronunciation of words are usually omitted in modern writing. Arabic morphology is complicated due to it being a highly inflectional language and the fact that Arabic written forms omit short vowels and a number of other marks. Stems are formed by a derivational combination of a root morpheme and a vowel melody; the two are arranged according to canonical patterns, suffixes such as nominal case and verb endings, and the nominal feminine
ending with the letter [ة، پ] تاء مربوطة `marbuwTap. Moreover, the verb classes are likely to use number, gender and person and the noun class uses number and gender.

This level of error classification relates to the word formation errors that can occur when adding, for example, a feminine marker [ة] taa mrbwtp to a masculine word, resulting in a feminine noun case marking error. It can also occur when adding the wrong verb tense marker, such as adding a present prefix with a past suffix to a verb. In some cases, adding incorrect present or past affixes to the verbs is like treating a weak verb as though it were an ordinary verb. Students often make mistakes with complement pronouns, which are generally enclitics, and English native speakers are not used to such items and are mostly frequently confused by the use of masculine or feminine plural or a broken plural. Table 4.1 below shows examples of incorrect word formation.

<table>
<thead>
<tr>
<th>Ill-formed word</th>
<th>Correct form</th>
<th>Type of error</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>تشربت</td>
<td>شربت</td>
<td>Present prefix with past suffix</td>
<td>To drink</td>
</tr>
<tr>
<td>كتابون</td>
<td>كتب</td>
<td>Third person masculine plural ending added to broken plural</td>
<td>Books</td>
</tr>
<tr>
<td>يسار</td>
<td>يسير</td>
<td>Weak verb with wrong tense marker</td>
<td>To walk</td>
</tr>
<tr>
<td>ولدة</td>
<td>بنت</td>
<td>Feminine marker on a masculine word</td>
<td>Girl</td>
</tr>
<tr>
<td>نلعبن</td>
<td>لعبن or لعبن</td>
<td>Verb with wrong tense marker</td>
<td>To play</td>
</tr>
<tr>
<td>يكتب</td>
<td>يكتب</td>
<td>Masculine present prefix and feminine past suffix</td>
<td>To write</td>
</tr>
</tbody>
</table>

Table 4.1: Types of word formation errors
Table 4.1 above shows the types of errors that can be made by students during learning and how these can be dealt with. These can be marked for morphological ill-formedness or case marking error. aRaBCALL aims to provide detection of morpho-syntactic errors and word order problems in our current investigation.

1. Local feature mismatches – Some words are required by the local context to have a specific set of features; for example, adjectives should have the same definiteness marker as the noun, and subjects should agree in gender and sometimes number with their verbs. This is shown below in table 4.2.

<table>
<thead>
<tr>
<th>Wrong sentence</th>
<th>Correct sentence</th>
<th>Type of error</th>
<th>Gloss</th>
</tr>
</thead>
</table>
| هذه قلم سوداء  
\textit{h*A qlm swdA'} | هذه قلم أسود  
\textit{h*A qlm Oswd} | Adjective put constraints on gender marker | This is a black pen |
| حضر المدرسين  
\textit{HDr Almdrsyn} | حضر المدرسون  
\textit{HDr Almdrswn} | Subject should be nominative and the case marker is accusative | The teachers arrived |
| هذه سيارة حمراء  
\textit{H*h syAvp AlhamrA'} | هذه سيارة حمراء  
\textit{H*h syAvp hamrA'} | Definite adjective with indefinite noun | This is a red car |
| هذه رجال طوال  
\textit{h*p rjAl TwAl} | هؤلاء رجال طويل  
\textit{h&lA' rjAl TwAl} | Demonstrative pronouns put constraints on gender marker | These are tall men |
| هذه كتاب كثيرة  
\textit{h*p ktb kvyrp} | هذه كتاب كثيرة  
\textit{h*h ktb kvyrp} | Adjective put constraints on word number | These are lots of books |
| لاذهب الولد الى المدرسة  
\textit{IY y*hb Alwld IY Olmdrsap} | لا يذهب الولد الى المدرسة  
\textit{IY *hb Alwld IY Olmdrsap} | Negation puts constraints on verb tense | The boy does not go to school' |

Table 4.2: Types of local feature mismatches
3. **Word order errors** – As explained in chapter 3, Arabic syntactic structure has multiple complexities. However, in the Arabic grammatical tradition, there are two distinguishing classifications of sentences: verbal and nominal. Verbal sentences are those that begin with a verb, whereas a nominal sentence begins with a noun. It should be noted that under this classification a nominal sentence might contain a verb so that a sentence in SVO order is classified as nominal. Since Arabic has rich agreement morphology, it allows agreement relations between various elements in the sentence to be shown. As we illustrate in chapter 3, Arabic has five morpho-syntactic features involved in agreement: person, definiteness, number, gender and case. The strongest agreement relation is that between the nouns and their adjectives where four of the five agreement features are involved. These features are number, case, gender and definiteness. Moreover, when the subject appears in the sentence after the verb, the verb shows partial agreement, in this case, agreeing with their subjects in gender and person only. Verbs take the default singular form whether subjects are singular, dual or plural, as shown in section 3.4.3. Therefore, students most often misuse these five syntactic features, for example when the adjective should have been matching the noun in gender and number. In Arabic we can see differences of word order when negation or complementisers are used; however, we devised constraints for all expected misuse of these formations. Generally speaking, despite Arabic sentences being of free word order, a noun at the beginning of the sentence must match the number of the verb, indefinite nouns cannot be at the beginning of the sentence and explicit pronouns must be in front of the verb. Examples of these are shown below in table 4.3.
### Table 4.3: Types of word order errors

aRaBCALL provides feedback to the learner once the error has been found or if there is insufficient information provided. Chapter 6 covers all the manipulation procedures related to aRaBCALL and in the following sections we demonstrate other prototypes for the classification of errors.

On the basis of the students' results, we have included some pedagogical implications for teachers, syllabus designers, textbook writers and text developers.

As the above discussion reveals, morphological and syntactic errors are the widest category of learners’ errors. Therefore, in order to deal with the morphosyntactic errors made by learners of Arabic, support must be provided in order to assist them with correcting their mistakes.
4.6 Summary

In this chapter, we considered a number of approaches to error classification, before proposing our own. The general discussion covered a wide range of error types. However, given our understanding of our students’ needs, our classification concentrates on providing a set of broad groups of morphosyntactic errors. We also addressed how we could deal with our learners’ errors. In the upcoming chapter, we discuss the computational treatment of Arabic morphosyntax within the framework of Parasite, based on HPSG and CG.
Chapter 5

Arabic Morphosyntax within the Parasite Framework

5.1 Introduction

The previous chapter discussed the historical perspective with the classification of all learner errors, a detailed explanation of the error taxonomies, James’ classification of language learning errors and a detailed survey of the kinds of errors that learners of the language can make. A narrowly focused study on the morphological and syntactic errors that learners of Arabic may come across has been previously looked into in chapter 4. This chapter is dedicated to the computational description of Arabic morphosyntax. It traces the roots of the system’s morphological analysis, which is related to the case markers, gender and clitics in CG (Bauer, 2001; Ramsay and Mansour, 2001) and describes how this is encoded in the programming language Prolog. Similarly, the relationship between HPSG (Pollard and Sag, 1994) and the parser Parasite’s syntactic analysis (including a number of modifications) is explained next, followed by a description of the implementation in the Parasite system.

5.2 Parasite Overview

We have chosen Parasite (Ramsay and Mansour, 2001), which has been used previously to analyse numerous languages such as Arabic, as a NLP system because it is almost entirely constraint-based, so that any kind of error can be dealt with uniformly by relaxing constraints. Parasite produces morphological, syntactic and semantic analyses
Chapter 5: Arabic Morphosyntax within the Parasite Framework

of textual input and is employed to develop the system-generated error detection and diagnosis that can be seen in chapter 6.

Although Parasite contains rules that describe the morphological and syntactic component of MSA, we require a much more detailed description of the dictionary, so we will extend it with adjectives, adjectival nouns and proper nouns denoting nationalities. The base entry that contains these words has been defined by Ahmed (2005). To implement the soft parsing techniques, our CALL system relies on the morpho-syntactic analysis. Parser-based CALL has been used before for similar CALL purposes for the different languages German, Persian and French (Schulze, 2001; Mirzaeian, 2003; Lusuardi, 2007). A detailed analysis of the existing computational Arabic parser will be described in the following sections. The morphological part of the parser is in CG and the syntactic component is in a modified form of HPSG (Ramsay, 1999).

Our reason for describing Parasite in detail is to allow the reader to understand how we have extended some parts and to reconstruct it in a way to employ it in our aRaBCALL system by relaxing constraints on unification. We will discuss how Parasite dealt with the full form of the surface written form. The full form is a form that contains the diacritic markers that indicate the presence of short vowels and other unwritten marks. The full forms provide all the information that is needed to produce a phonetic transcription (Ramsay and Mansour, 2004) which can be used for our CALL system aRaBCALL.

We use the notation of a ‘SIGN’ as a bundle of syntactic, semantic and other information relating to a single linguistic unit. In each lexical unit there are three aspects that SIGN constrains: i) a subcategorisation (SUBCAT) list that describes the set of arguments it requires to be saturated, which is when a word is completed by adding affixes to the root to fulfil the syntactic properties such as agreement markers; ii) the canonical order of arguments in SUBCAT list; iii) theta roles. Targets and Results describe the item and will modify what the resulting item will be like. These aspects are
exploited by two rules of combination: a lexical rule to cover the SUBCAT principle in HPSG and another rule for processing modifiers and targets.

Parasite that has been used adopts HPSG as the main underlying grammatical framework for Arabic grammar. This grammar has been adopted since “… [HPSG] relies crucially on complex lexical information, which determines, in accordance with general principles such as HFP [Head Feature Principle] and the Sub-categorization principle, the essential grammatical properties of phrasal expressions. This does not mean, however, that HPSG relies on complex lexical stipulations, or that the presence of distinct lexical entries with shared properties leads to massive redundancy within the lexicon”. (Pollard and Sag, 1994: 36)

Lexical Functional Grammar (LFG) and other linguistic theories are some of the unifications that are used in HPSG. Pollard and Sag (1994) explain that there is more emphasis and is a better use of structure sharing in HPSG than LFG on the phrase-structure aspects of the language. A single representation is integrated from syntactic and semantic information in HPSG. HPSG depends greatly on a system that specifies the kind of linguistic object that is being described. As well as this, it aims to radically reduce the number and detail of linguistic rules. As HPSG keeps detailed information about each different ‘SIGN’, it has a feature-based approach, which consequently reduces the grammatical rules. In addition, “… HPSG is heavily dictionary based” (Schulze, 2001: 45). A great deal of grammatical information with relevant lexical entries is stored in the dictionary. Words can be characterized as members of specific classes and subclasses. HPSG can be easily written when using word classes that are taken from traditional structural linguistics.

HPSG is relied upon by the Arabic system for the capturing of syntactic phenomena and it also uses categorial morphology for descriptions of word structures. Categorial morphology works with a limited number of categories and a number of rules that are able to form a word by explaining the combination of these categories. In the following section, we will present a survey of Arabic morphological analysers.
5.3 Survey of Arabic Morphological Analysers

A good notion of Arabic morphological analysers is the review of some approaches in regards to Arabic computation morphology; however, there is no accurate Arabic parser that describes all types of lexicon roots and patterns.

Arabic is one of the languages which have a non-concatenative morphology, in which morphemes are combined in more complex ways. Root and pattern and template morphology is considered as a kind of non-concatenative Arabic morphology (Beesley, 1998; Karttunen and Beesley, 1992).

Most computational treatments of Arabic morphology are based on linguistic models that describe Arabic in a non-concentenate way and focus primarily on analysis. All the systems are classified into three types according to the technique that researchers have built on. The three techniques are Two-level Morphology, multi-level and Finite State Transducers (FST) and unification-based frameworks.

5.3.1 Two-level Morphology

Two-level Morphology is characterized by an emphasis on phonological rules and has a very basic treatment of morphology itself. Koskenniemi (1983) has established achievable Two-level Morphology, which is particularly suited to concatenative morphology, though it has been extended to Arabic template morphology first by Beesley (1991; 1996), Kay (1987), and Kiraz (1994a; 199).

In general, Two-level Morphology implies a diverse framework for morphophonological variation. Two-level Morphology aims to optimize the implementation of traditional phonology by rewriting rules with parallel rules instead of the usual sequential cascade of morphophonological variation. The following are the characteristics of Two-level Morphology:
Chapter 5: Arabic Morphosyntax within the Parasite Framework

1. Only the morphophonemic and the surface forms exist, which means that there are no intermediate representations or results.

2. The order of the rules does not have an effect on the other rules.

3. The rules are logical constraints and define the relation between the morphophonemic and surface level representations rather than procedural actions mapping one string to another.

4. Two-level rules could refer to conditions also on the surface, or even refer to relations between the morphophonemic and the surface.

The Two-level rules could not deal with irregular forms, e.g. the relation between the surface form X and its morphophonemic form Y. Some complicated rule conflicts and other interactions were also more problematic to handle in Two-level frameworks than in the cascading scheme.

5.3.1.1 Beesley’s Approach

A study of Arabic morphology has been presented by Beesley (1990; 1991). He has described a system for a two level Arabic morphology. The system consists of two kinds of lexicon: root and pattern lexicon, which are both stored by inflectional patterns in the form ‘A_B’, as the underscores are called detours (ibid). The verbal and nominal patterns’ root can have different entry markers and all MSA lexicons contain about 5000 roots (Ahmed, 2005). The system presents Two-level Morphology of each root and pattern (Beesley, 1998). The formalism in rule 5.1 below shows the root and the pattern of the lexical description:

(5.1) [drs& CaCaC]

The bracket symbols define the output stem and the ampersand is for separating the root from the pattern. He proposed another approach in 2002 by using a finite-state lexical transducer in which each rule is compiled into a transducer to map the lexical entry [drs&CaCaC] into a surface form [darasa], consider the rule 5.2:

116
However, the disadvantage of this approach is that it needs a huge number of rules to be compiled, which is why he may have abandoned this approach (Beesley and Karttunen, 2003).

5.3.2 Finite-state Morphology

For a non-linear morphology, finite-state model of phonology and morphology have been proposed.

5.3.2.1 Kay’s Approach

Kay (1987) proposed a finite-state approach for handling Arabic non-concatenative morphology that followed the CV analysis of Arabic. This differs in the number of tiers that are used for mapping from the root to the surface form. A four-level account of how the Arabic root دَرَسَ d̄rs is mapped onto the stem d̄aras by means of templates ‘CVCVC’ is presented. The four levels of transducer architecture consist of: the first tape containing the root, the second containing the template, the third being interactive vowels and lastly, the surface form.

5.3.3 Multi Two-level FST Morphology

5.3.3.1 Kiraz’s Approach

A Two-level formalism which is proposed by Kiraz (1994a) is based on three linguistic frameworks: CV, moraic and affixational. Furthermore, it introduced a computational account applying especially for Arabic broken plural nouns. Two-level theory defines two levels of strings in recognition and synthesis: lexical strings represent morphemes, and surface strings represent surface form. The two strings are mapped by Two-level Morphological rules: the rules are compiled into FSTs. He introduces a multi-tape Two
level-Morphology and formalism where three tapes are used for the lexical level (root, pattern and vocalization) and one tape for the surface level as shown in figure 5.1.

<table>
<thead>
<tr>
<th>Lexical type</th>
<th>d</th>
<th>a</th>
<th>r</th>
<th>a</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consonant type</td>
<td>d</td>
<td>r</td>
<td>s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Template Morpheme</td>
<td>c</td>
<td>v</td>
<td>C</td>
<td>v</td>
<td>C</td>
</tr>
<tr>
<td>Vocalic Pattern</td>
<td>a</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5.1: Multi-level template Arabic morphology scheme**

The CV approach is computerized most frequently. The best approaches are moraic and affixation (Kiraz, 1994b). However, there are some difficulties in the CV analysis.

The CV approach describes the stem without referring to the inflectional morphemes. Two types of rules are needed: the first is the mapping rules from the surface form to the lexical form and the second being the eraser rules to delete the associated line for certain patterns. Therefore, the CV analysis is not a systematic model for the whole of verbal and nominal patterns. A large pattern lexicon results from indexed templates. The moraic analysis was applied to nominal stems. This is beneficial as the moraic analysis builds on the notion of extrametricality, allowing all the right edges of Arabic stems ending in a consonant. The moraic framework theory explains that the phonological word is made up of feet, which is composed of at least one stressed syllable. The moraic analysis builds on the notion of extrametricality. The right edge of all Arabic stems must end in a consonant, which was considered to be an
extrametrical syllable and was denoted by ($\sigma$) for the light syllable and two ($\sigma\sigma$) for the heavy syllable. (McCarthy and Prince, 1990)

The moraic analysis was applied to nominal stems. The pattern lexicon in this case is smaller than the CV. A Two-level rule has been applied to the moraic model. The affixational model was applied to the verb and its pattern lexicon contains only one template. Affix morphemes are placed in a separate lexicon and sit on a separate tier.

### 5.4 Treatment of Arabic Morphology in Parasite

A morphological analyser can be characterized in its simplest terms as a module that accepts words and outputs morphological analysis for a surface form. Parasite manipulates the word that we have got in a surface form by firstly finding the morphemes representing the base form and by secondly, combining the word which is perceived in a linguistic field as a syntactic description. (Beesley, 2001)

Generally speaking, a morphological analysis of any natural language is a crucial phase for all its practical applications. The typical morphological analyser tackles three issues: the morphological lexicon, rewrite rules and morphophonemic rules (Kiraz, 2001). The Parasite system includes three main components for morphological parsing.

- **Storing Morphemes**

  It is a lexical transducer that encodes all the lists of lexical entries, like roots and affixes, with basic information. This lexical transducer requires storing words in a lexical trie (see figure 5.2 below).

- **Boundary**

  In this stage, words combine by phonetic morphemes, by applying morphophonemic FSTs as the spelling rules, which are used to model the changes that occur for the word pronunciation. An example of this would be the handling of the phonological change and the orthography. In most languages, these kinds of change occur at boundaries of words.
• **How Words are Made out of Pieces**

In this component, a set of categorial affixation rules decomposes the surface form into affixes. This shows that a special rule is being applied to the rules causing the component to spot a combination of words. The rules needed here may be supplied from a variety of sources and with some constraints. Unification is used to carry the information by indicating the diacritics. Categorial morphology (Ramsay and Mansour, 2001) was used to analyse the input data. We will discuss these notations in detail in section 5.4.3.

### 5.4.1 Arabic Morphological Analyser

Nearly every practical application that uses natural language must contain a morphological analyser. As we claimed before there are three issues which the typical morphological analyser tackles: the morphological lexicon, rewrite rules and morphophonemic rules (Kiraz, 2001).

The existing morphological analyser has two main components for morphological parsing. The first of the two components is a lexical component transducer that encodes all the lists of lexical entries like roots and affixes with basic information. The second component is the spelling rules, which are morphophonemic FSTs that are used to represent the changes for word pronunciations by the handling of the phonological change and the orthography. Categorial affixation rules decompose the surface form into affixes and encode the vocalic patterns.

### 5.4.2 Root and Lexical Entry

This section is consisted of three parts: the lexical trie and slot-filler, Categorial Grammar and affix entry.
5.4.2.1 Lexical Trie and Slot-filler

Generally, the most vital task is to map from the written surface form to the underlying lexical entry because of the morphological and orthographic alternations. We did some initial experiments to decide what the lexical entry was. The first design for an entry was similar to Kay’s entry. Lexical entries are root forms, from which specific forms such as tensed forms for verbs are derived, as shown in example 5.4 below:

(5.4)
"d?r?s" lextype regular(i(1, "u"), a, 1)
:: verb, translation(study), sense(drs_study)
delayed vtype(valency(1, [agent:living, object: ~living]))).

The example declares that the lexical entry is the root درس $d?r?s$, which will accept the other morphemes for outlining the other written verbal forms. Now let us consider the lexical entry for a noun as the example 5.5 shows below:

(5.5)
"d?r?s" lextype regular(nominal,

"":[["a", "o"]:_:regular(_): lesson:

[not_plural, masculine, translation(lesson)],
["o", "w"]:_: broken(""): lesson:

[third_plural_only, translation(lesson)],
["A", "i"]:_:regular(_):human:[translation(student)],
["i", "A"]:_:regular(_): ~living:

[feminine, translation(study)].

This example shows that the root $drs$ has three interpretations as a noun, meaning lesson, student and study. When it is used to mean lesson, it has different diacritics in the singular/dual forms (درس $d?r?s$) and in the plural (دروس $drws$): this information is specified in the lexicon by noting that one set of diacritics corresponds to the ‘not plural’ form and the other to the plural. The other entries here are regular, and say nothing about
number (or indeed gender: \(dAr\)s (student) can occur with either masculine or feminine endings, as well as with all three numbers).

From the analysis above, we see that Arabic, unlike English, has very productive morphology, because the root gives many derivational forms that are made from attached morphemes and template patterns, so that storing all the possible stems for a given root requires a dictionary. For a root like 

\[
\text{درس} \quad \text{درس} \quad \text{درس}
\]

the diacritics vary so the best we can do is to put holes into the lexical item marking the places where the diacritics are needed. The general rule for lexical lookup is shown in the following formalism:

\[(5.6) \quad C \quad C \quad ? \quad C\]

The rule says a symbol ‘?’ would be inserted between any pair of consonants in a surface form. In the letter trie, lexical items are stored, so that words with the same initial characters lie down the same branch (Fredkin, 1960). The lexical entries for the roots 

\[
\text{درس} \quad \text{درس} \quad \text{درس}
\]

‘to study’ and 

\[
\text{درس} \quad \text{درس} \quad \text{درس}
\]

‘to train’ are therefore 

\[
\text{درس} \quad \text{درس} \quad \text{درس}
\]

and 

\[
\text{درس} \quad \text{درس} \quad \text{درس}
\]

as shown in figure 5.2 below.

![Figure 5.2: Root trie representation](image)

As noted in the above figure, the root 

\[
\text{درس} \quad \text{درس} \quad \text{درس}
\]

‘to study’ and 

\[
\text{درس} \quad \text{درس} \quad \text{درس}
\]

‘to train’ share the same branch starting by ‘d’. The root 

\[
\text{درس} \quad \text{درس} \quad \text{درس}
\]

‘to through’ is a quadrilateral root that consist of four consonants and three question mark symbols ‘?’.
5.4.2.2 Categorial Grammar - CG

Categorial Grammar is described as being in a logical form. The direct way of attaching semantics to syntactic operations is one of CG’s essential characteristics. Grammatically, components that have been logically stated and specified as to whether an expression is in its acceptable form, corresponds to presenting a derivation in the logic for these connectives. The words, in regards to functions, are then to be applied on the semantics of argument expressions (Steedman, 1985; Wood, 1993).

- Unsaturated CG Rules

This consists of unsaturated lexical items and spelling rules. Unsaturation is the stage of the word being incomplete in that it contains a root with no affixes, Hoeksma (1985) explains that category is defined as for example, A is category as long as A is a member of the set of primitive categories:

Or A is of the form V/W, where V and W are categories and V/W is used to denote an incomplete V which needs a following W,

Or A is of the form V\W, where V and W are categories and V\W is used to denote an incomplete V which requires a preceding W. (Hoeksma, 1985:13)

In order to conclude morphological processes, the association rules stated below, in time form a fully inflected lexical sign:

A \(\rightarrow\) A/B B

A \(\rightarrow\) B A\B

A sublexical sign can also be created using the composition rule stated below:

A/D \(\rightarrow\) A/B B/D \(\rightarrow\)

A\D \(\rightarrow\) B\D A\B \(\rightarrow\)

In structural analysis (at the word level, at the sentence level, etc.), the word occurs at a range of levels, however, the feature description remains the same. For the word درس
drs as shown in the description below, it may be a verb category which may have derived affixes. The derived affix is related to the tenses as shown in example 5.7, in which a third person that presents a singular verb will be joining a prefix يُ and the word will be يدرس ydrs.

(5.7) "y?" $$X:\text{-} \text{trigger}(	ext{active@X,}
\text{checkPresentPrefix}(V0, X, y, V1)),
\text{text@X} \text{-} \{V1\}, \text{lextype@X} \text{-} \text{regular}(T, V0, _),
X \text{-} [\text{verb, prefix, tensePrefix}],
\text{affix@TNS1} \text{-} *tns1, [\text{lextype, sort, syntax,}
\text{branch}@TNS1 \text{-} [\text{lextype, sort, syntax, branch}@X,
\text{affixes@TNS1} \text{-} [\text{AGR}], [\text{branch}@X \text{-} [\text{branch}@AGR},

A look up process has three stages in order to convert the surface form of MSA words to full form lexical items. This is achieved firstly, by using Finite State Automata (FSA) and a grapheme trie in order to compare the letter of the written form with branches of the root radicals and affixes in the lexicon (Ahmed, 2005). Secondly, morphophonemic rules are used to manipulate the phonological changes using a FST, and finally using these rules of CG for affixes to be able to constrain the content of diacritic placement (ibid).

The structure of a root entry representation is explained below. The root has specific verbal and nominal affixes that they combine with. Arabic roots that share all its syntactic properties are combined with a derivational affix. In the structure below, the way that Parasite can capture the diacritic patterns and the root entry in Parasite lexicon for verbal affixes are shown:

(5.9)

"d?r?s" le Dixte regular(ii, u, l)
::: (verb, translation(teach)) delayed

However, the structure below shows the root for the nominal affixes:
(5.10) "دروس" lextyped regular(nominal, ['\':[['ا', 'و']]:_:broken(''):
lesson:[masculine, translation(lesson)]],
['و', 'و']:_:_regular(_):
third_plural_only,
translation(lesson)],
['ا', 'و']:_:_regular(_):
translation(lesson)],
['ا', 'ا']:_:_regular(_):
translation(study)],
['أ', 'أ']:_:_regular(_):~living:
feminine,
translation(study)]},
t: [['و', 'و']:_:_regular(_):
~living:
[feminine,
third_plural_only, translation(teaching)]],
['و', 'و']:_:_regular(_):~living:
[feminine,
third_plural_only, translation(teaching)]],
{م,ع}:[[ا, 'ا']:[_C, _]:regular(_):human:
translation(teacher)],
[[ا, 'ا']:_:_regular(_):
~living:
masculine,
translation('old material')]],
{م,ع}:[[أ, 'ا']:_:_broken(''):
~living:
[feminine,
translation(school)]],
['و', 'و']:_:_broken(''):
~living:[third_plural_only,]]

The root entry specifies the verbal classes, the lexical type and the set of derivational morphemes that it accepts. The root stated above is for درس ḏṟs which contain two slots which can be instantiated by the vocalic pattern [___A, ___B] according to the affix that appears on the surface form.

- Saturated CG Rules

The general pattern for saturated CG rules is that a single root provides an increase to a larger number of different nouns and verbs. In addition, a number of specific derivational affixes each require a range of inflectional affixes such as tense markers, agreement markers and case markers.

The root always requires derivational affixes and in Arabic there is a significant number of derivational affixes. In the Arabic system, Parasite, there are 85 nominal affixes and 14 verbal derivation affixes. To indicate the derivation affix, it needs to
locate the conditional constraints that allow the combination with other affixes in order to choose the required diacritic patterns (Ahmed, 2005).

- **Verbal Marker**

To discuss the Arabic present tense verbal marker, the example 5.12 below shows the present tense marker indicating what is required for an additional agreement marker:

\[(5.12)\]

\[
\{\text{sform} = \{ \text{ydr} \}, \\
\text{uform} = \{ y + a + d?r? \}, \\
\text{diacritics} = (\text{present} = \text{“a”} + \text{“u”}, \\
\text{actual} = \text{“0”} + \text{“u”}), \\
\text{fform} = \{ y + a + dorus \}\}
\]

There are three types of forms that are represented in the notation 5.13 below, showing the encoding of the diacritics patterns. These forms are known as the surface form (sform), underlying form (uform) and the full form (fform). Also, the present tense marker \( \{ y \} \) that chooses the suitable diacritics patterns for its present verb is represented in the notation (ibid). In addition to this, the present tense marker indicates an additional agreement marker is required.

\[(5.13)\]

\[
\{\text{sform} = \{ \text{Al} \}, \text{uform} = \{ ? \text{a1} \}, \text{Cat} = \text{noun}, \\
\text{Diacritics} = \text{diacritics@noun}, \\
\text{Affixes} = [\text{def}, \text{dir=before}]\}
\]

- **Nominal marker**

Since Arabic nominal words are derived from the root with a derivational nominal affix, Parasite has set diacritic patterns that consist of vowel and consonant patterns in order to deal with this. There are 85 nominal patterns in Arabic, for example, adding the prefix مَ m to a root 

\[
\text{dr} \]

will generate two nominal words, one is either مُدَرِّسَة mudaris ‘teacher’ or مدرسة madorasp ‘school’.
The Arabic definite article ال Al ‘the’ is added to the nominal word as a prefix. The representation below (5.14) expresses article morpheme ال Al and identifies the fitting nominal diacritic pattern and not the verbal pattern. The entry for the definite article looks like:

(5.14)

"Al" X lexttype defarticle :
    text@X <- '؟al',
    X <- definitel,
    +specified@X,
    affixes@X <- [N],
    affixes@N <- [],
    -affix@N,
    specifier@X <- definite,
    [diacritics, consonants, def, syntax\specf, sort, meaning]@X
    <-> [diacritics, consonants, def, syntax\specf, sort, meaning]@N,
    dir@N <- xafter.

The bold lines represent the full form for the definite affix ‘؟al’. The noun that comes after the article does not need affixes.

In the next section, we will show how Parasite is able to represent the full form of written MSA for verbs and nouns by determining some examples.

5.4.2.3 Affix Entry

As morphemes are relatively fixed, the dictionary shows them as complete sub-lexical items. A sub-lexical entry is similarly looking to a sign, which includes morphological, phonological and syntactic features. Categorial rules help to implement and process them. There are two types of morpheme entries that the lexicon includes: Derivational
morphemes that encode the diacritics pattern set and the other being inflectional morphemes which are required. The structure of an affix entry used with Parasite demonstrates the agreement marker, which is a suffix. This lies on the left side of the stem and shares the stem category, without needing any more affixes, i.e., an empty affix or ‘.’.

5.5 Morphological Analysis: Roots and Affixes

In this section, we represent the implementation of the Arabic surface forms including nominal forms and verbal forms. In other words, this section deals with the link between the lexical trie, spelling rules and unsaturated lexical items.

5.5.1 Verbal Morphology

Arabic verbs are categorized into fifteen classes and three subclasses according to the interdigitated diacritic patterns (Ramsay and Mansour, 2004). The verbal inflectional standard of an Arabic root is determined by specific interrelated levels: the verb classes, the derivational affix and the tense’s consonant patterns. These levels are described by a rule that says the Arabic regular verbal type is defined according to the above parameters that appear in the root entry.

At this point, all the system knows is that the verbal stem requires an affix. That affix may be a tense marker or agreement marker affixes. Now the following example explains the morphological analysis when the system deals with a verb pattern (ii) tense marker. The rule below shows how Parasite sets the diacritics marks.
Chapter 5: Arabic Morphosyntax within the Parasite Framework

(5.15) \textbf{setDiacritics}(X, \text{ii}) :- \\
, \\
\text{passivepastdiacritics}@X \leftarrow \text{["u", "i"]}, \\
\text{passivepresentdiacritics}@X \leftarrow \text{["a", "a"]}, \\
\text{pastdiacritics}@X \leftarrow \text{["a", "a"]}, \\
\text{presentdiacritics}@X \leftarrow \text{["a", "i"]}. \\

The above rule (5.15) sets the diacritics pattern for a verb of class (ii) in past and present tenses and in active and passive voices. The diacritics will be chosen if the tense of the verb has been indicated as past or present.

(5.16)

\[
\text{\$\$ X :- lextype}@X \leftarrow \text{regular}(\text{LTYPE}, _, _), \\
\text{affix}@X \leftarrow \ast\text{deriv}(1), \ X \leftarrow [\text{verb, suffix}], \\
\text{affixes}@X \leftarrow \text{[FIRST]}, \text{affix@FIRST} \leftarrow \ast\text{tense}, \\
\text{text}@X \leftarrow \text{"0"}, \text{trigger}(\text{LTYPE},(\text{LTYPE} = \text{ii} \rightarrow \text{consonants}@X \leftarrow \text{[[_], [C, C], [\_]; true)}), \\
\text{[history, syntax, lextype]@FIRST} \leftarrow \text{[history, syntax, lextype]}@X.
\]

Now from the notation (5.16), we can see the empty derivational affix entry ‘ ‘ as it applies to class (ii) patterns. This affix is the first suffix that has one role which is to encode the consonantal and the vocalic patterns. After they are set, a pattern choice will be assigned by the tense affix as shown in the notation 5.17 below.

(5.17) \[
\text{\$\$ X :- affix}@X \leftarrow \ast\text{tense}, \\
x \leftarrow [\text{verb, suffix, past-tense}], \\
\text{active}@X \leftarrow \text{ACTIVE}, \\
\text{trigger} (\text{ACTIVE}, \\
\text{ACTIVE} = + \rightarrow \\
\text{pastdiacritics}@X \leftarrow \text{actual}@X; \\
\text{passivepastdiacritics}@X \leftarrow \text{actual}@X),...
\]
The codes in the above notation show a past tense marker entry. It describes the tense for both active and passive voice. The system applies more specification to choose one pattern of them. Triggers are used for delaying execution until some condition is satisfied. In (5.16), for instance, we may not know whether $X$ is active or not at the time when the constraint is set. The trigger says that when we know the value of $\text{ACTIVE}$ we should check the if-then-else condition that forms the second argument of the trigger:

We can apply this specification to the verb. If the verb is active then the diacritic is represented by the code $\text{pastdiacritics@X} \leftarrow \text{actual @X}$. If not then it represent counterpart passive pattern. The system output showing the full form of a verb of class (ii) for $\text{درس} \ dars$ ‘to study’ is shown below in 5.18:

\[(5.18)\]

```
?- in arabic.
   Input a sentence in Arabic
   | : drs ^
   | ?- surfaceForms.
   Word 1 : full form is darasa
   Word 2 : full form is durisa
   Word 3 : full form is darrasa
   Word 5 : full form is durrisa
   Word 2 : full form is drws?
```

The term $\text{surfaceForms}$ represents the full formula of the input. Parasite also treats present tense markers, affixes or future tense markers. In Parasite, there are 5 types of lexical entries. The following code (5.19) is the lexical entry $\mathcal{y}$ for the present tense prefix. The full form for each present prefix can contain $\check{u}$ or $\check{a}$ depending on the verb class.
(5.19)

"y?" $$ X :-
  trigger(active@X, checkPresentPrefix(V0, X, y, V1)),
  text@X <-> {V1},
  lextypo@X <-> regular(T, V0, _),
  X <-> [verb, prefix, checkPrefix, tensePrefix],
  affix@X <-> *tense,
  affixes@X <-> [TNS1],
  affix@TNS1 <-> *tns1,
  [lextypo, sort, syntax, branch]@TNS1 <-> [lextypo, sort, syntax, branch]@X,
  affixes@TNS1 <-> [AGR],
  [branch]@X <-> [branch]@AGR,
  constraintsOnYA(text@AGR, X).

In the next section, we will describe how Parasite deals with noun morphology.

5.5.2 Nominal Morphology

As mentioned before in chapter 3, in addition to the verbal sentences, Arabic allows nominal sentences, which contain an NP as a theme and a predication that could be NP, PP or VP. The case mark of the theme is always nominative and the predication should be accusative or marked by a preposition (Badawi et al., 2004; Attia, 2008). The Arabic noun category has sub-categorizations that cover the details of this class. Arabic nouns are split morphologically into two types: solid nouns and regular derived nouns. They come in a variety of numbers and genders. They are related to many template patterns as shown in the following section. We encode the most frequent nominal patterns in MSA. There are a large number of nominal derivational affixes that produce these patterns (Ryding, 2005). The Parasite system has 85 nominal patterns that work in combination with a number of nominal agreement suffixes. The diacritic placement for nominal patterns for the Arabic word root, to recreate the full form, is implemented by the
following factors: broken or regular nouns, the nominal diacritic patterns, and the case markers (Attia, 2008).

The system encodes the broken and regular derived nouns by reconstructing its surface forms. The full form is assigned by two ways depending if the noun is broken, such as the word رجل rjl ‘man’, or derived, otherwise the derivational nominal affixes select the diacritic set. The nominal morpheme can be either an empty affix or an overt affix. The example in the next section explains the details of how the system deals with the solid noun.

### 5.5.2.1 Case Study on Broken Noun

The Arabic solid noun like رجل rjl ‘man’ has an individual root that does not give rise to any derived form. Also, each noun has its own diacritic set, which is not predictable, as shown in the rule in table 5.1:

<table>
<thead>
<tr>
<th>Surface Form</th>
<th>Underlying Entry</th>
<th>Diacritic Pattern</th>
<th>Full Form</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>رجل rjl</td>
<td>r?j?l</td>
<td>[a,_u]</td>
<td>Rajul</td>
<td>Man</td>
</tr>
<tr>
<td>بنك bnk</td>
<td>b?n?k</td>
<td>[a,o]</td>
<td>Banok</td>
<td>Bank</td>
</tr>
<tr>
<td>ولد wld</td>
<td>w?l?d</td>
<td>[a,a]</td>
<td>Walad</td>
<td>Boy</td>
</tr>
<tr>
<td>طفل Tfl</td>
<td>T?f?l</td>
<td>[i,i]</td>
<td>Tifil</td>
<td>Child</td>
</tr>
</tbody>
</table>

**Table 5.1: Solid nouns diacritic patterns**

The system allows the lexicon to include the individual entries for these types of nouns. The following example (5.20) shows that the diacritic set will be indicated individually by the solid noun entry. It is an entry for a singular broken noun that has a different diacritic set.
Chapter 5: Arabic Morphosyntax within the Parasite Framework

The following example 5.21 shows the full form Parasite system procedure:

(5.21) 
?- in arabic. 
  Input a sentence in Arabic 
  : wld ^ 
  ?- surfaceForms. 
  Word 1 : full form is walad+0 (regular)

Or

?- in arabic. 
  Input a sentence in Arabic 
  | : bnk~ 
  ?- surfaceForms. 
  Word 1 : full form is banok+0(regular)

5.6 Syntax

5.6.1 Introduction

The reason for the detailed description in the preceding sections of the lexical and morphological analysers for Parasite was to form a basis for the discussion of how we extend them for use in the aRaBCALL system.

Similarly, the aim of the current section is to explain how Parasite deals with the syntax of MSA, in order to be able to discuss the changes that were needed for it to be used in aRaBCALL.

Syntax is the study of formal relationships between words (Jurafsky and Martin, 2009). The major characteristics of the Arabic language are that it is a clitic language
which makes Arabic morphological analysis more complex because of building words from adding affixes to the root (Crystal, 2003). Arabic is of free-word order sentence structure, thus, it can be the order VSO, SVO, VOS, or OVS. Besides being a free word order language, Arabic also has the features of pro-drop, invisible passive marking, optional agreements and construct NPS. As we are going to use a variant of HPSG, the next section is an introduction to HPSG.

### 5.6.2 Head-Driven Phrase Structure Grammar- HPSG

Some of the linguistic theories which were adapted to NLP are based on LFG (Kaplan and Bresnan, 1982), Phrase Structure Grammar (PSG), Generalized Phrase Structure Grammar (GPSG) (Gazdar, 1985) and Head-driven Phrase Structure Grammar (HPSG) in which subcategorisation frames are used (Pollard and Sag, 1994).

#### 5.6.2.1 Background Study on HPSG

In this section, key concepts and the history of HPSG will be focused on. HPSG was first introduced by Pollard and Sag (1987). This approach was formalized in 1994 in the standard HPSG reference book: Head-driven Phrase Structure Grammar by Pollard and Sag (1994). If the sentence satisfies all the constraints within the set of constraints that are defined in the grammar, a sentence is grammatical and has an HPSG analysis.

Bennett (1997) discusses a number of distinctive aspects of HPSG such as:

- HPSG makes great use of **structure sharing**.
- HPSG places more emphasis than LFG on the **phrase-structure** aspects of the language.
- HPSG aims at drastically reducing the number and detail of linguistic rules.
- HPSG relies heavily on a **type system** which specifies the kind of linguistic object being described.
- HPSG integrates syntactic and semantic information in a single representation.
HPSG grammars contain two elements, the sort hierarchy or the signature, and the set of constraints. The sort hierarchy is a partial order that defines the ontology of linguistic objects. The sorts are used to label the feature structures.

5.6.3 Subcategorisation in HPSG

Subcategorisation is one of the basic concepts used with HPSG. SUBCAT is the term used by HPSG, to store the information about the subcategorisation of a specific item. Pollard and Sag (1987: 67) describe subcategorisation or valence of a lexical sign as “a specification of the number and kind of other signs that the sign in question characteristically combines with in order to become complete”.

HPSG is more concerned with the role of phrase-structure configuration than most feature-based approaches. The common term in HPSG for feature representation is the SIGN, as discussed in section 5.2. Pollard and Sag (1987) use the term argument referring to the signs subcategorized for by the verb. They explain subcategorisation as “In a headed phrase (i.e. phrasal sign whose DTRS value is of sort HEADED-STRUCTURE), the SUBCAT value of the head daughter is the concatenation of the phrase’s SUBCAT list with the list (in order of increasing obliqueness) of SYNSEM values of complement daughters” (ibid:34).

HPSG has used the following formalism for describing the subcategorisation principle:

\[
\begin{align*}
\text{DTRS} & \quad \text{HEADED-STRUCTURE} \\
\rightarrow & \quad \text{SYNSEM} \\
& \quad \text{LOC} \mid \text{CAT} \mid \text{SUBCAT} < \\
& \quad \text{DTRS} \\
& \quad \text{HEAD-DTRS} \mid \text{SYNSEM} \\
& \quad \text{LOC} \mid \text{CAT} \mid \text{SUBCAT} < \\
& \quad \text{COMP-DTRS} \\
& \quad \text{1} \\
& \quad \text{1} + \text{2}
\end{align*}
\]
Chapter 5: Arabic Morphosyntax within the Parasite Framework

From the above diagram we notice this HPSG represents the structure of a headed phrase such as a sentence with its property of having DAUGHTERS (DTRS). These DTRS are divided into two types.

- Firstly, the head and the daughter that shares most of the mother’s properties.
- Secondly, the complement, there may be more than one complement.

There are two types of DTRS: HEAD-DTR and COMP-DTRS. In addition, the idea of paths is frequently mentioned in HPSG. The path is a vertical line between the attribute names as shown in the previous example (5.22). To declare the HPSG sign for a simple Arabic sentence, the following figure shows a simple tree of a simple verbal Arabic sentence:

(5.23)

\[
S : \text{كتبت الولد الدرس} \quad \text{‘The student wrote the lesson’}
\]

\[
V : \text{k*tb}
\]

\[
Np : \text{Alwld}
\]

\[
Np : \text{Aldr}\text{s}
\]

In the previous example, 5.23, the head is the V and the NPs are the complements. This information could be represented in HPSG sign as illustrated in the following example:
As seen in the previous example 5.24, it represents information about the attributes and values and more information than the previous example 5.23 indicating the differences between the values of the two of daughters. However, 5.24 has no information about the order of the DTRs.

The following example 5.25 illustrates the representation of the order of DTRs in HPSG which could be represented by giving a number to each DTR (Bennett, 1997).

In the previous examples, a clear representation of how HPSG sign aspect appears is given. Consequently, a brief explanation of a SIGN will be presented next. Any SIGN has three main aspects:
1. PHON feature which represents the phonological aspect.

2. DTRS feature, which carries information about the daughters.

3. SYNSEM feature which includes syntactic and semantic information, which is expressed via four features: firstly, HEAD and CAT for category of syntactic aspects, secondly, CONTENT for the meaning of the core parts of semantics and finally, CONTEXT, which includes matters such as tense and discourse.

The formalization 5.26 shows feature representations in the SIGN for Arabic transitive verbs:

(5.26)

The previous SIGN 5.26 shows the representation of transitive verb. The combination of the pronunciation of the daughters is represented in the PHON. The HEAD and COMP-DTR are represented by the DTRS. The general lexical sign, which has an
internal phrase structure, is called a phrasal sign. The following example describes one of the paths from (5.26):

\[(5.27)\]

\[
\text{DTRS} \mid \text{HEAD-DTR} \mid \text{CAT} = V
\]

The final point that needs to be made is that there are some changes of subcategorisation frame in HPSG versions (Pollard and Sag, 1987). This last version is called HPSG3 and uses three features for valency: SUBJ, COMP and SPR (subject, complement and specifiers, respectively) (Pollard and Sag, 1996; 1987).

### 5.6.4 HPSG Schemata

Before showing our Parasite framework, it is vital to represent the HPSG schemata, in which this importance can be verified by the fact that Parasite links with the schemata. We will represent the counterpart for each schema.

HPSG uses Linear Precedence (LP) rules and Immediate Dominance (ID) schemata instead of classical Phrase Structure (PS) rules. The relative order of the daughters in a local trie without making reference to the mother node is determined by LP rules. The Immediate Dominance Principle (IDP) is related to X-bar theory. Without specifying the order of the daughters, ID rules identify the mother and daughters in a local trie. To further constrain the set of local tries admitted by the ID schemata and LP rules, HPSG proposes several universal principles (Kordoni and Neu, 2005).

The fundamental unit in HPSG is the sign, which is a compound of phonological, morphological, syntactic, semantic, and discourse attributes. In terms of signs, the principles and grammar rules of HPSG are stated. Relationships between PHON, SYN and SEM information of a phrasal sign are described and contained in its daughter signs. IDP rules are grammar rules that are called schemata in HPSG. Without imposing any restriction on relative ordering, they constrain the structure of constituents (Pollard and Sag, 1997).
There are five universally available schemata for HPSG. These are disjunctive constraints on the immediate constituency of phrases, in which each language makes a selection. Of these five schemata, schemata 1 and 2 are complementary to schemata 3: some languages, such as English, make use of 1 and 2 but not 3, others, such as Arabic, make use of 3 but not 1 and 2. These Schemata constrain well-formed phrases. These schemata are represented in (ibid) as follows:

\[(5.28)\]

“(SCHEMA 1) a saturated ([SUBCAT<>]) phrase with DTRS value of sort head-comp-struc in which the HEAD-DTR value is a phrasal sign and the COMP-DTRS value is a list of length one” (ibid: 38).

\[S \rightarrow \text{NP VP}\]

\[(5.29)\]

“(SCHEMA 2) an almost-saturated (SUBCAT list of length one) phrase with DTRS value of sort head-comp-struc in which the HEAD-DTR value is a lexical sign” (ibid: 38).

\[\text{VP} \rightarrow \text{V O}\]

Schema 1 as shown in (5.28) has saturated phrases that contain a phrasal head daughter and another complement daughter. The SIGN structure must have the same SYNSEM value as the head daughter. The following PS (5.30) is licensed by schema 1 for any phrase:
Schema 2 contains phrases that have a lexical head daughter and zero or more complement daughters. It subsumes all the phrase structure rules that enlarge a phrase as a lexical head joined with its subcategorisation frame except the least oblique one, which is the subject. The phrase by Schema 2 is represented by the following structure (5.31):

\[(5.30)\]

\[
\begin{array}{c|c}
\text{LOC|CAT} & \text{HEAD} \\
\hline
\text{SUBCAT} & <2>
\end{array}
\]

In the first part of our framework, to code the arguments list and subcat list of each lexical item in our lexicon, we use Schema 1 and 2. Schema (3) is represented in 5.32:

\[(5.32)\]

\[
\begin{array}{c|c}
\text{LOC|CAT} & \text{HEAD} \\
\hline
\text{SUBCAT} & <2, 3, \ldots, n>
\end{array}
\]

“(SCHEMA 3) a saturated ([SUBCAT<>]) phrase with DTRS value sort head-comp-struc in which the HEAD-DTR value is a lexical sign” (ibid: 40).

\[S \rightarrow V \text{ SUBJ OBJ}\]

As can be seen, this schema is the last ID schema used for describing the lexical sign. Schema 3 states that all the complements, including the subject, are realized as sisters of
the lexical head, which controls the order of head phrase daughters. It is required by the specification of the ID schemata in standard HPSG handling complements, that the complements, with the exception of the subject, can be combined at once into a phrase. A similar technique is used in this schema to process the canonical word order.

(5.33)

“(SCHEMA 4) a phrase with DTRS value of sort head-marker-structure whose marker daughter is a marker whose SPEC value is structure-shared with the SYNSEM value of the head daughter, and whose MARKING value is structure-shared with that of the mother.” (ibid: 46).

COMPLEMENTISER + S

(5.34)

“(SCHEMA 5) phrase with DTRS value of sort head-adjunct-structure (head-adj-str), such that the MOD value of the adjunct daughter is token-identical to the SYNSEM value of the head daughter” (ibid: 56).

In Schema 5, a head is combined with an adjunct into one structure creating a head-adjunct-structure, to ensure that the constituent head is an element allowed by the mod feature of the adjunct. In a head-adjunct-structure via the Semantics Principle, the content of the mother is required to be token-identical with the content of the adjunct. The adjunct is allowed to appear immediately by the adjunct attachment schema, before or after the head it selects, or before or after the phrase containing the head and all of its complements. However, it will not allow the appearance of the adjunct within a group of complements (Kim, 2000).

Schemata 1 and 2 describe a situation where all the arguments but the least oblique (i.e., the subject) are found following the verb, and then the subject is found preceding it. In other words, they describe languages like English that have a distinct notion of a VP,
containing everything except the subject, which combines with the subject to make a full sentence. This is not appropriate for Arabic, where all the arguments follow the verb, with the subject occurring first in the sequence. This pattern is covered by Schema 3. Schema 4 is also appropriate for Arabic as it uses the complementiser. Schema 5 deals with adjectives making it appropriate for Arabic.

5.7 Parasite Framework

Parasite system is built on a mixture of CG and HPSG. It is about an adaptation of the general HPSG framework combined with CG rules in which each lexical item has a sign containing the following features: a SUBCAT list that describes the set of arguments it requires to be saturated, the canonical order of arguments in the SUBCAT list and theta roles, and the TARGET and RESULT, which describe the items it can modify and what the resulting item is like, respectively. These features are exploited by two rules of combination:

1. A lexical rule to cover the SUBCAT principle in HPSG

2. A rule of modifiers and targets (the following sections will explain these points)

Filter and Dynamic constraints are the two kinds of constraint that control the nature of Arabic word order. The filter is a part of the treatment responsible for allowing the arguments of the head to move from the canonical positions through a set of filters. The dynamic constraints are the case of marked orders governed by a set of constraints.

One of the important components in the Parasite system framework is the internal and external structure rule. In the following sections, a detailed representation of the other Parasite components will be discussed, such as argument and SUBCAT list, Arabic canonical order and the filter.
5.7.1 Constraints on Arguments

Arabic verbs have a range of possible arguments that are represented in a verb valency which is related to the number of arguments controlled by the verbal predicate. For example, in the case of a verb which can be transitive or intransitive, the lexical entry has to list it in both options. The subcategorisation frame determines the number of verb valency or arguments. A list of feature structures has been established in the domain of the valency. The lexical rule for the SUBCAT principle in HPSG is shown in the notation (5.35) below:

\[ \text{(5.35)} \]

![Feature structure notation](image)

The rule (5.35) above represents the standard rules from categorial grammar \( x \rightarrow x/y, y \). The rule shows the number of lexical items that can be combined with another item which is shown after the head of the SUBCAT list. The sign formalism was used to indicate the lexical verb type. The description of the verb came from the combination of specific features such as Argument lists and valency. An argument list uses non-terminal variables for the subcat lists for the verb. The features that are associated with verb arguments are transitivity, theta roles and selection restrictions. With regard to valency, it is the determined number of possible arguments which can be smaller than the listed arguments. This makes it possible to distinguish between obligatory and optional arguments. The number specifies how many of the arguments in the list are obligatory.
The following lexical entry from Parasite shows the encoding of the subcategorisation list in the system:

\[(5.36)\]

\%

"$rb=drink"

"$?r?b" $$ X lextype LEXTYPE

\[
\begin{align*}
\text{delayed} & \quad \left( (\text{LEXTYPE} = \text{regular}(i(1, "u"), a, 1) \\
\text{::: verb delayed vtype(valency}(1, \text{[agent:_, } \text{object:_]})\right) \right);…
\end{align*}
\]

It can be noticed from the above formula that the valency of class one (i) شرب \( \$rb \) ‘to drink’ includes two arguments as a transitive verb but it can have one argument as an intransitive verb.

### 5.7.2 Canonical Order and Theta Roles

The canonical word order of an Arabic sentence and especially MSA is VSO. The treatment by Parasite to a canonical word order Arabic sentence is based on two stages: argument order and argument combination. The technique used to represent the lextype entry is by recording the argument list for each lexical entry in the lexicon with the possible case marking and theta roles without declaring the word order. The following notation shows the word order for a transitive Arabic verb ركب \( rkb \) ‘to ride’

\[(5.37)\]

\[
\begin{pmatrix}
\text{NP1} & \text{NP2} \\
\text{agent} & \text{obj}
\end{pmatrix}
\]

After the theta roles order the head, according to the Arabic canonical word order the head is the verb. Therefore, combining arguments in canonical word order is secondly based on the assumption that it is supposed to be in canonical order. A traditional categorial grammar algorithm was used for combining arguments based on the unmarked order as shown in example 5.38:
5.7.3 Non–Canonical Word Order

The relatively free-word order in Arabic causes many structural ambiguities. A parser does not find it easy to detect which order is meant in a given sentence, since all these different word orders are possible in a given sentence. This is because the distinction between nominative subject and accusative object is made through diacritics which are missing in MSA. Thus, whereas SVO order is easily detected by the parser, VOS gets mixed up with VSO. This means that every VSO sentence has a VOS reading, which causes a serious ambiguity problem (Attia, 2008).

The following two part examples show the SVO order that needs to be filtered in the Parasite system.

(5.39a)

Consider the sentence العربية تدرس الدرس Albn tdr Aldr ‘the girl studies the lesson’ as S V O.
Now, consider the following example below as OSV:

(5.39b)

الدرس البنت تدرس Aldrs Albnr tdrs

The previous examples show the variation in word order for an Arabic sentence. Therefore, both examples give one meaning which is that ‘the girl is studying the lesson’.
5.7.3.1 Constraints on Out-of-Position Items

A filter technique was used to constrain (out of position) items. When combining two items to make a third, the filter technique needs to be checked to ensure that we are satisfied with what is being done. For example if the subject is in front of the verb it must be definite.

A filter module was applied to the Parasite system to treat the marked word order in Arabic. “The filter is the module that is responsible for allowing arguments to shift to non-canonical positions and banning unlikely argument movements according to a set of dynamic constraints” (Ahmed, 2005: 137). For example if the subject comes in front of the verb the filter techniques are important to allow only the definite subject to be in front of the verb.

The filter module is the most supportive part in Parasite (ibid). The roles of the filter are:

1. To check constraints on local subtrees, which manage the arguments’ movements of the embedded phrase
2. To check constraints on WH-marked items, which they are used to distinguish between declarative and interrogative mood. The rule is to allow WH-clauses as follows:

\[
(5.40) \quad \text{filter\_mood}(Z):-
\]
\[
Z \not\in \text{clause},
\]
\[
\text{language}@Z \not\in \text{arabic},
\]
\[
!,
\]
\[
( Z \not\in \text{no\_wh} \rightarrow
\]
\[
\text{default} ( Z \not\in \text{declarativel});
\]
\[
Z \not\in \text{wh\_interrogativel}).
\]

The previous rule shows the interrogative mood for Arabic.
5.7.4 Internal / External Syntax

An item which looks as though it belongs to one category can often be freely used in situations that require another. An English present participle VP, for instance, can often be used in contexts that require an NP (5.41). This is often referred to as the difference between its internal and external syntax. This is dealt with in Parasite by using an extension of Nerbonne (2003) and Malouf’s (1999) notion of lexical rules.

(5.41a)

He concluded the banquet by eating the owl.

(5.41b)

Eating raw meat will make you ill.

Each of these examples has a slightly different use of the verbal noun eating. The verbal gerund used in 5.41a retains some of its verbal nature. However, the nominal gerund eating used in 5.41b is a fully nominal word as other English nouns. See the analysis of both previous examples.

(5.41c)

Eating the owl

\[
\begin{align*}
\text{Cat} &= \text{prep} \\
\text{Args} &= \text{[NP]} \\
\text{Mod} &= \begin{cases} 
\text{Target} &= \text{VP} \\
\text{Result} &= \text{VP} 
\end{cases}
\end{align*}
\]

Eating raw meat \(\rightarrow\) VP

It will make him ill \(\rightarrow\) NP

The following notation from the example 5.41c for the VP make is as follows:
(5.41d)

\[ \text{make} \ [ \text{args} = \{ \text{NP}, \text{Adj}, \text{NP} \} ] \]

Therefore, the notation below shows that items that look like present participle VPs can be used as NPs. In other words, the notation 5.42 below shows the principle of external internal syntax:

(5.42)

\[
\begin{array}{c}
\text{NP (external)} \\
\to \text{VP as a present participle (internal)}
\end{array}
\]

In conclusion, the combination of verbal and nominal properties has made the framework of the syntax more challengeable. Generally speaking, the external syntax of the verbal gerunds is mostly the NP. The internal structure is the VP. Therefore, verbal gerunds take accusative NP complements, while the common nouns and nominal gerunds can only take PP complements.

5.7.4.1 Constraints on Verbless Sentences

Sentences in Arabic follow two classifications as mentioned in chapter 3. There are two types which are nominal sentences and verbal sentences. Arabic allows for sentences called nominal sentences. A ‘nominal sentence’ consists of an NP as a subject and a predicate (e.g. NP, PP and VP) (Fehri, 1993). Arabic nominal sentences, which look like English (small clauses), can be easily described by post-lexical rules by saying that something which looks internally like an NP can be seen externally as a sentence missing the predication (Ahmed, 2005). Consider the following notation 5.43 for Arabic verbless sentences:
Chapter 5: Arabic Morphosyntax within the Parasite Framework

(5.43)

\[
\begin{align*}
\text{Cat} &= V \\
\text{Args} &= [\text{NP}] > \text{NP}
\end{align*}
\]

To illustrate the constraint we need to consider the following sentences:

(5.44) \text{Aljaw Jameel} \quad \text{‘the weather is nice’}

Parasite has provided the following rule to treat the previous constraint:

(5.45) \text{ syn(nonfood(head (cat (xbar (+v, -n))),}
      \text{ subcat(args ([{struct (dir (+after ))},}
        \text{ meaning (+predicative)}]}))}

\text{ syn(nonfood(head (cat (xbar (-v, +n))),}
      \text{ minor(specf (kspec (+specified )))),}
      \text{ subcat(args ([]))})}

The rule 5.45 above represents that if there is an NP (+specified) it can be seen as unsaturated subject (S) which needs a (+predicative) argument after and which has the
NP as its subject. Moreover, the rule shows that the internal view of a nominal sentence is a specified NP and the external view looks like an unsaturated verbal sentence.

5.7.4.2 Relative Clauses

In MSA, the relative clauses modify the head noun. The head noun is definite, if for example 
نadv ‘that’ and التي alaty ‘which’, the relative clauses are introduced and headed with a relative pronoun (Habash, 2007; 2010). In other words, these pronouns link with the relative clauses. The relative pronoun agrees with the noun it modifies in gender and the number of the noun when it is present in the sentence. See example 5.46a below:

(5.46a)

الرجل الذي أحب Alrjl Alvy AHbh ‘the man who I like’

Therefore, the headed noun in the previous sentence is definite. But, if the headed noun is indefinite, then the relative clause will not be introduced with the relative pronoun.

(5.46b)

رجل أحبهُ rjl AHbh ‘a man who I like’

In a computational point of view and as mentioned before, the internal and external strategy means that in HPSG, words are treated as items that select for arguments of a specific category. Therefore internal and external views of Parasite relate to WH-clauses, such as interrogative and relative clauses (Ahmed, 2005).

Generally speaking, the Arabic language allows interrogative and relative clauses, which are introduced by a relative pronoun or an interrogative pronoun or particle, respectively. The external view of WH-clause will be inherited from the interrogative or relative pronouns that govern the clause and the internal view is a WH-marked clause. Consider the following rule:
(5.47)

\[
X \rightarrow \text{syn(head(cat(xbar(+v,-n))), subcat(args([ ])), foot(wh([X])))}
\]

Rule (5.47) says that, whatever the internal view for the relative-clause is, it will externally look like a clause governed by a relative-pronoun which will identify the relative-clause type.

5.8 Summary

Parasite–based HPSG relies on the interaction of two different sets of information: firstly, lexical information which is stored in the dictionary such as signs, and secondly, a set of principles and rules covering the combinations of lexical signs. Parasite will be capable to store morphemes in the dictionary and extend the set of governing rules and principles in such a way that they cover not only the possible combinations of lexical signs within sentences, but also the combination of smaller linguistic units to create lexical signs. In the following chapter, we will explain the architecture and formalization of our aRaBCALL system which is integrated from a NLP based Parasite as a learning aid for learners of Arabic.
Chapter 6

Implementation of aRaBCALL Prototype

6.1 Introduction

So far we have been dealing with the linguistic description of Arabic from a computational perspective, and looking at the different classifications of errors that are produced by learners in general and learners of Arabic in particular. From now on, we will move to the development and implementation of the tool that we have mentioned in previous chapters. This chapter documents and discusses aspects of the implementation of the aRaBCALL tool. We will provide a detailed discussion of the implementation of our tool by looking at the functionality of the system and the procedure of developing it.

The tool is concerned with the specific language problems identified when teaching Arabic as a second language. The system’s intended use is as a supplement to the face-to-face learning experience. One of the problems that students face is the lack of support to consolidate their knowledge immediately after a new topic has been introduced. The aim of aRaBCALL is to provide immediate feedback at various levels of homework that has been set to support the topic currently being taught. This helps the student, since they can work on the material while it is still fresh in their mind; and it also eases the burden on the teacher by relieving her of the need to mark large amounts of homework.

A soft parsing technique is used to allow one or more elements of ill-formed input to combine with other components and marks the resultant structure with the appropriate label. In other words, the constraint that ensures that the two values of the feature in question match is relaxed. In this way, the differing values can still be combined and pass the correct label upward. Once the system has joined the non-matching items, it
completes its parse of the whole sentence. If more than one candidate item has been
found, the parser will traverse each tree systematically and backtrack where necessary.
At the same time, the system will record what has actually been processed, the fact that
an error has occurred, what kind of error it was and so on. This process will now be
examined in detail.

We will consider soft parsing in relation to adjectives, verb tenses, use of personal
pronouns, use of weak verbs, and negation in Arabic. Subsequently, we will discuss how
to provide students with feedback on the basis of the results of soft parsing. The
following sections comprise this chapter:

- Explanation of the functionality of the system
- A brief description of the types of grammar aspects that aRaBCALL has
  implemented
- Demonstration of the system architecture
- Explanation of how the system handles students’ errors
- Detailed discussion of soft parsing techniques in relation to the grammatical
  system exercises

6.2 Pedagogical Principles
aRaBCALL has been introduced for learning Arabic as a second language. To describe
the prototype of the system, it is necessary to discuss the pedagogy of the learning
strategy and the underlying theoretical aspects of teaching and learning grammar and
learning through homework activity in the context of a second language.

6.2.1 Learning and Teaching Grammar
In chapter 2, we discussed the role of the teacher in language learning and teaching, and
we discussed the role of ICALL in teaching Arabic grammar. In this section, we will
argue that, in general, teaching grammar is an important field for teachers to focus on,
when teaching or learning a foreign language.
There is a complex relationship between teaching and learning and how something is taught is not directly related to how it is learned. There are some who in fact deny the important of explicit instruction in language acquisition. Krashen (1993:725) for example, describes the effects of grammar instruction as “peripheral and fragile”, arguing that explicit grammatical knowledge about structures and rules for their use may never become embedded knowledge underlying unconscious language production. Also rejecting the value of unambiguous grammar instruction is Truscott (1996), who on similar grounds argues that the effects are superficial and short-lived (ibid). Truscott proposes that if studies have shown form-focused instruction having many benefits, this means that such results have not been collated from the learner’s ability to use the target language in spontaneous communication, but more from tests that measure only explicit metalinguistic knowledge (Nassaji, 2010).

Ellis (2002) do not deny a role for explicit instruction, but do suggest that language learning is implicit in nature, “the slow acquisition of form-function mappings and the regularities therein. This skill, like others, takes tens of thousands of hours of practice, practice that cannot be substituted for by provision of a few declarative rules” (ibid: 175).

However, this does not necessarily mean that grammar instruction is completely useless. It is more suggested that learners should have opportunities to encounter, process and use instructed forms in their range of form-meaning relationships, in order for the forms to become part of their interlanguage behaviour.

In our discussion, we indicate that in order to attain high levels of proficiency in the target language, it is necessary to provide grammar feedback. Our aRaBCALL system demonstrates the need for formal grammar in order for learners to achieve high levels of accuracy, especially learners who are studying GCSE and A level.

If the aim in learning a second language is to develop communicative competence, a number of researchers have argued (e.g., Long and Doughty, 2011; Ellis, 1994; Robinson, 2001) that grammar and communication must be integrated in learning. However, as simple as it sounds, the main challenge is to identify the best ways of doing
so in Level 2 classrooms (Nassaji, 2010) and how we can maximize the opportunity to focus on the grammatical sides of the language without having to sacrifice the focus on meaning and communication.

We will outline the notion of the grammar that we will implement in our aRaBCALL system. We should base it on various situations such as our teaching experience, investigation of the British curriculum, and observing students’ needs. As a result, grammar is one of the important aspects of learners’ needs when learning a second language. We realized that learning the following fields are some of the important parts of our learning system:

1. Adjectives that cover colours, numbers, and nationalities
2. Verbs and how they change with regard to gender, person and tense
3. Negation article and its rule
4. Pronouns
5. Complementisers
6. Weak verbs

The grammar fields stated above have been extracted from the EdExcel learning criteria.

When the student inputs incorrect text for any of the grammar fields, the text is parsed, but a record is kept of the errors that have been made and a teacher can decide which errors should be brought to the student’s attention, which will be explained in detail in section 6.4.1.

6.2.2 Homework Marking and Feedback

Now we will describe the role of the aRaBCALL tool in language grammar learning. In this discussion, three related questions have to be addressed: What is homework for? What is the importance of homework? And how does it correlate to everyday life, its affairs and to lessons? In order to attempt to answer these three questions, we will need a formal understanding of the role of homework. Homework is one aspect of the general
education system that has been openly recognized as an important tool of success. Teachers have used homework to strengthen study and organizational skills and to provide additional learning time, and homework can teach student self-discipline and a sense of responsibility. From the perspective of classroom learning and teaching, homework allows learners to revisit topics they have been taught in class and cover them more thoroughly and with greater accuracy, so that the activity of self-study becomes a natural complement of teaching. It helps both teachers and students to see if what has been taught was understood fully or not.

There is another question that comes to mind, which is whether teachers should spend hours correcting student work. As Guénette (2007: 40) claimed in his answer “This is a fundamental question for second language teachers who are trying to help their students develop fluency and accuracy in their second language”. The correction feedback of student work could be either by direct correction, which can be verbal or written feedback, peer correction or group correction. There is another type of feedback, which is when stronger students may help weaker students and the weaker students constantly need to be encouraged and pushed to work harder. This brings us to the conclusion that students are different in getting the information and the help of other students becomes just as beneficial as the teachers. All learners would benefit from the teachers’ comments, but this is limited to the time and the amount of times a teacher can repeat their comments.

There have been many studies since the early 1980s on whether teachers should provide corrective feedback or not. Researchers left this as an opportunity to keep looking as there is no specific answer (Ferris, 2004). Tutors of foreign language believe homework exercises and self-correction either through online interaction or through a teaching CD/DVD will change the traditional way of teaching and learning a foreign language (Gibbs and Simpson, 2004; Lea and Street, 1998). In order to examine the effectiveness of online homework tasks and how error correction might be seen, a detailed study will be presented in the evaluation chapter (chapter 7).

Homework is important for the student as it can develop good habits and attitudes. It helps students to work independently and encourages self-discipline and personal
responsibility (Jha, 2006). Once the learner has been taught a topic, it is essential that they spend time on their own, reading and undertaking exercises at their own pace to ensure that they understand what they have been taught. From the learning perspective too, homework is very important, as it is a powerful way to extend classroom learning. Well formulated, appropriate and purposeful homework can reinforce and maintain. Homework can increase the involvement of each student with the learning task and can be one of the best tools to establish a unique relation between teacher and learner.

A teacher can examine the learners’ work in depth to find out their strengths and weaknesses and make comments and give guidance to correct and help them. Homework checking by the teacher can provide an opportunity for both the teacher and the student to identify which area is stronger or weaker among the major components of cognitive domain such as knowledge, understanding, application and skills. Students become more disciplined when studying on their own and can develop confidence.

In general, teachers assign homework for a variety of reasons: to help with reviewing a student’s progress, to practice and prepare them for the following lesson, to expand the students’ knowledge of topics more than class time permits and to help students gain skills in self-directed learning and use resources such as libraries. Hence, there are some kinds of standard homework exercises such as filling the gaps, comprehension, matching words, or to choose the correct answer. Our homework exercise is varied from the above exercises, in ways which we think are more challenging. With this exercise, the student has to decide whether the sentence is correct or not and if the sentence is wrong then they have to correct it. All the mistakes are related to the term of syntactic or morphological terms. Giving a student only an incorrect sentence and asking them to correct it would not be as beneficial as giving both correct and incorrect sentences. This is due to having the student being able to differentiate between the sentences, recognizing whether they are in fact correct or incorrect. This work is essential for students’ learning to ensure that they are on the right track and that they understand the work that is being set. This type of exercise plays an important role in the creation of the aRaBCALL system as it plays a core role in whether the student is learning key areas of the foreign language.
We want to provide effective learning feedback which will support the students’ learning progress. Normally, teachers give students one or two levels of feedback during an exercise. Teachers usually do not give a third level of feedback to weak students for many reasons such as time consumption, effectiveness. An advantage of having a computer provide third level feedback is that it provides feedback directly without the need of a teacher to write in a margin as a note. When providing feedback to the user, it is discouraged to receive additional feedback from the system after the user has already received feedback. This is done by scoring and penalizing the students for using and not using the feedback system. For some students the second level of feedback is not enough, so a third level of feedback, which is a detailed explanation of the source of the error, is provided. This is usually not provided by teachers because, as previously stated, it would be time consuming.

As Barr states that “…the learning experience can be made more enjoyable” (Barr, 2008:112), the scoring system we implemented helped to ensure this due to the students being much more motivated to achieve the best possible score. When using the computer system we give the students an opportunity to read and understand the error made and try to guide them to learn from the error by penalizing them so that it can be avoided in the future. Penalizing the students has two main advantages for the students; firstly, it tailors the amount of feedback they get and secondly, it increases the possibility for them to receive more feedback such as a detailed explanation of the error (see figure 6.6).

A type of feedback that can be used when helping students would be a three level feedback system. The three level of feedback are a sequence of feedback given. The first level of the sequence, a word message, is provided when the system highlights the word that had caused the sentence to be incorrect. The second level, a short error message, is when the student could not manage to correct the previous error to obtain a correct sentence. If the student requires more feedback after the second stage, a detailed feedback explanation is displayed in order to assist the student, which is the final stage of the three level feedback system. The reason for using this feedback system is that it allows the student to learn as to how they were incorrect in certain stages, whilst also allowing them to try and correct themselves. Due to these reasons, the three levels of
feedback that we have used with our learners are the steps that the system can offer with a reflection or an evaluation of the learners’ work. Always being able to offer students feedback is very beneficial as if a student was to incorrectly answer an exercise, minimal help would be provided for the student to try and correct themselves and if more feedback was needed, more detailed levels of feedback are provided. This also helps to distinguish between students to see whether or not they are capable of understanding the situation from a small amount of help. Using a three level feedback system can not only distinguish the levels at which the students are at, but also provide a more detailed explanation of the correction to improve their understanding and most importantly their learning.

6.3 The User Experience

In order to explain what our system does, we will take you through what a typical user will do. The login page has many instructions which we can point out. We want to provide our students with a three level feedback but discourage them to use them all. We cannot stop them from doing so but we will try to prevent them from doing so by giving a score if feedback is used too much. The login page contains more detail than a typical website. We have put a lot of information at the login page to help inform them on how to use the system. We do not use a help page because we suspected that they will not go to this page. Therefore, we put it on the login page to ensure that they do not miss it. Figure 6.1 shows the aRaBCALL login screen.
Chapter 6: Implementation of aRaBCALL Prototype

Figure 6.1: aRaBCALL login screen

When the user types their name the following page appears on the screen and it welcomes the learner using their name. See figure 6.2 below.

In these exercises, a series of Arabic sentences has been set. To test the learner understands such a topic, a mixture of correct and incorrect sentences has been set as explained in section 6.2.2. If the user thinks that what has been shown is correct, they can just submit it; if they think it has errors in it, there is a text box where they can enter
a corrected version. They can also copy the target sentence into the text box and edit it there if wanted; they can then submit it by clicking on the "submit" button. If the user makes the right decision they will score 5 points as shown in figure 6.3.

However, if there is an error in the sentence then the student will be deducted 5 points. If there are mistakes in what the user submits, they will be shown a copy of the text with the words that are wrong highlighted. The user can then choose to enter a corrected version once the words where the errors occurred have been seen, or they can click on the word to get a brief description of the problem. If they do not find this brief description useful, they can click to get a more detailed description of the issue in question. However, if the user chooses to click on the word, the score will decrease again. The user can then enter text into the box using English or Arabic characters, or through a virtual Arabic keyboard. Figure 6.4 below shows an error in the input.
If the user then decides to go on further, a brief description of the error on the page will occur as follows with the link as shown in figure 6.5.

By clicking on the determiner *ha*h another page will appear showing the detailed explanation of the error, shown in figure 6.6 below. The user will be deducted 7 points if they end up having to use this feedback level in the system. The page will explain the Arabic demonstrative pronouns *h*A, *h*h, *htAn, *h*An, and *hWlA* that have two ways in which they can be used. The two ways in which they can be used are as pronouns (words like ‘he’, ‘she’, and ‘it’) and as determiners (words like ‘a’, ‘the’, ‘every’). When these words are used as determiners, they have to agree with the nouns that they modify. The page will describe how the situation in Arabic is much more complicated than English. This is due to the fact that when the determiner attaches to a noun that denotes a human,
they have to agree in gender and number, but when the noun is not human, they only have to agree in gender. In addition, a precise reason why the answer is wrong will be clear as in the previous example, the user had used *ha*h to modify *nahor*?*, however, these two have to agree with the gender.

![Demonstrative determiners](image)

**Figure 6.6: aRaBCALL output screen – third level feedback**

When the user submits a correct answer, the system will report that there are no errors, as shown in figure 6.7.

![aRaBCALL output: no errors](image)

**Figure 6.7: aRaBCALL output: no errors**

We have seen the three levels of feedback in figures 6.5, 6.6 and 6.7 above which also consist of a short error description and a detailed explanation. What we hope is that once the learner sees the first feedback, they will be able to correct themselves and will not need to go to the next levels of feedback.
6.3.1 Log File

The server creates a log file automatically, which keeps the history of the page requested. A lot of information is then stored on this single log file such as date, time, page requested, HTML, user, etc. We can use this log file to analyse how the students use the system to study the most.

Our log file is also where we can find out about the students’ mistakes. Before analysing the log file, we have expected to find a correlation between these mistakes by mining the log file. As well as the log file being able to show us their mistakes, it can also show us their lack of Arabic vocabulary, stages of feedback the student has reached and many more stages of the system. To get information from students, we need to know what information we have got, how we will get it, how to analyse this information and what the learners will learn from it.

The log file is a XML file which contains various kinds of information. The reason it is made into a XML file is because we want to extract information from it, as it is easier to do it through XML. The top level entry within a log file is an exchange, which helps to determine who the user was and what they had asked. This file has two types of EXCHANGE functions. The following code shows the first type of EXCHANGE which represents the user’s log in time and period. The example code below shows that the user has chosen the Arabic keyboard:

(6.1)

```xml
<EXCHANGE DATE="Tue May 17 16:02:49 BST 2011">
  <QUERY>
    (RESULT=true, _V=chooseLanguage('majda', arabic)).
  </QUERY>
  <RESULT>
    <VAR NAME="RESULT"> True </VAR>
  </RESULT>
</EXCHANGE>
```
In the code (6.1) above, the query is what we want to send to Prolog, whereas the \texttt{RESULT} is what we will receive from it. In other words, to receive results, Prolog must submit the query: Prolog executes the command and returns back with the result.

For uniformity, we assume that the query will always contain a single non-anonymous variable called \texttt{RESULT}, which is where the information is returned. All user-generated events that we want to record lead to the generation of queries to Prolog. In some cases these are ‘dummy’ queries, included solely to ensure that all entries in the log file have the same format.

When the student submits a sentence to Parasite, it is recorded in order to know exactly when it has happened. This part of an exchange consists of the “query” mechanism, as we want to know when it happens and we have the result of it.

The other type of \texttt{EXCHANGE} function shows the labelled behaviour of the student’s work. When the student enters a word in the aRaBCALL system and that word was unknown, it would be highlighted in red and a message would appear on the screen explaining that the word was unknown. The example below shows how a query exercise presents a task when the student makes a mistake and accesses the first level of feedback. The second kind of exchange is established when we submit a sentence to Parasite.

(6.2)

\begin{verbatim}
<EXCHANGE DATE="Tue May 17 16:05:34 BST 2011">
  <QUERY>
    (retractall(soft(_, _)), assert(soft(subjAgree(_, _), 6)),
     assert(soft(_, 30)), javaCALL('Majda', 'lm *hb Albnt.', RESULT)).
  </QUERY>
  <RESULTS>
    <RESULT>
      <VAR NAME="RESULT">
        \textstyle{Your input had errors relating to the marked words. You can click on a word to see more about it: \&lt;br\&gt;\&lt;a href="basicerror.jsp?details=code/compTense!1/*ahaba!2/lm!msg/The verb '*ahaba' has the wrong tense for}
      </VAR>
    </RESULT>
  </RESULTS>
</EXCHANGE>
\end{verbatim}
The code above represents the second type of** EXCHANGE** which is when the input is processed by Prolog to provide information. The assertions **assert(soft(subjAgree(_, _), 6)),** and **assert(soft(_, 30)),** allow us to impose different penalties for different kinds of errors, in the present case, the exercise is targeted as the rules that deal with subject-verb agreement, so a lower penalty is applied for this kind of error than for any others.

Example 6.3 below shows a** QUERY** procedure without the** RESULT**.

(6.3)

<QUERY>
(retractall(soft(_, _)), assert(soft(subjAgree(_, _), 6)),
assert(soft(_, 30)), javaCALL('Majda', 'lm *hb Albnt.', RESULT)).
</QUERY>

The** visit** function shows us whether the learner has looked at the answer or not. The** visit** file is represented in the code 6.4 below.

(6.4)

<QUERY>
(RESULT=true, _V=visit('Allan',
"code/genderNomSent!1/sawodA`!2/qalam?!msg/You have written a nominal sentence, but the subject 'sawodA`' and predicate 'qalam?' don't agree properly.;"))
</QUERY>
The visit procedure in the code above is to indicate that the student has visited the second level of feedback (this is marked by the fact this result starts with the word “code”, which is used by the JSP page error.jsp as an identifier for second level feedback). In the above example the user has to match the gender marker of the word قلم qlm ‘pen’ with the adjective سوداء sawodA ‘black’.

### 6.4 Technical Details

Before we go through the detailed implementation stages, we need to go through the main components of our system.

Generally speaking, the user has interacted with the system, either through the virtual Arabic keyboard that is shown in the login page or through their normal keyboard. When the student manipulates an exercise by accessing one of the grammar questions set, pressing on the submit button will take the request to Parasite and process it. Thus, when the student has submitted a sentence to Parasite, the exchange function will have the time stamp to know when it has happened.

When we interact with the screen, there are two events that we can expect. The first event will tell Parasite to act on the procedure when the “submit” button is pressed. The other event does not include Parasite and will only reach the JSP Handler (see below) which is when for example the “next” button is pressed. Even when we change keyboards, Parasite is not involved in the procedure. For uniformity, we always ask Parasite to handle at least a dummy query, and what Parasite gives back to us, we may fix in order for Parasite to provide a variable result. So, what we send to Parasite is a query and what we receive is a result. In figure 6.8 below is the general picture of our system framework.
The above figure shows a brief overview of the current system in different steps. When the user accesses the login page, JSPHandler receives the message that this has happened resulting in two options. The first option that the JSP handler has is that the message can be interpreted as a request to load the arabic.jsp file with the specified arguments and send this back to the browser. The other option is that the message could be sent as an action to Parasite. If it is sent to Parasite it will process the query and return it to JSPHandler in order to create a URL that will be sent to the user. In other words, the browser behaves as the client and Prolog behaves as the server. The system above is run on a server because all the programs above cannot be stored locally.

The login page is a JSP page. When the user clicks on ‘login’, a message is sent from the browser on the user’s machine to Tomcat on the server. Tomcat will accept the JSP
file and compile it if necessary. When it runs, JSP handler reads the URL and receives the arguments. It then generates a page of JSP and sends it back to the client’s browser which displays it. When the user clicks on the submit button on this page, JSP handler looks at the submitted text and sends it to the Parasite server. The **Parasite** server analyses the text in Prolog as explained in the previous section, and the result of the query will be produced and be sent back as a string of HTML. This HTML code contains the following three components:

- A link to a file called basicerror.JSP
- A summary of the error
- The name of another JSP file

All this collectively gets put into a file and is sent directly to the browser. When the user clicks on a word, the browser asks to see basicerror.JSP with the argument supplied. Basicerror.JSP is then compiled and runs with these arguments creating a new file with a link to the special error file. When this link is clicked, the browser asks to see the error file, with the argument filled in. All the feedback messages that the student receives are written in the Roman alphabet.

### 6.4.1 Mechanism for Soft Parsing

Soft parsing is a functionality that is dependent on the existence of a mechanism for choosing whether to hard or soft parse. CALL systems should provide teachers with the means of deciding when to enforce and when not to enforce constraints and on exactly which elements, as well as the scope to moderate and even suppress error messages relating to those particular errors. Therefore, the use of soft parsing opens up a number of possibilities such as allowing a lot more flexibility and targeting of feedback, as well as being able to moderate feedback according to perceived degree of incorrectness.

One of the fundamental mechanisms of a soft parsing technique is constraints. To be able to make a constraint soft, you have to ensure you know the value of the variable
you are concerned with and see whether it has the value you want it to have. If it does not, then this indicates that there was a problem.

When you write a Prolog program you specify a set of constraints. For instance:

\[(6.5)\]

\[
\begin{align*}
\text{member}(H, [H|_T]). \\
\text{member}(X, [_H|T]):- \\
\hspace{1em}\text{member}(X, T).
\end{align*}
\]

The above code is a standard Prolog code for testing membership of a list. The first line of the code means if \(H\) matches the head of list then this means that these items looks the same. The ways it works is by unifying the first argument with the head of the second argument. The unification is expressing the relation between items. Basically everything we do for grammar involves constraints between features. To illustrate how we do soft parsing we look at the examples 6.6 and 6.7.

\[(6.6)\]

\[
\begin{align*}
X &= 0, \\
Y &= 1, \\
(X=Y).
\end{align*}
\]

In the last line of example 6.6, the values of variables \(X\) and \(Y\) are required to be equal, but since they have already been set to distinct values, the program will fail. Now, in soft parsing, we may want the program to succeed, even if some constraints are not satisfied. To have a program succeed even though the constraints are not satisfied, we could proceed as in example 6.7 below.

\[(6.7)\]

\[
\begin{align*}
X &= 0, \\
Y &= 1, \\
(X=Y \rightarrow \text{true}; \text{true}),
\end{align*}
\]
In the code above, the hard constraint \( X = Y \) fails because \( X \) is equal to zero and \( Y \) is equal to 1. This leads to the “else” part of the test construction being chosen as it is in fact the goal. Thus, the overall soft constraint succeeds even though the main test fails.

The above piece of Prolog succeeds but does not make any note of it. If the hard constraint is violated, the soft one is nonetheless satisfied. In order to provide the user with any feedback about which hard constraints were violated, we have to keep a note of what has happened. We do this by extending the information contained in a \texttt{SIGN}.

\begin{equation}
X = Y \text{ by } (X = Y \rightarrow \text{true}; \text{noteProblem} \ (\text{SIGN}))
\end{equation}

The above rule means that if \( x \) does not equal \( y \) then there is a problem. Now, the way we have noted the problem is by using \texttt{SIGN} as bundles of information about words and phrases. Problems are just bits of information: add them to the \texttt{SIGN}. Then we add an error code which contains a \texttt{short name}, and the \texttt{indices} of the \texttt{SIGNS} that have caused the problem.

As explained in chapter 5, our parser contains three parts: \texttt{STRUCTURE}, \texttt{MORPHOLOGY} and \texttt{SYNTAX}. A fourth category has been introduced, \texttt{REMARKS}, in order to store information on the violation of constraints in a feature of the structure of \texttt{SIGN}. In other words, we keep our softened message and the error code is stored. \texttt{SIGN} in Parasite looks like the following:

\begin{equation}
\text{SIGN} = \left[ \begin{array}{l}
\text{STRUCTURE} & \text{phon} \\
\text{MORPHOLOGY} & \text{morph} \\
\text{SYNTAX} & \text{syntax} \\
\text{MEANING} & \text{semantics} \\
\text{REMARKS} & \text{constraints}
\end{array} \right]
\end{equation}

It is the implementation of the last category which is related to the constraints, as it is the place to store the information about the problem. For example, if we want to soft
parse the case marker of the verb, such a constraint could state that a particular noun phrase has to be case-marked in a certain way. Each constraint has a name. This name gives the error class and usually stores the index number of the item that contains the constraint, in order to be able later on to retrieve the string and/or the linguistic analysis of this edge.

If $x$ or $y$ is currently unknown, $x = y$ will always succeed. If it turns out later that they were not equal, the problem will just fail as the code below shows that.

(6.10)

\[
\begin{align*}
X &= a, \\
(X = Y \rightarrow \text{true}; \text{problem}), \\
Y &= b
\end{align*}
\]

The problem is that when $x = y$ was tested, $y$ was not bound, and hence could match with $x$. It is only subsequently we realise that they are different, by when it is too late to treat the constraint softly. These are only if they are the same. The key is that we have to delay application of the constraint until we have all the relevant information. Consider the following ordinary Prolog in 6.11a and in 6.11b as these are equivalent, they both fail:

(6.11a)

\[
\begin{align*}
X &= 1, \\
Y &= 2, \\
X &= Y.
\end{align*}
\]

(6.11b)

\[
\begin{align*}
X &= 1, \\
X &= Y, \\
Y &= 2.
\end{align*}
\]
The following relaxed version of the 6.11b constraints will succeed, because although the conditional part of the expression \((X=Y \rightarrow \text{true}; \text{true})\) fails, the ‘false’ branch allows the whole thing to succeed anyway.

\[ \begin{align*}
(X = Y \rightarrow \text{true}; \text{true}).
\end{align*} \]

However, the code below fails because when we tried the constraints we did not know what \(Y\) was; so we bound it to 1, and then \(Y=2\) failed.

\[ \begin{align*}
X=1, \\
(X= Y \rightarrow \text{true}; \text{true}). \\
Y=2,
\end{align*} \]

Prolog lets you delay checking a constraint until you know everything you need to know. Consider the following examples:

\[ \begin{align*}
X=1, \\
Y=2, \\
\text{when } ((\text{nonvar}(X), \text{nonvar}(Y)), \\
(X=Y \rightarrow \text{true}; \text{true})).
\end{align*} \]

\[ \begin{align*}
\text{when } ((\text{nonvar}(X), \text{nonvar}(Y)), \\
(X=Y \rightarrow \text{true}; \text{true}), \\
X=1, \\
Y=2.
\end{align*} \]

In both cases, the test \(X=Y\) is not run until both \(X\) and \(Y\) are known. It therefore does not have the side effect of binding \(X\) to \(Y\). Thus, the conditional \((X=Y \rightarrow \text{true}; \text{true})\) succeeds, via the ‘else’ branch, without forcing \(X\) and \(Y\) to be the same.
In actual fact, when relaxed constraints are tested, there are two possible scenarios. This is either, the test succeeds and parsing continues as it would do with hard constraints, or the test fails and the name of the constraint is recorded in the list of failures. This list is part of the feature structure in one of the SIGN categories, REMARKS. In order to generate feedback, these failures or violated constraints will be inspected at the end of the parsing process.

To wrap up, the predicate soft is asserted to the Prolog database at runtime. This is done by the users when they choose to work with certain relaxed grammatical constraints.

When we want to soft parse with a specific constraint, there are four things we need to specify: the trigger, which specifies the information that we need to have before we can test the constraint; the test, which embodies the constraint that we want to check; the cost, which is a penalty to be imposed if the test fails; and the diagnosis, which is a brief error code describing what went wrong. The following example 6.16 represents the soft parse arguments:

(6.16)

\[
\text{softParse}(	ext{TRIGGER, TEST, COST, DIAGNOSIS, X}) : - \\
\phantom{=} \text{[remarks, score]}@X \leftarrow \text{[remarks, score]}@Y, \\
\phantom{=} \text{trigger}(	ext{TRIGGER, doSoftParse(TEST, COST, DIAGNOSIS, X)}).
\]

\[
\text{softParse}(	ext{TRIGGER, TEST, DIAGNOSIS, X}) : - \\
\phantom{=} \text{[remarks, score]}@X \leftarrow \text{[remarks, score]}@Y, \\
\phantom{=} \text{trigger}(	ext{TRIGGER, doSoftParse(TEST, DIAGNOSIS, Y)}).
\]

\[
\text{doSoftParse}(	ext{TEST, COST, DIAGNOSIS, Y}) : - \\
\phantom{=} \text{(TEST} \rightarrow \\
\phantom{=} \text{true;}
\phantom{=} \text{(soft(DIAGNOSIS, _COST),}
\phantom{=} \text{(COST} \gt 0 \rightarrow \text{addFailure(Y, COST, DIAGNOSIS); true))).
\]

The above code represented the soft constraint procedure. This means that if the constraint succeeds then it is true. Otherwise, DIAGNOSIS is what kind of constraint we
want to check. When the constraint fails, a penalty score will appear which is represented by the variable COST. The TRIGGER is used to decide when to actually check the constraint. The following codes soften the entry for the past tense:

\[(6.17)\]

\[
\text{softParse}(\text{lextype}@\text{X0}, \text{past}(\text{X1}), \text{tense}(\text{index}@\text{X0}, \text{text}@\text{X0}), \text{X1}).
\]

As seen in the code above, the constraint is attached to the affix, but we cannot check it until we know the form of the verb we are attaching it to. We use the LEXTYPE of the verb as the TRIGGER, since this is something that will not be known until we know what the affix is being attached to. The test is then that the verb should be a past form, and the error code indicates that there is a problem with the tense marker.

In other words, the constraints on past tense affix are delayed until the phrase that is in question has been found by the parser. The actual linguistic rule that has to be tested in order to confirm whether or not it has been violated is in the third variable TEST. It is sometimes necessary to join two lists of constraints of two different linguistic signs, for instance, combining morphs of a word with each other. In this case, every time the parser produces a new parsing edge, the constraints are checked. In other words, before each SIGN – lexical, phrasal or sentential – is built, the constraints are checked.

In our aRaBCALL system we have a set of exercises about subject and verb agreement. We allow learners to make mistakes with subject and verb agreement, but you get a penalty of six. For any other mistakes, the program will set a penalty of thirty as illustrated in the following example.

\[(6.18)\]

\[
\text{soft(subjagree}(\_\_\_), 6) \\
\text{soft(\_\_, 30)}.
\]

We tested the machinery of the system in the case to ensure the coverage of all the student errors. We will be setting these out in order for the system to accept these errors, so that testing will allow the system to see all the possibility of the students’ errors and
analyse it providing useful feedback. In the following table, we will be numerating all the words to ensure that the system is robust.

In the following table, all the errors a student could possibly make when dealing with Arabic verbs are listed. These errors are caused when the student adds a wrong affix, thus changing the structure and the meaning of the sentence.

<table>
<thead>
<tr>
<th>Gloss</th>
<th>Verb Empty Prefix</th>
<th>Verb Prefix ‘O’</th>
<th>Verb Prefix ‘n’</th>
<th>Verb Prefix ‘y’</th>
<th>Verb Prefix ‘t’</th>
</tr>
</thead>
<tbody>
<tr>
<td>$rb</td>
<td>شرب</td>
<td>O$rb</td>
<td>n$rb</td>
<td>y$rb</td>
<td>t$rb</td>
</tr>
<tr>
<td>$rbnA</td>
<td>شربنا</td>
<td>O$rbnA</td>
<td>n$rbnA</td>
<td>y$rbnA</td>
<td>t$rbnA</td>
</tr>
<tr>
<td>$rbi</td>
<td>شربت</td>
<td>O$rbi</td>
<td>n$rbi</td>
<td>y$rbi</td>
<td>t$rbi</td>
</tr>
<tr>
<td>$rbtA</td>
<td>شربتا</td>
<td>O$rbtA</td>
<td>n$rbtA</td>
<td>y$rbtA</td>
<td>t$rbtA</td>
</tr>
<tr>
<td>$rbtn</td>
<td>شربتن</td>
<td>O$rbtn</td>
<td>n$rbtn</td>
<td>y$rbtn</td>
<td>t$rbtn</td>
</tr>
<tr>
<td>$rbtnA</td>
<td>شربتانا</td>
<td>O$rbtnA</td>
<td>n$rbtnA</td>
<td>y$rbtnA</td>
<td>t$rbtnA</td>
</tr>
<tr>
<td>$rbtm</td>
<td>شربتم</td>
<td>O$rbtm</td>
<td>n$rbtm</td>
<td>y$rbtm</td>
<td>t$rbtm</td>
</tr>
<tr>
<td>$rbtmA</td>
<td>شربتام</td>
<td>O$rbtmA</td>
<td>n$rbtmA</td>
<td>y$rbtmA</td>
<td>t$rbtmA</td>
</tr>
<tr>
<td>$rbw</td>
<td>شربو</td>
<td>O$rbw</td>
<td>n$rbw</td>
<td>y$rbw</td>
<td>t$rbw</td>
</tr>
<tr>
<td>$rbA</td>
<td>شربا</td>
<td>O$rbA</td>
<td>n$rbA</td>
<td>y$rbA</td>
<td>t$rbA</td>
</tr>
<tr>
<td>$rbAn</td>
<td>شربان</td>
<td>O$rbAn</td>
<td>n$rbAn</td>
<td>y$rbAn</td>
<td>t$rbAn</td>
</tr>
<tr>
<td>$rbn</td>
<td>شربين</td>
<td>O$rbn</td>
<td>n$rbn</td>
<td>y$rbn</td>
<td>t$rbn</td>
</tr>
<tr>
<td>$rbwA</td>
<td>شربوا</td>
<td>O$rbwA</td>
<td>n$rbwA</td>
<td>y$rbwA</td>
<td>t$rbwA</td>
</tr>
<tr>
<td>$rbAt</td>
<td>شربات</td>
<td>O$rbAt</td>
<td>n$rbAt</td>
<td>y$rbAt</td>
<td>t$rbAt</td>
</tr>
<tr>
<td>$rbyn</td>
<td>شربين</td>
<td>O$rbyn</td>
<td>n$rbyn</td>
<td>y$rbyn</td>
<td>t$rbyn</td>
</tr>
<tr>
<td>$rbwn</td>
<td>شربون</td>
<td>O$rbwn</td>
<td>n$rbwn</td>
<td>y$rbwn</td>
<td>t$rbwn</td>
</tr>
</tbody>
</table>

Table 6.1: The expected verb morphological errors
6.4.2 Error Code and JSP

The error codes that our Prolog program will produce when soft parsing constraints are violated would not be very informative to our learners. These error codes are important in order to generate meaningful feedback. This is done through the JSPHandler causing the error code to become readable. Error code is used to provide information for JSP. All the error messages are sent to the JSPHandler. Most of the rules in the file will be explained below with relevant examples. At the end, the system will report any misused items in each given exercise.

In the previous subsection 6.3, we mentioned that the system provides three levels of feedback. We will now discuss how the error codes are used for generating this feedback. We are going to generate the feedback by accessing JSP pages with information that will be used to fill in the blanks in the JSP. Consider the following codes for the diagnosis of the agreement markers in 6.19:

\[(6.19)\]

\[
\text{diagnoseFailure(agr(I, T)) :-}
\]
\[
\text{retrieve(I, X),}
\]
\[
\text{underlyingForm(X, S),}
\]
\[
\text{error("The agreement marker '~w' doesn't match the rest of '~w'",
}
\]
\[
[T, S], agr, [T, S]).
\]

From the above code \texttt{agr(I, T)} we can see that \texttt{agr} has two items \texttt{I} and \texttt{T}, which are generated as a parser code. The first item is an index that identifies a particular word. The second item, \texttt{T}, is the form of the affix itself. The term \texttt{agr} is used in order to generate the link to the JSP file in the second level feedback page. The code is to print a message or a text to the user which represents the first level of feedback. It has four parts of arguments which are as follows:

- The first argument is part of the error message such as "The agreement marker... ."
- The second argument part is the note for the inference which represents by ' ~w'.

179
Chapter 6: Implementation of aRaBCALL Prototype

- The third argument in the error is “agr” which is the name of the JSP file.
- And the last argument is [T, S] which is represented the web page that appears to the user. All these codes will be generated by JSP pages.

Now let us discuss an example of a word Oktbwn أكتبون ‘to write’ with an agreement failure between the tense marking prefix O and the agreement suffix wn ون.

(6.20)

| ?- in arabic.  
| Input a sentence in Arabic  
| ?- Oktbwn ^

```
%% Time taken: 31 milliseconds  
%% Parse 12 (cost 150):

/**** LABELLED TREE ***************
[[{"'{'"Ou"}'', '{'('I', 'k?t?b')'}'}, o(tns1)''}, wuna}),
 (arg(agent) - ['0'])]

/**** DEPENDENCY TREE ***************
{{{{Ou},{I,k?t?b}},o(tns1)},{wuna}}

0

/**** MEANING 12: Oktbwn

exists(_A,
    {event(_A, 'k?t?b')
    & (theta(_A,
      agent,
      ref(lambda(B, centred(_B, lambda(C, human(_C))))!_D)
      & aspect(now, simple, _A))}

This analysis had the following problems: [agr(5,wuna)|_15200]+_15203
```

Most interactions with Prolog use a command line interface. When the student uses the system we bypass the command line and let them use the web based GUI. The example below shows that when we use the command line interface at the very end, we end up getting an error code. This error code is used to generate the HTML, which marks the word with the error on it as shown in code (6.20) above.
The code below is to show how we diagnose the failure of the use of the agreement marker which is ‘uwna’. The output contains a few parts, the code is `agr`, number 1 which represents the affix ‘uwna’, number 2 is for the underlying form of the full word which here is for the ‘Oukotibuwna’ and finally the full message which is ‘The agreement marker ‘uwna’ doesn't match the rest of ‘Oukotibuwna’’. All the string above is sent to the basicerror.jsp file. The JSP file will generate a string of HTML for the first level of feedback as shown in the code below.

(6.21)

```html
<a href="basicerror.jsp?details=code/agr!1/uwna!2/Oukotibuwna!msg/The agreement marker 'uwna' doesn't match the rest of 'Oukotibuwna';"><red>Oktbwn</red></a> <br>
```

The above code generates the screen below which is the first level of feedback, with the word ‘Oukotibuwna’ highlighted in red to show that there is an error associated with it.

![aRaBCALL output screen](image)

**Figure 6.9: aRaBCALL output screen - first level of feedback for the word أكتبون Oukotibuwna**

However, if the user wants to reach the second and third level of feedback, there is some HTML code link to the agr.jsp and that is shown in the code below. To reach the
second level of feedback and when the student clicks on the incorrect word, the link that was associated with the word ‘Oukotibuwna’ takes the user to the JSP page ‘basicerrors.JSP’ with the given arguments. This in turn generates a page of HTML containing the link shown in (6.22). The appearance of this page is shown in figure (6.10).

(6.22)

```html
<a href="agr.jsp?arg1=uwna&arg2=Oukotibuwna">The agreement marker 'uwna' doesn't match the rest of 'Oukotibuwna'</a>
```

Figure 6.10: aRaBCALL output screen - second level of feedback for the word أكتبون Oulkotibuwna

Clicking on the link in figure (6.10) takes us to another JSP page, namely agr.JSP, as specified in the original error code, again with appropriate arguments, which in turn generates the tailored HTML in (6.23) and figure (6.11).

(6.23)

```html
<body>
<h2 style="text-align: center;">Agreement markers</h2>
Arabic nouns and verbs have 'agreement markers' which tell you whether the word in question is singular, dual or plural. These have to match other elements of the word that they are attached to--you can't add a plural ending to a singular noun, or a singular ending to a dual form of a verb, or ... <p>In the current case the suffix "uwna" clashes with the rest of the word "OuIkotibuwna".<br><p><p>
<A HREF="http://ramapp.cs.man.ac.uk:8080/CALL/arabic.jsp">back to exercise page</A>
</body>
```
The final feedback screen represents the third level of feedback.

**Agreement markers**

Arabic nouns and verbs have ‘agreement markers’ which tell you whether the word in question is singular, dual or plural. These have to match other elements of the word that they are attached to -- you can't add a plural ending to a singular noun or a singular ending to a dual form of a verb, or ... In the current case the suffix "uwna" clashes with the rest of the word "Oulkitibuwna".

**Figure 6.11: aRaBCALL output screen - third level of feedback for the word أكتبون Oulkitibuwna**

6.5 Specific Errors

Our learners of Arabic may find the most difficult things to learn are the areas of morphology and syntax, as, from our previous experience of teaching, pronunciation and meaning are easier. This is because of firstly the lack of diacritics, secondly the nature of free word order and thirdly the zero items pronouns and copulas. In this section, we will illustrate the errors that learners of Arabic will encounter.

As explained in the previous chapters, learners are encouraged to check existing sentences for the various types of morpho-syntactic errors. In addition, they are encouraged to produce sentences freely in various situations and context, with the system providing them with a detailed explanation if mistakes are made.

6.5.1 Treatment for Morpho-syntactic Errors

Chapter 4 was devoted to the meaning of the morpho-syntactic errors, which leads to us demonstrating, in this section, all the types of the morphological and syntactic errors that Parasite has treated (appendix A shows all Parasite’s soft parsing codes).
1. Incompatible affixes

- **Misused verbal affixes**

  *e.g.* يكتبت الدارسة الدرس yktbt AldArsp Aldrs. ‘The learner wrote the lesson’.

  In this example, the learner uses the present tense prefix masculine ي y with the past tense suffix feminine ت t.

2. Inappropriate lexical classes

- **Misuse of tarmarbuta‘ت’**

  *e.g.* مدرسته بعيدة mdrsph bEydp ‘His school is far’.

  For this example, we did a soft constraint for the misuse of clitics such as تاء مربوطة taa` marbuwTap, as the feminine name can end with this, however, when we add a diacritic to the end to make it dual such as ان ان An, the تاء مربوطة taa` marbuwTap should be changed to the normal ت t as seen in examples مدرسة mdrsp ‘school’ and مدرستان mdrstAn ‘two schools’.

- **Misused word number clitics**

  *e.g.* قرأت كتابون كثير qroT ktabwn kvyrp ‘I read a lot of books’.

  Certain words do not take standard affixes when in the plural. The plural of these nouns is in different broken patterns. A regular masculine plural is formed by adding the suffix ‘ون wn’ to the singular nouns, and the suffix ‘ات At’ for the singular feminine noun. However, broken plurals are formed differently by mainly changing the diacritic patterns (see 3.3.5.1).

3. Word order constraints

- *e.g.* رجل يكتب كتب rjl yktb kib ‘A man writes book’.

184
The noun in the beginning of the sentence should be definite; therefore, the word ‘رجل’ must have a prefix ‘ال’.

4. Complementiser patterns

- Misused complementiser – On and On~
  e.g. إعتقد المعلم أن درس الطالب IEtqd AlmElm On drs AlTAlb ‘The teacher thinks the student has studied’.

- Misused Negation
  e.g. لا ذهب الولد إلى المدرسة LA *hb Alwld ILY Olmdrsap ‘The boy does not go to school’.

The complementisers are quite difficult with constraints especially with أن On and On~ لأن because they are written the same.

5. Agreement errors

- Misused gender-number
  e.g. أكتب الدارسان الكتب Oktb AldArsAn Alktb. ‘The learners write a written task’.

- Misused adjective - noun
  e.g. هذه نهر طويلة h*h nhr Twylp ‘This is a long river’.

- Misused definite determine ال Al with adjective – noun
  e.g. هذا نهر الطويل h*A nhr AlTwyl ‘This is a long river’

6. Misuse of Clitics

- Wrong affixes in terms of gender/ verb tenses
  e.g. أكتبون الدروس Oktbwn Oldrs ‘They write the lesson’.

The above subsections have outlined the morphological and syntactic errors that our system covers.
6.6 Summary

This chapter has discussed all the stages of aRaBCALL’s implementation. The initial explanation began with a detailed analysis of the system’s functionality, followed by the mechanism of the interface. It was noted that while it is clearly desirable to have the aRaBCALL system able to recognize correct input, the current system is also able to make use of the information contained in badly formed sentences. The implementation of soft parsing in Parasite was introduced and explained.

Relaxing all the constraints that can be relaxed is a far less effective strategy than targeting specific language points and then softening the relevant constraints. However, on occasion it may be desirable to have more than one constraint softened at any given time. Parasite avoids the problem of generating numerous and hence confusing feedback by selecting the parse tree with lowest cost errors to send to the screen.

By relaxing certain constraints on unification, correct information contained in a badly formed sentence may still be used to obtain a coherent overall analysis. Thus, the system is able to produce a model of a well-formed version of the user’s intended input. A number of examples were given where morphological errors had been soft-parsed. In particular, errors relating to verbs that require minor modification to the regular formation of rules and adjectival agreement were examined, as well as non-standard number suffixes.
Chapter 7

Evaluation

7.1 Introduction

In this chapter, we want to discuss how we evaluate our CALL system. We are not primarily interested in the usability of the system. The evaluation we carry out is mainly about the effects of the tool and the way that students use it. The standard way to evaluate a system is through the questionnaire in section 7.3.1. We want to ensure that our system has fulfilled its purpose by finding out two things from our users: whether they like it, and whether they benefit from it. We need to consider both these issues, since a system with excellent pedagogical outcomes that students refuse to use is not much more useful than one that engages their attention but which fails to provide useful feedback. This investigation will consider the usefulness of diagnosing errors and also for future reconstructions. Generally, the main reason why we want to evaluate aRaBCALL from different angles by using various methods is to see if the system is useful and how much can be learnt with a certain teaching method, namely the homework procedure that we have focused on in this research. This kind of testing will add strength to our decision making during our research.

Research concerning the evaluation of CALL systems has been made from various perspectives (Jakobsdottir and Hooper, 1995; Lam and Pennington, 1995; Lee, 2000; McEnery et al., 1997; Ma, Q. and Kelly, 2006). Many studies show that CALL systems are more effective than traditional human instruction when teaching a second language (Yang and Akahori, 1998; Hubbard and Levy, 2006; Hamel, 2008). We will thus be investigating the influence of the aRaBCALL learning aid from the learner’s point of view. A text file (log file) is also analysed, as it presents a trace of the
users’ behaviour when interacting with a Web-based technology, which will provide an objective counterpart to the subjective feedback from students.

Our initial aim was to use the log file to provide objective evidence that students’ actual behaviour corresponded to what they reported via the questionnaire. However, it turned out that the log files also showed different patterns of usage. When viewing the log file at various angles, it can show different student behaviour that can be useful for us as it helps to show how the students look at the system. There are a number of common paths; table 7.1 shows how the students perform and how they behave. Analysing these traces may provide useful information about individual students, and could also be used as a guide for subsequent development of the system. Exploring ways of using these behaviour traces falls outside the scope of this thesis, but could form a fruitful avenue for further work.

In this chapter, we will analyse different mechanisms, which we have adopted for the evaluation of aRaBCALL, such as questionnaires, interviews and the analysis of the log file. Evaluating the user viewpoint came through a questionnaire that was designed and distributed to students. To further refine the knowledge gained from the questionnaire, we interviewed the students in order to receive a more detailed elaboration of their answers. The items in the questionnaire consist of specific questions related to the system’s ease of use, features of the exercises and the detailed explanations of the parts of the exercise questions. These techniques are representative of the techniques used in any evaluation. On the other hand, analysis of the log file is used as it is created by the operation of the computer system in which it records all the user activities. The text file (log file) shows all the access information after the learners used the system.

Follow-up interviews were conducted with the learners to gain an understanding into their decision-making process and also develop an understanding of their willingness and ability to access the aRaBCALL exercises.

We did not discuss the usability of the tool, although there is a part in our questionnaire regarding it. King (2005) mentions this in her research and claimed that many academic evaluation exercises concentrate on a software system taken in isolation,
looking primarily at what it is supposed to do, and ignoring the context. User-oriented evaluation adopts a radically different perspective, taking as primary a user or set of users who need to accomplish some task and asking whether the system will help them to do so effectively, productively, safely, and with a sense of satisfaction (King, 2007). This implies looking at a large and complex set of factors which will contribute to whether, in the end, a decision to acquire and deploy the system will seem to have been a good decision (ibid). Frequently, the factors involved are not independent of one another, either conceptually or in the ways that each factor may contribute to an overall judgement.

The Advanced Research Projects Agency (ARPA), sometimes also known as the Defence Advanced Research Projects Agency (DARPA), initiated the earliest evaluation campaigns. The aim of the campaigns is focused on advancing a core technology by promoting competition amongst research teams working in that domain. Participants are typically expected to take part in regular conferences, where the results achieved by any one participating system are compared to the results of all other systems (King, 2005). There have been a number of such campaigns, some running over very long periods of time. They concentrate on the software system in which they have measurable criteria in which they work to. We will not be able to use this method in our system as our system concentrates on working with students and is not easy to measure nor is it easy to set as a measurable criteria task.

Thus, the following sections will focus on the detailed analysis of the evaluation techniques and a detailed study of the types of evaluations and pedagogical views denoting the view on a system task measurement and concentrating on individual feedback for current and future development. The chapter contains the following.
• We will present the purpose of measurement
• We will talk about the definition of evaluation and types of evaluation
• We will focus on system task measurement and specifically on questionnaire forms and results
• Then, we will create a relevant picture from the interviews that are conducted with the learners’ point of view
• We will analyse the log file in terms of evaluating the student interaction procedures

7.2 Definition of Evaluation

Ebel and Frisble (1991) define evaluation as “...an information-gathering process that results in judgments about the quality or worth of a performance, product, process, or activity” (Ebel and Frisble, 1991: 38). Evaluation involves analysis of information about the use or performance of the system that makes it possible to judge its effectiveness for the task for which it is intended (Ellington et al., 1993). The goal of our evaluation is to make a judgment about the quality or worth of our aRaBCALL system as an educational program, its performance or efficiency or its effect on student attainments. The evaluator must decide what kind of information is needed, how the information should be gathered and how the information should be synthesized to support the outcome and the value of the judgment. Thus, evaluation is a connection of information gathering with making decisions (ibid). In this section, we will introduce the idea of evaluation and some of the major terms and issues in the field in order to identify the best way of evaluation to improve the aRaBCALL learning aid in the future.

7.2.1 Review on Evaluation

In this section, we will go on to explain the many different purposes for the evaluation of CALL software, such as selection for a course, and explain the three major approaches to CALL evaluation which are checklists, methodological frameworks and SLA
Software evaluation is an important area of CALL and its role may be increasing, especially within the domain of empirical evaluation, which is the evaluation by means of direct or indirect observation or experience. For most language teachers, it is primarily a judgemental process, with some empirical follow up of evaluation but due to its time demanding process it can be rather impractical for some classroom teachers.

In support of our review we need to consider previous work on evaluating CALL systems to integrate a methodological framework to our evaluation. A report published in the CALICO journal (2006) by Hubbard explains that there are different purposes for the evaluation of CALL software such as selection of software for an individual course or for self-access and reviews. Selection for a course is when a teacher produces an evaluation to select appropriate software for his or her own class. In this evaluation process, information such as an understanding of technical infrastructure of the institution, data on other course materials, students’ characteristics and assumptions on how the language is learned can be obtained. Selection for self-access is similar to the former, however, it is used to recommend software selections for a self-access lab or an entire language program. Information on student characteristics, course objectives and materials may be less readily available. In addition, a great deal of variability in student and course characteristics makes the selection process more challenging. Reviews generally differ from other forms of evaluation as they typically focus on the software itself rather than the environment in which the software is used as they are aimed at a broader audience of potentially interested parties. A review in form of evaluation is a vital source of information in that others can use it in initial identification of possible candidates and in informing their own evaluations.

There are three separate domains when evaluating CALL systems, which are thought of as selection, implementation and assessment. Selection is the investigation of a piece of CALL software, judging its appropriateness for a given language learning setting. Implementation identifies ways in which it may be effectively implemented in that setting. The third stage of the evaluation of CALL is assessment, which our aRaBCALL system evaluation relies on, as it is the assessment of the degree of success and
determination of whether to continue use or to make adjustments in implementation for further use.

There are three major approaches to CALL evaluation that Hubbard (2006) identifies from research and describes as:

1. **Checklists** – These have been present from the earliest stages of CALL and remain widespread. A checklist usually consists of a series of questions or categories for judgment in which the evaluator makes an expected response based on information gathered from the review. Checklists may simply ask for a yes or no indication, whilst some may include space for open-ended commentary. However, these have been criticized due to the expense of pedagogy and also being biased and restrictive. Checklists can be adapted and updated for particular purposes and have the capacity to provide teachers with a useful tool in recognizing the various elements that make up a software application and for triggering reflection on some of their own assumptions on CALL.

2. **Methodological Frameworks** can be compatible with some checklists but vary in two specific ways; they concentrate on being descriptive rather than judgemental and they attempt to link with language teaching and learning considerations that take place out of technology. Hubbard (1988) explains that a framework in this context is an integrated description of the components of CALL materials with respect to evaluation. A framework provides a tool in which an evaluator can make their own questions rather than asking a specific set of questions. The framework Hubbard (1988) produced was an adaptation of the approach, design, and procedure constructs into categories describing the key elements of evaluation and hence renamed them teacher fit, learner fit, and operational description, respectively. The resulting framework developed into the evaluation module in a proposed comprehensive methodological framework, which included modules for courseware development and implementation.

3. **SLA-based Approaches** - Given that teaching languages with software is a form of language teaching, this is another reasonable procedure for developing
software evaluation. Ultimately, we might expect to have definitive SLA results specifically from research on learning with software, but to date there has not been a sufficiently established base for such results. When discussing SLA-based approaches, Chapelle (2001) represents a significant advance for the characterisation of evaluation on the basis of principles and offers a set of five principles when evaluating CALL. The five principles are that CALL evaluation is situation-specific, it should be evaluated both judgementally and empirically, the criteria should come from instructed SLA theory and research, be applied relative to the purpose of the CALL task and also should be language learning potential. Chapelle also proposes six general evaluation criteria useful in determining the appropriateness of a given CALL task in supporting language attainment: language learning potential, learner fit, meaning focus, authenticity, positive impact, and practicality. These are in some respect compatible with the methodological framework and checklist approaches, though are slightly different in structure and in underlying assumptions.

All these approaches relate to our study and can be applied to the aRaBCALL system. The checklist will be identified in the design of the questionnaire in the form of yes/no questions, the methodological framework will be integrated in the face to the face interviews and comments and the SLA approach is integrated in the syllabus. All these will be described in further detail in section 7.3. The resulting framework became the evaluation module in a proposed comprehensive methodological framework to our work. This section has outlined approaches to evaluation in the forms of checklists, methodological frameworks and SLA research. Even though these three may have their merits, the next section will concentrate more in detail on the methodological framework as it is the most neutral in terms of the language teaching approach.
7.2.1.1 General Evaluation Framework

The framework outlined in this section can be readily extended to any of the domains mentioned previously, that is, selection for self-access or other instructors. Software packages are often complex, including a number of different types of presentations, activities, exercises, and quizzes, so in a thorough evaluation it will be necessary to cover examples of each type.

Before the beginning of any evaluation, preliminaries are set by candidates that will be evaluated being identified that are in most cases designed for language learning, which is known as the identification of potential software. To begin these preliminaries, a basic understanding of what characteristics to look for in candidates is needed. The software evaluation framework is when core components at this level, even without further analysis, reflect a simple evaluation procedure embodying the following stages:

- Technical preview – Ensuring the software runs the way it should be on the equipment available for the students
- Operational description – Go through the software as a cooperative user and get an understanding of the flow of lessons and items within before making any judgements, recording any first impressions without judgement until the system software’s operation is completely understood.
- Teacher fit – Inferring the language teaching approach that the software reflects and determining the degree to which it is compatible or incompatible with our own.
- Learner fit – How well the software fits the interests of students and their preferred learning styles and how well it fits with the content, language levels and skills relate to the students’ needs, especially as mentioned by the objectives of the syllabus.
- Implementation schemes – How the software may be integrated into the course or a curriculum, what students will need to know for efficient use and the time taken for the process.
- Appropriateness judgments – Make a decision to use or not based on the degree and the quality of teacher and learner fit, with the considerations of the costs to
produce and benefits of implementations. It could be judged in the same way a human teacher would be judged.

There are technical considerations of several types but the most basic is which machine the student will be using when accessing the system. This could be from the simple Microsoft Windows-based PCs to the Apple Macintoshes. Some software can run all different platforms and are particularly web-based applications. When dealing with software that is online, there can be issues of bandwidth and server access as the speed of which data is transferred can impact the use of the system.

The operational description is a review of components of the software and how these can operate on their own or can be controlled by the user. This was originally presented as a set of more or less independent central and peripheral descriptive categories, remaining a useful classification. The peripheral categories include any text, documentation, tutorial, record keeping features and any other utilities. The presentational scheme describes the way a CALL activity can be presented to learners that can be defined by:

- Screen layout or interface and the aspects of basic appearance
- Timing for some software in which the content material or prompt appears on the screen
- Control options describes what is under learner and program control as well as the physical nature of those controls
- User input characterises how the learner responds to implicit or explicit prompts of the program
- Input judging describes the program’s procedure for handling user input such as the recording of a mouse click
- Feedback provided to the user by a program as the result of input judging, which is a key part of the presentational scheme as there are a number of options. These can clearly represent a more active teaching presence than others and can be implicit, when an incorrect answer disappears as a choice, or explicit. Feedback can simply indicate a correct or incorrect response or provide additional information in forms of hints or explanations.


- Help options represent the final element and is when assistance provided is contextualized and targeted for an item rather than being global, and whether it is available all the time or under certain conditions.

The final area of evaluation process is the determination of the degree to which the software can be used and the manner it is used has been successful. This helps to let the teacher decide whether or not the software can be used in the future and if they would use it in the same way or differently. This part of the evaluation also gives a general understanding to the teacher as to what the students do with the software and can influence future evaluations. Although this is an important part of evaluation, it can be quite challenging and also time consuming to achieve in practice. The most direct way in which this part of evaluation can be obtained is by watching the students as they use it, as the teacher can walk past the students, observe their reaction to it and interact with the students as they interact with the software. This method of retrieving information can be used both to evaluate the software and to inform the on-going learner training.

Possibly, the best method in retrieving objective information on student use is to track the students’ actions, observing any patterns emerging from it and to examine their behaviour. Another approach to gathering information can be through surveys and questionnaires. This information can be valuable but there are two concerns. Firstly, students know their responses are tied to a grade or other assessment and if they believe this, results can be compromised, thus, anonymity must be ensured if feasible. Secondly, when students try to be honest their reports do not necessarily correspond to their actions. If surveys are used then it must be advised to have them administered during or immediately after completion of a CALL activity to ensure accuracy.

### 7.2.2 Types of Evaluation

In general, there are two types of evaluation: (i) subjective evaluation, which tells the evaluator how the users feel about the tool that is being tested. The usual method of assessing this is to use a questionnaire; (ii) objective evaluation which is about evaluating the tool from the output of the user result.
The different types of evaluations are dependent on the object and purpose of the evaluation being carried out. Possibly the most important basic distinction in evaluation types is that between formative and summative evaluation. Formative evaluations improve or strengthen the object being evaluated; they help it by examining the delivery of the program or technology, the quality of its implementation, and the assessment of the organizational context. In contrast, summative evaluations examine the outcomes or effects of some object. They summarize these effects by describing what happens prior to delivery of the program or technology; assessing whether the object can be identified to be the cause of the outcome; determining the overall impact of the causal factor; and, estimating the relative costs associated with the object.

To evaluate our system, we have conducted both subjective and objective evaluation in order to carry out formative evaluation. It also helps to judge the strengths and weaknesses of the aRaBCALL developing stages, for the purpose of revising the system. The purpose of formative evaluation is to investigate how to improve our system in order to make the instruction more effective, efficient and also to assess its usability and appeal. For aRaBCALL, two types of subjective evaluation were used: questionnaires and interviews. The objective evaluation of aRaBCALL was carried out in order to investigate the relationship of the learning behaviour and the system style for future work. In section 7.3, we will have a detailed discussion with regard to the subjective evaluation, and in the same section we will analyse the objective trace of user behaviour provided by the log file.

**7.2.3 Questions and Methods: System Task Measurement**

Evaluators ask a variety of questions and use different methods to address them. Different kinds of questions are considered within the framework of subjective evaluation. The major questions and methodologies used with a research are:
• Formulating and conceptualizing methods may be used including group techniques, brainstorming, Delphi methods, lateral thinking, input-output analysis and concept mapping.

• The most common method used is ‘needs assessment’ which can include: analysis of existing data sources, and the use of sample surveys, interviews of constituent populations, qualitative research, and focus groups.

The evaluation techniques should be delivered to test the quantitative and qualitative monitoring techniques, the use of management information systems, and implementation assessment would be appropriate methodologies.

7.3 Evaluation of aRaBCALL

The author has developed an aRaBCALL learning system based on an NLP based parser. The system was developed using NLP techniques and the results of error analysis will be evaluated for the purpose of pedagogy. The evaluation of the analysis in the real world will be through online questionnaire and through an Internet log file. The log file is built up while the learner is manipulating and interacting with the aRaBCALL web page system. Both methods of evaluation show all the perspectives described above. The evaluation experiment in this chapter was designed to test the quality and the power of our Web-based CALL system with different methods of input and different methods of feedback, to teach and consolidate the learning. The target group in this study is learners of Arabic in the beginners, GCSE or GCE level stages or equivalent of starters, intermediate and advanced level.

7.3.1 Questionnaires

During our analysis, we found out that using a questionnaire seems to be the most effective and the simplest method for both the learner to complete and for us to create and collect data (Knight and Yourke, 2003; Ebel and Frisbie, 1991). However, this apparent simplicity contradicts the fact that creating and organizing the questions can be
Chapter 7: Evaluation

quite laborious and time consuming. Clarity about the areas and concerns that the questionnaire is to cover is essential. We had to avoid attempts to ask everything, as this would tend to make the questionnaire too long and by doing so decrease the willingness of the respondent to answer fully and thoughtfully.

In the questionnaire the items that are analysed cover three categories: the system’s ease of use, the power of the interface, and the features of the questions. In most of the questionnaire, rating questions were used rather than open/closed questions as the scoring system was much more accurate with levels from very poor to very useful. After operating the aRaBCALL system, the subjects were asked to fill in the online questionnaire and to submit it. In part C of the questionnaire, the subjects were asked to rate thirty items on a five-level scale. These thirty items were related to the sixth exercise and each exercise had five questions in order to obtain the final grade of information about what they thought. Part D allows the student to make general comments.

7.3.1.1 Designing a Questionnaire

As mentioned in section 7.2, for designing the layout of the questionnaire, it was important to leave enough space for responses to open-ended questions but at the same time keep such questions to a minimum. Often, these areas are better left to semi-structured interviews where a fuller answer can be sought. For evaluating the aRaBCALL questionnaire, short-answer questions were kept to allow the respondent some space to elaborate on their response. It was important to keep the questionnaires anonymous as this enables a freer response than if the learners can be identified, in which case they might not be as forthcoming. Even though questionnaires are less time consuming to administer than other instruments of evaluations, it is essential not to over use them. Learners can become uncooperative if every time they do something they are presented with a questionnaire. The following are some advantages and disadvantages of using a questionnaire:
Chapter 7: Evaluation

- **Advantages**
  1. Simple to administer.
  2. Easy to collate and analyse.
  3. Covers a lot of information in a simple format.
  4. Allows for anonymity, perhaps eliciting more genuine responses.

- **Disadvantages**
  1. Open-ended questions are more difficult to categorise.
  2. Necessary to keep the questions focused and to ask the right kind.
  3. Difficulty in getting a useful response rate (not everybody returns the questionnaire).
  4. Any follow-up must be done through interviews.
  5. Responses may be based on the more memorable events (either good or bad).

The purpose of the questionnaires is to evaluate the whole aRaBCALL system to develop it in a way that supports the project. Some questions relate to how easy the system is to use while others relate to the power of the interface. A number of the questions have enough space to enter text freely, which can be filled in or left blank. The questionnaire was split into four parts. Part A focuses on ease of use, Part B on features of exercises, Part C on the details of the question topics, and part D on general information.

**7.3.1.2 Result of Questionnaire**

As the system exists on a website, [http://ramapp.cs.man.ac.uk:8080/CALL/login.jsp](http://ramapp.cs.man.ac.uk:8080/CALL/login.jsp), it was easily accessed by our learners. There are around seventy-five students that had used the system from all levels of Arabic. In this section, we will focus on analysing the evaluation procedure for the questionnaire. The rating of the questionnaire was analysed on three factors: the effect of the ‘ease of use’, the ‘power of the system feature’, and the detailed description of each exercise question.
From the log, it could be seen that around 75 students had accessed the system. However, only 33 questionnaires were returned, which is the equivalent of 44%. The reason for the low amount of questionnaires could have been due to the distribution of the questionnaire at the end and it being regarded as a daunting task by some students. As was already the case the previous year, all students said they enjoyed the system as well as being able to learn and practice Arabic anytime and anywhere. They also felt quite confident when accessing the aRaBCALL system.

In section C of the questionnaire, the efficacy of the help files and how much assistance they provided when fixing the errors to reach the correct answer were investigated. It was explained that the system kept a running score dependent on the use of help files and sentences that were answered correctly. When asking whether or not this affected the use of the help file, a majority, 79%, answered yes and 21% answered no. This could be due to the students being penalised when having used the help file and concentrated more on achieving a high score rather than seeking help and learning from their mistakes. Furthermore, table 7.1 below confirms the answer to this question with the help of the log file. Furthermore, when investigating the next step of the user when having made a mistake, the majority of users aimed to fix it immediately. This was asked in order to try and work out whether or not the system fulfilled the needs of the student. As a majority of students tried to fix it immediately then clearly the system is not doing very much for them, as they are not taking advantage of the help provided. It seems that they are relying more on their second guess rather than trying to understand the reason behind their mistake. Table 7.1 displays the number of students that agree with the specific options.
<table>
<thead>
<tr>
<th>Option</th>
<th>No of students</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Try to fix it immediately</td>
<td>17</td>
<td>52%</td>
</tr>
<tr>
<td>Go to the summary information and try to fix it</td>
<td>5</td>
<td>15%</td>
</tr>
<tr>
<td>Go to help feedback file and then fix it</td>
<td>3</td>
<td>9%</td>
</tr>
<tr>
<td>Go on to the next sentence</td>
<td>7</td>
<td>21%</td>
</tr>
</tbody>
</table>

Table 7.1: Student’s score for the question about making mistakes

Moreover, in the questionnaire we asked the students to rank the various exercises as to whether they met their aim or not in terms of using the tool. Surprisingly, although there are 720 possible orders, the students who gave this information only made 7 rank orders. This made it fairly easy to interpret the ranks, since the patterns can be easily seen when there are only 7 orders to look at. If we look at the rank order that they used, we could see different views on verb and adjective topic order, which is represented by the verb and adjectives exercises. Some learners think they are both useful, whilst others do not. It is likely to find verbs and adjectives topics being useful for beginners that are studying verbs and adjectives but not on the complementisers, which beginners have not been introduced to. Although the detailed feedback page does describe how complementisers work, it is not, and was never intended as, a complete introduction to the topic. The second group look like the advanced learners because presumably they already know or have idea about this topic (we had told the students that we would not be using the data as part of their assessment, and to reinforce this message we anonymised the data, and hence cannot trace which students belong to which groups). If we split them into two groups we can then calculate an arranged rank for each exercise for each group. The rest of the exercises, which are topics for grammar such as pronouns, negation and weak verbs, were found to require the same level of learning so they were mixed in with both the groups. Table 7.2 below represents a detailed group point of view.
### Table 7.2: A detailed group point of view on subjective assessments

We calculate the average rank for each topic by taking the summation of multiplying the number of students who select the topic in the group by their rank and divide by the total number of the participating students who are involved in this questionnaire. Now consider the following measurement for the rank of each exercise regarding all groups:

\[
\text{Average rank} = \frac{\sum_{i=1}^{n} \text{rank}_i}{|\text{participant}|}
\]

As far as our task is concerned, \( n \) is the number of exercises. Consequently, the table below confirms the above argument that the exercise on topic verbs is the exercise that most meets the criteria of student aim with a value of 2.06 as shown below.
Chapter 7: Evaluation

Table 7.3: A representation of each exercise rank

<table>
<thead>
<tr>
<th>Verbs</th>
<th>Pronoun</th>
<th>Adjective</th>
<th>negation</th>
<th>Complementiser</th>
<th>weak verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.06</td>
<td>3.00</td>
<td>3.39</td>
<td>3.15</td>
<td>4.70</td>
<td>5.00</td>
</tr>
</tbody>
</table>

We also collect ranking with individual sentences, as shown in figure 7.1. From the results in figure 7.1, we can see that most of the learners indicated that the exercises that focused on the use of verb tenses or agreement are very useful. From our point of view, these users that benefit from this exercise could be from the beginners or intermediate level of learning Arabic.

**Figure 7.1: Scores for verb exercise**

From all the various kinds of exercises we illustrate that there are different levels of students who benefit from these exercises as a learning aid (see appendix D). This could be because of a few reasons. One reason is the difficulty of understanding some special verbs, such as weak verbs, when there is no background study on the Arabic literature, especially for learners who are not native speakers and are from the beginner’s level as
they do not study this subject of grammar. Another reason is due to Arabic having a rich morphology and a complex syntax, especially when using weak verbs and complementisers.

The significant differences that were found between the different groups’ points of views were evident from the questionnaire. Therefore, the students from the beginner’s level would not be familiar with the grammar topic, weak verbs for instance. However, Arabic is a free word order language and there are crucial rules to change the tenses of the verb, this makes this exercise useful to all levels of learners.

7.3.2 Interviews and Comments

The learners were asked to give reasons for their responses to the questionnaire items during the interviews. These interviews were conducted with the students to gain extra support for our evaluation of the aRaBCALL system, and insight into their decision-making process and to develop an understanding of their willingness and ability to access and manipulate the aRaBCALL exercises.

The aRaBCALL system was sent to all our learners who are in beginners, intermediate and advanced level. There were only seventy-five students who accessed the system and it seemed that from the questionnaire, log file and the interviews, they had a good learning period during their work with aRaBCALL.

The interviews were done individually with around twenty students⁶, most of them from beginner’s level, a couple from intermediate and three from advanced level of learning. Each interview also provided an opportunity to ask students about their individual contributions and any changes they might have overlooked.

Generally, the students’ opinions concerning the ease of use and the feedback of the system were especially requested. The students were also asked about the concept of self-correction and suggestions for improvement of the system.

---

⁶ The reason behind that we could not manage to interview all the learners because the students who are in Advanced level (A level) or intermediate (GCSE level) GCSE have left the college since June 2011, however we did manage to contact some and meet them.
For the ease of use of the system, almost all of the subjects answered that aRaBCALL was designed in a way that was interactive and interesting, because of being able to input their own sentences freely and easily change words to make new sentences. Besides that, almost all the students agreed that the feedback is very informative in relation to the typed sentence in the feedback of the aRaBCALL system. Our main goal for the interview is to ascertain which exercises were the most and least difficult, and to know exactly the reason why they are difficult for them. Most answers are similar, that they did not give their patience to this exercise as it is not in their syllabus and they think would not be useful to them (see appendix D).

We collected data when we interviewed some learners. Most of the information received leaned positively to its easy-to-use nature. Some students asked for extensions such as translation and speech, which would require very substantial further research, and we simply noted these interests for future work.

Generally speaking, all the students say this system is very helpful and brings technology to learning Arabic for all levels of the English syllabus for the first time.

7.3.3 User Behaviour

In this section, we will analyse the result through the log file. The log file represents and clears up who has benefited from the system the most and who has hardly benefited from it. The log file contains a record of what the student has done which provides us with objective information that can be used to supplement the subjective evaluation that we obtained from the questionnaires and interviews.

7.3.3.1 Analysis of Log File

The log file represents all the users’ work and behaviour. As explained in section 6.3.2, the log file is automatically created on the Internet and it begins to record information as soon as the student first enters http://ramapp.cs.man.ac.uk:8080/CALL/login.jsp. We can see all the student’s interactions with the system resulting in lots of information being collected, which relates to the possible pattern of behaviour. Figure 7.2 shows the
transition diagrams that represent the patterns of the user interaction when accessing the system. The table 7.4 below shows the property that is represented by each activity title.

<table>
<thead>
<tr>
<th>Activity title</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>next</td>
<td>go to next exercise</td>
</tr>
<tr>
<td>prev</td>
<td>go to the previous exercise</td>
</tr>
<tr>
<td>success</td>
<td>enter correct text</td>
</tr>
<tr>
<td>skip</td>
<td>go to some other exercise</td>
</tr>
<tr>
<td>failure</td>
<td>enter incorrect text</td>
</tr>
<tr>
<td>visit 1</td>
<td>look at level 2 feedback</td>
</tr>
<tr>
<td>visit 2</td>
<td>look at level 3 feedback</td>
</tr>
</tbody>
</table>

Table 7.4: The representation of the activities’ path

There are general observations that can be made about the user path through all the expected procedures. Figure 7.2 represents all the possible transitions. Looking at the active moves made by an individual student will provide us with information about how they have used the tool.

Figure 7.2: Network for all legal options included equally
From Figure 7.2 and table 7.5 above, we can see much different behaviour in the way the students use the system. For instance, a student who always moves on to the next exercise, regardless of whether they have got the current one right or not, does not get good use of the system. Such behaviour will be characterised as the sequence of ‘failure’ ‘next’ as the aim of the system is to allow students to build and practise on what they already know.

Our explanation was that a student who looks at the summary description after failure to answer it correctly and then reads up on the detailed description after looking at the summary, consequently fixing the problem successfully, will get the most benefit from the system. Figure 7.3 shows the set of transitions for a student who got most things right, and who used the feedback appropriately when they made a mistake\(^7\), hence the system fulfils the student’s needs.

\[(7.2)\]

<table>
<thead>
<tr>
<th>Event Sequence</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>failure-&gt;failure</td>
<td>0.24 (4)</td>
</tr>
<tr>
<td>failure-&gt;next</td>
<td>0.06 (1)</td>
</tr>
<tr>
<td>failure-&gt;visit1</td>
<td>0.06 (1)</td>
</tr>
<tr>
<td>next-&gt;success</td>
<td>0.24 (4)</td>
</tr>
<tr>
<td>success-&gt;failure</td>
<td>0.12 (2)</td>
</tr>
<tr>
<td>success-&gt;next</td>
<td>0.18 (3)</td>
</tr>
<tr>
<td>visit1-&gt;visit2</td>
<td>0.06 (1)</td>
</tr>
<tr>
<td>visit2-&gt;success</td>
<td>0.06 (1)</td>
</tr>
</tbody>
</table>

The code 7.2, shown diagrammatically in figure 7.3, represents the behaviour of a student who gets most things right first time, and uses the feedback files appropriately when they do make a mistake.

\(^7\) In the diagrammatic presentations, the thickness of a line corresponds to the frequency of taking that transition.
However, figure 7.4 below represents the behaviour of a student who fails to make good use of the system. We can see that when the student enters incorrect text and moves on to the next exercise without correcting their errors, it means that the student has not used the system the way it should be used to maximise their learning. A student who repeatedly moves from failure to next or from failure to prev is not attempting to correct their mistakes, and as such is not using the system very effectively.

Now, we will analyse some paths individually regarding the student’s benefit, which can all be summarized as follows (See Appendix E for all the users’ transitions):

(i) **An Ideal user** of the aRaBCALL tool was shown in figure 7.5. The intended use of aRaBCALL is when the student first accesses the system and they make a mistake, they
read the feedback or even the detailed explanation of the feedback in order to correct their mistake. For example, we saw that with beginner users who do not know about complementisers, they attempt the exercise and then read the feedback in order to try and learn from it. However, at the second visit they accessed the system without viewing the feedback. In this case, we found that the user completes their task and moves on to the next. From the analysis of the log file, we found that a good user of the system who has visited the system more than once will reduce the level of feedback each time. The first time they visit, they move from level one feedback, to level three, through level two. It was not easy to measure the effectiveness of such paths and continue on investigating to get a definite decision.

![Figure 7.5: The pattern of the ideal user of the system](image)

(ii) **A Sensible use** of the tool or effective learning for another group of users who use the system very effectively is when they move from incorrect text to correct text, and then go to the next text as they input a new sentence as represented in figure 7.6. In addition, the user who also tries several times to find out the correct answer uses the system effectively. However, these groups of learners do not necessarily benefit fully from the tools as the other groups of users described in part (i) would, since although they do generally correct their mistakes they do not make any use of the feedback. As noted earlier, this may be because our penalties discouraged them from doing so.
(iii) A poor use of the system, as figure 7.7 below represents, is a path taken in the system without seeking advice from the learning tool. This path shows a bad level of knowledge in students who in our point of view did not benefit from the system by learning or by consolidating learning. Figure 7.7 below shows a learner that has not put any effort in to learn or to use the learning procedure to their benefit. From this, we can see that the user may not have been responsible enough to try and learn, as the user has moved several times between next and success without actually coming to a result. It also seems that the learner has entered text with lots of mistakes.
7.4 Conclusion

In this chapter, we have talked about the evaluation of the system from both perceptions, subjective, which is represented by the questionnaire, and objective, which is represented by the log file. The analysis of how the student uses the system provided an interesting piece of evaluation. The questionnaire is essentially subjective as it is from the students’ perspective. We apply a rank measurement for each exercise and then we discuss the average result rank. Finally, we analyse all the expected paths of user interaction, which may end up as an effective learning tool, or not. From the results, it has been found that there was no significant difference in the effect of the evaluation method that was used. It was confirmed that the rating of each exercise did not depend only on the interviews, questionnaires or the log file; but the decision-making came from all the previous aspects.
Chapter 8

Conclusion

In this chapter, we will review the preceding chapters, discuss the contribution of the research and finally discuss any further research in the future.

8.1 Thesis Summary

In the introduction of this thesis, we indicated that the final goal of this project was to build an ICALL system that would provide computational assistance to learners of the Arabic language. At the outset of this research we posed two major hypotheses:

- That providing students with multi-level guidance about grammatical errors would be useful as a support to independent learning
- That relaxing constraints in a constraint-based grammar would enable us to diagnose a wide variety of errors

From our research, we are able to come to the conclusion that we can accept both hypotheses. The first hypothesis can be proved as a teacher may not provide the indepth guidance required for an individual to understand a certain topic, whereas easy accessible multi-level guidance can help to support the student and provide the exact amount of assistance required by the student. The second hypothesis can be accepted as we found out from the research in chapter five, that relaxing a constraint in a constraint-based grammar causes the system to accept and then deal with all expected errors. In particular, we have shown that constraint relaxation will let us deal with word internal problems such as the use of regular affixes with irregular words, and with complex
constraints on words such as those imposed by the various complementisers. These two hypotheses led to the research questions in section 1.4, repeated here:

i. What types of errors can be diagnosed by softening the constraints in the rule-based description of Arabic morphosyntax?

ii. What types of feedback do students’ find useful?

iii. Can student behaviour be characterised in ways that will allow guidance to be provided?

We aimed to provide a system to teach grammar and provide assistance by producing homework and issuing students with immediate feedback. The feedback process was considered limited in its effectiveness because, despite evidence of students’ desire for feedback (Ben Mohamed et al., 2011; Price et al., 2010; Rust 2002), students do not necessarily read their feedback (Hounsell, 1987) or, if they do, they may not understand or use it (Gibbs and Simpson, 2004; Lea and Street, 1998; McCune, 2004). Several levels of feedback were provided, and we tracked student behaviour and how the interface was used. Most of the work was based on various situations such as our teaching experience, investigation of the British curriculum and observing students’ needs. We found out that an important field to be confident in when learning a second language is grammar and in specific the rules of grammar as it is the core of the language, such as adjectives and verbs and how their form changes with regard to gender, person, tense, negation article and its rule, pronouns, complementiser and weak verbs. The use of a log file in the system helps to express a student’s behaviour when accessing the system, which helps with diagnosing areas where they have particular difficulty. This can be used to direct them to further study and exercises in those specific areas. However, this is not something new to us as any system which detects and diagnoses errors can provide a log which can be used in this way. The main benefit of our system is the way in which it records each action a student takes, which is characterised in the activity graphs in chapter 7. These activity graphs will let us distinguish between students who are working systematically through the exercises, and making use of the detailed guidance that is provided by the third level help files and
students who are just rushing through the exercises without reflecting on what they have
got right or wrong. We can use this information to provide guidance about how to study
as well as the simpler guidance about what to study that comes from just classifying
their errors.

Parasite (Ramsay and Mansour, 2001) was used in our thesis due to its constraint-
based approach and due to it previously being used to analyse numerous languages such
as Arabic. Parasite produces morphological, syntactic and semantic analysis of textual
input and is employed to develop the system-generated error detection and diagnosis.
The methodology we adopted was by relaxing certain constraints on unification; correct
information contained in a badly formed sentence may still be used to obtain a coherent
overall analysis. Thus, the system was able to produce a model of a well-formed version
of the user's intended input. A number of examples were also given where
morphological errors had been soft-parsed. In particular, errors relating to verbs that
required minor modification to the regular formation of rules and adjectival agreement
were examined, as well as non-standard number suffixes.

We looked into the research of the background on errors and showed that the term
‘grammatical errors’ in the field of language teaching is used to refer to morphological
and syntactic errors. We have reviewed various classifications of error and diagnosed
semantic errors introduced by Tudoka and Chen (2004), James (1998), Dulay et al.
(1982), Taylor (1998) and looked into the errors from our experience of aRaBCALL.
From our teaching experience we also understood that students’ errors were mostly in
the field of grammar. We divided our learners’ errors into three categories, word
formation errors, which occur when adding wrong clitics or a wrong case marker to a
word; local feature mismatch, which occurs when the student misuses the definitive or
agreement marker; and finally word order errors, which arise because Arabic permits a
variety of word orders, but imposes subtle constraints on their use in ways which have
no counterpart in English. We also examined how to deal with learner’s errors. It was
noted that the aRaBCALL system was able to recognize correct input, and was also able
to make use of the information contained in badly formed sentences.
We explained how the type of feedback we used when helping students was a three level feedback system, which was in fact a sequence of feedback provided. The first level of the sequence, a word message, was provided when the system highlights the word that had caused the sentence to be incorrect. The second level, a short error message, was when the student could not manage to correct the previous error to obtain a correct sentence. If the student required more feedback after the second stage, a detailed feedback explanation was displayed in order to assist the student, which was the final stage of the three level feedback system. The reason this feedback system was used was due to the benefit of allowing the students to learn why they were incorrect in stages, allowing them to try and correct themselves. Using a three level feedback system could not only distinguish the levels at which the students were at but also provide a more detailed explanation of the correction to improve their understanding and most importantly their learning. Due to all these reasons, the three levels of feedback that we used with our learners are the steps that the system can offer with a reflection or an evaluation of the learners’ work. Before we viewed the evaluation of the system, we examined how we investigated the different ways of assessments using a survey, in order to find out the effective way of assessing our system aRaBCALL.

We concluded that software evaluation remained an important area of CALL and that there are indications of its role increasing. In the evaluation of our system, we discussed some reports on the research of evaluation within the CALICO journal and understood that Hubbard (2006) had adapted three approaches to evaluation: checklists, methodological frameworks and SLA approach. Furthermore, we had an extra approach to the evaluation of our system through the analysis of user behaviour from the log file.

We evaluated the system from both perceptions, subjective, which was represented by the questionnaire, and objective, which was by analysing the log file. We also had a look at how we analysed how the student used the system. We explored how we incorporated a rank measurement for each exercise and discussed the average result rank. Finally, we analysed all the expected paths of user interaction. From the results, we found that there was no significant difference in the effect of the evaluation method that was used. It was confirmed that the rating of each exercise did not depend only on the interviews, questionnaires or the log file. However, the decision-making came from all
the previous aspects. The log file helps to portray to us the typical analysis of the user and their action regarding the three-level feedback system.

8.2 Future research

This thesis deals with morpho-syntactactic errors. We have done a wide range of soft parsing to handle the expected errors that the learner of Arabic might make. However, we have not handled the detailed analysis of user behaviour due to limited time.

We have the following plans for future study:

- Do a larger scale study on a large cohort to examine how the system effects the development of the student’s learning over a longer period
- Involve teachers to get feedback about their view of the tool, how they would use it in classroom situations, what kind of exercises they would get
- See whether there is a link between patterns of student behaviour from the log file and attainment as further analysis can be made on the log file

In conclusion, as noted by Hart (1995), trying to extract statistically reliable measures of the effect of using CALL tools is a long-term process (ibid). Ideally, we would like to study the effectiveness of the aRaBCALL by conducting a substantial longitudinal study, but a long-term analysis of this kind is beyond the scope of this thesis.
Appendix A: Soft Constraints

/**** CONSTRAINTS ON TENSE MARKERS ****/

notBlocked(X0) :-
    [person, remarks]@X0
    <- [person, remarks]@X1,
    person@X0 <- P,
    softParse(ground(P),
        \+ blockedAgreement(P),
        blocked(index@X0),
        X1).

constraintsOnYA(_AGRTEXT, X) :-
    X <> [pres_or_future, third].

constraintsOnNA(_AGRTEXT, X) :-
    X <> [pres_or_future, first, not_singular].

constraintsOnTA(_AGRTEXT, X) :-
    X <> [pres_or_future, not_first, singular].

constraintsOnO(_AGRTEXT, X) :-
    X <> [pres_or_future, first_sing_only].

constraintsOnEmptyTense(AGRTEXT, subject@X, agree@X) :-
    (AGRTEXT == '' ->
        X <> [third_sing_only, masculine];
    AGRTEXT == t ->
        trigger(index@subject@X, (X <> first_sing_only; X <> second_sing_only; X <>
            [third_sing_only, feminine]));
    AGRTEXT == n ->
        X <> [first, not_plural];
    true).

/**** CONSTRAINTS ON AGREEMENT MARKERS ***********************/

constraintsOnEmptyAgr(_N, Y, _TEXT, _UNDERLYING, [DET], _IRREAL, colour) :-
    affix@DET <-> *det,
    syntax@Y <-> syntax@DET,
    Y <- adj1.

constraintsOnEmptyAgr(+, Y, TEXT, TEXT, AFFIXES, _IRREAL, LEXTYPE) :-
    affix@DET <-> *det,
    syntax@Y <-> syntax@DET,
Appendix A: Soft Constraints

Y <> suffix,
notBlocked(Y),
!,
doSoftParse((singular(Y),
(LEXTYPE = pronoun ->
(AFFIXES = [], masculine(Y)));
(LEXTYPE = noun(pname) ->
AFFIXES = [];
(LEXTYPE = noun(_),
AFFIXES = [DET],
LEXTYPE = lextyp@DET,
third_sing_only(Y)))),
agr(index@Y, text@Y),
Y).

constraintsOnEmptyAgr(N, Y, TEXT, TEXT, AFFIXES, _IRREAL, LEXTYPE) :-
verb(Y),
notBlocked(Y),
doSoftParse(third_sing_only(Y), agr(index@Y, ''), Y),
AFFIXES = [],
(past(Y) ->
(masculine(Y), TEXT = a);
trigger(irreal@Z,
((indicative(Y); interrogative1(Y)) ->
TEXT = u;
subjunctive(Y) ->
TEXT = a;
jussive(Y) ->
TEXT = o)));
affix@current:fst@fstOut@Y <- *(_).

constraintsOnEmptyGender(X0, ANYNOUN) :-
agree@X0 <- agree@X1,
affixes@X0 <- [NUM],
[lextyp, syntax]@X0 <- [lextyp, syntax]@NUM,
affix@NUM <- *NEXT,
trigger(ANYNOUN,
(ANYNOUN = broken(BROKEN) ->
(doSoftParse(BROKEN = '', lextyp(index@X0, broken('')),
ANKYOUNG, X0),
third_plural_only(X1), NEXT=det);
ANYNOUN = solid ->
NEXT = det;
(NEXT = agr,
softParse(ANYNOUN, (masculine(X1) -> true;
third_plural_only(X1)), gender(index@X0, text@X0, masculine), X0))).

constraintsOnNA(X0) :-
[agree, vfeatures, index, remarks]@X0
<- [agree, vfeatures, index, remarks]@X1,
softParse(lextyp@X0,
(first(X1), not_singular(X1)),
agr(index@X0, text@X0),
X1),
notBlocked(X0),

219
softParse(lextype@X0,
past(X1),
tense(index@X0, text@X0),
X1).

actualConstraintsOnT(X) :-
X <> [masculine, second_sing_only],
\+ blockedAgreement(X),
characters("ta", underlying@X).

actualConstraintsOnT(X) :-
X <> [feminine, second_sing_only],
\+ blockedAgreement(X),
characters("ti", underlying@X).

actualConstraintsOnT(X) :-
X <> first_sing_only,
\+ blockedAgreement(X),
characters("tu", underlying@X).

actualConstraintsOnT(X) :-
X <> [feminine, third_sing_only],
\+ blockedAgreement(X),
characters("at", underlying@X).

constraintsOnT(X0) :-
[agree, vfeatures, index, remarks]@X0
<- [agree, vfeatures, index, remarks]@X1,
notBlocked(X0),
softParse(ground(gender@X0),
(pretty(xxx(agree@subject@X0+gender@X0+index@X0+index@subject@X0+agree@X0)), actualConstraintsOnT(X1)),
subjAgree(index@X0, index@subject@X0), X1).

constraintsOnTA1(X0) :-
[agree, vfeatures, index, remarks]@X0
<- [agree, vfeatures, index, remarks]@X1,
notBlocked(X0),
softParse(lextype@X0, third_dual_only(X1), agr(index@X0, text@X0),
X1),
shortGender(X1, G),
softParse(lextype@X0, feminine(X1), gender(index@X0, text@X0, G),
X1).

constraintsOnTN(X0) :-
[agree, vfeatures, index, remarks]@X0
<- [agree, vfeatures, index, remarks]@X1,
notBlocked(X0),
softParse(lextype@X0, second_plural(X1), agr(index@X0, text@X0),
X1),
shortGender(X1, G),
softParse(lextype@X0, feminine(X1), gender(index@X0, text@X0, G),
X1),
softParse(lextype@X0, past(X1), tense(index@X0, text@X0), X1).
Appendix A: Soft Constraints

constraintsOnTMA(X0) :-
    [agree, vfeatures, index, remarks]@X0
    <-> [agree, vfeatures, index, remarks]@X1,
    notBlocked(X0),
    softParse(lextype@X0, second_dual_only(X1), agr(index@X0, text@X0), X1),
    softParse(lextype@X0, past(X1), tense(index@X0, text@X0), X1).

constraintsOnTM(X0) :-
    [agree, vfeatures, index, remarks]@X0
    <-> [agree, vfeatures, index, remarks]@X1,
    notBlocked(X0),
    softParse(lextype@X0, second_plural_only(X1), agr(index@X0, text@X0), X1),
    shortGender(X1, G),
    softParse(lextype@X0, masculine(X1), gender(index@X0, text@X0, G), X1),
    softParse(lextype@X0, past(X1), tense(index@X0, text@X0), X1).

constraintsOnUWA(X0) :-
    [agree, vfeatures, index, remarks]@X0
    <-> [agree, vfeatures, index, remarks]@X1,
    notBlocked(X0),
    shortGender(X1, G),
    trigger(lextype@X0, (pres_or_future(X1) ->
        doSoftParse((dual(X1), not_first(X1)), agr(index@X0, text@X0), X1); 
        doSoftParse(masculine(X1), gender(index@X0, text@X0, G), X1),
        doSoftParse(third_dual_only(X1), agr(index@X0, text@X0), X1)));).

constraintsOnAA(X0) :-
    [agree, vfeatures, index, remarks]@X0
    <-> [agree, vfeatures, index, remarks]@X1,
    notBlocked(X0),
    shortGender(X1, G),
    affinees@X0 <-> [],
    notBlocked(X0),
    softParse(lextype@X0, pres_or_future(X1), tense(index@X0, text@X0), X1),
    softParse(index@subject@X0, (third_dual_only(X1); (second_dual_only(X1))), agr(index@X0, text@X0), X1).
Appendix A: Soft Constraints

+n:xbar@cat@X0,
lextype@X0 <- LEXTYPE,
    softParse(LEXTYPE, \+ LEXTYPE = colour, lexttype(index@X0,
        LEXTYPE, regular(_)), X0),
[agree, index, remarks]@X0
<- [agree, index, remarks]@X1,
affixes@X0 <- [DET],
affix@DET <- *det,
[syntax, lexttype]@X0 <- [syntax, lexttype]@DET,
notBlocked(X0),
    softParse(lextype@X0, third_dual_only(X1), agr(index@X0,
        text@X0), X1).

constraintsOnN(X0) :-
    X0 <- [verb, suffix, checkSuffixIsNeeded, immediate],
        [agree, vfeatures, index, remarks]@X0
<- [agree, vfeatures, index, remarks]@X1,
affixes@X0 <- [],
notBlocked(X0),
shortGender(X1, G),
    softParse(lextype@X0, feminine(X), gender(index@X0, text@X0,
        G), X1),
    softParse(lextype@X0, (plural(X), not_first(X)), agr(index@X0,
        text@X0), X1).

constraintsOnWA(X0) :-
    X0 <- [verb, suffix, checkSuffixIsNeeded, immediate],
        [agree, vfeatures, index, remarks]@X0
<- [agree, vfeatures, index, remarks]@X1,
affixes@X0 <- [],
notBlocked(X0),
    softParse(lextype@X0, pres_or_future(X1), tense(index@X0,
        text@X0), X1),
shortGender(X1, G),
    softParse(lextype@X0, masculine(X), gender(index@X0, text@X0,
        G), X1),
    softParse(lextype@X0, (plural(X), not_first(X)), agr(index@X0,
        text@X0), X1).

constraintsOnAT(X0) :-
    notBlocked(X0),
shortGender(X1, G),
lextype@X0 <- LEXTYPE,
    softParse(LEXTYPE,
        (LEXTYPE = noun(regular('N/Ap')); feminine(X0)),
        gender(index@X0, text@X0, G),
        X0),
    softParse(LEXTYPE, third_plural_only(X0), agr(index@X0, text@X0),
        X0),
    localConstraintsOnAT(X0).

localConstraintsOnAT(X0) :-
    X0 <- n,
[syntax, lexttype]@X0 <- [syntax, lexttype]@DET,
lextype@X0 <- noun(LEXTYPE),

softParse(LEXTYPE, LEXTYPE=regular(_), lextype(index@X0, LEXTYPE, regular(_)), X0),
affix@X0 <-- *gender,
affixes@X0 <-- [DET],
affix@DET <-- *det.

constraintsOnYN(X0) :-
+ n:xbar@cat@X0,
[agree, index, remarks]@X0 <- [agree, index, remarks]@X1,
lextype@X0 <-- noun(LEXTYPE),
affix@X0 <-- *agr,
affixes@X0 <-- [DET],
affix@DET <-- *det,
[syntax, lextype]@X0 <-- [syntax, lextype]@DET,
shortGender(X1, G),
softParse(LEXTYPE, LEXTYPE=regular(''), lextype(index@X0, LEXTYPE, regular('')), X1),
shortGender(X1, G),
softParse(lextype@X0, masculine(X1), gender(index@X0, text@X0, G), X1),
softParse(LEXTYPE, third_plural_only(X1), agr(index@X0, text@X), X1),
X0 <> [suffix, checkSuffixIsNeeded, immediate, acc].

constraintsOnYN(X0) :-
[agree, vfeatures, index, remarks]@X0 <- [agree, vfeatures, index, remarks]@X1,
affixes@X0 <-- [],
X0 <> [verb, suffix, checkSuffixIsNeeded, immediate, indicative],
notBlocked(X0),
shortGender(X1, G),
softParse(lextype@X0, pres_or_future(X1), tense(index@X0, text@X0), X1),
shortGender(X1, G),
softParse(lextype@X0, feminine(X1), gender(index@X0, text@X0, G), X1),
softParse(lextype@X0, second_sing_only(X1), agr(index@X0, text@X0), X1).

constraintsOnWN(X0) :-
+ n:xbar@cat@X0,
[agree, index, remarks]@X0 <- [agree, index, remarks]@X1,
lextype@X0 <-- noun(LEXTYPE),
affix@X0 <-- *agr,
affixes@X0 <-- [DET],
affix@DET <-- *det,
[syntax, lextype]@X0 <-- [syntax, lextype]@DET,
shortGender(X1, G),
softParse(LEXTYPE, LEXTYPE=regular(''), lextype(index@X0, LEXTYPE, regular('')), X1),
shortGender(X1, G),
softParse(lextype@X0, masculine(X1), gender(index@X0, text@X0, G), X1),
softParse(LEXTYPE, third_plural_only(X1), agr(index@X0, text@X0), X1),
Appendix A: Soft Constraints

X0 <> [suffix, checkSuffixIsNeeded, immediate, nom].

costraintsOnWN(X0) :-
    [agree, vfeatures, index, remarks]@X0
    <> [agree, vfeatures, index, remarks]@X1,
    affix@X0 <> *agr,
    affixes@X0 <> [],
    X0 <> [verb, suffix, checkSuffixIsNeeded, immediate, indicative],
    notBlocked(X0),
    softParse(lextype@X0, pres_or_future(X1), tense(index@X0, text@X0), X1),
    softParse(lextype@X0, (not_first(X0), plural(X1)), agr(index@X0, text@X0), X1),
    shortGender(X1, G),
    softParse(lextype@X0, masculine(X1), gender(index@X0, text@X0, G), X1).

Base entries files
/***** MISCELLANY *****/

% this one takes an object clitic and then a verb initial sentence
% of the required kind

"lky" $$ X lextype complementiser :-
    subject@Y <> nom,
    Y <> [s, tensed_form, genuine],
    text@X <> (likay),
    softParse(index@Y, (present(Y), declarative1(Y), subjunctive(Y)), compTense(index@X, index@core@Y), X),
    end@X <> start@core@Y,
    arabicComplementiser(X, Y),
    X <> adjunct,
    target@X <> s.

"lm" $$ X lextype complementiser :-
% This is where it says it's verb initial: the core of a clause
% is the verb itself
    Y <> [s, tensed_form, genuine, declarative1],
    subject@Y <> SUBJECT,
    softParse(index@SUBJECT, reallyNom(SUBJECT), subjcase(index@X), X),
    start@core@Y <> STARTY,
    end@X <> ENDX,
    softParse(STARTY, ENDX=STARTY, subjPosition(index@X, index@core@Y, STARTY, ENDX), X),
    softParse(index@Y, jussive(Y), mood(index@Y), X),
    softParse(index@Y, present(Y), compTense(index@X, index@core@Y), X),
    arabicComplementiser(X, Y),
    X <> adjunct,
    target@X <> s.

"ln" $$ X lextype complementiser :-
    Y <> [s, tensed_form, genuine, declarative1],
subject@Y <-> SUBJECT,
softParse(index@SUBJECT, reallyNom(SUBJECT), subjcase(index@X), X),
start@core@Y <-> STARTY,
end@X <-> ENDX,
softParse(STARTY, ENDX=STARTY, subjPosition(index@X, index@core@Y, STARTY, ENDX), X),
softParse(index@Y, indicative(Y), mood(index@core@Y), X),
softParse(index@Y, present(Y), compTense(index@X, index@core@Y), X),
arabicComplementiser(X, Y),
X <> adjunct,
target@X <> s.

"An" $$ X lextype complementiser :-
X <> initial(+),
X <> terminal(+),
Y <> [s, genuine, declarative],
arabicComplementiser(X, Y),
text@X <-> (COMP),
comp@X <-> *COMP,
subject@X <-> subject@Y,
trigger(COMP, '?anOr?anna'(X, Y)).

"In" $$ X lextype complementiser :-
Y <> [s, genuine, declarative, indicative],
subject@Y <> acc,
comp@X <-> *'In',
arabicComplementiser(X, Y).

"." $$ X :-
X <-> inflected,
S <-> [verb, saturated],
subject@X <-> subject@S,
cat@X <-> punct,
xstart@S <-> 0,
-interrogative@S,
-comp@S,
+before:dir@dir@S,
semantics@X <-> mood@S,
-modifier@X,
-moved:dir@displaced@S,
+compact@S,
args@X <-> [S],
theta@S <-> arg(*lambda(P, lambda(_Q, (mood@S):P))),
+main@S,
+fixed@X,
% so it can be the argument of "also"
+unspecified@X,
subject@S <-> SUBJ,
SUBJ <-> reallyNom.
Appendix B: System Output

B.1 Word order Constraints

Here let us test the system with a mistake in the noun phrase, i.e., using “a man” instead of “the man” as shown in sentence B.1 below.

(B.1) رجل يكتب كتاب

aRaBCALL displaying first level feedback on misused word order constraints

As Arabic has fairly free word order, the standard word order is verb-subject-object; the subject can be put in front of the verb in order to create particular attention to the subject. In much the same way, the object can also be put at the beginning of a sentence in English if you want to draw attention. This can generally only be done if the item you are trying to emphasize is a definite NP. Therefore, the figure B.1 above shows that putting an indefinite NP rjl in front of the verb yktb is not acceptable in Arabic. Figure B.2 below shows the detailed clarification on how such information can be used for displaying proper feedback to students.
Subject position

Arabic does have fairly free word order, so although the standard word order is verb-subject-object you can put the subject in front of the verb. You would do this to draw particular attention to the subject. In your example the subject “rajul?” was not definite, so it should not have been put in front of the verb.

Figure B.2: aRaBCALL displaying second level of feedback on misuse of word order constraints

Figure B.3: aRaBCALL displaying correct use of word order constraints

The above shows that the system responds as if the sentence was correct if the student inputs a definite NP as a subject in front of the verb.

B.2 Misused Tense

Now, this is the output of the system when the wrong tense is used, such as in the following sentence
Appendix B: System Output

(B.2) *hb Albnt ‘the girl went’

![Screenshot of aRaBCALL system]

**Figure B.4: aRaBCALL displaying first level feedback on misused tense**

One of the biggest advantages of using this system is that the system accepts the given verb, however, it complains about the tense used. In Arabic, affixes have to be added to the verb root to signify its tense and gender marker. Therefore, the verb ذهب *hb ‘to go’ is a masculine past tense singular verb and البنت Albnt ‘the girl’ is a feminine noun. As can be seen in figure B.5, the system displays feedback to the student for such occasion:

**Subject-verb agreement**

Arabic verbs have to 'agree' with their subjects--they have to have the same number and gender, in the same way that English present tense verbs have to agree with their subjects (which is why "he sleep" and "they sleeps" aren't acceptable). In your example the subject "AldAris@Ani" has different agreement marker from the verb "Oakotubuwna".

**Figure B.5: aRaBCALL displaying second level feedback on subject verb agreement**
B.3 Misused Complementiser Patterns- لا لا لا

Now let us see what happens if the student uses a completely different negation particle or a wrong verb tense with a specific negation particle. Consider the following sentence:

(B.3) لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا ولا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا ولا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا ولا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا ولا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا لا ل
The man thought that the learner went’

Figure B.7: aRaBCALL displaying first level feedback on misused ان On

The following figure B.8 shows what the system displays when the student needs further explanation as to why their answer is wrong.

Forms of the complementiser ‘An’

Some Arabic verbs expect a following subordinate clause. These clauses are often headed by the word ‘An’, in the same way that the subordinate clause ‘she saw me’ in ‘I believe that she saw me’ is headed by ‘that’.

The written form ‘An’, however, has two underlying forms, ‘An’ and ‘Anna’. You can’t tell from the written form which of these is intended, which is unfortunate because they have different properties: if the underlying form is ‘Anna’ then the following clause should be verb initial, whereas if it’s ‘An’ then the clause should be subject initial. The choice between the two forms is made by the embedding verb, in the same way that ‘know’ requires a subordinate clause headed by ‘that’ and ‘want’ requires one headed by ‘to’.

In the current case the suffix “ahaba” marks the word it is attached to as 3, which clashes with the rest of the word “4”.

Figure B.8: aRaBCALL displaying a detailed feedback on the misuse of the complementiser ان On

As you can see that some Arabic verbs expect complementiser particles. These clauses are often headed by the word ان On, in the same way that the subordinate clause
‘the learner went’ in ‘the man believes that the learner went’ is headed by ‘that’. The written form أنون On, however, has two underlying forms، أنون On and أنون Onna. You cannot tell from the written form which of these is intended, which is unfortunate because they have different properties: if the underlying form is أنون Onna then the following clause should be verb initial, whereas if it is أنون On then the clause should be subject initial. The choice between the two forms is made by the embedding verb, in the same way that ‘know’ requires a subordinate clause headed by ‘that’ and ‘want’ requires one headed by ‘to’.

B.4 Misused Weak verbs

One of the problematic areas is the use of the weak verbs, as shown in the following example:

(B.5) أنشكو أنك ‘You suffered’

Here the sentence started with a weak verb أنشكو $kw ‘to suffer’ which is neither past tense verb nor present, and أنك Ont ‘you (mas/fem)’. Therefore, the both do not match each other so the verb $kw needs a prefix يو ya or تو ta or requires changing the subject to a third person singular, such as adding the subject الولد Alwld ‘the boy’ rather than أنك Ont ‘you masculine or feminine’. Figure B.9 shown below shows if the student exchanges the pronoun أنك Ont with the word الولد Alwld ‘the boy’.
Figure B.9: aRaBCALL displaying first level feedback for the misuse of the weak verb 

يشكو $y$s$kw$

Weak nouns and verbs

When you add an affix to a word, you sometimes get sounds that are to pronounce in quick succession. When that happens, you often find that one of the sounds gets changed to something which is similar but which does easily low into the other. A good example is the English prefix 'in-', which means something like 'not'. So 'incomplete' means 'not complete', 'indecisive' means 'not decisive', and so on.

It's difficult to say 'in-' and then a word that begins with 'p'. You say 'in-' by putting your tongue between your teeth, and you say 'p' by putting your lips together: Getting from one of these to the other is tricky. So in many cases we substitute 'im-' for 'in-': they sound a bit alike, but for 'im-' you put your lips together, so it's easy to get from there to 'p'. So we say 'impossible' rather than 'inpossible' to mean 'not possible'.

This happens a lot in Arabic when one of the items involves what is known as a 'semivowel', i.e. something like 'w' or 'y' which sometimes gets used as a vowel and sometimes as a consonant. You often get changes when these occur next to items which begin or end with vowels that don't go nicely with them. All sorts of changes can occur--the semi-vowel may get deleted, or changed to something else, or something else may get inserted.

Figure B.10: aRaBCALL displaying third level feedback on the misuse of the weak verb 

يشكو $y$s$kw$
Appendix C: Questionnaire Template

aRaBCALL System Evaluation Questionnaire

General information
- Age: __________
- Gender: __________
- Education: ______________
- First language: ______________

The purpose of the questionnaire is to evaluate the whole aRaBCALL system to develop it in a way that supports the project. Some questions relate to how easy the system is to use while others relate to the power of the interface. A number of the questions have space for you to enter free text as well making a choice. You can fill these in or leave them blank as you choose.

The questionnaire is in four parts. Part A focuses on ease of use, Part B on features of exercises, Part C on the details of the question topics, and part D on general information.

Part A: Evaluation of ease of use

Please make either for yes or no box for each question that more suitable for your answer

<table>
<thead>
<tr>
<th>No.</th>
<th>Question</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1</td>
<td>Do you think that any previous knowledge about computers is necessary to</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>work with the system?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Comment:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A.2</td>
<td>Does the system meet with your expectation?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Comment:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A.3</td>
<td>Was the system easy to work with?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Comment:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A.4</td>
<td>Do you think that some training in how to use the system would have been</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>useful?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Comment:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A.5</td>
<td>Did the system help you learn?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Comment:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A.6</td>
<td>Was the system reliable?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Comment:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A.7</td>
<td>Do you think you would have improved more if you had worked more with the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>system?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Comment:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Part B: Evaluate the features of exercises

<table>
<thead>
<tr>
<th>No.</th>
<th>Question</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1</td>
<td>Describe two particularly strong features of the system exercises</td>
<td></td>
</tr>
<tr>
<td>B.2</td>
<td>Describe two areas of weakness you would like to see changed in a particular exercise.</td>
<td></td>
</tr>
<tr>
<td>B.3</td>
<td>List the ways you will change in your practice homework because of this system</td>
<td></td>
</tr>
</tbody>
</table>

### Part C: Evaluate the details of exercises

Tick *(ONLY ONE)* in the box that you most agree:

<table>
<thead>
<tr>
<th>Question numbers</th>
<th>Very useful</th>
<th>useful</th>
<th>Fair</th>
<th>Poor</th>
<th>Very poor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjectives</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>هذه نهر طويلة</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>هذا قلم سوداء</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>هذه بنت جميلة</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>لي ولد واحدة</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>هذا بنك كبير</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>لا ذهب الولد</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>لم ذهب البنت</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>لن شرب الطفل العصير</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ما يلعب الولد</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>لم تكتب البنت</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pronouns</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>هاتان دارسات</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>أنت يلعب</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>هذه رجل طويل</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>هذا بنت</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>هذان دارسون يكتبون الدرس</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verbs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>البنت لعبت</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>لعب البنت</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>الولدان يلعب</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>الرجل يكتب كتاب</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>الرجل يكتب كتاب</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Appendix C: Questionnaire Template

#### Weak verbs

<table>
<thead>
<tr>
<th>English</th>
<th>Arabic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$kY AITfl.</td>
<td>شكا الطفل</td>
</tr>
<tr>
<td>y$kw Ontm.</td>
<td>يشكو أنتم</td>
</tr>
<tr>
<td>y$kJ h*A.</td>
<td>يشكي هذا</td>
</tr>
<tr>
<td>$kJ h*h.</td>
<td>شكي هذه</td>
</tr>
<tr>
<td>$kw Ont.</td>
<td>شكو أنت</td>
</tr>
</tbody>
</table>

#### Subordinate causer/ complementiser

<table>
<thead>
<tr>
<th>English</th>
<th>Arabic</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEtqd Arjl On *hb AldArs.</td>
<td>اعتقد الرجل أن ذهب الدارس</td>
</tr>
<tr>
<td>dArsp IEtqd On Aldrs SEb.</td>
<td>دارسة تعتقد أن الدرس صعب</td>
</tr>
<tr>
<td>IEtqd Arjl On AldArs *hb.</td>
<td>اعتقد الرجل أن الدارس ذهب</td>
</tr>
<tr>
<td>IHtsb Alwld On Aldrs SEb.</td>
<td>احتسب الولد أن الدرس صعب</td>
</tr>
<tr>
<td>Hsb Aldrs mktwb.</td>
<td>حسب الدرس مكتوب</td>
</tr>
</tbody>
</table>

#### Part D: General comments

A. Order the exercises from 1 (very useful) to 6 (worse) in terms of how useful or better exercise

<table>
<thead>
<tr>
<th>adjectives</th>
<th>verbs</th>
<th>Negation</th>
<th>Pronoun</th>
<th>Weak verbs</th>
<th>complementiser</th>
</tr>
</thead>
</table>

B. When the system said you made a mistake, which of the following did you usually do? Pick just one option (by making a circle on the letter of the correct sentence) from the following:

a. Try to fix it immediately
b. Go to the summary information and then try to fix it
c. Go to the help feedback file and then fix it
d. Go on to the next sentence

C. The system keeps a running score of how you are doing based on how many sentences you get right and how much use you made of the help files:

1. Did this make any difference to how much you asked the help files?

   YES [ ] or NO [ ]

2. Would it have made more difference if this score was used by the teacher for assessing your progress?

   YES [ ] or NO [ ]

D. Were the detailed help files useful, or did you get just as much information from the initial summary messages?

   YES [ ] or NO [ ]

#### Part E: General comments

- If there is anything else you would like to add, please use the box below:
Appendix D: Questionnaire Analysis

The following are the results collected from the student feedback for individual questions:

**Figure D.1: Scores for adjective exercise**

**Figure D.2: Scores for pronoun exercise**
Appendix D: Questionnaire Analysis

Figure D.3: Scores for negation exercise

Figure D.4: Scores for use of weak verbs exercise
Appendix D: Questionnaire Analysis

**Figure D.5**: Scores for complementiser exercise
Appendix E: Transition Behaviour

(F.1)

(F.2)

(F.3)
Appendix E: Transition Behaviour

(F.4)

(F.5)

(F.6)
Bibliography


Bennett, P. (1997). Feature-Based Approaches to Grammar, Manuscript Language Engineering, UMIST, Manchester, UK.


Ellis, N. C. (2002). Frequency Effects in Language Processing: A Review with Implications for Theories of Implicit and Explicit Language Acquisition, SSLA, 24, pp.143–188


Bibliography


