It is necessary and possible to build (multilingual) NL-based restricted e-commerce systems with mixed sublanguage and content-oriented methods

Daoud Daoud

THESE
présentée et soutenue publiquement par

Daoud Maher DAOUD

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITÉ JOSEPH FOURIER – GRENOBLE 1

Spécialité
INFORMATIQUE

Il faut et on peut construire des systèmes de commerce électronique à interface en langue naturelle restreints (et multilingues) en utilisant des méthodes orientées vers les sous-langages et le contenu

It is necessary and possible to build (multilingual) NL-based restricted e-commerce systems with mixed sublanguage and content-oriented methods

Date: 20 September 2006

Jury:

M.	Yassine	LAKHNECH	Président
M.	Fathi	DEBILI	Rapporteur
M.	Yves	LEPAGE	Rapporteur
M.	Patrice	POGNAN	Rapporteur
M.	Igor	BOGUSLAVSKIJ	Examinateur
M.	Jesus	CARDENOSA	Examinateur
M.	Christian	BOITET	Directeur

**THÈSE PRÉPARÉE AU SEIN DU GETA,
LABORATOIRE CLIPS (IMAG, UJF, INPG & CNRS)**

Résumé

Aucun système de commerce électronique opérationnel n'est capable de traiter en ligne des requêtes d'utilisateurs exprimées en langue naturelle spontanée. Certains systèmes évitent le problème difficile du développement et du support d'une interface en langue naturelle en simplifiant le type d'interaction de l'utilisateur, par l'utilisation de formulaires à remplir ou d'un langage naturel contrôlé. D'autres systèmes ont cherché mais échoué à offrir une interface en langue naturelle spontanée, parce qu'ils utilisaient des techniques de TALN inadaptées.
Le but de cette thèse est de montrer qu'il est nécessaire et possible de construrie des systèmes de commerce électronique à interface en langue naturelle restreints (et multilingues) en utilisant des méthodes orientées vers les sous-langages et le contenu. L'analyse du sous-langage et l'intégration de méthodes orientées vers le contenu augmentent en effet l'exactitude et la robustesse du traitement de façon décisive.
Pour vérifier cela, nous avons construit un système expérimental, CATS, comme "preuve de concept". C'est un système de petites annonces en langue naturelle (actuellement l'arabe) basé sur les SMS destiné à mettre en contact des personnes désirant vendre ou acheter des voitures d'occasion, de l'immobilier, etc. Pour analyser le sous-langage très particulier de ces petites annonces en SMS, nous sommes partis d'un corpus web de messages de ce type (mais pas en SMS) pour construire un système de base couvrant l'occasion automobile et l'immobilier en Jordanie. Ce premier système a été déployé, ce qui nous a permis de collecter un corpus réel de SMS "spontanés" dans ces domaines, et d'ajuster finement CATS à ces domaines.
Le traitement sémantique étant nécessaire, nous avons défini CRL-cats, un langage de représentation du contenu très simple, mais suffisant pour exprimer le contenu de telles petites annonces. Nous avons écrit l'extracteur de contenu dans le langage spécialisé pour la programmation linguisitique EnCo, dans lequel nous avions déjà écrit le premier "enconvertisseur" arabe-UNL. Ce langage étant d'assez bas niveau, et n'offant aucune aide à la programmation modulaire, nous avons développé une méthodologie qui facilite l'écriture d'enconvertisseurs aussi bien que d'extracteurs de contenu, et permet un codage systématique et efficace.
La génération des réponses est basée sur une reconnaissance de patrons sémantiques, différents selon qu'il s'agit d'une annonce de recherche ("look for") ou d'offre ("sell"), et sur un mécanisme de raisonnement, de sorte qu'on peut traiter les situations "sans réponse". À la différence d'autres systèmes expérimentaux, CATS a été conçu dès le départ pour être un "système de production". Il est actuellement déployé en Jordanie par le plus grand opérateur de téléphonie mobile (Fastlink), qui lui a d'abord fait passer des tests sévères. Le test de l'extracteur de contenu sur du texte réel et bruité a donné une f-mesure de 90%. Le temps de réponse moyen est d'environ 10 à 30 secondes à une heure de pointe (10 annonces par minute).
Mots-clés : interface en LN spontanée, services SMS, sous-langages, extraction de contenu, petites annonces, traitement de l'arabe

Abstract

The survey of the available e-commerce systems shows that none of them is able to handle spontaneous users' requests online. Some systems avoid the hard problem of supporting free natural language interface by simplifying the user interaction styles either by using form filling or by using controlled languages. Other systems failed to support free natural language interface because they used inadequate NLP techniques.

The purpose of this thesis is to show that it is necessary and possible to build (multilingual) NL-based e-commerce systems with mixed sublanguage and content-oriented methods. The analysis of the sublanguage and the integration of content-oriented methods will definitely increase the accuracy and robustness of the processing.

To verify this assumption, we built an experimental system as a proof-of-concept. The system is a SMS-based classified ads selling and buying platform. It allows users to send classified ads of the articles/goods they would like to sell and to search for the goods/articles they desire using full natural language interface. To analyze the sublanguage, we first used a web based corpus to build the basic system which covers the Cars and Real Estate domains. This initial experimental deployment of the system was to collect real SMS-based spontaneous data, which were used to fine tune the system.

To enable semantic processing, a content representation language is defined to capture the meaning of a classified ad post. The semantic grammars of content extraction are coded using the EnCo specialized language for linguistic programming which we used previously in developing the first Arabic-UNL enconverter. To enhance the process of coding using EnCo, we have developed a methodology that facilitates this process and provides the means for a systematic and efficient coding.

Response generation is based on semantic matching ("looking for" and "sell" posts) and reasoning and is able to handle "no answer situations". Not like other experimental systems, CATS was designed from the beginning to be a "production system". It is currently deployed in Jordan by the largest mobile operator (Fastlink) after passing intensive testing by them. Testing the content extraction component with a real noisy free text shows a 90% F-measure. The average response time is around 10~30 seconds calculated during peak time (10 posts/minute).

Keywords: spontaneous NL interface, SMS services, sublanguages, content extraction, classified ads, Arabic processing

Thanks and acknowledgments

The work on this thesis has been an inspiring, often exciting, sometimes challenging, but always interesting experience. It has been made possible by many other people, who have supported me.

I am very grateful to my supervisor Christian Boitet who has given me the chance to resume my study. He has supported me with his encouragement and many fruitful discussions.

I would also like to express my sincere thanks to my brother Mustafa for his invaluable experience and advice.

Finally, I wish to thank my parents, my wife, and my children for their continuous support, encouragement and patience.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Introduction to the thesis

A natural language interface accepts users' inputs in natural language interacting with some system, typically a retrieval system, which then results in adequate responses to the commands or query statements. Hence, a natural language interface should be able to translate uncontrolled natural language statements into suitable actions for the system.

This type of unrestricted NL interface is an interesting option because, if it could be built, it would offer many advantages. Firstly, it does not require any learning and training, because its structure and vocabulary are already familiar to the user. Secondly, natural language enables users to encode complex meanings. Thirdly, this type of interface is text-based, making it suitable for all types of devices and medium. In contrast, form-based or graphical user interfaces need more sophisticated and specific resources.

Incorporating a NL interface requires translating ambiguous user's inputs into clear intermediate representations. Two main problems are associated with building such systems: the first one is handling linguistic knowledge and the second one is handling domain knowledge.

The study of the current scene shows that the deployed or operational e-commerce NL interface systems are rare and most of them are only prototypes. This problem is not related to the openness or restrictiveness of the domain. Although most e-commerce activities are domain-specific, we don't yet find any restricted NL interface-based e-commerce operational systems.

NL-based systems have the reputations of high development cost and low quality. Our goal in this thesis is to show that the most important factor in building NL based systems is the selection of the right methodologies for the development, regardless of the target language in terms resources richness, or type or complexity of the domain, or even cleanliness of the input text. If this is approach is combined with treating a NLP project as an engineering problem, and not only as a traditional linguistic problem, it is almost guaranteed to produce a system with industrial quality and high extensibility, with the minimum resources possible.

Hence, we built an experimental system as a proof of concept. The system is a SMS-based classified ads selling and buying platform. It allows users to send classified ads of the articles/goods they would like to sell and to search for the goods/articles they desire using full natural language interface. The system extracts content from both "sell" and "looking for" posts and transforms the natural language text into a corresponding content representation. For a "sell" post, the content representation is mapped into database records and stored into RDMS. For a "looking for" type of posts, the content representation is used to build a SQL query to retrieve information from the data that has previously been processed and stored in the RDMS.

This thesis is divided into three parts. The first part will describe the current scene concerning our assumptions and our proposed solution. In this part, we will describe the main requirements of the proposed system, its main components, and its internal and external data specifications.

In the second part, we will focus on the Content Extraction process. We will describe the programming language we used, our lingware methodology, and our approach to the extraction of content from Arabic spontaneous and noisy text.

In the final part, we will show some operational aspects of the CATS system and its current status, before evaluating and comparing it with other systems. We will also discuss issues related to porting the system to other languages and other domains.

Part A. Why and How to build Natural Language SMS-based E-Commerce System?

Introduction to Part A

Most of the NL-based systems are experimental projects that end in the labs. Very few of them are converted into real systems or products. Certainly, the main cause of this problem stems from their weak interpretative power which in turn is caused by their inability to deal with the variations in human use of natural language [Owei, et al. 1997]. This can be aggravated or minimized by the implementation methods used.

The key success factors in building NL systems are understanding how people encode their thoughts, and finding the right representation to model the concerned domain knowledge. Additionally, a NL-based system should also be treated as an engineering project, not to be treated only from the linguistic angle.

On the other hand, building a system that can be a "production system" has different development requisites from building a prototype. The re-implementation (from prototype to "production system") often doesn't happen [Leidner 2003]. This is because the objective of building a prototype is only to demonstrate the feasibility of a method. Alternatively, building a practical NLP system always forces you to take a lot of additional constraints into account that are negligible from a theorist's viewpoint.

This part is divided into three chapters. In the first chapter, we will analyze the current scene, and then we will introduce our assumptions and finally will introduce our solution.

In the second chapter, we present the requirements of the experimental system. We will also introduce our approach to sublanguage analysis.

In the third chapter, the architecture of the system will be introduced, with a focus on its modular architecture. Additionally, we will present the knowledge representation used in this system.

Chapter 1. The scene and problems with current approaches

Introduction

Although natural language (NL) interface is the best style of interaction, it is not used currently in most e-commerce systems. The reason for this observable fact is that NL-based systems are hard and expensive to build. However, we believe that this is not absolutely true if a right approach is used to build such systems.

In this chapter, we describe the current scene regarding some information systems from the natural language interface and processing perspective. Furthermore we would like to explore different natural language styles and processing techniques. We will try to find out why NL interface is not used in building e-commerce systems and to explore why these systems remain prototypes or research projects.

We will then propose a solution based on our analysis of the current picture.

This chapter is divided into three parts. In the first one we explore some e-commerce and information systems that use some NL interface. In the second part, we analyze the main NL interface and processing approaches. In the third part, we introduce our solution.

1.1 Existing systems and not covered needs

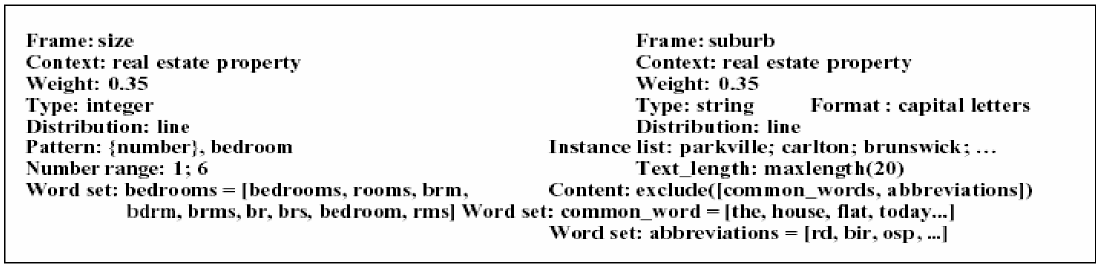
Electronic commerce may be defined as the set of activities of trading goods and services online. It can be structured into the market segments business-to-customer (B2C) such as online shops and auctions provided by portals, business-to-business (B2B), customer-to-business (C2B) such as reverse auctions, and customer-to-customer (C2C) e-commerce.

In this part, we will look at different e-commerce systems and discuss them, mainly from the interface and text processing perspectives.

1.1.1 E-Commerce systems

1.1.1.1 CASA (Classified Ads Search Agent)

This system was built by Sharon Gao [Gao and Sterling 1998]. CASA was tested specifically on house classified ads. It is a rule-based system that uses Information Extraction (IE) to extract knowledge from text. Parsing of semi-structured text is guided by knowledge units which are word groups or phrases with an independent and specific meaning. For example, "$200 per week" is a knowledge unit price with value 200. The main problem is that the text parser has low precision and especially low recall for suburb parsing. The main reason for this is that the text parser is not sufficiently flexible to parse the widely different advertisement structures. Users can query CASA through form filling, by entering specified fields such as location, number of bedrooms and price. The matching component compares the keywords expressing the user's requirements with every group of classified ads knowledge units. Finally, CASA lacks the functionalities of reasoning, relaxation, and handling no answer situations which are essential in satisfying user's requests.



```
Frame: size                                        Frame: suburb
Context: real estate property                      Context: real estate property
Weight: 0.35                                       Weight: 0.35
Type: integer                                      Type: string       Format : capital letters
Distribution: line                                 Distribution: line
Pattern: {number}, bedroom                  Instance list: parkville; carlton; brunswick; ...
Number range: 1; 6                                 Text_length: maxlength(20)
Word set: bedrooms = [bedrooms, rooms, brm,     Content: exclude([common_words, abbreviations])
        bdrm, brms, br, brs, bedroom, rms] Word set: common_word = [the, house, flat, today...]
                                            Word set: abbreviations = [rd, bir, osp, ...]
```

Figure 1: CASA knowledge units

Table 1: specs of the CASA system

Domain	Real Estate
Documents source	Web
Document type	Semi-structured text
Intermediate representation	knowledge components
User requests	Form based filling (key words inputs)
Matching	Matching each keyword entered by users with each knowledge units groups

1.1.1.2 TREE (TRans-European Employment)

The project, which ran from September 1995 to November 1998 and was funded by the Language Engineering sector of the EU Commission's Fourth Framework Programme, implemented an on-line multilingual employment service, through which prospective employees would be able to read job opportunity announcements in any of several European languages, independent of the language in which the employer originally drafted the announcement. A prototype was implemented on the Internet treating requests in English, Dutch, French, Swedish and Finnish. Unfortunately, it is no longer available.

The system permits users offering jobs to submit semi-structured job descriptions by e-mail. The system converts these texts into language-independent schematic representations which are then stored in the jobs database. The analysis methodology adopted is a hybrid of traditional pattern-matching techniques and example-based fuzzy matching. Job seekers can query the system by filling an HTML form consisting of a number of fields which correspond to job-schema object attributes (e.g. job-title, location etc.).

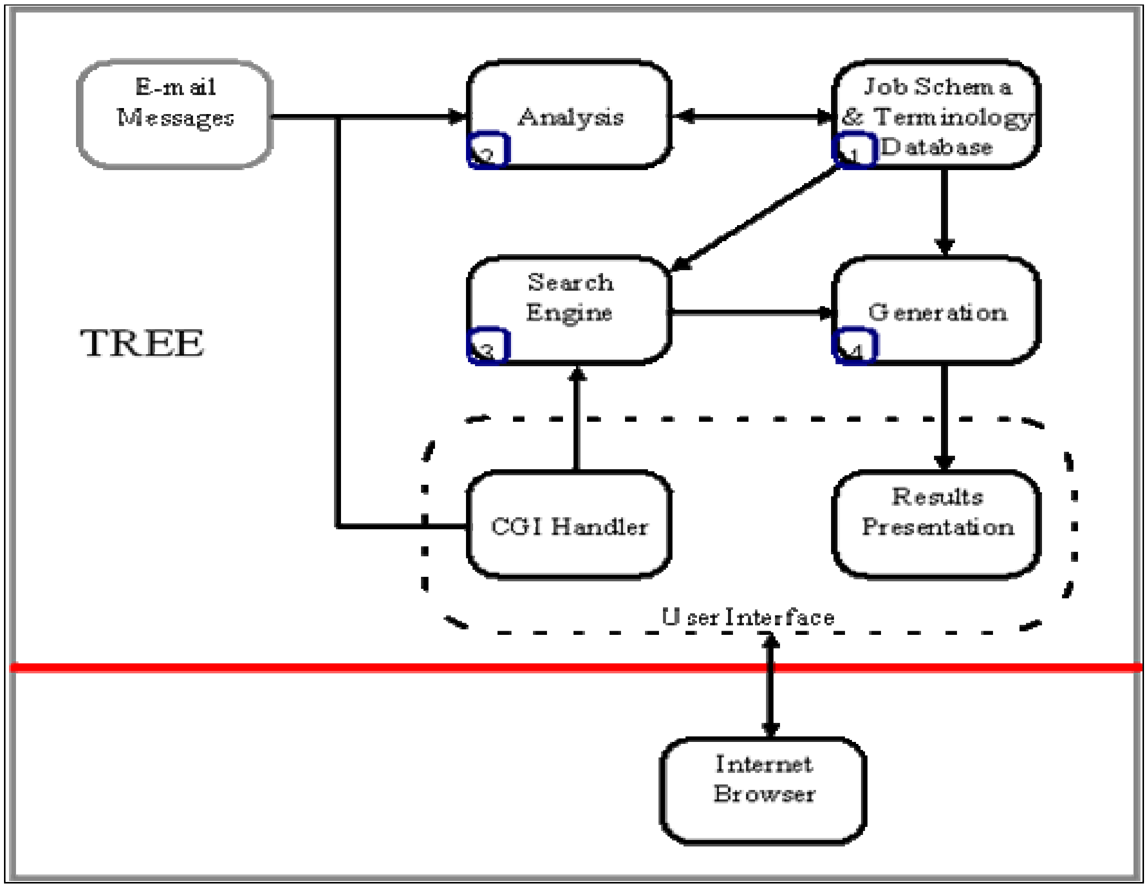


Figure 2: TREE system design

Database queries are conducted by matching the "ideal" job as specified by the user against job-schemas held in the database. The matching process yields a numeric result representing the "distance" between two objects. Identified jobs can then be ranked according to how closely they resemble the user's ideal job. The results of a database query are then fed to the generation module for subsequent presentation in the language specified by the user [Somers, et al. 1997].

Table 2: specs of the TREE project

Domain	Jobs
Documents source	Email
Document type	Semi-structured text, "Controlled"
Semantic	Language-independent templates / reasoning
User requests	Form-based filling (keywords inputs)
Matching	Distance-based keywords matching

1.1.1.3 The MKBEEM (Multilingual Knowledge Based European Electronic Marketplace)

MKBEEM is a European project (1st Feb. 2000 – 30th Nov. 2002) [MKBEEM 2005]. It centered on written language technologies and its use in commerce in two typical domains (B2C tourism, B2C mail order) for enabling several parties to perform commercial exchanges in a transparent way, independently of the language of the end user, the service, and the product provider. Ontologies are used for classifying and indexing catalogues, for filtering users' queries, for selecting relevant products and providers, for facilitating multilingual man-machine dialogues between users and software "agents", and for inferring information relevant to the users' requests and trading needs.

A product description text is processed and information is stored into the product database. This includes the translated product articles, the extracted properties, and the results of inferences based on the properties and the market-specific categories to which the product belongs.

Natural language query processing is used for testing the newly added products so that they are found easily and from the correct places in the catalogue. The queries are analyzed and the extracted properties are matched against saved properties of the products in the ontology. However, as described in [Heinecke and Toumani 2003], the matching procedure is very complicated: it involves a graph based-matching of the semantic representation of the user query with the graph representation of the product ontology.

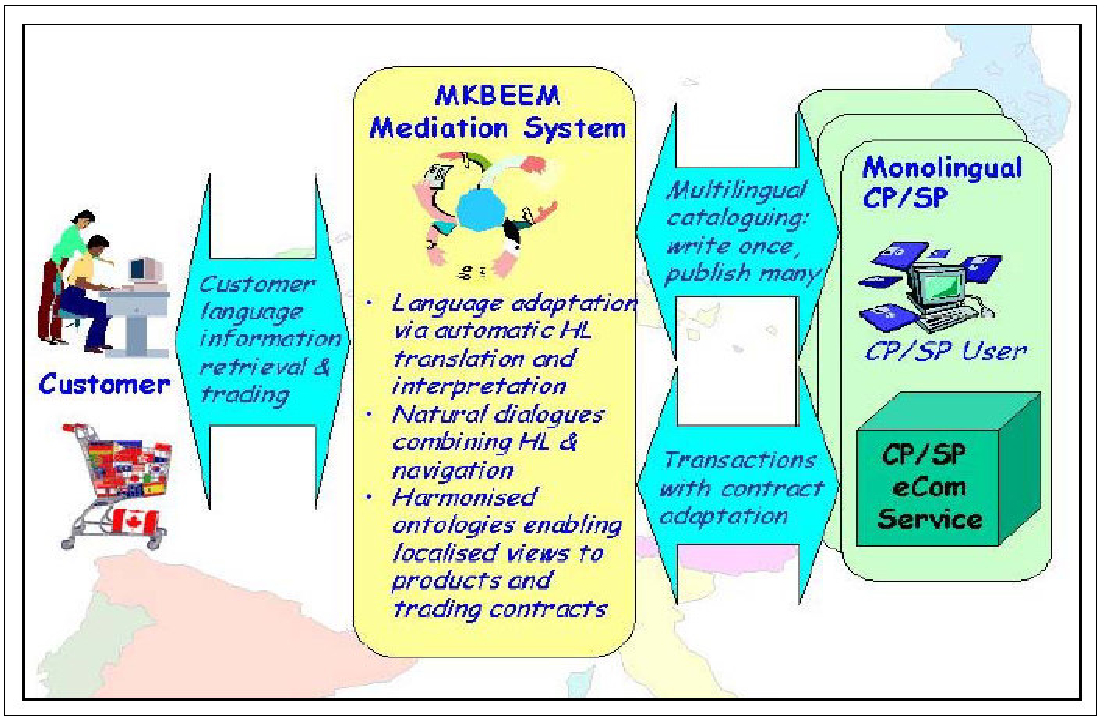

MKBEEM
Mediation System
Customer
Customer language information retrieval & trading
• Language adaptation via automatic HL translation and interpretation
• Natural dialogues combining HL & navigation
• Harmonised ontologies enabling localised views to products and trading contracts
Multilingual cataloguing: write once, publish many
Transactions with contract adaptation
Monolingual CP/SP
CP/SP User
CP/SP eCom Service

Figure 3: operating context of the MKBEEM system

As to the type of natural language interface implemented in the prototype, it is not specified clearly in any of the papers written about this project and no examples of results are given. Additionally, we could not test this product since the project homepage is not open for public access. However, since MKBEEM machine translation and content extraction is based on WebTran [Lehtola, et al. 2003] which is a machine translation engine that uses controlled language [Lehtola, et al. 1998, Lehtola, et al. 1999], we can conclude that MKBEEM does not offer full spontaneous natural language interface but only implements a predefined language model.

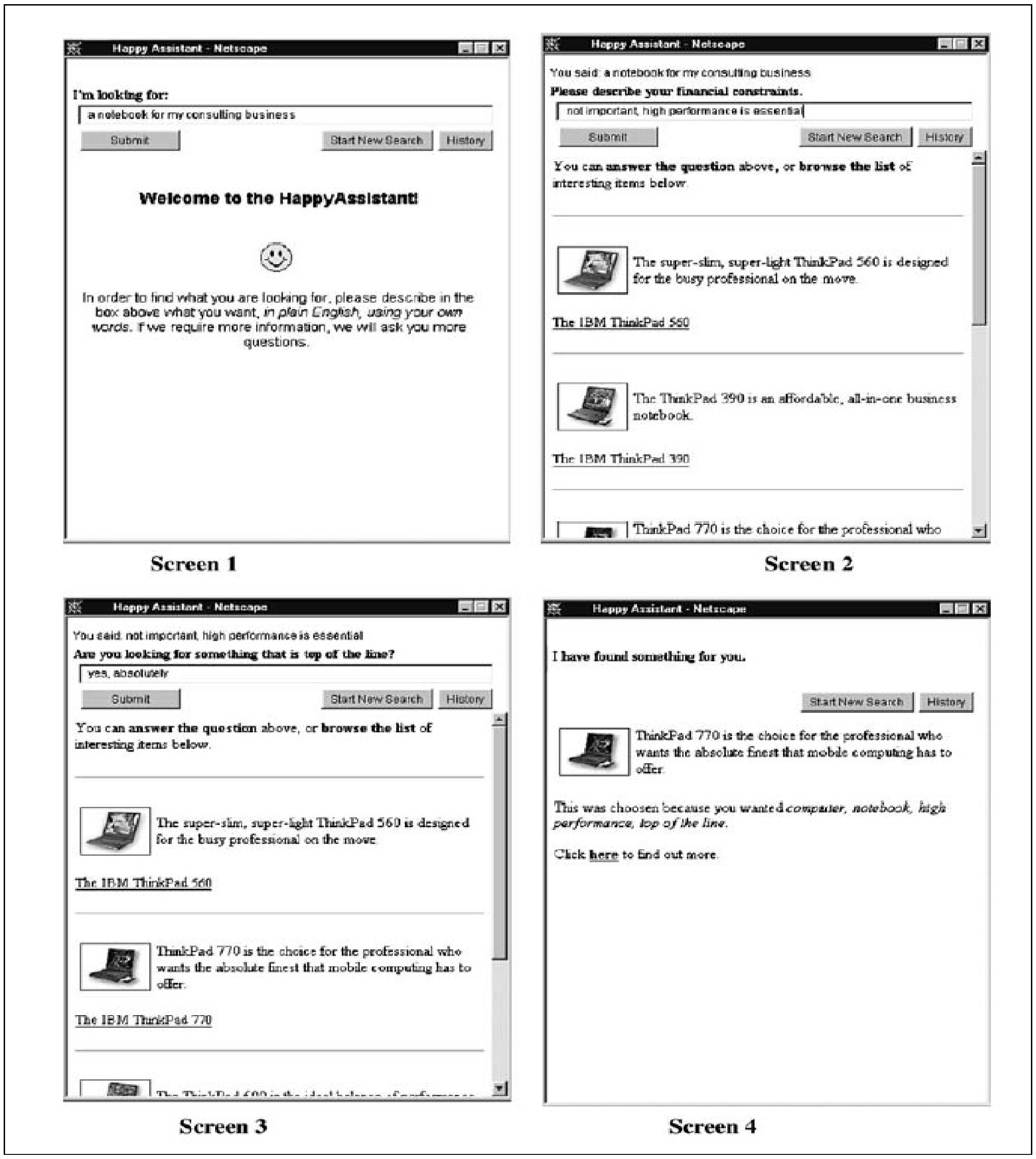
Table 3: specs of the MKBEEM project

Domain	Tourism, mail order
Documents source	Users
Document type	Controlled language
Intermediate Representation	Language independent templates
User requests	Controlled language
Matching/retrieval	Graph-based

1.1.1.4 HappyAssistant

The system allows customers to make requests in natural language and directs them towards appropriate web pages that sell the product or provide the service they need, in the field of computers. The task is to buy a personal computer.

In the prototype system, the Presentation Manager employs a natural language parser to transform the user's natural language query into a logical form, and sends the logical form to the Dialog Manager. It is also responsible for obtaining the system's response from the Dialog Manager and presenting it to the user. The Dialog Manager is responsible for determining the specific action(s) requested by the user and filling the parameters (e.g., the attributes of the computers in which users are interested) of the identified action by way of a dialog with the user. The Knowledge Base for business rules specifies the translation from user requests to action plans for the Action Manager to satisfy the requests, for example, retrieving information about particular computer models from the product catalog [Chai, et al. 2002].


Happy Assistant - Netscape

I'm looking for:

a notebook for my consulting business

Welcome to the HappyAssistant!

In order to find what you are looking for, please describe in the box above what you want, *in plain English, using your own words*. If we require more information, we will ask you more questions.

Screen 1

Happy Assistant - Netscape

You said: a notebook for my consulting business

Please describe your financial constraints.

not important, high performance is essential

You can **answer the question** above, or **browse the list** of interesting items below.

The super-slim, super-light ThinkPad 560 is designed for the busy professional on the move.

The IBM ThinkPad 560

The ThinkPad 390 is an affordable, all-in-one business notebook.

The IBM ThinkPad 390

ThinkPad 770 is the choice for the professional who

Screen 2

Happy Assistant - Netscape

You said: not important, high performance is essential

Are you looking for something that is top of the line?

yes, absolutely

You can **answer the question** above, or **browse the list** of interesting items below.

The super-slim, super-light ThinkPad 560 is designed for the busy professional on the move.

The IBM ThinkPad 560

ThinkPad 770 is the choice for the professional who wants the absolute finest that mobile computing has to offer.

The IBM ThinkPad 770

Screen 3

Happy Assistant - Netscape

I have found something for you.

ThinkPad 770 is the choice for the professional who wants the absolute finest that mobile computing has to offer.

This was choosen because you wanted *computer, notebook, high performance, top of the line.*

Click here to find out more.

Screen 4

Figure 4: examples of interactions using HappyAssistant

The initial matching algorithm iterates through all the rules in the knowledge base, and calculates the rank of each rule. The presence of any rule with rank zero indicates the triggering of that rule, i.e., that a particular product or service matches the user description. In this case, a detailed web page regarding that item is displayed in a separate browser window. If there are multiple rule triggerings, the rule with the highest weight is selected. If there are no rules of rank zero, then an additional refinement is needed to recommend a suitable item. HappyAssistant chooses from the rules of the lowest rank and the highest weight, and from that finds a concept that has not yet been identified.

The system retrieves the natural language question associated with that concept from the Domain Lexicon and poses it to the user, prompting for a response.

However, the natural language analysis is very limited, handling only noun phrases.

Table 4: specs of HappyAssistant project

Domain	Computers
Documents source	Manually processed
Document type	Structured
Intermediate Representation	Logical form
User requests	Limited sentences (Natural language Dialogue interfaces)
Matching/retrieval	Key word-matching

1.1.2 Information systems/assistants

1.1.2.1 MIETTA (Multilingual Information Extraction for Tourism and Travel Assistance)

MIETTA is a European Union funded project that integrates information retrieval with the areas of shallow natural language processing and information extraction [Buitelaar, et al. 1998]. The main objectives of the projects are:

- Providing full access to all information independent by of the language in which the information was originally encoded and independent by of the query language;
- Providing transparent natural language access to structured database information;
- Providing hybrid and flexible query options to enable users to obtain maximally precise information.

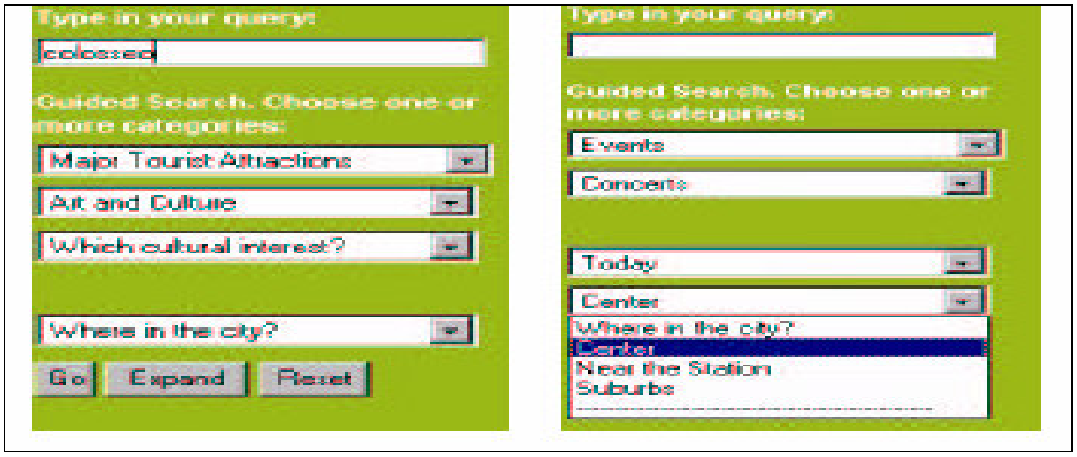


Figure 5: MIETTA form based query and search menu

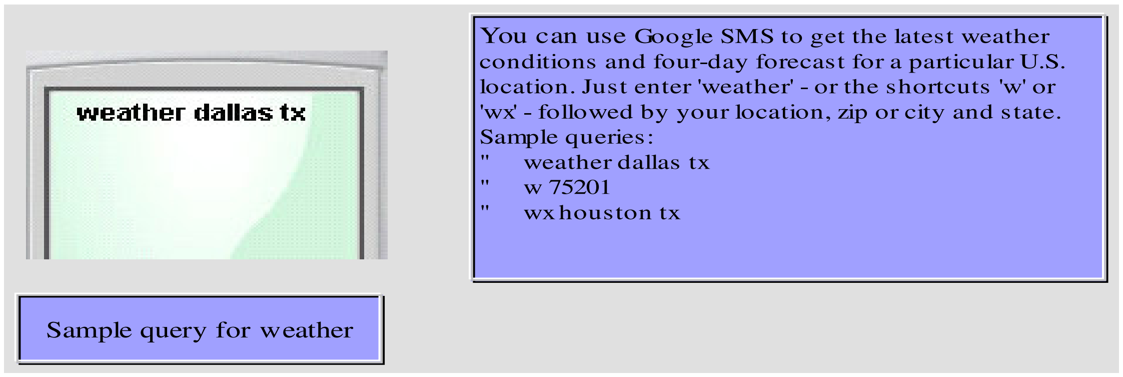
The system basically extracts information from web-based semi-structured text using Information Technology and fills slots of predefined Interlingua templates. Free text web-based documents are indexed using IR techniques and translated into each language incorporated in this project. The user can issue an IR based query to search for web-based documents. Alternatively, the user can perform form-based search by selecting fields in a search form to access the templates database.

Table 5: specs of MIETTA project

Domain	Tourism
Documents source	Web
Document type	Semi-structured text
Intermediate Representation	Language-independent templates
User requests	Form-based filling (key-words inputs), IR-based searching
Matching/retrieval	IR techniques, SQL

1.1.2.2 SMS Google

Google is targeting SMS customers, and has launched a variety of SMS based services based on controlled language interfaces. As shown in Fig 6, a very simple controlled language is used to query the latest weather information, reflecting the straightforwardness of the domain.



You can use Google SMS to get the latest weather conditions and four-day forecast for a particular U.S. location. Just enter 'weather' - or the shortcuts 'w' or 'wx' - followed by your location, zip or city and state.
Sample queries:
" weather dallas tx
" w 75201
" wx houston tx

Figure 6: an example of a controlled language interface "Google's SMS weather service"

Table 6: specs of the Google's SMS services

Domain	Variety of services (weather, movie showtimes, etc)
Documents source	database
Document type	structured
Intermediate Representation	-
User requests	Controlled language
Matching/retrieval	keywords

1.1.3 Needs not covered

The need for systems with natural language interfaces has become increasingly essential as more and more people access information through their web browsers and mobile phones. This increase in the use of the web to access information on research, commerce, business and finance, sports, health, etc., requires employing rapid, reliable and accurate means of interacting with those systems.

Most of these systems are web-based, despite the fact that the mobile medium is becoming more popular among people, specifically in the developing countries. Recent data shows that the world mobile usage reaches 2 billion [Smith 2005], while the world Internet usage is only 1 billion [*Internet usage statistics* 2006]. The cell-phone industry is one of the fastest-growing communication industries in the Arab world. According to recent studies, it is estimated that the number of mobile subscribers in Jordan will reach 4.5 million at the end of 2008. Recent (September 2005) statistical information indicates that a significant 41 per cent of Jordanians currently subscribe to a mobile phone service. This is a massive increase from the 8 per cent and 28 per cent penetration rate at the end of 2000 and 2004 respectively. On the other hand, 10 percent of the Jordanians use Internet, according to recently available data.

In the Arab world, the difference between mobile and Internet usage is even higher than in the world [ITUArabic 2005]. As shown in table 7, the mobile connections penetration is four time larger than that of Internet usage.

Table 7: mobile and Internet usage in the Arab world (2004)

	number	Penetration %
Mobile connections	45,929,800	14.51
Internet	11,755,000	3.71

Unlike the Internet, which requires expensive equipment, national infrastructure, and primarily knowing how to read and write, the cell phone provides an accessible, relatively inexpensive solution to being part of what is happening around you without requiring a high level of literacy.

Today users of natural language interfaces are not the specialized groups that comprised early NLI users. Users may be multilingual, global, and they may query using a computer, or a small device or appliance, or even their voice. Information is also much more varied and may include relations about online store inventory, maps of nearly every location on the planet, research data, stock exchanges, job banks, restaurants, local government information, and much more.

1.2 Analysis of NLP in existing systems

Different natural language styles have been used. Some of them developed to avoid the problems associated with full spontaneous NL-based systems. In the following we will review those styles.

1.2.1 Natural language interface styles

1.2.1.1 Free natural language interface

Evidently, natural language is considered the simplest technique of human-machine interaction. It allows a wide range of expressions by helping the user to specify as many parameters as needed in a single request. The defining characteristics of a natural language interface are that users need not explicitly learn the lexicon and syntax of the system, so that they are able to express what they want in the language they are used to [Long 1994].

Natural Language Interfaces were first used as means for querying databases "Natural Language Database Interfaces (NLDBIS) " [Androutsopoulos, et al. 1995]. The main idea was that, for a user with no deep computer science knowledge, it is easier to query the database in natural language instead of using SQL expressions. Moreover, natural language expressions are often shorter than SQL ones, and there are cases when it is difficult to formalize expressions like "some", "a few", "often" etc. Most natural language interfaces to database systems have been prototypes, built by the research community.

The first NLDBIS was the LUNAR system built by Woods in 1973 for NASA. It concerned chemical analyses of moon rocks. Two databases were implemented, including the chemical analyses and the literature references. It managed to handle 78% of requests without error, a figure that rose to 90% when dictionary errors were corrected. However, ungrammatical sentences were not handled well and the NLP component was not very flexible.

SHRDLU was an early natural language understanding computer system, developed by Terry Winograd at MIT for his Ph.D. Thesis 1968-1970. SHRDLU allowed user interaction using English terms. The user instructed SHRDLU to move various objects around in a small "blocks world" containing various basic objects: blocks, cones, balls, etc.

By the late 1970's, LIFER/LADDER was one of the first good database Natural Language Processing (NLP) systems. It was designed as a natural language interface to a database of information about US Navy Ships. This system used a semantic grammar to parse questions and query a distributed database.

Natural Language interfaces to databases were commercially available in the late 1970s, but largely died out by the 1990s: porting to new databases and especially to new domains requires very specialized skills and is essentially too expensive (automatic porting was attempted but never successfully developed) [Copestake 2003].

Those early prototype systems showed that the analysis and understanding of natural language input is a highly complex process, requiring linguistic knowledge (morphology, syntax, semantics, and pragmatics) as well as a well elaborated knowledge-base and a very complex dataflow control between the components [Vertan 2004].

Until now, such systems require careful use, because they strongly depend on the correctness of their input, which, however, often contains typing errors, grammatical re-formulations, or slightly ungrammatical everyday idioms.

Traditional techniques for handling these difficulties involved spelling correctors and very large grammars and lexicons, but these techniques turned out to be unsuccessful.

The second reason why natural language interfaces are not in common use today is the large amount of time it has traditionally taken to construct a natural language interface. Current technology is such that each interface must be constructed on a case by case basis for each application. Thus, only applications that can justify such large expenditure of manpower are candidates for possible applications. However, given the quality of the system that results, the effort has not proven to be worthwhile.

1.2.1.2 Controlled/restricted natural language interface:

A controlled natural language is a limited subset of the vocabulary and syntax of a full natural language. This allows ambiguity to be reduced and processing time to be kept within reasonable bounds [Long 1994]. To retain the properties of ease of use and ease of remembering, the limitations of the system must somehow be conveyed to the users without requiring them to learn the rules explicitly.

However, some of these systems impose limitations on the vocabulary and syntax to the extent that it is not any more a controlled natural language, but rather a command language.

In this technique, the system imposes on users the way they should encode their thoughts and utterances. It does not allow them to express their requests naturally, so that they have to remember the allowed lexicon and syntax of the controlled language. As an example, Google SMS[1] imposes a controlled language on its offered services. For example, to get movie showtimes theater listings and movie details, users should follow the following language model:

- If you're looking for showtimes of a movie that's currently playing, enter the movie's title followed by your location (a zip code or city and state).
- For a listing of theaters near you and the showtimes of their top movies, enter 'movie: theaters' or 'movie: showtimes' followed by your location.
- For a listing of movies playing near you and their theater location, enter 'movie: movies' or 'movie: films' followed by your location.
- For movie details such as running time, genre, MPA rating, and critics' rating, simply enter the movie title.
- Get local listings in addition to your movie showtimes when you type 'theaters' or 'films' followed by your location.

Get showtimes and theaters for a particular movie:
- king kong 94103
- king kong san francisco ca

Get movie details:
- star wars

Find the top movies playing near you and their theater location:
- movie: movies 94103
- movie: films san francisco ca

Find the theaters near you and their movie showtimes of the top movies:
- movie: theaters 94103
- movie: showtimes san francisco ca

Google SMS provides similar instructions for each offered service and users have to memorize them in order to use them correctly.

These systems might work well for simple tasks or under controlled environments where users know the task well and can be trained on the controlled language interface.

On the other hand, when the task becomes more complicated with many parameters to express, it is difficult to have a working system based on a controlled language without learning what structures are acceptable, which is impossible for public services.

For example, PENG is a machine-oriented controlled natural language that has been designed for non-specialists to write precise specification texts in a seemingly informal notation. To guarantee the efficient usage of this controlled natural language, a text editor with an intelligent feedback mechanism has been developed that guides the writing process and guarantees well-formed linguistic structures that can be translated unambiguously into first-order logic [Schwitter 2004].

Another controlled language system is the KANT System (Knowledge-based, Accurate Natural-language Translation), which has been primarily targeted towards the translation of technical documents of heavy equipments [Mitamura 1999].

[Pool 2005] evaluated 32 controlled natural languages which are available nowadays and came to the following conclusions:

> In many cases, controlled natural languages have not been designed for Web-scale use. Most often, they have been designed within commercial organizations for the

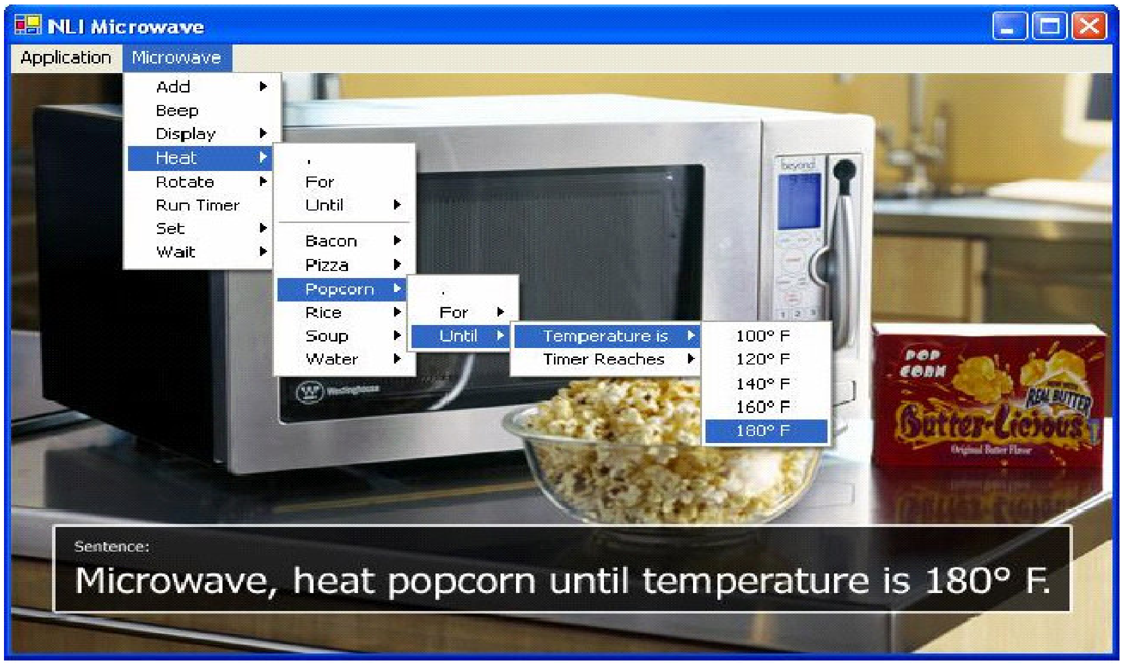
[1] Still in Beta stage

improvement of the processing of the language used in specialized functions performed in those same organizations. Utterances in that context are usually constrainable grammatically and lexically at relatively low cost, for three reasons. First, the domain is limited (e.g., heavy machinery maintenance). Second, the functions performed by utterances within the domain are limited (e.g., instructions). Third, a small and stable set of authors do the documentation, so that any investment they make in learning to comply with a controlled language is amortized over a large body of production. These advantages may account for the relative popularity of controlled natural languages as proprietary industrial tools.

1.2.1.3 Menu-based natural language interface (MBNLI)

To avoid the problem of training users which is associated with controlled natural language interfaces, menu-based natural language interfaces (MBNLI) have been introduced. They help users understand the capabilities of the system, thus avoiding any user negative or false expectation [Thompson, et al. 1983].

Menu-based natural language interfaces propose a broad variety of natural expressions: the user has the choice among a large number of alternatives at any place of the input, but the input language is still controlled in the sense that any choice has a well-defined (syntactic) structure with an associated precise semantic interpretation. Additionally, choosing from a menu eliminates typos. As the quality of a natural language processing system depends strongly already on the formal quality of the input, this kind of systems has been regarded as the ideal cost/effect compromise for human-computer interaction [Vertan and Hahn 2003].



Sentence:
Microwave, heat popcorn until temperature is 180° F.

Figure 7: example of MBNLI [Allen and Thompson 2005]

The first fully implemented interface, NLMenu, was developed in the eighties. It was a menu-based natural language understanding system. Rather than requiring the user to type his input to the system, input to NLMenu is made by selecting items from a set of dynamically changing menus. Active menus and items are determined by a predictive left-corner parser that accesses a semantic grammar and lexicon [Tennant, et al. 1983].

ROSY (RObust SYntactic analysis) (1987) was developed at the University of Saarbrücken and was meant as an extension of NLMenu, taking into account the particularities of the German language. With the addition of a morphological processor, the natural impression of the generated sentences was strongly improved. Several other interfaces based on the same principles were developed afterwards, mainly for database query tasks [Vertan and Hahn 2003].

One of the main disadvantages of this system is the use of an extensively controlled language. Additionally, it is not flexible and does not allow spontaneous input with obvious limitations on the menu design and updating of the dictionary. Finally, this approach is not adequate for all types of devices and communication modes and most likely will overwhelm users and frustrate them with lengthy interactions.

1.2.1.4 Natural language dialogue interfaces (NLDI)

A "Dialogue Interface" conducts a chat-like conversations with users, to extract exactly what they are looking for or trying to accomplish.

Natural language dialog has been used in many areas, such as for call-center/routing applications, e-mail routing information retrieval and database access, and for telephony banking [Chai, et al. 2002].

A good example of NLDI is the HappyAssistant dialogue system described by [Chai, et al. 2002]. Another example of NLDI systems is the Subjex.com search engine, which establishes two-way conversations with the users. Here's a sample dialogue:

User: WinZip
Subjex> *Would you like me to search for winzip?*

User:Yes
Subjex> *Here are some results. Should I look for* **2000** *or* **Winzip computing**
> *or something else.*

User: Something else
Subjex> what specific information about Winzip do you require?

User: Download sites

The approach of NLDI goes far beyond the keywords approach. It actually asks questions back to the user. This simple differentiation has a profound impact on the resulting user experience. A two-way conversation leads to fewer and highly relevant (targeted) search results; without forcing the user to wade through potentially millions of results or necessitate the learning of advanced search "Boolean" techniques using keywords. Keyword search give users only one opportunity to input what they are searching for before beginning another search. User queries are generally very short which naturally leads to a large number of documents being returned [Kruschwitz and Al-Bakour 2005]. Hence, adopting this interaction style is efficient in avoiding irrelevant information and in providing more precise results to users. Furthermore, this short and uncomplicated users' input will cause a more accurate processing of input text.

Specifically, in e-commerce environments, NLDI has also been proven to be more efficient in finding information than the keywords approach or MBNLI [Chai, et al. 2002].

In a similar manner with other NL interactions, the quality and complexity of such systems depends on the nature of the natural sublanguage used in the dialogues. Allowing unrestricted natural language dialogue in unrestricted domains would appear to require full human conversational competence, which does not seem feasible in the foreseeable future.

Additionally, this approach might work well with certain medium of communications such as Internet or PSTN, but, it is inappropriate for paid SMS-based services in which the users are charged for each message.

1.2.2 NLP approaches currently used

Different approaches are used to process natural language text. Strategies and techniques relevant for our goal of building NL-based interfaces for e-commerce include word-based, syntax-based and semantic-based processing.

1.2.2.1 Information retrieval (IR) techniques (bag of words)

Information retrieval involves returning a set of documents in response to a user query. The user specifies his information needs by providing sets of keywords and the information system retrieves the documents which best approximate the user query. Hence, the goal of this approach is to find relevant documents in a large collection, in response to user's query expressed as a sequence of words [Guarino, et al. 1999].

Search engines are a form of IR and one of the most essential tools on the Internet: they help find Web sites relating to a particular subject or topic using the bag of words approach.. Search engines are basically huge databases containing millions of records that include the URL of a particular Web page along with information relating to the content of the Web. Search engines crawl the Web and log the words from the web pages they find in their databases. The most popular search engines include: Google, AltaVista, Excite, Hotbot, Infoseek, Lycos, Webcrawler, Yahoo, and so on.

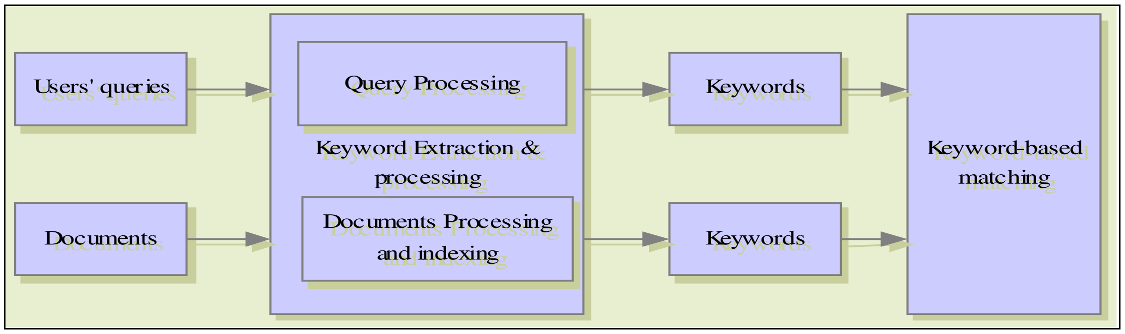


Figure 8: Keywords-based matching in IR-based e-commerce systems

Information retrieval systems are usually evaluated based on two metrics – precision and recall. Precision refers to the ratio of relevant (or correct) documents returned to the total number of documents returned. Recall refers to the number of relevant documents returned out of the total number of relevant documents available in the document collection being searched.

Retrieval based on keywords is often of fairly low quality. There are two possible reasons. First, the user query may be composed of too few terms which usually implies that the query context is poorly characterized. This is frequently the case, for instance, in the Web. This problem is dealt with through transformations of the query such as query expansion and user relevance feedback. Second, the set of keywords generated for a given document may fail to summarize its semantic content properly. This problem is dealt with through transformations in the text such as identification of noun groups to be used as keywords, stemming, and the use of a thesaurus. Additionally, for reasons of efficiency, text compression can be employed.

In the case of e-commerce applications this approach is unlikely to allow precise matching, because keywords based processing lacks the understanding of semantic meaning of the search words. [Chai, et al. 2002]. To demonstrate this, consider the following query:

مطلوب قطعة ارض في ابو السوس

"looking fore a piece of land in Abu Assoos[2]" .

All the returned results by Google in response to the above query are irrelevant. The top ranked page was an article about "secret negotiations between Israel and Hamas". We also discovered that the stemming techniques implemented by Google aggravated the results considerably. To show this:

قطعة [piece] is stemmed by removing (taa marboutah at the end) producing: **قطع**, which appeared in the document 6 times with a totally different meanings from the search query(suspend/ discontinue and cut). In the same document, ارض (land) appeared several times but with different meanings (territory, earth). The treatment of ابو السوس (abu assoos) as two separate words is another cause of these totally wrong results.

Additionally, bag of words processing is not adequate for handling numeric values such as price which are essential in e-commerce applications. Furthermore, given the dynamic nature of e-commerce contents, a more frequent crawling and indexing than in traditional content is required.

Despite a large body of research on keyword matching, the effectiveness of retrieval systems is still rather low. Although, it might work for search engines and large documents retrievals, it does not suits e-commerce systems. The types of dissimilarities between queries and documents cannot be solved using this approach.

1.2.2.2 Syntactic-based processing

In this type of systems, more processing is performed based on the syntactic structure of a sentence. The output of this processing is expressed usually by a parse tree. This approach is used mainly in traditional machine translation systems. However, for NL-based system that accepts users' queries and generate responses, different complexity factors [Appelt and Israel 1999] must be considered and studied before selecting the best level of syntactic analysis.

Different levels of text processing are used, for example, in the MUC events competing groups with a wide range of theoretical orientations converging on a common approach: domain-dependent templates for representing the critical patterns of concepts and a limited amount of syntactic processing to find appropriate phrases that fill slots in the templates [Sowa 1999]. This was illustrated by abandoning TACITUS which was a system that spent most of its time on syntactic analysis that were irrelevant to the ultimate goal. SRI replaced it with FASTUS, a system that is triggered by keywords, finds phrase patterns without attempting to link them into a formal parse tree, and matches the phrases to the slots in the templates [Sowa 1999].

[2] a location name in Amman

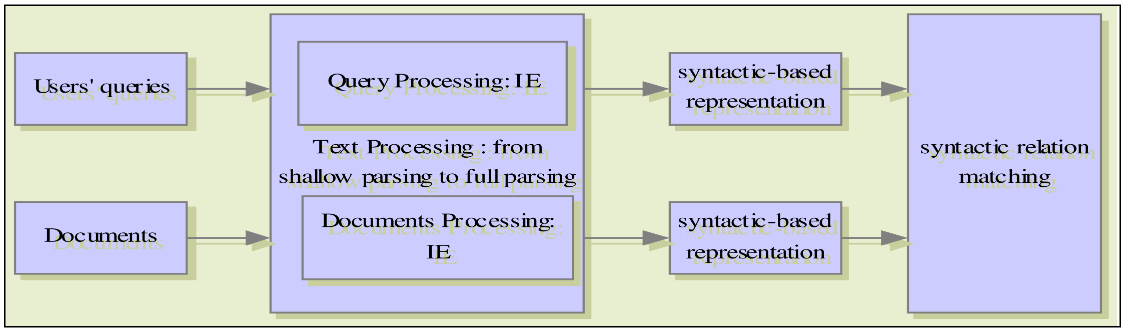

Users' queries
Documents
Query Processing: IE
Text Processing : from shallow parsing to full parsing
Documents Processing: IE
syntactic-based representation
syntactic-based representation
syntactic relation matching

Figure 9: a typical syntactic-based information system

Partial syntactic analysis of a sentence is called shallow parsing as an alternative to full sentence parsing. While earlier work in this direction concentrated on manual construction of rules, most of the recent work has been motivated by the observation that shallow syntactic information can be extracted using local information — by examining the pattern itself, its nearby context and the local part-of-speech information [Li and Roth 2001]. Learning and statistical methods has been used recently to recognize shallow parsing patterns such as syntactic phrase or words that participate in syntactic relationships.

On the other hand, to perform syntactic-based matching, both the words and the syntactic relations between them in the query are matched with words and relations in the document. Determining the syntactic relations requires full parsing of the sentence to identify the syntactic relations. However, reports show that using syntactic relation matching produces no improvement, over using just keywords. Indeed even more [Smeaton, et al. 1995] obtained worse results from relation matching using a tree-matching procedure than from keyword matching. One possible reason why syntactic relation matching has not yielded better results is the difficulty of identifying syntactic relations accurately [Khoo 1997].

Additionally, it is possible to use syntactic processing (shallow or full) to identify keywords and fill predefined templates. However, further processing and matching could be based on keywords only.

1.2.2.3 Semantic-based processing

Semantic-based systems process text with the objective of extracting content or meaning. These types of systems rely on semantic grammars, in which the sentence is not described by syntactic relations, but by using the semantic classes that categorize occurring words [Kittredge: 1982].

Semantic processing requires these topics to be addressed: knowledge representation, words meaning and the extension of knowledge representation with synonyms and related words [Engels and Bernt Bremdal 2000].

The LIFER/LADDER (1977) [Templeton and Burger 1983] system was the first to use a semantic grammar (that is, it used labels such as "SHIP" and "ATTRIBUTE" rather than syntactic labels such as noun and verb). This means that it was closely tied to the domain for which it was engineered.

The Conceptual Dependency Theory of Schank [Schank 1975] was the first attempt to apply knowledge representation to language processing [Engels and Bernt Bremdal 2000]. He wanted a representation that was unambiguous and unique. His aim was to express the meaning of any sentence in any language. The representations were intended to be language-independent. For any two sentences that are identical in meaning, regardless of language, there should be only one representation.

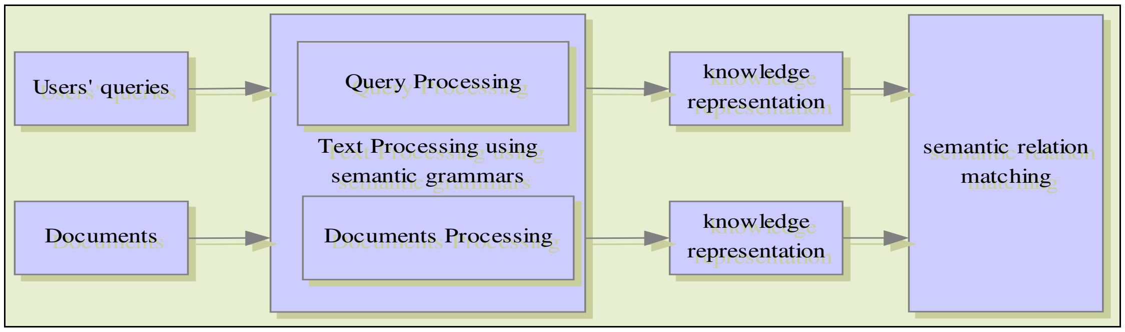


Figure 10: a typical semantic-based systems

The Conceptual Dependency formalism was used to build the first commercially deployed IE system ATRANS [Lytinen and Gershman 1986], which handled money transfer telexes. This system used semantics and did not require complete syntactic analysis of text and exploited the fact that the content of money transfer telexes is highly predictable. The text is processed by identifying actors (originating customer, originating bank, receiving bank, etc.) in order to fill in a template that was used, after human verification, to initiate automatic money transfers. It took something like thirty man-years to make it work [Schank 1991].

The main problem of semantic-based systems is that they are domain-specific. Hence, porting to other domains is often not straightforward. The development of such systems requires extensive domain knowledge. On the other hand, they have higher recall and precision than syntactic-based systems.

1.3 What should and could be done

In this section we will discuss the reasons that caused the failure of e-commerce systems to adopt full NL interface style. We will also set assumptions and propose a solution.

1.3.1 Assumptions based on the current scene

We have seen that no e-commerce system available today is able to handle spontaneous users' requests online. The study of the above systems confirms this fact. Those projects avoid this hard problem by simplifying the user interface either by using controlled languages, form filling, or NLDI.

Table 8: a comparison of user input NL styles and processing approaches for the considered systems

System	NL interface style	NLP approach
CASA	Form filling	
TREE	Form filling	
MKBEEM	Controlled language	Syntactic based on WEBTRAN
HappyAssistant	NLDI	Syntactic, for noun phrases only
MIETTA	Form filling and keywords	
GOOGLE SMS	Simple controlled language	

As an example, the failure of MKBEEM to provide full spontaneous NL interface is due to the use of methods and tools which are too complicated for the task. When we trace the project back to the beginning we find that one of its main objectives was to provide unrestricted NL interface. However, we could not find any evidence in the literature that this goal was ever achieved or demonstrated. The methodology used to extract content is very complicated. Initially, the input text is processed syntactically and several dependency parse trees are produced by WEBTRAN. Those dependency trees are then processed and mapped into semantic representations, which are finally transformed into CARIN (an ontological representation) (figure 11). Apparently, MKBEEM used these long and complicated steps of transforming one representation into another to meet the requirements of multilingualism which are provided by WEBTRAN. WEBTRAN is a machine translation system that analyzes input texts syntactically. The developers of this project decided to transform the syntactic representations into semantic ones which led to these complicated, long, and possibly error-prone processing steps.

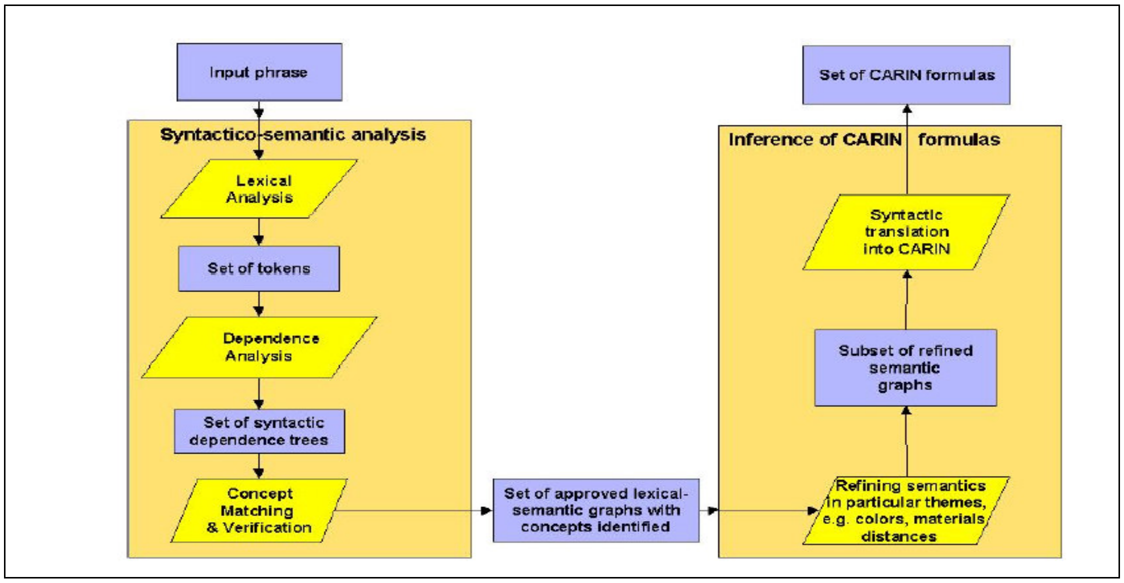

Input phrase
Syntactico-semantic analysis
Lexical Analysis
Set of tokens
Dependence Analysis
Set of syntactic dependence trees
Concept Matching & Verification
Set of approved lexical-semantic graphs with concepts identified
Set of CARIN formulas
Inference of CARIN formulas
Syntactic translation into CARIN
Subset of refined semantic graphs
Refining semantics in particular themes, e.g. colors, materials, distances

Figure 11: the method implemented by MKBEEM for the processing of NL inputs [Lehtola, et al. 2003]

- As for MIETTA, it is also a multilingual system. However, it avoided the use of full natural language interface and only was only used form filling interface and keywords processing.
- Similarly, TREE avoided the use of full natural language interfaces and used form filling to interact with users in different languages.
- The HappyAssistant prototype used a very limited NL processing for noun phrases only to provide NLDI.
- CASA had also a form filling interaction style with keywords-based processing.
- Finally, GOOGLE SMS uses a very restricted language (close to a command language) to interact with users.

On the document processing side, we have seen that some systems had a processing component for this task. CASA, TREE and MIETTA provided a shallow parsing for the semi-structured documents they processed. MKBEEM used full parsing to process controlled-language documents.

Looking carefully at the above systems, we see that many of their authors realized the importance of having internal representations for more precise processing. As an example, MIETTA and TREE used language-independent templates to store extracted information from documents. On the other hand, MKBEEM used several internal representations for mapping and inferring.

1.3.2 A proposed methodology

Thus, if the free natural language style is the best method for interactions with end users, why is it that most of the above systems avoided implementing it, or failed in delivering it in a robust way? There are different possible reasons:

- All of the above systems are Web-based. Hence, form filling and other graphical user interfaces are viable options, imposing only slightly more constraints on the users than a full NL interface.
- The developers of these systems did not take into account the restricted nature of their systems and the associated sublanguage that can be exploited in building a high quality system without settling for less interesting alternatives.
- Building a "production system" requires to take into consideration many constraints (concurrency, short response time, etc.) that are neglected when building a prototype. Therefore, transforming a prototype into a real system is often unfeasible because it requires major changes that may be impossible to perform.
- The use of inadequate techniques. This was manifested by MKBEEM project which imposed a controlled language on users' inputs, but with inadequate methods and techniques.

In total, we think that using inadequate techniques is the main source of this failure. As an example, using deep syntactic parsing for telegraphic ungrammatical sentences will certainly be unsuccessful. Similarly, using tools and techniques suitable for rigid word order languages will not certainly produce good results if applied on languages with free word order. Another example of inadequate technique is the use of open domain techniques for domain-dependent systems. It is necessary for such systems to take advantage of the narrow scope both linguistically and semantically for such restricted domains.

It is assumed that any applied system will be oriented toward the particular variety of natural language associated with a single knowledge domain. This follows from the now widely accepted fact that such systems require rather tight, primarily semantic, constraints to obtain a correct analysis, and that such constraints can at present be stated only for sublanguages, not for a whole natural language [Kittredge: 1982]. In that sense, incorporating the accurate linguistic description of a sublanguage into a natural language system will definitely increase the accuracy and robustness of processing.

On the other hand, knowledge representations and content-oriented methods are necessary for building accurate NL-based transactional systems such as e-commerce systems, because they provide the necessary mechanisms for normalization, unification, transformation, abstraction and compensation of information that exist in human language processing.

Therefore, our thesis will be that it is necessary and possible to build (multilingual) NL-based e-commerce systems for limited domains with mixed sublanguage and content-oriented methods.

1.3.3 Introduction of an experimental system

Having those issues in mind, we built an experimental system as a proof-of-concept. The system is a SMS-based classified ads selling and buying platform. It allows users to send classified ads of the articles/goods they would like to sell and to search for the goods/articles they desire using a full natural language interface. The system extracts content from both "sell" and "looking for" posts and transform the natural language text into a corresponding content representation. For a "sell" post, the content representation is mapped into database record and stored into RDMS. For a "looking for" type of posts, the content representation is used to build a SQL query to retrieve information from the data that has previously been processed and stored in the RDMS. In case of no match in a "looking for" post, relaxed match is attempted, and the exact query is remembered and applied again (until a certain time limit) when a new "sell" posts are received.

The current version of this system is in Arabic and is restricted to the Cars and Real Estate sub-domains. However, the system is designed with a view of easy porting into other domains and other languages.

Chapter 2. Requirements, approaches and sublanguage specifications

Introduction

The main objective of building CATS (classified ads transaction system) is to show that it is possible to develop an e-commerce NL-based system by using a methodology that combines sublanguage processing and higher knowledge and content representation.

Developing natural language based systems that handle spontaneous and unprocessed text requires techniques and approaches different from that used in the case of clean or preprocessed texts. The main challenge of such systems is the lack of data. The typology of the posts sent to the system affects the methodology and the techniques to be employed. As an example, the techniques used for extracting information from structured or semi-structured text is different from those used on free text.

Most of the current research focus is on web-based text which is available in large quantities. By contrast, to develop our experimental system, which is unique in using SMS text for interaction, we were faced with data scarcity.

In this chapter, we will focus on the requirements, specifications and resources needed to build our experimental system. In the first part of this chapter, we will specify the features and constraints of CATS. In the second part, we will describe the process of collecting corpora necessary to build an adequate lingware. In the third part, we will describe the sublanguage used, and justify the need of knowledge representation.

2.1 Overall requirements and constraints

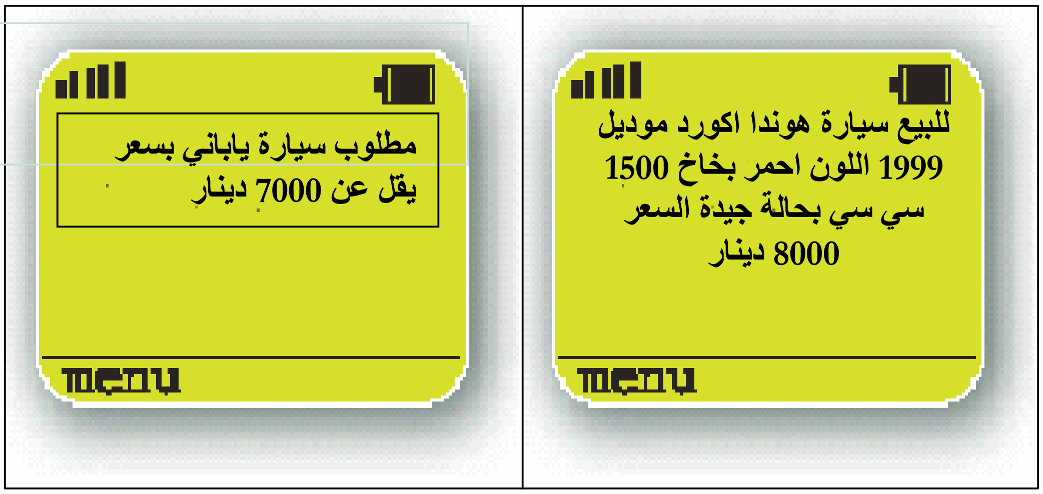
In this section, we will discuss some of the non-functional parameters that affect the implementation of CATS, such as the channel of communication, the domain, and other types of constraints.

2.1.1 Features of the CATS system

CATS is an experimental SMS-based selling and buying platform. However, there will not be any *monetary* transactions directly: CATS will help the subscribers to sell and buy their goods even while moving.

CATS offers a natural language interface allowing the user to express his/her demand by writing a sentence in Arabic and sending it as a SMS message to a short number assigned by the mobile operator.

With CATS, users can submit selling posts such as "for sale Honda Accord 1999, red, injection 1500 cc, in good condition, the price is 8000 JD." If a person is looking for a car, he/she can make queries like "I need a Japanese car with a price less than 7000 JD."


مطلوب سيارة ياباني بسعر
يقل عن 7000 دينار

للبيع سيارة هوندا اكورد موديل
1999 اللون احمر بخاخ 1500
سي سي بحالة جيدة السعر
8000 دينار

Figure 12: examples of "sell" and "looking for" posts

One aspect of CATS is that it must solve an Information Extraction (IE) problem, because it must transform the classified ads posted into corresponding knowledge representations. This is not completely similar to the filling of templates in traditional IE, because we are transforming the natural language representation into a more abstract representation that captures the meaning of the post, and is completely language-independent.

Additionally, one can relate CATS to the QA problem, since both aims at generating answers in response to users' natural language queries.

According to many researchers, a QA system is a special type of search engine which tries to pinpoint answers from documents. That necessitates a specialized form of information retrieval technology. However, CATS differs from QA systems, because of the following reasons:

1) **Type of the question**. QA systems, most questions handled ask for factoids such as:

- What is the Capital of Jordan?
- Who is the current president of France?

On the other hand, CATS handles different types of questions which are quite similar to list questions.

2) **Source of answer**. Usually, QA systems extract answers from large document collections existing in the web. By contrast, CATS formulates responses from the "sell" posts previously submitted by users.

3) **Domain**. CATS handles classified ads only, hence it is a restricted domain system. The text processed by CATS has a limited vocabulary and syntax. On the other hand, most of the research presented in the TREC focuses on open-domain QA techniques, which are not adequate for restricted–domain systems. Furthermore, CATS handles text with a special terminology and complex queries and questions that involve comparatives.

4) **Validity of answers**. As an example the following search request "I am looking for a Honda Civic above year 2000" can be rewritten as: "What are the contacts numbers of the owners of

Honda Civic cars which are available for sale". This is very similar to a list question. However, CATS differs in how it generates the responses. To demonstrate this, the answer to the above question might vary from time to time depending on the traffic to the systems (time-oriented, time-varying response). While the answer to "list all countries that has borders with Jordan" is the same, whether it is today or after one year. A traditional QA-system receives queries and searches for answers in documents that do not change often. On the other hand, CATS receives both types of posts and generates responses from the "sell" posts that it has previously received and processed.

Another aspect of the problem is reasoning, which is important for handling different types of requests. For example, there are questions that involve generalization, such as "looking for a Korean car" or "looking for small car".

Reasoning is also important in handling "no answer situations". For example, "looking for a Clio year 99". It is an option to relax the constraints in the initial query if no answer is found.

Additionally, CATS has a real-time constraint: it must provide answers in a very short time. To enhance the performance of the system, the "sell" posts should be processed, transformed and stored in advance. The usage of a RDBS is very important in providing a very robust and stable engineering solution.

2.1.2 The SMS as the chosen medium of communication

2.1.2.1 What is a SMS?

The SMS, **or Short Message Service,** is a data transmission technology which was created as part of the Global System for Mobiles communications (GSM), the digital mobile phone standard. SMS lets you send and receive text messages to and from a mobile phone. The text can include words or numbers, or a combination of the two. The SMS was introduced as a simple voicemail notification service and has evolved to become one of the most successful media for peer-to-peer communication and value added-services. The maximum data capacity of an SMS message is 1120 bits. Therefore, for the Latin alphabet, one can contain up to 160 characters. For the Arab alphabet, it can contain up to 70 characters only[3].

The SMS is becoming one of the most popular channels for exchanging information. The most important factor that explains this enormous success is its simple, immediate, and confidential way to communicate. Moreover, it has played a major role in narrowing down the digital gap caused by the low level of Internet penetration in some countries. As an example, SMS enables communication between more than 2.2 million Jordanian subscribers anywhere, anytime, and hence offers unmatched service coverage, beyond even that of the Internet, as mobile phone penetration is much higher than Internet penetration.

2.1.2.2 Why CATS is SMS-based?

The first reason to make CATS SMS-based is that it is not possible to interact with the system using this medium by a menu driven interface, or by form filling. Therefore, the only practical interaction style is spontaneous natural language text: the user should be allowed to post a complete sentence, without any restrictions on the used language. More to the point, natural language dialogue interfaces are inadequate and time consuming, since users need to send several messages to accomplish the task. On the contrary, a human language allows a wide range of expressions. It

[3] A 16-bit code, or Unicode (UCS2), is used for Greek and Arabic alphabets. The number of characters is limited to 70 (16*70 = 1120 bits).

helps the user specify many parameters as needed in a single request, in natural language, thus avoiding the tiresome aspect of dialogue-based systems.

Thus, the above reasons justify our decision to select SMS as the medium of communication for this fully NL-based system. In other words, a full NL-based interface is the only interaction method that is practical and possible in SMS.

The second reason is the popularity of the SMS medium, as people from different backgrounds can interact with CATS.

The final reason is the mobility advantage of SMS: people can interact with the system anywhere and anytime. That encourages people to use this service.

2.1.3 The Classified Ads Domain

In choosing classified ads as an experimental domain for CATS, we were motivated by different factors. Some are related to the objective of this experimental system and others are motivated by customer needs.

2.1.3.1 Reasons related to the objectives of the experiment

Since our objective is to build an e-commerce system, classified ads is an ideal domain for this experiment. In addition to the compatibility of e-commerce definition, the domain is well known by users, hence there is no need whatsoever for training them.

Additionally, we don't have to put investment in collecting the contents, as users are the source and destination of information. This strengthen the arguments for building this experimental system further, since unlike other systems, CATS processes both type of spontaneous natural language posts: "looking for" and "sell".

Furthermore, the classified ads domain is associated with a family of sublanguages, one for each domain (Cars, Real Estate, jobs...). To find ways to specify that family and the "parameters" of each sublanguage belonging to it is an interesting problem that can be solved only by higher abstraction of knowledge and the study of the sublanguages used.

2.1.3.2 Reasons related to making CATS successful

From the users' point of view, this domain is of high interest. Currently, most of the classified ads are published weekly through printed materials. You have to read thousands of classified ads to get what you are looking for. Looking at the web for ads, we find that the most popular publisher of one of the classified ads circulation has a website (www.ewaseet.com) that is updated weekly, without any searching capabilities. Hence, it is difficult to find what one is looking for. Furthermore, we discovered that most of the listed classified ads in the printed materials are posted by brokers and not by ordinary people. Hence, we thought of providing buyers and sellers with a more credible and transparent channel of communication, able of connecting them efficiently.

We picked as a beginning the Cars and Real Estate domains as both of them are very popular in Jordan and constitute the major shares of the classified ads industry.

2.1.4 Other constraints/ operating environment and dependencies

2.1.4.1 Integration with the mobile operator

To provide such a service, we have to partner with a Mobile Operator. The experimental system should comply with communication protocols used by the mobile operator. In turn, the operator provides the network access and assigns a number or more for accessing this service.

2.1.4.2 Using one short number (access point) to all services

We preferred the one short number setup. Assigning one short number implies that users have one access point to this service. It also means that the platform has to distinguish between "sell" and "looking for" posts for each domain. This setup is more interesting and straightforward to users as they have to remember one short number, rather than several one for each type of posts and for each domain. However, the One Short Number solution is harder and more challenging to implement.

2.1.4.3 Future expandability of the system

We took into consideration in the design of CATS its future expandability. The system should be able to accommodate with the minimum resources other sub-domains such as job announcement, matrimonial and home appliances, etc. Furthermore, it is also important to port this platform to other languages, in order to allow people to submit "sell" posts in any language, and also to query and search in any language.

2.2 Collection of corpora for development

2.2.1 Crucial role of corpora in development

A corpus-based approach will certainly lead to a better understanding of the sublanguage used and the way people encode their thoughts in this domain. In turn, this will help is selecting the right approach for development. As an example, systems developed for semi-structured text are not appropriate for free text and vice-versa. The assumption that SMS-based classified ads are semi-structured or free text needs to be verified. Developing information systems that depend on natural, spontaneous and unprocessed text requires techniques and approaches different from those used for edited text. Most of the current systems that process users queries and generate responses use shallow text processing techniques based on pattern extraction or information retrieval techniques [Benamara 2004]. However, systems such as CATS require deeper text understanding methods [Moldovan, et al. 2003].

2.2.1.1 Development methodology

The study of the corpus aims at identifying classes of objects and their hierarchal inclusions, the properties of these classes, the relations among them, and their lexico-semantic patterns.

We can distinguish between 4 phases in a corpus-based development:

- Design phase: in this phase, we study the corpus with the aim of assigning semantic classes, specifying most frequent words, and depict the lexicon, styles and types of queries that interest users. We also made decisions on what is relevant and what is not relevant to a particular domain. The two outputs of this phase are the design of the knowledge representation and the design of the dictionary.

- Basic implementation phase: in this phase, we build the basic NL system which consists of the extraction rules and the dictionary. Lexical items are added to the dictionary based on most frequent words. For the encoding of the rules, we use iterative procedures. We manually extract a first set of relevant patterns of the domain. These patterns are then encoded into extraction rules that are applied on the corpus. The coverage of the rules is increasingly expanded until good performance is achieved on the corpus.
- Experimental deployment phase: in this phase, we put the system into full operation, but for testing purposes. Each processed post is evaluated manually. Accordingly, corrective/updating measures are taken in the rules and/or the dictionary. When the number of maintenance tasks becomes smaller and smaller, we move to the full deployment phase.
- Full deployment phase: in this phase, the system is fully operational. Maintenance tasks are based on users' feedback and internal quality assurance procedures.

2.2.1.2 Type and size of corpora needed at various times

The design phase is crucial for the success or failure of such a system. Collecting irrelevant corpus or the wrong interpretation will lead to wrong design of the KR and the dictionary.

Hence, the corpus used in the design phase should be highly relevant to the domain. Having this in mind, building a system that uses a spontaneous input requires analysing a corpus composed of spontaneous texts.

As a minimum requirement, 200 messages are needed for each subdomain: Cars and Real Estate. This size of corpus is enough to cover also the needs of the basic implementation phase.

For the experimental deployment phase, the system was tested with another 400 posts for each domain.

During the first week of the full deployment phase, we performed many maintenance tasks. The modular design of CATS helped us in performing these tasks without interruption of the service. After 500 posts for each domain, the system became stable, and the maintenance task diminishes sharply especially for the rules[4].

To demonstrate this, consider the following message received in the first day full deployment:

4/16/2005 12:39:09 PM   Message In : +962796971018,**مطلوب أرض للاسكان في ع غ من المالك**

"Looking for a residential land in WA from owner"

In this message the poster used a non-standard abbreviation "**ع غ**" to denote West Amman which is in the dictionary. We took two actions to correct this situation: a new entry "**ع غ**" was added to the dictionary and the same query was issued again on behalf of the user to generate the correct list. It is worth mentioning that we sometimes anticipate new entries based on discovered facts. Hence, we did not add only "**ع غ**", but also "**ع ش**", to denote the east part of Amman.

[4] The size of the rules and dictionary will be discussed in the following chapters.

2.2.1.3 Scarcity of data

The shortage of data is one of the main obstacles in developing natural language systems. It is not easy to collect corpora for restricted domain, especially if they must come from a very private medium of communication such as SMS. A very interesting aspect of CATS is the study of its natural "sublanguages". That provides an unprecedented opportunity to analyze the SMS-based restricted domain sublanguages Therefore, the dataset collected from CATS is unique and very important. Firstly, it contains real, spontaneous, and unedited text. Secondly, it is written by thousands of authors from a diversity of backgrounds. Thirdly, it covers different domains and topics.

2.2.2 Finding a suitable corpus to initialize development

2.2.2.1 Hypothesis on the future SMS corpus

We could not find any references that discuss the features of Arabic SMS messages in any domain. Additionally, mobile operators refused to provide us with any excerpt of real SMS messages, to maintain the privacy of their customers.

Table 9: some examples from the web-based corpus

مطلوب باص هونداي موديل 99ماتور 3000 او موديل 2000 تيربو كامل الاضافات وعلى الفحص الكامل بدفعه الفين دينار والباقي تقسيط .	Looking for a Hyundai bus 99 motor 3000 or year 2000 turbo full option pass full check down payment 2000 dinar rest by installments
مطلوب سيارة متسوبيشي لانسر 95 فما فوق بحالة جيدة وسعر مناسب ولا يشترط الفحص الكامل....يفضل من المالك (للاتصال بعد 2 ظهرا).	Looking for Mitsubishi Lancer 95 and above in good condition and in good price not important to pass the check preferable from the owner to call after 2 pm
مطلوب سيارة أوبل 90 فما فوق أو جولف نفخ 92-94 بحالة جيدة جدا وبسعر مناسب..... للاتصال بعد 2 ظهرا (يفضل من المالك).	Looking for an Opel car 92 and above or Golf blow 92-94 in good condition and good price call after 2 pm preferable from the owner
سياره مرسيدس موديل 94 c180 فل اوبشن جير اوتوماتيك فتحة سنتر ريموت لون شامبين	A Mercedes car year 99 c180 full option automatic transmission roof center remote color Champaign.
للبيع سيارة نيسان لوريل 1984 بحالة جيدة جداً كاملة الاضافات عدا الجير بخاخ cc2400 السعر 2400 دينار قابل للتفاوض مع امكانية المبادله على سيارة أخرى ألماني أو ياباني.	For sale a car Nissan Laurel 1984 in good condition full option except the transmission injection 2400 cc the price is 2400 dinar negotiable with the possibility to exchange with other German or Japanese car.
سياره غواصه بحاله ممتازه موديل 2002للبيع بسعر مغري جدا للجادين فقط الاتصال السعر المطلوب 56000	A Gawwasah[5] car in excellent condition year 2002 for sale in good price for serious people only to call the price is 56000.

[5] a special term for Mercedes 500

مطلوب شقة 3 نوم بمنافعها في عمان الغربية بمساحةلا تقل عن150 م2 بسعر لا يتجاوز 35000 شاملا الرسوم. و يفضل أن تكون بالقرب من الجامعة الأردنية.	Looking for an apartment with 3 bedrooms with all utilities in West Amman with an area not less than 150 m^2 and a price not more than 35000 including tax preferable to be near the Jordanian University.
مطلوب قطعة ارض مساحتها من500الى 700 في طاب كراع	Looking for a piece of land its area from 500 to 700 in Tab Kra3
مطلوب شراء شقة بمساحة فوق 180م طابق أرضي في منطقة الشميساني ، جبل عمان ، الرابية ، أم أذينة ، دير غبار في عمارة جديدة أو قديمة نوعاما ويفضل من المالك مباشرة على أن لايتجاوز سعرها 40-45دينار .	Looking to buy an apartment with an area above 180 m ground floor in Shmeisany area, Amman Mountain, Rabiah, Om Ozaynah, Der Ghbar in somehow old or new building preferable from the owner directly on condition that the price should not be more than 40-45 dinar.
مطلوب شقة للإيجار 2 نوم و صالة و حمام و مطبخ في مناطق المدينة الرياضية أو عرجان / شارع الأقصى / ضاحية الأمير حسن . بإيجار لا يتجاوز 120 دينار شهري	looking for an apartment to rent 2 bedrooms and Saloon and bathroom and kitchen in areas of Sport City, 3arajan/Aqsa street/ Prince Hasan Suburb with a rent value not more than 120 dinar monthly.
أرض زراعية سكنية - خمسة دونمات تقريباً - ناعور قرب دار الدواء للبيع بمبلغ 60 ألف دينار نهائي.	An agricultural residential land five nearly five donems Naoor near Dar Addwaa for sale with 60 thousands dinar nonnegotiable.
جزء من فيلا طابق اول 3 غرف نوم فرش ممتاز مدخل مستقل وجميع الخدمات متوفرة للايجار للسكن العائلي فقط وبدون وسطاء ، خلدا -ام السماق .	A part of a Villa first floor 3 bedrooms good furniture a separate entrance all service are available for rent to families only and no brokers, Khalda.

Hence, the only feature we knew at the beginning of our development was that the length of an SMS in Arabic would be smaller than 70 characters. Even this assumption is not 100% correct, since users can use the option of long SMS which enables them to encode longer messages.

2.2.2.2 *Choice of a web-based e-commerce corpus*

[Sekine 1997] found in his experiment with different domains, that the best parsing performance was obtained for the same domain (religion, romance and love stories, etc.), followed by same class (fiction or non-fiction), and the worst is on domains within a different class.

In selecting a similar corpus, the main condition to consider is the spontaneous and unedited nature of the text. Therefore, texts from printed material were excluded. The only possibility we had was to look for a web site providing unedited Arabic classified ads services. Fortunately, we found a Jordanian one (http://www.almumtaz.com) that provides this service in Arabic for the Cars and Real Estate domains.

2.2.2.3 *Collection of the initial corpus*

This site enables Internet users to post their ads by typing them in a specified field within a given form. However, the site lacks any search capability and the data are not found in the right places. Hence, we have manually extracted the most relevant posts that we though would be similar to the

SMS posts. Therefore, we collected around 200 posts for each domain evenly distributed to cover "looking for" and "sell" types. In selecting those posts, we used the following guidelines:

- SMS users are aware of the message size constraint, hence we assume that they will avoid typing unnecessary expressions. In accordance with that, we focused on shorter posts.
- We looked for posts with variant structures to enrich our system.
- We also selected posts with errors, typos and different spelling variations.
- Additionally, we focused on posts that contain specific domain terminology.

The choice of a Web-based corpus for the design phase proved to be adequate. The initial system was sufficient to design and build our experimental system.

2.2.3 Phased-based approach for collecting further corpora

2.2.3.1 Collection of corpora for studying sublanguages

The experiments conducted by [Sekine 1997] clearly concluded that parsing a domain-specific text requires a grammar that suits that domain. Domain-specific grammars can produce better results than domain-independent grammars. From a practical perspective, corpora are the sources of these domain-dependent grammars. Sekine also concluded that the size of the corpus does not need to be large to extract these domain specific-grammars, as parsing performance is saturated with a very small-size training corpus.

Additionally, in order to build the right grammar the corpus should be representative for the medium and should be homogenous.

In our case, the same observation applies. It was not very long to extract the domain specific grammar that describing this sublanguage manually.

2.2.3.2 Collection of corpora for maintenance & coverage increase

The maintenance phase requires a corpus that should match the real working environment to fine tune the performance of the system. Basically, two types of maintenance are needed: corrective and adaptive. The corrective maintenance task is usually triggered by detecting a wrong response. Consequently, the taken actions are influenced by the predefined tolerance margins. Hence, a message sent in English can only be handled manually and requires no maintenance task. On the other hand, correct users' inputs that produce wrong results are studied carefully and the maintenance tasks are specified based on that.

The adaptive maintenance is used to enrich the system with new facts such as new cars models or new residential area.

2.2.3.3 Collection of corpora for developing new domains

To develop other domains, it is necessary to have corpora for other domains. We can follow the same procedure as that of Cars and Real Estate. However, we can also use the current corpus collected from CATS to anticipate or project the new domains. This corpus provides us with valuable information on how people encode their thoughts using the SMS medium. As an example the corpus we have, we know how people express numerical values, how they express "search requests", and the general structure of a message (length, word ordering, etc.). This information can be used in developing new domains effectively and shorten the development time dramatically (see chapter 8).

2.3 Typology of SMS-based task-oriented sublanguages

A sublanguage is the language used by a group of people in a restricted domain. It is generally perceived as subject oriented and this is reflected in the lexicon and semantic features.

To measure the lexical complexity of SMS-based classified ads sublanguage we used the type-token ratio (TTR). The higher this ratio, the more lexically complex the text, rich, and, on the contrary, the lower the ratio, the more words repeat themselves and the lower lexical complexity. We calculated the TTR for different corpora for the sake of comparison.

We measure the language complexity by the length of the sentence in words, and we compare between different domain and between SMS-based and Web-based ones. Finally, finding the words frequency in a corpus identifies the nature of text (telegraphic or normal), in particular the less the percentage of function words in a corpus the more it has a fragmentary style.

The analysis of the sublanguage also includes the manual study for lexico-semantic patterns found in within the posts. Our objective is extracting classes of objects that specify the domain knowledge described by the sublanguage.

To perform this analysis, we collected a corpus of the Cars domain, Real Estate domain, Job announcement, and SMS open domain. We also collected a web-based corpus for both Cars and Real Estate domains for the sake of comparison.

2.3.1 General corpus statistics

The SMS-based corpus consists of posts from Cars and Real Estate domains collected during a limited period of CATS operation. We also collected SMS based Job announcements sent to another mobile short number connected to a publisher of a printed circulation interested in jobs domain. The open domain sentences are the posts that are received by CATS or by the Job announcement short number and are not related to any mentioned domains or any related ones.

Table 10: examined SMS based Corpus

Domain	Number of sentences	Sentence average length (words)[6]	types[7]	Tokens	TTR (type token ratio)
Cars	771	9	1181	5875	.201
Real Estate	641	12.5	1441	6182	.233
Jobs	174	14.61	921	2452	.375
Open	231	6.42	1099	1538	.714

[6] Arabic numerals are considered in finding the sentence average length.

[7] Arabic numerals are not considered in calculating "types" or "tokens", that's why when you divide the number of tokens by the number of sentences you will not get the same sentence average length listed in the table.

Table 11: examined Web based Corpus

Domain	Number of sentences	Sentence average length (words)	types	Tokens	TTR (type token ratio)
Cars	537	25	1246	8168	.152
Real Estate	510	28.49	1484	9003	.165

As shown in table 10, the length of sentences in the Cars domain is less than that of the Real Estate domain, compared to 7.3 words for TREC questions. We also found that the length of sentence in the Job domain is the highest. These findings suggest that the language complexity, as measured by sentence length, is higher in Jobs domain than in Real Estate and Cars domain. In other words, the user needs a lesser amount of words to encode his thoughts in the Cars domain than that of the Real Estate domain or the Job domain. We also find the average length of SMS-based open domain sentences is only 6.42 words, making it the smallest among all other domains. When we compare SMS-based posts with Web-based post we find that the SMS based is generally smaller than that of Web-based. This might be explained by the fact that the SMS medium imposes a length and cost constraint on the advertisement. On the other hand, the Web poster has no strict constraint, encouraged to making their posts arbitrarily long and therefore can include more and more irrelevant information.

We also calculated the type-token ratio (TTR), the ratio of the number of word types to the number of word tokens in the text, as a measure of range of vocabulary used. That is, the greater the range of possible referents, the greater the number of word types which will figure in a text. For a given number of tokens a text tending toward greater explicitness will contain a higher number of types [Bruthiaux 1994].

As shown in table 10 the least TTR value was for Cars at 0.201, then for Real Estate at 0.233, then for Jobs at 0.375. Suggesting that the vocabulary is more limited in the Cars domain than in Real estate one, than in Jobs announcement. We also find that the TTR of the SMS-open domain is very high (0.714), suggesting a lexically complex and rich text. Indicating also, that posters were not focusing narrowly on particular topic of discussion in which the same words were not repeated often. These TTR results confirm the finding found in [Bruthiaux 1994] for the same domains.

The TTR values of Web-based post were even lower compared to SMS based ones, suggesting a higher lexical complexity and diversity in the SMS based text.

The TTR of general Arabic corpus of nearly the same text length (number of tokens) is 0.539 as calculated in [Goweder and De Roeck 2001]. Comparing this result with TTRs found for the Cars, Real Estate, and Jobs posts confirms the narrow scope and limited vocabulary characteristics of SMS based restricted domains. On the other hand, the SMS based open domain TTR is higher than that of [Goweder and De Roeck 2001], suggesting a more topical diversity.

Table 12: comparison of SMS based TTRs with the findings in [Goweder and De Roeck 2001]

Domain	Types	Tokens	TTR (type token ratio)	TTR calculated in [Goweder and De Roeck 2001] for the same number of tokens (text length)
Cars	1181	5875	.201	0.539
Real Estate	1441	6182	.233	0.539
Jobs	921	2452	.375	0.579
open	1099	1538	.714	0.618

2.3.2 Word frequencies

The top 50 most frequent used words percentage in SMS-based Cars, Real Estate, Jobs and open domains are 53.77%, 45.76%, 43.27 and 22.37% respectively .

Table 13: top most frequent words in SMS concerning the studied domains

Open domain			Jobs domain			Cars domain			Real Estate domain		
%		Word	%		word	%		word	%		word
1.50	from	من	4.77	in	في	5.91	wanted	مطلوب	5.87	In	في
1.11	if possible	ممكن	2.94	young man	شاب	5.86	car	سياره	4.69	wanted	مطلوب
0.98	on	على	2.12	for	عن	4.82	year	موديل	3.15	for sale	للبيع
0.91	to	الى	1.92	job	عمل	4.03	for sale	للبيع	2.35	apartment	شقه
0.91	I am	انا	1.88	experience	خبره	3.13	car	سيارة	1.96	land	ارض
0.91	in	في	1.75	from	من	1.70	Dinar	دينار	1.57	apartment	شقة
0.72	know	اتعرف	1.51	on	على	1.45	check	فحص	1.46	Dinar	دينار
0.72	ya	يا	1.51	phone	هاتف	1.34	full	كامل	1.29	Amman	عمان
0.65	on	علي	1.22	graduate	خريج	1.31	full	فل	1.12	than	عن
0.59	I am	أنا	1.14	looking	يبحث	1.29	Mercedes	مرسيدس	1.08	For a price	بسعر
0.59	abu	ابو	1.10	with a grade	بتقدير	1.21	color	لون	1.04	not	لا
0.59	to	الي	1.06	good	جيد	1.02	for a price	بسعر	0.95	for rent	للايجار
0.59	like	حاب	0.98	diploma	دبلوم	0.95	above	فوق	0.94	meter	م
0.52	about	عن	0.98	young girl	فتاه	0.94	Kia	كيا	0.92	from	من
0.46	know	أتعرف	0.94	and	و	0.92	year	م	0.91	or	او
0.46	duleimi	الدليمي	0.82	or	او	0.77	and	فما	0.86	land	أرض
0.46	like	حابه	0.82	university	جامعة	0.71	the price	السعر	0.82	house	بيت
0.46	morning	صباح	0.77	graduate(f)	خريجة	0.71	in condition	بحاله	0.73	thousand	الف
0.46	every	كل	0.77	year	سنه	0.71	Hyundai	هونداي	0.70	villa	فيلا
0.39	I would	ارجو	0.77	years	سنوات	0.70	options	الاضافات	0.70	bedroom	نوم
0.39	Jordan	الاردن	0.73	the job	العمل	0.70	Daewoo	دايو	0.65	on	على
0.39	I want	بدي	0.73	tel	ت	0.68	except	عدا	0.57	more	يزيد
0.39	greeting	تحيه	0.73	holding	حاصل	0.65	bus	باص	0.55	donom	دونم
0.39	congratulations	مبروك	0.73	experience	خبرة	0.63	gear	الجير	0.55	area	مساحة
0.33	to	ألى	0.73	field	مجال	0.63	buying	شراء	0.53	meter	متر
0.33	I love you	احبك	0.61	Bachelor	بكالوريوس	0.61	Honda	هوندا	0.50	marj	مرج
0.33	Allah	الله	0.57	young girl	فتاة	0.56	or	او	0.49	piece	قطعة
0.33	girl	بنت	0.53	company	شركه	0.54	Opel	اوبل	0.49	area	مساحه
0.33	Mohammad	محمد	0.49	any	اي	0.51		ما	0.49	with	مع
0.33	hi	مرحبا	0.49	specialty	تخصص	0.49	good	جيده	0.47	alhamam	الحمام

0.33	hi	هلا	0.49	accounting	محاسبه	0.49	payment	دفعه	0.45	al3ali	العلي
0.26	thousand	ألف	0.45	looking	تبحث	0.46	without	بدون	0.42	with area	بمساحة
0.26	sweeter	احلى	0.41	science	علوم	0.46	B	بي	0.40	the owner	المالك
0.26	good	الخير	0.41	engineer	مهندس	0.46	Toyota	تويوتا	0.40	street	شارع
0.26	Iraq	العراق	0.41	job	وظيفه	0.46	wanted	طلب	0.39	tela3	تلاع
0.26	doing	العمل	0.37	part time	بدوام	0.44	power	بور	0.39	jabal	جبل
0.26	present	اهداء	0.37	bachelor	بكالوريس	0.44	above	عن	0.37	saloon	صالون
0.26	I love you	بحبك	0.37	part	جزئي	0.44	Nissan	نيسان	0.37	house	منزل
0.26	good	بخير	0.37	holding	حاصله	0.43	BMW	bmw	0.36	thousand	ألف
0.26	hi	تحية	0.33	or	أو	0.41	Golf	جولف	0.36	west	الغربيه
0.26	how are you	حالك	0.33	looking	ابحث	0.39	installments	اقساط	0.36	3arajan	عرجان
0.26	Hussien	حسين	0.33	university	جامعه	0.39	with payment	بدفعه	0.36	near	قرب
0.26	number	رقم	0.33	computer	حاسوب	0.39	pick up	بكب	0.36	piece	قطعه
0.26	tribes	عشائر	0.33	year	سنة	0.39	Lancer	لانسر	0.36	area	منطقة
0.26	who	مين	0.33	company	شركة	0.39	from	من	0.36	and	و
0.20	with	مع	0.33	certificate	شهادة	0.37	Peugeot	بيجو	0.34	or	أو
0.20	with you	معكم	0.33	I have	لدى	0.37	not	لا	0.34	the price	السعر
0.20	if possible	أذا	0.33	looking	يطلب	0.37	and	و	0.34	west	الغربية
0.20	to	أن	0.29	icdl	icdl	0.36	or	أو	0.34	floor	طابق
0.20	Ahmad	احمد	0.29	ready	استعداد	0.34	new	جديد	0.34	shop	محل
22.37			43.27			53.77			45.76		**total**

Table 13 shows that as we move from Cars to Real Estate, to Jobs and finally to Open domain, the percentage of function words (such as preposition) increases. This finding can be correlated with the TTR of each sub-domain, indicating a less telegraphic text as we move from the Cars domain to the open domain.

2.3.3 Lexical characteristics

The low TTR values of SMS-based Cars or Real Estate domains compared to open domain confirms the fact that the most obvious feature of a classified ads sublanguage is its limited and specialized lexicon.

2.3.3.1 Terminology

We observe that some words in the Cars and Real Estate domain can have different meanings than in the open domain. Therefore, specialized dictionaries are required to process the text. For example, in the Cars domain *'duck'* denotes a Mercedes model, and a *'piece'* in the Real Estate domain means a land.

Multi-word concepts and terms are also very frequent to the extent that they appear in the topmost frequent words list. As an example:

"For sale"
"For rent"
"Looking for"
"Piece of land"
"Not more than"
"Full option"
"Except the transmission"

They take a special meaning as if they were a single word. Whether to treat them as a single words or not is issue debated among researchers. However, since they are very discriminating in restricted domains, handling them as single word concept is very useful to improve the precisions of such systems [Doan Nguyen and Kosseim 2004].

2.3.3.2 Abbreviations

The use of abbreviations is not frequent, and there is no common standard or agreement among posters. As an example “م” denotes *area*, *meter* (Real estate domain) or *year* (Cars domain). In the same context, most posters do not abbreviate “West Amman”. However when some of them do, we get several abbreviations: “ عمان غ“, “ ع غ”, “ غ ع” , “ غ عمان”.

2.3.3.3 Named entities

In the Cars domain, named entities are references to Car Makes and Models. In the Real Estate domain, they are references to Locations. For the Open domain, this definition might be extended to include other patterns such as dates, times, number expressions, dollar amounts, email addresses and web addresses.

Correct recognition of named entities is necessary in raising the quality of the processing. As an example, it is impossible to process “For sale Honda” properly without the recognition of “Honda” as a Car Make.

The study of the corpus of classified ads shows that Named Entities consist of one or more words. We also note that it is possible that each component (or combinations of them) of a multi-word named entity points to a different reference or has another meaning. For example, the multi-word named entity “Biader Wadi AlSir” is a location name referencing a well known area in Amman, but “Wadi AlSir” alone references other area. As Arabic is not like English in distinguishing named entities by capitalizing the first character, and sentences are very short, recognition of named entities is impossible without using lexical lookup.

2.3.3.4 Numerical values

The dataset under study is full of numerical values. In the Car domain, they represent price, year, motor size and sometime models for some car makes. In the Real Estate domain, they represent the price, area, no of bedrooms, etc. The posters encode numerical values differently. Some of them use non-arabic numerals such as “three thousands”. Others use arabic numerals such as “3000”. Finally, some posters combine the two approaches and write expressions such as “3 thousands”.

Usually, numerical values are preceded by hint words such as:

Area
Price
Motor size
year

Some of them are usually followed by unit words such as:

Meter
CC
Dinar

Both hint words and unit words are used by the rules to extract correctly the values. But, it becomes problematic when users fail to write both hints words and unit words, as demonstrated by the post:

“For sale Mercedes 200 1999”

Failing to write hint words or unit words near to numerical values is frequent within the studied corpora.

2.3.3.5 Slang words

The corpus also contains slang words and words from spoken Arabic like “بدي” “I want” . They are not so frequent (5 slang words per 100 SMS messages) in the SMS based classified ads domain as none of these slang words appear in the top 50 most frequent words shown above. Nevertheless, they need to be processed accurately.

2.3.4 Syntactic characteristics

The data that we studied contains many alternative surface structures for the same utterance. We believe this phenomenon reflects the diversity of the posters. It was evident from looking at the posts that there was no unique underlying syntactic structure in the sublanguage used. Some posts consist of fragmented phrases (telegraphic) rather than fully-formed sentences. Other posts are more cohesive and some are full sentences. Obviously, syntax-based parsing based methods would not prove very useful in dealing with the given data. As an example, a traditional parser looking for object and subject will fail in analyzing sentence 3 in table 14. Similarly, techniques used for semi-structured text relying on position, layout and format of text are bound to fail on the given data.

We also observe the variable order nature of the posts as shown in table 14. This is a demonstration of the extremely free ordered nature of the Arabic language in which the constituents are not identified by their positions but by their inflectional endings [Covington 1992].

Table 14: sample of posts in the Cars domain

1	سياره اوبل فكترا للبيع موديل 2003 فل اوبشن	Opel Vectra car for sale year2000 full option
2	للبيع سيارة بي ام دبليو 520 لون زيتي فحص كامل م 89 فل عدا الفتحه مرخصه بحال ممتازه بسعر 8500	For sale BMW 520 color dark green full check year 89 full except sunroof licensed in a good condition with a price 8500.
3	أوبل أسترا ستيشن لون أحمر(بورفتحه سنترزجاج ومريات كهرباء) فحص للبيع.	Opel Astra station color red (power sunroof Center Electrical windows and mirrors check for sale
4	اريدبيع سياره دايو ليمنز موديل 92 فحص كامل فل اوبشن	I want to sell a Daewoo Lemens car year 92 full check full option
5	شراء سيارة.	Buying a car

6	بحاجه لسياره لا تزيد عن 2000 دينار بحاله جيده واقتصاديه في البنزين	In need for a car not more than 2000 dinar in good condition and economical in fuel.
7	عندي سياره لاند روفر بدي ابيعها.	I have a Land Rover car I want to sell it
8	مطلوب سيارة بيجو406	Wanted a Peugeot 406 car
9	لاندروفر ديسكفري 2001 فل 8 سلندر بحاله ممتازه للبيع	Landrover Discovery 2001 full 8 cylinders in excellent condition for sale.
10	سياره للبيع بدفعه 500 هونداي مديل 97 و بالتقسيط	A car for sale Hyundai year 97 by installments.

2.3.5 Semantic characteristics

Many researchers believe that it is possible to group words into equivalence classes and to describe the occurring sentences within a sublanguage in terms of these classes [Kittredge: 1982].

Therefore we can view a classified ads post as sequence of properties restricting the main domain object (i.e. car, apartment). This statement is true for both Real Estate and Cars and for both "sell" and "looking for" posts. This information model is more efficient than relying on syntactic structure for the description of the SMS. For example, sentence number 1 in table 14 can be described as follows:

[vehicle] [make][model][Ads_type][year][feature]

Year = [year_hint] [number]

Likewise, sentence number 6 can be described as:

[Ads_type][vehicle][price][feature][feature]

Price = [not more][number][currency_unit]

It is clear that the above semantic categories depict the underlying subject matter of the domain.

This approach of describing sentences semantically achieves better results than using pure syntactic description. They were introduced as an engineering methodology, which allows semantic knowledge to be easily included in the system [Androutsopoulos, et al. 1995].

2.4 Which representation is the best for such an application?

In this section we will demonstrate the necessity of having a knowledge representation. Additionally, we will specify the features that this KR should have. We will also discuss available knowledge representations that are used nowadays.

2.4.1 Necessity of KR

The syntactic structure for different posts which express the same information can vary enormously. As shown in table 15, both sentences hold the same information:

- Both are of "looking for" type posts
- The main domain object is Land,
- The location is in Khalda,
- The area should be 500 and above.

Table 15: example for syntactically different posts, but semantically the same

1	اريد قطعة ارض في خلدا مساحتها لاتقل عن 500	I am looking for a piece of land in Khalda its area not less than 500
2	مطلوب 500 متر وأكثر في خلدا	Wanted 500 square meter and above in Khalda

Hence, what is required is an additional level of abstraction that represents the underlying meaning of a post.

Formulating correct responses for users' queries is another motivation for defining a unique knowledge representation for both types of posts. Suppose we have the following "sell" post:

للبيع منزل مستقل في خلدا "For sale an independent house in Khalda"

And that a poster sends the following query:

مطلوب فيلا في غرب عمان "Wanted *a villa in the West of Amman"*

Relying only on bag of words for finding answers is insufficient, and of course will lead to totally unacceptable results, since none of the tokens in the "looking for" post match any of those in the "sell" post. This example shows clearly the need to transform both posts into a language-independent structure that captures the meaning. This will enable the system to correctly find matches, because posts with similar meaning will be recognized, regardless of how they are structured grammatically and which particular terms are used.

2.4.2 Required features of a KR for CATS

The *Knowledge Representation Hypothesis:* Any mechanically embodied intelligent process will be comprised of structural ingredients that [Smith 1982]:
a) we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits, and
b) are independent of such external semantical attribution, play a formal but causal and essential role in engendering the behavior that manifests that knowledge.

Based on that, a Knowledge representation, then, must be in a form that is *understandable* by humans, and must cause the system using the knowledge to *behave* as if it knows it.
William Woods [Woods 1975] defines the properties of a KR Language assuming that it must be derived from a natural language:

> A KR language must unambiguously represent any interpretation of a sentence (logical adequacy), have a method for translating from natural language to that representation, and must be usable for reasoning.

[Davis, et al. 1993] specified 5 roles a KR can play:

- A knowledge representation (KR) is most fundamentally a surrogate, a substitute for the thing itself, used to enable an entity to determine consequences by thinking rather than acting, i.e., by reasoning about the world rather than taking action in it.
- It is a set of ontological commitments, i.e., an answer to the question: In what terms should I think about the world?
- It is a fragmentary theory of intelligent reasoning, expressed in terms of three components: (i) the representation's fundamental conception of intelligent reasoning; (ii) the set of inferences the representation sanctions; and (iii) the set of inferences it recommends.
- It is a medium for pragmatically efficient computation, i.e., the computational environment in which thinking is accomplished. One contribution to this pragmatic efficiency is supplied by the guidance a representation provides for organizing information so as to facilitate making the recommended inferences.
- It is a medium of human expression, i.e., a language in which we say things about the world.

In the next section I will discuss briefly different KR used, however in this section I will focus on the study of features needed for this specific experimental system.
The above examples show the knowledge representation should include a mechanism for generalization/specialization, and to present concepts in a hierarchy in which higher level concepts are more general than concepts below. Hence, there exists a "generalization/specialization" relation between entities E1 and E2 (for example E2 is a specialization of E1 and E1 a generalization of E2), if E2 evoke a specificity of E1, or is about specifics themes of E1. For example E1 = West Amman and E2 = Khalda, or E1 is about the French cars in general and E2 treats Renault cars only.

It should also enable the implementation of reasoning, which is the ability to state in a formal way that the existence of some piece of knowledge implies the existence of some other.

For example, when a "sell" ad is received saying "*for sale LANCER 1999*" the system recognizes by referring to the dictionary that it is a car, it is also a Japanese car and the maker is Mitsubishi. Therefore, the system will be capable of detecting and compensating for missing information in both types of messages. As a result, the above post would be one possible source for answering the following query: "*looking for a Japanese car*".

The knowledge representation should also be object-oriented: all the information about a specific concept is stored with that concept. In contrast, information about one concept in rule-based systems may be scattered throughout the rule base. [Motik, et al. 2002] summarizes the importance of the object-oriented paradigm:

> Object-oriented modeling paradigm owes its success largely to the fact that it is highly intuitive. Its constructs match well with the way people think about the domain they are modeling. Object-oriented models can easily be visualized, thus making them easily understandable. Thus, any successful conceptual modeling approach should incorporate the object-oriented paradigm.

In our case, we should model the main domain object along with its attributes. We have seen that this is the prevailing information structure in both types of posts that exists in our corpus. This encapsulation of information will then allow attribute based searching, for example:

"Looking for red Civic"

"Looking for a ground apartment in Amman"

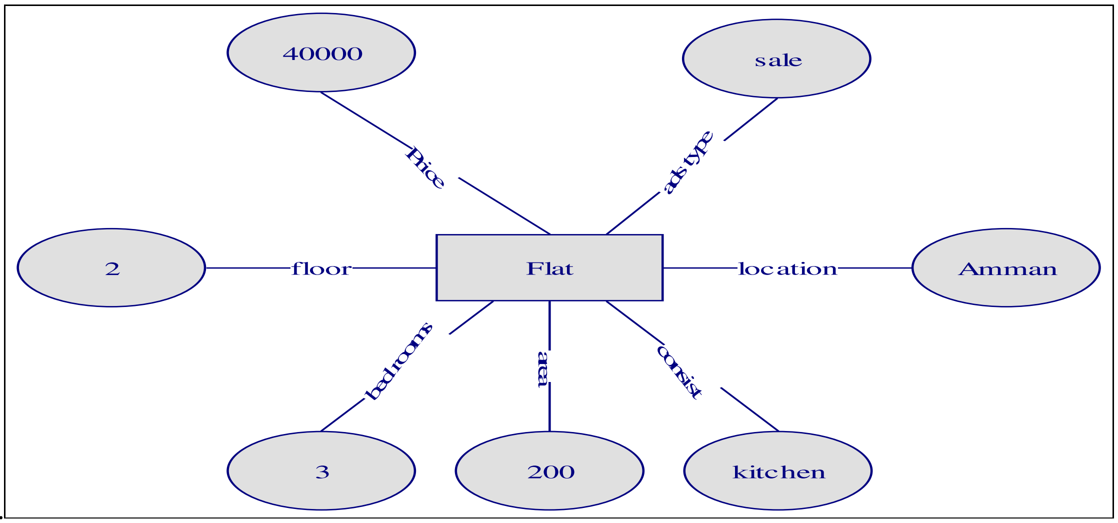


Figure 13: example of object-orientedness

The knowledge Representation should be expandable, that is capable of accommodating other domains which seem to have different or complex information structure. For example, in the following Jobs announcement:

أنا شاب متزوج . حاصل على دبلوم مختبرات طبيه . لدي خبره 4 سنوات في هذا المجال . أرغب بوظيفه بعد الرابعه عصرا إما في مستشفى أو مندوب مستلزمات طبيه	I am a married young man holding a diploma in Medical Labs. I have 4 years experience in this field looking for a job after 4 pm either in a hospital or as a sale-representative for medical goods.

We have two main domain objects: the first object is the person who is seeking employment and the other object is the job itself. The two objects are linked by a "look for" relation. Each one has its designating sets of properties.

Finally, porting this system to other languages requires that the knowledge representation should be language-independent.

2.4.3 Available knowledge representations

Having said that, the main question regarding producing automatically correct responses is what type of knowledge is involved, how and where we should store this knowledge. In our corpus we identified the following types of knowledge:

- General-purpose, factual information (locations names, car makers, car models, etc.).
- Descriptive information like the price of a car, the area of an apartment, model of a car, etc.
- Properties-values knowledge. This knowledge is often associated with properties that define the main domain object in a post, for example a car is characterized by its make, model, year, etc.
- Hierarchical knowledge: such as Honda is a kind of Japanese cars.

- Classification criteria of objects according to specific properties such as retrieving apartments according to their areas.

Different formal representations of meaning are used today such as logic (first order predicate calculus), semantic networks, conceptual dependency diagrams, or frame-based representations [Andrenucci and Sneiders 2005, Brachman and Levesque 2004].

Frames and semantic networks are useful for representing declarative information about collections of related objects/concepts, and in particular where there is a clear *class hierarchy*, and where one wants to use *inheritance* to infer the attributes of objects in subclasses from the attributes of objects in the parent class. On the other hand, for complex inferences, a logic-based representation is more appropriate.

To assess the possibility of using a logic-based representation of knowledge, we studied a restricted domain system (tourism) called WEBCOOP [Benamara 2004], which is a logic-based QA system that integrates knowledge representation and advanced reasoning procedures to generate intelligent and cooperative responses to natural language (NL) queries on the web. The system is written in Prolog and consists of the following components: the knowledge base, ontology, lexicon and indexed text.

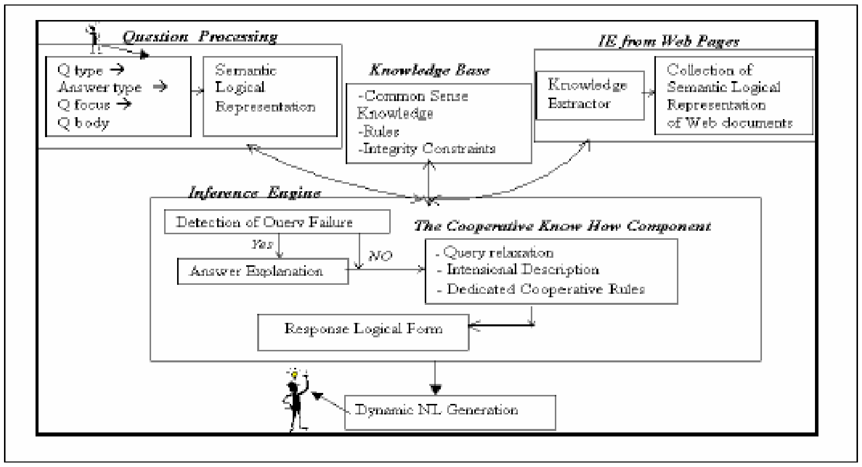


Figure 14: WEBCOOP architecture

For example, the question Give me the Royal Hotel rates in Paris! has the following semantic representation:

(Quantity, X : listof(rates), hotel(royal) ∧ in(place, royal, paris) ∧ rates(royal, X)).

Generating responses is complicated and is based on matching first logic queries with indexed texts. Additionally, it is accomplished by accessing the ontology and the KB of this system.

The study of the WEBCOOP system confirmed the following reasons that make a logic-based KR inadequate for our experimental system:

- Expressiveness of logic-based KR and complex reasoning comes at a cost in efficiency [Klein and Bernstein 2001]. We could not get any indication form WEBCOOP that disproves this claim. On the contrary, with the nonexistence of any type of evaluation, the reported matching procedure confirms this fact.

- Additionally, it can be prohibitively difficult to model and produce the semantics of non-trivial “looking for” queries and “sell” post using formal logic [Klein and Bernstein 2001].

- To provide the best computational medium, it is important to have a more autonomous KR that reflects the object-oriented structure of knowledge found in the SMS posts and is not scattered in different repositories like in WEBCOOP.
- In this thesis, our focus is to build an efficient experimental system "able to become a practical", to prove our hypothesis. Hence, we will adopt a representation that can facilitate this ultimate objective. It is beyond the scope of this thesis to draw a comparison between the best KRs for this application.

Considering these points, a more declarative structure, such as frame-based is more appropriate to our experimental application. No doubt, there exists a connection between frame systems and object-oriented programming (OOP) [Lassila 1990], Basic vocabulary is different, but what the terms denote are approximately the same, see table below [Lassila and McGuinness 2001]:

OOP Systems	**Frame Systems**	**Description Logics**
instance	frame, instance, individual	instance, individual
attribute, instance variable	slot	role
value	filler	filler
class, type	frame, schema	class, concept

This is also confirmed by the adoption of the resource description framework (RDF) as a formal frame-based representation of the semantic web. The RDF basic description model uses object/attribute/value triples: instead of viewing instances of the model as directed, labeled graphs (in this respect they resemble semantic networks), one can take a more object-oriented view and think of RDF as a frame-based representation system [Lassila and McGuinness 2001]. Additionally, the Semantic Computing consortium is using a common description language (CDL) [Uchida 2006] based on UNL (a Semantic Network) to represent knowledge.

A detailed description of our KR will be explained in the next chapter.

Chapter 3. The CATS System

Introduction

We constructed the CATS to experiment, validate and possibly improve our hypothesis.

CATS is not only a lab prototype, but a system in real use. This leads to:

- more valid scientific conclusions.
- some new (unforeseen) problems coming from real operation.

The CATS system is an integration of different technologies: NLP, databases and GSM communication. It is a real time system where high availability and low response time is essential.

In this chapter, we will provide a full description of the architecture of CATS. At the beginning we present the main functionalities, usage scenarios and working environment. Then we describe the general architecture of the system by presenting its major components. Finally, we introduce the KR used by CATS, CRL-CATS, and compare it with UNL.

3.1 Overview

3.1.1 Main functionalities

The CATS system is a C2C based e-commerce system that uses content extraction technology based on sublanguage analysis and knowledge representation to enable SMS users to post and search for classified ads in Arabic. It has two main functionalities: the submission for selling items and the answering of users' queries through interaction in spontaneous natural language. The system receives an entry in full text without a pre-specified layout, recognizes the various relevant entries, and produces a knowledge representation for further processing. We have two types of users' requests:

- "Sell" post: in which the user is a potential seller.
- "Looking for" post: in which the user is a potential buyer.

3.1.2 Usage scenarios

Suppose a user wants to sell his car, he can simply type a SMS message and send it to a specified short number (Figure 15). In the same way, the user can express his search request and send it to the same number. As shown in Figure 16, the system is capable of searching for exact values (e.g., Japanese) and specified ranges (e.g., the price should be less than 2500 JD). If the system finds records that match his request, the user will get a list of cars with contacts numbers (Figure 16) in a tabular format. Otherwise, the user will get a notifying message that no match was found at this time, with the possibility to get results later.

In addition, CATS implements a “no answer situations” strategy. For demonstration, consider the query “I am looking for Clio 1999.” Suppose the system fails to find any match, it will look for all cars manufactured by Renault. Then, if it still fails to find any match, it will look for any French car.

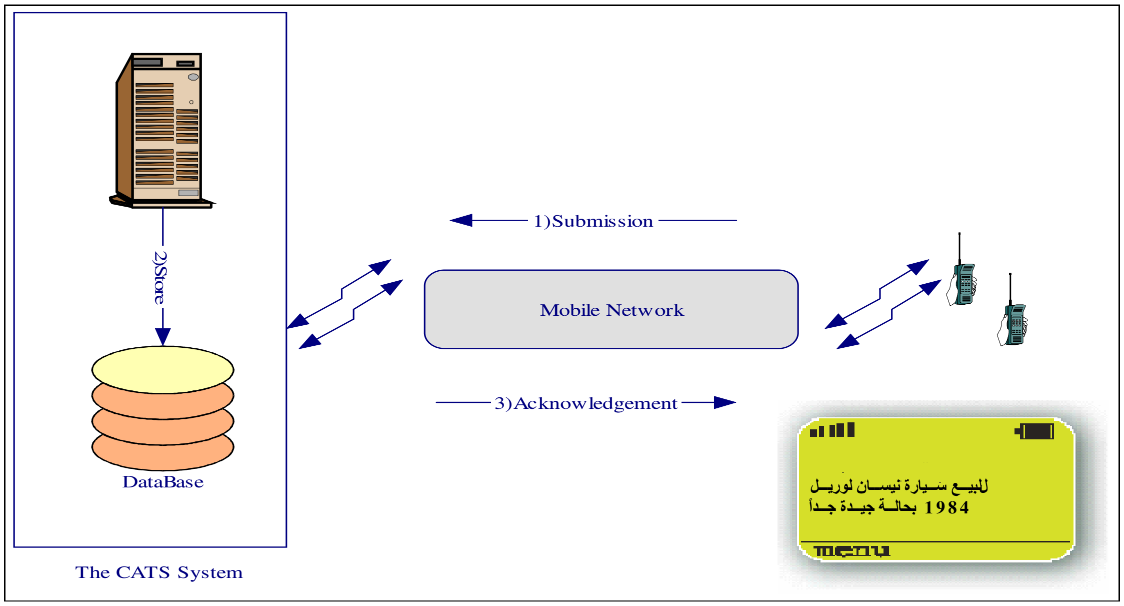


Figure 15 sending a selling classified ad “For sale Nissan Laurel 1984 in good condition”

In the current implementation of CATS, the mobile operator assigned one short number to this service (90050). This number represents the access point to CATS by mobile users. So, if a user wants to sell or buy, he sends a post to this short number. Note that the short number is the same for all domains handled by CATS (at this time: Real Estate and Cars).

This is more natural for users than to memorize several short numbers. This of course implies that CATS is able to distinguish between domains and post types.

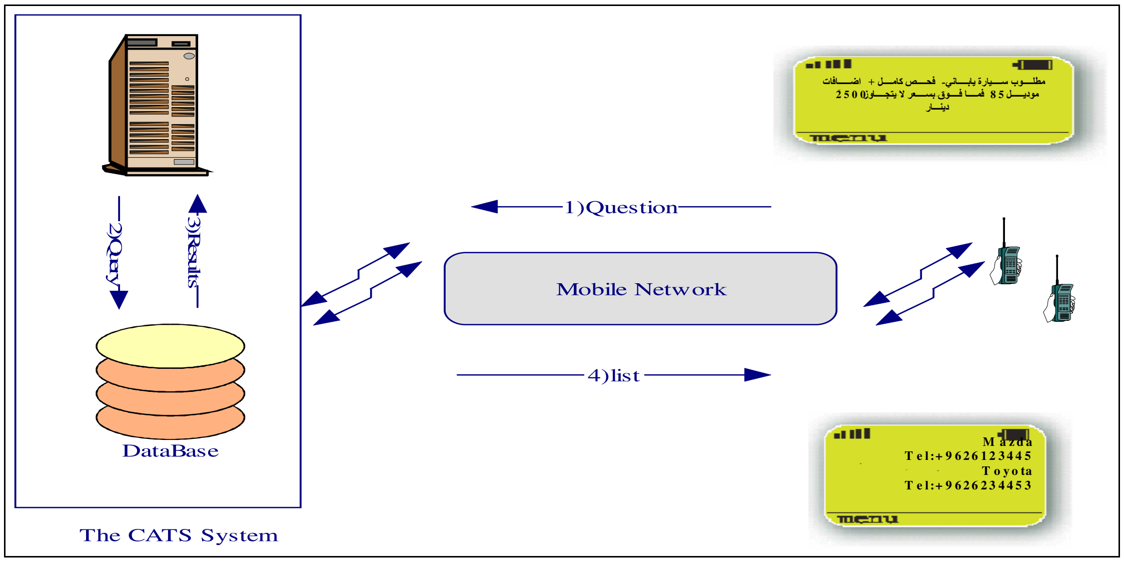


Figure 16: making a query and getting results through the CATS system. “A Japanese car is wanted, full check up, full option, above 1985; the price should be less than 2500 JD.”

3.1.3 Working environment

The system is implemented using the .NET framework class library. This framework provides full object-oriented support, supports a different programming languages, has a user friendly development, and is used for a wide range of applications like web, windows services, general assembly, interoperability with different window operating systems using MSIL (Microsoft intermediate language) and .NET CLR (Common Language Run time).

The SMS gateway (responsible for connecting with the mobile operator) and the EnCo (parser) are both Windows applications. As a consequence, CATS operates under Windows OS.

3.2 Overall architecture

The overall structure of the CATS system reflects both the corpus analysis and the adopted knowledge representation. The CATS system consists of a content extraction (CE) component and a query manager (QM) component.

The CE component receives SMS text and decodes it into the corresponding knowledge representation using a domain-specific lexicon. The system is able to extract knowledge from both types of messages.

The QM component takes the KR and converts it into SQL statements. It also issues the SQL statements (query or insert), and checks, validates and formats the results. It also handles situations where no answer found.

One important aspect of this design is that both questions and postings (documents) are processed by the same engine, using the same knowledge representation, leading to accurate matching of questions with answers.

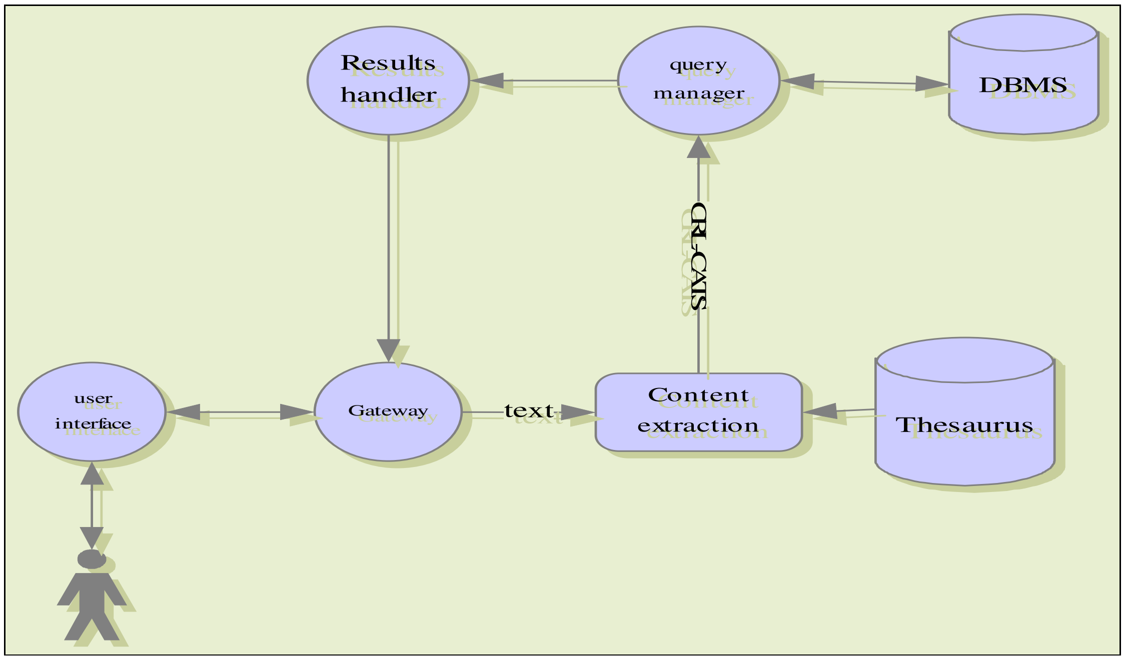


Figure 17: the overall architecture of the CATS system

3.2.1 The gateway

The gateway is a middleware tool that manages simultaneous connections to the mobile operator using the SMPP (short message peer-to-peer) protocol. When it receives an SMS message, it dispatches that message to a script running on an HTTP server, to a local executable program which sends it to the CE component. It also gets the response from result handlers and sends it through the mobile operator to the specified customer.

3.2.2 The dictionary

The dictionary is the backbone of the CATS system. It maps the domain-specific Arabic words to concepts. It also encodes morphological, syntactic, and semantic features for each word used for parsing the classified ads text, in addition to the semantic information.

This structure minimizes the effect of the alternative representations of text (including different orthographic forms, spelling errors, and abbreviations) on the overall performance of the system, specifically in the search process. Dictionary coverage is an especially challenging problem, since classified ads can be filled with all manner of jargon, abbreviations, and proper names, not to mention typos and fragmented phrases instead of fully-formed proper Arabic sentences. Currently, the dictionary contains 30,000 entries. Most of them are generated automatically by applying certain normalization algorithms. A detailed structure of the dictionary will be shown in chapter 6.

3.2.3 Content Extraction

When dealing with classified ads text, it is important to recognize different entities such as car makes (e.g., "Honda"), car models (e.g., "Accord"), locations (e.g.," Amman"), property types (e.g., "flat"), vehicle types (e.g., "saloon"). CE also includes the identification of other patterns such as numerical expressions: price, area, motor size, etc. To extract text attributes (location names, cars makers names, etc.), parsing is rule-based, directed by a semantic grammar.

We can view a classified ad as a group of information entities assembled in a natural language structure. A systematic approach is necessary for the analysis of free text. We believe that the development of a methodology to specify the language grammar used in a particular domain is necessary for building efficient content extractions systems. CE needs a good interface specification between NL and domain knowledge [Neumann and Xu 2004].

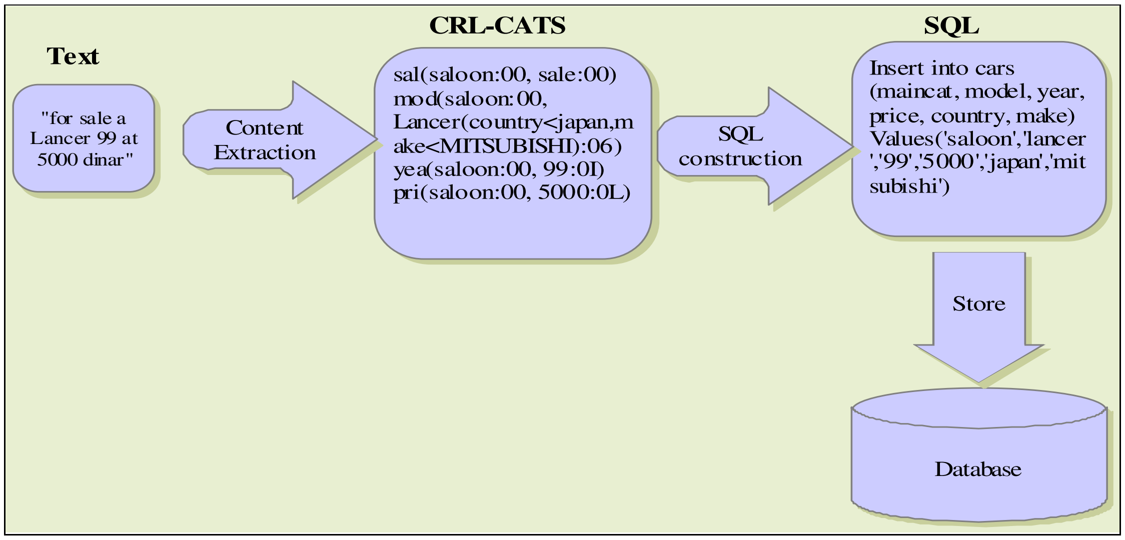

Text
"for sale a Lancer 99 at 5000 dinar"
Content Extraction
CRL-CATS
sal(saloon:00, sale:00)
mod(saloon:00, Lancer(country<japan,make<MITSUBISHI):06)
yea(saloon:00, 99:0I)
pri(saloon:00, 5000:0L)
SQL construction
SQL
Insert into cars (maincat, model, year, price, country, make) Values('saloon','lancer','99','5000','japan','mitsubishi')
Store
Database

Figure 18: information flow of "sell" post

The first step is the design of the content description language (CRL-CATS). It is a link between the unstructured free text and the structured database (figures 18, 19). By defining the sets of valid objects, relations, properties and attributes within the description, we set the boundary for each particular domain. This content representation reflects the natural sublanguage studied and analyzed in a specific domain. It is also language-independent representation of the information.

The analysis of a received classified ad is affected by the following:

- the structure of CRL-CATS
- the structure of the natural language.

In the following chapters we will discuss in detail the process of content extractions and the tools used. A full description of the CRL-CATS will be presented in the next section.

3.2.4 Query Manager

The Query Manager (QM) performs the following functions:

- It receives the CRL-CATS expressions and extracts from them the needed information such as the name of the property, the value of the property and the domain object. Additionally, it extracts semantic information. This extraction process is performed using regular expressions.
- It identifies the type of post: “looking for” or “sell” post according to the extracted information from the first step.
- It identifies the classified ads domain: Cars or Real Estate.
- It issues a corresponding Structure Query Language (SQL) request. For demonstration, if the CRL-CATS corresponds to a “looking for” post and in the Car domain, the QM generates an SQL query and directs it to the Cars Database. Similarly, if the CRL-CATS is related to a “sell” post and in the Real Estate domain, it generates an “insert” statement to the Real Estate database.

If no results are received after issuing a SQL query to one of the databases, the QM issues another query with relaxed constraints.

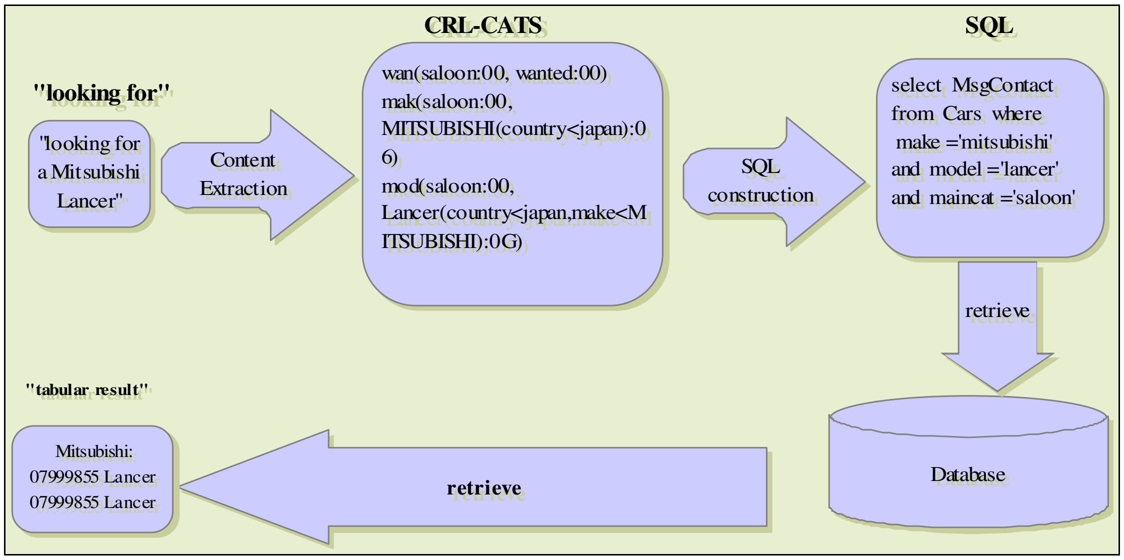

CRL-CATS
SQL
"looking for"
"looking for a Mitsubishi Lancer"
Content Extraction
wan(saloon:00, wanted:00)
mak(saloon:00, MITSUBISHI(country<japan):06)
mod(saloon:00, Lancer(country<japan,make<MITSUBISHI):0G)
SQL construction
select MsgContact from Cars where make ='mitsubishi' and model ='lancer' and maincat ='saloon'
retrieve
"tabular result"
Mitsubishi: 07999855 Lancer 07999855 Lancer
retrieve
Database

Figure 19: information flow when processing a "looking for" post

3.2.5 DBMS

The usage of a database is very important to achieve good performance. As we have shown in figure 18, the natural language “sell” posts are eventually structured, organized and stored in a database (SQL Server); enabling CATS to provide responses accurately and in a very short time.

The precision and efficiency of information access improves when digital content is organized into tables within a relational database. Once information is encoded in a database, it can be organized into taxonomy or searched over by textual attribute or feature. This stands as a vast improvement over the usual search protocol: index content and query full-text documents by keyword [Adams 2001].

To enhance performance further, we allocate one separate table for each domain (Cars, Real Estate), even if we have to add redundant semantic data in it. This design enhances both the insertion and the query processes.

3.2.6 Results Handler (RH)

This component is responsible for formatting the tabular result structure to fit within the constraints of SMS message length (this can be changed, since CATS supports long SMS messages). Hence, if the user is interested in getting the rest of the results set, she/he can send “more” and the RH memorizes his previous request and sends new results.

3.3 The Content Representation Language for CATS

We have chosen a minimal but sufficient formalism to express the content of SMS used in posting or querying classified ads.

In CRL-CATS (Content Representation Language for CATS), a posted SMS is represented as a set of binary relations between objects. There are no variables, but the dictionary is used as a type lattice allowing specialization and generalization.

Table 16: specs of the CATS project

Domain	SMS-based classified ads (Cars and Real Estate)
Documents source	Users
Document type	Spontaneous unedited text
Intermediate Representation	CRL-CATS
User requests	Spontaneous unedited text
Matching/retrieval	semantic

There is big advantage for us to use such a restricted formalism: as it is formally very near to the UNL formalism, we can use the same tool for CE as the tool we used a few years ago for writing the first Arabic-UNL enconverter, namely the EnCo specialized programming language.

We will now present in detail the components of CRL-CATS: objects and properties.

3.3.1 Syntax and semantics of CRL-CATS

3.3.1.1 Objects, properties and statements

The basic data model of CRL-CATS consists of three object types:

Main Domain Object (MDO). The central notion in CRL-CATS is that there are *things* that we wish to make assertions about. Examples of such things in the Cars domain are "Saloon" and "Pickup" and in the Real Estate domain are "Apartment" and "Villa".

Properties. A property is a specific aspect, characteristic, attribute, or relation used to describe a MDO. A "property" and its value are pieces of information that may be attached to things, but which are not sufficiently important in the specific domain to be considered things in their own right.

Some examples of properties of the thing "Red" is: the color of my car. In CRL-CATS, color is simply a property of the MDO "Saloon" and is encoded using the following statement:

Col (saloon, red)

Statement. A specific MDO together with a named property plus the value of that property for that MDO is a CRL-CATS statement:

mak(bus:06, HYUNDAI(country<korea):0R)

A CRL-CATS expression that encodes one classified ad post contains one or more CRL-CATS statements:

```
[S]
wan(saloon:06, wanted:00)
mak(saloon:06, KIA(country<korea):0C)
yea(saloon:06, 95:0L)
[/S]
```

This data model is quite similar to the object-property-value triples defined in Resources Description Framework (RDF) recommended by the *World Wide Web Consortium* (W3C), to model meta-data about the resources of the web [Lassila and Swick 1999].

3.3.1.2 Binary relations (properties)

Binary relations are the building blocks of CRL-CATS expressions. They are made up of one relation and two CWs.

<relation> (<CW1>, <CW2>)

A binary relation is written as follows:

<Binary Relation>	::= <Relation Label> "("
	< CW $_1$> [":" < CW -ID$_1$>] [<Attribute List>] ","
	< CW $_2$> [":" < CW -ID$_2$>] [<Attribute List>] ")"
<Relation Label>	::= a relation
< CW >	::= a CW
<Attribute List>	::= { "." <Attribute> } …
<Attribute>	::= an attribute
< CW -ID>	::= two characters of '0' – '9' and 'A' – 'Z'

In the Cars and Real Estate domains, a relation label is a property name. CW1 represents the MDO and CW2 represents the value of the attribute.

Property (MDO, value)

3.3.1.3 CATS Words (CWs)

CATS Word (concept). A CW is made up of a character string (an English language word) followed by a list of semantic classification.

<CW> ::= <Head Word> [<semantic label>]

Head Word. it is an English word that is interpreted as a label for a set of concepts that may correspond to word senses of that word in English.

Semantic label. it relates to the concept hierarchy or taxonomy and specifies the nature of the categories in terms of generalization and specialization.

<semantic label>::= ClassName "<" ClassInstance

For example, the following CW

Honda (country < japan)

means that the Honda car makers belong to the country of Japan.

3.3.1.4 Attributes

Attributes specify how a CW is being used. They are usually attached to CW2 which represents the property value in a CRL-CAT statement.

They also give more details or to put some restriction on the property value. Specifically, they are used in normalization of numerical attributes values and in capturing mentioned ranges used to perform attribute-based searching. As an example, in the following expression:

Price (car, 5000@less)

"@less" means that the price of the car should be less than 5000.

Also, "@meter" means that the unit of measure is meter. Similarly, the "@build" attribute is used to indicate a building area and "@space" indicates the surrounding area. In the same line, when a numerical value is attached to "@thousand", this number should be multiplied by 1000 for further processing.

3.3.2 Examples of CRL-Expressions

A full inventory of relations and attributes for each domain is given in appendices A and B.

3.3.2.1 CRL-CATS for a "sell" post

CRL-CATS represents information post by post. For example, consider the following "sell" post:

للبيع سياره هوندا موديل 1997 جير اوتوماتيك مكيف سنتر بسعر 7750 دينار.

For sale Honda year 1997 automatic transmission air condition center lock price 7750 dinar

The CRL-CATS expression extracted from it is:

```
[S]
sal(saloon:06, sale:00)
mak(saloon:06, HONDA(country<japan):0C)
yea(saloon:06, 1997:0U)
fea(saloon:06, automatic gear:0Z)
fea(saloon:06, air condition:1D)
fea(saloon:06, center lock:1I)
pri(saloon:06, 7750:1S)
[/S]
```

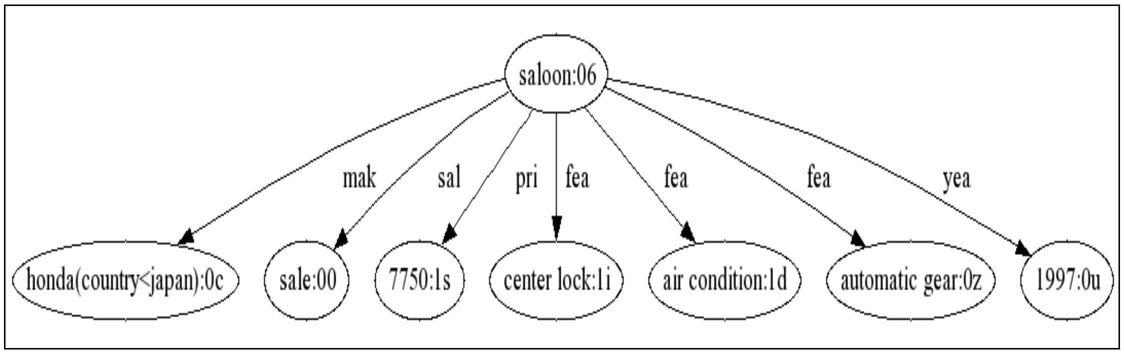


Figure 20: CRL-CATS graph for "For sale Honda year 1997 Automatic transmission Air condition Center Lock price 7750 Dinar"

In Figure 20, the arcs labeled with *mak* (make), *sal* (Sale), *pri* (price), *fea* (feature) and *yea* (year) are property labels. The nodes *saloon, sale, HONDA (country<japan), automatic gear, air condition* and *center lock* are CATS Words (CWs). The CW *saloon* represents the MDO; other CWs represent the values of the properties. The label *country<japan* is the semantic label for HONDA, providing information about the country of the manufacturer.

Note that a property such as *fea* (feature) can have multiple values ("air condition", "automatic", "center lock"). In other formalisms, we might have:

fea(saloon, [air condition, automatic, center lock]), where [ ] stands for "and". Here, we simply allow any number of arcs with the same label going out of a node in the graphical representation.

3.3.2.2 CRL-CATS expression for a "looking for" post

To demonstrate a "looking for" post in CRL-CATS, consider the following post:

أريد شقة تمليك بعمان الغربية مساحتها لا تقل عن 150م مربع وسعرها 25 ألف.

I am looking for an apartment in West Amman, its area not less than 150 square meters and its price is 25 thousand.

```
[S]
wan(flat:05, wanted:00)
loc(flat:05, wAmman:0G.@area)
are(flat:05, 150:1B.@meter.@more)
pri(flat:05, 25:1S.@ALF)
[/S]
```

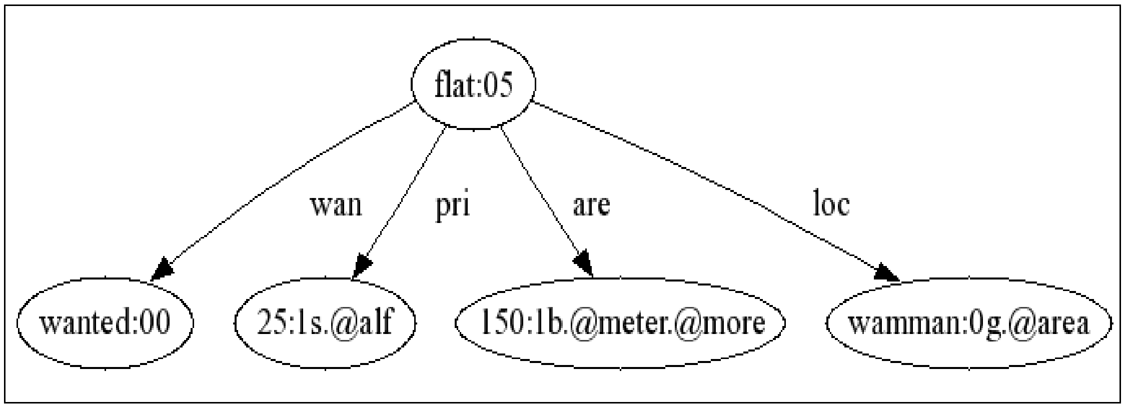


Figure 21: CRL-CATS graph for "I am looking for an apartment in West Amman, its area not less than 150 square meters and its price is 25 thousand"

In the example shown in figure 21, the property labels are *wan* (wanted), *pri* (price), *are* (area) and *loc* (location). The CWs are *flat* (the MDO), *wanted* and *wamman* which represent property values. The attribute (@alf) indicates that the price is 25000. The attribute @meter indicates that the area is measured in m^2, and @more indicates that the area should be more than 150 m^2. The @area attribute attached to *wamman* indicates that it is an area consisting of districts.

3.3.3 Adequacy of CRL-CATS and comparison with UNL

3.3.3.1 Adequacy

Generalization/specialization and reasoning are handled by the semantic labels attached to CWs. Object-orientedness is achieved by having the object-property-value data model. The same goes for ease of porting to databases, in which we have a regular language that can easily be processed. As to language independence, CRL-CATS has an inventory of CWs, relations and attributes that can be produced from any language. Likewise, natural language text can also be generated from CRL-CATS expressions, irrespective of the language itself.

Additionally, CRL-CATS provides an abstraction level sufficient to express the meaning regardless of the structure of the original post.

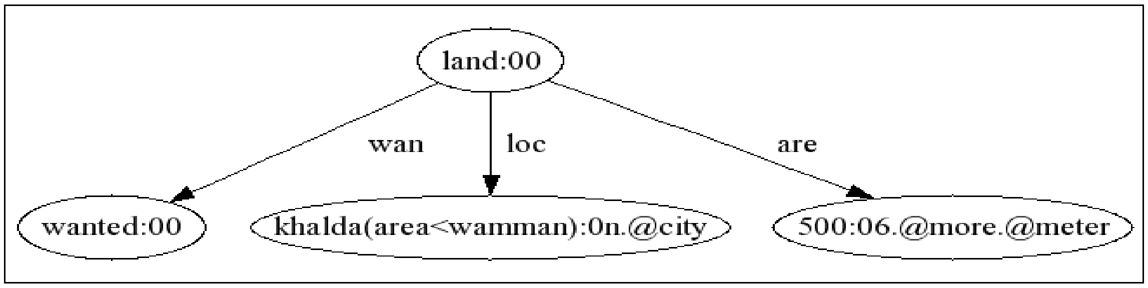


Figure 22: CRL-CATS for "wanted 500 square meter and above in Khalda"

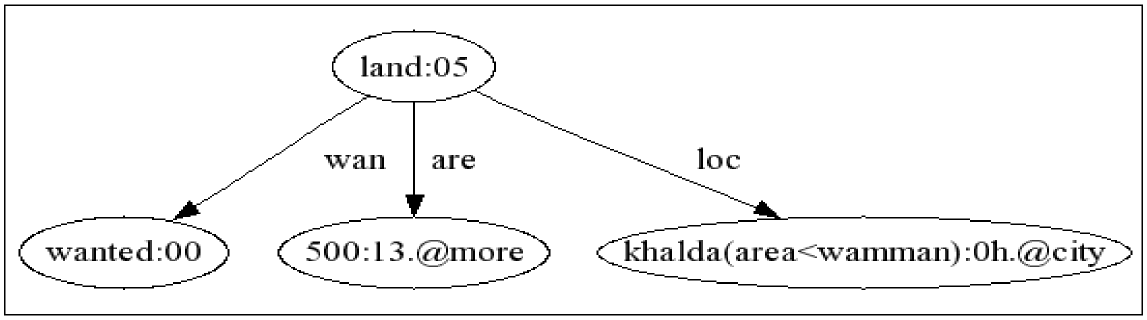


Figure 23: CRL-CATS for "I am looking for a piece of land in Khalda its area not less than 500"

To demonstrate this, consider the two sentences listed in table 15 chapter 2. Although, both are syntactically different, they have the same meaning. Figures 22 & 23 show how a CRL-CATS expression produced by CE is able to express this meaning.

3.3.3.2 *Comparison with UNL*

CRL-CATS has a similar syntax that of UNL [Uchida and Zhu 2003, Uchida and Zhu 2005] because we are using the Enconverter tool (EnCo) [Uchida 1999]. However, these two representations are semantically extremely different. While CRL-CATS is a purely semantic interlingua independent of the NL expressions, UNL is a “linguistico-semantic” formalism, where a graph represents the abstract structure of an English utterance.

During the development of this project, we first tried to use UNL to represent the classified ads. However, this attempt proved not feasible computationally, due to the deep-syntactic nature of UNL relations [Uchida 2006]. Hence, we had to derive new specifications to describe accurately the classified ads for different restricted domains. This does not contradict the claim that UNL can describe any type of information [Boitet 2005], because it is true that any information can in principle be expressed in UNL. Simply, a purely semantic pivot is far more adequate for computing with the content of a NL utterance.

To demonstrate the differences between CRL-CATS and UNL, consider the following post:

أرغب ببيع سيارتي ميتسوبيشي اللون ابيض موديل 1990

"I want to sell my Mitsubishi car, White color year 1990"

Its CRL-CATS expression is

[S]
sal(saloon:0A, sale)
mak(saloon:0A, MITSUBISHI(country<japan):0G)
col(saloon:0A, white:0W)
yea(saloon:0A, 1990:17)
[/S]

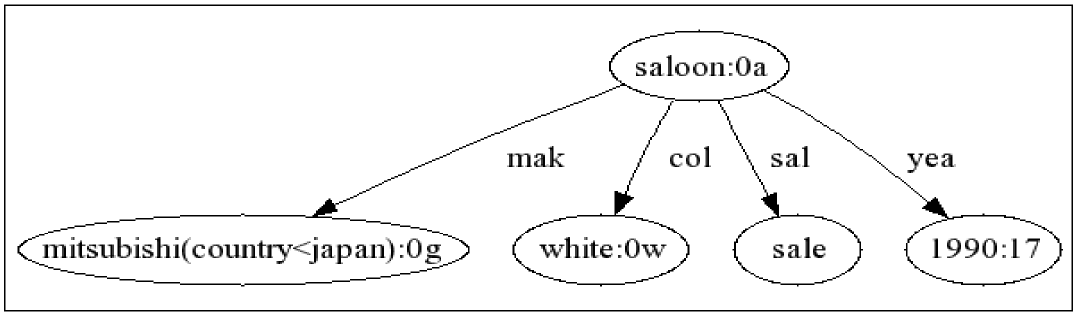


Figure 24: CRL-CATS for "I want to sell my Mitsubishi car, White color year 1990"

Its UNL representation:

[S]
agt(want.@entry.@present , I)
gol(want.@entry.@present, sale)
obj(sale, car)
pos(car, I)
mod(car, mitsubishi)
aoj(color, car)
mod(color, white)
aoj(year, car)
qua(year, 1990)
[/S]

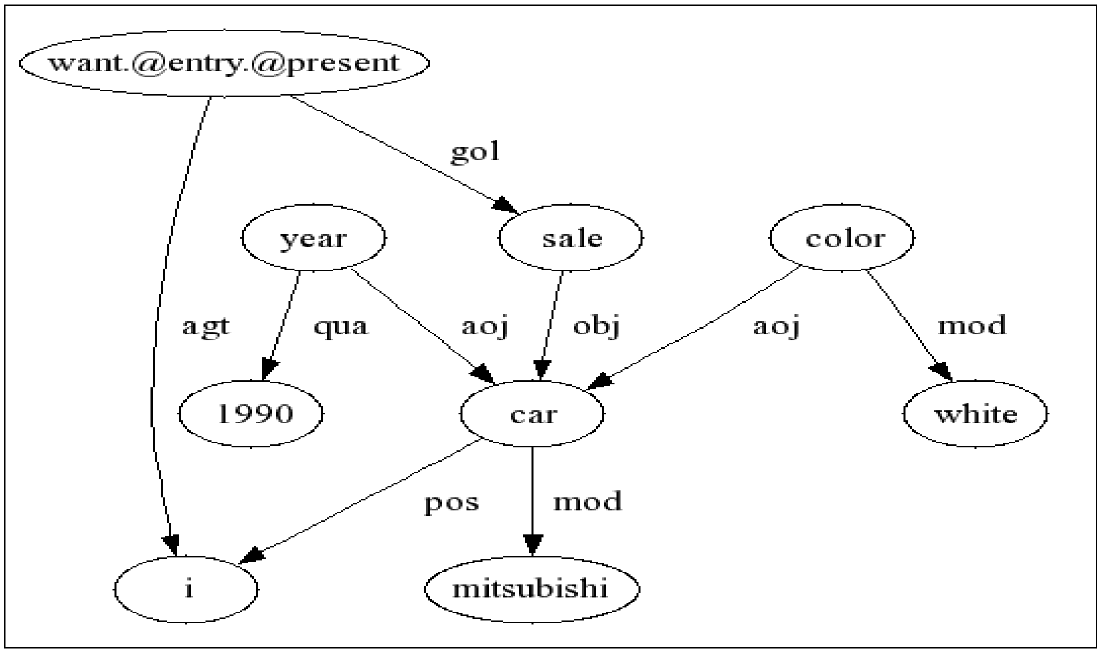


Figure 25: UNL for "I want to sell my Mitsubishi car, White color year 1990"

Even if we could "normalize" the ads to correct Arabic sentences, we could not stop on the UNL level and do content processing on it. A first reason is that there are many ways to represent the above classified ads in UNL, depending in the source language and the quality of the analysis. Also, it is worth mentioning that there are amounts of ambiguity in the UNL expressions which fail in many occasions to deliver the correct meaning.

UNL was designed for multilingual machine translation with two processes associated with it: the enconversion and the deconversion. Hence, the purpose is to produce a correct sentence from the source one. We see also in this specific example that the UNL representation does not directly provide the exact knowledge one is looking for in a specific domain, making sometimes the processing of these UNL graphs extremely difficult. In other words, the semantic content is distorted by the language information present in the UNL graph.

Conclusion of Part A

The analysis of some NLP-based e-commerce system shows clearly that there is a problem in developing them. The few of them which have really tried to use different forms of full NL interface systems have failed. Our analysis shows that this failure is stems from using inadequate techniques in building those systems.

We also showed our approach, which aim at handling both the linguistic and the domain knowledge components. To demonstrate it, we built CATS not only as a prototype but as a "production system".

By using a web-based corpus a priori similar to our intended SMS-based corpus, we produced a first version of the description of the sublanguage and the domain (microworld) at hand.

That led us to choose suitable techniques to implement the linguistic component (as a content extractor) and the semantic component (as a QA system using a relational DB), and to define CRL-CATS, a simple but efficient and sufficient content type language as the intermediate point between CE and QA.

Part B. Content Extraction

Introduction to Part B

Content extraction is the most important component of the CATS system. We could have tried implemented it in many programming languages and various techniques, such as using pattern-matching with regular expressions in tcl/tk or Perl (as done by H. Blanchon to produce the French-IF extractor in the Nespole project), or using Xerox XFST tools for Arabic (but they are proprietary), or using Prolog and unifications, or converting the string to a flat tree and using the ROBRA tree-transformation language in Ariane-G5 or directly programming in Java, etc.

We choose to work with EnCo simply because we had used in 1998-1999 to work the first Arabic-UNL enconverter, knew how to develop in it in a systematic way despite its relatively low-level, and were certain its speed would be sufficient to support a real-time application. We would like to thank H. Uchida, who allowed us to use it to build this experimental system.

EnCo is a rule-based programming language specialized for the writing of enconverters (translators from a NL into UNL), and provided by the UNL center. From our experience in developing Arabic-UNL module using this programming language, it is a time and effort consuming process.

In this part we will describe the EnCo programming language, its usage and operations. Then we will introduce our lingware engineering method to provide a more systematic and productive way of EnCo programming. Finally we will describe the CE process, showing how we adapted a general purpose translator to perform a domain-specific extraction. Additionally we will show how to perform this task using a semantic grammar. Finally, we will give some examples and mention the remaining problems and limitations of the system.

Chapter 4. The EnCo rule-based language

Introduction

EnCo is a rule-based programming language specialized for the writing of enconverters (translators from a NL into UNL), and provided by the UNL center. It provides a framework for performing morphological, syntactic, and semantic analysis synchronously.

EnCo works in the following way. An input string is scanned from left to right. During the scan, all matched morphemes with the same starting characters are retrieved from the dictionary and become candidate morphemes. The rules are applied to these candidate morphemes, according to the rule priority, in order to build a syntactic tree and semantic network for the sentence. The character string not yet scanned is then scanned from the beginning according to the applied rule; the process continues in the same manner. The output of the whole process is a semantic network expressed in the UNL format. If the dictionary retrieval or the rule application fails, it backtracks.

This language has been used by the UNL-Arabic center and some other UNL language centers to analyze the input natural sentences and produce UNL expressions.

During the development of the Arabic-UNL module, we developed a set of enconversion rules that covers the following:

- Verbal sentences and verbal modifiers
- Morphological Analysis
- Idafa construct[8]
- Adjectival phrase
- Prepositional phrase
- Adverbial phrase
- Relative clause and embedded sentences
- Scope generation
- Disambiguation and backtracking
- Nominal phrase

The use of this tool has been productive in the analysis of usual Arabic sentences. However, at that time we had some problems related to the process of encoding those rules which are hard to control and maintain. Other problems are related to the support of the EnCo tool itself.

[8] The IDAFA construction is an important grammatical structure in Arabic. It is a genitive construction in which two nouns are linked in such a way that the second (second part of the construction) qualifies or specializes the first (first part of the construction).

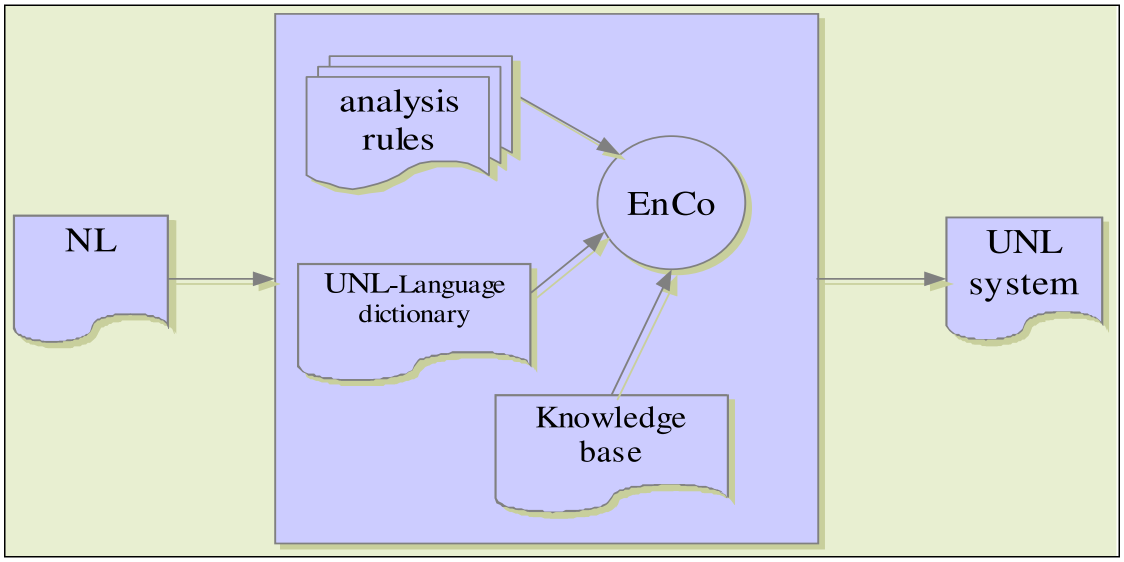


Figure 26: EnCo Architecture

4.1 Architecture and structure of enconverters written in EnCo

4.1.1 Resources

An enconverter[9] written in EnCo consists of the following components

- Analysis Rules
- Dictionary (NL-UNL)
- Knowledge base

The knowledge base is optional. It is foreseen to be used in disambiguation, but that was never implemented during the first phase of the UNL project.

4.1.2 Architecture and functions

The abstract model underlying EnCo is a computing device consisting of:

- an input tape (node-list), containing a t the beginning the input text (in one node) and then the input morpheme or lexemes recognized so far, (each in one node), followed by the remaining text (in one node).
- 2 active heads on that tape (left analysis window and right analysis window)
- a group of "context" heads (condition windows) surrounding the 2 active heads.
- an output "node-net" sharing some nodes with the node-list.

[9] We use the term "enconverter", and not "parser", because the process involves a lexical transfer from the "lexical space" of the NL at hand (while many have several "levels" such are morphs, morphemes, word forms, lexemes, lemmas, derivational lexical families, and word senses) to the "lexical space" of UNL (the UWs, and their hierarchy).

When started, the EnCo executes a series of discrete transitions, as determined by its rules and by the initial contents of the tape.

The tape (node-list) and the node-net contain nodes having 4 fields. Each node contains an entry from the language dictionary: a notation of a word of each language, an UW, associated grammatical features and UNL attributes (Figure 27).

Nodes in the node-net may be linked by arcs bearing semantic relations. Some nodes may appear in the node-list and the node-net. That should be the case, in particular, for the entry node of the resulting graph.

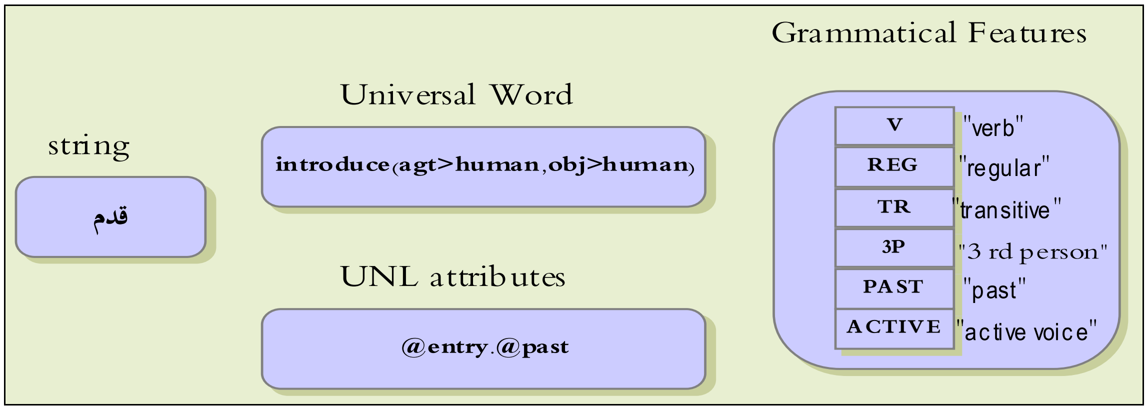


Figure 27: Example of a node in the tape or the node-net

EnCo operates on the nodes of the Node-list through its windows. There are two types of windows, namely *Analysis Window* and *Condition Window*. The 2 windows called *Analysis Windows (AW)* are circumscribed by the windows called *Condition Windows (CW)*.

Figure 28 shows the structure of EnCo. EnCo uses the Condition Windows for checking the neighboring nodes on both sides of the Analysis Windows in order to judge whether the neighboring nodes satisfy the conditions for applying an Analysis Rule or not. The Analysis Windows are used to check two adjacent nodes in order to apply one of the Analysis Rules. If there is an applicable rule, EnCo adds grammatical features to or deletes grammatical features from these nodes, and/or creates a partial syntactic tree and/or UNL network, according to the type of the rule.

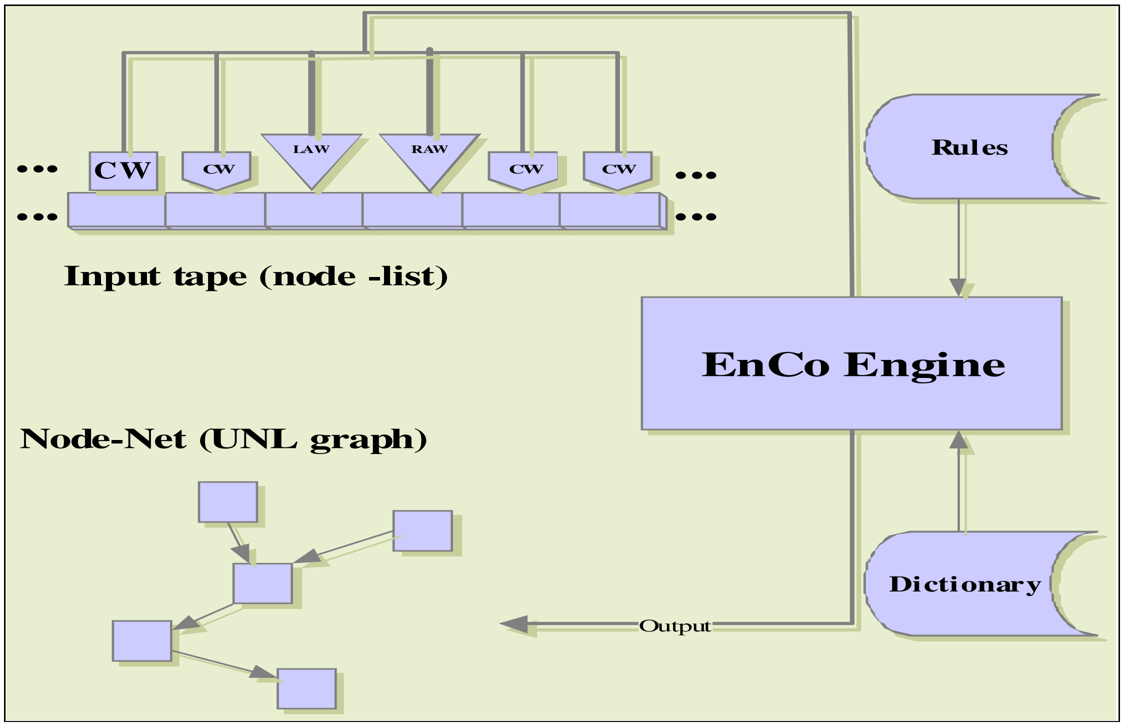


Figure 28: computing model of EnCo

If the dictionary retrieval or the rule application fails, EnCo backtracks. The rule has no "goto" field; the sequence of execution is controlled by the symbols (grammatical attributes) as we will explain later.

4.1.3 Sentence segmentation

EnCo handles a "segment" at a time. A segment is normally a sentence, or a title, or even an isolated term. It implements a lexical, or dictionary-based, segmentation which utilizes a lexicon accessed by morphs (full words, or affixes) of the language being analyzed. The input text is scanned (left-to-right)[10] and matches are returned. The longest (or "maximal") match at any given point is returned.

The segmentation part of the EnCo engine uses the strategy of maximal match segmentation, or "best" segmentation. The maximal match segmentation attempts to minimize the number of words in a sequence of *characters* by finding the longest matches in the dictionary at each point in the input.

EnCo accesses the dictionary to retrieve all dictionary entries matching the beginning of characters string beginning under the Right Analysis Window (RAW), or at any one of the right side Condition Windows.

Priority is given to longer match, however this can be changed by giving higher frequency values to shorter access strings. For demonstration, consider the following character sequence in the input tape:

.........فريق.......

[10] Arabic is stored physically left-to-right

When the EnCo accesses the dictionary, it retrieves the following entries:

1. [فريق] “team”
2. [فر] “escape”
3. [ف] “and then”

Since “1” is the longest match, it will be selected by the system. However, if one of the other entries has a higher frequency than entry “1”, the string in the tape will be segmented based on it. To show this, suppose “3” has the highest frequency, and then the original string will be segmented as follows:

“spittle “ ريق+ “and then”ف

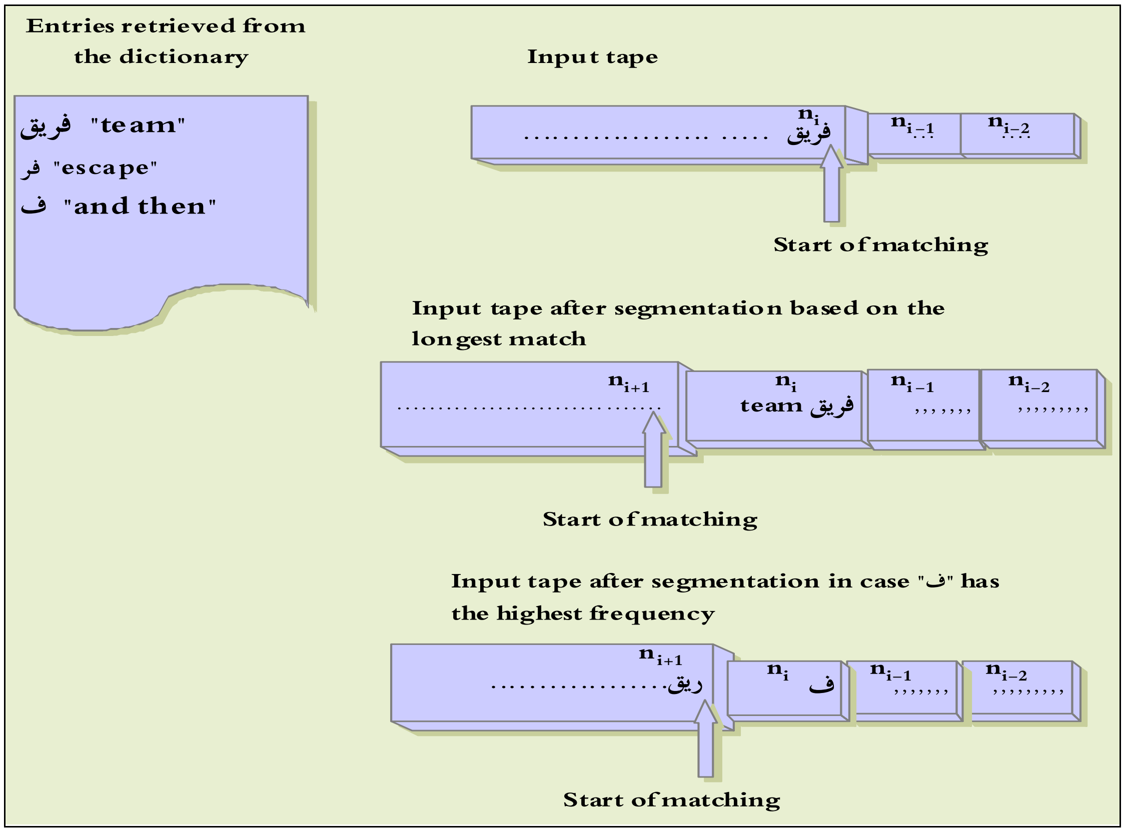


Figure 29: example of a segmentation (R-L view)

Finally, the node in the tape is filled with the selected morpheme along with the corresponding UW and the grammatical features.

4.1.4 Basic operations

We will explain the basic operation of EnCo through a step by step example of execution. EnCo starts operating from the configuration state. In its initial configuration the tape or Node-list consists of:

- the Sentence Head node (SHEAD) ,
- one node containing the input text, and

- The Sentence Tail node (STAIL).

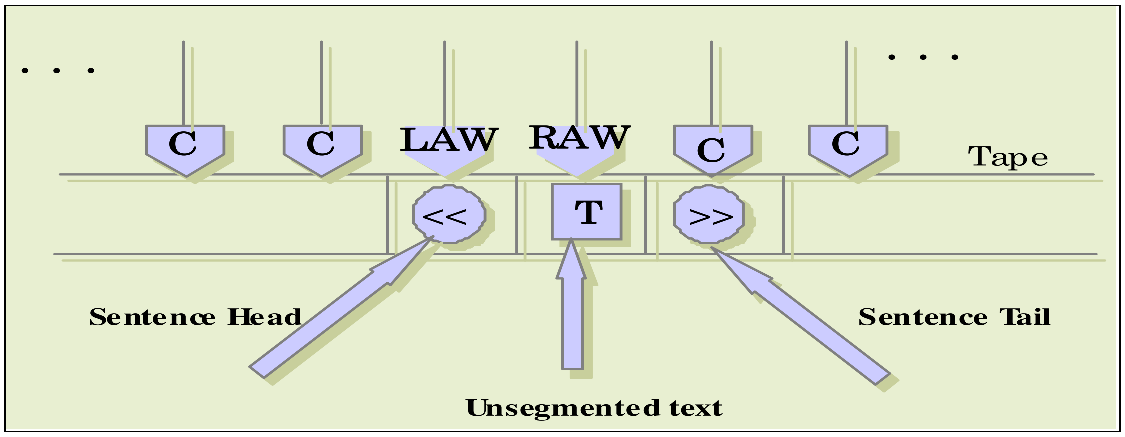


Figure 30: initial configuration of EnCo

For each transition, the EnCo engine checks what configuration it is in and what content (grammatical attributes retrieved initially from the dictionary) is written on the tape below the head(s). Based on those, it then changes to a new configuration and moves the head one space left or right. The EnCo stops when arriving in any HALT configuration, that is, the Sentence Tail comes under the left Analysis Window.

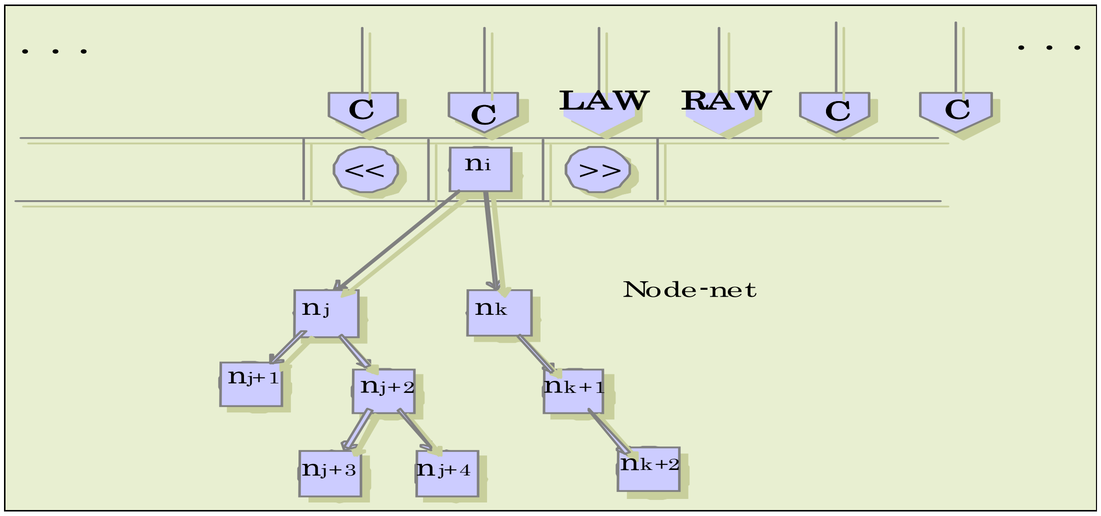


Figure 31: the Halt configuration of EnCo

Between the initial configuration and final configuration, EnCo uses the grammatical features in the nodes under analysis windows to move from one configuration to another.

At any particular moment in time, EnCo is in a describable configuration. Between this moment and the next discrete time stamp, the machine reads its input from the tape, refers to rules controlling its behavior, and considering both the input and the current configuration, determines what behavior to exhibit (i.e. erase/write on tape, move left, move right, create a an arc in the UNL graph, etc.), which determine the next configuration.

To demonstrate the behavior of EnCo, consider the following simple Arabic sentence given as input to our Arab-UNL enconverter (not to CATS).

فهم خالد **{fahima Khalid}** --------- > *Khalid understood*

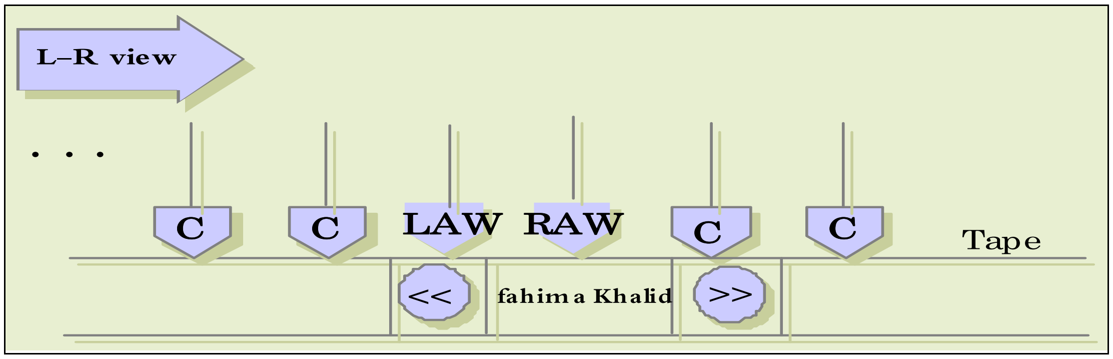


Figure 32: the initial configuration of "fahima Khalid"

Figure 32 shows the initial configuration: the LAW is pointing to the SHEAD, and the RAW is pointing to the unsegmented text. The first action by EnCo at this point is to access the dictionary and find the best match.

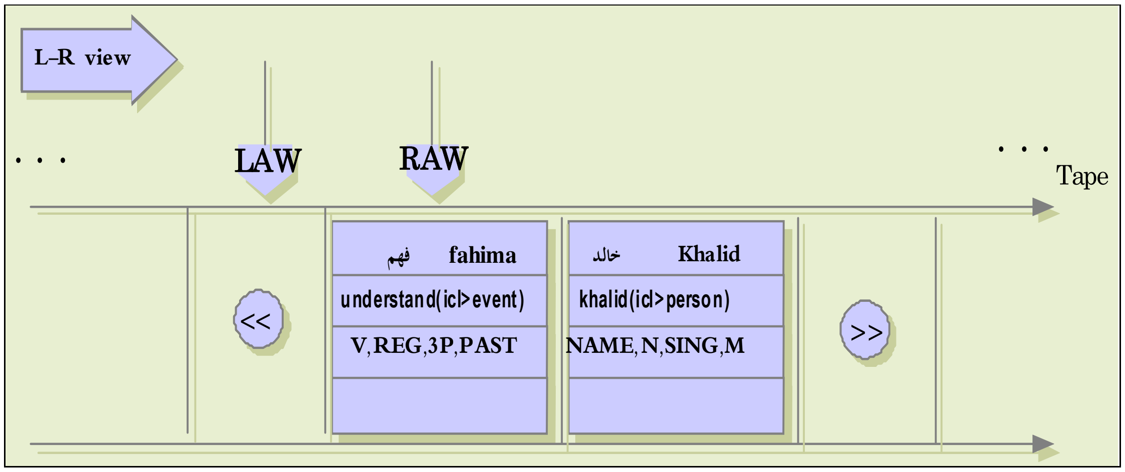


Figure 33: segmentation of "fahima Khalid"

Figure 33 shows the tape after the segmentation. Under the RAW the node contains the headword "fahima", the UW "understand" and the grammatical attributes[11] as loaded from the dictionary. At this position, EnCo looks for rules that satisfy the current configuration such as: LAW is pointing to the SHEAD and RAW is pointing to a verb. The rule should also satisfy what action to take (For the sake of demonstration supposes it is "shift right").

[11] "NAME" indicates a named entity, "N" indicates a noun, "SING" indicates singular noun, and M indicates a masculine noun.

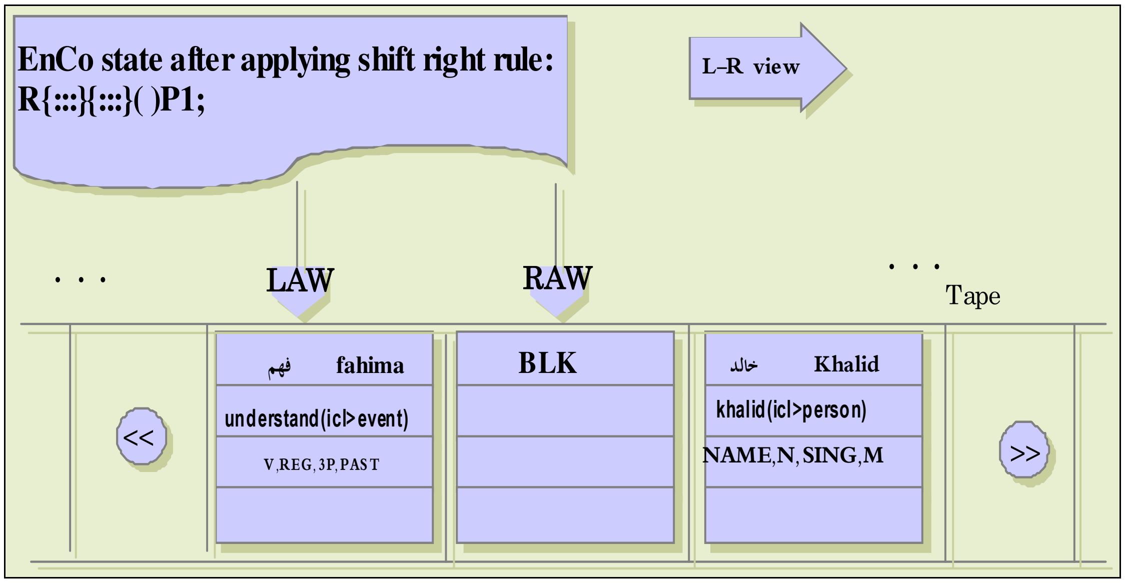


Figure 34: EnCo configuration after a shift right rule

As shown in figure 34, at this configuration after a move right action, LAW is pointing to "fahima" and RAW is pointing to the "BLANK" or white space.

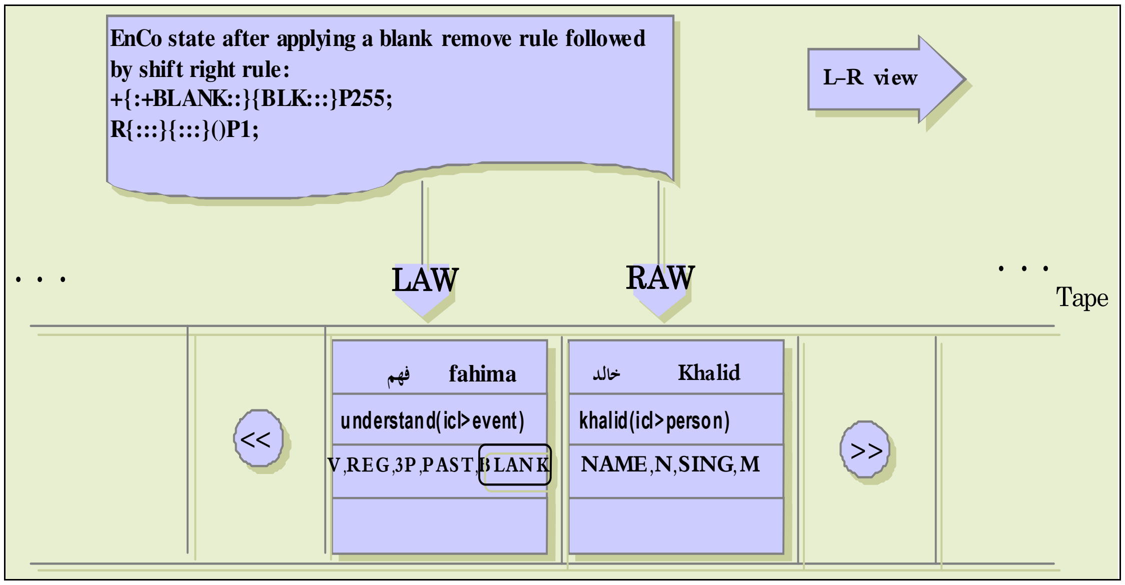
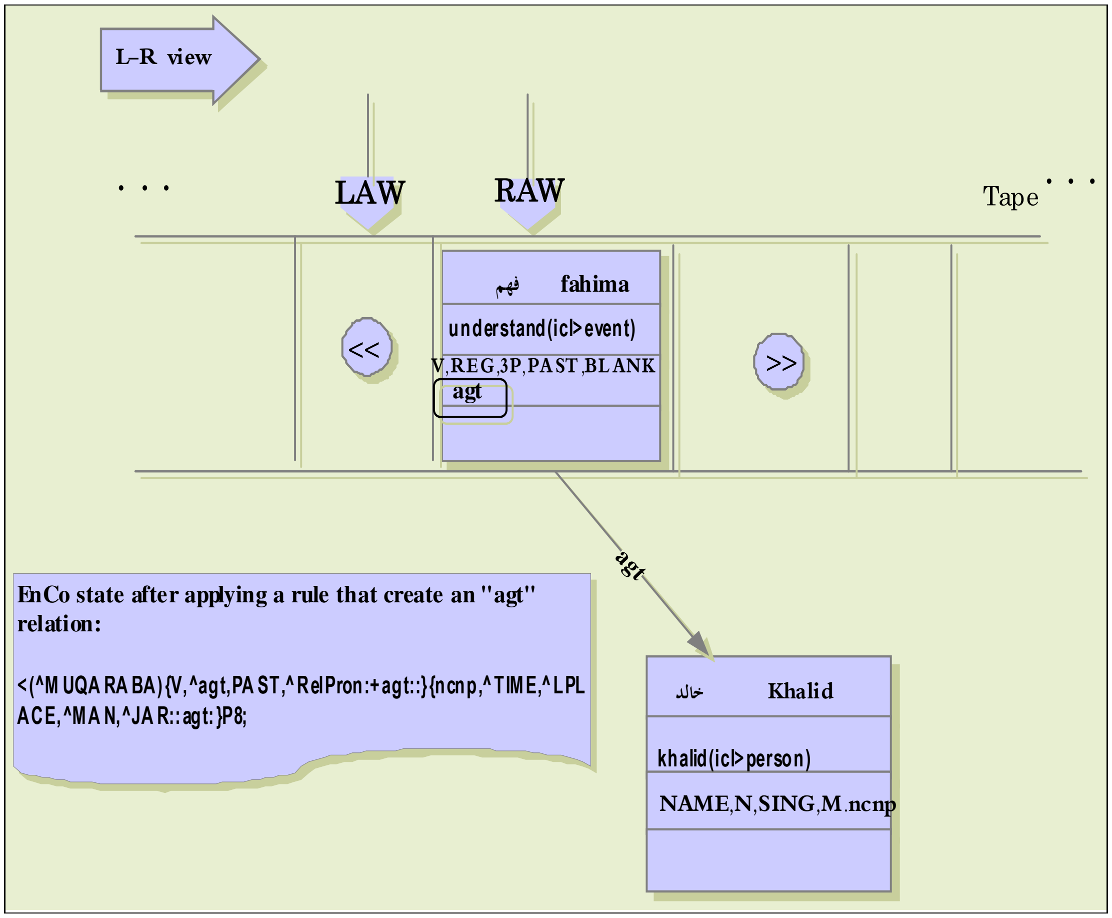


Figure 35: removing the blank and move right

At this point, a rule that removes the "BLANK" node is fired by combining it to "fahima", followed by a rule to shift right[12].

The positions of LAW and RAW are shown in figure 35. If we have a rule saying "if LAW is pointing to a verb and RAW is pointing to a named entity then create an *agent* relation from *fahima* to *khalid* in the UNL graph", it will fire. Figure 36 shows the tape after execution of such a rule and the generated UNL graph.

[12] Due to its non-deterministic nature, EnCo always shift to the left after any reduction operation to examine all the possibilities that may occur. Hence, directly after the removal of the BLANK, LAW points to SHEAD and RAW points to "fahima".



EnCo state after applying a rule that create an "agt" relation:

<(^MUQARABA){V,^agt,PAST,^RelPron:+agt::}{ncnp,^TIME,^LPLACE,^MAN,^JAR::agt:}P8;

Figure 36: creating a relation in the UNL graph

The above example shows that according to the grammatical attributes of the nodes under the two Analysis Windows, EnCo decides whether they are to be combined into a single headword or a relation is to be set up between them while combining or modifying the two nodes and one of the nodes is deleted from the node-list. Rules ensure that the entry node of the sentence is never deleted until the end of the analysis.

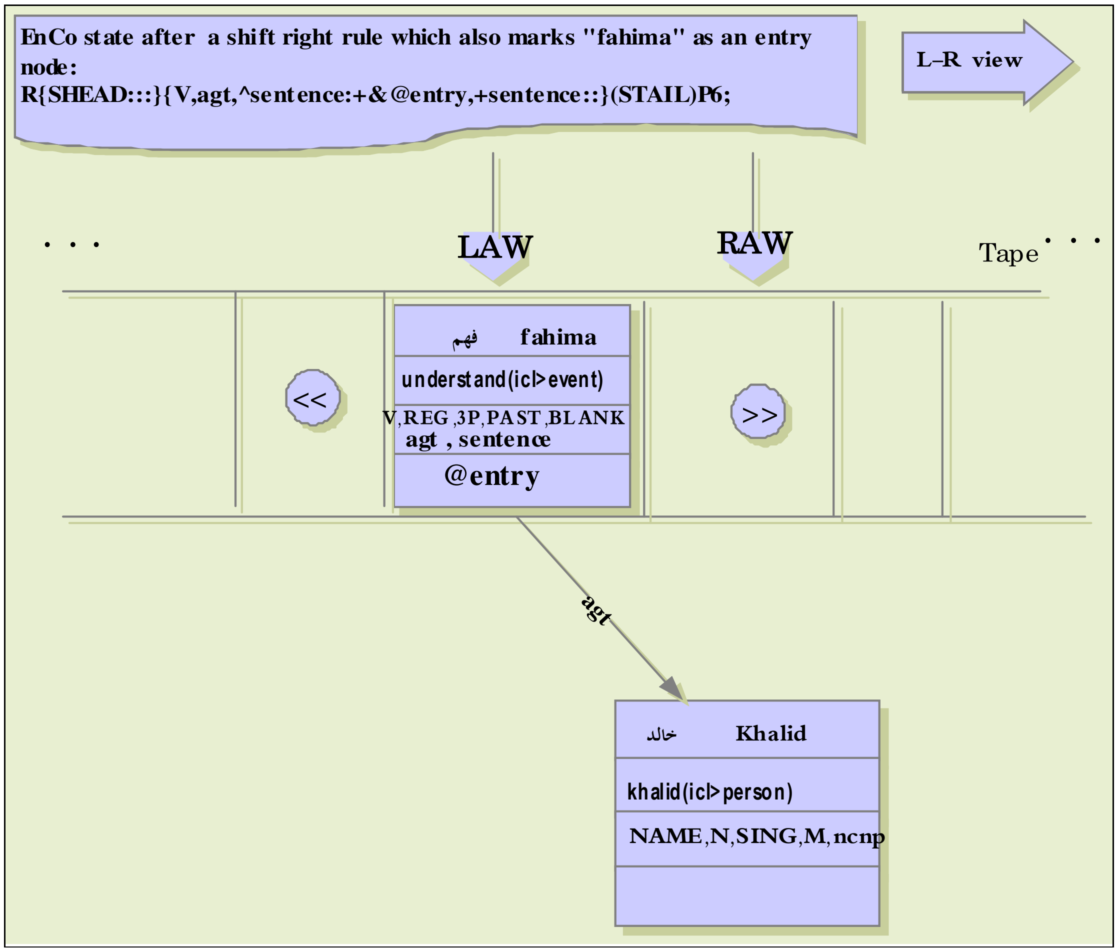


Figure 37: EnCo configuration after another shift right

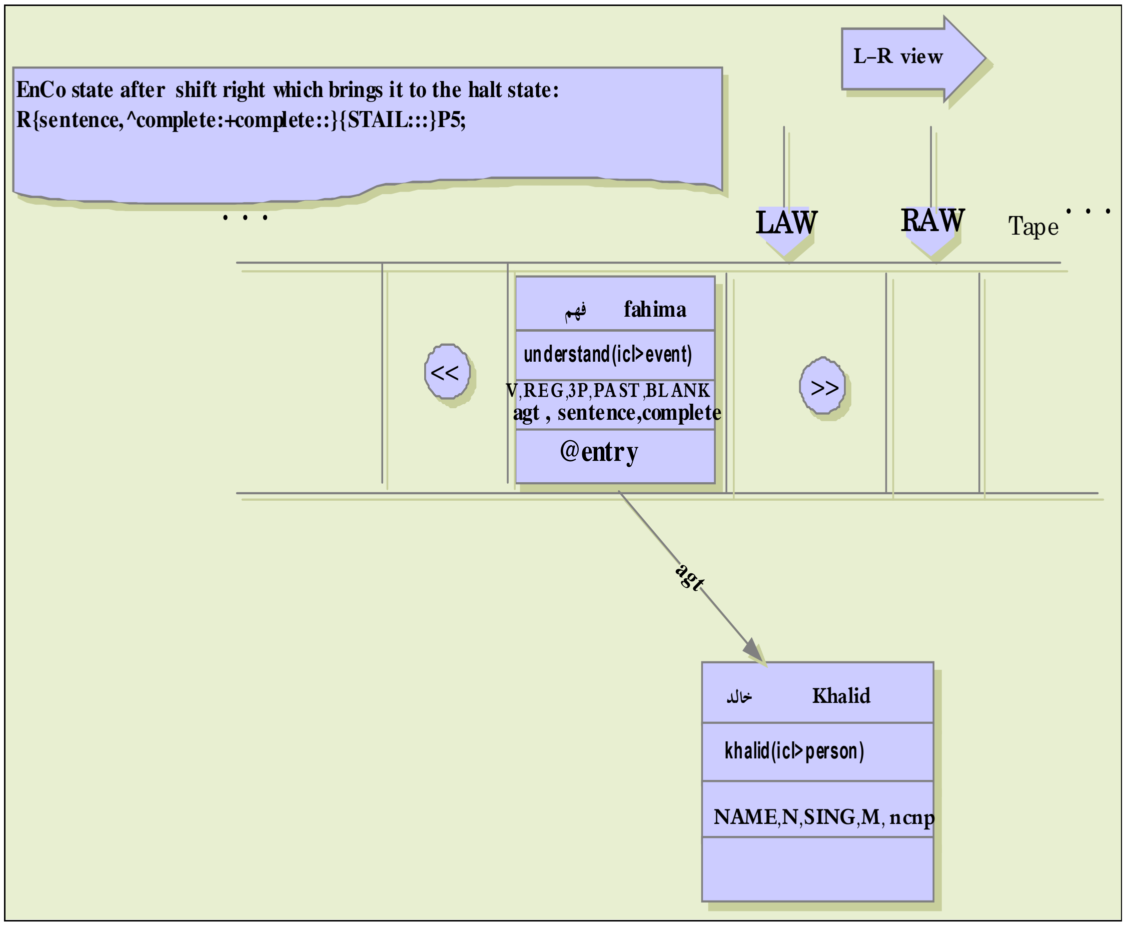


Figure 38: EnCo arriving at a halting configuration

As shown in figure 37, another rule is executed which shifts right and at the same time marks "fahima" as an entry node. Finally, as shown in figure 38, another shift right is applied, forcing EnCo to halt.

4.2 Analysis rules

4.2.1 Format

The analysis rules have the following syntax (EnCo 1999):

<TYPE>...(<PRE2>)(<PRE1>){<LNODE>} {<RNODE>} (<SUF1>) (<SUF2>)… P<PRI>;
Where,

<LNODE>:="{" [<COND1>] ":" [<ACTION1>] ":" [<RELATION1>]":" [<ROLE1>] "}"
<LNODE>:="{" [<COND2>] ":" [<ACTION2>] ":" [<RELATION2>]":" [<ROLE2>] "}"

For a given analysis rule, it is possible to insert or delete only one node into or from the Node-list. Here, LNODE stands for the node under the Left Analysis Window and RNODE for that under the Right Analysis Window.

The meaning of a rule is as follows.

A rule may apply: if under the Left Analysis Window there is a node that satisfies the conditions on <COND1>. And under the Right Analysis Window there is a node that satisfies the conditions on <COND2>. When there are nodes that fulfill the conditions in <PRE> and <SUF> in the order of left and right sides of Analysis Windows respectively, the grammatical features in Analysis Windows are rewritten according to the <ACTION1> and <ACTION2>, respectively.

The operations are done on the Node-list depending on the type of the rule shown in field <TYPE>.

<COND1> and <COND2> stand for lists of grammatical attributes whose presence (ATTR) or absence (indicated by "^ATTR") is required. <ACTION> contains a set of grammatical features to be inserted or deleted from the node under the Analysis Windows. The <RELATION> fields are used to create an arc bearing a UNL relation between the nodes under the Analysis Windows.

<ROLE1> and <ROLE2> are KB attributes which are optional and are not used in this thesis.

<PRI> indicates the priority value of the rules, which is in the range of 0-255. Larger numbers indicate higher priorities. The rules without priority values are deemed to have 0 priority.

To give examples of the rules, consider the above example in more detail:

{fahima Khalid}

The first rule to apply is:

R{:::}{:::}()P1;

R denotes the operation type, which is "shift right". No precondition is set to trigger this rule. Also, no change to the grammatical attributes. P1 indicates that it is a low priority rule.

The next applied rule is:

+{:+BLANK::}{BLK:::}P255;

Type of Operation = "+" which mean combination of right node to left node
Cond1 = nothing
Cond2 = BLK (white space)
Action1 = +BLANK (add the BLANK symbol to the existing list of grammatical attributes or symbols found in the left node)
Action2 = nothing
P255 = Priority 255 (High)

This rule will combine the right node that contains the BLANK (white space) to the left node which contains (fahima). LAW is pointing to SHEAD and RAW is pointing to "fahima".

Another shift right rule will be fired:

R{:::}{:::}()P1;

Figure 35 shows the configuration of EnCo after the execution of the last rule.

The next rule to apply is:

<(^MUQARABA){V,^agt,PAST,^RelPron:+agt::}{ncnp,^TIME,^LPLACE,^MAN,^JAR::agt:}P8 ;

Type of Operation= Left modification "<" (deletes the right node from the node-list, the left node becomes the head of this partial syntax tree and is left in the Node-list; a UNL semantic relation is created according to the designation of relation in the <RELATION> field.

Left condition window = ^MUQARABA (the node under the left condition window should not contain the symbol MUQARABA).

Cond1 = V,^agt,PAST,^RelPron (the node under LAW should contain V and PAST symbols and should not contain agt and RelPron symbols)

Cond2 = ncnp,^TIME,^LPLACE,^MAN,^JAR (The node under RAW should contain the ncnp symbol and should not contain any of the following symbols: TIME, LPLACE, MAN, JAR.

Action1 = +agt (add agt symbol to the left node)

Action2 = nothing

Relation2 = agt (the agt relation from left node to the right node)

The configuration after applying this rule is shown in figure36 .

The UNL expression produced by EnCo for the above phrase is:

```
;======================= UNL ===============
;فهم خالد
[S]
agt(understand:00.@entry.@past, Khalid:04)
[/S]
;=========================================
```

4.2.2 Types of operation

As in classical shift-reduce parsers, the main decision is whether to *shift* or to *reduce*. Here, *reduce* refers to the deletion of a node from the Node-list when it is no longer required. The rule types performing the *shift* process are **L** and **R,** while the rules performing *reduce* are the combination (+ or -) and modification (< or >) rules.

Those are the most used types of operations and we have demonstrated some of them in the above example.

There are other types of operations that have been less frequently used during the development of the Arabic enconverter such as:

"?" Backtrack

Example:

```
?{V:::}{PREF:::}P80;
```

This rule will cause EnCo to backtrack if the left window contains a verb and the right window contains a prefix. To demonstrate its use, in Arabic [ي] has two different interpretations: the first it is a verb prefix, the second it is pronoun which is usually attached to a verb or to a nouns. Hence, if the retrieved dictionary entry [ي] is a prefix for this situation, the rules will force EnCo to backtrack and select the correct entry which is the pronoun.

4.2.3 Scope generation

EnCo provides a mechanism to generate SCOPE constructs. As a matter of fact, UNL graphs can be hypergraphs. A "scope" is a subgraph consisting of all arcs bearing the same "scope id", and the nodes they connect.

A whole scope can play the role of a simple node: an arc can link 2 simple nodes, 1 simple node and a scope, or 2 scopes. The difference with usual recursive graphs is that an arc can go from a node to a node inside a scope to which it does not belong.

The general description format of binary relations for a hyper-node of a UNL expression is the following:

<relation>:<scope-id> (<node1>, <node2>)

Where,

- <scope-id> is the ID for distinguishing a scope and is optional: by default, the arc is in the outermost scope (scope 0 with empty id)
- <node1> and <node2> can be a UW or a <scope node>.
- A <scope node> is written ":<scope-id>".

To demonstrate the scope generation, consider the following example:

لكن نزول خالد لم يفعل الكثير في وقت بدأ المنتخب السعودي يفقد سيطرته على مجريات اللقاء;

But the introduction of Khalid did not do a lot as the Saudi team starts to loose control over the match.

The UNL expression produced by our general Arabic-UNL enconverter is given as graph in figure 39. Its linear form is:

```
[S]
obj(introduction:04, Khalid:09)
agt(do(obj>action):0I.@entry.@not, introduction:04)
man(do(obj>action):0I.@entry.@not, but:00)
aoj:02(saudi:1E, team(equ>squad):16.@def)
mod:02(domination(equ>angel):1Q, his:1V)
mod:01(course of events:21.@entry, game(equ>sport):2A.@def)
obj:02(domination(equ>angel):1Q,:01)
obj:02(lose(icl>event):1L, domination(equ>angel):1Q)
agt:02(start(icl>event):10.@past, team(equ>squad):16.@def)
obj:02(start(icl>event):10.@past, lose(icl>event):1L)
mod:02(as:0T.@entry, start(icl>event):10.@past)
obj(do(obj>action):0I.@entry.@not, much(aoj>quantity):0O)
man(do(obj>action):0I.@entry.@not,:02)
[/S]
```

To generate a scope, the attribute "@entry" is added to the node that will become the entry node of the scope. Then, all of the arcs (bearing semantic relations) directly or indirectly accessible from the entry node are included.

In the above example, the following rule produced SCOPE "01":

<{MASDAR,^obj,after_prep,على:+obj::}{ncnp:+&@entry:obj:}P17;

This rule means:

when

the node on left Analysis Window is a deverbal noun such as (*domination*) and its object is not analyzed yet,

and

the node on right Analysis Window is a noun phrase such as (*course of events*),

then

add the attribute "@entry" to the noun phrase, and a scope will be automatically created. In this scope, the noun on the Right Analysis Window becomes the entry node of this scope, and all relations between nodes accessible from the noun will be included in the scope.

and

link this scope to the verb by "obj" so that it becomes the object of the verb.

SCOPE "02" in the above example is produced by the following rule:

<{V:+man::}{MAN,TYPE_1:+&@entry:man:}P7;

This rule means:

when

the node on the left Analysis Window is a verb such as(*do*),

and

the node on the right Analysis Window is a manner such (*as*)

then

add the attribute "@entry" to the manner phrase so that a scope is automatically created. In this scope, the noun on the Right Analysis Window becomes the entry node, and all relations between nodes accessible from it will be included in the scope.

and

link this scope to the verb by the "man" relation.

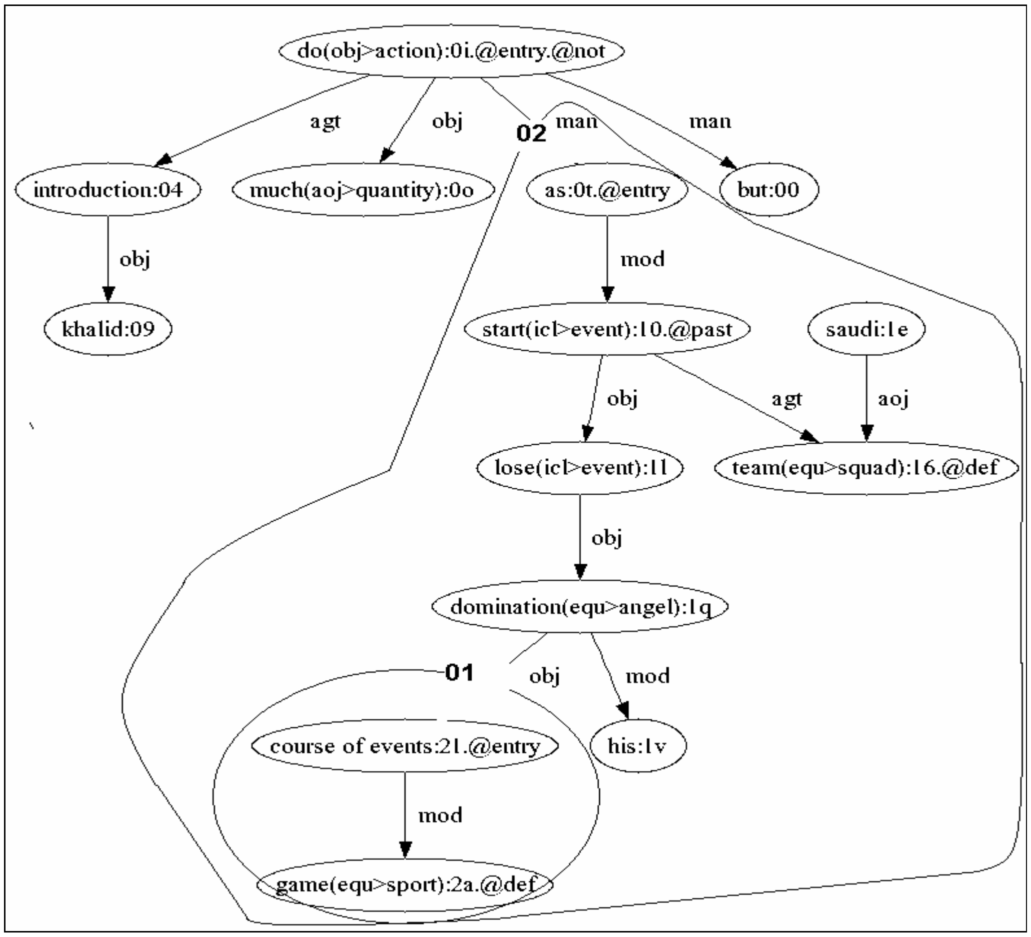


Figure 39: UNL graph with nested scopes

4.3 Overall analysis strategy using EnCo

Writing an enconverter in EnCo, and in particular the rules, is a complicated process that requires a good knowledge of the EnCo structure and rule syntax, the source natural language, and UNL.

4.3.1 Full synchronization with the lexicon

A full synchronization between the rules and the dictionary is necessary, since the execution of rules is highly dependent on the grammatical features specified in the dictionary. This has to be kept in mind when designing the dictionary and the categorization of words.

4.3.2 Control by assigning priorities to rules

In the same line, assigning priorities to rules is essential to avoid wrong results. In the 1500 rules written for Arabic, the prioritization strategy is as follows.

- The highest priority is given to the "removing of space" rule.

- The backtracking rules have the next highest priorities. They are given the highest priority to prevent further execution when a wrong situation or assumption is recognized. They are quite useful in circumstances where we have wrong selection of words.
- The next priority is given to morphological analysis rules. They are important because when they are executed they provide information (by using symbols) that might be useful in executing other rules. For example, an accusative noun cannot be a subject. Morphological analysis is mainly done by combination type rules (+ or -).
- The next priority is given to "left shift" rules. They act as information collectors and are used in controlling and specifying the next rule to fire.
- The next priority is given to the modification rules (<, >), which are, to a large extent, responsible for producing the UNL graph; they depend on other rules.
- The lowest priority is assigned to the "shift right" rules.

4.3.3 Symbol-based control

Developing EnCo rules requires a controlling mechanism that specifies which rule should be fired and which rules should not be fired. For that, we use tactical symbols written or removed from the input tape. Without using the KB (knowledge base), the only way to analyze Arabic is to depend on linguistic knowledge and on what exists in the sentence. Without having this controlling mechanism, this task would be impossible.

For example, suppose we have the following sentence:

ساق خالد السيارة الجديدة بسرعة كبيرة

Khalid drove the new car at a high speed.

To analyze this sentence correctly, we should discover the boundaries of the entities that exist in the sentence. Since "Khalid" is not followed by an adjective, it is allowed to be an agent of the verb "drive" and it is removed from the node-list (tape). On the other hand, since "car" is followed by an adjective which has the same gender, it is not allowed for "car" to be an object before handling the adjective first ("car" is a dependent of "drive", and "new" is a dependent of "car": it is not allowed to process the head before its dependents).

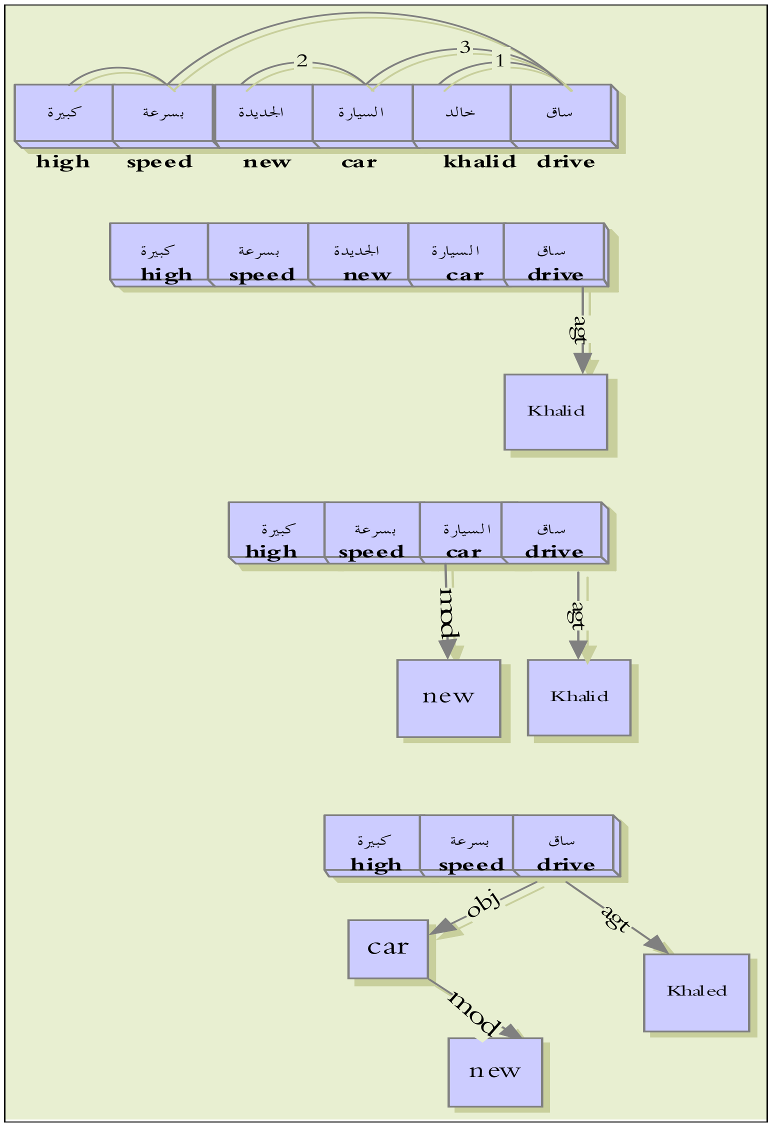


Figure 40: sequence of rule execution example

Chapter 5. The lingware

Introduction

Programming in EnCo is a time and effort consuming process. It is like programming in machine language, in which it is very difficult to control the structure and to use any solid engineering technology. The development environment consists only of a text editor, which in a way is good, but denies the developers of any engineering methodology that ensures the needed quality in a minimal time. Furthermore, with that naive development methodology, it is very hard to maintain an enconverter, especially for persons not having participated to its programming.

In this chapter, we introduce a methodology that helps in systematic EnCo programming. We will also introduce a methodology for diagrammatically specifying a language grammar along with mapping procedures. We will also test this methodology in real sentences from Arabic.

5.1 The problem of the naive development methodology

5.1.1 EnCo rules as a low level formalism for NL parsing

A set of rules written in EnCo constitute a formal system which ensures the unidirectional correspondence between texts and UNL graphs. On the other hand, rules written in DeCo (the other specialized language of the UNL center, used to write deconverters) ensure the correspondence between the UNL graph and the text. Both types of rules formulate the bidirectional formal system between texts and semantic representation (UNL).

Ideally, we should be able to develop at the same time an enconverter (in EnCo) and a deconverter (in DeCo) from the same "static" formal or semi-formal specifications, which themselves should be modular and extensible.

To demonstrate how a language grammar is transformed into EnCo rules, consider the following example:

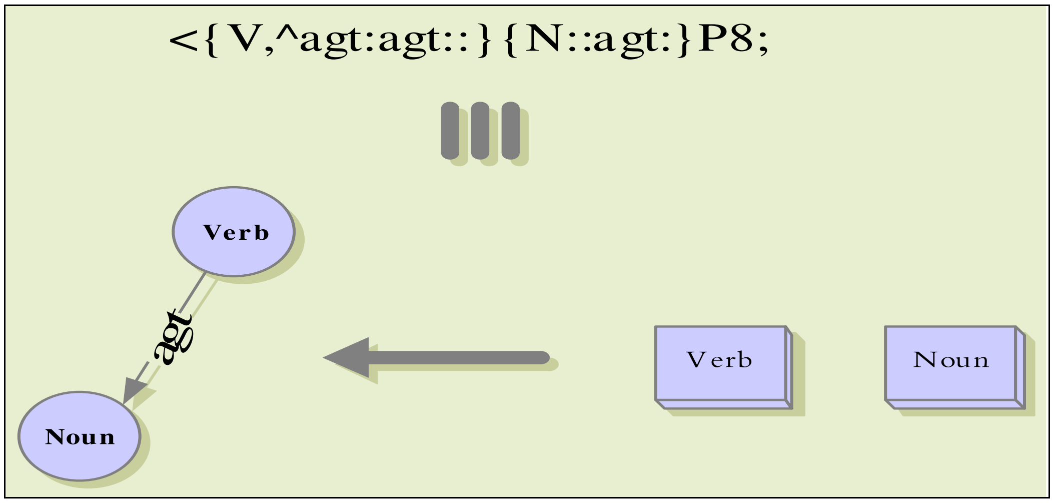


Figure 41: "agt" rule

The rule shown in figure 41 will fire if the LW is a verb with no agent and the right window is a noun. The “noun” node will be removed from the node-list and added to node-net, and an “agt” arc will be added to the node-net between the “verb” node and the “noun” node.

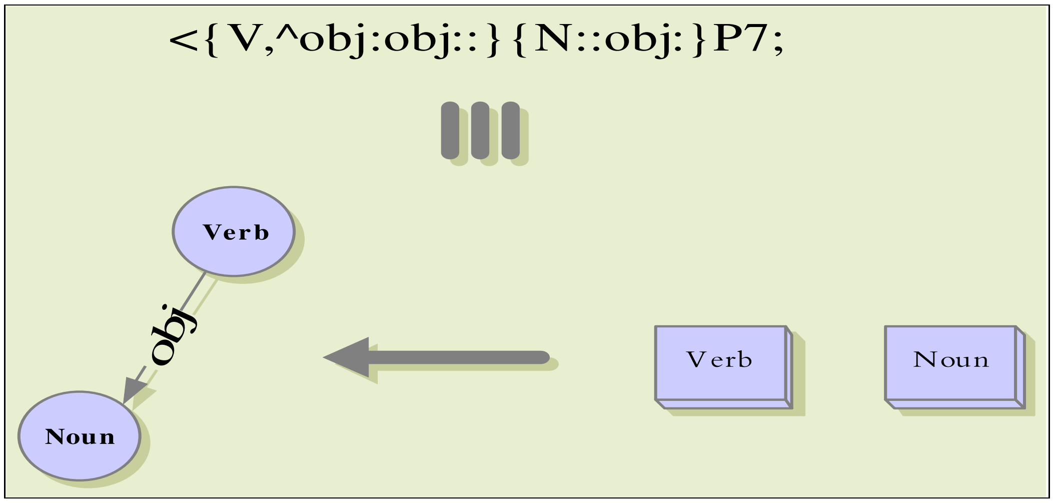


Figure 42: "obj" rule

In the same way, the rule in figure 42 will fire if the LW is a verb with no object and the right window is a noun. The “noun” node will be removed from the node-list and added to the node-net and an “obj” relation will be added to the node-net, connecting the “verb” node to the “noun” node.

The rule shown in figure 43 will build a “mod” relation if the left window is a “noun” and the right window is an “adjective”.

The formalism provided by EnCo rules embeds the language description despite the fact that this description it is not clear or understandable by humans. This is because this formalism is more oriented to the process of building a practical application more than to describing the language. Nevertheless, there is no computable means of extracting “static” specifications from such a program.

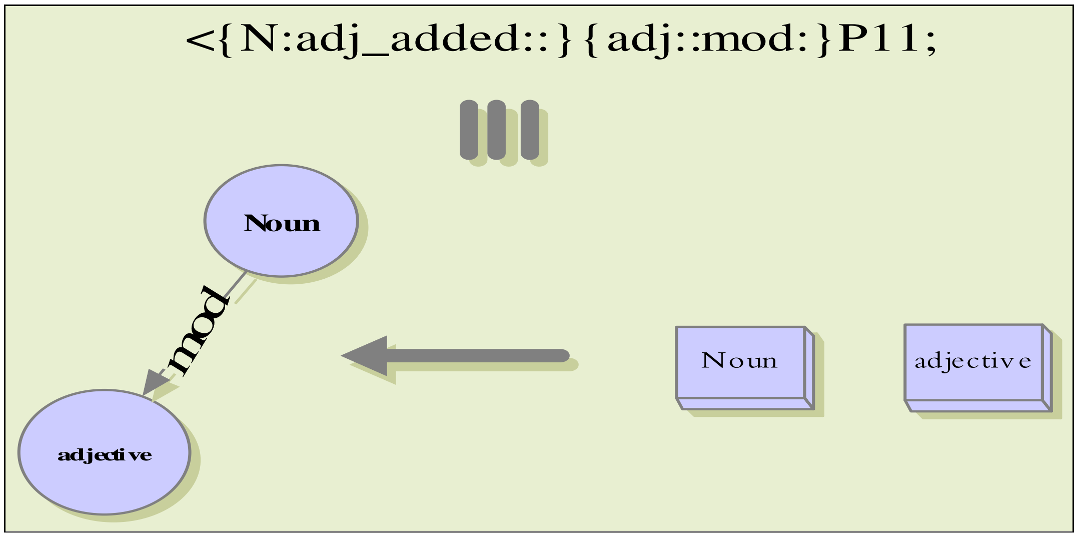
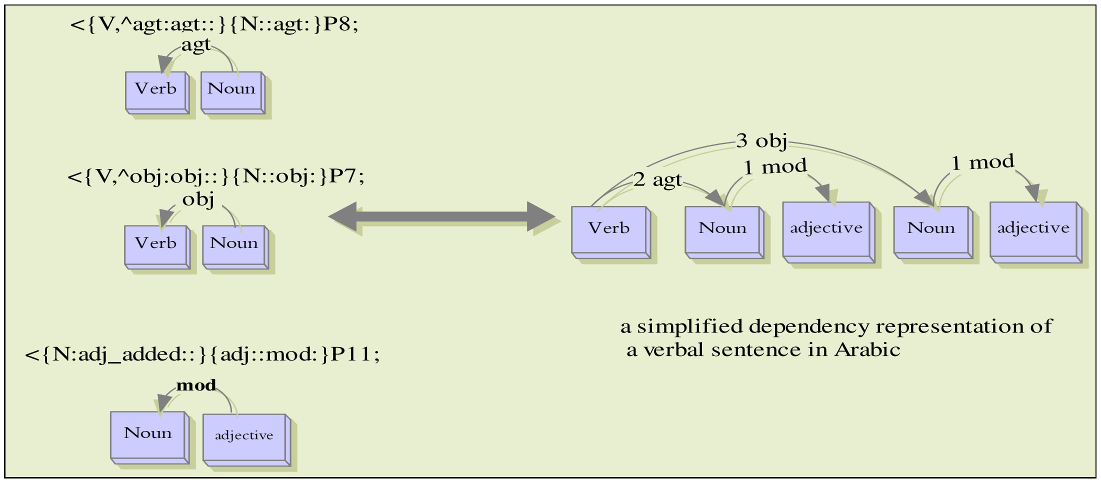


Figure 43: "mod" rule

By combining the set rules in figure 41, 42 and 43 using unification [Kahane 2001], we can generate the grammar shown in figure 44 which is a very simplified description of a verbal sentence in Arabic.

The above example clearly shows that EnCo is oriented towards the production of dependency graphs. However, it is possible to produce constituent structures (phrase-structures-trees) with EnCo. It analyses a sentence by establishing links between individual words and specifying the type of link in each case [Covington 1992]. Each link connects a word (the "head") with one of its "dependents" (an argument or modifier). A head can have many dependents, but each dependent can have only one head. Of course, the same word can be the head in one link and the dependent in another.

Figure 44 shows also that the dependency representation of a sentence (arrows point from each word to its dependents: modifiers or arguments) is inferred from the EnCo rules. On the other hand, this figure and the above example show that it is possible to transfer a dependency grammar into EnCo rules.



<{V,^agt:agt::}{N::agt:}P8;
agt
Verb
Noun
<{V,^obj:obj::}{N::obj:}P7;
obj
Verb
Noun
<{N:adj_added::}{adj::mod:}P11;
mod
Noun
adjective
3 obj
2 agt
1 mod
1 mod
Verb
Noun
adjective
Noun
adjective
a simplified dependency representation of
a verbal sentence in Arabic

Figure 44: the bidirectional mapping of EnCo rules and DG

Looking carefully at each rule, we find that it establishes a linking between two words, one is dependent on the other. Some links are shown clearly in the UNL- graph; others are implicit and are used within rules only. Each rule also indicates head-dependent order which is very important in specifying the word order typology. Dependent-Dependent order (the mutual order of two dependent of the same head) is specified by the priority strategy or by using symbols.

In the above example, the "agt" rule has a higher priority than "obj" rule, reflecting the fact that the subject of a verb is before its object.

In EnCo, this dependent-dependent order can be implemented alternatively by using symbols. As an example, consider the following two rules:

<{V,^agt:agt::}{N::agt:}P8;

<{V,^objt,agt:obj::}{N::obj:}P9;

The second rule executes after the first one independently of their priorities. This is because the "agt" symbol is added after the first rule and is a condition of the second rule. This shows how dependent-dependent relation can be implemented.

As we have seen in the previous chapter, there is no intermediate representation between the text and the output graph. EnCo takes the input text and transforms it into the corresponding UNL graph directly. It is the responsibility of the rules to ensure the right sequence of execution as we have shown previously.

EnCo provides two mechanisms to ensure the right execution of the rules: rule prioritization and use of tactical symbols. The developers have to use them correctly as EnCo does not provide any other means to assist or to enforce this mechanism.

5.1.2 Problems of the current methodology of rules encoding

EnCo is a true SLLP (specialized language for linguistic programming), but still of quite "low level" in the hierarchy of programming models[13]. It does not provide an integrated development environment to assist the developers to encode the rules correctly by controlling the symbols, or by providing any debugging capabilities.

Writing EnCo rules tends to become unmanageable as their number grows. Usually such rules are written by a small group of people. These people are the only ones who are able to understand its structure. If these people are assigned to another position or leave their job, the size as well the complexity of the lingware prevents anyone from gaining insight into it.

[13] In particular, it has no typed variables, and no control structures.

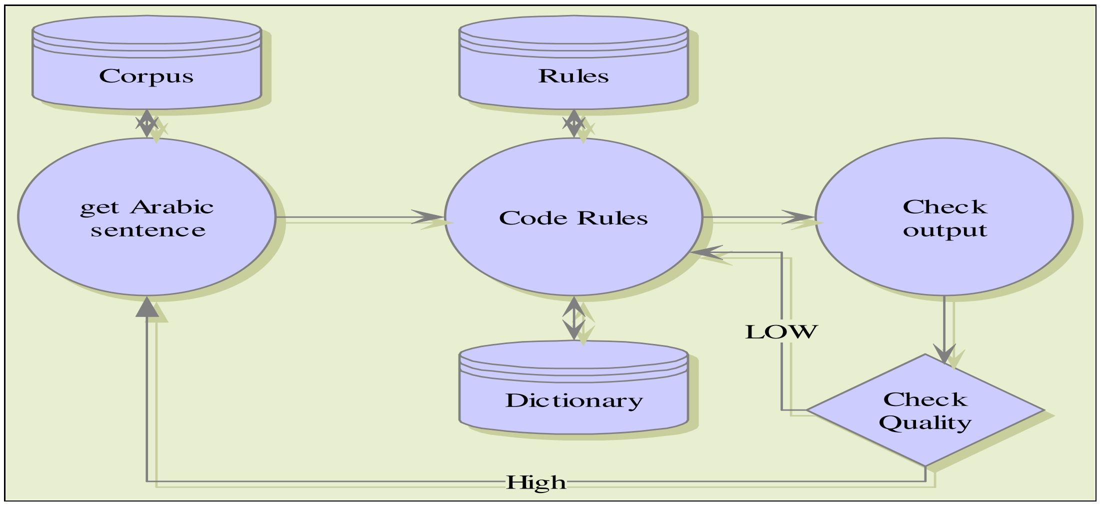


Figure 45: iterative methodology of writing rules

The iterative methodology of writing rules shown in Figure 45 proved to be inadequate. In this methodology of development, the rules are written for each sentence in the corpus and there is no guarantee that any modification will not have harmful side effects. The iterative methodology might work for small systems or prototypes, but to create robust systems with some industrial quality level, one should use methodologies used in software engineering.

5.1.3 Introducing a lingware engineering approach for EnCo

Therefore, a systematic development methodology is necessary to produce the natural language representation of a sentence using a transductive formalism based on dynamic rules. The main function of this methodology is to specify diagrammatically the language grammar using language components, which are entities that embed syntactic and semantic information that can be identified in the source language from their unique function in the sentence and to relate these diagrams to rules or sets of rules in the SLIP at hand (here, EnCo). These diagrams can be integrated into one development environment enabling systematic development of rules and ease of maintenance.

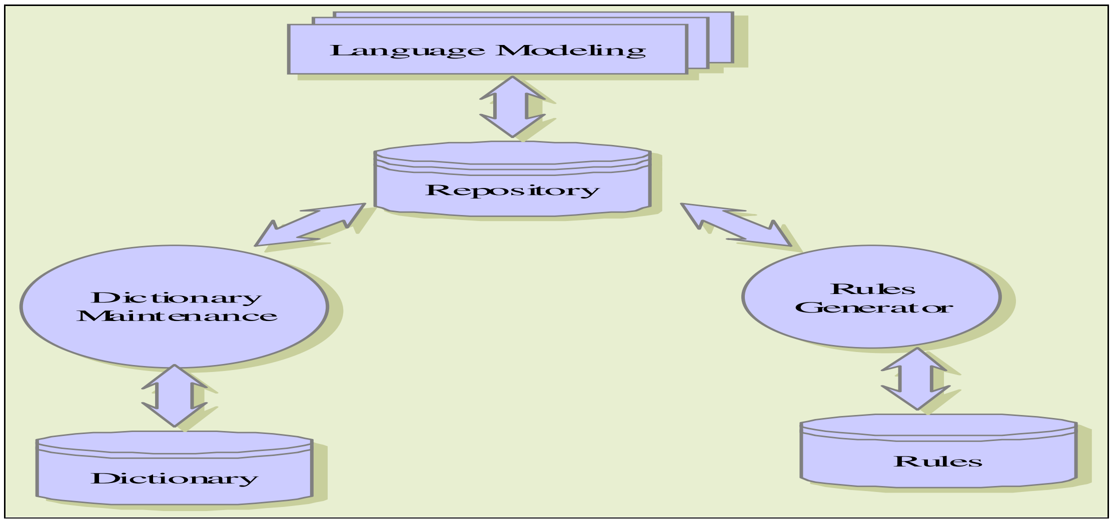


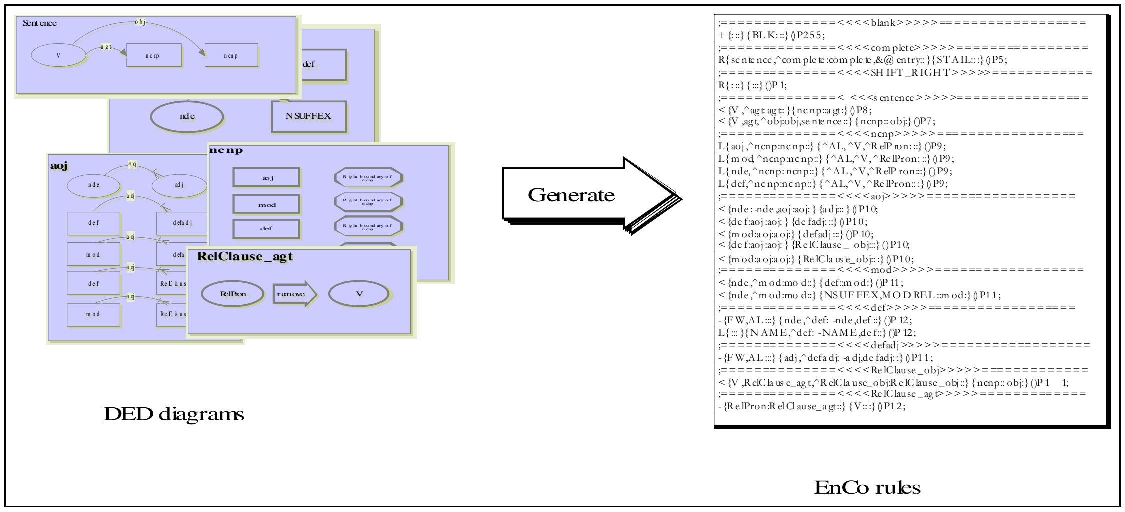
Figure 46: the architecture of CARE

5.2 The basics of our methodology (DED)

The proposed methodology name has been dubbed Daoud's EnCo Diagrams (DED). It uses diagrams to specify language grammars. In the following sections, we also describe a Computer-Aided Rules Engineering (CARE) that uses DED. It is an integrated environment, which provides a set of tools for the production and maintenance of the rules and the dictionary items.

5.2.1 General description

The aim of a CARE environment is to help the developer to build the rules for analysis of language with the minimum effort and time. It allows the user to design the rules graphically using DED. It also allows the user to store, modify and generate the rules from these diagrams.



```
;================<<<<blank>>>>>==================
+ {: ::} {BLK: ::} ()P255;
;================<<<<complete>>>>>==================
R{sentence,^complete:complete,&@ entry:: }{STAIL:: :} ()P5;
;================<<<<SHIFT_RIGHT>>>>>==============
R{: ::} {:::} ()P 1;
;================< <<<sentence>>>>>==================
< {V ,^agt: agt:: }{ncnp::agt:} ()P8;
< {V ,agt,^obj:obj,sentence ::} {ncnp:: obj:} ()P7;
;================<<<<ncnp>>>>>====================
L{aoj,^ncnp:ncnp::} {^AL,^V,^RelPron: ::} ()P9;
L{mod,^ncnp:ncnp::} {^AL,^V,^RelPron: ::} ()P9;
L{nde,^ncnp: ncnp::} {^AL,^V,^RelPron:::} ()P9;
L{def,^ncnp:ncnp::} {^AL,^V,^RelPron:: :} ()P9;
;================<<<<aoj>>>>>=====================
< {nde : -nde ,aoj :aoj: } {adj::: } ()P10;
< {def:aoj :aoj: } {defadj: ::} ()P10;
< {mod:aoj:aoj:} {defadj :::} ()P 10;
< {def:aoj :aoj: } {RelClause_ obj:::} () P10;
< {mod:aoj:aoj:} {RelClause_obj:: :} ()P10;
;================<<<<mod>>>>>====================
< {nde,^mod:mod::} {def::mod:} ()P 11;
< {nde,^mod:mod::} {NSUFFEX,MODREL ::mod:} ()P11;
;================<<<<def>>>>>=====================
- {FW,AL :::} {nde,^def: -nde,def ::} ()P 12;
L{::: }{NAME,^def: -NAME,def::} ()P 12;
;================<<<<defadj>>>>>===================
- {FW,AL :::} {adj,^defadj: -adj,defadj: :} ()P11;
;================<<<<RelClause_obj>>>>>==============
< {V ,RelClause_agt,^RelClause_obj:RelClause_obj ::} {ncnp:: obj:} ()P1 1;
;================<<<<RelClause_agt>>>>>==============
- {RelPron:RelClause_agt::} {V:: :} ()P12;
```

Figure 47: rule's generation from DED

5.2.1.1 *High-level specification using diagram*

DED provides the necessary graphical specification for building a verified specification of the needed linguistic knowledge for the desired application. As we will see later, the DED diagrams are oriented to the EnCo environment. By using language components, DED emphasizes the modularity approach.

5.2.1.2 *Semi-automatic generation of low level code*

This linguistic knowledge specified by DED is transformed into EnCo rules. This process can be fully automatic, semi- automatic or manual. We will show later how the transformation is possible from DED to EnCo rules. We will also show how this process is accomplished.

5.2.2 Architecture of the proposed system

Basically, the proposed system consists of the following main components:

- Language Modeling
- Repository
- Rules Generator
- Dictionary Maintenance

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

As shown in figure 46, linguistics knowledge, UNL, and mapping rules are stored in the repository.

The repository is then interfaced with a rules generation component that facilitates the automatic production of the EnCo rules.

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

5.2.3 Grammar specification methodology (DED)

The suggested methodology implemented by the DED is to define COMPONENTS interacting with each other to model the sentence. This work is inspired by the static grammar formalism and methodology [Vauquois and Chappuy 1985]. The notion of frame is similar to the notion of component in [Vauquois and Chappuy 1985].

5.2.3.1 Language modelling using Diagrams

DED models language by using components and relations.

Components

Components are entities that embed syntactic and semantic information that can be identified in the source language from their unique function in the sentence. There are different types of components:

Dictionary components (DC). The simplest form of all components and are derived directly from grammatical attributes found in the dictionary. Examples: V (verb), nde (non definite entity), NAME (named entity), adj (adjective). A DC is depicted as an oval shape with a specified name.

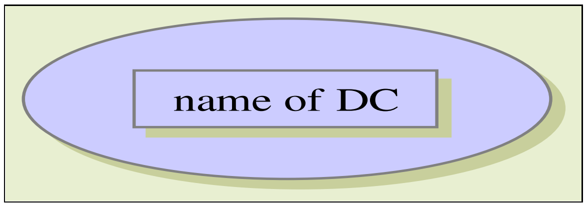


Figure 48: Dictionary Component

Composite Component (CC). This is a complex type of component that may consist of other CCs, or DCs or other types of DED components. It also models the operations and relations connecting theses constituents. A CC is depicted by a rectangular shape as shown in figure 49.

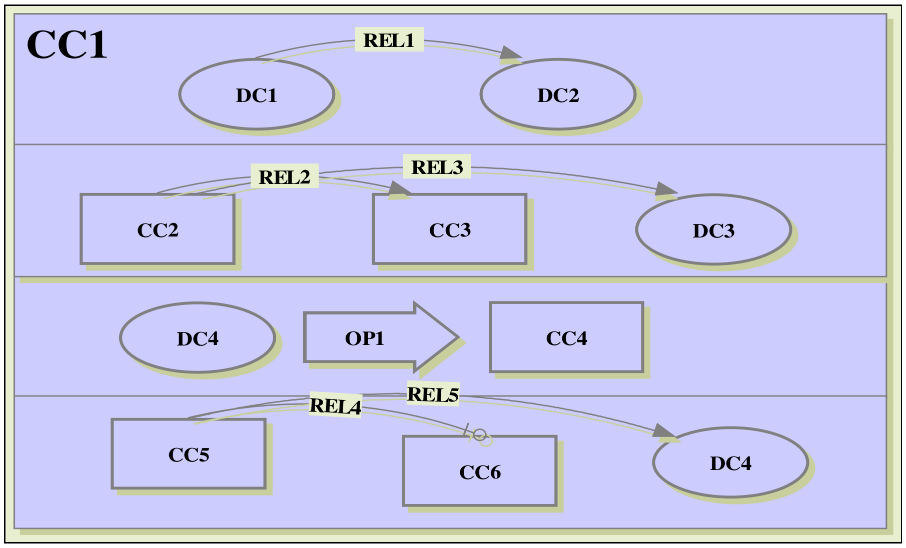


Figure 49: an example of a Composite Component

It is possible that a CC contains different structures as demonstrated in Figure 49. In the above figure CC1 is a composite of:

DC1 and DC2
Or,
CC2, CC3 and DC3
Or,
DC4 and CC4

Conditional Component (CnC). This type of components stores conditions only. It is depicted by an octagonal shape.

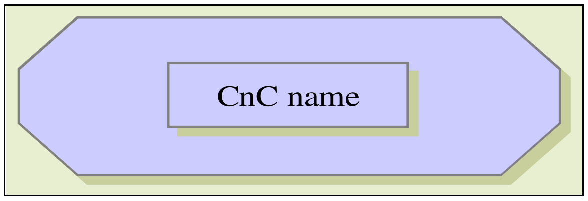


Figure 50: Conditional Component

Relations

A relation links two components. The direction of the link is specified along with its name. The name should be selected from the UNL inventory of relations. As shown in figure 51, REL1 is linking DC1 to DC2: DC1 is the head component and DC2 is the dependent.

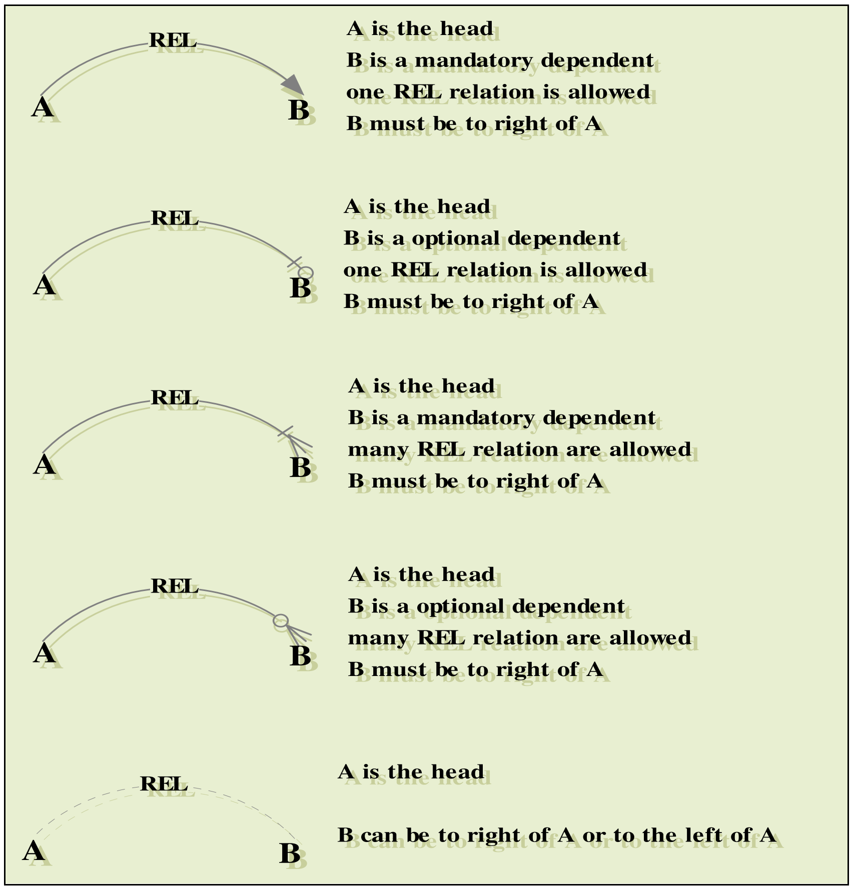

REL
A
B
A is the head
B is a mandatory dependent
one REL relation is allowed
B must be to right of A
REL
A
B
A is the head
B is a optional dependent
one REL relation is allowed
B must be to right of A
REL
A
B
A is the head
B is a mandatory dependent
many REL relation are allowed
B must be to right of A
REL
A
B
A is the head
B is a optional dependent
many REL relation are allowed
B must be to right of A
REL
A
B
A is the head
B can be to right of A or to the left of A

Figure 51: types of relations

Operations

Specify an operation that does not generate any UNL relations, such as removing a node or inserting a node as shown in figure 52. The direction of operation decides which node to remove or to keep.

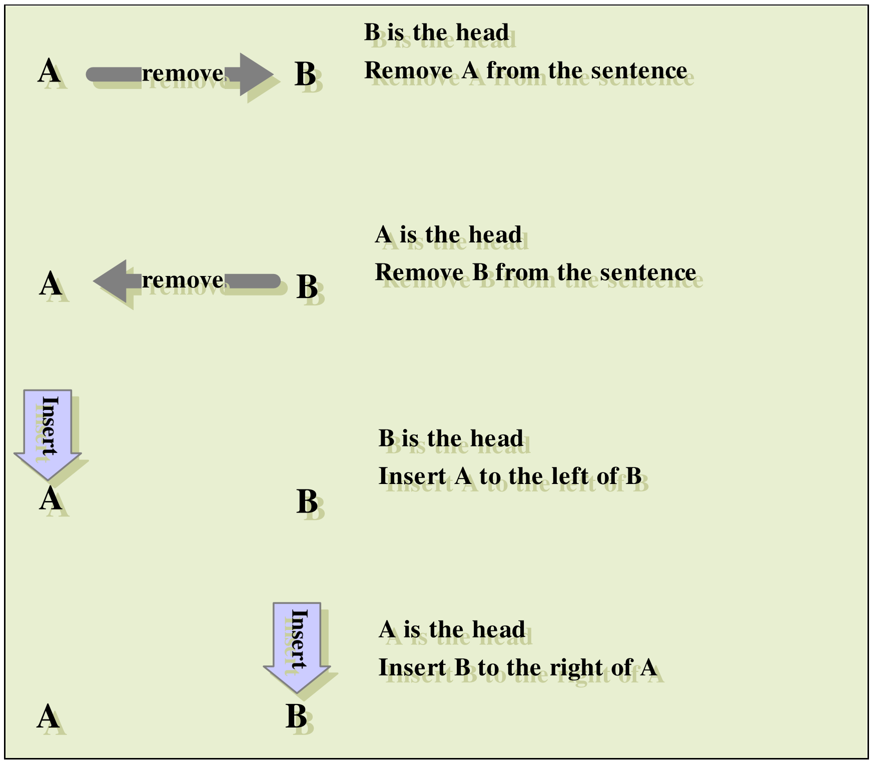

B is the head
Remove A from the sentence
A
remove
B
A is the head
Remove B from the sentence
A
remove
B
Insert
B is the head
Insert A to the left of B
A
B
Insert
A is the head
Insert B to the right of A
A
B

Figure 52: types of operations

5.2.3.2 Generating EnCo rules from DED

The mapping uses both the prioritization and symbol based-control to produce the corresponding rules from DED. The rule is not to process a head before its dependent(s).

In DEDs we have two types of dependencies: internal and external.

The internal dependencies (head-dependent order, dependent-dependent order) are links and relations within the CC, and they specify the head (source) components and their dependents (destination). If a relation is connecting a head with a dependent, the parsing process removes the dependent and updates the node-net.

On the other hand, the external dependencies are the relations between a referring CC and a referent one. In figure 49, CC1 refer to CC2, hence CC2 should be available before CC1. Internal dependencies (horizontal) specify what to parse, on the other hand external dependencies (vertical) specify how and when to parse. Both are important in the DED methodology.

To demonstrate the mapping process, consider the first row of CC1 shown in Figure 49, which illustrates a link between two DCs. The generated rule for this specification is:

```
<{DC1,^REL1:CC1,REL1::}{DC2::REL1:}()P8;
```

DC1 and ^REL1 are the conditions on the left window (head window), and DC2 is the condition on the right window (dependent). This rule removes DC2 and engages it into a REL1 relation with the head. The addition of CC1 to the action part of the head node informs other CCs that this component has become CC1 and should be used on this basis. REL1 is a symbol that demonstrates the fact that CC1 is engaged in a REL1 link. The Condition "^REL1" prevents the head node from having another relation of type "REL1".

For the second row in CC1, we have two types of orders: head-dependent and dependent-dependent orders. The following are the generated rules for CC1 definition.

```
<{CC2,^REL2:REL2::}{CC3::REL2:}()P8;
<{CC2,REL2,^REL3:CC1,REL3::}{DC3::REL3:}()P8;
```

In the first rule CC3 removed, and a relation is built from CC2 to CC3. Additionally, in the action part of the left window, the REL2 symbol is added. In the second rule, DC3 is removed and a relation is built from CC2 to DC3. To ensure the right dependent-dependent order, the REL2 symbol is required to appear in the left window of the second rule. This forces the second rule to be fired after the first one.

Note also that the CC1 symbol (which is the name of the component) is added to the action part of the head window (in this example the left window) of the second rule to mark the new name of the component. CC1 is not added by the first rule because the new name is valid only after the completion of the second rule.

In the third definition of CC1 shown in figure 49, DC4, which is a dependent, is removed. This is translated in EnCo as

```
-{DC4:::}{CC4:CC1::}P8;
```

It is also optionally possible to remove the first name CC4 from the head node as following:

```
-{DC4:::}{CC4:CC1, -CC4::}P8;
```

The fourth specification shown in figure 49 contains an optional relation. This is translated in EnCo as:

```
<{CC5,^REL4:REL4::}{CC6::REL4:}()P8;
<{CC5,REL4,^REL5:CC1,REL5::}{DC4::REL5:}()P8;
<{CC5,^REL4,^REL5:CC1,REL5::}{DC4::REL5:}()P7;
```

This says that if CC6 exists, then it should be between CC5 and DC4. This structure is handled by the first two rules. On the other hand, if there is no CC6 in the sentence, the third rule will apply.

Finally, the components that are at the lowest level of the hierarchy should have the highest priority in the generated rules and vice-versa. This is logical since the rules at the highest level depend on them.

5.3 A full example

To explain the DED methodology, we will use it to write the grammar for the Arabic verbal sentence. We will produce the rules based on this grammar. Finally, we will test these rules on a long Arabic sentence to show the efficiency of this approach.

5.3.1 Specification diagrammatically of a verbal sentence grammar with DED

A sentence in Arabic may have the following structure:
Verb Agent Object
Here we may have a component which we will call *sentence*. In figure 53, it consists of:

"*V*" denotes a verb component which should be a dictionary item. "*ncnp*" denotes a "non complete noun phrase" and is a composite component.

The graphical representation shown in figure 53 specifies the head and the dependent along with head-dependents, dependent-dependent relations.

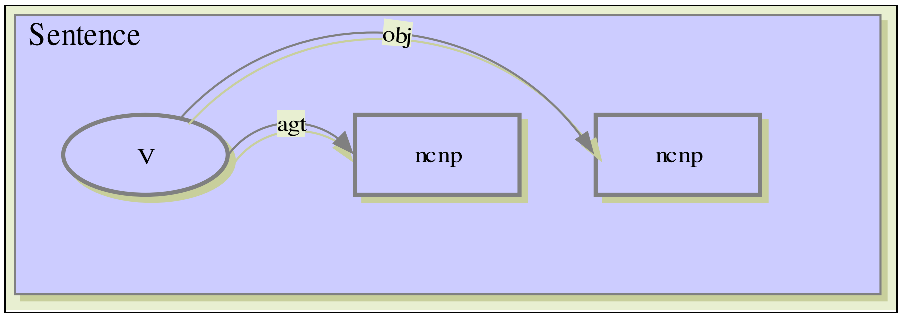


Figure 53: the sentence component

The information presented in figure 53 is sufficient to produce the following rules:

```
;==============<<<<sentence>>>>>================
R{SHEAD:::}{V,agt,obj,^sentence:+&@entry,sentence::}()P6;
;==============<<<<agt>>>>>==================
<{V,^agt:agt::}{ncnp::agt:}()P8;
;==============<<<<obj>>>>>==================
<{V,agt,^obj:obj::}{ncnp::obj:}()P7;
```

The sentence component is virtually created when the sentence symbol is added to the head component (the verb). Also, we see how is dependent- dependent order is specified as we have explained earlier.

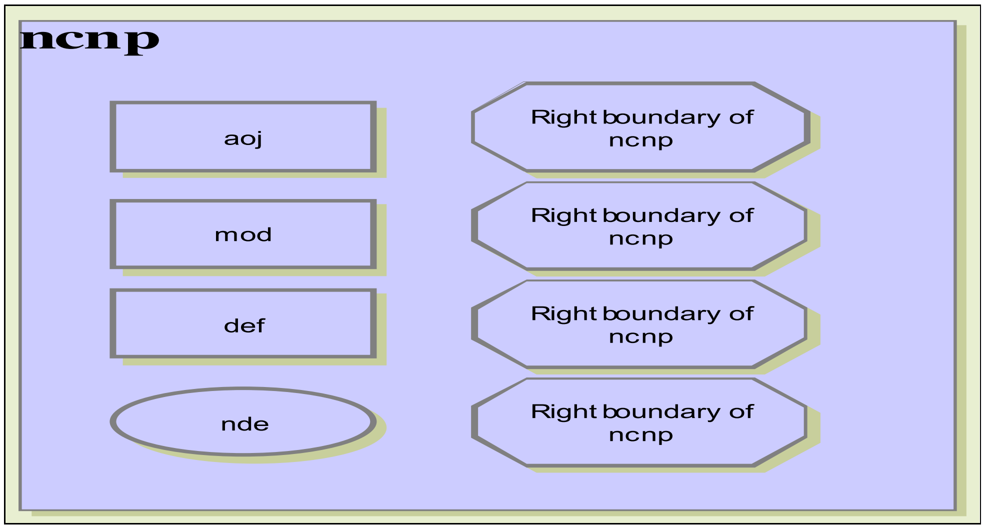


Figure 54: the definition of a "non complete noun phrase"

The symbol "ncnp" stands for "non complete noun phrase". In Arabic this could be:

1	nde		(indefinite noun) dictionary component
2	def	(definite noun)	complex component
3	mod	(IDAFA)	complex component
4	aoj		(adjective phrase) a complex component

Example of the above forms of ncnp:

Man, boy for "1"
The man, the boy "2"
The door of the house (باب البيت) for "3"
The tall man (الرجل الطويل) for "4"

The graph in figure 54 shows the specification of a ncnp component. One important observation is that there are no dependency relations between any of its constituents. The "right boundary component of ncnp" only specifies when the ncnp ends. It is a conditional component containing a list of conditions setting such that all must be satisfied by the node to the right of the ncnp component. In our example, it should not be a verb, not an article and not a relative pronoun.

The generated rules for this model are implemented using the "left shift" operation. These rules show that the ncnp symbol is added when the condition on the right node (RAW) is fulfilled.

```
;==============<<<<sentence>>>>===================
R{SHEAD:::}{V,agt,obj,^sentence:+&@entry,sentence::}()P6;
;==============<<<<agt>>>>===================
<{V,^agt:agt::}{ncnp::agt:}()P8;
;==============<<<<obj>>>>===================
<{V,agt,^obj:obj::}{ncnp::obj:}()P7;
;==============<<<<ncnp>>>>===================
L{aoj,^ncnp:ncnp::}{^AL,^V,^RelPron:::}()P9;
L{mod,^ncnp:ncnp::}{^AL,^V,^RelPron:::}()P9;
L{nde,^ncnp:ncnp::}{^AL,^V,^RelPron:::}()P9;
L{def,^ncnp:ncnp::}{^AL,^V,^RelPron:::}()P9;
```

The next step is to specify the "aoj" component:

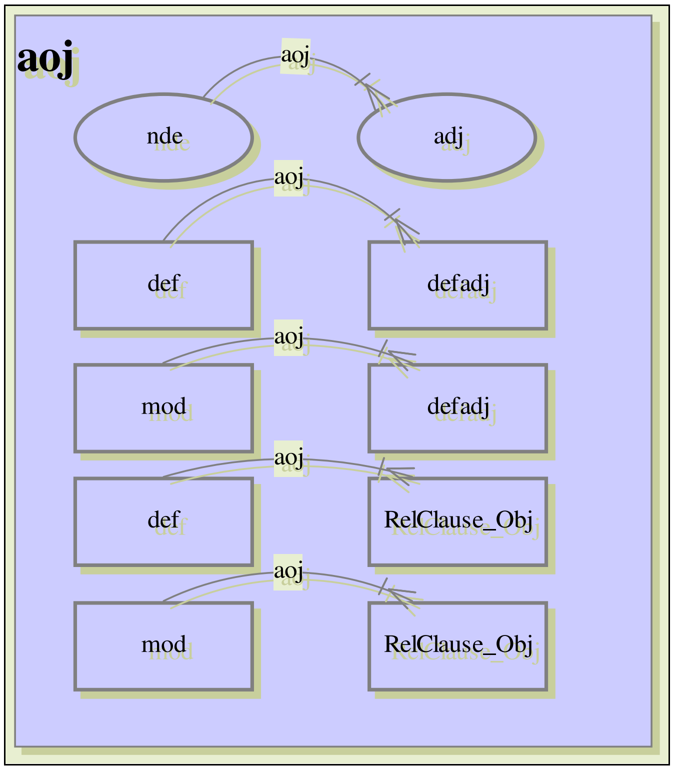


Figure 55: aoj component

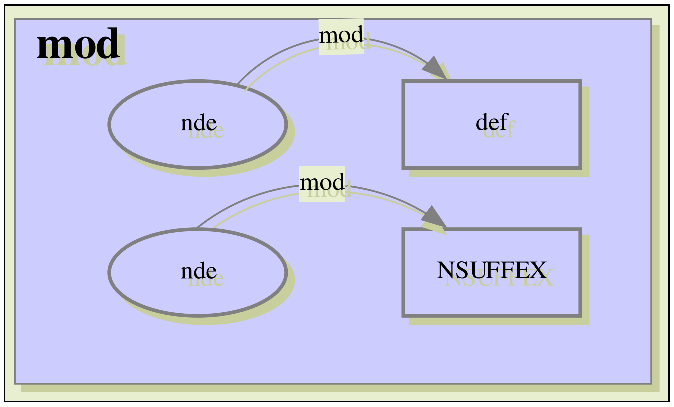


Figure 56: mod component

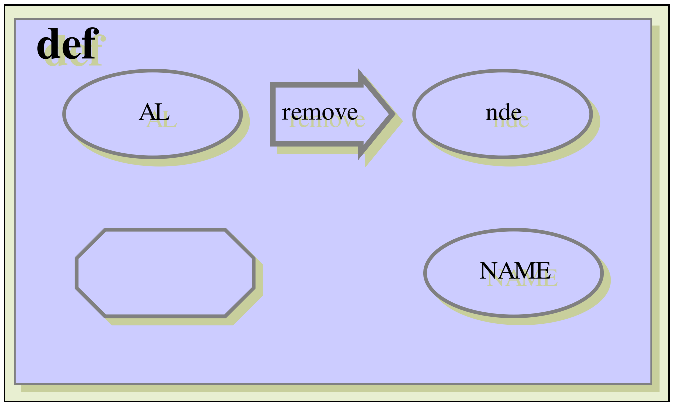


Figure 57: definite component

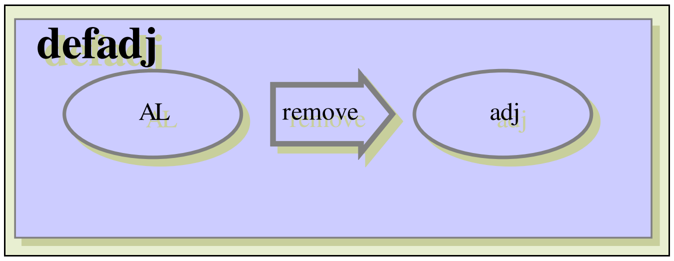


Figure 58: definite adjective component

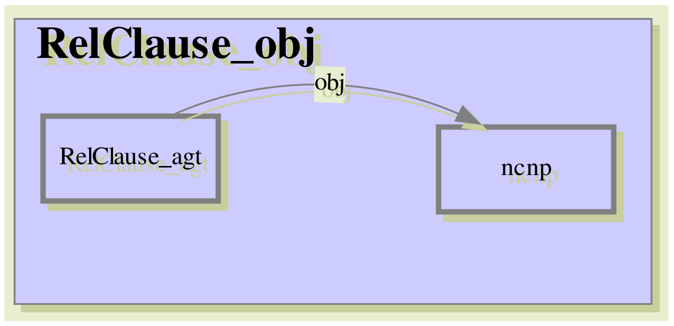


Figure 59: RelClause_obj component

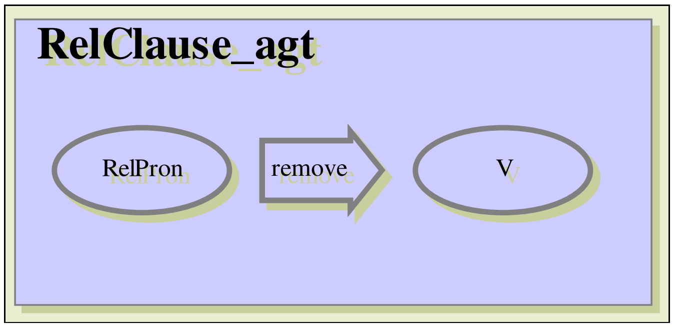


Figure 60: RelClause_agt component

5.3.2 Mapping these specification into EnCo rules

The output rules for the above model, generated based in our methodology, are:

```
;=============<<<<sentence>>>>==================
R{SHEAD:::}{V,agt,obj,^sentence:+&@entry,sentence::}()P6;
;=============<<<<agt>>>>==================
<{V,^agt:agt::}{ncnp::agt:}()P8;
;=============<<<<obj>>>>==================
<{V,agt,^obj:obj::}{ncnp::obj:}()P7;
;=============<<<<ncnp>>>>==================
L{aoj,^ncnp:ncnp::}{^AL,^V,^RelPron:::}()P9;
L{mod,^ncnp:ncnp::}{^AL,^V,^RelPron:::}()P9;
L{nde,^ncnp:ncnp::}{^AL,^V,^RelPron:::}()P9;
L{def,^ncnp:ncnp::}{^AL,^V,^RelPron:::}()P9;
;=============<<<<aoj>>>>==================
<{nde:-nde,aoj:aoj:}{adj:::}()P10;
<{def:aoj:aoj:}{defadj:::}()P10;
<{mod:aoj:aoj:}{defadj:::}()P10;
<{def:aoj:aoj:}{RelClause_agt:::}()P10;
<{mod:aoj:aoj:}{RelClause_agt:::}()P10;
;=============<<<<mod>>>>==================
<{nde,^mod:mod::}{def::mod:}()P11;
<{nde,^mod:mod::}{NSUFFEX,MODREL::mod:}()P11;
;=============<<<<def>>>>==================
-{FW,AL:::}{nde,^def:-nde,def::}()P12;
L{:::}{NAME,^def:-NAME,def::}()P12;
;=============<<<<defadj>>>>==================
-{FW,AL:::}{adj,^defadj:-adj,defadj::}()P11;
;=============<<<<RelClause_obj>>>>==================
<{V,RelClause_agt,^RelClause_obj:RelClause_obj::}{ncnp::obj:}()P11;
;=============<<<<RelClause_agt>>>>==================
>{RelPron:::}{V:RelClause_agt::}()P12;
```

To complete this process, the developer should add three more components:

- *a complete* component which informs the EnCo that the analysis is completed, this component uses the *sentence* component.
- a SHIFT_RIGHT component, to shift right analysis window if no rules applicable. This rule should have the lowest priority.
- a BLANK component used to remove spaces to the right of a word. This is an independent rule of highest priority.

The complete set of rules generated is:

```
;=============<<<<blank>>>>==================
+{:::}{BLK:::}()P255;
;=============<<<<complete>>>>==================
R{sentence,^complete:complete::}{STAIL:::}()P5;
;=============<<<<SHIFT_RIGHT>>>>==================
R{:::}{:::}()P1;
;=============<<<<sentence>>>>==================
R{SHEAD:::}{V,agt,obj,^sentence:+&@entry,sentence::}()P6;
```

```
;=============<<<<agt>>>>>==================
<{V,^agt:agt::}{ncnp::agt:}()P8;
;=============<<<<obj>>>>>==================
<{V,agt,^obj:obj::}{ncnp::obj:}()P7;
;=============<<<<ncnp>>>>>==================
L{aoj,^ncnp:ncnp::}{^AL,^V,^RelPron:::}()P9;
L{mod,^ncnp:ncnp::}{^AL,^V,^RelPron:::}()P9;
L{nde,^ncnp:ncnp::}{^AL,^V,^RelPron:::}()P9;
L{def,^ncnp:ncnp::}{^AL,^V,^RelPron:::}()P9;
;=============<<<<aoj>>>>>==================
<{nde:-nde,aoj:aoj:}{adj:::}()P10;
<{def:aoj:aoj:}{defadj:::}()P10;
<{mod:aoj:aoj:}{defadj:::}()P10;
<{def:aoj:aoj:}{RelClause_agt:::}()P10;
<{mod:aoj:aoj:}{RelClause_agt:::}()P10;
;=============<<<<mod>>>>>==================
<{nde,^mod:mod::}{def::mod:}()P11;
<{nde,^mod:mod::}{NSUFFEX,MODREL::mod:}()P11;
;=============<<<<def>>>>>==================
-{FW,AL:::}{nde,^def:-nde,def::}()P12;
L{:::}{NAME,^def:-NAME,def::}()P12;
;=============<<<<defadj>>>>>==================
-{FW,AL:::}{adj,^defadj:-adj,defadj::}()P11;
;=============<<<<RelClause_obj>>>>>==================
<{V,RelClause_agt,^RelClause_obj:RelClause_obj::}{ncnp::obj:}()P11;
;=============<<<<RelClause_agt>>>>>==================
>{RelPron:::}{V:RelClause_agt::}()P12;
```

5.3.3 Testing the generated rules

Using these set of rules we have been able to analyze long sentences such as:

فهم صديق داود الذكي الكتاب المفيد
Daoud's clever friend understood the useful book[14].

```
[S]
mod(friend:04, daoud:09)
aoj(clever:0G, friend:04)
aoj(useful:0T, book:0M)
agt(understand:00.@entry, friend:04)
obj(understand:00.@entry, book:0M)
[/S]
```

[14] we use here simplified UWs, without constraints lists.

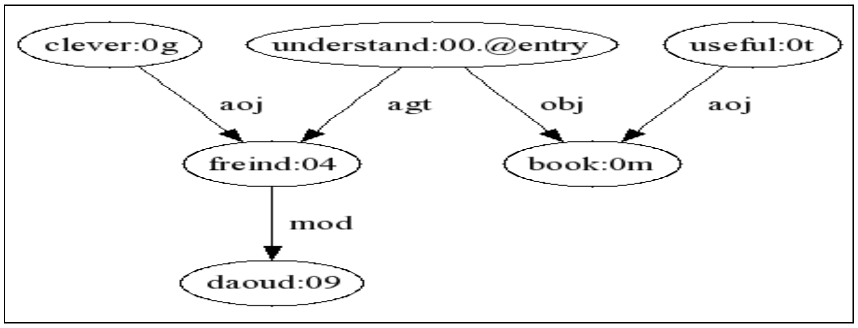


Figure 61: UNL graph for "Daoud's clever friend understood the useful book"

فهم خالد الذكي الذي قابل محمد القوي الذي قرأ كتاب داود المفيد الرائع الدرس
The clever Khalid who met the strong Moh'd who read Daoud's fantastic useful book understood the lesson.
[S]
aoj(clever:0B, Khalid:04)
aoj(strong(icl>state):0V, mohammad:0P)
mod(book:18, daoud:1D)
aoj(useful:1K, book:18)
aoj(fantastic:1Q, book:18)
obj(read:14, book:18)
aoj(read:14, mohammad:0P)
obj(meet:0K, mohammad:0P)
aoj(meet:0K, Khalid:04)
agt(understand:00.@entry, Khalid:04)
obj(understand:00.@entry, lesson:1X)
[/S]

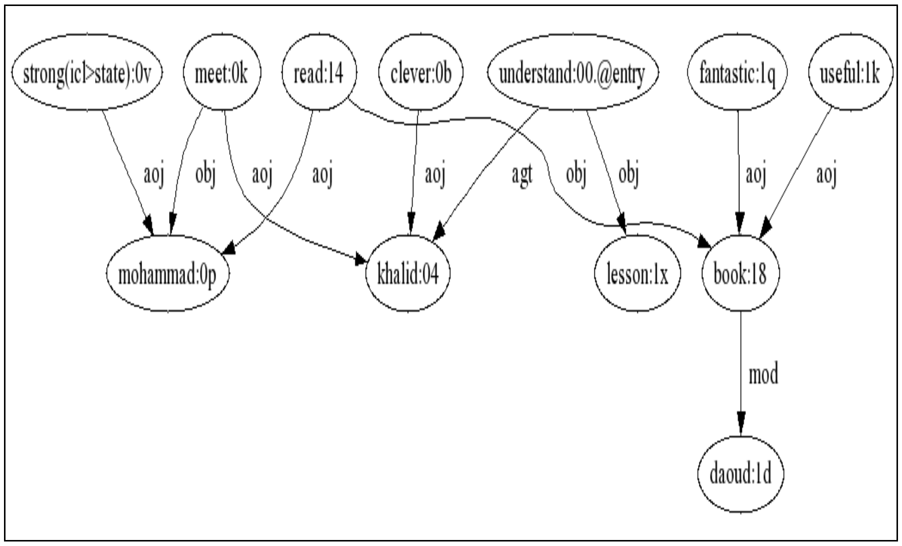


Figure 62: UNL graph for "The clever Khalid who met the strong Moh'd who read Daoud's fantastic useful book understood the lesson"

Chapter 6. Content Extraction in Arabic

Introduction

Content extraction is the most important component of the CATS system. The success of our experiment is highly dependent on this part. We have shown that choosing the right approach is decisive in successfully implementing NL systems. The input of this phase is natural spontaneous SMS text, and the output is CRL-CATS expressions.

This is a very challenging task specifically for a language such like Arabic. Add to this the noisy nature of the input text. We will also discuss how we adapted our general purpose translator to perform a domain-specific extraction. Additionally we will show how to perform this task using a semantic grammar. Finally, we will give some examples and demonstrate problems and limitations of the system.

6.1 Challenges of CE in Arabic

CE from Arabic SMS presents not only the usual problems encountered when handling western languages, due to several characteristics:

1. because people usually don't write the "small vowels", an orthographic word is much more ambiguous than in English, French, Italian, etc.
2. in some domains, such as Cars, there are many foreign words, which are transliterated in many different ways in the Arabic script by posters.

Let us detail the challenges we encountered by order of cruciality, as seen from a developer point of view. The main difficulty for us was the absence of freely usable lexical and syntactic resources and tools: Arabic is still a "pi-language" (poorly informatized). The other difficulties concern the treatment of named entities, the problem posed by spelling variations (dictionary size, need to handle "unknown" forms of known words), the free word order, and the presence of unpredictable long compound words.

6.1.1 Arabic is not a resource-rich language

Not all languages have received equal investment in linguistic resources and tool development [Riloff, et al. 2002]. As an example, most of the research published on IE discusses problems related to English, which is a resource-rich language. While some of the existing English-based IE systems performance is comparable to that of human experts, by comparison, Natural Language Processing (NLP) in the Arabic language is still in its initial stage [Hammo, et al. 2002].

Regarding Content Extraction, Arabic was not one of the languages considered in the MUCs events (1987-1998). In this competition, English has always been the unique target language, with the exception of MUC-6 (where Spanish and Chinese were considered as well [Dini 1998].

6.1.2 Named entities

On the level of entity extraction, Named Entities (NE) were defined as proper names and quantities of interest. Person, organization, and location names were marked as well as dates, times, percentages, and monetary amounts.

Classified ads are rich in proper names (such as cars makes, models and locations names), which in Arabic are not distinguished by using upper case letters like English. This makes them more difficult to locate in Arabic text than in English text [Abuleil and Evens 1998].

6.1.3 Spelling variations

The variations of spelling of the Arabic text add more challenges to the processing of Arabic text. As an example people, write Alef letter "ا"، or with Hamza (ء) over it "أ" or under it "إ. Also, we fin confusions between the Ha' "ه" and Ta' "ة'", and between Ya' "ي" and Alef-Maqsoura "ى". Another problem is the wrong insertions of spaces. In Arabic, spaces are normally used to separate words. Most Arabic letters are connected from both sides (cursive writing system), causing them to have different shapes depending on their positions (first, middle, or last). But the letters "و", "ر" ,"ز" , "د", "ى" and "ذ" can be connected only from the right side, making their shapes unchanging at any position of the word. After any of these letters, people tend to wrongly insert a space, or to (also wrongly) omit it (e.g., "أبو بكر" or "أبوبكر" {Abu-Baker}).

The inconsistency of the Arabic spelling of transliterated proper nouns is a major challenge. This appears frequently in the classified ads text where many of the proper names (car make and model as an example) are transliterated from other languages. This phenomenon is noticeable within unedited and spontaneous classified ads, reflecting the cultural and educational background of the text writer. For example, the car-make CITROEN has different spellings in our corpus: {SATARWEN}"سترون", {SA:TERWEN} "ساترون " , {SATERWE:N} "ستروين", {SA:TERWE:N} "ساتروين ", {SE:TERWE:N} "سيتروين".

6.1.4 Free word order

Arabic is a language with extremely free word order, as many richly inflected languages. Verbs often start sentences and sometimes come after the subject.

<Verb><Subject><Object>

<Subject><Verb><Object>

The first order is the default word order. The second one is used when putting greater attention to the subject, for example when the answer to the question is the subject. The identification of the subject and object depends on the semantics and inflectional endings.

The classified ads sublanguage inherited this feature from Arabic as we have shown in Chapter 2.

6.1.5 Arabic compound forms

Arabic uses a diverse system of prefixes, suffixes, and pronouns that are attached to the words, creating compound forms that further complicate text manipulation. For instance, articles such as "an" and "the" are not separate words as they are in languages like English but are actually appended to the words to which they refer (for example, "their two cars" is written as a single token, سيارتيهم).

This can cause ambiguity of forms (especially if short vowels are omitted). As an example, in Arabic,"ستروين " has two interpretations, which are (you will see) and (CITROEN a name of a car brand).

6.2 The CE CATS structure

Taking into consideration all the above mentioned problems and challenges in developing this significant component. On the other hand, we are interested in developing this experimental system with the minimum investment, time and efforts, while maintaining the overall robustness of the final system, to prove our assumptions and hypothesis.

In this section we will describe the rule-based CE component. We will demonstrate the use of EnCo as a CE engine and the development of rules using the DED methodology.

6.2.1 Feasible approaches and selected approach

IE approaches can be divided into knowledge engineering, rule-based approaches and automatically trainable approaches [Neumann and Xu 2004]. In the fist approach, grammars and rules are constructed by hand. The second approach learns rules from annotated corpora using statistical methods.

Automated learning procedures have proven effective for content extraction from semi-structured text, and material with a regular, repetitive structure. For more complex extraction tasks on free text, where there is greater linguistic variation, they have had some success, but so far they are not able to create complete extraction systems comparable in performance to manually developed systems [Grishman 2003].

We conclude from our review of the literature that the rule-based approach is more suitable for building CATS. An automatically trainable approach cannot be as accurate as a rule-based approach and requires a huge set of structured or semi-structured data as training corpus, and is not available in our case.

6.2.2 Using EnCo to write a CE and not a parser

[Sophie AUBIN, et al. 2005] show how they adopt a general purpose parser for extracting information in a restricted domain. They argued that, since parsing is domain and language-dependent, a general parser must be adapted to each given sublanguage.

We have chosen to write our CE in EnCo because it was available and we could reuse and adapt to this new context (CE) what we had already developed while writing an Arabic-UNL enconverter (development methodology, dictionary and rules).

The task is different: we are not trying to translate the classified posts into another language, but we want to transform the posts into a higher abstraction that captures the meaning of the sentence, regardless of the original surface form. The results will be a CRL-CATS expression (already descried in chapter 3). Syntactically, UNL and CRL-CATS are similar, but they depict different information. UNL expresses a (deep) linguistic-semantic structure while CRL-CATS captures only the meaning.

EnCo has shown a great flexibility in handling Arabic which is a free order language. The parsing mechanism which is based on dependency analysis is more efficient than using phrase structure grammars [Covington 1992] in handling that type of language. Additionally, the implementation of the longest match algorithm suits the processing of Arabic composite forms and compound words.

The EnCo rules use the grammatical attributes derived initially from the lexicon. They determine the action plan of the EnCo as presented earlier. On the other hand, the UNL relations determine the shape of the output.

In the case of CE, the attributes don't concern deep syntax (abstract time, aspect, modality, number, definiteness, etc.), but they are used to point to semantic classification of a sublanguage. The relations are no more semantic relations linking abstract lexemes, as in UNL (agent, object, goal, manner, etc.), but they become the property names in CRL-CATS.

In this way, it is possible to use EnCo to parse SMS Arabic language with the intention of producing a CRL-CATS expression, and not a UNL graph. To do this, we cannot use the full analysis rules and the associated dictionary. We have to develop a new rules based on the analysis of the classified ads sublanguage and to collect a new dictionary (or adopt the existing dictionary) to reflect the semantic classes of the domain.

6.2.3 Structure of dictionary

The dictionary of CATS is manually constructed for the Cars and Real Estate domains. It is the backbone of CATS since it drives the CE process, compensates for lexical inconsistency by providing synonym relations and by connecting words to concepts (CWs), and finally provides the semantic information needed for reasoning.

6.2.3.1 Mapping words to concepts

Different word forms are connected to one concept. A concept is a meaning pointed to by the CW. In a sense, a CW denotes a unique meaning while an unrestricted UW can denote to different word senses [Sérasset and Boitet 2000].

This structure minimizes the effect of the alternative representations of text (including different orthographic forms, spelling errors, and abbreviations) on the overall performance of the system, specifically in the searching process.

The number of CWs in the dictionary for both domains is 10828, while the total number of lexical forms is 30982. On average around 3 forms point to the same CW.

The entries for the dictionary are collected from the corpus and many are generated automatically as we will see in the coming sections.

6.2.3.2 Semantic classification

Each concept has a semantic category which is the direct outcome of the study of the corpus. These categories form the basis of the CE process as we have shown earlier. For example in the Cars domain we have semantic categories for: vehicle type, car manufacturer, model, color, motor size unit, motor size hint word, price hint word, currency, features etc.

In the same manner, we have different categories for the Real Estate domain such as: property type, area hint, area unit, locations, floor hint, bedroom hint, feature, etc.

Each classification is denoted by a unique symbol which is attached to each lexical item in the dictionary. These symbols are loaded from the dictionary during the CE process.

```
[دبليو بى ام]{}"B.M.W(country<germany,country<europe)"(make,nmodel,car) <A,3,3>;
[دبليو بى أم]{}"B.M.W(country<germany,country<europe)"(make,nmodel,car) <A,3,3>;
[بى.ام.دبليو]{}"B.M.W(country<germany,country<europe)"(make,nmodel,car) <A,3,3>;
[بى.أم.دبليو]{}"B.M.W(country<germany,country<europe)"(make,nmodel,car) <A,3,3>;
[دبليو بي ام]{}"B.M.W(country<germany,country<europe)"(make,nmodel,car) <A,3,3>;
[دبليو بي أم]{}"B.M.W(country<germany,country<europe)"(make,nmodel,car) <A,3,3>;
[بي.ام.دبليو]{}"B.M.W(country<germany,country<europe)"(make,nmodel,car) <A,3,3>;
[بي.أم.دبليو]{}"B.M.W(country<germany,country<europe)"(make,nmodel,car) <A,3,3>;
```

6.2.4 Using a "Semantic grammar" for CE

The DED methodology described in the previous chapter is applicable to our strategy for CE, in defining the "semantic grammar" which will drive the process of "domain analysis". We will use the same diagrams and symbols in defining "knowledge components" (the analogs of language components).

The basic element used to describe our semantic grammar is the Knowledge Component. A classified ads post is a group of KCs, and each can be further decomposed until we reach the atomic structure of knowledge such as: numeral values, year hint word, and currency unit word.

The main function of this methodology is to specify diagrammatically the natural language patterns using knowledge components (KC), which are entities that consists of groups of words that can be identified in the natural language text from their unique function in a particular domain.

Through the Knowledge Component, we can specify adjacency relations, priority levels, constraints, and conditions.

A Cars classified ad may contain the following components:

- Main domain object (MDO): is a DC (saloon, pickup, bus, etc)
- Sale: a DC (for sale, looking to sell,)
- Want: a DC (wanted, looking for, looking to buy,)
- Price: a KC
- Year: a KC
- Country: DC (Japan, Germany, France…)
- Color: a KC
- Model: a KC
- Make: DC

Each component has one or more surface structures (variants), but serves one knowledge function. As an example, each of the following phrases:

Table 17: variants of motor size

محرك 1500 سي سي	Motor is 1500 cc
1500 سي سي	1500 cc
سعة المحرك cc1500	motor size is 1500
1.5 ليتر	1.5 L

has a different syntactic structure, but all hold the same knowledge, which is described by:

Mot (car, 1500)

In the same context, consider the following phrases found in the Real Estate sub-domain that is describing a property:

Table 18: variants of property type component

تصلح لبناء بيت	……fit to build a house,………
تقع وسط فلل	……..Surrounded by villas ……….
سكني	………residential ……..
في منطقة سكنية	……….Located in a residential area,………
لبناء بيت او فيلا	….. to build a house or villa ………..

All of the above phrases indicate that the property is residential (not commercial or industrial) which is expressed in CRL-CAT as:

Typ(land, residential)

Defining a Knowledge Component involves specifying the knowledge function and the surface structure(s) that might appear in a free text within a particular domain.

Each surface structure consists of simple components derived from the dictionary and/or composite ones.

To show how to use DED in describing the grammar of the classified ads domain, consider the following table.

Table 19: variants of price component

السعر 500 دينار	The price is 500 dinar	Pri(flat,500@dinar)
500 دينار	500 dinar	Pri(flat,500@dinar)
السعر 500	The price is 500	Pri(flat,500)
لا يزيد السعر عن 500 دينار	Not more the price 500 dinar	Pri(flat,500@dinar@less)
السعر لا يزيد عن 500	The price is not more 500	Pri(flat,500@less)
500 دينار و اكثر	500 dinar and above	Pri(flat,500@dinar@more)
500 و مافوق دينار	500 and above dinar	Pri(flat,500@dinar@more)

All of the above forms should produce the same CRL-CATS expression (with or without the necessary attributes). It is not trivial to extract content correctly from the above phrases due to their variable word order. We will now show how the modular approach of DED can help in correctly generating EnCo rules that able to process the above structures correctly.

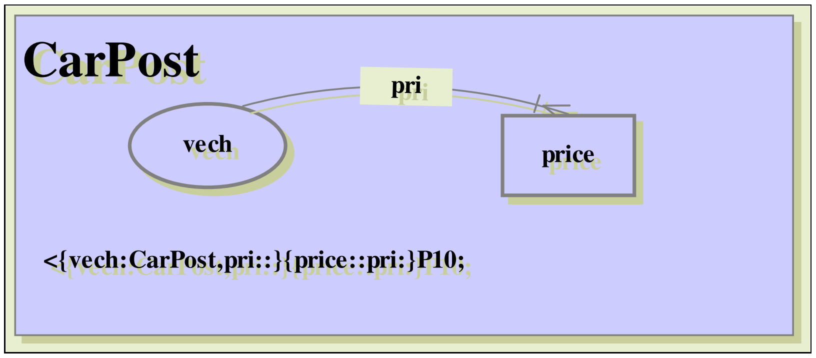


Figure 63: a Car Post KC

Figure 63 shows one possible description of Car Post KC. The headword is the MDO which is a DC (vech denotes any type of cars); the dependent is the *price* KC. In this description, we allowed the head to have many *price* components. Also the above figure shows the corresponding rule.

To define the *price* KC, it is necessary to understand its structure first. The core of this component is the number. The number is disambiguated by the existence of *PriceHint* word before and/or a *currency unit* after. Additionally, the Comparatives (if any) either surround the number, or are positioned before the *PriceHint* or after the *currency unit* as shown in table 19.

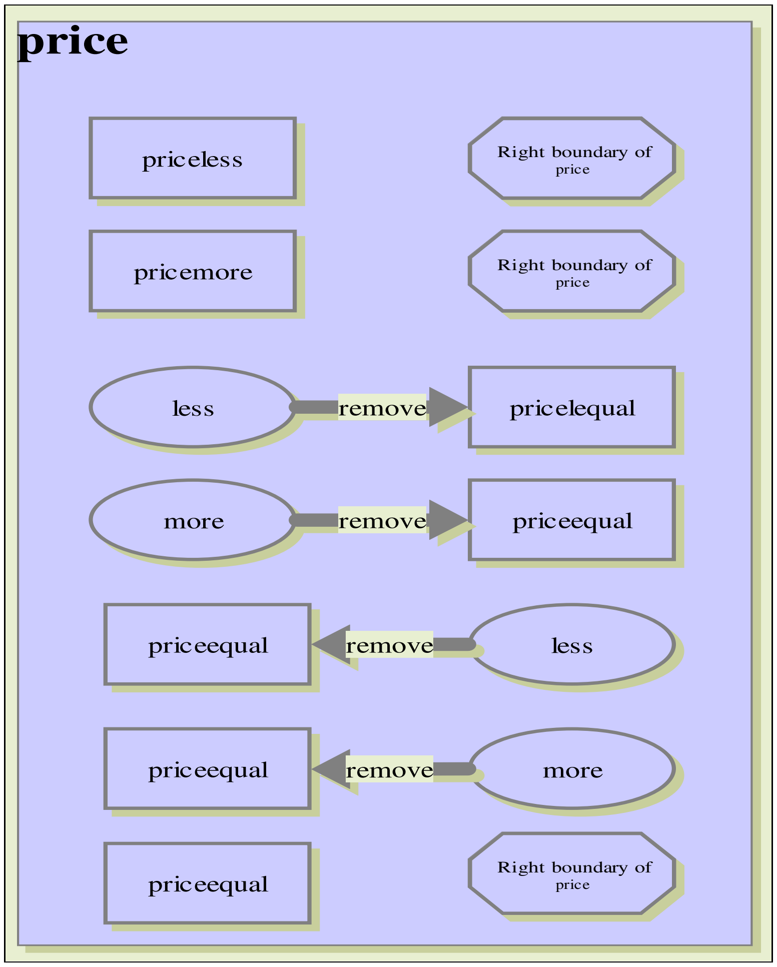


Figure 64: the price KC

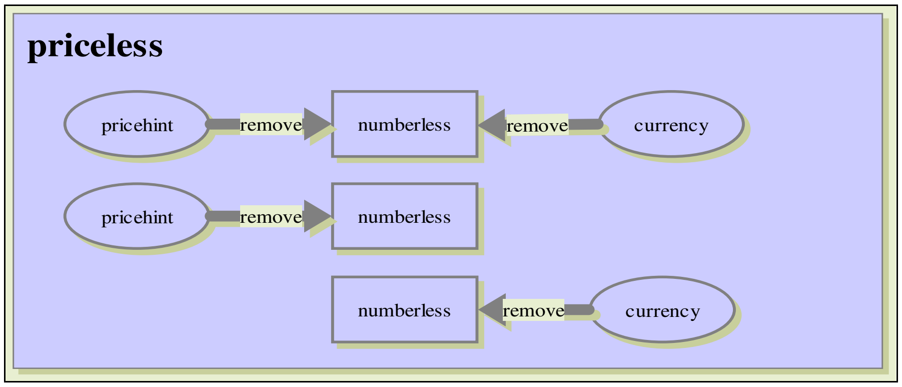


Figure 65: the priceless KC

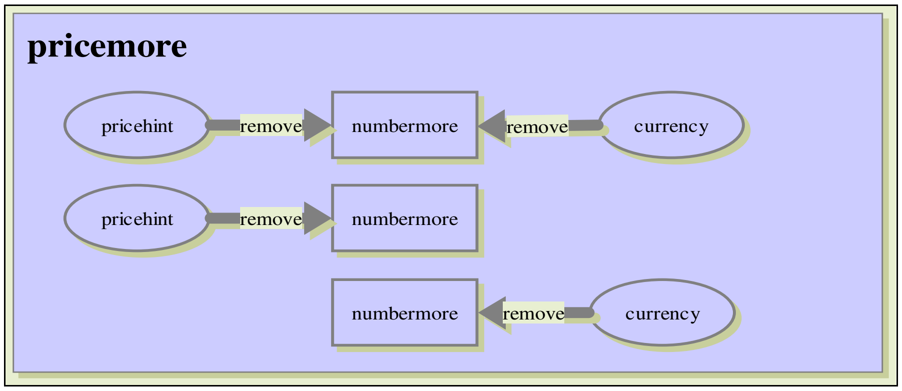


Figure 66: the pricemore KC

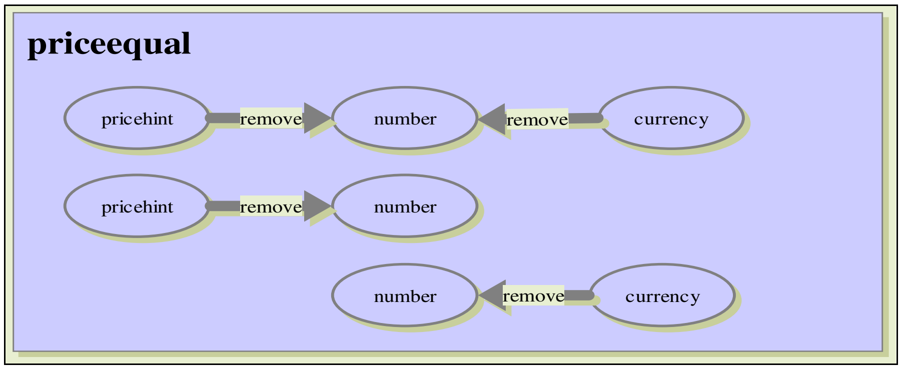


Figure 67: the priceequal KC

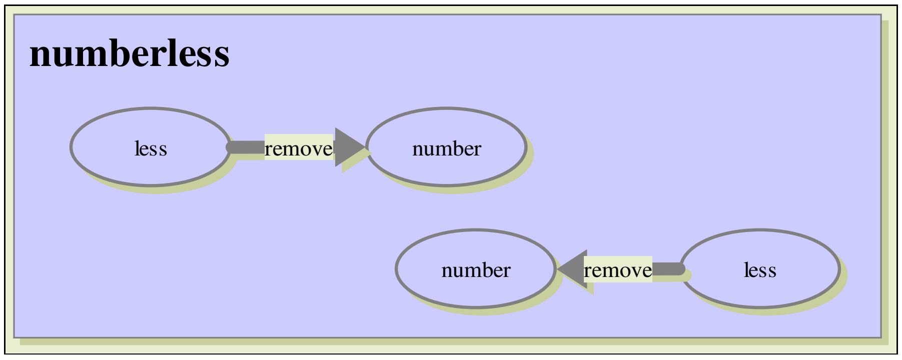


Figure 68: the numberless KC

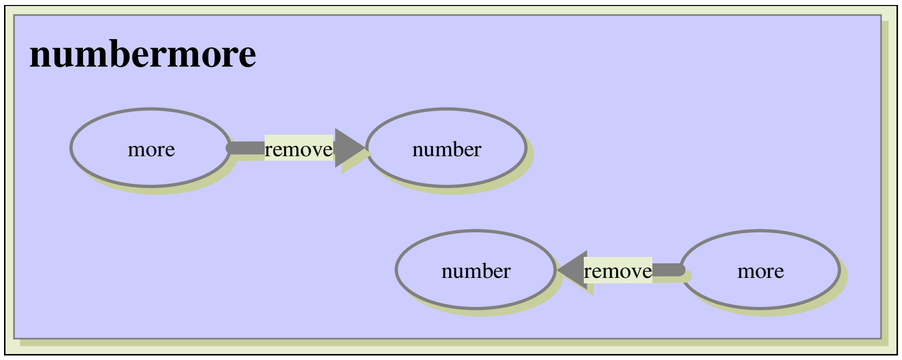


Figure 69: the numbermore KC

In the above diagrams used to define the price KC, we used the DED methodology to define the semantic grammar used to generate the corresponding rules. We defined the *numbermore* and *numberless* KCs to handle the cases when comparatives are directly adjacent to the number. In the middle level, we defined *priceless*, *pricemore*, and *priceequal* components that set the conditions to recognize a number as a price value. The *price* KC is a top level component that sets the boundary of this component.

By using this methodology we guarantee accuracy, maintainability, reusability and portability of the CE. Likewise, we define the knowledge components for each particular domain and all their constituents.

These KCs encapsulate domain specific knowledge along with natural language structures. They show our parsing strategy in focusing to extract domain dependent information from the free text, regardless of the surface structure they appear in.

```
;==============<<<<blank>>>>>==================
+{:::}{BLK:::}()P255;
;==============<<<<complete>>>>>=================
R{carpost,^complete:complete,&@entry::}{STAIL:::}()P5;
;==============<<<<SHIFT_RIGHT>>>>>============
R{:::}{:::}()P1;
;==============<<<<carpost>>>>>=================
<{vech:carpost,pri::}{price::pri:}P10;
============price===============
L{priceless,^price:price:}{^less,^more::}()P9;
L{pricemore,^price:price:}{^less,^more::}()P9;
-{less::}{priceequal:price,&@less::}()P9;
-{more::}{priceequal:price,&@more::}()P9;
+{priceequal:price,&@less::}{less::}()P9;
+{priceequal:price,&@more::}{more::}()P9;
L{piceequal,^price:price:}{^less,^more::}()P9;
============priceless================
-{pricehint::}{numberless:pricehint::}()P10;
+{numberless,pricehint:priceless,currency::}{currency::}()P10;
-{pricehint::}{numberless:priceless,pricehint::}()P10;
+{numberless:priceless,currency::}{currency::}()P10;
============pricemore================
-{pricehint::}{numbermore:pricehint::}()P11;
+{numbermore,pricehint:pricemore,currency::}{currency::}()P11;
-{pricehint::}{numbermore:pricemore,pricehint::}()P11;
+{numbermore:pricemore,currency::}{currency::}()P11;
===========priceequal================
-{pricehint::}{number:pricehint::}()P12;
+{number,pricehint:priceequal,currency::}{currency::}()P12;
-{pricehint::}{number:priceequal,pricehint::}()P12;
+{number:priceequal,currency::}{currency::}()P12;
==========numberless=================
-{less:::}{ANUM:&@less,numberless::}P13;
+{ANUM:&@less,numberless::}{less:::}P13;
==========numbermore=================
-{more::}{ANUM:&@more,numbermore::}P13;
+{ANUM:&@more,numbermore::}{more:::}P13;
```

This methodology defines the partial parsing for a certain domain by indicating domain relevant entities, relations, attributes, and different surface structures. Therefore, it is steering the parsing process, by mapping these DED to EnCo rules, which are processed easily, robustly and rapidly.

This systematic methodology of development of a CE system such as CATS proved to be not only feasible but efficient. As a matter of fact, we were able to build up a robust solution in a short time. This methodology is also very useful in expanding the system to accommodate more subdomains.

6.2.5 Extraction rules

To perform the CE task, we have written 710 rules for both the Cars and Real Estate domains. The rules were written based on our analysis of the sublanguage used for the classified ads. The study of those posts in the corpus enabled us to design the CRL-CATS as a higher abstraction of knowledge. In the same manner, the EnCo rules are the outcome of sublanguage analysis, in which we collected all structures and patterns used by users. The written rules for CE reflects our conceptual view of the classified ads post expressed by DED.

As we have shown in the previous section, a car post consists of components: *make, model, color, sale, want, year, price, feature, country and motor size* in addition to the MDO which is a vehicle.

As to the Real Estate post, it consists of the following components *sale, want, purpose, location, area, number of bedrooms, consist of, price, type, floor* and *feature* in addition to the MDO.

Some of these components are derived directly from the lexicon, such as *country* or *location* (DCs). Others are KCs that need to be defined as we did with *price* KC.

6.2.5.1 Numerical values recognition

A numerical value can have different interpretations: a price, motor size, year, area, number of bed rooms, floor number and for some cars it identifies a model. The rules use both semantic (domain) and syntactic structures to interpret correctly the meaning of numerical values. This has been demonstrated in the definition of the *price* component above. This mechanism depends on identifying the words immediately before and the words immediately after a number to identify its function correctly along with any attributes such comparatives. We also have seen that the positions of the surrounding words are extremely variable.

Additionally, numbers are written differently. Hence, it is necessary to convert (three thousands) or (3 thousand) into Arabic numerals (3000).

+{ANUM:&@ALF::}{ALF:::}P200;

The above rule demonstrates how we handle these variations. If the left window contains a number and the right window contains (thousand), add @ALF attribute to the left window and delete the right window.

It is impossible to correctly interpret the number if they are not hinted correctly. To demonstrate this, consider the following post.

للبيع مرسيدس 99	[S]
	sal(saloon:00, sale:00)
	mak(saloon:00,
	MERCEDES(country<germany,country<europe):06)
	mod(saloon:00, 99:0D)
	[/S]

For sale Mercedes 99	

In the above CRL-CATS, "99" was interpreted as model although it denoted the year of manufacturing. The cause of this error is the lack of any hint word helping the rules in finding the correct meaning. As a sequence, this rule will fire:

<{vech,add_nmod,^model_add,add_year:+model_add::}{ANUM::model:}()P70;

which means: if a make has models designated by a numbers (such as Mercedes and Peugeot) followed immediately by a number then consider this number as a model. The only solution to these cases is by validation of the "content" of the number. Unfortunately, this is not possible with EnCo[15].

Beside rules for correctly extracting numerical values, it is also important to attach to them the right attributes, such as @meter (for square meter) or @donem (1000 square meter), @permonth (per month), etc.

6.2.5.2 Named entities Extraction

The CATS CE uses both lexical lookup and internal structure (shallow parsing approach) to recognize named entities. Named entities include car makes, models, countries and locations. The segmentation is performed automatically by EnCo, based on the longest match algorithm.

Without using lexical lookup it is impossible to recognize many named entities found in the classified ads posts, knowing that Arabic does not use any orthographic means to distinguish named entities. Take the post: *"looking for Honda"*. It is only possible to recognize Honda as a car manufacturer by referring to the lexicon.

In order to handle unknown named entities (named entities that are not in the dictionary), we implemented an approach in which we look for an evidence or hint words (usually stored in the dictionary):

[15] "as EnCo has no numerical type, but only symbols, no numerical condition can be expressed in EnCo rules, hence any disambiguation procedure based on the knowledge of the probabilities of intervals of numerical attributes has to be postponed to the knowledge manipulation phase, which uses a relational database with several data types, including numbers".

{City, Forest, Center, River} + Word

CATS will assume the word is a location named entity, provided that the MDO is from the Real Estate domain.

6.2.5.3 Extraction of relations

Identifying relations between the MDO and the property values is also an essential part of CE engine. This is performed by identifying the MDO, linking it to the property values found in the text, and finally producing the CRL-CATS expressions.

<{vech:color_add::}{color::col:}()P70;

If the MDO is any type of vehicle and the right window contains a word representing *color value*, a *col* relation is built between *vech* and *color value*.

Similarly, the following rule will fire if the left window is a Real Estate MDO and the right window contains a node indicating "for sale". A *sal* relation is built connecting the MDO to the *sale* node.

<{flat:sale_add::}{sale::sal:}()P70;

Hence, the extraction of relationships between the MDO and its property values depends first on identifying the MDO. Subsequently, any extracted entity from the text is linked to the MDO, assuming that it is a property value for it. To demonstrate this, consider the following post:

سياره ميتسوبيش لانسر موديل 93 حره بور سنترلوك إنجكشن أللون فيراني ألفحص كرت أبيض ألسعر 5800 دينار	mod(saloon:00,
	Lancer(country<japan,make<MITSUBISHI):0F)
	yea(saloon:00,93:0R)
	fea(saloon:00,power steering:0Y)
	fea(saloon:00,center lock:12)
	col(saloon:00,gray:1U)
	col(saloon:00,white:2B)
	pri(saloon:00,5800:2M)

A Mitsubishi lancer car year 93 (free zone) power steering, centerlock, injection color dark gray check is white card price is 5800 dinar	

In the above car post, CATS recognizes the MDO which is "*saloon*". As can be seen in the CRL-CATS expression, all extracted information is linked to this MDO. The system could not extract "*mak*" relation because "*Mitsubishi*" was misspelt; nevertheless this was compensated by extracting "*Lancer*" as a *model*. Also, two "*fea*" relations exist in the expression reflecting the content of the post.

When we come to the "*col*" relation, we also find two: the first one is correct. However the second one is not. "*White card*" in this restricted domain terminology means "*perfect*", but since this term is not in the lexicon, *card* is ignored and *white* is recognized as a color and linked again to the MDO.

If we need deeper syntactic analysis, we would reject the second relation because there is no gender agreement between the color value and the MDO. However, this parsing strategy will not improve the results because many posts are ungrammatical and have a telegraphic style.

Some posts do not contain a MDO. For example, consider the following one:

للبيع مرسيدس
For sale Mercedes

To compensate for the missing information, the following rules are executed:

```
L{sale,^add_vech:+add_vech:::}{make:::}P20;
:"[سيارة]:+#INSE::"{add_vech:-add_vech::}P250;
```

The first rule explores the sentence and if it finds a word of type "*for sale*" right before a word of type "*make*" it will inform the second rule to insert a node [سيارة] or "*saloon*" before "*for sale*" node as shown below:

```
>>>>> lnode  INSERTED
[سيارة]{}  "saloon" (#INSE,vech,car) <A,3,3>;
>>>>> rnode
[للبيع]{}  "sale" (BLANK,sale,pur) <A,3,3>;
```

The insertion of the MDO is necessary to execute correctly and build the relevant relations between other constituents and the MDO.

6.2.5.4 Extraction of significant information only

All content extraction systems process relevant information and ignore unnecessary ones. The same approach is adopted in CATS. Unwanted information is simply discarded and removed during the processing. In the above example, we have seen that "*free*" is completely ignored because no match is found in the dictionary. However, if an unrecognized morpheme contains a morpheme which has a match in the dictionary, segmentation will be performed accordingly. This was shown in the "white card" example. Certainly it is more appropriate to ignore a whole morpheme than to recognize part of it, causing the system to produce wrong relations. This problem is manifested also in the Real Estate domain where we get frequently wrong location relations due to finding falsely a location name within an unrecognized morpheme.

We followed two strategies to minimize the erroneous effects of this problem:

- The first strategy to remove the unnecessary short morphemes from the dictionary to reduce the possibilities of finding a match within other longer morphemes.
- The second strategy is to populate the dictionary with frequently used extra expressions that we are not interested in to prevent EnCo to do any kind of segmentation.

6.3 Examples from various classifieds ads domains

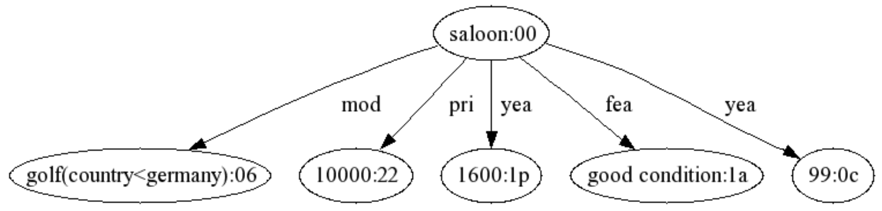
For further clarifications of the content extraction task, we will provide some examples in the Cars and Real Estate domains, for both types of posts. In the following examples, we will also present problems associated with working in these domains.

6.3.1 Example from the Cars domain

6.3.1.1 "Sell" posts examples:

Table 20: example 1 of a Car post

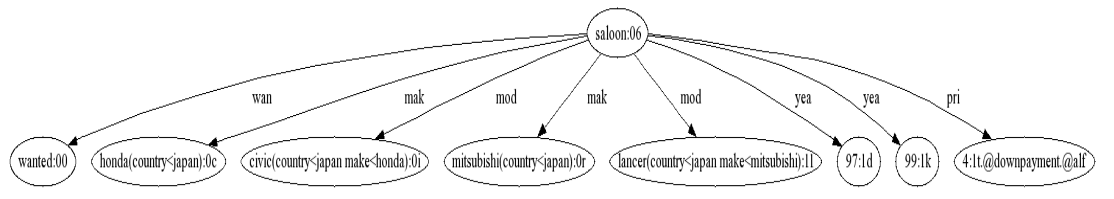
سياره جولف م99 فل أبشن مواصفات خليجيه ع الفحص بحاله جيده جدا 1600 CC. السعر 10000دينار غيرقابل للتفاوض.مهند الرمحي.	[S]
	mod(saloon:00,
	Golf(country<germany,country<europe,make<VOLKSWAGEN):06)
	yea(saloon:00,99:0C)
	fea(saloon:00,good condition:1A)
	yea(saloon:00,1600:1P)
	pri(saloon:00,10000:22)
	[/S]

A Golf year 99 full option Gulf features check in very good condition CC 1600 price is 10000 dinar not negotiable Muhannad Ramahi	
 saloon:00 mod pri yea fea yea golf(country<germany):06 10000:22 1600:1p good condition:1a 99:0c	
The above example shows the wrong interpretation of 1600, caused by the wrong position of unit word. However, this can be corrected by looking at the content or the value itself which is not possible with the EnCo. This example shows that some posts do not contain any explicit indication that it is of type "sell" or "looking for". In these cases, CATS considers them as "sell" posts, which is not always correct. Some posts implicitly indicate that they are of type "looking for", such as the following: "A car not more than 5000 Dinar". Compensation of missing information can then be done within the CE phase or later.	

6.3.1.2 *"Looking for" posts*

Table 21: example 2 of a Car post

مطلوب سياره هوندا سيفيك او ميتسوبيشي لانسر موديل 97. الى 99 بدفعه 4 آلاف دينار والباقي اقساط شهريه	[S]
	wan(saloon:06,wanted:00)
	mak(saloon:06,HONDA(country<japan):0C)
	mod(saloon:06,Civic(country<japan,make<HONDA):0I)
	mak(saloon:06,MITSUBISHI(country<japan):0R)
	mod(saloon:06,Lancer(country<japan,make<MITSUBISHI):11)
	yea(saloon:06,97:1D)
	yea(saloon:06,99:1K)
	pri(saloon:06,4:1T.@downpayment.@ALF)
	[/S]

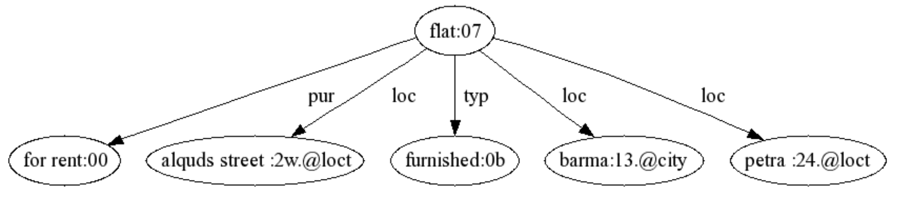
Looking for a Honda civic car or Mitsubishi lancer year 97 to 99 with a down payment 4 thousands dinar and the rest by monthly installments.	
 saloon:06 wan mak mod mak mod yea yea pri wanted:00 honda(country<japan):0c civic(country<japan make<honda):0i mitsubishi(country<japan):0r lancer(country<japan make<mitsubishi):1l 97:1d 99:1k 4:1t.@downpayment.@alf	
The above example shows a "looking for" post containing disjunctions and how these structures are mapped into CRL-CATS expressions. In the current CATS implementation, we do not distinguish between disjunctions or conjunctions, both produce the same CRL-CATS expressions. However, it would be possible to distinguish between them simply by putting the attribute ***@or*** for disjunction and ***@and*** for conjunction.	

6.3.2 Example from the Real Estate domain

6.3.2.1 *"Sell" post examples*

Table 22: example 1 of a Real Estate post

للإيجارشقق مفروشة كاشفة و مميزة+موقع سوبرماركت لايوجد غيره أمام بوابة جامعة البترا القمة تقاطع ط مطار وشارع القدس..	[S]
	pur(flat:07,for rent:00)
	typ(flat:07,furnished:0B)
	loc(flat:07,13:برما.@city)
	loc(flat:07,24: البترا.@loct)
	loc(flat:07,2: شارع القدسW.@loct)
	[/S]

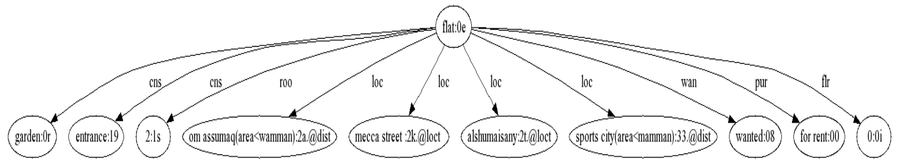
For rent furnished apartments high and distinguished + a location for supermarket facing alone the main gate of Al–Petra university the crossing of airport road and alquds street	
 flat:07 pur loc typ loc loc for rent:00 alquds street :2w.@loct furnished:0b barma:13.@city petra :24.@loct	

This above post shows how condensed and unorganized some posts can be. The poster is offering two things, furnished flats and a supermarket, which CE cannot process. One location is recognized correctly (*Alquds Street*). *Al-Petra University* is not recognized because is misspelled (hamza is omitted) and it is identified as *Petra city*. Additionally, *airport road* is not also recognized because of the abbreviation. Hence, CE added a new location in the CRL-CATS expression [“برما” barma] which exists in the lexicon (it is a location name in Jordan) because it is part of [“سوبرماركت” supermarket]. And, [“سوبرماركت” supermarket] is not in the lexicon, EnCo performed the segmentation based on the presence of “barma” as longest match present in the lexicon, and hence [“برما” barma] appeared as a location name in the output expression.

6.3.2.2 “Looking for” example

Table 23: example 2 of a Real Estate post

للايجار مطلوب شقة أرضية مع حديقة واسعةوكراج ومدخل مستقلين جديدة 2_3نوم في الرابية أم السماق شارع مكة الشميساني المدينة الرياضيةلغاية3500سنوي	[S]
	wan(flat:0E,wanted:08)
	pur(flat:0E,for rent:00)
	flr(flat:0E,0:0I)
	cns(flat:0E,garden:0R)
	cns(flat:0E,entrance:19)
	roo(flat:0E,2:1S)
	loc(flat:0E,الرابية (area<wAmman):2A.@dist)
	loc(flat:0E,أم لسماق(area<wAmman):2A.@dist)
	loc(flat:0E,2: شارع مكةK.@loct)
	loc(flat:0E,2:الشميسانيT.@loct)
	loc(flat:0E,المدينةالرياضية(area<mAmman):33.@dist)
	[/S]

Looking for to rent a ground apartment with a large garden and a garage with private entrance new 2-3 bedroams in Rabiah, Om Alsumaq Mecca street Alshumesani Sports city up to 3500 annual	
 flat:0e cns cns roo loc loc loc loc wan pur flr garden:0r entrance:19 2:1s om assumaq(area<wamman):2a.@dist mecca street :2k.@loct alshumaisany:2t.@loct sports city(area<mamman):33.@dist wanted:08 for rent:00 0:0i	
The above “looking for” post illustrates some orthographic problems of Arabic: in “ **حديقة واسعةوكراج and المدينة الرياضيةلغاية”,** spaces have been wrongly omitted. However, the system is able to extract correctly, because of the longest match algorithm used by EnCo. The CE failed to identify the price, due to absence of hint or unit words. One final note, regarding the locations named used in CRL-CATS. We used Arabic to express locations (CWs) in violation of the definition. This was due to the large number of locations name which were very difficult to transliterate into English. However, we kept important locations in Latin characters.	

6.4 Techniques for higher recall and precision

6.4.1 Handling of unknown words

It is not possible to anticipate every user input. Unknown words are handled during the CE phase where the system tries to identify them from their positions or any surrounding hint word. For example if a preposition precedes unknown word in the Real Estate domain, it is assumed a location name.

Also in some cases, unknown words can be deduced from the semantic information in the dictionary, for example, if CE could not recognize a car make but recognized a car model, the car make is deduced automatically.

However, unknown words can still cause other problems if they are wrongly segmented as we have shown before.

6.4.2 Handling of compound words

Multi-words terms in the classified ads sublanguage are frequent and must be treated as integrated concepts. This is very important in enhancing the quality and the precision of the system [Doan Nguyen and Kosseim 2004].

In our case, multi-words such as the following are processed and handled as one word lexical items.

"manual transmission" <جير عادي>

"except the transmission" <ما عدا الجير>

"separate house" <منزل مستقل>

"not more" <لا يزيد عن>

Table 24: multi-words entries in the dictionary

Multi-word	CW
فكس واجن	"VOLKSWAGEN(country<germany,country<europe)"
فوكسواجن	"VOLKSWAGEN(country<germany,country<europe)"
فكس واجن	"VOLKSWAGEN(country<germany,country<europe)"
فوكس فاجن	"VOLKSWAGEN(country<germany,country<europe)"
سانج يونج	"SangYong(country<korea)"
فولكسفاغن	"VOLKSWAGEN(country<germany,country<europe)"

Additionally, many location names consist of two or even more words and are treated as one unit.

6.4.3 Handling of spelling errors and variations

CATS does not implement any spell checking. In order to cope with spelling errors and orthographic variations, we have adopted full forms listing. We tried to generate automatically all possible orthographic forms of lexical items and populate the dictionary with them. Around 20,000 (65%) entries have been generated by this way. Similarly, different forms of transliterated named entities are manually generated and stored in the dictionary.

Table 25: spelling variations

Orthographical forms	CW
تدفئة مركزى	"heating"
تدفئة مركزي	"heating"
تدفئةمستقلة	"heating"
تدفئةمستقله	"heating"
تدفئه مركزى	"heating"
تدفئه مركزي	"heating"
تدفئة	"heating"
تدفئه	"heating"
تدفئةمركزى	"heating"
تدفئةمركزي	"heating"

Although this approach is not the best, it enabled the system to increase precision and recall. On the other hand, since these forms are generated in advance, we can claim that this approach is faster than using spell checking techniques.

Table 26: transliterated forms

word	CW
كارينز	"Carens(country<korea,make<KIA)"
كارنز	"Carens(country<korea,make<KIA)"
كرنز	"Carens(country<korea,make<KIA)"
كرينز	"Carens(country<korea,make<KIA)"
اوبيروس	"Operus(country<korea,make<KIA)"
بيروس	"Operus(country<korea,make<KIA)"
بروس	"Operus(country<korea,make<KIA)"
بيرس	"Operus(country<korea,make<KIA)"
CRV	"CRV(country<japan,make<HONDA)"
CR-V	"CRV(country<japan,make<HONDA)"
سي ار في	"CRV(country<japan,make<HONDA)"
سي آر في	"CRV(country<japan,make<HONDA)"
تريل	"X-Trail(country<japan,make<NISSAN)"
trail	"X-Trail(country<japan,make<NISSAN)"
مورانو	"Morano(country<japan,make<NISSAN)"
مرانو	"Morano(country<japan,make<NISSAN)"
برادو	"prado(country<japan,make<TOYOTA)"

Conclusion of Part B

We have shown how we used the EnCo, designed to write enconverter, to implement a content extractor. We have also introduces the DED methodology for writing EnCo rules. We have tested successfully this method for both CE and general enconversion.

As to CE, we have demonstrated the structure of our system. We have also shown the details of the using the EnCo. Additionally, we have described the structures of the rules and the dictionary. Additionally, we have used DED in specifying the semantic grammar. Furthermore, we have demonstrated the system by giving some real examples in both domains. Finally, we have described techniques used to enhance the quality.

Part C. The QA component and end-to-end evaluation

Introduction to Part C

CE handled mismatches in the local level or within the post “sell” or “looking for” only. On the other hand, CATS should also formulate responses (from previously processed and stored “sell” posts) to users’ “looking for” posts. In a sense, variations between the two types are handled by using the semantic matching. This will trigger another question: what type of storage is needed? Is it necessary to use storage with very general inference capabilities? Or we can perform the task with light-weight inference storage but has other features such as reliability and concurrency?

To evaluate the success of this experiment, we have to check the current status of the CATS. We also should use some metrics and compare it with other systems. It is also important to evaluate the expandability to other domains and other languages.

This part is divided into two chapters. In chapter 7, we describe the process of response generation. In chapter 8, we evaluate the system and describe its short and middle term prospectives.

Chapter 7. The QA component: database design, semantic matching and response generations

Introduction

During the past two decades, relational databases have been developed to a level that is cannot be emulated by other storage means, semantic or non-semantic. This is because they accumulated essential and critical features such as scalability, reliability and concurrency, needed in building robust applications in various sectors.

On the other hand, relational databases alone are not adequate to build ontologies or perform semantic-based tasks.

In this chapter, we will show how we use a relational database to generate responses based on semantic matching, and show the techniques used to overcome this inadequacy and utilize well-established technologies to build a robust system.

We will first discuss the database design, and present the basic implementation. Additionally we will then compare between two scenarios to implement reasoning and semantic matching. In the third section, we will present how we process "sell" and "looking for" posts. Finally, we will explain the process of generating responses.

7.1 Database design

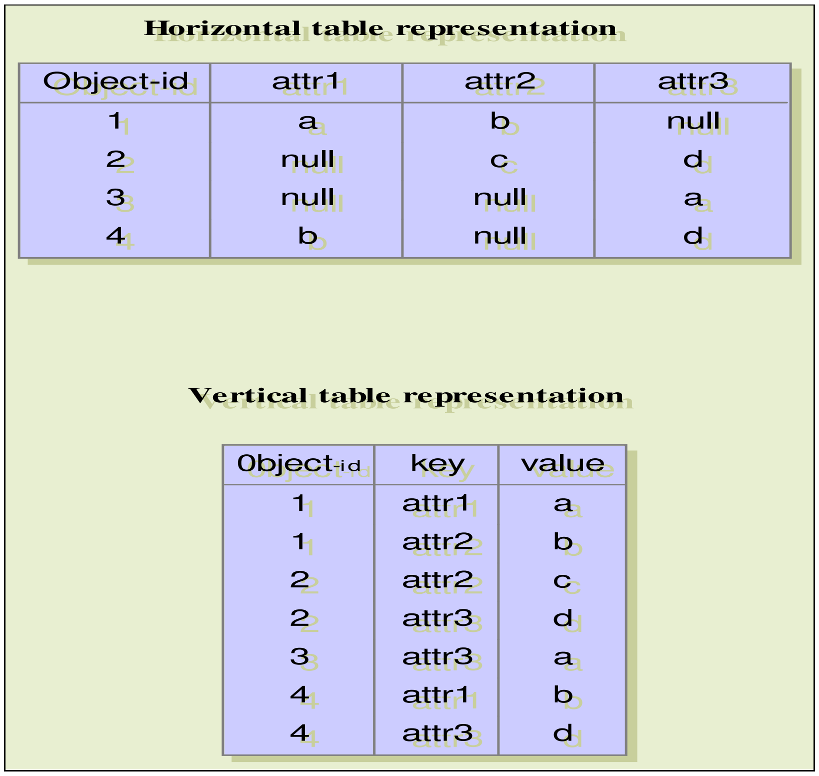
In this section, we will describe the basic design of the DB used. We will discuss the method of implementing reasoning and semantic based matching used by CATS.

7.1.1 Basic implementation

In relational database systems, data objects are normally stored using a horizontal scheme [Agrawal, et al. 2001]. A data object is represented as a row of a table. There are as many columns in the table as the number of attributes the objects have (figure 70). The horizontal representation has some disadvantages:

- The existence of many null values increases the overhead on the database.
- To accommodate new attributes, we would need repeated changing of the table.

To address the above problems, the vertical table representation has been introduced. Figure 70 shows that the vertical representation table stores tuples for those attributes that have values. Adding new attributes does not require performing any schema alteration. Simply, add a new row corresponding to the new attribute. However, these apparent advantages have a tradeoff; writing SQL for this scheme is awkward and error-prone.



Horizontal table representation

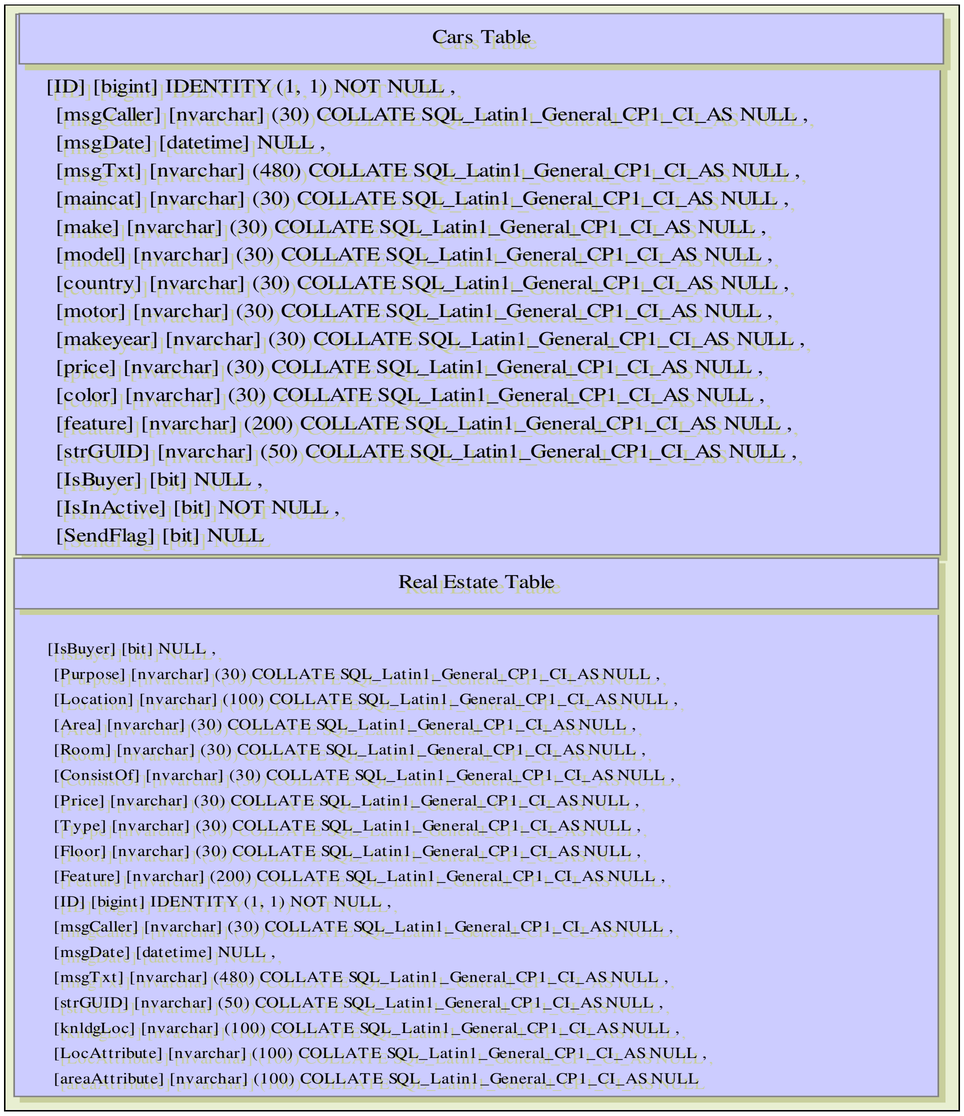
Object-id	attr1	attr2	attr3
1	a	b	null
2	null	c	d
3	null	null	a
4	b	null	d

Vertical table representation

Object-id	key	value
1	attr1	a
1	attr2	b
2	attr2	c
2	attr3	d
3	attr3	a
4	attr1	b
4	attr3	d

Figure 70: horizontal and vertical table representations

In our case, we have basically two types of objects corresponding to Cars and Real Estates. The number of attributes for each domain is limited, and it is not expected to modify the attributes. Hence, we used the horizontal representation and minimized the number of null values by using two tables, one for each domain, to store the data as shown in figure 71



Cars Table

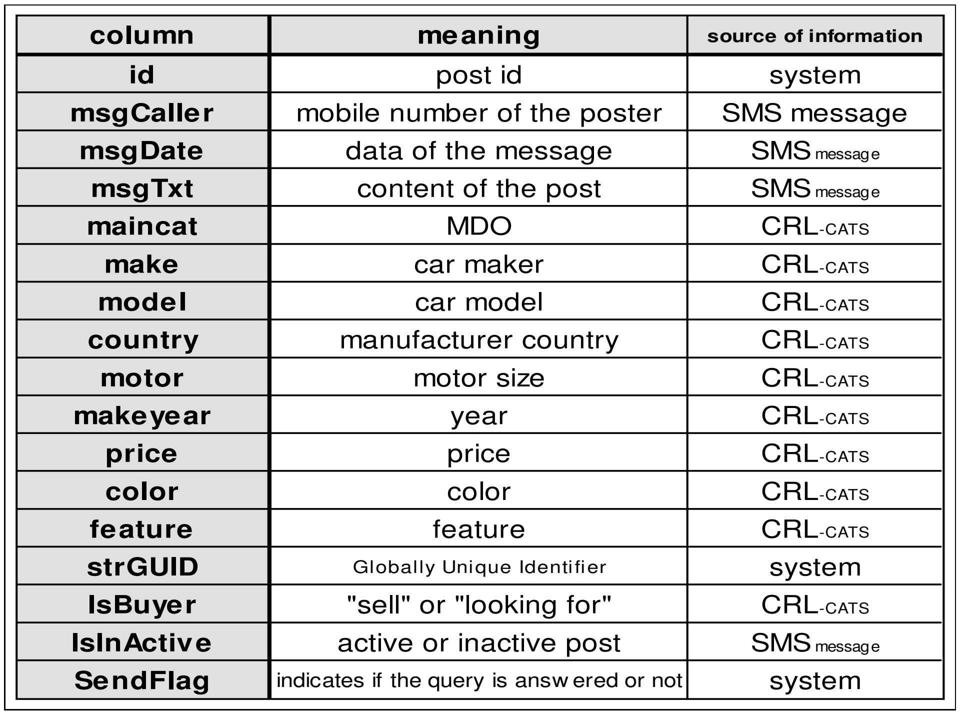
```
[ID] [bigint] IDENTITY (1, 1) NOT NULL ,
 [msgCaller] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [msgDate] [datetime] NULL ,
 [msgTxt] [nvarchar] (480) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [maincat] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [make] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [model] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [country] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [motor] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [makeyear] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [price] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [color] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [feature] [nvarchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [strGUID] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [IsBuyer] [bit] NULL ,
 [IsInActive] [bit] NOT NULL ,
 [SendFlag] [bit] NULL
```

Real Estate Table

```
[IsBuyer] [bit] NULL ,
 [Purpose] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [Location] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [Area] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [Room] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [ConsistOf] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [Price] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [Type] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [Floor] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [Feature] [nvarchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [ID] [bigint] IDENTITY (1, 1) NOT NULL ,
 [msgCaller] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [msgDate] [datetime] NULL ,
 [msgTxt] [nvarchar] (480) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [strGUID] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [knldgLoc] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [LocAttribute] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [areaAttribute] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
```

Figure 71: Cars and Real Estate tables

As shown in figure 72, in addition to columns that correspond to object's attributes, we have other types of columns that either correspond to certain values or serve a certain function. Hence, CRL-CATS expressions are the source of the columns. Some columns are filled with different sources as shown.



column	meaning	source of information
id	post id	system
msgCaller	mobile number of the poster	SMS message
msgDate	data of the message	SMS message
msgTxt	content of the post	SMS message
maincat	MDO	CRL-CATS
make	car maker	CRL-CATS
model	car model	CRL-CATS
country	manufacturer country	CRL-CATS
motor	motor size	CRL-CATS
makeyear	year	CRL-CATS
price	price	CRL-CATS
color	color	CRL-CATS
feature	feature	CRL-CATS
strGUID	Globally Unique Identifier	system
IsBuyer	"sell" or "looking for"	CRL-CATS
IsInActive	active or inactive post	SMS message
SendFlag	indicates if the query is answered or not	system

Figure 72: columns of the Cars table

7.1.2 Reasoning and semantic-based matching

We have shown that this experimental system uses semantic processing in performing CE. We also have shown that CRL-CATS is a representation that captures the content and the meaning of a post. During the process of CE and the production of CRL-CATS expressions, CATS handles different causes of mismatch at the local level (within the posts) such as the following:

- **Equivalent alternatives (syntactic variations).** We have seen that posts can have different syntactic structures caused by different word orders and grouping patterns of its constituents. The variations extend to the constituents level as we explained in the "price" KC. In chapter 2, we have seen how the positions of some constituents vary from one post to another: "for sale Honda ….." "A Honda car ……. For sale". Also from Real Estate domain, consider the following posts, which all have the same meaning but in different structure. "…..located between villas), "…..surrounded by villas" "……class A residential area".

- **Inconsistent vocabulary.** Although the vocabulary used is narrow and limited, posters use different words to express the same concept. For example, to express the concept "more", users use around 30 words (including spelling variations).

- **Omission.** In some posts, we find that some constituents are not present because it does not interest the poster or it is irrelevant for him such as "looking for a car above 2001". In this post, the user omits all other criteria that can restrict his query and mentions only one. Other causes of omissions arise when information is deemed to be implicitly known such as "looking for a Clio" in which "car" is omitted or "for sale 500 square meter" in which "land" is omitted. In some posts, we don't find any indication of the type ("sell" or "looking for"): "a Toyota Corolla above 99 and with less than 7000 dinar" because the poster thinks it can be known from the context of the post.

On the other hand, other sources of mismatch are handled at the global level, that is within the database such as:

- **Granularity.** Some posters encode the knowledge but at different levels of detail. For example: "looking for a CIVIC" or "A Japanese Honda Civic car for sale". Both posts hold the same knowledge at different details.
- **The use of generalization in the query**. Another possible source of mismatch is related to the use a generalization concept for searching such "looking for a French car", "looking for a villa in West Amman" or "looking for economical car". Usually these words ("French", "West Amman" and "economical") do not appear in the "sell" post since they are implicitly known.
- **No answer situations**. And the generation of approximate answers.

To address the above problems, exact matching is not sufficient. Hence, we use semantic matching [Das, et al. 2004]. As a sequence, the DB has to be designed to identify related concepts and to have an inference mechanisms for deduction of information not explicitly asserted.

For example, when a "sell" post is received saying "for sale LANCER 1999", the system recognizes that it is a car, it is a Japanese car, and that the maker is Mitsubishi. Therefore, the system is capable of detecting and compensating for missing information in both types of messages. As a result, the above record would be one of the answers of the following post: "looking for a Japanese car" (figure 73).

To provide reasoning functionalities, we use feature-based categorization of concepts. In which concepts are assigned to categories according to commonalties in specific features. For example, Renault, Peugeot and Citroen share the same features that they are all French cars. In a similar manner, Clio and Megan are car models that share the same manufacturer. Similarly, in the Real Estate domain, locations names are grouped together according to the larger area they belong to.

Without this knowledge structure, it is impossible to handle situations that require relaxing constraints in the original "looking for" post caused by a "no match". The relaxation algorithm needs to know the neighboring information. Also, in situation where the user is not specific or is using general constraints.

The advantages of using this approach results in automated system that combines the benefits and flexibility of free-text as an interface with the power of semantic-based information retrieval e-commerce systems. Hence, we attain the following goals:

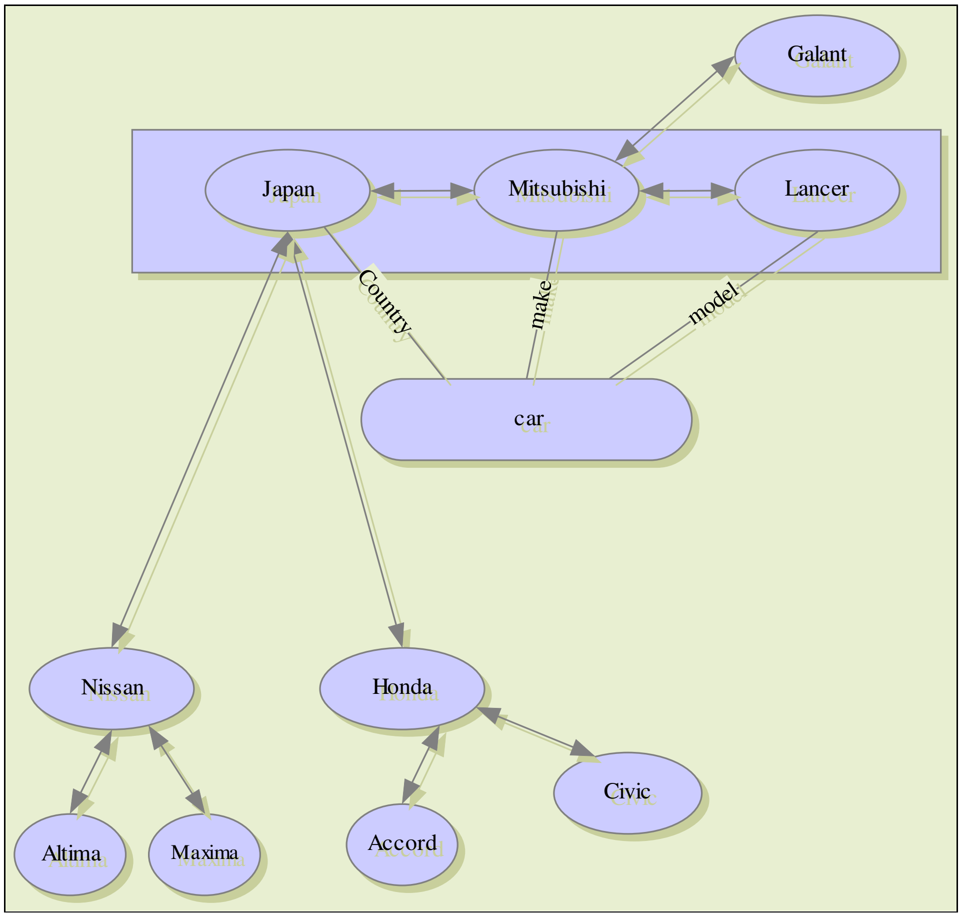


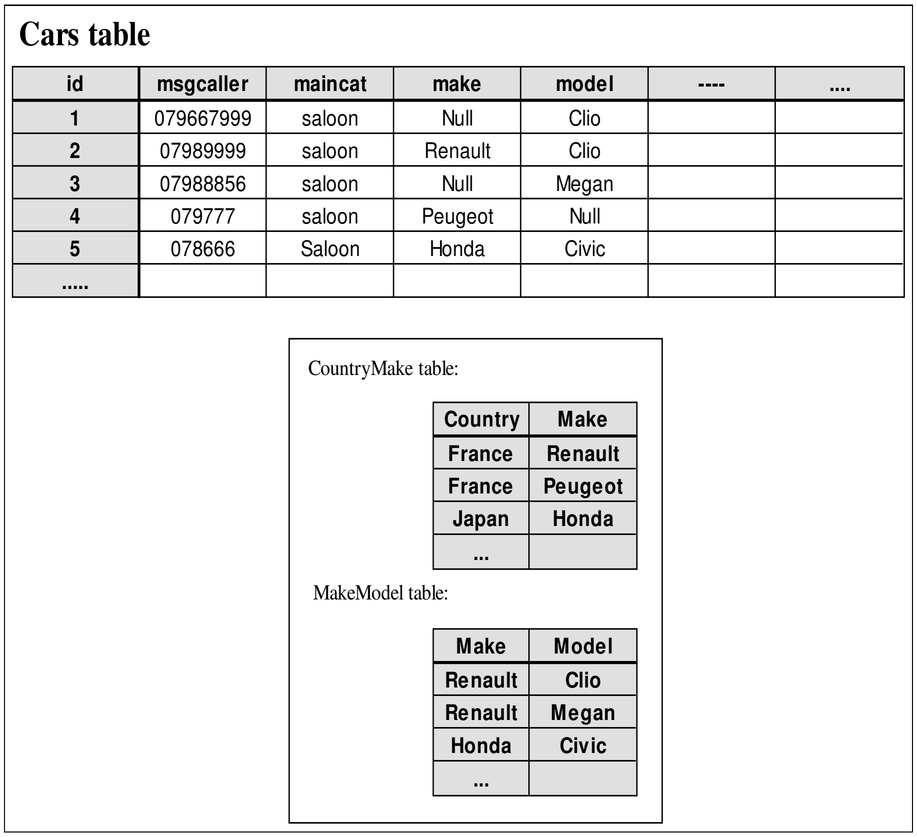
Figure 73: an instance of Cars semantic taxonomy

- **Response consistency**: Answers do not depend on the choice of words for describing a request or for submitting a question. "Looking for" posts with the same meanings lead to same answers.
- **Accuracy and powerful similarity-based retrieval.** Relations between concepts in the model are essential to uncovering similarities between "looking for" and "sell" posts.

Different scenarios can be used to realize this light-weight inference engine [Motik, et al. 2002] using a RDBMS. In the following sections we will discuss two possible scenarios and we will explain in more detail the selected one.

7.1.2.1 Scenario 1

In this design schema, the DB has its own "Ontology" implemented by using table(s) to store the concept hierarchy and taxonomy as shown in figure 74.



Cars table

id	msgcaller	maincat	make	model	----	....
1	079667999	saloon	Null	Clio		
2	07989999	saloon	Renault	Clio		
3	07988856	saloon	Null	Megan		
4	079777	saloon	Peugeot	Null		
5	078666	Saloon	Honda	Civic		
.....						

CountryMake table:

Country	Make
France	Renault
France	Peugeot
Japan	Honda
...	

MakeModel table:

Make	Model
Renault	Clio
Renault	Megan
Honda	Civic
...	

Figure 74: scenario 1 implementation

In this design, to answer a "looking for" post such as "looking for a French car", requires the consultation of the ontology to get all makes and models of the French makers:

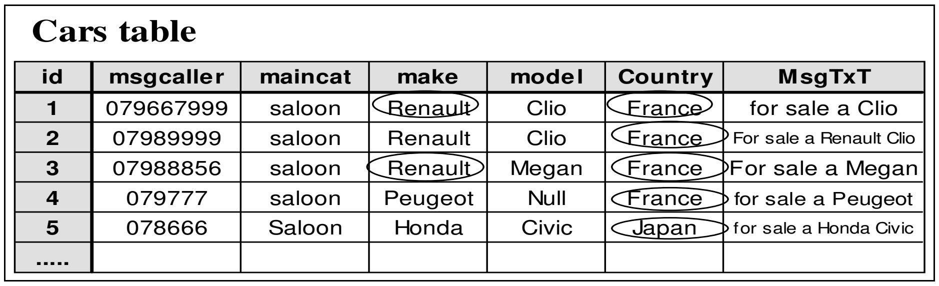
```
SELECT msgcaller
FROM cars
WHERE make in
(select make from  countrymake  where country  ="France")
Or
model in (select model from makemodel where make in (select make from  countrymake  where
country  ="France") );
```

The above example shows the level of complexity of generating a SQL query for this design schema. Usually, queries that consult this ontology are cumbersome and error-prone to generate. This is because they have to address different issues such as: classes, subclasses, transitive relations, etc. Additionally, the execution of such queries is computationally expensive and has a low performance due to their nested nature.

On the other hand, this schema provides a good environment for expanding the system to perform more advanced semantic matching (which includes heuristic terms) such as answering "looking for a family car", or "looking for a small apartment" without affecting other tables or modules.

7.1.2.2 Scenario 2

In this design schema, we don't allocate any table for the ontology, but we use the semantic labels embedded within the CWs to fill concerned columns values and insuring that there are no null values in them.



Cars table

id	msgcaller	maincat	make	model	Country	MsgTxT
1	079667999	saloon	Renault	Clio	France	for sale a Clio
2	07989999	saloon	Renault	Clio	France	For sale a Renault Clio
3	07988856	saloon	Renault	Megan	France	For sale a Megan
4	079777	saloon	Peugeot	Null	France	for sale a Peugeot
5	078666	Saloon	Honda	Civic	Japan	for sale a Honda Civic
.....						

Figure 75: scenario 2 implementation

As shown in figure 75, the system inserts "make" and "country" values regardless of their presence in the original "sell" post. In CATS, we used this design, because it performs the semantic matching with simpler queries and consequently at a higher performance. More details in different aspects of this design will be presented in the next sections.

7.2 Storing "sell" posts

As we have shown in chapter 2, the QM receives the CRL-CATS expressions and extracts from them the needed information such as the name of the property, the value of the property and the MDO. Additionally, it extracts semantic information embedded within the CWs. It then identifies the domain (Cars or Real Estates) by looking at the value of the MDO; if it is ("saloon", "bus", "van", "pickup"), it belongs to the Cars. On the other hand, if it is ("flat", "roof", "cabana", "villa", "land", "shop", "office", "orangefarm"), it belongs to Real Estate. Finally, QM identifies the type of the post by checking the existence of a "wan" relation. If yes, it is a "looking for", otherwise it is a "sell" post.

For a "sell" post, the extracted information from the CRL-CATS and from other sources (figure 72) is passed to a stored procedure to generate the insert SQL statement. This is because stored procedures provide a separation between the layers and save network traffic (sending parameters is more efficient than sending a long SQL statement). Fore example, the parameters for a "sell" post in the Cars domain are passed to the following stored procedure:

```
CREATE  PROCEDURE ST_AddCarMsg (
@ID as bigint,
@msgCaller as nvarchar(30),
@msgDate as datetime,
@msgTxt  as nvarchar(480),
@maincat  as nvarchar(30),
@make  as nvarchar(30),
@model  as nvarchar(30),
@country  as nvarchar(30),
@motor  as nvarchar(30),
@makeyear  as nvarchar(30),
@price  as nvarchar(30),
@color  as nvarchar(30),
@feature  as nvarchar(200),
@strGUID  as nvarchar(50),
@IsBuyer as bit
)
 AS
SET NOCOUNT ON

INSERT INTO [KnowledgeBase].[dbo].[cars](
[msgCaller], [msgDate], [msgTxt], [maincat], [make], [model], [country], [motor],
[makeyear], [price], [color], [feature], [strGUID], [IsBuyer])
VALUES(
--@ID, @msgCaller, @msgDate, @msgTxt, @maincat, @make, @model, @country, @motor,
@makeyear, @price, @color, @feature, @strGUID, @IsBuyer
)
return 0
```

To demonstrate the process of transformation, consider the following "sell" post *"for sale a Lancer 99 at 5000 dinar"*

The CRL-CATS for the above post is:

```
[S]
sal(saloon:00, sale:00)
mod(saloon:00, Lancer(country<japan,make<MITSUBISHI):06)
yea(saloon:00, 99:0I)
pri(saloon:00, 5000:0L)
[/S]
```

Since it is a "sell" post, the system issues an insert SQL statement (as we have shown, this is performed in reality by using a stored procedure and involves more parameters) to populate the database with this post:

```
Insert into cars (maincat, model, year, price, country, make)
Values('saloon','lancer','99','5000','japan','mitsubishi')
```

Each property value in CRL-CATS fills the corresponding column in the Cars table in the database. Note that, the semantic information (country and make) is extracted and mapped into pre-specified columns to enable the process of semantic matching.

7.3 Processing of "looking for" posts

Once a CRL-CATS expression is processed and identified as a "looking for" post, it is stored in the Cars or Real Estate tables along with "sell" posts. It is marked as a "looking for" post by setting the "IsBuyer" value. Then, the extracted values are used to generate a SQL query dynamically, again using a stored procedure (see appendix G).

For example, the following CRL-CATS which corresponds to the query "*looking for a Mitsubishi Lancer*":

```
[S]
wan(saloon:00, wanted:00)
mak(saloon:00, MITSUBISHI(country<japan):06)
mod(saloon:00, Lancer(country<japan,make<MITSUBISHI):0G)
[/S]
```

is converted to the following SQL query:

```
select  MsgCaller  from  Cars  where
 make ='mitsubishi'
and  model ='lancer'
and  maincat ='saloon'
```

Hence, the method of extracting semantic relations and storing them in the corresponding columns, regardless of their existence in the original "sell" post, makes possible the generation of that kind of simple and efficient queries.

7.3.1 No answer situations

We are first try to answer a user's query as it is asked. If it has no answers, we relax it to a more general one, and try again [Gaasterland, et al. 1992]. For example, If no answer is found for the above query "*looking for a Mitsubishi Lancer*", the following SQL query will be issued:

```
select MsgCaller from Cars  where
(
make ='mitsubishi' or
model ='lancer'
)
and  maincat ='saloon'
```

As for processing "sell" posts, a stored procedure is used within the DB to dynamically generate those queries. At the beginning, it will generate a query based on conjunctive conditions. If no answer retrieved, it will issue another query but this time with "or" operator connecting these conditions.

In case that no answer found even with this relaxation, the query is marked as an unanswered by setting the SendFlag in the main table. A service (an agent) will periodically check at predefined time intervals the availability of any is found answer, it will be sent to the poster.

7.3.2 Generating approximate answers

From the end user perspective, this behavior might not be the best. Some people believe that no answer is better than "approximate" answer. Another issue is how to select an approximate answer. In this implementation, we choose to move up the hierarchy.

However, in the car domain, we can have different definition of proximity such as motor size, or the price, which needs to be defined in advance. The same behavior applies to the Real Estate domain, in which we categorize geographic locations into a hierarchy. As an example: Amman is divided into 4 major areas (north, south, west and east), each location in Amman is attached to one of these areas. Therefore, the system can answer questions such as "*looking for a land in West Amman*". It also can handle cases of no answer in the following question "*looking for a land in Khalda*".

7.4 Generating responses

Given the length constraint put by SMS, we used a tabular form to display the results as shown in figure 76.

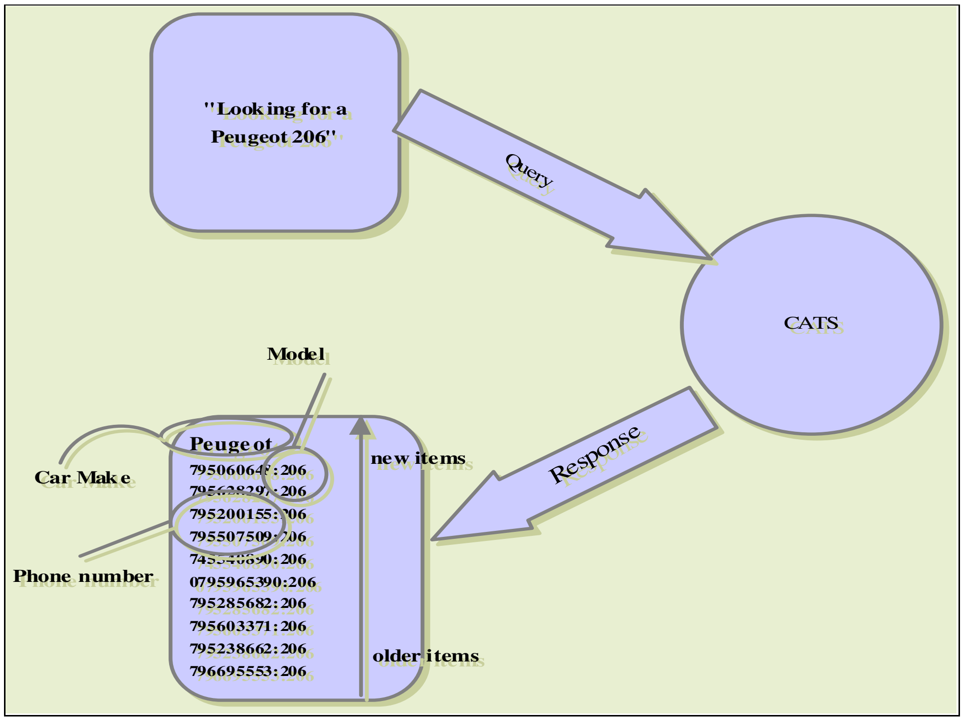


Figure 76: example of a response in Cars

Adding more information to the response, such as year, and price would reduce the number of displayed items. Also, many of the "sell" post lack information about year or price, which would cause irregularity in the response format.

The items within a response are ordered according to the sell post's time: the most recent one appears at the top of the list.

In some situations, it is not possible to send all the results because they do not fit in one message. To enable a user to view unsent items, simply he can post "more", and the system replies him with the remainder of results.

The RH (response handler) checks the size of the results against the specified limit, and sends the items within that limits. The rest are stored in the database with the phone number of the query poster. If he sends "more", RH recognizes him and the remaining results are sent to him as shown in figure 77.

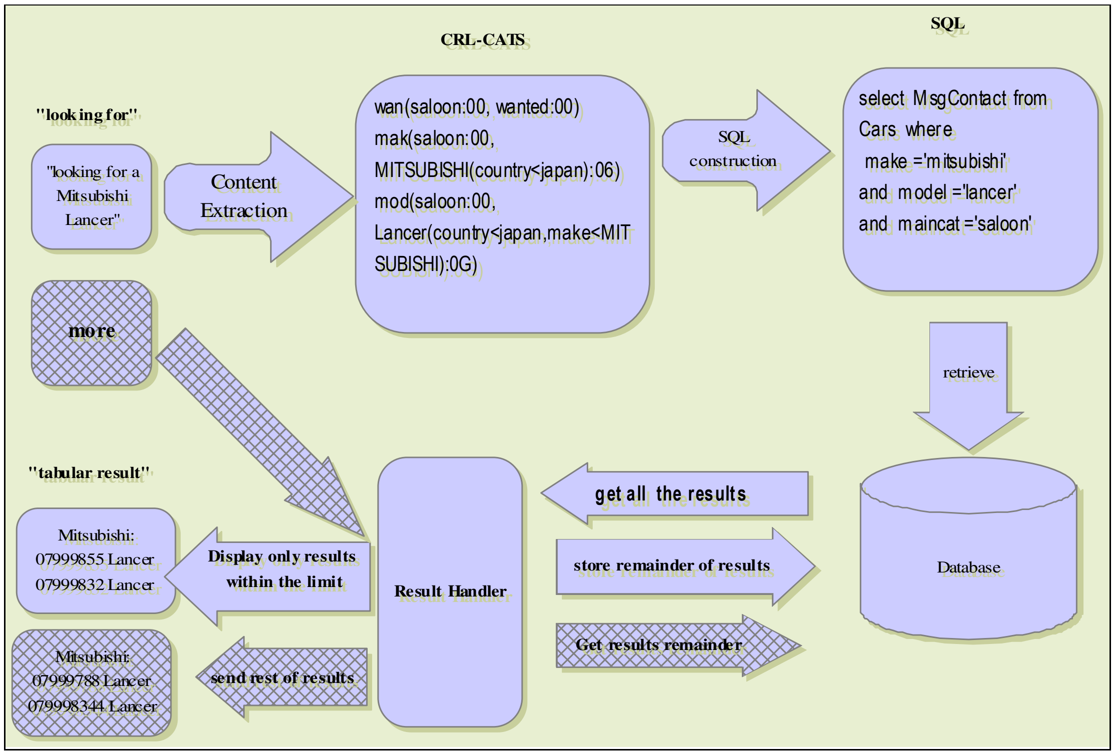

CRL-CATS
SQL
"looking for"
"looking for a Mitsubishi Lancer"
Content Extraction
wan(saloon:00, wanted:00) mak(saloon:00, MITSUBISHI(country<japan):06) mod(saloon:00, Lancer(country<japan,make<MITSUBISHI):0G)
SQL construction
select MsgContact from Cars where make ='mitsubishi' and model ='lancer' and maincat ='saloon'
more
retrieve
"tabular result"
Mitsubishi: 07999855 Lancer 07999832 Lancer
Display only results within the limit
Result Handler
get all the results
store remainder of results
Get results remainder
Database
Mitsubishi: 07999788 Lancer 079998344 Lancer
send rest of results

Figure 77: information flow of "more" command

A poster of a "sell" ad can prevent it from being listed within generated responses by sending "deactivate" command. He can make it active again by sending "activate".

7.4.1 Generating "natural" responses

We know from our experience that some people like to have the complete original "sell" ad, instead of the current format. As we have seen, it is possible technically, the only problem is the message size constraints.

One possible solution to this problem is to enable the user, after receiving the initial tabular list, to request full text of certain item. For example, he can send "list 1" to ask for the original text of the top item.

Another possibility is not to send the original message but one generated from the CRL-CATS form. In that case, the CRL-CATS expressions for each "sell" post are also stored in the table. When a user asks for full text, the CRL-CATS for a sell post is deconverted into Arabic (or possibly in other NL in the future) and sent to the user. The advantage of this approach is that the generated text is more concise and is to the point, with the possibility of producing results in different languages. However, this solution requires a system for the deconversion process which needs development.

7.4.2 Human intervention

In some cases human intervention is necessary to understand users' posts. An operator monitors the posts online and acts accordingly. The operator can correct the user input and issue it again, or, if information is missing, the operator calls the poster and asks him for more information.

The question is whether this can also be 100% automatic? Can we add the ability to CATS to detect when a query is unclear and to interact with a user to clarify it? Because *a CRL-CATS expression sets expectations*, it is possible to make CATS detect missing information. For example, the "sell" post "for sale a car", may trigger a clarification question because the system knows that a car can be further specified and it can send to the poster a question such as: "can you tell what type of car?"

This opens many possibilities which we plan to explore further in the future. For the moment, we use a human operator, because we wanted to avoid a possibility burden from man-machine dialogue, and there are many cases in which more intelligence is required from our human operator far than can be expected from a state-of-the art AI-based dialogue system

Chapter 8. Operational experimentation, evaluations and discussion

Introduction

To evaluate the success of this experiment, we have to check the current status of the CATS. We also should use some metrics and compare it with other systems.

Due to the nature of the system in targeting end users, it is difficult to perform end-to-end evaluation without surveying them directly. However, this was impractical and beyond our capabilities. In particular, we could not (yet) persuade the mobile operator to include questions about the CATS service in their user satisfaction previews campaign. The only thing we can say is that the system is in operation, with a steady flow of about 30 posts per day, jumping to 40 posts per hour in the 2 to 3 days following advertisement mentioning CATS. We also made ourselves a small users' survey.

Hence, we used well known metrics to evaluate the CE component, assuming it plays a major role in the overall performance of the system.

The first part of this chapter presents the strategy we used in developing and deploying CATS. We then present the evaluation of the system. Finally we discuss the scalability of CATS.

8.1 Deployment and development strategy

8.1.1 Phased-based approach

CATS integrates different technologies to perform its tasks. As shown in Figure 78: higher level Architecture of the CATS system CATS is a sophisticated integration system that synchronizes between different layers and technologies: GSM, NLP and RDBMS. The software has been implemented using the .Net development environment. The choice of Windows environment was constrained by the usage of other components such as EnCo and NowsSMS.

A modular approach has been used for building CATS, so that each module can be tested and developed separately. Having this in mind, the NLP part was developed in full separation of other components. This included the development of the dictionary and the extraction rules. Testing is performed, at this level for this part, using the posts used for the development of the elementary rules and dictionary. This strategy of modular approach is crucial for successive steps of modification and maintenance, because we can perform any modification in the rules or the dictionary without affecting the overall system or interrupting the operation of the system.

The next component is responsible for handling the database, it is in charge of collecting each CRL-CATS expression produced by the CE, process it to produce the corresponding SQL statements, and forward the results (if any) to the SMS gateway adapter.

The GSM adapter is responsible for the communication tasks. Its main function it to interface with NowSMS, a middleware that handles the SMS messages through the connection with the GSM operator.

Once each component has been tested alone, integrated testing is necessary to validate the system. Since we did not have at this point any agreement with any mobile operator, we used a GSM modem to interface with the mobile operator network and to perform the integration testing.

To complete the process of this experimentation, we have to put the system online and to make it accessible by customers in order to verify and validate the hypothesis mentioned at the beginning of the thesis. This step was impossible without the cooperation of one mobile operator, to provide us with a short number and the needed support to deploy this service.

After 2 months of correspondence with FASTLINK, the largest mobile operator in Jordan, with 2 million subscribers, we convinced them to accept this novel service. They agreed to provide us with a short number, opened for free for testing purposes. They also performed testing from their side. Finally 3 months of assessment, they were convinced and fully deployed this service.

8.1.2 Availability of the service

This service is available for FASTLINK subscribers. They only have to send their posts to the assigned short number 90050. The service is available in Arabic for the Cars and Real Estate domains only. The mobile operator provides the short number, the link and does the billing.

From the users' point of view, CATS is considered a Pull service: users ask for the information and initiate the requests.

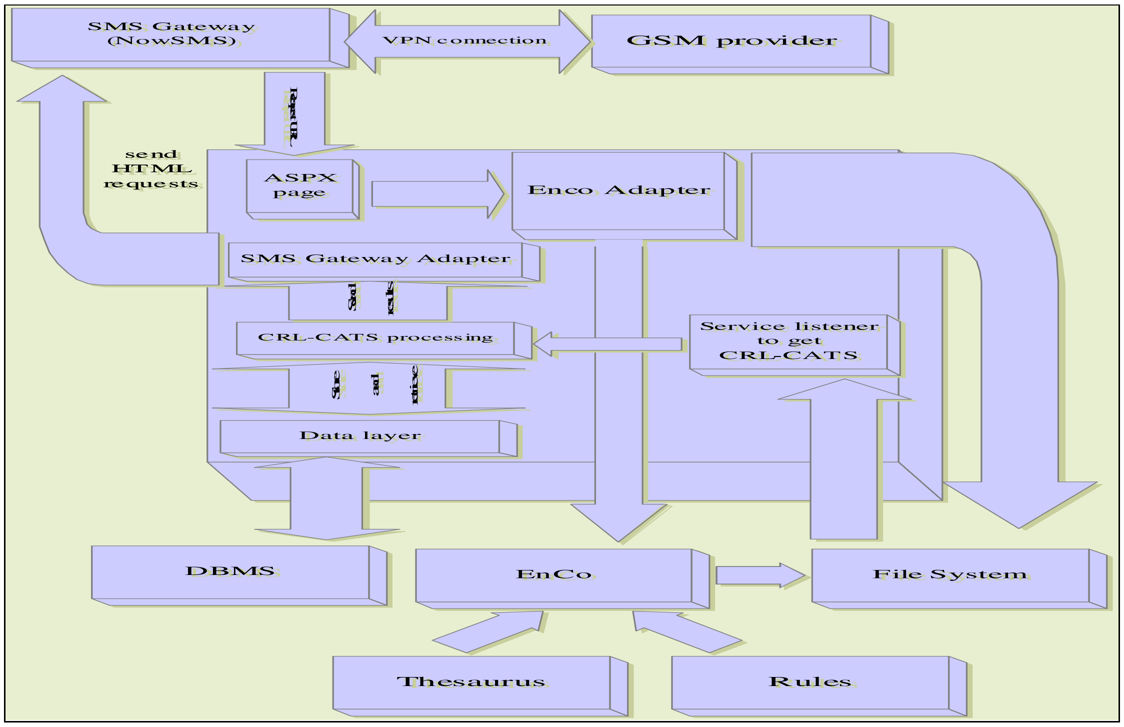


Figure 78: higher level Architecture of the CATS system

By contrast, most of the current services provided via SMS are Push based, which means that information is sent from a server through a channel to clients or users who have subscribed to that channel. The channels are chosen (subscribed to) by the users. Afterwards, the client waits for the information to be pushed to him. When a channel has a new piece of information fed into it, the server initiates a push all subscribers, completing one cycle of the server-push workflow.

Client-Pull (Pull), also know as On-Demand Service, is a technology used by the client to obtain information. Based on a user's manual action, an information request is sent to the server. The server responds with the results of the request and gives it back to the client. In this type of communication, the user plays a more active role, he is the one who initiates and asks for the information not the reverse.

8.1.3 Status of the system

This service is currently available in Jordan, where thousand of people have already used it to sell or buys cars or properties. The number of posts received depends on many factors such as the season or the marketing campaign by the mobile operators. Usually after some marketing we get on average 1000 posts per day, otherwise we get 20 ~30 posts per day.

8.2 Evaluation

8.2.1 Methods of measuring the efficiency of CATS

Because the CATS system is targeting end users, we performed an end-to-end evaluation of the system by surveying users directly. Initially, the system has been deployed with the cooperation of the largest mobile operator in Jordan (FASTLINK) to perform this evaluation. We have explained the system to a sample of around 200 users from different backgrounds, asking them to test the system by posting "sell" and "looking for" SMS messages.

Generally, the feedback was positive: 95% of the participants said that results were accurate. The rest said that the results should be more precise. We have noticed that 70% of the messages are of the "looking for" type.

However, it is important to provide a quantitative metrics to measure performance and accuracy. As a restricted domain information system, CATS is a task-oriented system, and that should be considered in the evaluation. [Diekema, et al. 2004] specifies different user evaluation dimensions for this type of systems. In our case, CATS is a multi-component system and the CE component is the most important in evaluating the completeness, relevance, and accuracy of the responses. Additionally, it is also important to measure the performance of CATS in terms of the time it takes to respond to the users.

We used precision and recall rates to measure the quality of our answers. They were calculated as follows [Cardie 1997, Sitter, et al. 2004]:

$$precision = \frac{\text{number of correct entities identified by the system}}{\text{number of entities identified by the system}}$$

$$recall = \frac{\text{number of correct entities identified by the system}}{\text{number of entities identified by a human}}$$

$$F - Measure = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

8.2.2 Experiment and results

We designed and conducted an experiment to evaluate the usefulness and performance of our content extractor. A set of real posts was used as the testbed. It consisted of 100 posts per type per domain, not used in the development of the systems, and randomly selected from the received posts during the real operation of CATS.

A human experimenter manually processed these posts to identify all entities of interest. Theses posts contained a significant amount of typos, spelling errors, and grammatical mistakes. This added difficulty to the entity extraction process. On the other hand, however, it allowed us to test our system's robustness for noisy data sets.

Table 27 shows the results for 100 "looking for" posts in the Cars domain. The results show high precision and recall for the query posts. This is true for both numerical and textual entities. This table shows also what interests users when they formulate a request in the Cars domain (make, year, model and price respectively).

Table 27: extraction results for "looking for" posts in the Cars domain

	Precision	Recall	F-measure	number of correct entries extracted by the system	number of total entities extracted by the system	number of total entities extracted by human
want	0.951	0.980	0.966	98	103	100
year	0.910	0.973	0.940	71	78	73
price	0.913	1.000	0.955	21	23	21
make	0.968	0.989	0.978	90	93	91
model	0.941	0.923	0.932	48	51	52
average	0.937	0.973	0.954			
Color entities (6), motor size (3) and country (16) where not included due to low frequency in the experimental data.						

The results for "sell" posts in the Cars domain presented in table 28 shows less precision and recall. Specifically, the numerical entities have lower measures than textual entities. This problem is associated with the drop of any hint words or units with numbers in the text, making these values difficult to identify (*BMW 99 520 at 8000*). On the hand, other entities have shown higher P&R. The results also show that the posters of "sell" type ads are more interested in specifying the following attributes: features, year, make, model, the price and color.

Table 28: extraction result for "sell" posts in the Cars domain

	Precision	Recall	F-measure	number of correct entries extracted by the system	number of total entities extracted by the system	number of total entities extracted by human
sale	0.960	0.960	0.960	96	100	100
year	0.762	0.952	0.846	99	130	104
price	0.750	0.750	0.750	39	52	52
make	0.989	0.938	0.963	90	91	96
model	0.875	0.787	0.828	70	80	89
feature	0.972	0.875	0.921	140	144	160
color	0.950	0.809	0.874	38	40	47
average	0.907	0.832	0.867			
Mot "motor size" entity (3), country (2) where not included due to low frequency.						

For the "sell" posts in the Real Estate domain shown in table 29, the results indicate generally higher precision values than recall. The entities with lowest recall are "area size" and "price"; both are numerical, followed by location. The results show also that users, when formulating a search post, are interested more in specifying the location than other entities.

Table 29: extraction results for "looking for" posts in the Real Estate domain

	Precision	Recall	F-measure	number of correct entries extracted by the system	number of total entities extracted by the system	Number of total entities extracted by human
want	0.980	0.980	0.980	98	100	100
property	0.904	0.870	0.887	94	104	108
area	0.978	0.726	0.833	45	46	62
price	0.965	0.714	0.821	55	57	77
purpose	0.938	0.968	0.952	30	32	31
location	0.939	0.842	0.888	123	131	146
consist	0.947	0.947	0.947	36	38	38
average	0.950	0.864	0.901			
room = 14, type =7, feature = 2, flr = 2 are not included						

Table 30 shows the results of "sell" posts in the Real Estate domain. Precision is higher than recall. The price has the lowest recall despite its high precision, suggesting how ambiguous numerical values can be. The location entity has a high recall, not like for the "looking for" posts, suggesting that the potential sellers are more accurate in specifying the location than potential buyers.

Table 30: extraction results for "sell" posts in the Real Estate domain

	Precision	Recall	F-measure	number of correct entries extracted by the system	number of total entities extracted by the system	number of total entities extracted by human
sale	0.960	0.960	0.960	96	100	100
property	0.931	0.896	0.913	95	102	106
area	0.942	0.739	0.828	65	69	88
price	0.963	0.578	0.722	26	27	45
feature	0.931	0.818	0.871	27	29	33
location	0.907	0.972	0.938	137	151	141
consist	0.952	0.984	0.968	60	63	61
average	0.938	0.838	0.875			
purpose = 2, room= 13, type=5 are not included						

We also remark that, "looking for" posts show higher F-measure than "sell" posts. On the other hand, the Cars domain has a higher F-measure than the Real Estate domain, reflecting its higher complexity. We also observe that "numerical values" entities have lower F-measure than textual entities, suggesting that numerical entities are harder to detect or to identify correctly.

8.2.3 Assessment

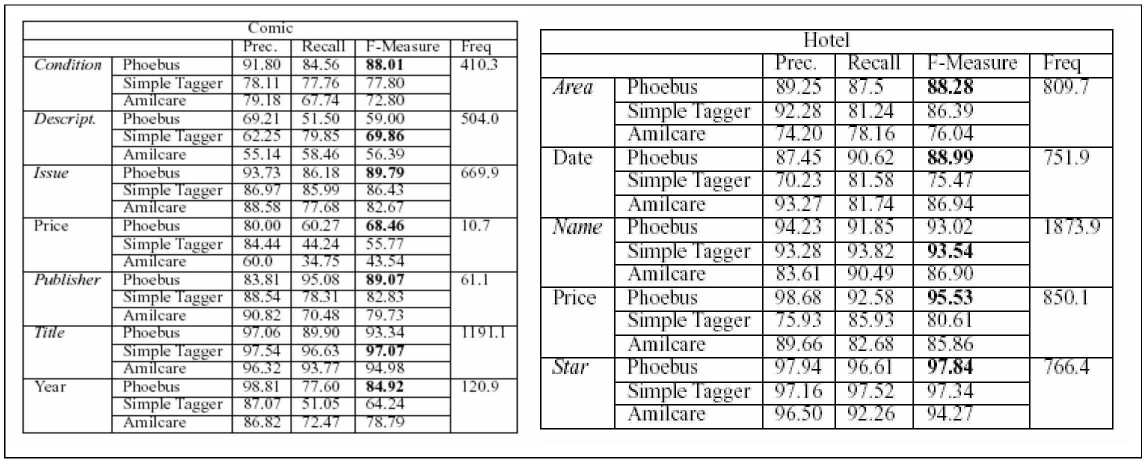
In general, the results indicate that our content extractor performs well in identifying different entities. Considering that the spontaneous free posts collected to conduct this evaluation were much noisier than the news articles used in MUC evaluations, CATS has a higher recall and precision than the results reported by MUC (unrestricted text: 60-70% R, 65-75% P, Semi-structured text: 90% R/P) [Cardie 2005].

For further assessment of our system, we compare our results with other more recent systems that use English: Phoebus [Michelson and Knoblock 2005], SimpleTagger [McCallum 2002], and AmilCare [Ciravegna 2001]. Phoebus uses semantic annotation for handling ungrammatical and unstructured text. SimpleTagger is a suite of text processing tools that is an implementation of Conditional Random Fields (CRF) which has been used in information extraction. Amilcare uses shallow natural language processing for information extraction. Unfortunately, we could not use any of the above directly for comparisons. Therefore, we use the comparison study conducted by [Michelson and Knoblock 2005] in two domains: hotel postings and comics books. The results are shown, for the three systems, in table 31.

For the *price* entity CATS, scored an F-Measure (for all types and domains) of 81%, higher than the three systems in the comics books domain. For the hotel postings, it is better than Simpletagger and Amilcare but worse than Phoebus. For the *year* entity, in the Cars domain, CATS scores 89% higher than all other systems under consideration. For the *location* entity (in the Real Estate domain) which corresponds to the *area* in the hotel domain, CATS scored 91%, again higher than all other systems.

Hence, CATS despite the free, spontaneous and noisy nature of its input, has surpassed other systems in quality.

Table 31: extraction results for comics books and hotel



Comic					
		Prec.	Recall	F-Measure	Freq
Condition	Phoebus	91.80	84.56	**88.01**	410.3
	Simple Tagger	78.11	77.76	77.80	
	Amilcare	79.18	67.74	72.80	
Descript.	Phoebus	69.21	51.50	59.00	504.0
	Simple Tagger	62.25	79.85	**69.86**	
	Amilcare	55.14	58.46	56.39	
Issue	Phoebus	93.73	86.18	**89.79**	669.9
	Simple Tagger	86.97	85.99	86.43	
	Amilcare	88.58	77.68	82.67	
Price	Phoebus	80.00	60.27	**68.46**	10.7
	Simple Tagger	84.44	44.24	55.77	
	Amilcare	60.0	34.75	43.54	
Publisher	Phoebus	83.81	95.08	**89.07**	61.1
	Simple Tagger	88.54	78.31	82.83	
	Amilcare	90.82	70.48	79.73	
Title	Phoebus	97.06	89.90	93.34	1191.1
	Simple Tagger	97.54	96.63	**97.07**	
	Amilcare	96.32	93.77	94.98	
Year	Phoebus	98.81	77.60	**84.92**	120.9
	Simple Tagger	87.07	51.05	64.24	
	Amilcare	86.82	72.47	78.79	

Hotel					
		Prec.	Recall	F-Measure	Freq
Area	Phoebus	89.25	87.5	**88.28**	809.7
	Simple Tagger	92.28	81.24	86.39	
	Amilcare	74.20	78.16	76.04	
Date	Phoebus	87.45	90.62	**88.99**	751.9
	Simple Tagger	70.23	81.58	75.47	
	Amilcare	93.27	81.74	86.94	
Name	Phoebus	94.23	91.85	93.02	1873.9
	Simple Tagger	93.28	93.82	**93.54**	
	Amilcare	83.61	90.49	86.90	
Price	Phoebus	98.68	92.58	**95.53**	850.1
	Simple Tagger	75.93	85.93	80.61	
	Amilcare	89.66	82.68	85.86	
Star	Phoebus	97.94	96.61	**97.84**	766.4
	Simple Tagger	97.16	97.52	97.34	
	Amilcare	96.50	92.26	94.27	

As to the performance of the system in terms of capacity and time to respond, it has shown high performance. CATS was tested for one post per second and it has performed well. We also noted, as presented in Appendix E, that during some times it was able to process more than 10 posts/minute efficiently (including response generation). The average response time is around 10~30 seconds. Compare this with the 12 minutes to process 100 messages using FASTUS (8 posts/minute, and 36 hours to process 100 messages (more than 3 hours/post) using TACITUS [Hobbs, et al. 1996].

Table 32: sample of input posts and corresponding CATS response

Input Post	CATS response
5/24/2005 **2:23:39** PM Message In : +962795412191, مطلوب سياره بطه م 85 الي 88 سعر طري مش فحص	5/24/2005 **2:23:55** PM Out Message: +962795412191, *mercedes 795313472:200 795280784:200 795510100:200 795444474:200 795594258:200 795145032:200 795400445:200 795642424:200 795167818:200 795040777:200
5/24/2005 2:21:10 PM Message In : +962795562722, مطلوب سياره أوبل أوميغا 95 بسلفه 2000 واقساط شهريه	5/24/2005 2:21:55 PM Out Message: +962795562722, *opel 795412191:omega 795682213:omega 795996005:omega 795928857:omega 795061726:omega 796870777:omega

5/24/2005 2:25:41 PM Message In : +962795581985,مطلوب أرض أو شقه بسعر معقول في العقبه للمراجعه	5/24/2005 2:25:57 PM Out Message: +962795581985, *aqaba 795996677

8.3 Scalability of CATS

8.3.1 Adaptation to other domains

Adapting CATS to other domains is possible. To measure the resources needed to add another domain to CATS, we performed an experiment of adding Marriage Service as a new domain. In this service, a person (man or woman) encodes his/her request as a (personal ad) in natural language. In developing this new domain we went through the following steps.

	Task	resources
1	Collecting web-based corpus (300 posts)	2 man-hours
2	Analysis of the sublanguage used and extending CRL-CATS to accommodate this domain	1 man-week
3	Adding domain-specific entries to the dictionary (600 entries)	3 man–days
4	Developing the rules (100 rules were added)	1 man-week
5	Updating the database and other related components	1 man-week (estimate)

By the end of step 4, we had a complete extraction system for this new domain.

The SMS-based corpus collected for the Cars and Real Estate is useful in adapting CATS to new domains. As a matter of fact, it can be projected to new domains for certain entities such as numerical values or the way users express "looking for". The time taken to port to new domains depends on how close the new domain is to the current SMS-based corpora. The Marriage domain is relatively more distant than a service for buying and selling "mobile phones".

However, the collected corpus from CATS is unique. If used effectively, it can be useful on building applications that require spontaneous inputs from users. The usefulness of this corpus is not restricted to classified ads domains but can be extended to other domains.

8.3.2 Extensions to other languages

Different approaches can be used to port CATS to other languages. One approach to be considered is by using machine translation (MT) (figure 79). The classified ads posts are translated from source language to Arabic and then Content extraction is performed to extract CRL-CATS.

This approach might look applicable and straightforward. However, from the practical perspective it is not. As a matter of fact, for restricted domains where we have special sublanguage with specific terminology, traditional MT is not useful. To demonstrate this, suppose we have the following "sell" posts for a car in English: "*Volvo station 1998 duty unpaid full options white good condition*". Using

“Golden Al Wafi translator” which is English –Arabic MT produced by ATA, we get the following result:

“محطة خياراتِ غير مدفوعةٍ كاملةٍ واجبَ Volvo الـ1998 حالة جيدة بيضاء”

Which has nothing to do with the original post: ”full unpaid station option duty Volvo 1998 good condition white”.

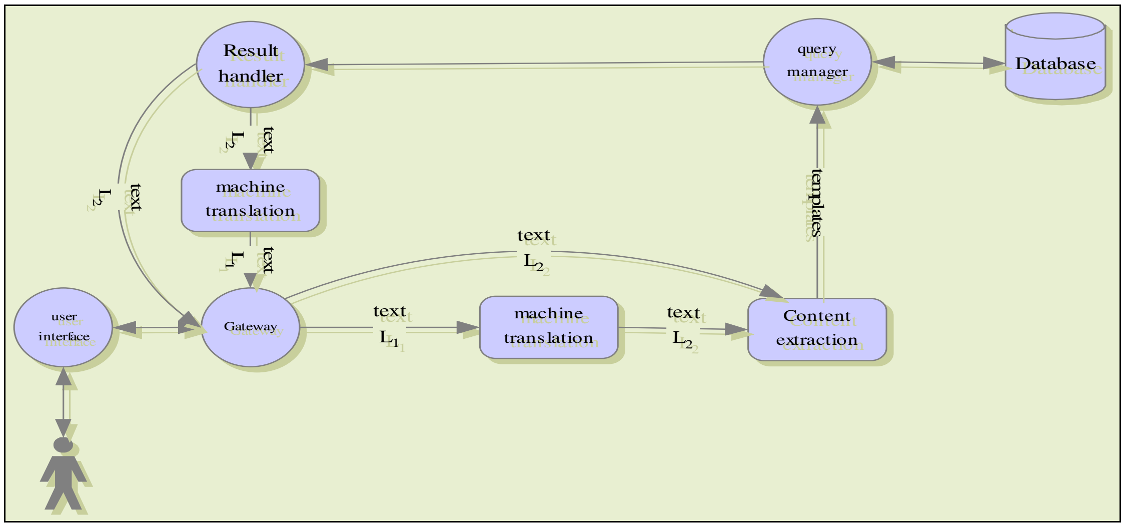


Figure 79: extension to other languages by machine translation

The other approach is by localization of the CE (figure 80), where each language has its own CE that produces the same CRL-CATS. The localization task can be implemented by cloning the Arabic CE to other languages.

For comparison, two approaches are experimented; the first one is by cloning and adapting the CATS CE to French. The resulted cloned CE will process French posts (translated from Arabic Corpus) and will produce CRL-CATS expressions. The second approach is by adapting an existing French tourism-based CE system (IF) to the Cars domain and testing this system using the same translated corpus. For comparing with the first approach, “IF” format should be translated into the CRL-CATS representation and both approaches should be evaluated based on recall and precision. Other factors should also be considered, such as the resources required to perform the cloning or the adaptation.

However, as the sublanguages of different languages for the same task (e.g. Cars) are quite near to each other[16], and quite restricted, it should be possible to develop example-based or statistical MT systems. To develop for example, a MT system from SMS-Cars-fr to SMS-Cars-ar, we would only need a “large enough” corpus.

The question is (again) how to get such a corpus and what “large enough” means in this case. By comparison with the BTEC used by CSTAR we hypothesize that about 15000 SMS (in the 2 languages) would be more than enough. French versions could actually be produced by a special “deconverter” from CRL-CATS to SMS-Cars-fr.

[16] Kittredge said the distant between SL-en to SL-fr, for example, is much smaller than that between SL-en and English, or SL-fr and French, and we can confirm it in the case of the SMS.

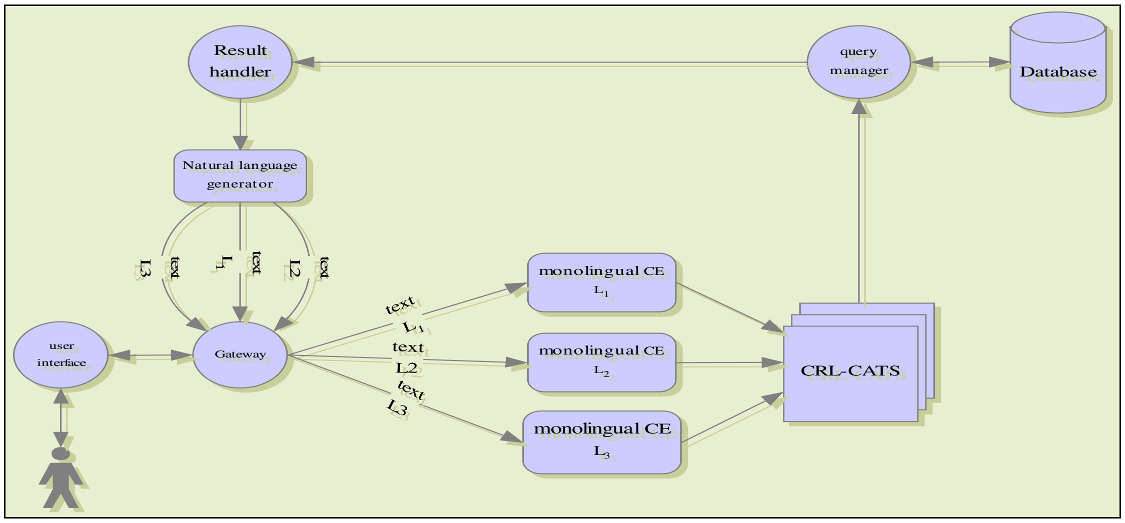


Figure 80: porting CATS to other languages by localization of the extractors

To avoid localization of CE for each language, one can think of using UNL for building a language-independent version of CATS. This architecture will take advantage of the current UNL infrastructure. [Boitet 2005] has shown that "abstract pivot" technique can work, confuting previous thoughts that it is impossible to construct an all-purpose interlingua [Arnold, et al. 1994]. Hence, UNL can describe the classified ads posts despite their telegraphic structure sometimes.

In this architecture, the posts are directed to a language server to be enconverted to the UNL expressions. In turn, these UNL expressions are transformed into CRL-CATS by using a knowledge extractor (Universal extractor). The CRL-CATS expressions are then processed as by the current CATS system. To formulate a response, results can be deconverted into the target language using the language centers (Figures 81-82).

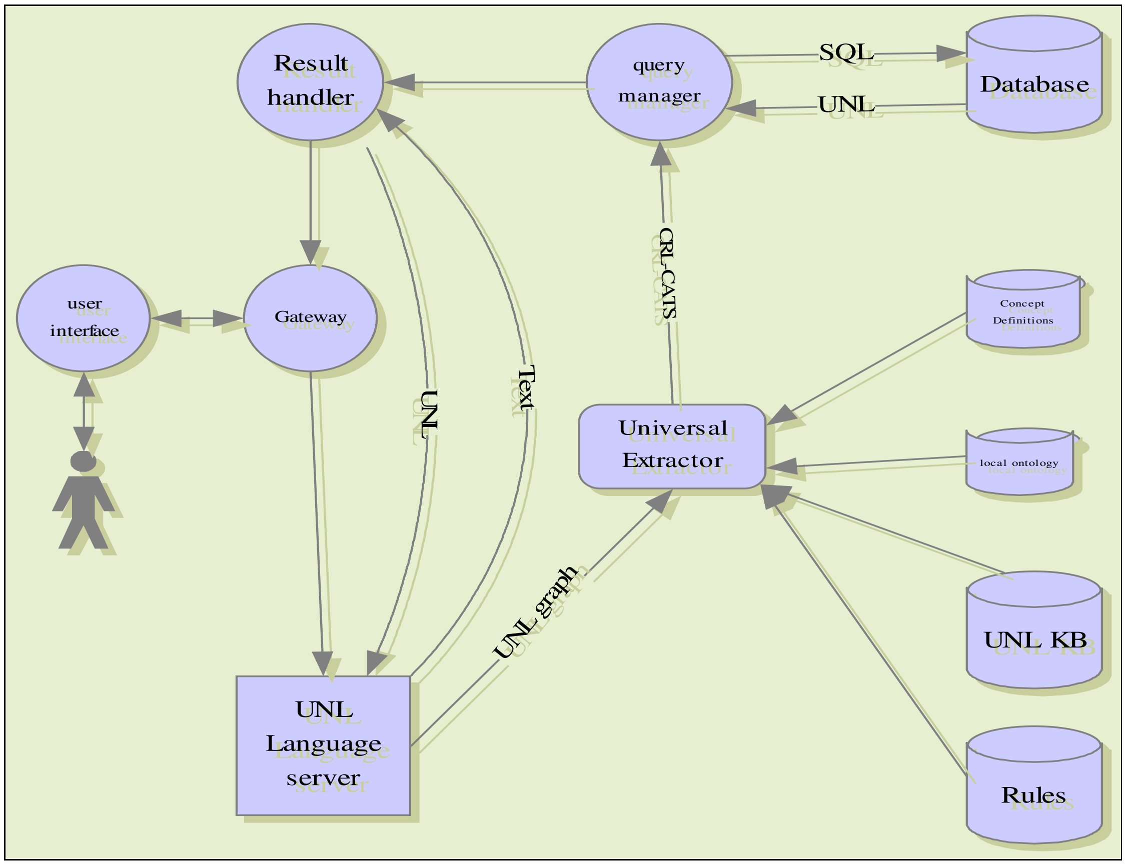


Figure 81: language-independent architecture of CATS

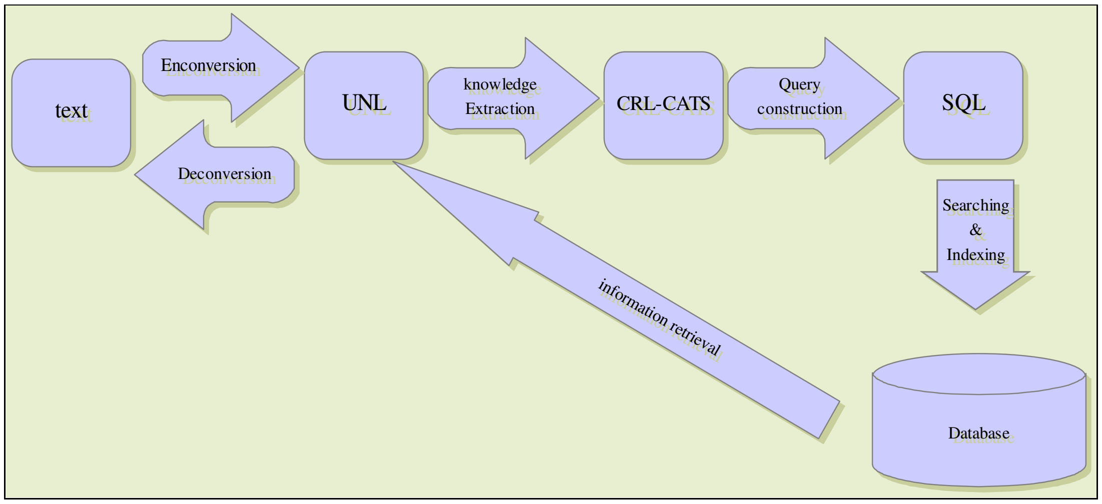


Figure 82: main operations in the language-independent architecture

Theoretically, this can be the best solution. Practically, is it is impossible to produce correct UNL expressions from classified ads posts without the modification of the general purpose enconversion systems that currently exist.

This operation would imply "normalization" of the input posts as well-formed , complete utterances, which is much more difficult than extracting content (of type shown in advance) from noisy, incomplete and ungrammatical input.

Trying to use "fragmentary UNL" as an intermediate representation also cannot work (or could mean transforming "chunks" into small UNL graphs). Why? Simply because, anyway, one cannot translate these chunks into UNL without a study of the sublanguage.

Conclusion of Part C

The whole QA part of CATS has been described, including the design and implementation of the database, the semantic processing (including inference of missing information), and response generation (with possible question relaxation and follow up in case of no match or partial match). We have shown that a RDBMS is used to perform the matching inferring efficiently using the semantic information embedded within the CWs.

The evaluation quality and response time shows that, despite the nature of the free, spontaneous and noisy input, CATS has surpassed other systems. We have also shown that it is possible to expand CATS to accommodate other domains. Porting it to other language is also possible and under investigation in Geta-Imag-Clips to find the best method.

Conclusion of the thesis

We have shown in this thesis by surveying some e-commerce systems that none of them is able to handle spontaneous users' requests online. Some of the studied systems avoided the hard problem of supporting free natural language interface by simplifying the user interaction styles either by using form filling or by using controlled languages. Other systems felled short of supporting full natural language interface because they used inadequate NLP techniques.

The hypothesis that it is necessary and possible to build (multilingual) NL-based e-commerce systems with mixed sublanguage and content-oriented methods has been verified by building CATS. It is a SMS-based classified ads selling and buying platform, which allows users to send classified ads of the articles/goods they would like to sell and to search for the goods/articles they desire using full natural language interface. We first studied the classified ads sublanguage to determine the linguistic features and the domain knowledge, both are essential in determining the adequate NL processing method. We first used a web based corpus to build the basic system which covers the Cars and Real Estate domains. This initial experimental deployment of CATS was to collect real SMS-based spontaneous data, which we used to fine tune the system.

To enable semantic processing, CRL-CATS was defined to capture the meaning of a classified ad post. The semantic grammars of content extraction are coded using the EnCo specialized language for linguistic programming which we used previously in developing the first Arabic-UNL enconverter. To enhance the process of coding using EnCo, we have developed the DED methodology that facilitates this process and provides the means for a systematic and efficient coding.

We have shown how we implemented the response generation component, which is based on semantic matching ("looking for" and "sell" posts) and reasoning and is able to handle "no answer situations". We have shown different follow up functionalities implemented by the current version which were enabled by the usage of DB.

We have shown that CATS is not like other experimental NL systems, it was designed from the beginning to be a "production system". Software engineering aspects issues such as modularity, efficiency, robustness, etc., are not top priorities in NLP research. Many questions concerning software design treated as detailed information not necessary to the main task. This is in contrast to our approach shown in this thesis in which all software engineering aspects are considered from the beginning.

CATS is currently deployed in Jordan by the largest mobile operator (Fastlink) after passing intensive testing by them. Testing the content extraction component with a real noisy free text shows a 90% F-measure. The average response time is around 10~30 seconds calculated during peak time (10 posts/minute).

The corpus produced by CATS is unique and can be exploited in building spontaneous NLP systems. Additionally, we can explore different methodologies to build similar system. We can also explore other techniques to for enhancing the quality of CATS. As an example, we can test the use of spell checkers to handle spelling variations in these types of spontaneous inputs and measure the effects of this approach on quality and to check for any performance tradeoff. Additionally, we would like to enhance CE in general. We think this can be achieved with the help of the corpus produced by CATS.

Additionally we will work in porting CATS to other domains and other languages. We have given an example of porting CATS to other domain. We would like to explore this further by using DED.

Furthermore, CATS is being investigated for multilinguality by exploring different approach to localization. This work is part of research which is currently conducting in GETA.

BIBLIOGRAPHY

Abuleil S, and Evens M (1998) Discovering Lexical Information by Tagging Arabic Newspaper Text. *Proc. of Workshop on Semitic Language Processing. COLING-ACL'98*, Université de Montréal, 10-14 August 1998, UdM & ACL.

Adams KC (2001) The Web as a Database: New Extraction Technologies and Content Management. *Online Magazine,* Vol. 25, N° 2.

Agrawal R, Somani A, and Xu: Y (2001) Storage and Querying of E-Commerce Data. . *Proc. of 27th International Conference on Very Large Data Bases* Roma, Italy, September 11-14, Morgan Kaufmann 2001, ISBN 1-55860-804-4 pp. 149-158.

Allen J, and Thompson C (2005) *Interfacing agents with natural language* 2005 [cited 2006]. Available from http://www.uark.edu/rd_vcad/images/allen.pdf.

Andrenucci A, and Sneiders E (2005) Automated Question Answering: Review of the Main Approaches. *Proc. of the 3rd International Conference on Information Technology and Applications (ICITA'05)*, Sydney, Australia, IEEE, 1/, pp. 514-519.

Androutsopoulos I, Ritchie GD, and Thanisch P (1995) Natural Language Interfaces to Databases: An Introduction. *Journal of Natural Language Engineering,* Vol. 1, N° 1.

Appelt DE, and Israel DJ (1999) Introduction to Information Extraction Technology: a Tutorial. *Proc. of 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, Stockholm, Sweden, August 1999.

Arnold D, Balkan L, Meijer S, Humphreys RL, and Sadler L (1994) Machine Translation: an Introductory Guide. Blackwells-NCC, London.

Benamara F (2004) Cooperative Question Answering in Restricted Domains:the WEBCOOP Experiment. *Proc. of ACL'04 Workshop on Question Answering in Restricted Domains*, Barcelona.

Boitet C (2005) Gradable quality translations through mutualization of human translation and revision, and UNL-based MT and coedition. In *Universal Networking Language, advances in theory and applications*, edited by Cardeõsa J, Gelbukh A and Tovar E, Mexico, pp. 393—410.

Brachman RJ, and Levesque HJ (2004) Knowledge Representation and Reasoning. Morgan Kaufmann

Bruthiaux P (1994) Functional variation in the language of classified ads. . *Perspectives: Working Papers of the Department of English, City Polytechnic of Hong Kong,* Vol. 6, N° 2, pp. 21-40.

Buitelaar P, Netter K, and Xu F (1998) Integrating Different Strategies for Cross-Language Information Retrieval in the MIETTA Project. *Proc. of TWLT14, Enschede*, Netherlands, December 1998.

Cardie C (1997) Empirical Methods in Information Extraction. *AI Magazine,* Vol. 18, N° 4, pp. 65-79

Cardie C (2005) *Information extraction* 2005 [cited 2005]. Available from http://www.cs.cornell.edu/courses/cs674/2005sp/Handouts/IE-intro-4up.pdf.

Chai J, Horvath V, Nicolov N, Stys M, Kambhatla N, Zadrozny W, and Melville P (2002) Natural Language Assistant - A Dialog System for Online Product Recommendation. *AI Magazine,* Vol. 23, N° 2, pp. 63–75.

Ciravegna F (2001) Adaptive information extraction from text by rule induction and generalisation. . *Proc. of the 17th International Joint Conference on Artificial Intelligence*, Seattle, Usa, August 2001.

Copestake A (2003) *Natural Language Processing* University of Campridge, 2003 [cited 2005]. Available from http://www.cl.cam.ac.uk/Teaching/2002/NatLangProc/nlp1-4.pdf.

Covington MA (1992) A dependency parser for variable–word–order languages. In *Computer assisted modeling on the IBM 3090: Papers from the 1989 IBM Supercomputing Competition*, edited by Billingsley KR, Brown III HU and Derohanes E, Athens, Greece, Baldwin Press, pp. 799–845.

Das S, Chong EI, Eadon G, and Srinivasan: J (2004) Supporting Ontology-Based Semantic matching in RDBMS. *Proc. of Thirtieth International Conference on Very Large Data Bases*, Toronto, Canada, August 31 - September 3, pp. 1054-1065.

Davis R, Shrobe H, and Szolovits P (1993) What is A Knowledge Representation? . *AI Magazine* Vol. 14, N° 1, pp. 17-33.

Diekema AR, Yilmazel O, and Liddy ED (2004) Evaluation of Restricted Domain Question-Answering Systems. *Proc. of ACL'04 Workshop on Question Answering in Restricted Domains*, Barcelona.

Dini L (1998) Parallel Information Extraction Systems For Multilingual Information Access. *Proc. of Third European Robotics, Intelligent Systems & Control Conference (Euriscon 98)*, session 6 (Adaptive and Multilingual Information Extraction Systems), June, 1998.

Doan Nguyen H, and Kosseim L (2004) The Problem of Precision in Restricted-Domain Question Answering – Some Proposed Methods of Improvement. *Proc. of ACL'04 Workshop on Question Answering in Restricted Domains*, Barcelona.

Engels R, and Bernt Bremdal (2000) Information extraction: State-of-the-Art-Report. On-To-Knowledge Consortium, July 2000.

Gaasterland, Godfrey P, and Minker J (1992) Relaxation as a platform of cooperative answering. *J. Intell. Inform. Syst.,* Vol., N° 1, pp. 101-120.

Gao X, and Sterling L (1998) Classified Advertisement Search Agent (CASA): a knowledge-based information agent for searching semi-structured text. *Proc. of Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, pp. 621-624.

Goweder A, and De Roeck A (2001) Assessment of a significant Arabic corpus. *Proc. of Arabic NLP Workshop at ACL/EACL 2001*, Toulouse, France, July 2001.

Grishman R (2003) Information Extraction. In *The Oxford Handbook of Computational Linguistics* edited by Mitkov R, Oxford university Press,

Guarino N, Masolo C, and Vetere G (1999) OntoSeek: Content-Based Access to the Web *Intelligent Systems and Their Applications, IEEE [see also IEEE Intelligent Systems],* Vol. 14, N° 3, pp. 70-80.

Hammo B, Abu-Salem H, Lytinen S, and Evens M (2002) QARAB: A Question Answering System to Support the Arabic Language. *Proc. of Computational Approaches to Semitic Languages workshop*, University of Pennsylvania., 11th July 2002.

Heinecke J, and Toumani F (2003) A Natural Language Mediation System for E-Commerce applications: an ontology-based approach. *Proc. of Workshop on Human Language Technology for the Semantic Web and Web Services,2nd International Semantic Web Conference*, Sanibel Island, Florida, October 20th 2003,.

Hobbs JR, Douglas Appelt, John Bear, Israel D, Kameyama M, Stickel M, and Tyson aM (1996) A Cascaded Finite-State Transducer for Extracting Information from Natural Language Text. In *Finite State Devices for Natural Language Processing*, edited by Roche E and Schabes Y, MIT Press.,

Internet usage statistics (2006) 2006 2006 [cited. Available from http://www.internetworldstats.com/stats.htm.

ITUArabic (2005) *Telecommunications indicators* 2005 [cited 2005]. Available from http://www.ituarabic.org/teleindicators.asp.

Kahane S (2001) What Is a Natural Language and How to Describe It? Meaning-Text Approaches in Contrast with Generative Approaches. *Proc. of CICLing-01*, Mexico, Feb. 2001, pp. 1-17.

Khoo CS-G (1997) The Use of Relation Matching in Information Retrieval. *Electronic Journal,* Vol. 7, N° 2.

Kittredge: RI (1982) Sublanguages. *American Journal of Computational Linguistics,* Vol. 8, N° 2, pp. 79-84.

Klein M, and Bernstein A (2001) Searching for Services on the Semantic Web using Process Ontologies. *Proc. of The First Semantic Web Working Symposium (SWWS-1).* Stanford, CA USA.

Kruschwitz U, and Al-Bakour H (2005) Users want more sophisticated search assistants: Results of a task-based evaluation. *American Society for Information Science and Technology,* Vol. 56, N° 13, pp. 1377 - 1393.

Lassila O (1990) Frames or Objects, or Both? *Proc. of Workshop Notes from the 8th National Conference on AI (AAAI-90): Object-Oriented Programming in AI*, Boston (MA).

Lassila O, and Swick RR (1999) *Resource Description Framework (RDF): Model and Syntax Specification. Recommendation* 1999 [cited. Available from http://www.w3.org/TR/REC-rdf-syntax/.

Lassila O, and McGuinness D (2001) The Role of Frame-Based Representation on the Semantic Web. *Linkping Electronic Articles in Computer and Information Science,* Vol. 6.

Lehtola A, Bounsaythip C, and Tenni J (1998) Controlled Language Technology in Multilingual User Interfaces. *Proc. of 4th ERCIM Workshop on "User Interfaces for All" Special Theme "Towards an Accessible Web"*, Stockholm, Sweden, 19-21 October 1998.

Lehtola A, Tenni J, Bounsaythip C, and Jaaranen aK (1999) *WEBTRAN: A Controlled Language Machine Translation System for Building Multilingual Services on Internet* 1999 [cited 2006]. Available from http://www.vtt.fi/tte/language/publications/mtsummit1999.pdf.

Lehtola A, KÄPYLÄ Y, BOUNSAYTHIP C, and TALLGREN M (2003) Multilingual and Ontological Product Cataloguing Tool – User Experiences. *Proc. of eChallenges-2003 - Building the Knowledge Economy: Issues, Applications, Case Studies.*, Bologna, Italy, 22-24 October 2003, IOS Press, pp. 947 - 954.

Leidner J (2003) Current Issues in Software Engineering for Natural Language Processing *Proc. of the HLT-NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS)*, Edmonton, Alberta, Canada.

Li X, and Roth D (2001) Exploring evidence for shallow parsing. *Proc. of the 2001 workshop on Computational Natural Language Learning*, Toulouse, France, 7/.

Long B (1994) *"Natural Language as an Interface Style* Dept. of CS, Univ. Toronto, May 1994 1994 [cited 2006]. Available from http://www.dgp.toronto.edu/people/byron/papers/nli.html.

Lytinen SL, and Gershman A (1986) ATRANS: automatic processing of money transfer messages. . *Proc. of the Fifth National Conference on Artificial Intelligence (AAAI-86)*, Philadelphia.

McCallum AK (2002) *MALLET: A Machine Learning for Language Toolkit* 2002 [cited. Available from http://mallet.cs.umass.edu.

Michelson M, and Knoblock CA (2005) Semantic annotation of unstructured and ungrammatical text. *Proc. of the 19th International Joint Conference on Artificial Intelligence(IJCAI-2005))*, Edinburgh, UK, August, 2005.

Mitamura T (1999) Controlled Language for Multilingual Machine Translation. *Proc. of Machine Translation Summit VII*, Singapore, September 13-17.

MKBEEM (2005) *(web site)* 2005 [cited. Available from http://www.mkbeem.com.

Moldovan D, Clark C, Harabagiu S, and Maiorano S (2003) COGEX: A Logic Prover for Question Answering. *Proc. of HLT-NAACL 2003*, Edmonton, May-June 2003, pp. 87-93.

Motik B, Maedche A, and Volz R (2002) A Conceptual Modeling Approach for Semantics-Driven Enterprise Applications. *Proc. of First International Conference on Ontologies, Databases and Application of Semantics (ODBASE-2002).* Irvine, USA, Springer.

Neumann G, and Xu F (2004) Intelligent information extraction. *Proc. of 16th European Summer School in Logic, Language and Information*, Université Henri Poincaré Nancy, France, 9-20 August, 2004.

Owei V, Rhee H-SS, and Navathe S (1997) Natural Language Query Filtration in the Conceptual Query Language. *Proc. of The Thirtieth Annual Hawwaii International Conference on System Sciences*

Pool J (2005) *Can Controlled Languages Scale to the Web?*
DISCUSSION DRAFT ONLY 2005 [cited 2006]. Available from http://utilika.org/pubs/etc/ambigcl/.

Riloff E, Schafer C, and Yarowsky D (2002) Inducing Information Extraction Systems for New Languages via Cross-Language Projection. *Proc. of COLING-02*, Taipeh, Aug. 2002.

Schank R (1975) Conceptual Information Processing. North Holland, Amsterdam.

Schank R (1991) Where's the AI? , Northwestern University, Institute for the Learning Sciences.

Schwitter R (2004) Representing Knowledge in Controlled Natural Language: A Case Study. *Proc. of Knowledge-Based Intelligent Information and Engineering Systems, 8th International Conference, KES2004*, Wellington, New Zealand, pp. 711-717.

Sekine S (1997) The Domain Dependence of Parsing. *Proc. of Applied Natural Language Processing(ANLP'97)*, Washington D.C., USA,, pp. 96-102.

Sérasset G, and Boitet C (2000) On UNL as the future "html of the linguistic content" & the reuse of existing NLP components in UNL-related applications with the example of a UNL-French deconverter. *Proc. of COLING-2000*, Saarbrücken, 31/7—3/8/2000, ACL & Morgan Kaufmann, 2/2, pp. 768—774.

Sitter AD, Calders T, and Daelemans W (2004) *Formal Framework for Evaluation of Information Extraction*
2004 [cited. Available from http://citeseer.ist.psu.edu/648270.html.

Smeaton AF, O'Donnell R, and Kelledy F (1995) Indexing structures derived from syntax in TREC-3: System description. *Proc. of Overview of the Third Text REtrieval Conference (TREC-3)*, Gaithersburg, pp. 55-67.

Smith BC (1982) Reflection and Semantics in a Procedural Language. Massachusetts Institute of Technology.

Smith M (2005) *Worldwide cellular connections exceeds 2 billion* 2005 2005 [cited Press Release 2005. Available from http://www.gsmworld.com/news/press_2005/press05_21.shtml.

Somers H, Black B, Nivre J, Lager T, Multari A, Gilardoni L, Ellman J, and Rogers A (1997) Multilingual Generation and Summarization of Job Adverts: the TREE Project. *Proc. of Fifth Conference on Applied Natural Language Processing*, Washington, DC, April 1997, pp. 269-276.

Sophie AUBIN, NAZARENKO A, and NÉDELLEC C (2005) Adapting a General Parser to a Sublanguage *Proc. of International Conference RANLP - 2005 (Recent Advances in Natural Language Processing)*, Borovets, Bulgaria, 21-23 September 2005.

Sowa JF (1999) Relating Templates to Language and Logic. In *Information Extraction: Towards Scalable, Adaptable Systems*, edited by Pazienza MT, Springer, Templeton M, and Burger JD (1983) Problems In Natural-Language Interface To DSMS With Examples From EUFID. *ANLP,* Vol., pp. 3-16.

Tennant HR, Ross KM, Saenz RM, Thompson CW, and Miller JR (1983) Menu-based natural language understanding. *Proc. of 21st annual meeting on Association for Computational Linguistics*, Cambridge, Massachusetts, pp. 151 - 158

Thompson CW, Ross KM, Tennant HR, and Saenz RM (1983) Building Usable Menu-Based Natural Language Interfaces To Databases. *Proc. of VLDB-1983*, pp. 43-55.

Uchida H (1999) *Enconverter Specifications* UNU/IAS UNL Center, 1999 1999 [cited. Available from http://www.undl.org.

Uchida H, and Zhu M (2003) *The Universal Networking Language specification, version 3.0* UNDL Foundation, 2003 2003 [cited. Available from http://www.undl.org.

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

Uchida H (2006) A Common Web Language (CDL). *Proc. of The Semantic Web Interest Group SwigAtTp2006 , as part of the W3C Technical Plenary Week.*, France, Thursday, 2nd March, 2006.

Vauquois B, and Chappuy S (1985) Static grammars: a formalism for the description of linguistic models. *Proc. of TMI-85 (Conf. on theoretical and metholodogical issues in the Machine Translation of natural languages)*, Colgate Univ., Hamilton, N.Y., Aug. 1985, pp. 298-322.

Vertan C, and Hahn Wv (2003) Menu choice translation: a flexible menu-based controlled natural language system. *Proc. of Controlled language translation Workshop, EAMT-CLAW-03*, Dublin City University, 15-17 May 2003, pp. 194-199.

Vertan C (2004) Querying Multilingual Semantic Web in Natural Language. *Proc. of Semantic Web Applications and Perspectives (SWAP),1st Italian Semantic Web Workshop*, Ancona, Italy, 10th December 2004.

Woods W (1975) What's in a Link: Foundations for Semantic Networks. In *Representation and Understanding: Studies in Cognitive Science*, Academic Press, pp. 35-82.

APPENDIX A
CRL-CATS FOR THE CARS DOMAIN

List of properties:

Property	meaning	Example
mak	Car make	mak(saloon:06, TOYOTA(country>japan):0C)
mod	model	mod(saloon:06, Corona(country>japan,make>TOYOTA):0J)
col	color	col(saloon:00.@entry, green:16)
sal	For sale	sal(saloon:06,sale:00)
Wan	wanted	wan(saloon:06.@entry, wanted:00)
Yea	year	yea(saloon:06,86:0W)
Pri	price	pri(saloon:00.@entry,3000:0Y)
Fea	feature	fea(saloon:06.@entry,good condition:18)
Cou	country	cou(saloon:06, germany:0C)
mot	Motor size	mot(saloon:00.@entry,1800:0 Z)

Allowed main domain objects(MDO)

MDO
saloon
bus
van
pick up

Property Values:

Property values should be "text" for the following relations: ***mak, col, sal, wan, fea, cou***. For ***pri, yea, mot*** they should be numeric. The property valuse of ***mod*** relation can be both text and numeric.

Semantic Information:

Semantic information	example
Country>"name of the country"	mak(saloon:06,**MERCEDES**(country>germany,country>europe):0K)
Make>"name of the make"	Corona(country>japan,make>TOYOTA):0J)

List of attributes:

attribute	Usage	Example
@less	<=	pri(saloon:06.@entry,
@more	>=	yea(saloon:06.@entry,
@downpayment	Down payemnet	

@installment	installment	
@ALF	thousand	pri(saloon:06.@entry,

These attributes are attached to the following relations: ***pri, mot, and yea***

APPENDIX B
CRL-CATS FOR REAL ESTATE DOMAIN

List of properties:

rel	meaning	
sal	For sale	sal(land:00, sale:04)
wan	wanted	wan(flat:06, wanted:00)
pur	purpose	pur(flat:00, for rent:0A)
loc	location	loc(land:00,0: قاع خنا J.@loct)
are	Area	are(land:00, 10:0T.@donem)
roo	Bed rooms	roo(flat:00, 3:0I)
cns	Consist of	cns(flat:06, kitchen:2Q)
pri	price	pri(flat:06, 1600:1E.@year.@less)
typ	Industrial, agricultural, residential	typ(flat:06, empty:0I)
flr	floor	flr(flat:00, 0:04)
fea	feature	fea(flat:00, floor from villa:1N)

Allowed main domain objects(MDO)

MDO
flat
Roof
cabana
Villa
Land
Shop
Office
Orange Farm
Building
Farm
small house

Property Values:

Secondword should be "text" for the following relations: ***sal, wan, pur, loc, type, fea, cns.*** For ***pri, roo, are*** and ***flr*** they should be numeric.

Ontological Information:

Semantic information	Usage	example
area>larger area	Area is belong to larger rea	loc(flat:06,jabal husseienن(area>mAmman):1S.@dist)

List of attributes:

attribute	Usage	example
@less	<=	pri(saloon:06.@entry,3000:1Z.@less)
@more	>=	yea(saloon:06.@entry,9:13.@more)
@downpayment		
@installment		
@ALF		pri(saloon:06.@entry,3:1Z.@ALF.@less)
@city		To denote city in loc relation
@area		To denote are in loc relation
@town		To denote town in loc relation
@dist		To denote district in loc relation
@loct		To denote location in loc relation
@donem		Area is measured by donem
@meter		Are is measured by meter
@build		That this is an building area
@space		That this is whole area
@year		Payment is due per year
@month		Payment is due per month
@permeter		The price is per meter
@perdonem		The price is per donem
@dual		Two entities comes with cns relation
@plural		More than two entities

APPENDIX C
UNL2005 SPECIFICATIONS

Releations of UNL :

Relation	Definition	Example
agt	Agent	a thing in focus which initiates an action
and	conjunction	a conjunctive relation between concepts
aoj	thing with attribute	a thing which is in a state or has an attribute
bas	basis	a thing used as the basis (standard) for expressing a degree
ben	beneficiary	an indirectly related beneficiary or victim of an event or state
cag	co-agent	a thing not in focus which initiates an implicit event which is done in parallel
cao	co-thing with attribute	a thing not in focus, as in a state in parallel
cnt	content	an equivalent concept
cob	affected co-thing	a thing which is directly effected by an implicit event done in parallel or an implicit state in parallel
con	condition	a non-focused event or state which conditions a focused event or state
coo	co-occurrence	a co-occurrent event or state for a focused event or state
dur	duration	a period of time during which an event occurs or a state exists
fmt	range	a range between two things
frm	origin	an origin of a thing
gol	goal/final state	the final state of an object or the thing finally associated with the object
ins	instrument	the instrument to carry out an event
int	intersection	A partner to take an intersection
man	manner	the way to carry out an event or characteristics of a state
met	method	the means to carry out an event
mod	modification	a thing which restricts a focused thing
nam	Name	a name of a thing
obj	affected thing	a thing in focus which is directly effected by an event or state
opl	affected place	a place in focus where an event takes effect
or	disjunction	a disjunctive relation between two concepts
per	proportion, rate of distribution	a basis or unit of proportion, rate of distribution
plc	place	the place where an event occurs, or a state is true, or a thing exists
plf	initial place	the place where an event begins or a state becomes true
plt	final place	the place where an event ends or a state becomes false
pof	part-of	a concept of which a focused thing is a part
pos	possessor	the possessor of a thing

Relation	Definition	Example
ptn	partner	an indispensable non-focused initiator of an action
pur	purpose or objective	the purpose or objective of an agent of an event or the purpose of a thing which exists
qua	quantity	quantity of a thing or unit
rsn	reason	a reason why an event or a state happens
scn	scene	a virtual world where an event occurs, or a state is true, or a thing exists
seq	sequence	a prior event or state of a focused event or state
src	source/initial state	the initial state of an object or thing initially associated with the object of an event
tim	time	the time an event occurs or a state is true
tmf	initial time	the time an event starts or a state becomes true
tmt	final time	the time an event ends or a state becomes false
to	destination	a destination of a thing
via	intermediate place or state	an intermediate place or state of an event
Icl	included	a concept of which a focused concept is a proper subset
Iof	instance of	an instance of a class
equ	Equal	an acronym of an original word

UNL attributes :

Speaker's viewpoint @ability, @ability-past, @admire, @although, @ask-back, @conclusion, @custom, @doubt, @expectation, @grant, @grant-not, @induce, @inevitability, @insistence, @intention, @may, @obligation, @obligation-not, @possibility, @probability, @regret, @request, @should, @unexpected-presumption, @unexpected-consequence, @will
Convention @angle_bracket, @double_parenthesis, @double_quotation, @parenthesis, @pl, @single_quotation, @square_bracket
Time with respect to speaker @past @present @future
Speaker's view of aspect @begin-soon @begin-just @progress @end-soon @end-just @complete @state @repeat
Speaker's view of reference @generic @def @indef @not @ordinal
Speaker's focus @emphasis @entry @qfocus @theme @title @topic

Speaker's attitude	
@affirmative @confirmation @exclamation @imperative @interrogative @invitation @politeness @respect @vocative	
Attribute for Describing Logicality	
@transitive	attached to an UW that has transitivity
@symmetric	attached to an UW that has symmetricity
@identifiable	attached to an UW that can identify the subject
@disjointed	attached to an UW or a group of UWs to show
Attribute for Times Respect to Writer	
@past	happened in the past
@present	happening at present
@future	will happen in future
Attribute for Writer's View on Aspect of Event	
@begin	beginning of an event or a state
@complete	finishing/completion of a (whole) event
@continue	continuation of an event
@custom	customary or repetitious action
@end	end/termination of an event or a state
@experience	experience
@progress	an event is in progress
@repeat	repetition of an event
@state	final state or the existence of the object on
Attribute for modifying Attribute on Aspect	
@just	expresses an event or a state that has just begun
@soon	expresses an event or a state that is about to
@yet	expresses an event or a state that has not yet
Attribute for Writer's View of Reference on Concept	
@generic	generic concept
@def	already referred
@indef	non-specific class
@not	complement set
@ordinal	ordinal number
Attribute for View of Emphasis, Focus, and Topic	
@contrast	contrasted UW
@emphasis	emphasized UW
@entry	entry or main UW of a sentence or a scope
@qfocus	focused UW of a question
@theme	instantiates an object from a different
@title	title
@topic	topic
Attribute for Writer's Attitude	
@affirmative	affirmation
@confirmation	confirmation
@exclamation	exclamation
@humility	in a humility manner

@imperative	imperative
@interrogative	interrogation
@invitation	inducement
@polite	polite way
@request	request
@respect	respectful way
@vocative	vocative
<u>Attribute for Writer's Felling and Judgment</u>	
@although	something follows against [contrary to] or
@discontented	discontented feeling of the speaker about
@expectation	expectation of something
@wish	wishful feeling, to wish something is true
@insistence	strong determination to do something
@intention	intention about something or to do
@want	desire to do something
@will	determination to do something
@need	necessity to do something
@obligation	obligation to do something according to
@obligation-not	obligation not to do something, forbid to
<u>Attribute for Writer's Felling and Judgment</u>	
@should	to do something as a matter of course
@unavoidable	unavoidable feeling of the speaker about
@certain	certainty that something is true or
@inevitable	logical inevitability that something is true
@may	practical possibility that something is true
@possible	logical possibility that something is true
@probable	(practical) probability that something is
@rare	rare logical possibility that something is
@unreal	unreality that something is true or
<u>Attribute for Writer's Felling and Judgment</u>	
@admire	admiring feeling of the speaker about something
@blame	blameful feeling of the speaker about something
@contempt	contemptuous feeling of the speaker something
@regret	Regretful feeling of the speaker about something
@surprised	surprised feeling of the speaker about something
@troublesome	troublesome feeling of the speaker about the occurrence of something
<u>Attribute for Convention</u>	
@passive	passive form
@pl	more than one
@angle_bracket	< > are used
@brace	{ } are used
@double_parenthesis	(( )) are used
@double_quote	" " are used
@parenthesis	( ) are used

@single_quote	‘ ’ are used
@square_bracket	[ ] are used

APPENDIX D
THE ENCO TRACE OF SAMPLE SENTENCE

The clever Khalid who met the strong Moh'd who read Daoud's fantastic useful book understood the lesson.

```
=================== NODE LIST ===================
/ [<<] / "فهم خالد الذكي الذي قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" /
=======================================================
lnode >>>>>
-- candidates --
[<<]{} "" (SHEAD) <.,0,0>;
rnode >>>>>
-- candidates of next word --0
[فهم]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
[ف]{} "" (FW,SEQ) <A,1,1>;
[فهم]{} "understand" (N,SAHIH,MASDAR,M,SING,NC,NDE) <A,0,5>;
[فهم]{} "understand" (V,REG,3P,PAST,TR) <A,0,5>;
[فهم]{} "understand" (N,SAHIH,MASDAR,M,SING,NC,NDE) <A,0,0>;
[فهم]{} "understand" (V,REG,3P,PAST,TR) <A,0,0>;
[فهم]{} "apprehend(obj>meaning)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "apprehend(icl>understand)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "catch(aoj>question)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "cotton(equ>understand,obj>entity)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "figure(icl>event)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "get(equ>understand,obj>matter)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "get(obj>religion)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "grasp(equ>understand)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "grip(icl>ability)" (N,MASDAR,M,NC,SAHIH) <A,0,1>;
[فهم]{} "have(goal>activity,obj>human)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "have(obj>subject)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "hold(obj>plan)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "hold(obj>word)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "intelligence(icl>wit)" (N,MASDAR,M,NC,SAHIH) <A,0,1>;
[فهم]{} "master(icl>event)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "read(equ>comprehend,agt>human,obj>mistery)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "read(equ>understand,agt>human,obj>notation)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "realization(icl>understanding)" (N,MASDAR,M,NC,SAHIH) <A,0,1>;
[فهم]{} "savvy(equ>understand(" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "savvy(equ>understand)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "see(equ>understand,agt>human)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "see(equ>understand,agt>human,icl>mental acrivity,obj>point)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V)
<A,0,1>;
[فهم]{} "see(equ>understand,agt>human,icl>mental activity,obj>point)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V)
<A,0,1>;
[فهم]{} "see(equ>understand,agt>human,icl>mental activity,obj>matters)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V)
<A,0,1>;
[فهم]{} "see(equ>understand,agt>human,icl>mental acrivity,obj>matters)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V)
<A,0,1>;
[فهم]{} "seize(agt>human,obj>idea)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "seize(agt>human,obj>point)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "seize(agt>human,obj>suggestion)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "seize(agt>human,obj>meaning)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "sink(agt>human,obj>letters)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
```

```
[فهم]{} "take(agt>human,obj>language)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "take(agt>human,obj>grammar)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "take(agt>human,obj>study)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "tumble(equ>understand)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "understanding(icl>ability)" (N,MASDAR,M,NC,SAHIH) <A,0,1>;
[فهم]{} "understanding" (N,MASDAR,M,NC,SAHIH) <A,0,1>;
[فهم]{} "uptake(equ>comprehension)" (N,MASDAR,M,NC,SAHIH) <A,0,1>;
[فهم]{} "wit(icl>talent)" (N,MASDAR,M,NC,SAHIH) <A,0,1>;
[فهم]{} "wit(icl>ability)" (N,MASDAR,M,NC,SAHIH) <A,0,1>;
[فهم]{} "understand" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[فهم]{} "understanding" (N,MASDAR,M,NC,SAHIH) <A,0,1>;
-- candidates of next word --1
[ ]{} "" (BLK) <.,0,0>;
==================== APPLIED RULE ==================
R{:::}{:::}()P1;
-- other matched rules --
======================================================
>>>>> lnode
[<<]{} "" (SHEAD) <.,0,0>;
>>>>> rnode
[فهم]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ << / [فهم|uw=uNDErstand] / [] / "خالد الذكي الذي قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" /
======================================================
lnode >>>>>
[فهم]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[ ]{} "" (BLK) <.,0,0>;
-- candidates of next word --1
[خالد]{} "Khalid" (N,NAME,SING,M,SAHIH) <A,1,1>;
[خالد]{} "endless(equ>eternal)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[خالد]{} "endless(aoj>machine)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خالد]{} "eternal(equ>god)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خالد]{} "immortal(aoj>human)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[خالد]{} "immortal(aoj>fame)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خالد]{} "immortal(aoj>book)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خالد]{} "immortal(icl>gods)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[خالد]{} "immortal(icl>state)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[خالد]{} "immortal(icl>human)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[خالد]{} "immortal(aoj>god)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خالد]{} "immortal" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[خالد]{} "imperishable(aoj>creature,icl>existence)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خالد]{} "perennial" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[خالد]{} "timeless(aoj>entity)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خالد]{} "timeless" (N,adj,M,FtSFX,SING,RGMPL) <A,0,1>;
[خالد]{} "timeless(icl>state)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[خالد]{} "undying" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[خالد]{} "undying(icl>state)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[خال]{} "beauty spot" (N,NDE,M,SING,RGFPL,SAHIH) <A,0,1>;
[خال]{} "empty(goal>place,obj>container)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خال]{} "empty" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[خال]{} "empty(equ>vacant)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خال]{} "empty(obj>content,src>container)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خال]{} "empty(obj>container)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خال]{} "empty(goal>place,obj>content,src>container)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خال]{} "open(equ>unoccupied,aoj>position)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
```

```
[خال]{} "unoccupied(aoj>house)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خال]{} "unoccupied(aoj>seat)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خال]{} "unoccupied" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خال]{} "unoccupied(aoj>entity)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خال]{} "uncle(icl>relative)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خال]{} "vacancy(icl>room)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[خال]{} "void(icl>location)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
=================== APPLIED RULE ==================
+{:::}{BLK:::}()P255;
-- other matched rules --
R{:::}{:::}()P1;
=====================================================
>>>>> lnode
[<<]{} "" (SHEAD) <.,0,0>;
>>>>> rnode
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ [<<] / [خالد / فهم ]|uw=Khalid / " الذكي الذي قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" /
=====================================================
lnode >>>>>
[<<]{} "" (SHEAD) <.,0,0>;
rnode >>>>>
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== APPLIED RULE ==================
R{:::}{:::}()P1;
-- other matched rules --
=====================================================
>>>>> lnode
[<<]{} "" (SHEAD) <.,0,0>;
>>>>> rnode
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ << / [خالد] / [ فهم|uw=Khalid] / " الذكي الذي قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" /
=====================================================
lnode >>>>>
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[خالد]{} "Khalid" (N,NAME,SING,M,SAHIH) <A,1,1>;
-- candidates of next word --1
[ ]{} "" (BLK) <.,0,0>;
=================== APPLIED RULE ==================
L{:::}{NAME,^def:-NAME,+def::}()P12;
-- other matched rules --
R{:::}{:::}()P1;
=====================================================
>>>>> lnode
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[خالد]{} "Khalid" (def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ [<<] / [خالد / فهم ]|uw=Khalid /   / "الذكي الذي قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" /
=====================================================
lnode >>>>>
[<<]{} "" (SHEAD) <.,0,0>;
rnode >>>>>
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== APPLIED RULE ==================
R{:::}{:::}()P1;
```

```
-- other matched rules --
==================================================
>>>>> lnode
[<<]{} "" (SHEAD) <.,0,0>;
>>>>> rnode
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ << / [خالد] / [ فهم|uw=Khalid] /   / "الذكي الذي قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" /
==================================================
lnode >>>>>
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[خالد]{} "Khalid" (def,N,SING,M,SAHIH) <A,1,1>;
=================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
==================================================
>>>>> lnode
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[خالد]{} "Khalid" (def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / خالد] / فهم|uw=Khalid] / [] / "الذكي الذي قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" /
==================================================
lnode >>>>>
[خالد]{} "Khalid" (def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[ ]{} "" (BLK) <.,0,0>;
-- candidates of next word --1
[ال]{} "" (FW,AL,ال) <A,8,1>;
[ا]{} "" (PREF,ا) <A,1,1>;
[ا]{} "" (SUFFEX,ا) <A,1,1>;
[ا]{THEY} "" (VSUFFEX,ا) <A,1,1>;
=================== APPLIED RULE ===================
+{:::}{BLK:::}()P255;
-- other matched rules --
L{def,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
R{:::}{:::}()P1;
==================================================
>>>>> lnode
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[خالد ]{} "Khalid" (def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / [ذكي الذي قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" / ال / [خالد ] / [ فهم" /
==================================================
lnode >>>>>
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[خالد ]{} "Khalid" (def,N,SING,M,SAHIH) <A,1,1>;
=================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
==================================================
>>>>> lnode
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[خالد ]{} "Khalid" (def,N,SING,M,SAHIH) <A,1,1>;
```

```
================== NODE LIST ==================
/ "فهم / [خالد ] / [ال] / "ذكي الذي قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس / >> /
====================================================
lnode >>>>>
[خالد ]{} "Khalid" (def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[ال]{} "" (FW,AL,ال) <A,8,1>;
-- candidates of next word --1
[ذكي]{} "clever" (N,adj,SING,M,SAHIH,FtSFX) <A,1,1>;
[ذكي]{} "agile(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "astute" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "astute(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "astute(equ>perceptive)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "brainy" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "brainy(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "bright(equ>clever,aoj>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "clever(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "clever(aoj>idea)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "clever(aoj>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "clever" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "cute(equ>smart,aoj>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "diabolic(equ>clever)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "heady(aoj>thought)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "heady(aoj>action)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "intelligent(aoj>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "intelligent" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "intelligent(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "intelligent(aoj>machine)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "keen(aoj>human,icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "knowledgeable(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "nimble(aoj>action)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "penetrative(aoj>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "quick(equ>clever,aoj>animate)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "quick(equ>acute,aoj>animate)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "savvy(aoj>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "sagacious" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "sagacious(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "sensuous(equ>keen)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "sharp(ant>blunt/dull,icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "sharp(aoj>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "sharp-witted(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "smart(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "smart(equ>intelligent,aoj>machine,icl>function)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "smart(equ>bright,aoj>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "smart(aoj>missile,icl>function)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "witty(aoj>thinking)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "witty(aoj>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "witty(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[ذكي]{} "witty(aoj>remark)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
================== APPLIED RULE ==================
R{:::}{:::}()P1;
-- other matched rules --
====================================================
>>>>> lnode
[خالد ]{} "Khalid" (def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
```

```
[ال]{} "" (FW,AL,ال) <A,8,1>;
=================== NODE LIST ===================
/ >> / [ذكي / [ال] / خالد / فهم|uw=clever] / " الذي قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" /
====================================================
lnode >>>>>
[ال]{} "" (FW,AL,ال) <A,8,1>;
rnode >>>>>
[ذكي]{} "clever" (N,adj,SING,M,SAHIH,FtSFX) <A,1,1>;
-- candidates of next word --1
[ ]{} "" (BLK) <.,0,0>;
==================== APPLIED RULE ===================
>{FW,AL:::}{adj,^defadj:-adj,+defadj::}()P11;
-- other matched rules --
R{:::}{:::}()P1;
====================================================
>>>>> lnode
[خالد ]{} "Khalid" (def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[ذكي]{} "clever" (defadj,N,SING,M,SAHIH,FtSFX) <A,1,1>;
=================== NODE LIST ===================
/ >> / الذي قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" /  / [ذكي] / [خالد ] / فهم" /
====================================================
lnode >>>>>
[خالد ]{} "Khalid" (def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[ذكي]{} "clever" (defadj,N,SING,M,SAHIH,FtSFX) <A,1,1>;
==================== APPLIED RULE ===================
<{def:+aoj:aoj:}{defadj:::}()P10;
-- other matched rules --
L{def,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
R{:::}{:::}()P1;
====================================================
>>>>> lnode
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ >> / [الذي قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" /  / [خالد ] / [فهم ] "/
====================================================
lnode >>>>>
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
==================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
====================================================
>>>>> lnode
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ >> / الذي قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" / [ ] / [ ] / [خالد ] / فهم" /
====================================================
lnode >>>>>
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[ ]{} "" (BLK) <.,0,0>;
```

```
-- candidates of next word --1
[الذي]{} "who" (RelPron,SING,M) <A,9,9>;
[ال]{} "" (FW,AL,ال) <A,8,1>;
[ا]{} "" (PREF,ا) <A,1,1>;
[ا]{} "" (SUFFEX,ا) <A,1,1>;
[ا]{THEY} "" (VSUFFEX,ا) <A,1,1>;
==================== APPLIED RULE ===================
+{:::}{BLK:::}()P255;
-- other matched rules --
L{aoj,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
L{def,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
R{:::}{:::}()P1;
=====================================================
>>>>> lnode
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / [الذي / [خالد ] / [ فهم|uw=who / " قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" /
=====================================================
lnode >>>>>
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
==================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
=====================================================
>>>>> lnode
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / [الذي] / [ خالد / فهم|uw=who] / " قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" /
=====================================================
lnode >>>>>
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[الذي]{} "who" (RelPron,SING,M) <A,9,9>;
-- candidates of next word --1
[ ]{} "" (BLK) <.,0,0>;
==================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
=====================================================
>>>>> lnode
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[الذي]{} "who" (RelPron,SING,M) <A,9,9>;
=================== NODE LIST ===================
/ << / [الذي / خالد / فهم|uw=who] / [ ] / "قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" /
=====================================================
lnode >>>>>
[الذي]{} "who" (RelPron,SING,M) <A,9,9>;
rnode >>>>>
[ ]{} "" (BLK) <.,0,0>;
-- candidates of next word --1
[قابل]{} "meet" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
```

```
[قابل]{} "match" (V,REG,3P,PAST) <A,0,5>;
[قابل]{} "match" (V,REG,3P,PAST) <A,0,0>;
[قابل]{} "accepting" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[قابل]{} "capable(aoj>entity)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قابل]{} "capable(aoj>matter)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قابل]{} "face(equ>turn,agt>human)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[قابل]{} "face(icl>event)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[قابل]{} "face(gol>human,icl>event)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[قابل]{} "face(agt>human,icl>event)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[قابل]{} "midwife(icl>helper)" (N,adj,F,SING,RGFPL,FtSFX) <A,0,1>;
[قابل]{} "midwife" (N,adj,F,SING,RGFPL,FtSFX) <A,0,1>;
[قابل]{} "midwife(icl>entity)" (N,adj,F,SING,RGFPL,FtSFX) <A,0,1>;
[قابل]{} "midwife(icl>human)" (N,adj,F,SING,RGFPL,FtSFX) <A,0,1>;
[قابل]{} "meet(obj>entity,ptn>entity)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[قابل]{} "meet(icl>event)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[قابل]{} "meet(agt>organization,ptn>human)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[قابل]{} "meet(agt>human,ptn>human)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[قابل]{} "requite(gol>kindness)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[قابل]{} "subtend" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[قابل]{} "subtend(icl>event)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
==================== APPLIED RULE ===================
+{:::}{BLK:::}()P255;
-- other matched rules --
R{:::}{:::}()P1;
======================================================
>>>>> lnode
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[الذي ]{} "who" (RelPron,SING,M) <A,9,9>;
==================== NODE LIST ====================
/ << / قابل / [الذي ] / [خالد ] / فهم|uw=meet / " محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" /
======================================================
lnode >>>>>
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[الذي ]{} "who" (RelPron,SING,M) <A,9,9>;
==================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
======================================================
>>>>> lnode
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[الذي ]{} "who" (RelPron,SING,M) <A,9,9>;
==================== NODE LIST ====================
/ << / [قابل / [الذي ] / [ خالد / فهم|uw=meet] / " محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس" /
======================================================
lnode >>>>>
[الذي ]{} "who" (RelPron,SING,M) <A,9,9>;
rnode >>>>>
[قابل]{} "meet" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
-- candidates of next word --1
[ ]{} "" (BLK) <.,0,0>;
==================== APPLIED RULE ===================
>{RelPron:::}{V:+RelClause_agt::}()P12;
-- other matched rules --
R{:::}{:::}()P1;
```

```
====================================================
>>>>> lnode
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[قابل]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ "فهم / [خالد ] / [قابل] / / "محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس / >> /
====================================================
lnode >>>>>
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[قابل]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
====================================================
>>>>> lnode
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[قابل]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ "فهم / خالد / [قابل] / [ ] / "محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس / >> /
====================================================
lnode >>>>>
[قابل]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[ ]{} "" (BLK) <.,0,0>;
-- candidates of next word --1
[محمد]{} "MOHAMMAD" (N,NAME,SING,M,SAHIH) <A,1,1>;
=================== APPLIED RULE ===================
+{:::}{BLK:::}()P255;
-- other matched rules --
R{:::}{:::}()P1;
====================================================
>>>>> lnode
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ "القوي الذي قرأ كتاب داود المفيدالرائع الدرس" / uw=MOHAMMAD|فهم / [خالد ] / [قابل ] / محمد / >> /
====================================================
lnode >>>>>
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
====================================================
>>>>> lnode
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ "القوي الذي قرأ كتاب داود المفيدالرائع الدرس" / [uw=MOHAMMAD|فهم / خالد / [قابل ] / [محمد / >> /
====================================================
lnode >>>>>
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
```

```
rnode >>>>>
[محمد]{} "MOHAMMAD" (N,NAME,SING,M,SAHIH) <A,1,1>;
-- candidates of next word --1
[ال]{} "" (FW,AL,ال) <A,8,1>;
[ا]{} "" (PREF,ا) <A,1,1>;
[ا]{} "" (SUFFEX,ا) <A,1,1>;
[ا]{THEY} "" (VSUFFEX,ا) <A,1,1>;
=================== APPLIED RULE ==================
L{:::}{NAME,^def:-NAME,+def::}()P12;
-- other matched rules --
R{:::}{:::}()P1;
==================================================
>>>>> lnode
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[محمد]{} "MOHAMMAD" (def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / محمد / [قابل ] / [خالد  ] / فهم|uw=MOHAMMAD / قوي الذي قرأ كتاب داود المفيدالرائع الدرس" / ال" /
==================================================
lnode >>>>>
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== APPLIED RULE ==================
R{:::}{:::}()P1;
-- other matched rules --
==================================================
>>>>> lnode
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ << / [محمد / [قابل ] /  خالد / فهم|uw=MOHAMMAD] / قوي الذي قرأ كتاب داود المفيدالرائع الدرس" / ال" /
==================================================
lnode >>>>>
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[محمد]{} "MOHAMMAD" (def,N,SING,M,SAHIH) <A,1,1>;
=================== APPLIED RULE ==================
R{:::}{:::}()P1;
-- other matched rules --
==================================================
>>>>> lnode
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[محمد]{} "MOHAMMAD" (def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / [محمد / قابل /  خالد / فهم|uw=MOHAMMAD] / [قوي الذي قرأ كتاب داود المفيدالرائع الدرس" / ال]" /
==================================================
lnode >>>>>
[محمد]{} "MOHAMMAD" (def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[ال]{} "" (FW,AL,ال) <A,8,1>;
-- candidates of next word --1
[قوي]{} "strong(icl>state)" (N,adj,SING,M,RGMPL,SAHIH,FtSFX) <A,1,1>;
[قو]{} "potency(equ>power)" (N,F,NDE,SING,IRGPL,SAHIH,FtSFX) <A,1,1>;
[قوي]{} "strong" (N,adj,SING,M,RGMPL,SAHIH,FtSFX) <A,0,0>;
[قوي]{} "big(aoj>wind)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
```

```
[قوي]{} "big" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "big(icl>sports)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "big(equ>eager)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "big(icl>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "brawny(icl>characteristic)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "brawny(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "brawny" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "cast-iron(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "cast-iron(aoj>stomach)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "chunk(icl>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "commanding(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "compression(obj>thought)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "demoniac(equ>violent,aoj>animate)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "devil(equ>powerful person,icl>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "forceful" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "forcible(aoj>speech,icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "forcible(aoj>reasoning,icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "forcible(aoj>expression,icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "fruity(aoj>voice)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "full-blooded(aoj>entity)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "forte(icl>sound)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "great(aoj>sound)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "hefty(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "hefty" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "high(aoj>card,icl>point)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "high-pitched(aoj>purpose)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "high-pitched(aoj>plan)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "high-spirited(equ>vigorous,aoj>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "high-spirited(equ>vigorous,aoj>behavior)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "hard(aoj>human,gol>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "hard(equ>stark,aoj>outline)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "indurate(aoj>body)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "inflexible(aoj>determination)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "jinn(icl>vehicle)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "juicy(aoj>rumor)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "iron(aoj>will,icl>attribute)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "iron(aoj>human,icl>attribute)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "keen(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "living(aoj>belief)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "lusty(aoj>voice)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "lusty(aoj>body)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "lusty" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "mighty(man>event)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "mighty(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "mighty(icl>manner)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "muscular(aoj>expression)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "muscular(equ>brawny,aoj>body)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "muscular(aoj>strength)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "muscular(aoj>movement)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "pithy(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "pithy(aoj>style)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "powerful" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "powerful(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "powerful(aoj>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "power(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
```

```
[قوي]{} "pungent(aoj>smell)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "racy(equ>vigorous,aoj>style,icl>style)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "racy(equ>vigorous,aoj>work,icl>style)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "racy(equ>vigorous,aoj>talk,icl>style)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "rugged(aoj>sound)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "rich(aoj>entity)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "rich(aoj>sound)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "rich(aoj>smell)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "rich(aoj>voice)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "robust(aoj>belief)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "robust(aoj>physique)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "robust(aoj>body)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "sforzando" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "sforzando(aoj>performance)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "sledge(equ>sledgehammer)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "sledgehammer(aoj>entity)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "sledgehammer" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "sledgehammer(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "smacking(aoj>kiss)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "smacking" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "smacking(aoj>degree)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "smacking(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "smacking(aoj>reaction)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "sinewy(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "sinewy(aoj>meat)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "sinewy" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "sinewy(aoj>entity)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "solidity(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "stark(icl>quality)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "stark(aoj>body)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "stark(icl>manner)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "stark(icl>appearance)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "stark(aoj>human)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[قوي]{} "staunch(aoj>human,icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "strongman(icl>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "strong(aoj>proof)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "strong(aoj>mind)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "strong(aoj>medicine)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "strong(aoj>lens)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "strong(aoj>human)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "strong(aoj>smell)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "strong(aoj>will)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "strong(icl>strengh)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "strong(fld>commerce,icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "sturdy(aoj>characteristics)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "sturdy(icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "turbulent(aoj>wind)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "two-handed(icl>manner)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "unrelenting(aoj>speed)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "virile(equ>forceful)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "vigorous(aoj>behavior)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "vigorous(aoj>plant)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "vigorous" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قوي]{} "youthful(aoj>activity)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[قو]{} "force" (N,NDE,SING,F,IRGPL,SAHIH,FtSFX) <A,0,5>;
```

```
[قو]{} "potency" (N,F,NDE,SING,IRGPL,SAHIH,FtSFX) <A,0,0>;
[قو]{} "agency(icl>force)" (N,MASDAR,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "balance(equ>superiority)" (N,MASDAR,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "birr" (N,MASDAR,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "clout(icl>power)" (N,MASDAR,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "faculty(icl>function)" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "heft(icl>power)" (N,MASDAR,F,NC,FtSFX) <A,0,1>;
[قو]{} "heartiness" (N,MASDAR,F,NC,FtSFX) <A,0,1>;
[قو]{} "heartiness(equ>vigor)" (N,MASDAR,F,NC,IRGPL,FtSFX) <A,0,1>;
[قو]{} "horn(icl>automobile)" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "intensity" (N,MASDAR,F,NC,FtSFX) <A,0,1>;
[قو]{} "intensity(icl>condition)" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "intension(icl>degree)" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "intension" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "intensity(icl>emotion)" (N,MASDAR,F,NC,FtSFX) <A,0,1>;
[قو]{} "leverage(obj>entity)" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "might(icl>power)" (N,MASDAR,F,NC,FtSFX) <A,0,1>;
[قو]{} "might" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "nerve(icl>power)" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "power(equ>force)" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "power(icl>strength)" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "power(icl>entity)" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "power(icl>strenght)" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "power(icl>ratio)" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "power(equ>strength)" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "power(icl>force)" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "power(icl>energy)" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "potentiality(icl>power)" (N,MASDAR,F,NC,FtSFX) <A,0,1>;
[قو]{} "puissance" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[قو]{} "sinew(icl>strength)" (N,MASDAR,F,NC,RGFPL,FtSFX) <A,0,1>;
[قو]{} "solidity(icl>situation)" (N,MASDAR,F,NC,FtSFX) <A,0,1>;
[قو]{} "solidity(icl>character)" (N,MASDAR,F,NC,FtSFX) <A,0,1>;
[قو]{} "strength" (N,MASDAR,F,NC,IRGPL,FtSFX) <A,0,1>;
[قو]{} "strength(icl>support)" (N,MASDAR,F,NC,IRGPL,FtSFX) <A,0,1>;
[قو]{} "strength(icl>power)" (N,MASDAR,F,NC,IRGPL,FtSFX) <A,0,1>;
[قو]{} "strength(icl>mental toughness)" (N,MASDAR,F,NC,IRGPL,FtSFX) <A,0,1>;
[قو]{} "strength(icl>concentration)" (N,MASDAR,F,NC,IRGPL,FtSFX) <A,0,1>;
[قو]{} "strength(icl>advantage)" (N,MASDAR,F,NC,IRGPL,FtSFX) <A,0,1>;
[قو]{} "strength(icl>manpower)" (N,MASDAR,F,NC,IRGPL,FtSFX) <A,0,1>;
[قو]{} "strength(icl>resistance)" (N,MASDAR,F,NC,IRGPL,FtSFX) <A,0,1>;
[قو]{} "strength(icl>intensity)" (N,MASDAR,F,NC,IRGPL,FtSFX) <A,0,1>;
[قو]{} "strength(goal>activity,icl>power)" (N,MASDAR,F,NC,IRGPL,FtSFX) <A,0,1>;
[قو]{} "strong-arm" (N,MASDAR,F,NC,RGFPL,FtSFX) <A,0,1>;
[قو]{} "tonicity(icl>property)" (N,MASDAR,F,NC,RGFPL,FtSFX) <A,0,1>;
[قو]{} "vigor(icl>physical energy)" (N,MASDAR,F,NC,FtSFX) <A,0,1>;
[قو]{} "vigor(icl>ability)" (N,MASDAR,F,NC,FtSFX) <A,0,1>;
[قو]{} "vigor(icl>quality)" (N,MASDAR,F,NC,FtSFX) <A,0,1>;
[قو]{} "virility(icl>power)" (N,MASDAR,F,NC,FtSFX) <A,0,1>;
[قو]{} "forces" (N,NDE,F,PLUR,MAQ2) <A,0,1>;
[قو]{} "strength(icl>manpower)" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
=================== APPLIED RULE ==================
R{:::}{:::}()P1;
-- other matched rules --
====================================================
>>>>> lnode
[محمد]{} "MOHAMMAD" (def,N,SING,M,SAHIH) <A,1,1>;
```

```
>>>>> rnode
[ال]{} "" (FW,AL,ال) <A,8,1>;
=================== NODE LIST ===================
/ << / محمد / قابل / خالد / فهم|uw=MOHAMMAD / [قوي] / [ال|uw=strong(icl>state)] / " الذي قرأ كتاب داود المفيدالرائع
الدرس" /
================================================
lnode >>>>>
[ال]{} "" (FW,AL,ال) <A,8,1>;
rnode >>>>>
[قوي]{} "strong(icl>state)" (N,adj,SING,M,RGMPL,SAHIH,FtSFX) <A,1,1>;
-- candidates of next word --1
[ ]{} "" (BLK) <.,0,0>;
=================== APPLIED RULE ==================
>{FW,AL:::}{adj,^defadj:-adj,+defadj::}()P11;
-- other matched rules --
R{:::}{:::}()P1;
================================================
>>>>> lnode
[محمد]{} "MOHAMMAD" (def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[قوي]{} "strong(icl>state)" (defadj,N,SING,M,RGMPL,SAHIH,FtSFX) <A,1,1>;
=================== NODE LIST ===================
/ << / محمد] / قابل / خالد / فهم|uw=MOHAMMAD] / [الذي قرأ كتاب داود المفيدالرائع الدرس" / / [قوي" /
================================================
lnode >>>>>
[محمد]{} "MOHAMMAD" (def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[قوي]{} "strong(icl>state)" (defadj,N,SING,M,RGMPL,SAHIH,FtSFX) <A,1,1>;
=================== APPLIED RULE ==================
<{def:+aoj:aoj:}{defadj:::}()P10;
-- other matched rules --
L{def,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
R{:::}{:::}()P1;
================================================
>>>>> lnode
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[محمد]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / الذي قرأ كتاب داود المفيدالرائع الدرس" / [محمد] / [قابل ] / خالد / فهم" /
================================================
lnode >>>>>
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[محمد]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
=================== APPLIED RULE ==================
R{:::}{:::}()P1;
-- other matched rules --
================================================
>>>>> lnode
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[محمد]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / الذي قرأ كتاب داود المفيدالرائع الدرس" / [ ] / [محمد] / قابل / خالد / فهم" /
================================================
lnode >>>>>
[محمد]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
```

```
rnode >>>>>
[ ]{} "" (BLK) <.,0,0>;
-- candidates of next word --1
[الذي]{} "who" (RelPron,SING,M) <A,9,9>;
[ال]{} "" (FW,AL,ال) <A,8,1>;
[ا]{} "" (PREF,ا) <A,1,1>;
[ا]{} "" (SUFFEX,ا) <A,1,1>;
[ا]{THEY} "" (VSUFFEX,ا) <A,1,1>;
==================== APPLIED RULE ===================
+{:::}{BLK:::}()P255;
-- other matched rules --
L{aoj,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
L{def,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
R{:::}{:::}()P1;
=====================================================
>>>>> lnode
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / الذي / [محمد ] / [قابل ] /  خالد / فهم|uw=who / " قرأ كتاب داود المفيدالرائع الدرس" /
=====================================================
lnode >>>>>
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
==================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
=====================================================
>>>>> lnode
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / الذي] / [محمد ] / قابل /  خالد / فهم|uw=who] / " قرأ كتاب داود المفيدالرائع الدرس" /
=====================================================
lnode >>>>>
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[الذي]{} "who" (RelPron,SING,M) <A,9,9>;
-- candidates of next word --1
[ ]{} "" (BLK) <.,0,0>;
==================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
=====================================================
>>>>> lnode
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[الذي]{} "who" (RelPron,SING,M) <A,9,9>;
=================== NODE LIST ===================
/ << / الذي] /  محمد / قابل /  خالد / فهم|uw=who] / [ ] / "قرأ كتاب داود المفيدالرائع الدرس" /
=====================================================
lnode >>>>>
[الذي]{} "who" (RelPron,SING,M) <A,9,9>;
rnode >>>>>
[ ]{} "" (BLK) <.,0,0>;
```

```
-- candidates of next word --1
[قرأ]{} "read" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
[قرأ]{} "run(equ>say,obj>writing)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
==================== APPLIED RULE ===================
+{:::}{BLK:::}()P255;
-- other matched rules --
R{:::}{:::}()P1;
=========================================================
>>>>> lnode
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[الذي ]{} "who" (RelPron,SING,M) <A,9,9>;
==================== NODE LIST ====================
/ << / قرأ / [ الذي] / [ محمد] / قابل / خالد / فهم|uw=read / " كتاب داود المفيدالرائع الدرس" /
=========================================================
lnode >>>>>
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[الذي ]{} "who" (RelPron,SING,M) <A,9,9>;
==================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
=========================================================
>>>>> lnode
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[الذي ]{} "who" (RelPron,SING,M) <A,9,9>;
==================== NODE LIST ====================
/ << / [قرأ] / [ الذي] / محمد / قابل / خالد / فهم|uw=read] / " كتاب داود المفيدالرائع الدرس" /
=========================================================
lnode >>>>>
[الذي ]{} "who" (RelPron,SING,M) <A,9,9>;
rnode >>>>>
[قرأ]{} "read" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
-- candidates of next word --1
[ ]{} "" (BLK) <.,0,0>;
==================== APPLIED RULE ===================
>{RelPron:::}{V:+RelClause_agt::}()P12;
-- other matched rules --
R{:::}{:::}()P1;
=========================================================
>>>>> lnode
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
==================== NODE LIST ====================
/ << / كتاب داود المفيدالرائع الدرس" / / [قرأ] / [ محمد] / قابل / خالد / فهم" /
=========================================================
lnode >>>>>
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
==================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
=========================================================
>>>>> lnode
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
```

```
>>>>> rnode
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ >> / كتاب داود المفيدالرائع الدرس" / [ ] / [قرأ] / محمد / قابل / خالد / فهم" /
===================================================
lnode >>>>>
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[ ]{} "" (BLK) <.,0,0>;
-- candidates of next word --1
[كتاب]{} "BOOK" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
[ك]{} "you" (MODREL,ك) <A,1,1>;
[ك]{} "YOU" (MODREL,ك) <A,1,1>;
[كتاب]{} "book" (N,NDE,M,SING,IRGPL,SAHIH) <A,0,1>;
[كتاب]{} "book(icl>writing)" (N,MASDAR,F,SING,RGFPL,FtSFX) <A,0,1>;
[كتاب]{} "book(icl>lylic)" (N,NDE,M,SING,IRGPL,SAHIH) <A,0,1>;
[كتاب]{} "handwriting" (N,MASDAR,F,SING,RGFPL,FtSFX) <A,0,1>;
[كتاب]{} "handwriting(equ>script)" (N,MASDAR,F,SING,RGFPL,FtSFX) <A,0,1>;
[كتاب]{} "letter(icl>writing)" (N,MASDAR,F,SING,RGFPL,FtSFX) <A,0,1>;
[كتاب]{} "letter(gol>writing)" (N,MASDAR,F,SING,RGFPL,FtSFX) <A,0,1>;
[كتاب]{} "lettering(icl>writing)" (N,MASDAR,F,SING,RGFPL,FtSFX) <A,0,1>;
[كتاب]{} "lettering(icl>style)" (N,MASDAR,F,SING,RGFPL,FtSFX) <A,0,1>;
[كتاب]{} "lettering" (N,MASDAR,F,SING,RGFPL,FtSFX) <A,0,1>;
[كتاب]{} "script" (N,MASDAR,F,SING,RGFPL,FtSFX) <A,0,1>;
[كتاب]{} "script(icl>writing)" (N,MASDAR,F,SING,RGFPL,FtSFX) <A,0,1>;
[كتاب]{} "script(icl>font)" (N,MASDAR,F,SING,RGFPL,FtSFX) <A,0,1>;
[كتاب]{} "write" (N,MASDAR,F,SING,RGFPL,FtSFX) <A,0,1>;
[كتاب]{} "writ" (N,MASDAR,F,SING,RGFPL,FtSFX) <A,0,1>;
[كتاب]{} "writing(icl>practice)" (N,MASDAR,F,SING,RGFPL,FtSFX) <A,0,1>;
[كتاب]{} "writing" (N,MASDAR,F,SING,RGFPL,FtSFX) <A,0,1>;
=================== APPLIED RULE ==================
+{:::}{BLK:::}()P255;
-- other matched rules --
R{:::}{:::}()P1;
===================================================
>>>>> lnode
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[قرأ ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ >> / كتاب / [قرأ ] / [محمد ] / قابل / خالد / فهم|uw=BOOK / " داود المفيدالرائع الدرس" /
===================================================
lnode >>>>>
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[قرأ ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== APPLIED RULE ==================
R{:::}{:::}()P1;
-- other matched rules --
===================================================
>>>>> lnode
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[قرأ ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ >> / [كتاب] / [قرأ ] / محمد / قابل / خالد / فهم|uw=BOOK] / " داود المفيدالرائع الدرس" /
===================================================
lnode >>>>>
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[كتاب]{} "BOOK" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
-- candidates of next word --1
[ ]{} "" (BLK) <.,0,0>;
==================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
=====================================================
>>>>> lnode
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[كتاب]{} "BOOK" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== NODE LIST ====================
/ << / كتاب] / قرأ / محمد / قابل / خالد / فهم|uw=BOOK] / [ ] / "داود المفيدالرائع الدرس" /
=====================================================
lnode >>>>>
[كتاب]{} "BOOK" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
rnode >>>>>
[ ]{} "" (BLK) <.,0,0>;
-- candidates of next word --1
[داود]{} "DAOUD" (N,NAME,SING,M,SAHIH) <A,1,1>;
[دا]{} "disease(icl>condition)" (N,NDE,M,NC,RGFPL,MAMD) <A,0,1>;
==================== APPLIED RULE ===================
+{:::}{BLK:::}()P255;
-- other matched rules --
L{NDE,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
R{:::}{:::}()P1;
=====================================================
>>>>> lnode
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[كتاب ]{} "BOOK" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== NODE LIST ====================
/ << / داود / [ كتاب] / [ قرأ] / محمد / قابل / خالد / فهم|uw=DAOUD / " المفيدالرائع الدرس" /
=====================================================
lnode >>>>>
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[كتاب ]{} "BOOK" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
==================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
=====================================================
>>>>> lnode
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[كتاب ]{} "BOOK" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== NODE LIST ====================
/ << / داود] / [ كتاب] / قرأ / محمد / قابل / خالد / فهم|uw=DAOUD] / " المفيدالرائع الدرس" /
=====================================================
lnode >>>>>
[كتاب ]{} "BOOK" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
rnode >>>>>
[داود]{} "DAOUD" (N,NAME,SING,M,SAHIH) <A,1,1>;
-- candidates of next word --1
[ ]{} "" (BLK) <.,0,0>;
```

```
=================== APPLIED RULE ==================
L{:::}{NAME,^def:-NAME,+def::}()P12;
-- other matched rules --
L{NDE,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
R{:::}{:::}()P1;
====================================================
>>>>> lnode
[كتاب ]{} "BOOK" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
>>>>> rnode
[داود]{} "DAOUD" (def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / داود / [ كتاب] / [قرأ ] / محمد / قابل / خالد / فهم|uw=DAOUD / / "المفيدالرائع الدرس" /
====================================================
lnode >>>>>
[قرأ ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[كتاب ]{} "BOOK" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== APPLIED RULE ==================
R{:::}{:::}()P1;
-- other matched rules --
====================================================
>>>>> lnode
[قرأ ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[كتاب ]{} "BOOK" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== NODE LIST ===================
/ << / داود] / [ كتاب] / قرأ / محمد / قابل / خالد / فهم|uw=DAOUD] / / "المفيدالرائع الدرس" /
====================================================
lnode >>>>>
[كتاب ]{} "BOOK" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
rnode >>>>>
[داود]{} "DAOUD" (def,N,SING,M,SAHIH) <A,1,1>;
=================== APPLIED RULE ==================
<{NDE,^mod:+mod::}{def::mod:}()P11;
-- other matched rules --
L{NDE,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
R{:::}{:::}()P1;
====================================================
>>>>> lnode
[قرأ ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[كتاب ]{} "BOOK" (mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== NODE LIST ===================
/ << / المفيدالرائع الدرس" / / [ كتاب] / [قرأ ] / محمد / قابل / خالد / فهم" /
====================================================
lnode >>>>>
[قرأ ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[كتاب ]{} "BOOK" (mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== APPLIED RULE ==================
R{:::}{:::}()P1;
-- other matched rules --
====================================================
>>>>> lnode
[قرأ ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[كتاب ]{} "BOOK" (mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== NODE LIST ===================
/ >> / المفيدالرائع الدرس" / [ ] / [كتاب ] / قرأ / محمد / قابل / خالد / فهم" /
========================================================
lnode >>>>>
[كتاب ]{} "BOOK" (mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
rnode >>>>>
[ ]{} "" (BLK) <.,0,0>;
-- candidates of next word --1
[ال]{} "" (FW,AL,ال) <A,8,1>;
[ا]{} "" (PREF,ا) <A,1,1>;
[ا]{} "" (SUFFEX,ا) <A,1,1>;
[ا]{THEY} "" (VSUFFEX,ا) <A,1,1>;
=================== APPLIED RULE ===================
+{:::}{BLK:::}()P255;
-- other matched rules --
L{mod,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
L{NDE,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
R{:::}{:::}()P1;
========================================================
>>>>> lnode
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[كتاب ]{} "BOOK" (mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== NODE LIST ===================
/ >> / مفيدالرائع الدرس" / ال / [كتاب ] / [قرأ ] / محمد / قابل / خالد / فهم" /
========================================================
lnode >>>>>
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[كتاب ]{} "BOOK" (mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
========================================================
>>>>> lnode
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[كتاب ]{} "BOOK" (mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== NODE LIST ===================
/ >> / مفيدالرائع الدرس" / [ال] / [كتاب ] / قرأ / محمد / قابل / خالد / فهم" /
========================================================
lnode >>>>>
[كتاب ]{} "BOOK" (mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
rnode >>>>>
[ال]{} "" (FW,AL,ال) <A,8,1>;
-- candidates of next word --1
[مفيد]{} "useful" (N,M,adj,INDEF,SING,RGMPL,SAHIH,FtSFX) <A,1,1>;
[مفيد]{} "advantageous(aoj>entity)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[مفيد]{} "advantageous(icl>state)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[مفيد]{} "advantageous" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[مفيد]{} "beneficial" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[مفيد]{} "beneficent" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[مفيد]{} "beneficial(icl>state)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[مفيد]{} "helpful(aoj>entity)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[مفيد]{} "instrumental(icl>state)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[مفيد]{} "instrumental(aoj>activity)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[مفيد]{} "instrumental" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[مفيد]{} "profitable(icl>state)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
```

```
[مفيد]{} "purposive(icl>state)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[مفيد]{} "purposive" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[مفيد]{} "salutary(aoj>lesson,aoj>penalty)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[مفيد]{} "salutary(aoj>advice,aoj>lesson)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[مفيد]{} "salutary(aoj>medicine,icl>property)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[مفيد]{} "serviceable" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[مفيد]{} "serviceable(icl>state)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[مفيد]{} "serviceable(equ>useful,icl>efficiency)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[مفيد]{} "useful(icl>state)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[مفيد]{} "useful(equ>helpful)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[مفيد]{} "wholesome(aoj>movie,icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[مفيد]{} "wholesome(aoj>reading,icl>state)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
==================== APPLIED RULE ==================
R{:::}{:::}()P1;
-- other matched rules --
====================================================
>>>>> lnode
[كتاب ]{} "BOOK" (mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
>>>>> rnode
[ال]{} "" (FW,AL,ال) <A,8,1>;
=================== NODE LIST ===================
/ >> / [مفيد / [ال] /  كتاب / قرأ / محمد / قابل /  خالد / فهم|uw=useful] / "الرائع الدرس" /
====================================================
lnode >>>>>
[ال]{} "" (FW,AL,ال) <A,8,1>;
rnode >>>>>
[مفيد]{} "useful" (N,M,adj,INDEF,SING,RGMPL,SAHIH,FtSFX) <A,1,1>;
-- candidates of next word --1
[ال]{} "" (FW,AL,ال) <A,8,1>;
[ا]{} "" (PREF,ا) <A,1,1>;
[ا]{} "" (SUFFEX,ا) <A,1,1>;
[ا]{THEY} "" (VSUFFEX,ا) <A,1,1>;
==================== APPLIED RULE ==================
>{FW,AL:::}{adj,^defadj:-adj,+defadj::}()P11;
-- other matched rules --
R{:::}{:::}()P1;
====================================================
>>>>> lnode
[كتاب ]{} "BOOK" (mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
>>>>> rnode
[مفيد]{} "useful" (defadj,N,M,INDEF,SING,RGMPL,SAHIH,FtSFX) <A,1,1>;
=================== NODE LIST ===================
/ >> / "رائع الدرس / ال / [مفيد] / [كتاب ] / قرأ / محمد / قابل /  خالد / فهم" /
====================================================
lnode >>>>>
[كتاب ]{} "BOOK" (mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
rnode >>>>>
[مفيد]{} "useful" (defadj,N,M,INDEF,SING,RGMPL,SAHIH,FtSFX) <A,1,1>;
==================== APPLIED RULE ==================
<{mod:+aoj:aoj:}{defadj:::}()P10;
-- other matched rules --
L{mod,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
L{NDE,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
R{:::}{:::}()P1;
====================================================
>>>>> lnode
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
```

```
>>>>> rnode
[كتاب ] {} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== NODE LIST ===================
/ "فهم / خالد / قابل / محمد / [قرأ ] / [كتاب ] / ال / "رائع الدرس / >> /
==========================================================
lnode >>>>>
[قرأ] {} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[كتاب ] {} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
==================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
==========================================================
>>>>> lnode
[قرأ] {} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[كتاب ] {} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== NODE LIST ===================
/ "فهم / خالد / قابل / محمد / قرأ / [كتاب ] / [ال] / "رائع الدرس / >> /
==========================================================
lnode >>>>>
[كتاب ] {} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
rnode >>>>>
[ال] {} "" (FW,AL,ال) <A,8,1>;
-- candidates of next word --1
[رائع] {} "fantastic" (N,adj,SING,M,RGMPL,SAHIH,FtSFX) <A,1,1>;
[رائع] {} "superb" (N,adj,SING,M,RGMPL,SAHIH,FtSFX) <A,0,5>;
[رائع] {} "superb(icl>#state)" (N,adj,SING,M,RGMPL,SAHIH,FtSFX) <A,0,5>;
[رائع] {} "superb" (N,adj,SING,M,RGMPL,SAHIH,FtSFX) <A,0,0>;
[رائع] {} "superb(icl>#state)" (N,adj,SING,M,RGMPL,SAHIH,FtSFX) <A,0,0>;
[رائع] {} "adorable(icl>property)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع] {} "adorable(icl>state)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "adorable" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "awesome(equ>very impressive)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "arresting(icl>activity)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع] {} "arresting" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "brave(equ>gorgeous)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "brilliant(equ>fine,aoj>matter,icl>quality)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع] {} "brilliant(icl>type)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "champion(icl>manner)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "cracking(equ>wonderful,aoj>entity)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع] {} "cracking(equ>wonderful,aoj>art)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع] {} "copacetic(equ>wonderful)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "corking" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "corking(icl>manner)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "dazzling(aoj>human)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "effective" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "elegant(equ>wonderful)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "elegant(aoj>literature)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع] {} "ducky(icl>human)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "ducky(aoj>entity)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع] {} "extraordinary(equ>peculiar)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "felicitous(aoj>human,icl>state)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع] {} "famous(equ>wonderful,aoj>entity)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع] {} "gorgeous(aoj>entity)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع] {} "gorgeous" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
```

```
[رائع]{} "grand(aoj>landscape)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "grand(aoj>appearance)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "great(equ>wonderful)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "great(icl>manner)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "hunky-dory" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "hunky-dory(icl>state)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "incredible(equ>extraordinary)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "jim-dandy(icl>abstraction)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "jim-dandy(aoj>human)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "jolly" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "keen(aoj>remark)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "keen(equ>wonderful)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "magnificence(equ>grandeur,icl>quality)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "magnificence(equ>superb,icl>quality)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "marvelous" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "lulu" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "masterwork" (N,NDE,F,SING,IRGPL,FtSFX) <A,0,1>;
[رائع]{} "neat(aoj>habit)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "neat(aoj>style)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "neat(equ>wonderful)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "nifty" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "nifty(aoj>word)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "outstanding(gol>talent)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "outstanding(aoj>performance)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "peachy(equ>nice)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "picturesque(aoj>style)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "picturesque(aoj>word)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "rattling(aoj>story)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "rattling(aoj>door)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "rattling(aoj>instrument)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "rattling(aoj>action)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "rattling(icl>manner)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "remarkably(equ>extraordinary,icl>degree)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "remarkable" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "royal(equ>wonderful)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "royalty(equ>magnificence,icl>appearance)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "rip-roaring(equ>wonderful)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "ripping" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "scrumptious" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "showy" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "showy(icl>state)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "showy(equ>striking)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "showy(icl>attitude)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "smashing" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "spanking(icl>manner)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "spanking" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "spanking(man>event)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "some(equ>wonderful,icl>quality)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "some(aoj>human)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "some(equ>remarkable)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "superb" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "super(icl>state)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "super(icl>manner)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "super" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "superfine(aoj>object)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
```

```
[رائع]{} "superfine(aoj>goods)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "superfine(aoj>product)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "superfine(aoj>human)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "terrific(equ>wonderful,icl>quality)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "theatrical(equ>showy,aoj>manner)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "theatrical(equ>showy,aoj>gesture)" (N,adj,M,FtSFX,SING,IRGPL,SAHIH) <A,0,1>;
[رائع]{} "tough(equ>wonderful)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "tremendous(equ>wonderful)" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "wonderful" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
[رائع]{} "wondrous" (N,adj,M,FtSFX,SING,RGMPL,SAHIH) <A,0,1>;
==================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
=====================================================
>>>>> lnode
[كتاب ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
>>>>> rnode
[ال]{} "" (FW,AL,ال) <A,8,1>;
==================== NODE LIST ===================
/ " الدرس" / [فهم|uw=fantastic] / خالد / قابل / محمد / قرأ / كتاب / [ال] / [رائع] / >> /
=====================================================
lnode >>>>>
[ال]{} "" (FW,AL,ال) <A,8,1>;
rnode >>>>>
[رائع]{} "fantastic" (N,adj,SING,M,RGMPL,SAHIH,FtSFX) <A,1,1>;
-- candidates of next word --1
[]{} "" (BLK) <.,0,0>;
==================== APPLIED RULE ===================
>{FW,AL:::}{adj,^defadj:-adj,+defadj::}()P11;
-- other matched rules --
R{:::}{:::}()P1;
=====================================================
>>>>> lnode
[كتاب ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
>>>>> rnode
[رائع]{} "fantastic" (defadj,N,SING,M,RGMPL,SAHIH,FtSFX) <A,1,1>;
==================== NODE LIST ===================
/ "فهم / خالد / قابل / محمد / قرأ / [كتاب ] / [رائع] / / "الدرس / >> /
=====================================================
lnode >>>>>
[كتاب ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
rnode >>>>>
[رائع]{} "fantastic" (defadj,N,SING,M,RGMPL,SAHIH,FtSFX) <A,1,1>;
==================== APPLIED RULE ===================
<{mod:+aoj:aoj:}{defadj:::}()P10;
-- other matched rules --
L{aoj,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
L{mod,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
L{NDE,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
R{:::}{:::}()P1;
=====================================================
>>>>> lnode
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[كتاب ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
==================== NODE LIST ===================
/ "فهم / خالد / قابل / محمد / [قرأ ] / [كتاب ] / / "الدرس / >> /
=====================================================
lnode >>>>>
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[كتاب  ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== APPLIED RULE ==================
R{:::}{:::}()P1;
-- other matched rules --
==================================================
>>>>> lnode
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[كتاب  ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== NODE LIST ===================
/ "فهم / خالد  / قابل / محمد / قرأ / [كتاب  ] / [ ] / الدرس" / >> /
==================================================
lnode >>>>>
[كتاب  ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
rnode >>>>>
[ ]{} "" (BLK) <.,0,0>;
-- candidates of next word --1
[ال]{} "" (FW,AL,ال) <A,8,1>;
[ا]{} "" (PREF,ا) <A,1,1>;
[ا]{} "" (SUFFEX,ا) <A,1,1>;
[ا]{THEY} "" (VSUFFEX,ا) <A,1,1>;
=================== APPLIED RULE ==================
+{:::}{BLK:::}()P255;
-- other matched rules --
L{aoj,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
L{mod,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
L{NDE,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
R{:::}{:::}()P1;
==================================================
>>>>> lnode
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[كتاب   ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== NODE LIST ===================
/ "فهم / خالد  / قابل / محمد / [قرأ ] / [كتاب   ] / ال / درس" / >> /
==================================================
lnode >>>>>
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[كتاب   ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== APPLIED RULE ==================
R{:::}{:::}()P1;
-- other matched rules --
==================================================
>>>>> lnode
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[كتاب   ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
=================== NODE LIST ===================
/ "فهم / خالد  / قابل / محمد / قرأ / [كتاب   ] / [ال] / درس" / >> /
==================================================
lnode >>>>>
[كتاب   ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
rnode >>>>>
[ال]{} "" (FW,AL,ال) <A,8,1>;
```

```
-- candidates of next word --1
[درس]{} "lesson" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,1,1>;
[درس]{} "con(icl>manner)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "cover(equ>study)" (N,NDE,M,SING,IRGPL,SAHIH) <A,0,1>;
[درس]{} "deliberate(obj>entity)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "elaborate(equ>polish,agt>human,obj>plan)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "elaborate(agt>human,obj>entity)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "excogitate(icl>event)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "instruction" (N,NDE,M,SING,IRGPL,SAHIH) <A,0,1>;
[درس]{} "learn(agt>human,obj>thought)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "learn" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "lesson" (N,NDE,M,SING,IRGPL,SAHIH) <A,0,1>;
[درس]{} "lesson(pos>textbook)" (N,NDE,M,SING,IRGPL,SAHIH) <A,0,1>;
[درس]{} "peruse" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "read(equ>study,agt>human,obj>science)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "read(obj>rule)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "read(equ>study,agt>human,obj>cource)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "read(equ>study,agt>human)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "sit(agt>organization,obj>report)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "sit(agt>human,obj>report)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "study(fld>music)" (N,NDE,M,SING,IRGPL,SAHIH) <A,0,1>;
[درس]{} "study" (N,NDE,M,SING,IRGPL,SAHIH) <A,0,1>;
[درس]{} "thresh(agt>human,obj>grain)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "traverse(agt>human,obj>opinion)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "traverse(agt>organization,obj>plan)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "traverse(agt>human,obj>plan)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
[درس]{} "traverse(agt>organization,obj>opinion)" (V,REG,TR,3P,PAST,ACTIVE,CONF_V) <A,0,1>;
=================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
====================================================
>>>>> lnode
[كتاب  ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
>>>>> rnode
[ال]{} "" (FW,AL,ال) <A,8,1>;
=================== NODE LIST ===================
/ << / درس / [درس] / [ال] /  كتاب / [كتاب / قرأ / محمد / قابل /  خالد / فهم|uw=lesson] /
====================================================
lnode >>>>>
[ال]{} "" (FW,AL,ال) <A,8,1>;
rnode >>>>>
[درس]{} "lesson" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,1,1>;
=================== APPLIED RULE ===================
>{FW,AL:::}{NDE,^def:-NDE,+def::}()P12;
-- other matched rules --
R{:::}{:::}()P1;
====================================================
>>>>> lnode
[كتاب  ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
>>>>> rnode
[درس]{} "lesson" (def,N,M,IRGPL,INDEF,SING,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / درس / [درس] / [  كتاب] / قرأ / محمد / قابل /  خالد / فهم] /
====================================================
lnode >>>>>
[كتاب  ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
rnode >>>>>
```

```
[درس]{} "lesson" (def,N,M,IRGPL,INDEF,SING,SAHIH) <A,1,1>;
==================== APPLIED RULE ===================
L{aoj,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
-- other matched rules --
L{mod,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
L{NDE,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
R{:::}{:::}()P1;
=====================================================
>>>>> lnode
[كتاب  ]{} "BOOK" (ncnp,aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
>>>>> rnode
[درس]{} "lesson" (def,N,M,IRGPL,INDEF,SING,SAHIH) <A,1,1>;
==================== NODE LIST ====================
/ << / درس / [   كتاب] / [ قرأ] / محمد / قابل /  خالد / فهم /
=====================================================
lnode >>>>>
[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[كتاب  ]{} "BOOK" (ncnp,aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
==================== APPLIED RULE ===================
<{V,RelClause_agt,^RelClause_obj:+RelClause_obj::}{ncnp::obj:}()P11;
-- other matched rules --
<{V,^agt:+agt::}{ncnp::agt:}()P8;
R{:::}{:::}()P1;
=====================================================
>>>>> lnode
[محمد]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[قرأ]{} "read" (RelClause_obj,RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
==================== NODE LIST ====================
/ << / درس / [ قرأ] / [ محمد] / قابل /  خالد / فهم /
=====================================================
lnode >>>>>
[محمد]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[قرأ]{} "read" (RelClause_obj,RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
==================== APPLIED RULE ===================
<{def:+aoj:aoj:}{RelClause_obj:::}()P10;
-- other matched rules --
R{:::}{:::}()P1;
=====================================================
>>>>> lnode
[قابل]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[محمد]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
==================== NODE LIST ====================
/ << / درس / [ محمد] / [ قابل] /  خالد / فهم /
=====================================================
lnode >>>>>
[قابل]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[محمد]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
==================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
=====================================================
>>>>> lnode
[قابل]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
```

```
>>>>> rnode
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / درس] / [ محمد] / قابل / خالد / فهم] /
=====================================================
lnode >>>>>
[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[درس]{} "lesson" (def,N,M,IRGPL,INDEF,SING,SAHIH) <A,1,1>;
==================== APPLIED RULE ===================
L{aoj,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
-- other matched rules --
L{def,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
R{:::}{:::}()P1;
=====================================================
>>>>> lnode
[محمد ]{} "MOHAMMAD" (ncnp,aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[درس]{} "lesson" (def,N,M,IRGPL,INDEF,SING,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / درس [ محمد] / [ قابل] / خالد / فهم /
=====================================================
lnode >>>>>
[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[محمد ]{} "MOHAMMAD" (ncnp,aoj,def,N,SING,M,SAHIH) <A,1,1>;
==================== APPLIED RULE ===================
<{V,RelClause_agt,^RelClause_obj:+RelClause_obj::}{ncnp::obj:}()P11;
-- other matched rules --
<{V,^agt:+agt::}{ncnp::agt:}()P8;
R{:::}{:::}()P1;
=====================================================
>>>>> lnode
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[قابل ]{} "meet" (RelClause_obj,RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ << / درس [ قابل] / [ خالد] / فهم /
=====================================================
lnode >>>>>
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[قابل ]{} "meet" (RelClause_obj,RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
==================== APPLIED RULE ===================
<{def:+aoj:aoj:}{RelClause_obj:::}()P10;
-- other matched rules --
R{:::}{:::}()P1;
=====================================================
>>>>> lnode
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / [درس / [ خالد] / [ فهم /
=====================================================
lnode >>>>>
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
```

```
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
=================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
==================================================
>>>>> lnode
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / درس] / [ خالد] / فهم] /
==================================================
lnode >>>>>
[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
rnode >>>>>
[درس]{} "lesson" (def,N,M,IRGPL,INDEF,SING,SAHIH) <A,1,1>;
=================== APPLIED RULE ===================
L{aoj,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
-- other matched rules --
L{def,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
R{:::}{:::}()P1;
==================================================
>>>>> lnode
[خالد ]{} "Khalid" (ncnp,aoj,def,N,SING,M,SAHIH) <A,1,1>;
>>>>> rnode
[درس]{} "lesson" (def,N,M,IRGPL,INDEF,SING,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / [درس / [ خالد] / [ فهم /
==================================================
lnode >>>>>
[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[خالد ]{} "Khalid" (ncnp,aoj,def,N,SING,M,SAHIH) <A,1,1>;
=================== APPLIED RULE ===================
<{V,^agt:+agt::}{ncnp::agt:}()P8;
-- other matched rules --
R{:::}{:::}()P1;
==================================================
>>>>> lnode
[<<]{} "" (SHEAD) <.,0,0>;
>>>>> rnode
[فهم ]{} "uNDErstand" (agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ [<<] / [درس / [ فهم /
==================================================
lnode >>>>>
[<<]{} "" (SHEAD) <.,0,0>;
rnode >>>>>
[فهم ]{} "uNDErstand" (agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== APPLIED RULE ===================
R{:::}{:::}()P1;
-- other matched rules --
==================================================
>>>>> lnode
[<<]{} "" (SHEAD) <.,0,0>;
>>>>> rnode
[فهم ]{} "uNDErstand" (agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ << / [درس] / [ فهم] /
====================================================
lnode >>>>>
[فهم ]{} "uNDErstand" (agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[درس]{} "lesson" (def,N,M,IRGPL,INDEF,SING,SAHIH) <A,1,1>;
==================== APPLIED RULE ==================
R{:::}{:::}()P1;
-- other matched rules --
====================================================
>>>>> lnode
[فهم ]{} "uNDErstand" (agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[درس]{} "lesson" (def,N,M,IRGPL,INDEF,SING,SAHIH) <A,1,1>;
=================== NODE LIST ===================
/ << / درس] / فهم] /
====================================================
lnode >>>>>
[درس]{} "lesson" (def,N,M,IRGPL,INDEF,SING,SAHIH) <A,1,1>;
rnode >>>>>
==================== APPLIED RULE ==================
L{def,^ncnp:+ncnp::}{^AL,^V,^RelPron:::}()P9;
-- other matched rules --
R{:::}{:::}()P1;
====================================================
>>>>> lnode
[درس]{} "lesson" (ncnp,def,N,M,IRGPL,INDEF,SING,SAHIH) <A,1,1>;
>>>>> rnode
[>>]{} "" (STAIL) <.,0,0>;
=================== NODE LIST ===================
/ << / [درس] / [ فهم] / >> /
====================================================
lnode >>>>>
[فهم ]{} "uNDErstand" (agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[درس]{} "lesson" (ncnp,def,N,M,IRGPL,INDEF,SING,SAHIH) <A,1,1>;
==================== APPLIED RULE ==================
<{V,agt,^obj:+obj,+sentence::}{ncnp::obj:}()P7;
-- other matched rules --
R{:::}{:::}()P1;
====================================================
>>>>> lnode
[<<]{} "" (SHEAD) <.,0,0>;
>>>>> rnode
[فهم ]{} "uNDErstand" (obj,sentence,agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ [<<] / [فهم ] / >> /
====================================================
lnode >>>>>
[<<]{} "" (SHEAD) <.,0,0>;
rnode >>>>>
[فهم ]{} "uNDErstand" (obj,sentence,agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
==================== APPLIED RULE ==================
R{:::}{:::}()P1;
-- other matched rules --
====================================================
>>>>> lnode
[<<]{} "" (SHEAD) <.,0,0>;
```

```
>>>>> rnode
[فهم ]{} "uNDErstand" (obj,sentence,agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
=================== NODE LIST ===================
/ << / [فهم ] / [>>] /
=====================================================
lnode >>>>>
[فهم ]{} "uNDErstand" (obj,sentence,agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
rnode >>>>>
[>>]{} "" (STAIL) <.,0,0>;
==================== APPLIED RULE ==================
R{sentence,^complete:+complete,+&@entry::}{STAIL:::}()P5;
-- other matched rules --
R{:::}{:::}()P1;
=====================================================
>>>>> lnode
[فهم ]{} "uNDErstand" (complete,&@entry,obj,sentence,agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
>>>>> rnode
[>>]{} "" (STAIL) <.,0,0>;
=================== NODE LIST ===================
/ << / فهم  / [>>] /
=====================================================
lnode >>>>>
[>>]{} "" (STAIL) <.,0,0>;
rnode >>>>>

[<<]{} "" (SHEAD) <.,0,0>;

[فهم ]{} "uNDErstand" (complete,&@entry,obj,sentence,agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
| <{V,agt,^obj:+obj,+sentence::}{ncnp::obj:}()P7;
| >obj
|-[فهم ]{} "uNDErstand" (agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
| | <{V,^agt:+agt::}{ncnp::agt:}()P8;
| | >agt
| |-[فهم ]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
| | | +{:::}{BLK:::}()P255;
| | |-[فهم]{} "uNDErstand" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
| | |-[ ]{} "" (BLK) <.,0,0>;
| |-[خالد  ]{} "Khalid" (ncnp,aoj,def,N,SING,M,SAHIH) <A,1,1>;
|   | <{def:+aoj:aoj:}{RelClause_obj:::}()P10;
|   | <aoj
|   |-[خالد  ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
|   | | +{:::}{BLK:::}()P255;
|   | |-[خالد ]{} "Khalid" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
|   | | | <{def:+aoj:aoj:}{defadj:::}()P10;
|   | | | <aoj
|   | | |-[خالد ]{} "Khalid" (def,N,SING,M,SAHIH) <A,1,1>;
|   | | | | +{:::}{BLK:::}()P255;
|   | | | |-[خالد]{} "Khalid" (def,N,SING,M,SAHIH) <A,1,1>;
|   | | | |-[ ]{} "" (BLK) <.,0,0>;
|   | | |-[ذكي]{} "clever" (defadj,N,SING,M,SAHIH,FtSFX) <A,1,1>;
|   | |   | >{FW,AL:::}{adj,^defadj:-adj,+defadj::}()P11;
|   | |   |-[ال]{} "" (FW,AL,ال) <A,8,1>;
|   | |   |-[ذكي]{} "clever" (N,adj,SING,M,SAHIH,FtSFX) <A,1,1>;
|   | |-[ ]{} "" (BLK) <.,0,0>;
|   |-[قابل ]{} "meet" (RelClause_obj,RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
|     | <{V,RelClause_agt,^RelClause_obj:+RelClause_obj::}{ncnp::obj:}()P11;
|     | >obj
|     |-[قابل ]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
|     | | +{:::}{BLK:::}()P255;
|     | |-[قابل]{} "meet" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
|     | | | >{RelPron:::}{V:+RelClause_agt::}()P12;
|     | | |-[الذي ]{} "who" (RelPron,SING,M) <A,9,9>;
|     | | | | +{:::}{BLK:::}()P255;
|     | | | |-[الذي]{} "who" (RelPron,SING,M) <A,9,9>;
|     | | | |-[ ]{} "" (BLK) <.,0,0>;
|     | | |-[قابل]{} "meet" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
|     | |-[ ]{} "" (BLK) <.,0,0>;
|     |-[محمد ]{} "MOHAMMAD" (ncnp,aoj,def,N,SING,M,SAHIH) <A,1,1>;
|       | <{def:+aoj:aoj:}{RelClause_obj:::}()P10;
|       | <aoj
|       |-[محمد ]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
|       | | +{:::}{BLK:::}()P255;
|       | |-[محمد]{} "MOHAMMAD" (aoj,def,N,SING,M,SAHIH) <A,1,1>;
|       | | | <{def:+aoj:aoj:}{defadj:::}()P10;
|       | | | <aoj
|       | | |-[محمد]{} "MOHAMMAD" (def,N,SING,M,SAHIH) <A,1,1>;
|       | | |-[قوي]{} "strong(icl>state)" (defadj,N,SING,M,RGMPL,SAHIH,FtSFX) <A,1,1>;
|       | |   | >{FW,AL:::}{adj,^defadj:-adj,+defadj::}()P11;
|       | |   |-[ال]{} "" (FW,AL,ال) <A,8,1>;
|       | |   |-[قوي]{} "strong(icl>state)" (N,adj,SING,M,RGMPL,SAHIH,FtSFX) <A,1,1>;
|       | |-[ ]{} "" (BLK) <.,0,0>;
|       |-[قرأ ]{} "read" (RelClause_obj,RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
|         | <{V,RelClause_agt,^RelClause_obj:+RelClause_obj::}{ncnp::obj:}()P11;
|         | >obj
|         |-[قرأ ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
|         | | +{:::}{BLK:::}()P255;
|         | |-[قرأ]{} "read" (RelClause_agt,V,REG,3P,PAST,TR,NPREP) <A,1,1>;
|         | | | >{RelPron:::}{V:+RelClause_agt::}()P12;
|         | | |-[الذي ]{} "who" (RelPron,SING,M) <A,9,9>;
|         | | | | +{:::}{BLK:::}()P255;
|         | | | |-[الذي]{} "who" (RelPron,SING,M) <A,9,9>;
|         | | | |-[ ]{} "" (BLK) <.,0,0>;
|         | | |-[قرأ]{} "read" (V,REG,3P,PAST,TR,NPREP) <A,1,1>;
|         | |-[ ]{} "" (BLK) <.,0,0>;
|         |-[كتاب  ]{} "BOOK" (ncnp,aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
|           | +{:::}{BLK:::}()P255;
|           |-[كتاب ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
|           | | <{mod:+aoj:aoj:}{defadj:::}()P10;
|           | | <aoj
|           | |-[كتاب ]{} "BOOK" (aoj,mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
|           | | | <{mod:+aoj:aoj:}{defadj:::}()P10;
|           | | | <aoj
|           | | |-[كتاب ]{} "BOOK" (mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
|           | | | | +{:::}{BLK:::}()P255;
|           | | | |-[كتاب]{} "BOOK" (mod,N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
|           | | | | | <{NDE,^mod:+mod::}{def::mod:}()P11;
|           | | | | | >mod
|           | | | | |-[كتاب]{} "BOOK" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
|           | | | | | | +{:::}{BLK:::}()P255;
|           | | | | | |-[كتاب]{} "BOOK" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,3,3>;
|           | | | | | |-[ ]{} "" (BLK) <.,0,0>;
|           | | | | |-[داود]{} "DAOUD" (def,N,SING,M,SAHIH) <A,1,1>;
|           | | | |-[ ]{} "" (BLK) <.,0,0>;
|           | | |-[مفيد]{} "useful" (defadj,N,M,INDEF,SING,RGMPL,SAHIH,FtSFX) <A,1,1>;
|           | |   | >{FW,AL:::}{adj,^defadj:-adj,+defadj::}()P11;
|           | |   |-[ال]{} "" (FW,AL,ال) <A,8,1>;
|           | |   |-[مفيد]{} "useful" (N,M,adj,INDEF,SING,RGMPL,SAHIH,FtSFX) <A,1,1>;
|           | |-[رائع]{} "fantastic" (defadj,N,SING,M,RGMPL,SAHIH,FtSFX) <A,1,1>;
|           |   | >{FW,AL:::}{adj,^defadj:-adj,+defadj::}()P11;
|           |   |-[ال]{} "" (FW,AL,ال) <A,8,1>;
|           |   |-[رائع]{} "fantastic" (N,adj,SING,M,RGMPL,SAHIH,FtSFX) <A,1,1>;
|           |-[ ]{} "" (BLK) <.,0,0>;
|-[درس]{} "lesson" (ncnp,def,N,M,IRGPL,INDEF,SING,SAHIH) <A,1,1>;
  | >{FW,AL:::}{NDE,^def:-NDE,+def::}()P12;
  |-[ال]{} "" (FW,AL,ال) <A,8,1>;
  |-[درس]{} "lesson" (N,M,NDE,IRGPL,INDEF,SING,SAHIH) <A,1,1>;

[>>]{} "" (STAIL) <.,0,0>;
;======================= UNL =======================
;فهم خالد الذكي الذي قابل محمدالقوي الذي قرأ كتاب داود المفيدالرائع الدرس
[S]
aoj(clever:0B, Khalid:04)
aoj(strong(icl>state):0V, MOHAMMAD:0P)
mod(BOOK:18, DAOUD:1D)
aoj(useful:1K, BOOK:18)
aoj(fantastic:1Q, BOOK:18)
obj(read:14, BOOK:18)
aoj(read:14, MOHAMMAD:0P)
obj(meet:0K, MOHAMMAD:0P)
aoj(meet:0K, Khalid:04)
agt(uNDErstand:00.@entry, Khalid:04)
obj(uNDErstand:00.@entry, lesson:1X)
[/S]
;==================================================
;;Time  0.0Sec
;;Done!
```

APPENDIX E
CONTENT EXTRACTION RULES

```
;=============<<<<blank>>>>>==================
<{vech:::}{TEMP:::}P200;
<{flat:::}{TEMP:::}P200;
+{vech:::}{TEMP:::}P200;
+{flat:::}{TEMP:::}P200;
<{vech:+blank::}{BLK:::}(TEMP)P200;
+{vech:+blank::}{BLK:::}(TEMP)P200;

<{vech:+add_motor,-add_nmod,-price_right,-add_year,-phone_added::}{motor:::}()P200;
<{vech:+add_year,-add_nmod,-add_motor,-price_right,-phone_added::}{year:::}()P200;
<{vech:+price_right,-add_year,-add_nmod,-add_motor,+DP,-phone_added::}{price,downpayment:::}()P200;
<{vech:+price_right,-add_year,-add_nmod,-add_motor,+INS,-phone_added::}{price,installment:::}()P200;
<{vech:+phone_added,-price_right,-add_year,-add_nmod,-add_motor::}{phone:::}()P70;
<{vech,want_add:+price_right,-add_year,-add_nmod,-add_motor,-phone_added::}{price:::}()P200;
<{vech,sale_add:+price_right,-add_year,-add_nmod,-add_motor,-phone_added::}{price:::}()P200;

<{vech:+price_right,-add_year,-add_nmod,-add_motor,-phone_added::}{price:::}()P5;

<{vech:::}{noise:::}P20;
<{vech:::}{TEMP:::}P200;
<{vech:::}{noise:::}(TEMP)P21;
<{vech:::}{noise:::}(^ANUM,^noise,^extra,^TEMP,^BLK,^SP)P200;
<{vech:::}{more:::}(^ANUM,^noise,^extra,^TEMP,^BLK,^SP)P200;
<{vech:::}{extra:::}(^ANUM,^noise,^extra,^TEMP,^BLK,^SP)P200;
<{vech:::}{from:::}(^ANUM,^noise,^extra,^TEMP,^BLK,^SP)P200;
<{vech:::}{more:::}(^ANUM,^noise,^extra,^TEMP,^BLK,^SP)P200;
<{vech:::}{extra:::}(^ANUM,^noise,^extra,^TEMP,^BLK,^SP)P200;
<{vech:::}{to:::}(^ANUM,^noise,^extra,^TEMP,^BLK,^SP)P200;
;<{:::}{more:::}P200;
<{flat:::}{extra,ok:::}P200;
L{extra,^ok:+ok::}{^BLK,^ANUM}P210;
;<{:::}{more:::}P200;

<{extra:::}{extra:::}P2;
<{noise:::}{noise:::}P2;
<{zarf:::}{benum:::}P2;
L(vech){:+end::}{STAIL:::}P2;
L(flat){:+end::}{STAIL:::}P2;
<{vech:::}{end:::}P2;
<{vech:::}{end:::}P2;
<{flat:::}{end:::}P2;
<{vech:::}{pur:::}P2;
<{vech:::}{pur:::}P2;
-{pur:::}{vech:::}P2;
<{vech:::}{zarf:::}P2;

<{want:::}{temp:::}P200;
<{sale:::}{temp:::}P200;
<{vech,price_right:::}{currency:::}P20;
<{vech,price_right:::}{currency:::}P20;
<{vech:::}{vech:::}P20;
```

```
<{vech:::}{vech:::}P20;
<{^SHEAD:::}{noise,^ANUM:::}(^ANUM,^BLK)P100;
<{^SHEAD:::}{SP,^ANUM:::}(^ANUM,^BLK)P200;
<{vech:+add_motor,-add_nmod,-price_right,-add_year::}{motor:::}P200;
<{vech:+add_year,-add_nmod,-add_motor,-price_right::}{year:::}P200;
;<{vech:+price_right,-add_year,-add_nmod,-add_motor::}{price:::}()P200;

;-{price:::}{ANUM:+price_left::}()P200;
;<{ANUM:+currency::}{currency:::}()P200;
<{ANUM:+cc::}{motorsize:::}()P200;
+(SHEAD){vech,make_add:+BLANK::}{BLK:::}()P200;
;<{vech,make_add:+BLANK::}{^model,^color,^make,^ANUM,^feature,^motor:::}()P200;
<{vech:+BLANK::}{BLK:::}(TEMP)P200;
<{make:+BLANK::}{BLK:::}(^area)P200;
<{model:+BLANK::}{BLK:::}(^area)P200;
<{vech:+BLANK::}{BLK:::}(^area)P20;
<{:+BLANK::}{BLK:::}()P20;
<{:+BLANK::}{BLK:::}(TEMP)P200;
<{:+BLANK::}{TEMP:::}()P2;
<{:+BLANK::}{TEMP:::}P2;
<{^SHEAD:::}{SP:::}P200;
<{vech,#INSE:::}{TEMP:::}P200;
<{vech,#INSE:::}{BLK:::}()P7;
;<{:+add_year::}{year:::}P200;
;<{vech,#INSE:+price_right::}{price:::}()P200;
;<{vech,make_add:+BLANK::}{^model,^color,^make,^ANUM,^feature,^motor:::}()P200;
<{vech,make_add:+BLANK::}{aqar:::}()P20;
<{vech:::}{aqar:::}()P20;
<{want:::}{want:::}()P20;
<{vech:+SC::}{SP:::}(TEMP)P200;
<{make:+SC::}{SP:::}(^area)P200;
<{model:+SC::}{SP:::}(^area)P200;
<{vech:++SC::}{SP::}(^area)P20;

<{:+SC::}{SP:::}(TEMP)P200;

<{vech:+BLANK::}{prep:::}()P200;
?{year:::}{^ANUM:::}P200;
?{year:::}{TEMP:::}P200;

?{area:::}{TEMP:::}P200;
?{SHEAD:::}{year:::}P210;
?{vech,want:::}{want:::}P210;
?{flat:::}{make:::}p210;
?{flat:::}{model:::}p210;
?{flat:::}{year:::}p210;
?R{make:::}{area:::}p199;
?R{model:::}{area:::}p199;
?{vech:::}{area:::}p200;
?{vech,add_motor,sale_add,year_add,BLANK,model_add,make_add,#INSE,vech:::}{want:::}P199;


;=============<<<<SHIFT_RIGHT>>>>>=================
<{vech:::}{ALF:::}P10;
<{vech:::}{currency:::}P5;
<{vech:::}{aqar:::}P5;
<{vech:::}{zawaj:::}P5;
<{vech:::}{ALF:::}P10;
<{vech:::}{currency:::}P5;
```

```
<{vech:::}{aqar:::}P5;
<{vech:::}{zawaj:::}P5;
<{flat:::}{ALF:::}P10;
<{flat:::}{car:::}P10;
<{flat:::}{zawaj:::}P10;
;L{^BLANK:+BLANK::}{STAIL:::}P10;
;L{^BLANK:+BLANK::}{NULL:::}P10;
L{STAIL:::}{BLK:::}P10;
L(BLANK){BLK}{^TEMP:::}P255;
R{vech:::}{:::}()P1;
R{make:::}{:::}()P1;
R{model:::}{:::}()P1;
R{color:::}{:::}()P1;
R{sale:::}{:::}()P1;
R{want:::}{:::}()P1;
R{price:::}{:::}()P1;
R{year:::}{:::}()P1;

R{SHEAD:::}{vech:::}(TEMP)P20;
R{SHEAD:::}{make:::}()P20;
R{SHEAD:::}{vech,sale_add:::}(BLK)(price)P2;
R{SHEAD:::}{vech:::}(^aqar)P2;
R{SHEAD:::}{:::}P2;
R{SHEAD:::}{want:::}P20;
R{SHEAD:::}{vech:::}P20;
R{flat:::}{ANUM:::}(area)P20;
R{want:::}{flat:::}P20;
R{flat,loc_added:::}{ftype:::}P20;
R{flat,loc_added:::}{F_feature:::}P20;
R{sale:::}{ANUM:::}(BLK)(area)P20;
R{sale:::}{ANUM:::}(area)P20;
R{SHEAD:::}{ANUM:::}(BLK)(area)P20;
R{SHEAD:::}{ANUM:::}(area)P20;
R{:::}{:::}P2;

L{make,^add_vech:+add_vech::}{year:::}P20;
L{make,^add_vech:+add_vech::}{model:::}P20;
L{make,^add_vech:+add_vech::}{feature:::}P20;
L{make,^add_vech:+add_vech::}{motor:::}P20;
L{make,^add_vech:+add_vech::}{extra:::}P20;
L{make,^add_vech:+add_vech::}{noise:::}P20;
L{make,^add_vech:+add_vech::}{price:::}P20;
L{make,^add_vech:+add_vech::}{TEMP:::}P20;
L{make,nmodel,^add_vech:+add_vech::}{ANUM:::}P20;
L{model,^add_vech:+add_vech::}{make:::}P20;
L{model,^add_vech:+add_vech::}{year:::}P20;
L{model,^add_vech:+add_vech::}{feature:::}P20;
L{model,^add_vech:+add_vech::}{noise:::}P20;
L{model,^add_vech:+add_vech::}{price:::}P20;
L{model,^add_vech:+add_vech::}{TEMP:::}P20;
L{model,^add_vech:+add_vech::}{motor:::}P20;
L{ANUM,^add_vech:+add_vech::}{make,nmodel:::}P20;
L{make,^add_vech:+add_vech::}{sale:::}(model)P20;
L{model,^add_vech:+add_vech::}{sale:::}(make)P20;
L{make,^add_vech:+add_vech::}{want:::}(model)P20;
L{model,^add_vech:+add_vech::}{want:::}(make)P20;
L{want,^add_vech:+add_vech:::}{make:::}P20;
L{sale,^add_vech:+add_vech:::}{make:::}P20;
L{sale,^add_vech:+add_vech:::}{model:::}P20;
```

```
;==============<<<<sentence>>>>>==================
;
R{sentence,^complete:complete::}{STAIL:::}P5;
R{SHEAD:::}{vech,^sentence:+&@entry,sentence::}(STAIL)P6;
R(SHEAD){BLK:::}{vech,^sentence:+&@entry,sentence::}(STAIL)P6;
;==============<<<<vech insetion>>>>>==================


:"[سيارة]:+#INSE,+add_nmod::"{add_vech,ANUM:-add_vech::}P250;
:"[سيارة]:+#INSE::"{add_vech:-add_vech::}P250;


;==============<<<<make>>>>>==================
;<{vech,^make_add,#INSE:make_add::}{make,BLANK:-make:make:}()P80;
<{vech:make_add,+add_nmod::}{make,nmodel:-make:make:}()P80;
<{vech:make_add,add_year::}{make:-make:make:}()P80;
<{vech:make_add,+add_nmod::}{make,nmodel:-make:make:}P80;
<{vech:make_add,add_year::}{make:-make:make:}P80;
;==============<<<<model>>>>>==================
;<{vech,#INSE:model_add::}{model,BLANK:-model:model:}()P7;
<{vech:model_add,add_year::}{model:-model:model:}()P70;
<{vech,add_nmod,^model_add,add_year:model_add::}{ANUM::model:}()P70;
<{vech,add_nmod,^model_add,^add_year:model_add::}{ANUM::model:}()P70;
;==============<<<<color>>>>>==================
;<{vech,#INSE:color_add::}{color,BLANK:-color:color:}()P7;
<{vech:color_add::}{color:-color:color:}()P70;
;==============<<<<sale>>>>>==================
;<{vech,#INSE:sale_add::}{sale,BLANK:-sale:sale:}()P7;
<{vech:sale_add,add_year::}{sale:-sale:sale:}()P70;
>{sale:-sale:sale:}{vech:sale_add,add_year::}()P70;
;==============<<<<want>>>>>==================
<{vech,^want_add,^sale_add:want_add,add_year::}{want,BLANK:-want:want:}()P7;
>{want,BLANK:-want:want:}{vech,^want_add:want_add,add_year::}()P70;
>{want:-want:want:}{vech,^want_add:want_add,add_year::}()P70;
;==============<<<<year>>>>>==================

<{vech,add_year:year_add::}{ANUM,okyear::year:}P70;
L(add_year){ANUM,^okyear:+okyear::}{^notmore,^notless,^more,^less,^BLK:::}()P200;

;==============<<<<price>>>>>==================


<{vech,price_right,INS:price_add::}{ANUM,currency,BLANK:&@installment:price:}()P70;
<{vech,price_right,DP:price_add::}{ANUM,currency,BLANK:&@downpayment:price:}()P70;
<{vech,price_right:price_add::}{ANUM,currency,BLANK::price:}()P70;

<{vech,DP:price_add::}{ANUM,currency:&@downpayment:price:}()P60;
<{vech,INS:price_add::}{ANUM,currency:&@installment:price:}()P60;
<{vech:::}{ANUM,currency_ok::price:}()P80;
<{vech:price_add::}{ANUM,currency::price:}()P60;

;<{vech,price_right:price_add::}{ANUM,currency,BLANK::price:}()P70;
;<{vech,price_right:price_add::}{ANUM::price:}()P70;
;<{vech,^price_right:price_add::}{ANUM,currency::price:}()P70;

;<{vech:price_add::}{ANUM,price_left::price:}()P70;
;==============<<<<feature>>>>>==================
<{vech:sale_add::}{feature::feature:}()P70;
<{car:::}{tofemale:::}()P70;
```

```
<{aqar:::}{tofemale:::}()P70;
;=============<<<<country>>>>>==================
<{vech:country_add::}{country::country:}()P70;
;=============<<<<motor size>>>>>==================
<{vech,add_motor:motor_added::}{ANUM,okmotor::motor:}()P70;
<{vech,^add_motor:motor_added::}{ANUM,cc,okmotor::motor:}()P70;
<{vech,^add_motor:motor_added::}{ANUM,cc::motor:}()P70;
L(add_motor){ANUM,^okmotor:+okmotor::}{^notmore,^notless,^more,^less,^motorsize,^BLK:::}()P200;
;=============<<<<sale-flat>>>>>==================

<{flat:sale_add::}{sale:-sale:sale:}()P70;
>{sale:-sale:sale:}{flat:sale_add::}()P70;
;=============<<<<want-flat>>>>>==================
<{flat,^want_add:want_add::}{want,BLANK:-want:want:}()P7;
>{want:-want:want:}{flat,^want_add:want_add::}()P70;

;===============================================
<{flat,want_add:::}{pur::pur:}()P7;
<{flat:+pur_add::}{pur::pur:}()P7;
>{pur::pur:}{flat:+pur_add::}()P7;
;===============================================
<{flat:prep_add::}{prep:::}P200;
<{flat:zarf_add::}{zarf:::}P200;
<{flat:nArea_add::}{nArea:::}P200;
<{flat:::}{TEMP:::}P20;
-{prep:::}{flat:::}P200;
-{zarf:::}{flat:::}p200;
;===============================================
<{flat,prep_add:-prep_add,+loc_added::}{city_name:&@city:loc:}()P70;
<{flat,zarf_add:-zarf_add,+loc_added::}{city_name:&@city:loc:}()P70;
<{flat:+loc_added::}{city_name:&@city:loc:}()P70;
<{flat,prep_add:-prep_add,+loc_added::}{area_name:&@area:loc:}()P70;
<{flat,zarf_add:-zarf_add,+loc_added::}{area_name:&@area:loc:}()P70;
<{flat:+loc_added::}{area_name:&@area:loc:}()P70;
<{flat,prep_add:-prep_add,+loc_added::}{town_name:&@town:loc:}()P70;
<{flat,zarf_add:-zarf_add,+loc_added::}{town_name:&@town:loc:}()P70;
<{flat:+loc_added::}{town_name:&@town:loc:}()P70;
<{flat,prep_add:-prep_add,+loc_added::}{district_name:&@dist:loc:}()P70;
<{flat,zarf_add:-zarf_add::}{district_name:&@dist:loc:}()P70;
<{flat:+loc_added::}{district_name:&@dist:loc:}()P70;
<{flat,prep_add:-prep_add::}{location_name:&@loct:loc:}()P70;
<{flat,zarf_add:-zarf_add,+loc_added::}{location_name:&@loct:loc:}()P70;
<{flat:,+loc_added::}{location_name:&@loct:loc:}()P70;



;<{flat,nArea_add,^loc_added:+loc_added::}{TEMP::loc:}()P210;
;<{flat,prep_add,^loc_added:+loc_added::}{TEMP::loc:}()P210;
;<{flat,zarf_add,^loc_added:+loc_added::}{TEMP::loc:}()P210;

<{flat,nArea_add,^loc_added:+loc_added::}{TEMP:::}()P210;
<{flat,prep_add,^loc_added:+loc_added::}{TEMP:::}()P210;
<{flat,zarf_add,^loc_added:+loc_added::}{TEMP:::}()P210;


>{city_name::loc:}{flat:::}()P70;

>{area_name::loc:}{flat:::}()P70;
```

```
>{town_name::loc:}{flat:::}()P70;
>{location_name::loc:}{flat:::}()P70;

;==========================================================
<(^SHEAD){ANUM:+area_added,&@donem::}{area,donem:::}P200;
<(^SHEAD){ANUM:+area_added,&@meter::}{area,meter:::}P200;
<(^SHEAD){ANUM:+area_added::}{area:::}P200;

<{benum:+area_added,&@donem,+ANUM::}{area,donem:::}P200;
<{benum:+area_added,&@meter,+ANUM::}{area,meter:::}P200;
<{benum:+area_added,+ANUM::}{area:::}P200;
;<{area:::}{ANUM:::}P200;
;<{area:::}{ANUM:::}P200;
;<{area:::}{ANUM:::}P200;

<{flat:+area_right,+build,-space,-price_right::}{masaha,build:::}()P200;
<{flat:+area_right,+space,-build,-price_right::}{masaha,space:::}()P200;
<{flat:+area_right,-price_right::}{masaha:::}()P200;

L{ANUM,^area_ok:+area_ok::}{^notmore,^notless,^more,^less,^ALF,^area,^BLK:::}()P200;

<{flat,area_right,build:area_add,-area_right::}{ANUM,area_ok:&@build:area:}()P70;

;<{flat,area_right,build:area_add,-area_right::}{ANUM:&@build:area:}()P70;

<{flat,area_right,space:area_add,-area_right::}{ANUM,area_ok:&@space:area:}()P70;

;<{flat,area_right,space:area_add,-area_right::}{ANUM:&@space:area:}()P70;

<{flat,area_right,^area_add:area_add::}{ANUM,area_ok::area:}()P70;
<{flat:area_add::}{ANUM,area_ok,area_added::area:}()P70;
;<{flat,^area_add,area_right:area_add::}{ANUM::area:}(BLK)P70;
;<{flat,area_add,area_right:::}{area,^ANUM:::}()P200;
;=================================================
<(^SHEAD){ANUM:+room::}{room:::}()P200;
<{benum:+room,+ANUM::}{room:::}()P200;
L{room:add_room::}{:::}()P200;
<{flat:room_add::}{ANUM,room::room:}()P70;
<{flat:room_add::}{room,add_room::cns:}()P70;
;====================================================


;+:*{ANUM,^FIG:::}{NUM,FIG:::}P200;
<{ANUM:&@ALF::}{ALF:::}P200;
<{flat:+price_right,+DP,-area_right::}{price,downpayment:::}()P200;
<{flat:+price_right,+INS,-area_right::}{price,installment:::}()P200;
<{flat:+price_right,-area_right::}{price:::}()P200;
-{notless:::}{price:::}()P200;
-{notmore:::}{price:::}()P200;
-{less:::}{price:::}()P200;
-{more:::}{price:::}()P200;

>{price:::}{flat:+price_left::}()P200;
>{currency,currency_ok::price:}{flat,price_left:::}()P200;

<{ANUM:::}{noise:-noise::}()P100;

<{ANUM:&@year::}{peryear:::}()P200;
<{ANUM:&@month::}{permonth:::}()P200;
```

```
;<{ENUM,FIG:&@year::}{peryear:::}()P200;
;<{ENUM,FIG:&@month::}{permonth:::}()P200;

-{peryear:::}{ANUM:&@year::}P200;
-{permonth:::}{ANUM:&@month::}P200;

-{notless,before:::}{ANUM:&@more::}P200;
-{notmore,before:::}{ANUM:&@less::}P200;
-{more:::}{ANUM:&@more::}P200;
-{less:::}{ANUM:&@less::}P200;

-{notless,before:::}{masaha:+more_add::}P200;
-{notmore,before:::}{masaha:+less_add::}P200;
-{more:::}{masaha:+more_add::}P200;
-{less:::}{masaha:+less_add::}P200;

<{ANUM:&@more::}{notless:::}P200;
<{ANUM:&@more::}{more:::}P200;
<{ANUM:&@less::}{notmore:::}P200;
<{ANUM:&@less::}{less:::}P200;

<{ANUM:&@permeter::}{permeter:::}P200;
<{ANUM:&@perdonem::}{perdonem:::}P200;
-{permeter:::}{ANUM:&@permeter::}P200;
-{perdonem:::}{ANUM:&@perdonem::}P200;

-{zarf:::}{ANUM:::}P200;
-{noise:::}{ANUM:::}P100;
-{TEMP:::}{ANUM:::}P200;
<{ANUM:+currency_ok::}{currency:::}()P240;
;<{ENUM,FIG:::}{currency:::}()P200;

L(price_right){ANUM,^currency:+currency::}{STAIL:::}()P200;

L(^vech){ANUM,^currency:+currency::}{price:::}()P250;
L(price_right){ANUM,^currency,^BLK:+currency::}{^notmore,^notless,^more,^less,^peryear,^permonth,^pemeter,^
perdonem,^ALF,^BLK:::}()P200;

<{flat,price_right,INS:price_add::}{ANUM,currency,BLANK:&@installment:price:}()P70;
<{flat,price_right,DP:price_add::}{ANUM,currency,BLANK:&@downpayment:price:}()P70;
<{flat,price_right:price_add::}{ANUM,currency,BLANK::price:}()P70;

<{flat,DP:price_add::}{ANUM,currency:&@downpayment:price:}()P60;
<{flat,INS:price_add::}{ANUM,currency:&@installment:price:}()P60;
<{flat:price_add::}{ANUM,currency,currency_ok::price:}()P60;
<{flat:price_add::}{ANUM,currency_ok::price:}()P60;

;===========================================================
<{flat:type_add::}{ftype::type:}()P70;
;============================================================
<{flat:+consist::}{consist:::}()P200;
-{ANUM:::}{pof:num_add}()P200;
<{flat:::}{pof,dual:&@dual:cns:}()P70;
<{flat:::}{pof,plur:&@plural:cns:}()P70;
<{flat:::}{pof,^dual,^plur,^room::cns:}()P70;
;============================================================
<{flat:+floor:::}{floor:::}()P200;
<{flat,floor:-floor::}{enum,adj::flr:}()P70;
<{flat,floor:-floor::}{ANUM::flr:}()P70;
```

```
<{flat:::}{enum,adj::flr:}()P70;
;-----------------------------------------------
<{F_feature:::}{ANUM:::}(^area,^year)P220;
L{F_feature,^OK:+OK::}{^ANUM,^BLK:::}P240;
L{feature,^OK:+OK::}{^ANUM:::}P240;
<{flat:::}{F_feature,OK::fea:}()P70;
<{flat:::}{feature,ok::fea:}()P70;
;================================================
?L{masaha,space}{SALE}p250;
<{flat:::}{noise:::}()P100;
;===================================
L{dim,^OK:+OK::}{^ANUM,^BLK,^TEMP:::}P240;
<{dim,anum_add:::}{area:::}P250;
<{dim:+anum_add::}{ANUM:::}P250;
<{flat:::}{dim,^F_feature,OK:::}P250;
<{flat:::}{year:::}P250;
;=========================================
<{extra:::}{ANUM:::}P250;
<{noise:::}{ANUM:::}P100;
<{TEMP:::}{ANUM:::}P250;
+(vech){pof:end::}{ANUM:::}(STAIL)P250;
<{vech:::}{pof,end:::}P250;
;+(vech){TEMP:end::}{ANUM:::}(STAIL)P250;
;<{vech:::}{TEMP,end:::}P250;
+(vech){noise:end::}{ANUM:::}(STAIL)P250;
<{vech:::}{noise,end:::}P250;
<{vech:::}{noise,end:::}P250;
+(vech){extra:end::}{ANUM:::}(STAIL)P250;
<{vech:::}{extra,end:::}P250;
<{vech:::}{extra,end:::}P250;

;<{flat:::}{TEMP,end:::}P250;
+(flat){noise:end::}{ANUM:::}(STAIL)P250;
<{flat:::}{noise,end:::}P250;
+(flat){extra:end::}{ANUM:::}(STAIL)P250;
<{flat:::}{extra,end:::}P250;
>{noise:::}{flat:::}()P220;
>{noise:::}{vech:::}()P220;
>{extra:::}{flat:::}()P7;
>{extra:::}{vech:::}()P7;
>{TEMP:::}{flat:::}()P7;
>{TEMP:::}{vech:::}()P7;
<{pmethod:::}{pmethod:::}()P7;
<{flat:::}{pmethod:::}()P7;
>{extra:::}{pof:::}P7;
;===================zawaj======================================
<{nofemale,#INSE:::}{BLK:::}()P7;
<{nofemale,#INSE:::}{noise:::}()P7;
<{nofemale,#INSE:::}{TEMP:::}()P7;
<{nomale,#INSE:::}{BLK:::}()P7;
<{nomale,#INSE:::}{noise:::}()P7;
<{nomale,#INSE:::}{TEMP:::}()P7;

L(SHEAD){male,^nofemale:+add_male::}{:::}P75;
L{female,tomarry_add:::}{male,^nofemale:+add_male::}P100;
:"[شاب]:+#INSE::"{add_male,male:-add_male::}P250;
:"[شاب]:+#INSE::"{add_male,tomarry:-add_male::}P250;
```

```
L(SHEAD){female,^nomale:+add_female::}{:::}P75;
L{male,tomarry_add:::}{female,^nomale:+add_female::}P100;
L(SHEAD){tomarry:+add_male::}{female,nomale:::}P75;
L(SHEAD){tomarry:+add_female::}{male,nofemale:::}P75;
L(SHEAD){want:+add_male,+tomarry,-want::}{female,nomale:::}P75;
L(SHEAD){want:+add_female,+tomarry,-want::}{male,nofemale:::}P75;

:"[فتاة]:+#INSE::"{add_female,female:-add_female::}P250;
:"[فتاة]:+#INSE::"{add_female,tomarry:-add_female::}P250;



;==================================================================

<{male:origin_add::}{origin::origin:}(^tofemale,^BLK)P70;
;============================================================
<{male:carry_add::}{carry:::}()P70;
<{male,carry_add:education_add::}{education::edu:}()P70;
<{male:education_add::}{education::edu:}()P70;
;============================================================
<{male:status_add::}{maritalstatus::status:}(^tofemale,^BLK)P70;
;============================================================
<{male:zarf_add::}{zarf:::}()P70;
<{male:in_add::}{prep,in:::}()P70;
<{male:::}{prep:::}()P70;
<{male:residence_add::}{residence:::}()P70;
<{male:workin_add::}{workin:::}()P70;
<{male,zarf_add:residence_loc::}{country::residence:}()P70;
<{male,residence_add:residence_loc::}{country::residence:}()P70;
<{male,workin_add:residence_loc::}{country::residence:}()P70;
<{male,in_add:residence_loc::}{country::residence:}()P70;

<{male,zarf_add:residence_loc::}{location_name::residence:}()P70;
<{male,residence_add:residence_loc::}{location_name::residence:}()P70;
<{male,workin_add:residence_loc::}{location_name::residence:}()P70;
;============================================================
<{male:job_add::}{job::job:}(^tofemale,^BLK)P70;
;============================================================
<{male:Z_feature_add::}{Z_feature::fea:}(^tofemale,^BLK)P70;
;============================================================
<{male:relegion_add::}{relegion::rel:}(^tofemale,^BLK)P70;
;=========================================
<{male:tomarry_add::}{tomarry:::}()P70;
<{male:tomarry_add::}{want:::}()P70;
;L(male){female:ok_add::}{^country,^tomarry,^BLK,^tofemale,^relegion,^Z_feature,^job,^residence,^workin,^countr
y,^carry,^education,^maritalstatus,^origin,^aqar,^TEMP,^car:::}P60;
L(male){female:ok_add::}{STAIL:::}P70;
<{male,tomarry_add:::}{female,ok_add::marry}P70;

<{female:tomarry_add::}{tomarry:::}()P70;
;L(female){male,nofemale:ok_add::}{^country,^tomarry,^BLK,^tofemale,^relegion,^Z_feature,^job,^residence,^worki
n,^country,^carry,^education,^maritalstatus,^origin,^aqar,^TEMP,^car:::}P60;
L(female){male,nofemale:ok_add::}{STAIL:::}P70;
<{female,tomarry_add:::}{male,ok_add::marry}P70;
;===================Female===========================================
============
<{male,^nofemale:-male,+female::}{tofemale:::}()P250;
;========================================================
<{female:origin_add::}{origin::origin:}(^tofemale,^BLK)P70;
```

```
;=============================================================
<{female:carry_add::}{carry:::}(^tofemale,^BLK)P70;
<{female,carry_add:education_add::}{education::edu:}()P70;
;=============================================================
<{female:status_add::}{maritalstatus::status:}(^tofemale,^BLK)P70;
;=============================================================
<{female:zarf_add::}{zarf:::}()P70;
<{female:in_add::}{prep,in:::}()P70;
<{female:::}{prep:::}()P70;
<{female:residence_add::}{residence:::}()P70;
<{female:workin_add::}{workin:::}()P70;
<{female,zarf_add:residence_loc::}{country::residence:}()P70;
<{female,residence_add:residence_loc::}{country::residence:}()P70;
<{female,workin_add:residence_loc::}{country::residence:}()P70;
<{female,in_add:residence_loc::}{country::residence:}()P70;

<{female,zarf_add:residence_loc::}{location_name::residence:}()P70;
<{female,residence_add:residence_loc::}{location_name::residence:}()P70;
<{female,workin_add:residence_loc::}{location_name::residence:}()P70;
;=============================================================
<{female:job_add::}{job::job:}(^tofemale,^BLK)P70;
;=============================================================
<{female:Z_feature_add::}{Z_feature::fea:}(^tofemale,^BLK)P70;
;=============================================================
<{female:relegion_add::}{relegion::rel:}(^tofemale,^BLK)P70;
;=========================================
;======================================================================
===
<{male,nofemale:::}{tofemale:::}()P70;
<{female:::}{tofemale:::}()P70;
<{male:::}{aqar:::}()P70;
<{male:::}{car:::}()P70;
<{male:::}{TEMP:::}()P70;
<{female:::}{aqar:::}()P70;
<{female:::}{car:::}()P70;
<{female:::}{TEMP:::}()P70;
;===============age================================

<{female,nomale:+age_added::}{age:::}()P70;
<{male,nofemale:+age_added::}{age:::}()P70;
<{ANUM:+sanah_added::}{sanah:::}P200;
L{ANUM,^sanah_ok:+sanah_ok::}{^notmore,^notless,^more,^less,^sanah,,^BLK:::}()P200;
<{female,nomale,age_added:-age_added::}{ANUM,sanah_ok::age:}()P70;
<{female,nomale:-age_added::}{ANUM,sanah_ok,sanah_added::age:}()P70;
<{male,nofemale,age_added:-age_added::}{ANUM,sanah_ok::age:}()P70;
<{male,nofemale:-age_added::}{ANUM,sanah_ok,sanah_added::age:}()P70;

;===============height================================
<{female,nomale:+height_added::}{height:::}()P70;
<{male,nofemale:+height_added::}{height:::}()P70;

L{ANUM,^height_ok:+height_ok::}{^notmore,^notless,^more,^less,^BLK:::}()P200;
<{female,nomale,height_added:-height_added::}{ANUM,height_ok::height:}()P70;

<{male,nofemale,height_added:-height_added::}{ANUM,height_ok::height:}()P70;
;===============weight================================
<{female,nomale:+weight_added::}{weight:::}()P70;
<{male,nofemale:+weight_added::}{weight:::}()P70;
```

```
L{ANUM,^weight_ok:+weight_ok::}{^notmore,^notless,^more,^less,^BLK:::}()P200;
<{female,nomale,weight_added:-weight_added::}{ANUM,weight_ok::weight:}()P70;

<{male,nofemale,weight_added:-weight_added::}{ANUM,weight_ok::weight:}()P70;
;===========================================================
-{pron:::}{female:::}P75;
-{pron:::}{male:::}P75;
-{pron:::}{tomarry:::}P75;
;================================================
-{takoon,^not:::}{female:::}P75;
-{takoon,^not:::}{male:::}P75;
-{takoon,not:::}{female:&@not::}P75;
-{takoon,not:::}{male:&@not::}P75;
-{not:::}{male:&@not::}P75;
;=============================================
<{^SHEAD:::}{from:::}()P6;
<{town_name:::}{ANUM:::}P10;
<{area_name:::}{ANUM:::}P10;
<{location_name:::}{ANUM:::}P10;
<{district_name:::}{ANUM:::}P10;

-{weight_ok,height_ok,sanah_ok,currency,area_ok,okmotor,okyear,BLANK,NUM,ANUM:::}{:::}P4;
<{^male:::}{tofemale:::}P10;
<{flat:::}{country:::}P10;
+(SHEAD){want:::}{noise:::}P10;
+(SHEAD){want:::}{extra:::}P10;
+(SHEAD){want:::}{TEMP:::}P10;
+(SHEAD){want:::}{prep:::}P10;
+(SHEAD){want:::}{zarf:::}P10;
+(SHEAD){sale:::}{prep:::}P10;
+(SHEAD){sale:::}{zarf:::}P10;
;<{flat:::}{more:::}P10;
;<{flat:::}{notmore:::}P10;
;<{flat:::}{less:::}P10;
;<{flat:::}{notless:::}P10;
;<{vech:::}{more:::}P10;
;<{vech:::}{notmore:::}P10;
;<{vech:::}{less:::}P10;
;<{vech:::}{notless:::}P10;
<{flat:::}{to:::}P10;
<{vech:::}{to:::}P10;
;================================================
L(SHEAD){area_added,ANUM:+add_land::}{:::}P75;
L(currency,currency_ok){price:::}{area_added,ANUM:+add_land::}P75;
L(SHEAD){sale:add_land::}{area_added,ANUM:+add_land::}P75;
L(SHEAD){want:add_land::}{area_added,ANUM:+add_land::}P75;
L(SHEAD){sale:add_land::}{district_name:+add_land::}P75;
L(SHEAD){sale:add_land::}{town_name:+add_land::}P75;
L(SHEAD){sale:add_land::}{area_name:+add_land::}P75;
L(SHEAD){sale:add_land::}{location_name:+add_land::}P75;
:"[ارض]:+#INSE::"{add_land:-add_land::}P250;

<(SHEAD){ANUM:+area_added,&@donem::}{area,donem:::}P200;
<(SHEAD){ANUM:+area_added,&@meter::}{area,meter:::}P200;
<(SHEAD){ANUM:+area_added::}{area:::}P200;


;<{SHEAD:::}{TEMP:::}()P200;
<{TEMP:::}{TEMP:::}()P200;
```

```
<{SP:::}{TEMP:::}()P200;
<{SP:::}{noise:::}()P200;
<{noise:::}{noise:::}()P200;
<{noise:::}{TEMP:::}()P200;
<{TEMP:::}{noise:::}()P200;
<{TEMP:::}{SP:::}()P200;
<{TEMP:::}{BLK:::}()P200;
<{extra:::}{ANUM:::}()P200;

;DR{SHEAD:::}{ANUM:::}()P250;
;DR{SHEAD:::}{["]:::}()P250;
;DR{SHEAD:::}{[?]:::}()P250;
;DR{SHEAD:::}{SP:::}()P250;
;DR{SHEAD:::}{BLK:::}()P250;
;R{:::}{SHEAD:::}()P25;
<{flat:::}{area:::}()P200;
>{ANUM,BLANK:::}{extra:::}P200;
>{ANUM,BLANK:::}{noise:::}P200;
-{weight_ok,height_ok,sanah_ok,area_ok,BLANK,NUM,ANUM:::}{^price:::}P4;
;===================phone=========================
L(phone_added){ANUM,^phone_ok:+phone_ok::}{^ANUM,^BLK:::}()P200;
L(phone_added){ANUM,^phone_ok:+phone_ok::}{STAIL:::}()P200;
+{ANUM:::}{ANUM:::}P200;
<{vech,phone_added:::}{ANUM,phone_ok:::}()P70;
<{flat:+phone_added::}{phone:::}()P70;

<{flat,phone_added:::}{ANUM,phone_ok:::}()P70;
```

APPENDIX F
CATS : TABLES DESIDGN

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Cars]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Cars]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[RealEstate]') and OBJECTPROPERTY(id, N'IsUserTable') =
1)
drop table [dbo].[RealEstate]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[knowledgeBase]') and OBJECTPROPERTY(id, N'IsUserTable')
=
1)
drop table [dbo].[knowledgeBase]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[msgs]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[msgs]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[msgsLog]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[msgsLog]
GO

CREATE TABLE [dbo].[Cars] (
 [ID] [bigint] IDENTITY (1, 1) NOT NULL ,
 [msgCaller] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [msgDate] [datetime] NULL ,
 [msgTxt] [nvarchar] (480) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [maincat] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [make] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [model] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [country] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [motor] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [makeyear] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [price] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [color] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [feature] [nvarchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [strGUID] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [IsBuyer] [bit] NULL ,
 [IsInActive] [bit] NOT NULL ,
 [SendFlag] [bit] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[RealEstate] (
 [IsBuyer] [bit] NULL ,
 [Purpose] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [Location] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [Area] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [Room] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [ConsistOf] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [Price] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [Type] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [Floor] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [Feature] [nvarchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [ID] [bigint] IDENTITY (1, 1) NOT NULL ,
 [msgCaller] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [msgDate] [datetime] NULL ,
 [msgTxt] [nvarchar] (480) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [strGUID] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [knldgLoc] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [LocAttribute] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
,
 [areaAttribute] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
,
 [priceAttribute] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL
,
 [estateType] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [isInActive] [bit] NOT NULL ,
 [SendFlag] [bit] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[knowledgeBase] (
 [make] [nvarchar] (255) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [model] [nvarchar] (255) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [country] [nvarchar] (255) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[msgs] (
 [ID] [bigint] IDENTITY (1, 1) NOT NULL ,
 [msgCaller] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [msgDate] [datetime] NULL ,
 [msgTxt] [nvarchar] (480) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [strGUID] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[msgsLog] (
 [msgCaller] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [msgTxt] [nvarchar] (2000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
 [msgLength] [int] NULL
) ON [PRIMARY]
GO
```

APPENDIX G
CATS : STORED PROCEDURES

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ST_ActivateDeactivate]') and OBJECTPROPERTY(id,
N'IsProcedure') = 1)
drop procedure [dbo].[ST_ActivateDeactivate]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ST_AddCarMsg]') and OBJECTPROPERTY(id, N'IsProcedure') =
1)
drop procedure [dbo].[ST_AddCarMsg]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ST_AddMsg]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[ST_AddMsg]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ST_AddMsgsLog]') and OBJECTPROPERTY(id, N'IsProcedure')
= 1)
drop procedure [dbo].[ST_AddMsgsLog]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ST_AddRealEstate]') and OBJECTPROPERTY(id,
N'IsProcedure') = 1)
drop procedure [dbo].[ST_AddRealEstate]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ST_DeleteMsgsLog]') and OBJECTPROPERTY(id,
N'IsProcedure') = 1)
drop procedure [dbo].[ST_DeleteMsgsLog]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ST_GetCarMsgs]') and OBJECTPROPERTY(id, N'IsProcedure')
= 1)
drop procedure [dbo].[ST_GetCarMsgs]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ST_GetMsg]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[ST_GetMsg]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ST_GetMsgsLog]') and OBJECTPROPERTY(id, N'IsProcedure')
= 1)
drop procedure [dbo].[ST_GetMsgsLog]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ST_GetRealEstateMsgs]') and OBJECTPROPERTY(id,
N'IsProcedure') = 1)
drop procedure [dbo].[ST_GetRealEstateMsgs]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ST_GetUnSentMsgs]') and OBJECTPROPERTY(id,
N'IsProcedure') = 1)
drop procedure [dbo].[ST_GetUnSentMsgs]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ST_SetSendFlag]') and OBJECTPROPERTY(id, N'IsProcedure')
= 1)
drop procedure [dbo].[ST_SetSendFlag]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ST_UpdateCarMsg]') and OBJECTPROPERTY(id,
N'IsProcedure') = 1)
drop procedure [dbo].[ST_UpdateCarMsg]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ST_UpdateRealEstate]') and OBJECTPROPERTY(id,
N'IsProcedure') = 1)
drop procedure [dbo].[ST_UpdateRealEstate]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ST_reset]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[ST_reset]
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE  PROCEDURE ST_ActivateDeactivate (
@msgCaller as nvarchar(30),
@activateFlag as bit
)
 AS
SET NOCOUNT ON

Update Cars set
IsInActive = @activateFlag
where msgCaller = @msgCaller

Update RealEstate set
IsInActive = @activateFlag
where msgCaller = @msgCaller

return 0
GO
SET QUOTED_IDENTIFIER OFF
GO
```

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS OFF
GO

CREATE  PROCEDURE ST_AddCarMsg (
@ID as bigint,
@msgCaller as nvarchar(30),
@msgDate as datetime,
@msgTxt  as nvarchar(480),
@maincat  as nvarchar(30),
@make  as nvarchar(30),
@model  as nvarchar(30),
@country  as nvarchar(30),
@motor  as nvarchar(30),
@makeyear  as nvarchar(30),
@price  as nvarchar(30),
@color  as nvarchar(30),
@feature  as nvarchar(200),
@strGUID  as nvarchar(50),
@IsBuyer as bit
)
 AS
SET NOCOUNT ON

INSERT INTO [KnowledgeBase].[dbo].[cars](
[msgCaller], [msgDate], [msgTxt], [maincat], [make], [model], [country],
[motor],
[makeyear], [price], [color], [feature], [strGUID], [IsBuyer])
VALUES(
--@ID,
@msgCaller,
@msgDate,
@msgTxt,
@maincat,
@make,
@model,
@country,
@motor,
@makeyear,
@price,
@color,
@feature,
@strGUID,
@IsBuyer
)

return 0
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
```

```
SET ANSI_NULLS OFF
GO

CREATE  PROCEDURE ST_AddMsg (
@ID as bigint,
@msgCaller as nvarchar(30),
@msgDate as datetime,
@msgTxt  as nvarchar(480),
@strGUID  as nvarchar(50)
)
 AS
SET NOCOUNT ON

INSERT INTO [KnowledgeBase].[dbo].[msgs](
[msgCaller], [msgDate], [msgTxt], [strGUID])
VALUES(
--@ID,
@msgCaller,
@msgDate,
@msgTxt,
@strGUID
)

return 0
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE  PROCEDURE ST_AddMsgsLog (

@msgCaller as nvarchar(30),
@msgTxt  as nvarchar(2000),
@msgLength as int

)
 AS
SET NOCOUNT ON
If Exists( select * from msgsLog where msgCaller = @msgCaller) begin
select * from msgsLog

Update [msgsLog] set
[msgTxt]=@msgTxt,
[msgLength]=@msgLength
where msgCaller = @msgCaller

end
else begin

INSERT INTO [msgsLog](
[msgCaller],[msgTxt], [msgLength])
VALUES(
@msgCaller,
@msgTxt,
@msgLength
)
end


return 0
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE ST_AddRealEstate (
@ID as bigint
,@msgCaller as nvarchar(30)
,@msgDate as datetime
,@msgTxt  as nvarchar(480)
,@IsBuyer as bit
,@Purpose as nvarchar(30)
,@Location as nvarchar(100)
,@Area as nvarchar(30)
,@Room as nvarchar(30)
,@ConsistOf as nvarchar(30)
,@Price as nvarchar(30)
,@Type as nvarchar(30)
,@Floor as nvarchar(30)
,@Feature as nvarchar(200)
,@strGUID  as nvarchar(50)
,@knldgLoc  as nvarchar(100)
,@LocAttribute  as nvarchar(100)
,@areaAttribute  as nvarchar(100)
,@priceAttribute  as nvarchar(100)
,@estateType as nvarchar(30)

)
 AS
SET NOCOUNT ON

INSERT INTO [RealEstate]
([msgCaller],[msgDate],[msgTxt],[IsBuyer], [Purpose], [Location], [Area],
[Room], [ConsistOf], [Price], [Type], [Floor],
[Feature],[strGUID],[knldgLoc],[LocAttribute],[areaAttribute],[priceAttribu
te],[estateType])
VALUES(
@msgCaller
,@msgDate
,@msgTxt
,@IsBuyer
,@Purpose
,@Location
,@Area
,@Room
,@ConsistOf
```

```
,@Price
,@Type
,@Floor
,@Feature
,@strGUID
,@knldgLoc
,@LocAttribute
,@areaAttribute
,@priceAttribute
,@estateType
)
Return 0
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE  PROCEDURE ST_DeleteMsgsLog (

@msgCaller as nvarchar(30),
@msgTxt  as nvarchar(2000),
@msgLength as int

)
 AS
delete from msgslog where msgCaller = @msgCaller

return 0
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS OFF
GO

CREATE    PROCEDURE ST_GetCarMsgs(
@ID as bigint,
@msgCaller as nvarchar(30),
@msgDate as datetime,
@msgTxt  as nvarchar(480),
@maincat  as nvarchar(30),
@make  as nvarchar(30),
@model  as nvarchar(30),
@country  as nvarchar(30),
@motor  as nvarchar(30),
@makeyear  as nvarchar(30),
@price  as nvarchar(30),
@color  as nvarchar(30),
@feature  as nvarchar(200),
@strGUID  as nvarchar(50),
@IsBuyer as bit,
@queryDatePeriod as int,
@AndOR as smallint
)
 AS
SET NOCOUNT ON
Declare
@slc as nvarchar(4000),
@slcConst as nvarchar(4000),
@slcwhere as nvarchar(4000),
@slcGUID as nvarchar(4000),
@str As nvarchar(4000) ,
@str1 As nvarchar(4000),
@str2 As nvarchar(4000),
@strCondition As nvarchar(4000),
@condstr As nvarchar(4000),
@strConditionOr as nvarchar(4000)

set @slcConst = 'select id,* from Cars '
set @str = ''
set @str1 = ''
set @str2 = ''
set @strCondition = ''
set @condstr = ''
set @strConditionOr = ''

--Building Car Seach Criteria
set @country = rtrim(ltrim(@country))
If (@country <> Null)and (@country <>'')  begin
    Declare @countystr  nvarchar(4000)
     set  @countystr='%'' or country like ''%'
    set @countystr = replace(@country,',',@countystr)
                set @str = ' (country like ''%' + @countystr + '%'')'
 print @str
    set @condstr = ' and '
End
set @make = rtrim(ltrim(@make))
If (@make <> Null) and (@make <> '')  begin
    set @str1 = @condstr + ' make =''' + @make + ''''
    set @condstr = ' and '
End
set @model = rtrim(ltrim(@model))
if (@model <> Null) and (@model <> '') begin
                set @str2 = @condstr + ' model =''' + @model + ''''
                set @condstr = ' and '
end

If @str + @str1 + @str2 <> '' begin
                set @strCondition = 'where (' + @str + @str1 + @str2 + ')'
                set @strConditionOr = Replace(@strCondition,' and ', ' or
')

                set @condstr = @condstr + ' (maincat ='''+ @maincat +'''
and isbuyer=0 and IsInActive=0 ' + ' and MsgDate >=  Cast(''' +
Cast(DateAdd(Day,@queryDatePeriod,@msgDate) as nvarchar) + ''' as
datetime)) order by MsgDate'
                set @strCondition = @strCondition + @condstr
                set @strConditionOr = @strConditionOr + @condstr
```

```
End else begin
 set @price = rtrim(ltrim(@price))
 if (@price <> Null) and (@price <> '') begin
                set @str =  ' price <=' + @price
                set @condstr = ' and '
 end
 print(@str)
 set @condstr = ' where maincat ='''+ @maincat +''' and isbuyer=0 and
IsInActive=0 ' + @condstr+@str+' and MsgDate >=  Cast(''' +
Cast(DateAdd(Day,@queryDatePeriod,@msgDate) as nvarchar) + ''' as datetime)
order by MsgDate'
 set @strCondition =  @condstr
            set @strConditionOr =  @condstr
end
print @condstr + ' (maincat ='''+ @maincat +''' and isbuyer=0 and
IsInActive=0 ' + ' and MsgDate >=  Cast('''

if ((@AndOR = 0)  and (@strGUID  <>null)) begin --Guid Search
 set @slcwhere  = ' where '
 set @slcGUID =   ' strGuid=''' + @strGUID+''' '
 set @slc = @slcConst + @slcwhere + @slcGUID
end else  if ((@AndOR = 1)) begin  -- And  Search
 set @slc = @slcConst + @strCondition
end else  if ((@AndOR = 2)) begin  -- OR search
 set @slc = @slcConst + @strConditionOr
end

if @slc <> null begin
 print @slc
 exec (@slc)
end

--return 0
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE    PROCEDURE ST_GetMsg(
@ID as bigint,
@msgCaller as nvarchar(30),
@msgDate as datetime,
@msgTxt  as nvarchar(480),
@strGUID  as nvarchar(50)
)
 AS
SET NOCOUNT ON

select * from msgs where strGUID = @strGUID

Delete msgs where strGUID = @strGUID

return 0
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE  PROCEDURE ST_GetMsgsLog (

@msgCaller as nvarchar(30),
@msgTxt  as nvarchar(2000)

)
 AS
SET NOCOUNT ON

select * from msgsLog where  msgCaller = @msgCaller

return 0
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE    PROCEDURE ST_GetRealEstateMsgs(
@ID as bigint
,@msgCaller as nvarchar(30)
,@msgDate as datetime
,@msgTxt  as nvarchar(480)
,@Purpose as nvarchar(30)
,@Location as nvarchar(100)
,@Area as nvarchar(30)
,@Room as nvarchar(30)
,@ConsistOf as nvarchar(30)
,@Price as nvarchar(30)
,@Type as nvarchar(30)
,@Floor as nvarchar(30)
,@Feature as nvarchar(200)
,@strGUID  as nvarchar(50)
,@knldgLoc  as nvarchar(100)
,@LocAttribute  as nvarchar(100)
,@areaAttribute  as nvarchar(100)
,@priceAttribute  as nvarchar(100)
,@estateType as nvarchar(30)
,@IsBuyer as bit
,@queryDatePeriod as int
,@AndOR as smallint
)
 AS
```

```
SET NOCOUNT ON
Declare
@slc as nvarchar(4000),
@slcConst as nvarchar(4000),
@slcwhere as nvarchar(4000),
@slcGUID as nvarchar(4000),
@str As nvarchar(4000) ,
@str1 As nvarchar(4000),
@str2 As nvarchar(4000),
@str3 As nvarchar(4000),
@strCondition As nvarchar(4000),
@condstr As nvarchar(4000),
@strConditionOr as nvarchar(4000)

set @slcConst = 'select id,* from RealEstate '
set @str = ''
set @str1 = ''
set @str2 = ''
set @str3 = ''
set @strCondition = ''
set @condstr = ''
set @strConditionOr = ''

--Building Real Estate Seach Criteria
set @location = rtrim(ltrim(@location))
If (@location <> Null)and (@location <>'')  begin
    Declare @locationStr  nvarchar(4000)
    set  @locationStr='%'' or location like N''%'
    set @locationStr = replace(@location,',',@locationStr)
                set @str = ' (location like N''%' + @locationStr + '%'')'
    set @condstr = ' and '
End
set @knldgLoc = rtrim(ltrim(@knldgLoc))
If (@knldgLoc <> Null) and (@knldgLoc <> '')  begin
    Declare @knldgLocStr  nvarchar(4000)
    set  @knldgLocStr='%'' or knldgLoc like N''%'
    set @knldgLocStr = replace(@knldgLoc,',',@knldgLocStr)
                set @str1 = @condstr + ' (knldgLoc like N''%' +
@knldgLocStr + '%'')'
    set @condstr = ' and '
End

set @Purpose = rtrim(ltrim(@Purpose))
if (@Purpose <> Null) and (@Purpose <> '') begin
                set @str2 = @condstr + ' Purpose =''' + @Purpose + ''''
                set @condstr = ' and '
end
else
begin
                set @str2 = @condstr + ' Purpose is null '
                set @condstr = ' and '
end

set @Area = rtrim(ltrim(@Area))
if (@Area <> Null) and (@Area <> '') begin
                set @str3 = @condstr + ' Area between(' + @Area - @Area*40%
+','+ @Area + @Area*40% + ')'
                set @condstr = ' and '
end
```

```
If @str + @str1 +  @str3 <> '' begin
--                 set @strCondition = 'where (' + @str + @str1 + @str2 +
@str3+ ')'
  --               set @strConditionOr = Replace('where (' + @str + @str1,'
and ', ' or ') + @str2 + ')'
                 if  @str + @str1 + @str3 <> '' begin
  set @strCondition = 'where (' + @str + @str1 + @str3+ ')' + @str2
  set @strConditionOr = Replace('where (' + @str + @str1,' and ', ' or ')
+ ')' + @str2
   end
                else begin
                set @strCondition = 'where '  + @str2
                 set @strConditionOr = 'where '  + @str2
                 end

                 set @condstr = @condstr + ' (estateType='''+ @estateType +
''' and isbuyer=0 and IsInActive=0' + ' and MsgDate >=  Cast(''' +
Cast(DateAdd(Day,@queryDatePeriod,@msgDate) as nvarchar) + ''' as
datetime)) order by MsgDate'
                 set @strCondition = @strCondition + @condstr
                 set @strConditionOr = @strConditionOr + @condstr
End else begin
 set @condstr = ''
 set @price = rtrim(ltrim(@price))
 if (@price <> Null) and (@price <> '') begin
                  set @str =  ' cast(price as decimal) <=' + @price
                  set @condstr = ' and '
 end
 set @priceAttribute = rtrim(ltrim(@priceAttribute))
 if (@priceAttribute <> Null) and (@priceAttribute <> '') begin
                  set @str1 = @condstr + @priceAttribute + ' priceAttribute
=' + @priceAttribute
                  set @condstr = ' and '
 end

 set @condstr = ' where estateType='''+ @estateType + ''' and isbuyer=0 and
IsInActive=0 and ' + @str2 +@condstr+@str+@str1 + ' and MsgDate >=
Cast(''' + Cast(DateAdd(Day,@queryDatePeriod,@msgDate) as nvarchar) + '''
as datetime) order by MsgDate'
 set @strCondition =  @condstr
             set @strConditionOr =  @condstr
end


if ((@AndOR = 0)  and (@strGUID  <>null)) begin --Guid Search
 set @slcwhere  = ' where '
 set @slcGUID =   ' strGuid=''' + @strGUID+''' '
 set @slc = @slcConst + @slcwhere + @slcGUID
end else  if ((@AndOR = 1)) begin  -- And  Search
 set @slc = @slcConst + @strCondition

end else  if ((@AndOR = 2)) begin  -- OR search
 set @slc = @slcConst + @strConditionOr
end

if @slc <> null begin
 print @slc
 exec (@slc)
end

--return 0
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE  PROCEDURE ST_GetUnSentMsgs (
@ValidityPeriod  as integer
)
 AS
SET NOCOUNT ON
--it is impossible to have two messages with the same GUID ID
select * from Cars where
SendFlag = 0 and
isbuyer = 1 and
msgDate>= getDate() + @ValidityPeriod
select * from RealEstate where
SendFlag = 0 and
isbuyer = 1 and
msgDate>= getDate() + @ValidityPeriod

return 0
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE  PROCEDURE ST_SetSendFlag (
@strGUID  as nvarchar(50),
@SendFlag as bit
)
 AS
SET NOCOUNT ON
--it is impossible to have two messages with the same GUID ID
update Cars set SendFlag =  @SendFlag where strGUID=@strGUID and isbuyer =
1
update RealEstate set SendFlag =  @SendFlag where strGUID = @strGUID and
isbuyer = 1

return 0
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
```

```
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS OFF
GO

CREATE  PROCEDURE ST_UpdateCarMsg (
@ID as bigint,
@msgCaller as nvarchar(30),
@msgDate as datetime,
@msgTxt  as nvarchar(480),
@maincat  as nvarchar(30),
@make  as nvarchar(30),
@model  as nvarchar(30),
@country  as nvarchar(30),
@motor  as nvarchar(30),
@makeyear  as nvarchar(30),
@price  as nvarchar(30),
@color  as nvarchar(30),
@feature  as nvarchar(200),
@strGUID  as nvarchar(50),
@IsBuyer as bit
)
 AS
SET NOCOUNT ON

Update Cars set
--[ID] = @ID,
--[msgCaller] = @msgCaller,
--[msgDate] = @msgDate,
--[msgTxt] = @msgTxt,
[maincat] = @maincat,
[make] = @make,
[model] = @model,
[country] = @country,
[motor] = @motor,
[makeyear] = @makeyear,
[price] = @price,
[color] = @color,
[feature] = @feature,
--[strGUID] = @strGUID,
[IsBuyer] = @IsBuyer
where [id] = @id

return 0
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE ST_UpdateRealEstate (
@ID as bigint
,@msgCaller as nvarchar(30)
,@msgDate as datetime
,@msgTxt  as nvarchar(480)
,@IsBuyer as bit
,@Purpose as nvarchar(30)
,@Location as nvarchar(100)
,@Area as nvarchar(30)
,@Room as nvarchar(30)
,@ConsistOf as nvarchar(30)
,@Price as nvarchar(30)
,@Type as nvarchar(30)
,@Floor as nvarchar(30)
,@Feature as nvarchar(200)
,@strGUID  as nvarchar(50)
,@knldgLoc  as nvarchar(100)
,@LocAttribute  as nvarchar(100)
,@areaAttribute  as nvarchar(100)
,@priceAttribute  as nvarchar(100)
,@estateType as nvarchar(30)
)
 AS
SET NOCOUNT ON

Update [RealEstate] set
[msgCaller] = @msgCaller
,[msgDate] = @msgDate
,[msgTxt] = @msgTxt
,[strGUID] = @strGUID
,[IsBuyer] = @IsBuyer
, [Purpose] = @Purpose
,[Location] = @Location
,[Area] = @Area
, [Room] = @Room
,[ConsistOf] = @ConsistOf
,[Price] = @Price
,[Type] = @Type
,[Floor] = @Floor
, [Feature] = @Feature
,[knldgLoc] = @knldgLoc
,[LocAttribute] = @LocAttribute
,[areaAttribute] = @areaAttribute
,[priceAttribute] = @priceAttribute
,[estateType] = @estateType
where [id] = @id
Return 0
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE ST_Reset  AS
delete   realestate
delete cars
delete msgs
delete msgslog
return 0
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

APPENDIX H
CATS : SYSTEM VARIABLES

```
  <?xml version="1.0" encoding="utf-8" ?>
- <systemvariables>
  <EnchoBasePath>d:\Work Place\Projects\New Pattern Recognishion\enco</EnchoBasePath>
  - <!--
  <EnchoBasePath>D:\Projects Source Code\New Pattern
    Recognishion\enco</EnchoBasePath>
    -->
  <rootDirectory>d:</rootDirectory>
  <EnchoAppPath>EnCoL20.exe</EnchoAppPath>
  <EnchoDicPath>estate.dic</EnchoDicPath>
  <EnchoRulesPath>7-7-zawaj.rul</EnchoRulesPath>
  <OutMsgPathIn>parsedfilesIn\</OutMsgPathIn>
  <OutMsgPathOut>parsedfilesOut\</OutMsgPathOut>
  <options>-go -l0 -t5 -n1</options>
  <BatchfilePath>d:\Work Place\Projects\New Pattern Recognishion\enco\</BatchfilePath>
  - <!--
  <BatchfilePath>D:\Projects Source Code\New Pattern
    Recognishion\enco\</BatchfilePath>
    -->
  <yea>yea</yea>
  <fea>fea</fea>
  <wan>wan</wan>
  <mod>mod</mod>
  <mak>mak</mak>
  <pri>pri</pri>
  <mot>mot</mot>
  <col>col</col>
  <cou>cou</cou>
  <sal>sal</sal>
  <pur>pur</pur>
  <loc>loc</loc>
  <are>are</are>
  <roo>roo</roo>
  <cns>cns</cns>
  <typ>typ</typ>
  <flr>flr</flr>
  - <!--
  Data Source=D:\Projects Source Code\New Pattern Recognishion\Pattern
    Recognishion\DataBase\KnowledgeBase.mdb;
    -->
  <ConnectionString>data source=localhost;initial catalog=knowledgeBase;password=123;persist security info=True;user id=sa;workstation id=localhost;packet size=4096</ConnectionString>
```

```
          <ConnectionString2>data                    source=localhost;initial
   catalog=JobLink;password=123;persist        security          info=True;user
   id=sa;workstation id=localhost;packet size=4096</ConnectionString2>
    - <!--
    <ConnectionString>
    Provider=Microsoft.Jet.OLEDB.4.0;
    Password="";
    User ID=Admin;
    Data Source=C:\Work Place\Projects\New Pattern Recognishion\Pattern
    Recognishion\DataBase\KnowledgeBase.mdb;
    Mode=Share Deny None;
    Extended Properties="";
    Jet OLEDB:System database="";
    Jet OLEDB:Registry Path="";
    Jet OLEDB:Database Password="";
    Jet OLEDB:Engine Type=5;
    Jet OLEDB:Database Locking Mode=1;
    Jet OLEDB:Global Partial Bulk Ops=2;
    Jet OLEDB:Global Bulk Transactions=1;
    Jet OLEDB:New Database Password="";
    Jet OLEDB:Create System Database=False;
    Jet OLEDB:Encrypt Database=False;
    Jet OLEDB:Don't Copy Locale on Compact=False;
    Jet OLEDB:Compact Without Replica Repair=False;
    Jet OLEDB:SFP=False</ConnectionString>
  -->
- <!--
<dateFormates>73</dateFormates>
  -->
<dateFormates>27</dateFormates>
<httpResponse>http://127.0.0.1:8800/</httpResponse>
<queryDatePeriod>-100</queryDatePeriod>
<queryUnSetDatePeriod>-2</queryUnSetDatePeriod>
<MsgLength>160</MsgLength>
<MsgLength_Ar>137</MsgLength_Ar>
 <RegExp>(?<rel>\b\w{3}\b)\((?<firstword>\b\w*\b)(\s*)(\:)(\d{2}|\d\w|\w
   \d|\w{2})(\.\@\w+)*?(\,)(\s*)(?<secondword>[^(]+)(\s*)\(?(?<knowledge
   >\w+\<\w+?\,?)*?\)?(\:)(\d{2}|\d\w|\w\d|\w{2})(?<attribute>\.\@\w+)
   *?\)</RegExp>
<NextWords>next,تالى،تالي،مزيد</NextWords>
<HelpWords>help,مساعدة،مساعده</HelpWords>
<ActivateWords>activate,فعل،شغل،تشغيل،تفعيل</ActivateWords>
<DeActivateWords>deactivate,ايقاف،وقف،توقيف</DeActivateWords>
<HelpMessage>تشغيل، لاستكمال عرض المعلومات ابعث التالي لايقاف الخدمة ابعث ايقاف و لاعادة تشغيلها ابعث</HelpMessage>
<NoDataFoundMessage>ببعثها اتوماتيكيا لا يوجد معلومات حاليا فور توفرها سيقوم النظام</NoDataFoundMessage>
<errMsgPath>C:\RESULTS.txt</errMsgPath>
<logMsgPath>C:\log.txt</logMsgPath>
<MsgCallerStrip>+962</MsgCallerStrip>
<shortNumber>90050</shortNumber>
- <!--
Log
  -->
<logSize>100000</logSize>
<logPath>C:\logs\</logPath>
```

```
<confirmationMsg>Your advertisement has been received.</confirmationMsg>
  </systemvariables>
```

APPENDIX I
CATS : SMS SERVICE

```
Imports System.ServiceProcess
Imports System.Threading

Public Class SMSService
    Inherits System.ServiceProcess.ServiceBase
    Private SMSServiceReader, UnsentSMSServiceReader As Thread
#Region " Component Designer generated code "

    Public Sub New()
        MyBase.New()

        ' This call is required by the Component Designer.
        InitializeComponent()

        ' Add any initialization after the InitializeComponent() call

    End Sub

    'UserService overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    ' The main entry point for the process
    <MTAThread()> _
    Shared Sub Main()
        Dim ServicesToRun() As System.ServiceProcess.ServiceBase

        ' More than one NT Service may run within the same process. To add
        ' another service to this process, change the following line to
        ' create a second service object. For example,
        '
        '   ServicesToRun = New System.ServiceProcess.ServiceBase () {New
Service1, New MySecondUserService}
        '
        ServicesToRun = New System.ServiceProcess.ServiceBase() {New
SMSService()}

        System.ServiceProcess.ServiceBase.Run(ServicesToRun)
    End Sub

    'Required by the Component Designer
    Private components As System.ComponentModel.IContainer

    ' NOTE: The following procedure is required by the Component Designer
    ' It can be modified using the Component Designer.
    ' Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
        components = New System.ComponentModel.Container()
        Me.ServiceName = "SMSService"
    End Sub

#End Region

    Protected Overrides Sub OnStart(ByVal args() As String)
        ' Add code here to start your service. This method should set
things
        ' in motion so your service can do its work.


        SMSServiceReader = New Thread(AddressOf Me.ReadFolder)
        UnsentSMSServiceReader = New Thread(AddressOf Me.ReadUnSentMsgs)
        Thread.Sleep(10000)
        SMSServiceReader.Start()
        UnsentSMSServiceReader.Start()
    End Sub

    Protected Overrides Sub OnStop()
        ' Add code here to perform any tear-down necessary to stop your
service.
        SMSServiceReader.Suspend()
    End Sub
    Protected Sub ReadFolder()
        While True
            SMSServiceReader.Sleep(60000)
            Dim msgsServices As New SMSMsgsServices.MsgsParser
            msgsServices.ParseInOutMessages(msgsServices.InOutEnum.InState)

msgsServices.ParseInOutMessages(msgsServices.InOutEnum.OutState)
            msgsServices = Nothing
        End While

    End Sub

    Protected Sub ReadUnSentMsgs()
        While True
            SMSServiceReader.Sleep(60000)
            Dim msgsServices As New SMSMsgsServices.MsgsParser
            msgsServices.ParseUnSentMsgs()
            msgsServices = Nothing
        End While

    End Sub
End Class
```

APPENDIX J
CATS : CRL-CATS PROCESSING

```
Imports System.IO
Imports System.Text.RegularExpressions

Public Class MsgsParser
#Region "General Part"
    Const LogSep = "*"
    Public Enum InOutEnum As Byte
        GeneralState = 0
        InState = 1
        OutState = 2
    End Enum

    Public Enum MsgsTypesEnum As Byte
        MsgGen = 0
        CarsMsg = 1
        EstateMsg = 2
        MaryMsg = 3
    End Enum

    Private msgTxt As String
    Private PhoneNo As String
    Private MsgDate As Date
    Private DataLayerObj As New DataLayer()

#End Region
#Region "General Integration Part"
#Region "constants"


    'Const xmlDocPath As String = "D:\Projects Source Code\New Pattern 
Recognishion\Pattern 
Recognishion\PatternsReciognishion\SMSMsgs\SystemVariables.xml"
    Const RelStr = "rel"
    Const firstWordStr = "firstword"
    Const SecondWordStr = "secondword"
    Const AttributeStr = "attribute"
    Const knowledgeStr = "knowledge"

    Const EnchoBasePath = "EnchoBasePath"
    Const EnchoAppPath = "EnchoAppPath"
    Const EnchoDicPath = "EnchoDicPath"
    Const EnchoRulesPath = "EnchoRulesPath"
    Const EnchoMsgReadFromPath = "EnchoMsgReadFromPath"
    Const OutMsgPathIn = "OutMsgPathIn"
    Const OutMsgPathOut = "OutMsgPathOut"
    Const BatchfilePath = "BatchfilePath"
    Const rootDirectory = "rootDirectory"
    Const options = "options"

    Private yea As String = GetSystemVariables("yea")
    Private fea As String = GetSystemVariables("fea")
    Private wan As String = GetSystemVariables("wan")
    Private _mod As String = GetSystemVariables("mod")
    Private mak As String = GetSystemVariables("mak")
    Private pri As String = GetSystemVariables("pri")
    Private mot As String = GetSystemVariables("mot")
    Private col As String = GetSystemVariables("col")
    Private cou As String = GetSystemVariables("cou")

    Private sal As String = GetSystemVariables("sal")
    Private pur As String = GetSystemVariables("pur")
    Private loc As String = GetSystemVariables("loc")
    Private are As String = GetSystemVariables("are")
    Private roo As String = GetSystemVariables("roo")
    Private cns As String = GetSystemVariables("cns")
    Private typ As String = GetSystemVariables("typ")
    Private flr As String = GetSystemVariables("flr")

    Private STRRegExp As String = GetSystemVariables("RegExp")

    Private dateFormates As Integer =
CInt(GetSystemVariables("dateFormates"))
    Private queryDatePeriod As Integer =
CInt(GetSystemVariables("queryDatePeriod"))
    Private queryUnSetDatePeriod As Integer =
CInt(GetSystemVariables("queryUnSetDatePeriod"))

    Private MsgLength As Integer = CInt(GetSystemVariables("MsgLength"))
    Private MsgLength_Ar As Integer =
CInt(GetSystemVariables("MsgLength_Ar"))


    Private strNextWords As String = GetSystemVariables("NextWords")
    Private strHelpWords As String = GetSystemVariables("HelpWords")
    Private strActivateWords As String =
GetSystemVariables("ActivateWords")
    Private strDeActivateWords As String =
GetSystemVariables("DeActivateWords")
    Private strHelpMessage As String = GetSystemVariables("HelpMessage")
    Private strNoDataFoundMessage As String =
GetSystemVariables("NoDataFoundMessage")
    Private strMsgCallerStrip As String =
GetSystemVariables("MsgCallerStrip")

    Public Shared errMsgPath As String = GetSystemVariables("errMsgPath")
    Public Shared logMsgPath As String = GetSystemVariables("logMsgPath")

    Private MsgStatmentRec As New MsgStatmentRec
#End Region
    Public Shared Function GetSystemVariables(ByVal StrID As String) As
String
        Dim xmlDoc As New System.Xml.XmlDocument
        'Dim arr() As String =
System.Environment.CurrentDirectory().Split(IO.Path.DirectorySeparatorChar)
        'ReDim Preserve arr(arr.Length_u45 ? 3)
        'Dim xmlDocPath As String = Join(arr,
IO.Path.DirectorySeparatorChar) + "\SMSMsgs\SystemVariables.xml"

        xmlDoc.Load(ConstantsClass.xmlDocPath)
        If xmlDoc.GetElementsByTagName(StrID).Count > 0 Then
            Return xmlDoc.GetElementsByTagName(StrID).ItemOf(0).InnerXml()
        Else
```

```
            Return ""
        End If
    End Function
    Private Function FormateMsgCaller(ByVal msgCaller As String) As String
        If msgCaller = Nothing Then
            Return ""
        Else
            Return msgCaller.Replace(strMsgCallerStrip, "")
        End If

    End Function
    Private Sub CreateTextFile(ByVal strPath As String, ByVal strTxt As
String)
        'Dim enco As New System.Text.ASCIIEncoding()
        Dim StrmWriter As New System.IO.StreamWriter(strPath, False,
StrmWriter.Encoding.GetEncoding(1256))
        StrmWriter.Write(strTxt)
        StrmWriter.Close()

    End Sub
    Public Shared Sub GetHttpResponse(ByVal sender As String, ByVal msgtxt
As String, Optional ByVal shortNumber As String = "")
        'Exit Sub
        Dim sendMsgURL As String = GetSystemVariables("httpResponse")

        'Dim server As String = System.Web.HttpServerUtility.
        Dim server As New Web.HttpUtility
        sendMsgURL += "?PhoneNumber=" + server.UrlEncode(sender) + "&Text="
+ server.UrlEncode(msgtxt)
        If shortNumber <> "" Then
            sendMsgURL += "&sender=" + server.UrlEncode(shortNumber)
        Else
            sendMsgURL += "&sender=" +
server.UrlEncode(GetSystemVariables("shortNumber"))
        End If
        Dim HttpWReq As Net.HttpWebRequest = _
        CType(Net.WebRequest.Create(sendMsgURL), Net.HttpWebRequest)
        Dim HttpWResp As Net.HttpWebResponse = _
           CType(HttpWReq.GetResponse(), Net.HttpWebResponse)
        HttpWResp.Close()

        ErrorHandler.OutLogHandler("Out Message: " + sender + ",  " +
msgtxt)
    End Sub

    Public Function DBNullCheck(ByVal val As Object) As Object
        If Convert.IsDBNull(val) Then
            Return Nothing
        Else
            Return val
        End If
    End Function
#End Region
#Region "Parse The In/Out Messages"
    Private Sub GetRequestedData(ByVal type As MsgsTypesEnum, ByRef
wantedBool As Boolean)
        If wantedBool Then
            Select Case type
                Case MsgsTypesEnum.CarsMsg
```

```
                    GetRequestedCarsData(wantedBool)
                Case MsgsTypesEnum.EstateMsg
                    GetRequestedEstateData(wantedBool)
                Case MsgsTypesEnum.MaryMsg
            End Select
        Else
            Try
                Me.GetHttpResponse(Me.PhoneNo,
GetSystemVariables("confirmationMsg"))
            Catch exp As Exception
                ErrorHandler.ErrorHandler(exp, "Confirmation Message
Error")
            End Try
        End If
    End Sub
    Private Sub GetNextMsg(ByVal MsgPhoneNo As String, ByVal phonesNOsParam
As String, ByVal Type As Integer, Optional ByVal MsgLength As Integer = 0)
        Dim phonesNOs As String = ""
        If Type = 0 Then '0 Next Statment, 1 Query
            Dim ds As DataSet = DataLayerObj.GetMsgLogDB(MsgPhoneNo)
            If ds.Tables.Count > 0 AndAlso ds.Tables(0).Rows.Count Then
                phonesNOs = ds.Tables(0).Rows(0)("msgTxt")
                MsgLength = ds.Tables(0).Rows(0)("MsgLength")
                If phonesNOs = "" Or phonesNOs Is Nothing Then
                    'Exit Sub
                    phonesNOs = ""
                End If
            Else
                'Exit Sub
                phonesNOs = ""
            End If
        Else
            phonesNOs = phonesNOsParam
        End If

        If (Trim(phonesNOs) = "" OrElse phonesNOs = Nothing) AndAlso
(MsgPhoneNo <> Nothing _
            AndAlso MsgPhoneNo <> "") And strNoDataFoundMessage <> "" Then
            GetHttpResponse(MsgPhoneNo, strNoDataFoundMessage)
        End If

        If MsgPhoneNo <> Nothing AndAlso phonesNOs <> Nothing AndAlso _
        MsgPhoneNo <> "" AndAlso phonesNOs <> "" Then

            Dim counter As Integer = phonesNOs.Length \ MsgLength
            Dim counter2 As Integer
            Dim strMsgTxt As String = ""
            Dim MsgArr As String() = phonesNOs.Split(vbNewLine)
            Dim strTmp As String
            For Each strTmp In MsgArr
                strTmp = strTmp.Replace(Chr(13), "").Replace(Chr(10), "")
                If (strMsgTxt + vbNewLine + strTmp).Length <= MsgLength
Then
                    If strMsgTxt = "" Then
                        strMsgTxt = strTmp
                    Else
                        strMsgTxt = strMsgTxt + vbNewLine + strTmp
                    End If
                Else
```

```
                        Exit For
                    End If
                Next

                If strMsgTxt <> "" Then
                    Dim logStr As String = ""
                    logStr = phonesNOs.Substring(strMsgTxt.Length)
                    Dim LogStrTmp As String =
strMsgTxt.Substring(strMsgTxt.LastIndexOf(LogSep))
                    Dim indx As Integer = LogStrTmp.IndexOf(vbNewLine)
                    If indx = -1 Then
                        indx = LogStrTmp.Length
                    End If

                    If logStr.Length > 0 AndAlso
logStr.IndexOf(LogStrTmp.Substring(0, indx)) < 0 Then
                        logStr = LogStrTmp.Substring(0, indx) + logStr
                        DataLayerObj.InsertMsgLogDB(MsgPhoneNo, logStr,
MsgLength)
                    Else
                        DataLayerObj.DeleteMsgLogDB(MsgPhoneNo)
                    End If
                    GetHttpResponse(MsgPhoneNo, strMsgTxt)
                End If
            End If
        End Sub
        Private Sub GetRequestedEstateData(ByRef wantedBool As Boolean)

            Dim stmtRec As MsgStatmentRec.EstateStatmentRecord
            'Dim msgsReceived As String
            Dim i, j As Integer
            For i = 0 To MsgStatmentRec.StatmentRecordArr.Count - 1
                Dim phonesNOs As String = ""
                stmtRec = MsgStatmentRec.StatmentRecordArr(i)
                stmtRec.MsgDate = Me.MsgDate
                stmtRec.MsgPhoneNo = Me.PhoneNo
                Dim ds1 As DataSet
                ds1 = DataLayerObj.GetEstateMsgs(stmtRec,
CInt(queryDatePeriod), 1, "")
                If ds1.Tables.Count > 0 AndAlso ds1.Tables(0).Rows.Count > 0
Then
                Else
                    ds1 = DataLayerObj.GetEstateMsgs(stmtRec, queryDatePeriod,
2, "")
                End If

                Dim sep As String
                Dim phonesNOs2 As String = ""
                For j = 0 To ds1.Tables(0).Rows.Count - 1
                    Dim strMake As String
                    If Not ds1.Tables(0).Rows(j)("location") Is Convert.DBNull
Then
                        strMake = ds1.Tables(0).Rows(j)("location")
                    ElseIf Not ds1.Tables(0).Rows(j)("knldgLoc") Is
Convert.DBNull Then
                        strMake = ds1.Tables(0).Rows(j)("knldgLoc")
                    End If
                    Dim strCaller As String =
FormateMsgCaller(ds1.Tables(0).Rows(j)("msgcaller"))
```

```
                Dim finalMsgTxt As String = strMake + "," + strCaller
                If strMake <> Nothing AndAlso strMake <> "" AndAlso
phonesNOs2.IndexOf(finalMsgTxt) < 0 Then
                    phonesNOs2 += sep + finalMsgTxt
                    If phonesNOs.IndexOf(strMake) < 0 Then
                        phonesNOs += sep + LogSep + strMake + vbNewLine +
strCaller
                    Else
                        phonesNOs =
phonesNOs.Insert(phonesNOs.IndexOf(strMake) + strMake.Length +
vbNewLine.Length, strCaller + sep)
                    End If
                    'msgsReceived += sep + ds1.Tables(0).Rows(j)("msgtxt")
+ "," + ds1.Tables(0).Rows(j)("msgcaller")
                End If
                sep = vbNewLine
            Next
            If stmtRec.MsgPhoneNo <> Nothing AndAlso
Trim(stmtRec.MsgPhoneNo) <> "" Then
                Me.PhoneNo = stmtRec.MsgPhoneNo
            End If
            If Me.PhoneNo <> Nothing AndAlso phonesNOs <> Nothing AndAlso _
            Me.PhoneNo <> "" AndAlso phonesNOs <> "" Then
                wantedBool = True
            Else
                wantedBool = False
            End If
            GetNextMsg(Me.PhoneNo, phonesNOs, 1, MsgLength_Ar)
            'If Me.PhoneNo <> Nothing AndAlso phonesNOs <> Nothing AndAlso
_
            'Me.PhoneNo <> "" AndAlso phonesNOs <> "" Then

            '    Dim counter As Integer = phonesNOs.Length \ MsgLength
            '    Dim counter2 As Integer
            '    Dim strMsgTxt As String = ""
            '    'For counter2 = 0 To counter
            '    '    If counter2 = counter Then
            '    '        strMsgTxt = phonesNOs.Substring(counter *
MsgLength, phonesNOs.Length - counter * MsgLength)
            '    '    Else
            '    '        strMsgTxt = phonesNOs.Substring(counter2 *
MsgLength, MsgLength)
            '    '    End If
            '    '    If strMsgTxt <> "" Then
            '    '        GetHttpResponse(Me.PhoneNo, strMsgTxt)
            '    '        Exit For
            '    '    End If
            '    'Next
            '    Dim MsgArr As String() = phonesNOs.Split(vbNewLine)
            '    Dim strTmp As String
            '    For Each strTmp In MsgArr
            '        strTmp = strTmp.Replace(Chr(13), "").Replace(Chr(10),
"")
            '        If (strMsgTxt + vbNewLine + strTmp).Length <=
MsgLength Then
            '            If strMsgTxt = "" Then
            '                strMsgTxt = strTmp
            '            Else
            '                strMsgTxt = strMsgTxt + vbNewLine + strTmp
            '            End If
            '        Else
            '            Exit For
            '        End If
            '    Next

            '    If strMsgTxt <> "" Then
            '        Dim logStr As String = ""
            '        logStr = phonesNOs.Substring(strMsgTxt.Length)
            '        'logStr = logStr.Replace(Chr(13), "").Replace(Chr(10),
"")
            '        Dim LogStrTmp As String =
strMsgTxt.Substring(strMsgTxt.LastIndexOf(LogSep))
            '        If logStr.Length > 0 AndAlso
logStr.IndexOf(LogStrTmp.Substring(0, LogStrTmp.IndexOf(vbNewLine))) < 0
Then
            '            logStr = LogStrTmp.Substring(0,
LogStrTmp.IndexOf(vbNewLine)) + logStr
            '            DataLayerObj.InsertMsgLogDB(Me.PhoneNo, logStr,
MsgLength)
            '        End If
            '        GetHttpResponse(Me.PhoneNo, strMsgTxt)
            '    End If
            'End If
        Next
    End Sub
    Private Sub GetRequestedCarsData(ByRef wantedBool As Boolean)
        'Const LogSep = "*"
        Dim stmtRec As MsgStatmentRec.StatmentRecordCars
        'Dim msgsReceived As String

        Dim i, j As Integer
        For i = 0 To MsgStatmentRec.StatmentRecordArr.Count - 1
            Dim phonesNOs As String = ""
            stmtRec = MsgStatmentRec.StatmentRecordArr(i)
            stmtRec.MsgDate = Me.MsgDate
            stmtRec.MsgPhoneNo = Me.PhoneNo
            Dim ds1 As DataSet
            ds1 = DataLayerObj.GetCarsMsgs(stmtRec, CInt(queryDatePeriod),
1, "")
            If ds1.Tables.Count > 0 AndAlso ds1.Tables(0).Rows.Count > 0
Then
            Else
                ds1 = DataLayerObj.GetCarsMsgs(stmtRec, queryDatePeriod, 2,
"")
            End If

            Dim sep As String
            Dim phonesNOs2 As String = ""
            If ds1.Tables.Count > 0 Then
                For j = 0 To ds1.Tables(0).Rows.Count - 1
                    Dim strMake As String
                    Dim strModel As String = ""
                    If Not ds1.Tables(0).Rows(j)("make") Is Convert.DBNull
Then
                        strMake = ds1.Tables(0).Rows(j)("make")
                    End If
                    Dim strCaller As String =
FormateMsgCaller(ds1.Tables(0).Rows(j)("msgcaller"))
```

```
                    If Not ds1.Tables(0).Rows(j)("model") Is Convert.DBNull
Then
                        strModel = ":" + ds1.Tables(0).Rows(j)("model")
                    End If
                    strCaller += strModel
                    If strMake = Nothing Then
                        strMake = ds1.Tables(0).Rows(j)("country")
                        If strMake = Nothing Then
                            strMake = ""
                        End If
                    End If
                    Dim finalMsgTxt As String = strMake + "," + strCaller
                    If phonesNOs2.IndexOf(finalMsgTxt) < 0 And strMake <>
"" Then
                        phonesNOs2 += sep + finalMsgTxt
                        If phonesNOs.IndexOf(strMake) < 0 Then
                            phonesNOs += sep + LogSep + strMake + vbNewLine
+ strCaller
                        Else
                            phonesNOs =
phonesNOs.Insert(phonesNOs.IndexOf(strMake) + strMake.Length +
vbNewLine.Length, strCaller + sep)
                        End If
                        'msgsReceived += sep +
ds1.Tables(0).Rows(j)("msgtxt") + "," + ds1.Tables(0).Rows(j)("msgcaller")
                    End If
                    sep = vbNewLine
                Next
            End If
            If stmtRec.MsgPhoneNo <> Nothing AndAlso
Trim(stmtRec.MsgPhoneNo) <> "" Then
                Me.PhoneNo = stmtRec.MsgPhoneNo
            End If
            If Me.PhoneNo <> Nothing AndAlso phonesNOs <> Nothing AndAlso _
            Me.PhoneNo <> "" AndAlso phonesNOs <> "" Then
                wantedBool = True
            Else
                wantedBool = False
            End If
            GetNextMsg(Me.PhoneNo, phonesNOs, 1, MsgLength)
            'If Me.PhoneNo <> Nothing AndAlso phonesNOs <> Nothing AndAlso
_
            'Me.PhoneNo <> "" AndAlso phonesNOs <> "" Then

            '    Dim counter As Integer = phonesNOs.Length \ MsgLength
            '    Dim counter2 As Integer
            '    Dim strMsgTxt As String = ""
            '    'For counter2 = 0 To counter
            '    '    If counter2 = counter Then
            '    '        strMsgTxt = phonesNOs.Substring(counter *
MsgLength, phonesNOs.Length - counter * MsgLength)
            '    '    Else
            '    '        strMsgTxt = phonesNOs.Substring(counter2 *
MsgLength, MsgLength)
            '    '    End If
            '    '    If strMsgTxt <> "" Then
            '    '        GetHttpResponse(Me.PhoneNo, strMsgTxt)
            '    '        Exit For
            '    '    End If
            '    'Next
            '    Dim MsgArr As String() = phonesNOs.Split(vbNewLine)
            '    Dim strTmp As String
            '    For Each strTmp In MsgArr
            '        strTmp = strTmp.Replace(Chr(13), "").Replace(Chr(10),
"")
            '        If (strMsgTxt + vbNewLine + strTmp).Length <=
MsgLength Then
            '            If strMsgTxt = "" Then
            '                strMsgTxt = strTmp
            '            Else
            '                strMsgTxt = strMsgTxt + vbNewLine + strTmp
            '            End If
            '        Else
            '            Exit For
            '        End If
            '    Next

            '    If strMsgTxt <> "" Then
            '        Dim logStr As String = ""
            '        logStr = phonesNOs.Substring(strMsgTxt.Length)
            '        'logStr = logStr.Replace(Chr(13), "").Replace(Chr(10),
"")
            '        Dim LogStrTmp As String =
strMsgTxt.Substring(strMsgTxt.LastIndexOf(LogSep))
            '        If logStr.Length > 0 AndAlso
logStr.IndexOf(LogStrTmp.Substring(0, LogStrTmp.IndexOf(vbNewLine))) < 0
Then
            '            logStr = LogStrTmp.Substring(0,
LogStrTmp.IndexOf(vbNewLine)) + logStr
            '            DataLayerObj.InsertMsgLogDB(Me.PhoneNo, logStr,
MsgLength)
            '        End If
            '        GetHttpResponse(Me.PhoneNo, strMsgTxt)
            '    End If
            'End If
        Next
    End Sub

    Private Sub AssignVal(ByRef Val1 As String, ByVal Val2 As String,
Optional ByVal notAccumelaticeBool As Boolean = False)
        Val2 = Val2.Replace(Chr(10), "")
        Val2 = Val2.Replace(Chr(13), "")
        Val2 = Val2.Replace(Chr(9), "")
        Val2 = Trim(Val2)
        If (Val1 = Nothing) OrElse (Val1 = "") Then
            Val1 = Val2
        Else
            If notAccumelaticeBool Then
            Else
                If Val1.IndexOf(Val2) = -1 Then
                    Val1 = Val1 + "," + Val2
                End If

            End If

        End If
    End Sub
    Private Sub parseknowledgeBase(ByVal knldgStr As String, ByVal MsgType
As MsgsTypesEnum)
        Dim arrKnowledge As String() = knldgStr.Split(",")
        Dim strTmp As String
        For Each strTmp In arrKnowledge
            If strTmp.Length > 0 Then

                Select Case MsgType
                    Case MsgsTypesEnum.CarsMsg
                        Select Case strTmp.Substring(0, 3)
                            Case cou

AssignVal(MsgStatmentRec.CarsRecordObj.country, strTmp.Replace("country<",
""))
                            Case mak

AssignVal(MsgStatmentRec.CarsRecordObj.make, strTmp.Replace("make<", ""),
True)
                        End Select
                    Case MsgsTypesEnum.EstateMsg
                        Select Case strTmp.Substring(0, 3)
                            Case are

AssignVal(MsgStatmentRec.EstateStatmentRecordObj.knldgLoc,
strTmp.Replace("area<", ""), True)
                        End Select
                    Case MsgsTypesEnum.MaryMsg
                End Select
            End If
        Next

    End Sub
    Private Function ParseMsgLine(ByVal str As String) As String
        Dim rlStr, fwStr, swStr, KnldgStr, attStr As String
        KnldgStr = ""
        str = Trim(str)
        If str.ToLower = "[/s]" Then
            Return "end"
        End If
        'comment line
        If str.Length = 0 OrElse str.Chars(0) = ";" OrElse str.ToLower =
"[s]" _
        OrElse str.ToLower = "[/s]" OrElse _
        str.Chars(0) = "[" Then
            Return ""
        Else
            'typeStr = GetToken(str, "", "(")
            'maincat = GetToken(str, "(", ":")
            'valStr = GetToken(str, ",", ":")
            Dim matchObj As Match

            Dim regExp As New Regex(STRRegExp.Replace("&lt;",
"<").Replace("&gt;", ">"),
System.Text.RegularExpressions.RegexOptions.Compiled)
            matchObj = regExp.Match(str)
            fwStr = Trim(matchObj.Groups(firstWordStr).Value.ToLower())
            rlStr = Trim(matchObj.Groups(RelStr).Value.ToLower())
            swStr = Trim(matchObj.Groups(SecondWordStr).Value.ToLower())
            Dim capture As Text.RegularExpressions.Capture
            For Each capture In matchObj.Groups(knowledgeStr).Captures
                KnldgStr = KnldgStr + Trim(capture.Value.ToLower())
            Next
            attStr = Trim(matchObj.Groups(AttributeStr).Value.ToLower())
            Select Case fwStr
                Case "flat", "roof", "cabana", "villa", "land", "shop",
"office", "orangefarm", "building", "farm", "smallhouse", "storageroom",
"floorfromvilla"

MsgStatmentRec.InitStatmentRecordObj(MsgsTypesEnum.EstateMsg)
                    parseknowledgeBase(KnldgStr, MsgsTypesEnum.EstateMsg)

AssignVal(MsgStatmentRec.EstateStatmentRecordObj.estateType, fwStr, True)
                    Select Case rlStr
                        Case wan

AssignVal(MsgStatmentRec.EstateStatmentRecordObj.wanted, swStr, True)
                        Case sal

AssignVal(MsgStatmentRec.EstateStatmentRecordObj.sal, swStr, True)
                        Case pur

AssignVal(MsgStatmentRec.EstateStatmentRecordObj.pur, swStr, True)
                            'If MsgStatmentRec.EstateStatmentRecordObj.pur
= "for rent" Then

'AssignVal(MsgStatmentRec.EstateStatmentRecordObj.wanted, "wanted", True)
                            'End If
                        Case loc
                            If attStr = ".@area" Then

AssignVal(MsgStatmentRec.EstateStatmentRecordObj.knldgLoc, swStr)
                            Else

AssignVal(MsgStatmentRec.EstateStatmentRecordObj.loc, swStr)
                            End If
                        Case are
                            If attStr.IndexOf("@donem") >= 0 Then
                                Try
                                    swStr = CStr(1000 * CDbl(swStr))
                                Catch e As Exception
                                End Try
                            Else
                            End If

AssignVal(MsgStatmentRec.EstateStatmentRecordObj.areaAttribute, attStr,
True)

AssignVal(MsgStatmentRec.EstateStatmentRecordObj.are, swStr, True)
                        Case roo

AssignVal(MsgStatmentRec.EstateStatmentRecordObj.roo, swStr, True)
                        Case cns

AssignVal(MsgStatmentRec.EstateStatmentRecordObj.cns, swStr, True)
                        Case typ

AssignVal(MsgStatmentRec.EstateStatmentRecordObj.typ, swStr, True)
                        Case flr
```

```
AssignVal(MsgStatmentRec.EstateStatmentRecordObj.flr, swStr, True)
                        Case fea

AssignVal(MsgStatmentRec.EstateStatmentRecordObj.feature, swStr)
                        Case pri
                            If attStr.IndexOf("@alf") >= 0 Then
                                Try
                                    swStr = CStr(1000 * CDbl(swStr))
                                Catch e As Exception
                                End Try
                            End If
                            If attStr.IndexOf("@year") >= 0 Then
                                Try
                                    swStr = CStr(CDbl(swStr) / 12)
                                Catch e As Exception
                                End Try
                            End If
                            If attStr.IndexOf("@perdonem") >= 0 Then
                                Try
                                    swStr = CStr(1000 * CDbl(swStr))
                                Catch e As Exception
                                End Try
                            End If

AssignVal(MsgStatmentRec.EstateStatmentRecordObj.price, swStr, True)

AssignVal(MsgStatmentRec.EstateStatmentRecordObj.priceAttribute, attStr,
True)
                    End Select
                Case "saloon", "bus", "van", "pickup"

MsgStatmentRec.InitStatmentRecordObj(MsgsTypesEnum.CarsMsg)
                    parseknowledgeBase(KnldgStr, MsgsTypesEnum.CarsMsg)
                    AssignVal(MsgStatmentRec.CarsRecordObj.maincat, fwStr,
True)
                    Select Case rlStr
                        Case yea
                            AssignVal(MsgStatmentRec.CarsRecordObj.year,
swStr, True)
                        Case fea
                            AssignVal(MsgStatmentRec.CarsRecordObj.feature,
swStr)
                        Case wan
                            AssignVal(MsgStatmentRec.CarsRecordObj.wanted,
swStr, True)
                        Case _mod
                            'If MsgStatmentRec.CarsRecordObj.make = "" And
MsgStatmentRec.CarsRecordObj.model <> "" Then

'MsgStatmentRec.NewElement(MsgStatmentRec.CarsRecordObj)

'MsgStatmentRec.InitStatmentRecordObj(MsgsTypesEnum.CarsMsg)
                            'End If
                            AssignVal(MsgStatmentRec.CarsRecordObj.model,
swStr, True)
                        Case mak
                            'If MsgStatmentRec.CarsRecordObj.make <> ""
Then
```

```
'MsgStatmentRec.NewElement(MsgStatmentRec.CarsRecordObj)
                            'MsgStatmentRec.InitStatmentRecordObj()
                            'End If
                            AssignVal(MsgStatmentRec.CarsRecordObj.make,
swStr, True)
                        Case pri
                            If attStr.IndexOf("@alf") >= 0 Then
                                Try
                                    swStr = CStr(1000 * CDbl(swStr))
                                Catch e As Exception
                                End Try
                            End If
                            AssignVal(MsgStatmentRec.CarsRecordObj.price,
swStr, True)
                        Case mot
                            AssignVal(MsgStatmentRec.CarsRecordObj.motor,
swStr)
                        Case col
                            AssignVal(MsgStatmentRec.CarsRecordObj.color,
swStr)
                        Case cou
                            AssignVal(MsgStatmentRec.CarsRecordObj.country,
swStr)
                    End Select
            End Select
        End If
    End Function
    Private Sub ParseMsgStream(ByVal MsgStreamReader As IO.StreamReader,
ByVal MsgFileStr As String, ByVal InOut As InOutEnum)
        MsgStatmentRec.InitStatmentRecordArr()
        MsgStatmentRec.InitStatmentRecordObj()
        While MsgStreamReader.Peek >= 0
            If ParseMsgLine(MsgStreamReader.ReadLine()) = "end" Then
                Exit While
            End If
        End While

        If Not MsgStatmentRec.CarsRecordObj Is Nothing Then
            MsgStatmentRec.NewElement(MsgStatmentRec.CarsRecordObj)
        ElseIf Not MsgStatmentRec.EstateStatmentRecordObj Is Nothing Then

MsgStatmentRec.NewElement(MsgStatmentRec.EstateStatmentRecordObj)
        End If

        MsgFileStr = IO.Path.GetFileNameWithoutExtension(MsgFileStr)
        'Dim ds As DataSet =
DataLayerObj.ExecuteSelecetCommands(DataLayerObj.DB_SLC_MSGSID + "strGuid=
'" + MsgFileStr + "'")
        Dim ds As DataSet = DataLayerObj.GetMsgDB(MsgFileStr)
        If ds.Tables.Count > 0 AndAlso ds.Tables(0).Rows.Count > 0 Then
            Me.MsgDate = DBNullCheck(ds.Tables(0).Rows(0)("msgDate"))
            Me.PhoneNo = DBNullCheck(ds.Tables(0).Rows(0)("msgCaller"))
            Me.msgTxt = DBNullCheck(ds.Tables(0).Rows(0)("msgTxt"))
        End If
        Dim stmtRec As MsgStatmentRec.StatmentRecordGen
        Dim i As Integer
        Dim wantedBool As Boolean
        For i = 0 To MsgStatmentRec.StatmentRecordArr.Count - 1
            stmtRec = MsgStatmentRec.StatmentRecordArr(i)
            If Trim(stmtRec.wanted) <> "" Or InOut = InOutEnum.OutState _
            Or wantedBool Then
                stmtRec.wantedBool = True
                wantedBool = True
            Else
                stmtRec.wantedBool = False
            End If
            stmtRec.fileCUID = MsgFileStr
            stmtRec.MsgDate = MsgDate
            stmtRec.MsgPhoneNo = PhoneNo
            stmtRec.MsgTxt = msgTxt
            If ds.Tables.Count > 0 AndAlso ds.Tables(0).Rows.Count > 0
AndAlso i = 0 Then
                If stmtRec.GetType Is
GetType(MsgStatmentRec.StatmentRecordCars) Then
                    DataLayerObj.InsertCarsDB(CType(stmtRec,
MsgStatmentRec.StatmentRecordCars))
                    GetRequestedData(MsgsTypesEnum.CarsMsg, wantedBool)
                    DataLayerObj.SetSendFlag(stmtRec.fileCUID, wantedBool)
                ElseIf stmtRec.GetType Is
GetType(MsgStatmentRec.EstateStatmentRecord) Then
                    DataLayerObj.InsertEstateDB(CType(stmtRec,
MsgStatmentRec.EstateStatmentRecord))
                    GetRequestedData(MsgsTypesEnum.EstateMsg, wantedBool)
                    DataLayerObj.SetSendFlag(stmtRec.fileCUID, wantedBool)
                End If
                Exit For  'Case more than one record not implemented
            End If
        Next

    End Sub
    Public Sub ParseInOutMessages(ByVal InOut As InOutEnum)
        Dim OutMsgPath As String
        Dim MsgsFolderPath As String
        Dim MsgsFolder As IO.Directory
        If InOut = InOutEnum.InState Then
            OutMsgPath = OutMsgPathIn
        Else
            OutMsgPath = OutMsgPathOut
        End If
        MsgsFolderPath = GetSystemVariables(BatchfilePath) +
GetSystemVariables(OutMsgPath)
        'MsgsFolder.SetCurrentDirectory(MsgsFolderPath)
        Dim MsgFileStr As String
        For Each MsgFileStr In MsgsFolder.GetFiles(MsgsFolderPath)
            Try
                Dim MsgStreamReader As New IO.StreamReader(MsgFileStr,
System.Text.Encoding.GetEncoding(_u50 ?56))
                Try
                    ParseMsgStream(MsgStreamReader, MsgFileStr, InOut)
                Catch e As Exception
                    ErrorHandler.ErrorHandler(e, Me.msgTxt)
                End Try
                MsgStreamReader.Close()
                IO.File.Delete(MsgFileStr)
            Catch e As Exception
                ErrorHandler.ErrorHandler(e, Me.msgTxt)
            End Try
        Next
    End Sub
    Public Sub HandleUnSentCarsMsgs(ByVal tbl As DataTable)
        Dim rw As DataRow
        For Each rw In tbl.Rows
            MsgStatmentRec.InitStatmentRecordArr()
            MsgStatmentRec.InitStatmentRecordObj(MsgsTypesEnum.CarsMsg)
            Dim wantedBool As Boolean = True
            Me.MsgDate = DBNullCheck(rw("msgDate"))
            Me.PhoneNo = DBNullCheck(rw("msgCaller"))
            MsgStatmentRec.CarsRecordObj.MsgDate =
DBNullCheck(rw("msgDate"))
            MsgStatmentRec.CarsRecordObj.MsgPhoneNo =
DBNullCheck(rw("msgCaller"))

            MsgStatmentRec.CarsRecordObj.country =
DBNullCheck(rw("country"))
            MsgStatmentRec.CarsRecordObj.maincat =
DBNullCheck(rw("maincat"))
            MsgStatmentRec.CarsRecordObj.make = DBNullCheck(rw("make"))
            MsgStatmentRec.CarsRecordObj.model = DBNullCheck(rw("model"))

            MsgStatmentRec.CarsRecordObj.feature =
DBNullCheck(rw("feature"))
            MsgStatmentRec.CarsRecordObj.fileCUID =
DBNullCheck(rw("strGUID"))
            MsgStatmentRec.CarsRecordObj.motor = DBNullCheck(rw("motor"))

            MsgStatmentRec.CarsRecordObj.year = DBNullCheck(rw("makeyear"))
            MsgStatmentRec.CarsRecordObj.color = DBNullCheck(rw("color"))
            MsgStatmentRec.NewElement(MsgStatmentRec.CarsRecordObj)

            GetRequestedData(MsgsTypesEnum.CarsMsg, wantedBool)

DataLayerObj.SetSendFlag(MsgStatmentRec.StatmentRecordArr(0).fileCUID,
wantedBool)
        Next

    End Sub
    Public Sub HandleUnSentRealEstateMsgs(ByVal tbl As DataTable)
        Dim rw As DataRow
        For Each rw In tbl.Rows
            MsgStatmentRec.InitStatmentRecordArr()
            MsgStatmentRec.InitStatmentRecordObj(MsgsTypesEnum.EstateMsg)
            Dim wantedBool As Boolean = True

            MsgStatmentRec.EstateStatmentRecordObj.MsgDate =
DBNullCheck(rw("msgDate"))
            MsgStatmentRec.EstateStatmentRecordObj.MsgPhoneNo =
DBNullCheck(rw("msgCaller"))
            Me.MsgDate = DBNullCheck(rw("msgDate"))
            Me.PhoneNo = DBNullCheck(rw("msgCaller"))
            MsgStatmentRec.EstateStatmentRecordObj.are =
DBNullCheck(rw("Area"))
            MsgStatmentRec.EstateStatmentRecordObj.cns =
DBNullCheck(rw("ConsistOf"))
            MsgStatmentRec.EstateStatmentRecordObj.estateType =
DBNullCheck(rw("estateType"))
```

```
            MsgStatmentRec.EstateStatmentRecordObj.feature =
DBNullCheck(rw("feature"))
            MsgStatmentRec.EstateStatmentRecordObj.fileCUID =
DBNullCheck(rw("strGUID"))
            MsgStatmentRec.EstateStatmentRecordObj.flr =
DBNullCheck(rw("Floor"))
            MsgStatmentRec.EstateStatmentRecordObj.knldgLoc =
DBNullCheck(rw("knldgLoc"))
            MsgStatmentRec.EstateStatmentRecordObj.loc =
DBNullCheck(rw("Location"))
            MsgStatmentRec.EstateStatmentRecordObj.price =
DBNullCheck(rw("Price"))
            MsgStatmentRec.EstateStatmentRecordObj.pur =
DBNullCheck(rw("Purpose"))
            MsgStatmentRec.EstateStatmentRecordObj.roo =
DBNullCheck(rw("Room"))
            MsgStatmentRec.EstateStatmentRecordObj.typ =
DBNullCheck(rw("Type"))


MsgStatmentRec.NewElement(MsgStatmentRec.EstateStatmentRecordObj)

            GetRequestedData(MsgsTypesEnum.EstateMsg, wantedBool)

DataLayerObj.SetSendFlag(MsgStatmentRec.StatmentRecordArr(0).fileCUID,
wantedBool)
        Next
    End Sub
    Public Sub ParseUnSentMsgs()
        Try
            strNoDataFoundMessage = ""
            Dim ds As DataSet =
DataLayerObj.GetUnSentMsgsDB(queryUnSetDatePeriod)
            Dim tbl As DataTable
            If ds.Tables.Count > 0 Then
                For Each tbl In ds.Tables
                    If tbl.Rows.Count > 0 Then
                        If tbl.Columns.Contains("location") Then
                            HandleUnSentRealEstateMsgs(tbl)
                        Else
                            HandleUnSentCarsMsgs(tbl)
                        End If
                    End If
                Next
            End If
        Catch e As Exception
            ErrorHandler.ErrorHandler(e, Me.msgTxt)
        End Try
    End Sub
#End Region
#Region "Integrate with Enco System"

    Private Sub CreateBatch(ByVal FileGuid As String, Optional ByVal InOut
As InOutEnum = InOutEnum.InState)

        Dim OutMsgPath As String
        If InOut = InOutEnum.InState Then
            OutMsgPath = OutMsgPathIn
        Else
```

```
            OutMsgPath = OutMsgPathOut
        End If
        Dim NwLine As String = Chr(13) + Chr(10)
        Dim PathStr As String

        'Dim OutputFileName As String = Me.PhoneNo + "_" +
CStr(Me.MsgDate.ToOADate())
        Dim OutputFileName As String = FileGuid
        Dim BatchFileContent As String = "@echo off" + NwLine

        PathStr = GetSystemVariables(BatchfilePath) + FileGuid
        BatchFileContent += GetSystemVariables(rootDirectory) + NwLine
        BatchFileContent += "cd  " + GetSystemVariables(EnchoBasePath) +
NwLine
        BatchFileContent += " " + GetSystemVariables(EnchoAppPath)
        BatchFileContent += " " + GetSystemVariables(EnchoDicPath)
        BatchFileContent += " " + GetSystemVariables(EnchoRulesPath)
        BatchFileContent += " " + FileGuid + ".txt"
        BatchFileContent += " " + GetSystemVariables(OutMsgPath) +
OutputFileName + ".txt"
        BatchFileContent += " " + GetSystemVariables(options)


        CreateTextFile(PathStr + ".bat", BatchFileContent)
        CreateTextFile(GetSystemVariables(BatchfilePath) +
GetSystemVariables(OutMsgPath) + OutputFileName + ".txt", "")
        CreateTextFile(PathStr + ".txt", Me.msgTxt)

        Shell(PathStr + ".bat", AppWinStyle.Hide, True)

        File.Delete(PathStr + ".bat")
        File.Delete(PathStr + ".txt")
        'Return FileGuid
    End Sub

#End Region
#Region "Class Constructor"
    Sub PhraseExists(ByVal PhraseStr As String, ByVal MsgArr As String(),
ByRef BoolVar As Boolean, ByRef Index As Integer)
        BoolVar = False
        Index = 0
        Dim strTmp, StrTmp2 As String
        For Each strTmp In MsgArr
            strTmp = strTmp.ToLower()
            For Each StrTmp2 In PhraseStr.Split(",")
                StrTmp2 = StrTmp2.ToLower()
                If strTmp.IndexOf(StrTmp2) >= 0 Then
                    BoolVar = True
                    Exit For
                End If
            Next
            If BoolVar Then
                Exit For
            Else
                Index = Index + 1
            End If
        Next
    End Sub
    Sub ActivateDeactivateMsgs(ByVal PhoneNumber As String, ByVal
ActDeActFlag As Integer)
        DataLayerObj.ActivateDeactivate(PhoneNumber, ActDeActFlag)
    End Sub
    Public Sub New()

    End Sub
    Public Sub New(ByVal msgTxtTmp As String, ByVal PhoneNoTmp As String,
ByVal MsgDateTmp As Date, Optional ByVal InOut As InOutEnum =
InOutEnum.InState)
        Me.msgTxt = msgTxtTmp
        msgTxt = Trim(msgTxt)
        Dim arr As String() = msgTxt.Split(" ")
        Dim strTmp, StrTmp2 As String
        Dim Index As Integer = 0
        Dim boolVar As Boolean
        ' Next
        PhraseExists(strNextWords, arr, boolVar, Index)
        'For Each strTmp In arr
        '    For Each StrTmp2 In
        'strNextWords.Split(",")
        '        If strTmp.IndexOf(StrTmp2) >= 0 Then
        '            boolVar = True
        '            Exit For
        '        End If
        '    Next
        '    If boolVar Then
        '        Exit For
        '    Else
        '        Index = Index + 1
        '   End If
        'Next
        If boolVar And Index < 4 Then
            GetNextMsg(PhoneNoTmp, "", 0)
            Exit Sub
        End If
        'Help
        PhraseExists(strHelpWords, arr, boolVar, Index)
        If boolVar And Index < 4 Then
            GetHttpResponse(PhoneNoTmp, strHelpMessage)
            Exit Sub
        End If
        'Activate
        PhraseExists(strActivateWords, arr, boolVar, Index)
        If boolVar And Index < 4 Then
            ActivateDeactivateMsgs(PhoneNoTmp, 0)
            Exit Sub
        End If

        'DeActivate
        PhraseExists(strDeActivateWords, arr, boolVar, Index)
        If boolVar And Index < 4 Then
            ActivateDeactivateMsgs(PhoneNoTmp, 1)
            Exit Sub
        End If

        Me.PhoneNo = PhoneNoTmp
        Me.MsgDate = MsgDateTmp
        Try
```

```
            Dim FileGuid As String = System.Guid.NewGuid().ToString()
            MsgStatmentRec.StatmentRecordGenObj.MsgDate = MsgDate
            MsgStatmentRec.StatmentRecordGenObj.MsgPhoneNo = PhoneNo
            MsgStatmentRec.StatmentRecordGenObj.MsgTxt = msgTxt
            MsgStatmentRec.StatmentRecordGenObj.fileCUID = FileGuid
            MsgStatmentRec.StatmentRecordGenObj.wantedBool = True
            MsgStatmentRec.NewElement(MsgStatmentRec.StatmentRecordGenObj)
            CreateBatch(FileGuid, InOut)
            If InOut = InOutEnum.InState Then
                Dim stmtRec As MsgStatmentRec.StatmentRecordGen
                For Each stmtRec In MsgStatmentRec.StatmentRecordArr
                    DataLayerObj.InsertMsgDB(stmtRec)
                Next
            End If
        Catch e As Exception
            ErrorHandler.ErrorHandler(e, Me.msgTxt)
        End Try
    End Sub

#End Region
End Class

#Region "Structure class"
Public Class MsgStatmentRec
#Region "shared object"
    Class StatmentRecordGen
        Public sal As String
        Public MsgTxt As String
        Public MsgPhoneNo As String
        Public fileCUID As String
        Public feature As String 'Gear,condition..
        Public price As String
        Public wanted As String
        Public wantedBool As Boolean
        Public MsgDate As Date

        'Price Att
        Public priceAttribute As String
    End Class
    Class StatmentRecordCars
        Inherits StatmentRecordGen
        Public maincat As String 'bus.saloon
        Public make As String  'Honda
        Public model As String  'Civic
        Public country As String 'Japan
        Public color As String  'Metalic
        Public year As String
        Public motor As String
    End Class
    Class EstateStatmentRecord
        Inherits StatmentRecordGen
        Public estateType As String  'flat,roof,..
        Public pur As String
        Public loc As String
        Public are As String
        Public roo As String
        Public cns As String
        Public typ As String  'صناعي زراعي
        Public flr As String
        'loc attributes and knowledge
        Public knldgLoc As String  'mAmman,...
        Public LocAttribute As String '.@dist,.@town
        'area atttributes
        Public areaAttribute As String  'donem,  meter

    End Class

    Public StatmentRecordArr As New System.Collections.ArrayList()
    Public StatmentRecordGenObj As StatmentRecordGen
    Public CarsRecordObj As StatmentRecordCars
    Public EstateStatmentRecordObj As EstateStatmentRecord
#End Region
#Region "Methods"
    Public Sub New(Optional ByVal ModuleType As MsgsParser.MsgsTypesEnum =
MsgsParser.MsgsTypesEnum.MsgGen)
        Select Case ModuleType
            Case MsgsParser.MsgsTypesEnum.CarsMsg
                CarsRecordObj = New StatmentRecordCars()
            Case MsgsParser.MsgsTypesEnum.EstateMsg
                EstateStatmentRecordObj = New EstateStatmentRecord()
            Case MsgsParser.MsgsTypesEnum.MaryMsg
            Case Else
                StatmentRecordGenObj = New StatmentRecordGen()
        End Select
    End Sub
    Public Sub InitStatmentRecordObj(Optional ByVal ModuleType As
MsgsParser.MsgsTypesEnum = MsgsParser.MsgsTypesEnum.MsgGen)
        Select Case ModuleType
            Case MsgsParser.MsgsTypesEnum.CarsMsg
                If CarsRecordObj Is Nothing Then
                    CarsRecordObj = New StatmentRecordCars
                End If
            Case MsgsParser.MsgsTypesEnum.EstateMsg
                    If EstateStatmentRecordObj Is Nothing Then
                        EstateStatmentRecordObj = New EstateStatmentRecord
                    End If
            Case MsgsParser.MsgsTypesEnum.MaryMsg
            Case Else
                    'If StatmentRecordGenObj Is Nothing Then
                StatmentRecordGenObj = New StatmentRecordGen
                CarsRecordObj = Nothing
                EstateStatmentRecordObj = Nothing
                'If Not CarsRecordObj Is Nothing Then
                '    CarsRecordObj = New StatmentRecordCars
                'End If
                'If Not EstateStatmentRecordObj Is Nothing Then
                '    EstateStatmentRecordObj = New EstateStatmentRecord
                'End If
                'End If
        End Select

    End Sub
    Public Sub InitStatmentRecordArr()
        While StatmentRecordArr.Count > 0
            StatmentRecordArr.RemoveAt(0)
        End While
    End Sub
    Public Sub NewElement(ByVal stmntRec As StatmentRecordGen)
```

```
        StatmentRecordArr.Add(stmntRec)
    End Sub
#End Region
End Class
#End Region
#Region "Error Handle class"
Public Class ErrorHandler
    'Log Informtation
    Private Shared logSize As Integer =
CInt(MsgsParser.GetSystemVariables("logSize"))
    Private Shared logPath As String =
MsgsParser.GetSystemVariables("logPath")
    Shared processName As String =
System.Diagnostics.Process.GetCurrentProcess.ProcessName()
    Shared traceErrorLstinerName = processName + "Error_"
    Shared traceLogLstinerName = processName + "Log_"
    Shared traceOutLogLstinerName = processName + "OutMsgs_"

    Private Shared Sub CreateTrace(ByVal traceLstinerName As String, ByVal
Path As String)

        Dim filePath As String = Path + traceLstinerName + "{0}" + ".txt"
        ' Creates when no Listner exists
        If IsNothing(Trace.Listeners(traceLstinerName)) Then
            filePath = String.Format(filePath, "1")
            Trace.Listeners.Add(New TextWriterTraceListener(filePath,
traceLstinerName))
        End If

    End Sub

    Private Shared Sub CreateLog(ByVal traceLstinerName As String, ByVal
Path As String)

        CreateTrace(traceLstinerName, Path)

        Dim filePath As String = Path + traceLstinerName + "{0}" + ".txt"
        Dim txtWriter As TextWriterTraceListener =
Trace.Listeners(traceLstinerName)
        Dim sw As IO.StreamWriter = txtWriter.Writer
        ' check if the existing log is in the normal size
        If sw.BaseStream.Length > logSize Then
            Dim fs As IO.FileStream = sw.BaseStream
            Dim fileNameArr As String() =
IO.Path.GetFileNameWithoutExtension(fs.Name()).Split("_")
            Dim strLogNumber As String =
CStr(CInt(fileNameArr(fileNameArr.Length - 1)) + 1)
            filePath = String.Format(filePath, strLogNumber)
            txtWriter = New TextWriterTraceListener(filePath,
traceLstinerName)
            Trace.Listeners(traceLstinerName).Close()
            Trace.Listeners.Remove(traceLstinerName)
            Trace.Listeners.Add(txtWriter)
            CreateLog(traceLstinerName, Path)
        End If

    End Sub
    Public Shared Sub ErrorHandler(ByVal e As Exception, ByVal msgtxt As
String)

        CreateLog(traceErrorLstinerName, logPath)

        Trace.AutoFlush = True
        Trace.Indent()
        Trace.Listeners(traceErrorLstinerName).WriteLine(Now.ToString() + "
" + msgtxt)
        Trace.Listeners(traceErrorLstinerName).WriteLine(e.Source)
        Trace.Listeners(traceErrorLstinerName).WriteLine(e.Message)
        Trace.Listeners(traceErrorLstinerName).WriteLine("")
        Trace.Listeners(traceErrorLstinerName).Flush()
        Trace.Unindent()
    End Sub

    Public Shared Sub LogHandler(ByVal msgtxt As String)

        CreateLog(traceLogLstinerName, logPath)

        Trace.AutoFlush = True
        Trace.Indent()
        Trace.Listeners(traceLogLstinerName).WriteLine(Now.ToString() + "
" + msgtxt)
        Trace.Listeners(traceLogLstinerName).WriteLine("")
        Trace.Listeners(traceLogLstinerName).Flush()
        Trace.Unindent()
    End Sub

    Public Shared Sub OutLogHandler(ByVal msgtxt As String)

        CreateLog(traceOutLogLstinerName, logPath)

        Trace.AutoFlush = True
        Trace.Indent()
        Trace.Listeners(traceOutLogLstinerName).WriteLine(Now.ToString() +
"   " + msgtxt)
        Trace.Listeners(traceOutLogLstinerName).WriteLine("")
        Trace.Listeners(traceOutLogLstinerName).Flush()
        Trace.Unindent()
    End Sub

    Protected Overrides Sub Finalize()
        MyBase.Finalize()
        If Not IsNothing(Trace.Listeners(traceErrorLstinerName)) Then
            Trace.Listeners(traceErrorLstinerName).Close()
        End If
        If Not IsNothing(Trace.Listeners(traceLogLstinerName)) Then
            Trace.Listeners(traceLogLstinerName).Close()
        End If
        If Not IsNothing(Trace.Listeners(traceOutLogLstinerName)) Then
            Trace.Listeners(traceOutLogLstinerName).Close()
        End If
    End Sub

End Class
#End Region
```

APPENDIX K
CATS : DATALAYER

```
Public Class DataLayer
#Region "Declaration General Vaiables"
    'General
    Private Const DB_ADD_Msg_SP = "ST_AddMsg"
    Private Const DB_Get_Msg_SP = "ST_GetMsg"
    'Cars
    Private Const DB_ADD_Car_SP = "ST_AddCarMsg"
    Private Const DB_Update_Car_SP = "ST_UpdateCarMsg"
    Private Const DB_Get_Cars_SP = "ST_GetCarMsgs"
    'Log
    Private Const DB_ADD_Msgs_Log = "ST_AddMsgsLog"
    Private Const DB_Get_Msgs_Log = "ST_GetMsgsLog"
    Private Const DB_Delete_Msgs_Log = "ST_DeleteMsgsLog"
    'Real Estate
    Private Const DB_ADD_Estate_SP = "ST_AddRealEstate"
    Private Const DB_Update_Estate_SP = "ST_UpdateRealEstate"
    Private Const DB_Get_Estate_SP = "ST_GetRealEstateMsgs"
    'Activate Deactivate and Send Flag
    Private Const DB_Activate_Deactivate_SP = "ST_ActivateDeactivate"
    Private Const DB_Set_Send_Flag_SP = "ST_SetSendFlag"
    Private Const DB_Get_UnSent_Msgs_SP = "ST_GetUnSentMsgs"
    Public SMSConnection As System.Data.SqlClient.SqlConnection
'.SqlClient.OleDbConnection
    'Msgs
    Friend MsgsCommand As System.Data.SqlClient.SqlCommand
    'General
    Friend GeneralDataAdapter As System.Data.SqlClient.SqlDataAdapter
    Friend GeneralSelectCommand As System.Data.SqlClient.SqlCommand
#End Region
#Region "Constructor and connectivity DBNull General Select Execute part"
    Public Sub New()
        'InitConnection
        Me.SMSConnection = New System.Data.SqlClient.SqlConnection()
        Me.SMSConnection.ConnectionString =
MsgsParser.GetSystemVariables("ConnectionString")

        'General Data Adapter
        Me.GeneralDataAdapter = New System.Data.SqlClient.SqlDataAdapter()
        Me.GeneralDataAdapter.SelectCommand = Me.GeneralSelectCommand
        'General Select Command
        Me.GeneralSelectCommand = New System.Data.SqlClient.SqlCommand()
        Me.GeneralSelectCommand.Connection = Me.SMSConnection
        Me.GeneralDataAdapter.SelectCommand = Me.GeneralSelectCommand

    End Sub
    Public Sub OpenConnectvity()
        Try
            If Me.SMSConnection.State = ConnectionState.Closed Or
Me.SMSConnection.State = ConnectionState.Broken Then
                Me.SMSConnection.Open()
            End If
        Catch e As Exception
            ErrorHandler.ErrorHandler(e, "open connectivity")
        End Try
    End Sub
    Public Sub CloseConnectvity()
        Try
            Me.SMSConnection.Close()
        Catch e As Exception
            ErrorHandler.ErrorHandler(e, "close connectivity")
        End Try
    End Sub
    Public Function ExecuteSelecetCommands(ByVal ConstStr As String) As
DataSet
        Me.OpenConnectvity()
        Dim dsObj As New DataSet()
        GeneralSelectCommand.CommandText = ConstStr
        GeneralDataAdapter.Fill(dsObj)
        Me.CloseConnectvity()
        Return dsObj
    End Function
    Public Function DBNullCheck(ByVal val As Object) As Object
        val = Trim(val)
        If CStr(val) = "" Then
            Return Convert.DBNull
        Else
            Return val
        End If
    End Function
#End Region
#Region "Car Insert Update Get"
#Region "params"
    Public Sub CreateGeneralPrams()
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@msgCaller",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@msgDate",
System.Data.SqlDbType.DateTime, 8))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@msgTxt",
System.Data.SqlDbType.NVarChar, 480))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@strGUID",
System.Data.SqlDbType.NVarChar, 50))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@ID", System.Data.SqlDbType.BigInt, 8,
System.Data.ParameterDirection.Input, False, CType(19, Byte), CType(0,
Byte), "", System.Data.DataRowVersion.Current, Nothing))
    End Sub
    Public Sub CreateParamsCars()
        CreateGeneralPrams()
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@RETURN_VALUE",
System.Data.SqlDbType.Int, 4, System.Data.ParameterDirection.ReturnValue,
False, CType(10, Byte), CType(0, Byte), "",
System.Data.DataRowVersion.Current, Nothing))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@maincat",
System.Data.SqlDbType.NVarChar, 30))
```

```
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@make", System.Data.SqlDbType.NVarChar,
30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@model",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@country",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@motor",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@makeyear",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@price",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@color",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@feature",
System.Data.SqlDbType.NVarChar, 200))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@IsBuyer", System.Data.SqlDbType.Bit,
1))
    End Sub
    Public Sub CreateParamsEstate()
        CreateGeneralPrams()
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@RETURN_VALUE",
System.Data.SqlDbType.Int, 4, System.Data.ParameterDirection.ReturnValue,
False, CType(10, Byte), CType(0, Byte), "",
System.Data.DataRowVersion.Current, Nothing))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@IsBuyer", System.Data.SqlDbType.Bit,
1))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@Purpose",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@Location",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@Area", System.Data.SqlDbType.NVarChar,
30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@Room", System.Data.SqlDbType.NVarChar,
30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@ConsistOf",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@Price",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@Type", System.Data.SqlDbType.NVarChar,
30))
```

```
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@Floor",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@Feature",
System.Data.SqlDbType.NVarChar, 200))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@knldgLoc",
System.Data.SqlDbType.NVarChar, 100))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@LocAttribute",
System.Data.SqlDbType.NVarChar, 100))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@areaAttribute",
System.Data.SqlDbType.NVarChar, 100))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@priceAttribute",
System.Data.SqlDbType.NVarChar, 100))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@estateType",
System.Data.SqlDbType.NVarChar, 30))

    End Sub
#End Region
    Public Sub CreateMsgsCommand(ByVal sp_name_str As String)
        Me.MsgsCommand = New System.Data.SqlClient.SqlCommand()
        Me.MsgsCommand.Connection = Me.SMSConnection
        MsgsCommand.CommandText = sp_name_str
        MsgsCommand.CommandType = System.Data.CommandType.StoredProcedure
    End Sub
#Region "Cars"
    Public Sub InsertCarsDB(ByVal MsgRec As
MsgStatmentRec.StatmentRecordCars)
        Me.OpenConnectvity()
        CreateMsgsCommand(DB_ADD_Car_SP)
        CreateParamsCars()
        MsgsCommand.Parameters("@ID").Value = Convert.DBNull
        MsgsCommand.Parameters("@msgCaller").Value = (MsgRec.MsgPhoneNo)
        MsgsCommand.Parameters("@msgDate").Value = (MsgRec.MsgDate)
        MsgsCommand.Parameters("@msgTxt").Value =
DBNullCheck(MsgRec.MsgTxt)

        MsgsCommand.Parameters("@maincat").Value =
DBNullCheck(MsgRec.maincat)
        MsgsCommand.Parameters("@make").Value = DBNullCheck(MsgRec.make)
        MsgsCommand.Parameters("@model").Value = DBNullCheck(MsgRec.model)
        MsgsCommand.Parameters("@country").Value =
DBNullCheck(MsgRec.country)
        MsgsCommand.Parameters("@motor").Value = DBNullCheck(MsgRec.motor)
        MsgsCommand.Parameters("@makeyear").Value =
DBNullCheck(MsgRec.year)
        MsgsCommand.Parameters("@price").Value = DBNullCheck(MsgRec.price)
        MsgsCommand.Parameters("@color").Value = DBNullCheck(MsgRec.color)
        MsgsCommand.Parameters("@feature").Value =
DBNullCheck(MsgRec.feature)
        MsgsCommand.Parameters("@strGUID").Value =
DBNullCheck(MsgRec.fileCUID)
        MsgsCommand.Parameters("@IsBuyer").Value = MsgRec.wantedBool
        MsgsCommand.ExecuteNonQuery()
        Me.CloseConnectvity()
    End Sub
    Public Sub UpdateCarsDB(ByVal id As Integer, ByVal MsgRec As
MsgStatmentRec.StatmentRecordCars)
        Me.OpenConnectvity()
        CreateMsgsCommand(DB_Update_Car_SP)
        CreateParamsCars()

        MsgsCommand.Parameters("@msgCaller").Value = Convert.DBNull
        MsgsCommand.Parameters("@msgDate").Value = Convert.DBNull
        MsgsCommand.Parameters("@msgTxt").Value = Convert.DBNull
        MsgsCommand.Parameters("@strGUID").Value = Convert.DBNull

        MsgsCommand.Parameters("@maincat").Value =
DBNullCheck(MsgRec.maincat)
        MsgsCommand.Parameters("@make").Value = DBNullCheck(MsgRec.make)
        MsgsCommand.Parameters("@model").Value = DBNullCheck(MsgRec.model)
        MsgsCommand.Parameters("@country").Value =
DBNullCheck(MsgRec.country)
        MsgsCommand.Parameters("@motor").Value = DBNullCheck(MsgRec.motor)
        MsgsCommand.Parameters("@makeyear").Value =
DBNullCheck(MsgRec.year)
        MsgsCommand.Parameters("@price").Value = DBNullCheck(MsgRec.price)
        MsgsCommand.Parameters("@color").Value = DBNullCheck(MsgRec.color)
        MsgsCommand.Parameters("@feature").Value =
DBNullCheck(MsgRec.feature)
        MsgsCommand.Parameters("@IsBuyer").Value = MsgRec.wantedBool
        MsgsCommand.Parameters("@id").Value = id

        MsgsCommand.ExecuteNonQuery()
        Me.CloseConnectvity()
    End Sub
    Public Function GetCarsMsgs(ByVal MsgRec As Object, ByVal
queryDatePeriodInt As Integer, ByVal AndOrInt As Integer, ByVal GuidIDSTR
As String) As DataSet
        Me.OpenConnectvity()
        CreateMsgsCommand(DB_Get_Cars_SP)
        CreateParamsCars()

        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@queryDatePeriod",
System.Data.SqlDbType.Int, 4, System.Data.ParameterDirection.Input, False,
CType(10, Byte), CType(0, Byte), "", System.Data.DataRowVersion.Current,
Nothing))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@AndOR",
System.Data.SqlDbType.SmallInt, 2, System.Data.ParameterDirection.Input,
False, CType(5, Byte), CType(0, Byte), "",
System.Data.DataRowVersion.Current, Nothing))

        If (Not MsgRec Is Nothing) Then
            'MsgStatmentRec.StatmentRecord()
            MsgsCommand.Parameters("@msgDate").Value = (MsgRec.MsgDate)
            MsgsCommand.Parameters("@make").Value =
DBNullCheck(MsgRec.make)
            MsgsCommand.Parameters("@model").Value =
DBNullCheck(MsgRec.model)
```

```
            MsgsCommand.Parameters("@country").Value =
DBNullCheck(MsgRec.country)
            MsgsCommand.Parameters("@maincat").Value =
DBNullCheck(MsgRec.maincat)
        Else
            MsgsCommand.Parameters("@msgDate").Value = Convert.DBNull
            MsgsCommand.Parameters("@make").Value = Convert.DBNull
            MsgsCommand.Parameters("@model").Value = Convert.DBNull
            MsgsCommand.Parameters("@country").Value = Convert.DBNull
            MsgsCommand.Parameters("@maincat").Value = Convert.DBNull
        End If

        MsgsCommand.Parameters("@strGUID").Value = DBNullCheck(GuidIDSTR)
        MsgsCommand.Parameters("@queryDatePeriod").Value =
DBNullCheck(queryDatePeriodInt)
        MsgsCommand.Parameters("@AndOr").Value = AndOrInt

        MsgsCommand.Parameters("@ID").Value = Convert.DBNull
        MsgsCommand.Parameters("@msgCaller").Value = Convert.DBNull
        MsgsCommand.Parameters("@msgTxt").Value = Convert.DBNull
        MsgsCommand.Parameters("@motor").Value = Convert.DBNull
        MsgsCommand.Parameters("@makeyear").Value = Convert.DBNull
        MsgsCommand.Parameters("@price").Value = Convert.DBNull
        MsgsCommand.Parameters("@color").Value = Convert.DBNull
        MsgsCommand.Parameters("@feature").Value = Convert.DBNull
        MsgsCommand.Parameters("@IsBuyer").Value = Convert.DBNull

        Dim ds As New DataSet()
        Dim DA As New System.Data.SqlClient.SqlDataAdapter(MsgsCommand)
        DA.Fill(ds)
        Me.CloseConnectvity()
        Return ds
    End Function
#End Region
#Region "Log"
    Public Sub DeleteMsgLogDB(ByVal MsgPhoneNo As String)
        Me.OpenConnectvity()
        CreateMsgsCommand(DB_Delete_Msgs_Log)

        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@msgCaller",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@msgTxt",
System.Data.SqlDbType.NVarChar, 2000))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@msgLength", System.Data.SqlDbType.Int,
4, System.Data.ParameterDirection.Input, False, CType(10, Byte), CType(0,
Byte), "", System.Data.DataRowVersion.Current, Nothing))

        MsgsCommand.Parameters("@msgCaller").Value =
DBNullCheck(MsgPhoneNo)
        MsgsCommand.Parameters("@msgTxt").Value = Convert.DBNull
        MsgsCommand.Parameters("@msgLength").Value = Convert.DBNull

        MsgsCommand.ExecuteNonQuery()
        Me.CloseConnectvity()
    End Sub
    Public Sub InsertMsgLogDB(ByVal MsgPhoneNo As String, ByVal MsgTxt As
String, ByVal msgLength As Integer)
        Me.OpenConnectvity()
        CreateMsgsCommand(DB_ADD_Msgs_Log)

        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@msgCaller",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@msgTxt",
System.Data.SqlDbType.NVarChar, 2000))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@msgLength", System.Data.SqlDbType.Int,
4, System.Data.ParameterDirection.Input, False, CType(10, Byte), CType(0,
Byte), "", System.Data.DataRowVersion.Current, Nothing))

        MsgsCommand.Parameters("@msgCaller").Value =
DBNullCheck(MsgPhoneNo)
        MsgsCommand.Parameters("@msgTxt").Value = DBNullCheck(MsgTxt)
        MsgsCommand.Parameters("@msgLength").Value = DBNullCheck(msgLength)


        MsgsCommand.ExecuteNonQuery()
        Me.CloseConnectvity()
    End Sub
    Public Function GetMsgLogDB(ByVal MsgPhoneNo As String) As DataSet
        Me.OpenConnectvity()
        CreateMsgsCommand(DB_Get_Msgs_Log)

        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@msgCaller",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@msgTxt",
System.Data.SqlDbType.NVarChar, 2000))

        MsgsCommand.Parameters("@msgCaller").Value =
DBNullCheck(MsgPhoneNo)
        MsgsCommand.Parameters("@msgTxt").Value = Convert.DBNull

        Dim ds As New DataSet
        Dim DA As New System.Data.SqlClient.SqlDataAdapter(MsgsCommand)
        DA.Fill(ds)

        Me.CloseConnectvity()
        Return ds
    End Function
#End Region
#Region "Estate"
    Public Sub InsertEstateDB(ByVal MsgRec As
MsgStatmentRec.EstateStatmentRecord)
        Me.OpenConnectvity()
        CreateMsgsCommand(DB_ADD_Estate_SP)
        CreateParamsEstate()
        MsgsCommand.Parameters("@ID").Value = Convert.DBNull
        MsgsCommand.Parameters("@msgCaller").Value = (MsgRec.MsgPhoneNo)
        MsgsCommand.Parameters("@msgDate").Value = (MsgRec.MsgDate)
        MsgsCommand.Parameters("@msgTxt").Value =
DBNullCheck(MsgRec.MsgTxt)
```

```
        MsgsCommand.Parameters("@Purpose").Value = DBNullCheck(MsgRec.pur)
        MsgsCommand.Parameters("@Location").Value = DBNullCheck(MsgRec.loc)
        MsgsCommand.Parameters("@Area").Value = DBNullCheck(MsgRec.are)
        MsgsCommand.Parameters("@Room").Value = DBNullCheck(MsgRec.roo)
        MsgsCommand.Parameters("@ConsistOf").Value =
DBNullCheck(MsgRec.cns)
        MsgsCommand.Parameters("@Price").Value = DBNullCheck(MsgRec.price)
        MsgsCommand.Parameters("@Type").Value = DBNullCheck(MsgRec.typ)
        MsgsCommand.Parameters("@Floor").Value = DBNullCheck(MsgRec.flr)
        MsgsCommand.Parameters("@feature").Value =
DBNullCheck(MsgRec.feature)
        MsgsCommand.Parameters("@strGUID").Value =
DBNullCheck(MsgRec.fileCUID)
        MsgsCommand.Parameters("@IsBuyer").Value = MsgRec.wantedBool
        MsgsCommand.Parameters("@knldgLoc").Value =
DBNullCheck(MsgRec.knldgLoc)
        MsgsCommand.Parameters("@LocAttribute").Value =
DBNullCheck(MsgRec.LocAttribute)
        MsgsCommand.Parameters("@areaAttribute").Value =
DBNullCheck(MsgRec.areaAttribute)
        MsgsCommand.Parameters("@priceAttribute").Value =
DBNullCheck(MsgRec.priceAttribute)
        MsgsCommand.Parameters("@estateType").Value =
DBNullCheck(MsgRec.estateType)

        MsgsCommand.ExecuteNonQuery()
        Me.CloseConnectvity()
    End Sub
    Public Sub UpdateEstateDB(ByVal id As Integer, ByVal MsgRec As
MsgStatmentRec.EstateStatmentRecord)
        Me.OpenConnectvity()
        CreateMsgsCommand(DB_Update_Estate_SP)
        CreateParamsEstate()

        MsgsCommand.Parameters("@msgCaller").Value = Convert.DBNull
        MsgsCommand.Parameters("@msgDate").Value = Convert.DBNull
        MsgsCommand.Parameters("@msgTxt").Value = Convert.DBNull
        MsgsCommand.Parameters("@strGUID").Value = Convert.DBNull

        MsgsCommand.Parameters("@Purpose").Value = DBNullCheck(MsgRec.pur)
        MsgsCommand.Parameters("@Location").Value = DBNullCheck(MsgRec.loc)
        MsgsCommand.Parameters("@Area").Value = DBNullCheck(MsgRec.are)
        MsgsCommand.Parameters("@Room").Value = DBNullCheck(MsgRec.roo)
        MsgsCommand.Parameters("@ConsistOf").Value =
DBNullCheck(MsgRec.cns)
        MsgsCommand.Parameters("@Price").Value = DBNullCheck(MsgRec.price)
        MsgsCommand.Parameters("@Type").Value = DBNullCheck(MsgRec.typ)
        MsgsCommand.Parameters("@Floor").Value = DBNullCheck(MsgRec.flr)
        MsgsCommand.Parameters("@feature").Value =
DBNullCheck(MsgRec.feature)
        MsgsCommand.Parameters("@IsBuyer").Value = MsgRec.wantedBool
        MsgsCommand.Parameters("@knldgLoc").Value =
DBNullCheck(MsgRec.knldgLoc)
        MsgsCommand.Parameters("@LocAttribute").Value =
DBNullCheck(MsgRec.LocAttribute)
        MsgsCommand.Parameters("@areaAttribute").Value =
DBNullCheck(MsgRec.areaAttribute)
```

```
        MsgsCommand.Parameters("@priceAttribute").Value =
DBNullCheck(MsgRec.priceAttribute)
        MsgsCommand.Parameters("@estateType").Value =
DBNullCheck(MsgRec.estateType)
        MsgsCommand.Parameters("@id").Value = id

        MsgsCommand.ExecuteNonQuery()
        Me.CloseConnectvity()
    End Sub
    Public Function GetEstateMsgs(ByVal MsgRec As
MsgStatmentRec.EstateStatmentRecord, ByVal queryDatePeriodInt As Integer,
ByVal AndOrInt As Integer, ByVal GuidIDSTR As String) As DataSet
        Me.OpenConnectvity()
        CreateMsgsCommand(DB_Get_Estate_SP)
        CreateParamsEstate()

        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@queryDatePeriod",
System.Data.SqlDbType.Int, 4, System.Data.ParameterDirection.Input, False,
CType(10, Byte), CType(0, Byte), "", System.Data.DataRowVersion.Current,
Nothing))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@AndOR",
System.Data.SqlDbType.SmallInt, 2, System.Data.ParameterDirection.Input,
False, CType(5, Byte), CType(0, Byte), "",
System.Data.DataRowVersion.Current, Nothing))

        If (Not MsgRec Is Nothing) Then
            MsgsCommand.Parameters("@estateType").Value =
DBNullCheck(MsgRec.estateType)
            MsgsCommand.Parameters("@Location").Value =
DBNullCheck(MsgRec.loc)
            MsgsCommand.Parameters("@msgDate").Value =
DBNullCheck(MsgRec.MsgDate)
            MsgsCommand.Parameters("@knldgLoc").Value =
DBNullCheck(MsgRec.knldgLoc)
        Else
            MsgsCommand.Parameters("@estateType").Value = Convert.DBNull
            MsgsCommand.Parameters("@Location").Value = Convert.DBNull
            MsgsCommand.Parameters("@msgDate").Value = Convert.DBNull
            MsgsCommand.Parameters("@knldgLoc").Value = Convert.DBNull
        End If

        MsgsCommand.Parameters("@strGUID").Value = DBNullCheck(GuidIDSTR)
        MsgsCommand.Parameters("@queryDatePeriod").Value =
DBNullCheck(queryDatePeriodInt)
        MsgsCommand.Parameters("@AndOr").Value = AndOrInt

        MsgsCommand.Parameters("@msgCaller").Value = Convert.DBNull
        MsgsCommand.Parameters("@msgTxt").Value = Convert.DBNull

        MsgsCommand.Parameters("@Purpose").Value = Convert.DBNull
        MsgsCommand.Parameters("@Area").Value = Convert.DBNull
        MsgsCommand.Parameters("@Room").Value = Convert.DBNull
        MsgsCommand.Parameters("@ConsistOf").Value = Convert.DBNull
        MsgsCommand.Parameters("@Price").Value = Convert.DBNull
        MsgsCommand.Parameters("@Type").Value = Convert.DBNull
        MsgsCommand.Parameters("@Floor").Value = Convert.DBNull
        MsgsCommand.Parameters("@feature").Value = Convert.DBNull
        MsgsCommand.Parameters("@IsBuyer").Value = Convert.DBNull
        'MsgsCommand.Parameters("@knldgLoc").Value = Convert.DBNull
        MsgsCommand.Parameters("@LocAttribute").Value = Convert.DBNull
        MsgsCommand.Parameters("@areaAttribute").Value = Convert.DBNull
        MsgsCommand.Parameters("@priceAttribute").Value = Convert.DBNull
        MsgsCommand.Parameters("@id").Value = Convert.DBNull

        Dim ds As New DataSet
        Dim DA As New System.Data.SqlClient.SqlDataAdapter(MsgsCommand)
        DA.Fill(ds)
        Me.CloseConnectvity()
        Return ds
    End Function
#End Region
#Region "Msgs"
    Public Sub InsertMsgDB(ByVal MsgRec As
MsgStatmentRec.StatmentRecordGen)
        Me.OpenConnectvity()
        CreateMsgsCommand(DB_ADD_Msg_SP)
        CreateGeneralPrams()
        MsgsCommand.Parameters("@ID").Value = Convert.DBNull
        MsgsCommand.Parameters("@msgCaller").Value = (MsgRec.MsgPhoneNo)
        MsgsCommand.Parameters("@msgDate").Value = (MsgRec.MsgDate)
        MsgsCommand.Parameters("@msgTxt").Value =
DBNullCheck(MsgRec.MsgTxt)
        MsgsCommand.Parameters("@strGUID").Value =
DBNullCheck(MsgRec.fileCUID)
        MsgsCommand.ExecuteNonQuery()
        Me.CloseConnectvity()
    End Sub
    Public Function GetMsgDB(ByVal strGuid As String) As DataSet
        Me.OpenConnectvity()
        CreateMsgsCommand(DB_Get_Msg_SP)
        CreateGeneralPrams()

        MsgsCommand.Parameters("@msgCaller").Value = Convert.DBNull
        MsgsCommand.Parameters("@msgDate").Value = Convert.DBNull
        MsgsCommand.Parameters("@msgTxt").Value = Convert.DBNull
        MsgsCommand.Parameters("@id").Value = Convert.DBNull
        MsgsCommand.Parameters("@strGUID").Value = strGuid

        Dim ds As New DataSet
        Dim DA As New System.Data.SqlClient.SqlDataAdapter(MsgsCommand)
        DA.Fill(ds)
        Me.CloseConnectvity()
        Return ds
    End Function
#End Region
#Region "Activate Deactivate and Set Send Flag"
    Public Sub ActivateDeactivate(ByVal msgCaller As String, ByVal
ActDeActFlag As Integer)
        Me.OpenConnectvity()
        CreateMsgsCommand(DB_Activate_Deactivate_SP)
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@msgCaller",
System.Data.SqlDbType.NVarChar, 30))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@activateFlag",
System.Data.SqlDbType.Bit, 1))
```

```
        MsgsCommand.Parameters("@msgCaller").Value = msgCaller
        MsgsCommand.Parameters("@activateFlag").Value = ActDeActFlag

        MsgsCommand.ExecuteNonQuery()
        Me.CloseConnectvity()
    End Sub
    Public Sub SetSendFlag(ByVal strGUID As String, ByVal sendFlag As
Integer)
        Me.OpenConnectvity()
        CreateMsgsCommand(DB_Set_Send_Flag_SP)
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@strGUID",
System.Data.SqlDbType.NVarChar, 50))
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@SendFlag", System.Data.SqlDbType.Bit,
1))

        MsgsCommand.Parameters("@strGUID").Value = strGUID
        MsgsCommand.Parameters("@SendFlag").Value = sendFlag

        MsgsCommand.ExecuteNonQuery()
        Me.CloseConnectvity()
    End Sub
    Public Function GetUnSentMsgsDB(ByVal ValidityPeriod As Integer) As
DataSet
        Me.OpenConnectvity()
        CreateMsgsCommand(DB_Get_UnSent_Msgs_SP)
        MsgsCommand.Parameters.Add(New
System.Data.SqlClient.SqlParameter("@ValidityPeriod",
System.Data.SqlDbType.Int, 4, System.Data.ParameterDirection.Input, False,
CType(10, Byte), CType(0, Byte), "", System.Data.DataRowVersion.Current,
Nothing))
        MsgsCommand.Parameters("@ValidityPeriod").Value = ValidityPeriod

        Dim ds As New DataSet
        Dim DA As New System.Data.SqlClient.SqlDataAdapter(MsgsCommand)
        DA.Fill(ds)
        Me.CloseConnectvity()
        Return ds
    End Function

#End Region
#End Region
End Class
```

APPENDIX L
A SMAPLE OF THE CATS INPUT LOG

```
5/24/2005 1:52:28 PM   90050

    5/24/2005 1:52:28 PM   Message In : +962795016664,230 للبيع مرسيدس
89 م
قير اتوماتيك
بدون مكيف
قزاز عادي
حره غير مجمرك

    5/24/2005 1:52:58 PM   90050

    5/24/2005 1:52:58 PM   Message In : +962795015050, ب م دبليو م 83 . 518 . مكيف بور كراسي كهربا ذهبي
1800سي سي

    5/24/2005 1:52:58 PM   90050

    5/24/2005 1:52:58 PM   Message In : +962795060879,زاز موديل1993 نفظ كامل ترخيص جديد للبيع بالف دينار

    5/24/2005 1:53:28 PM   90050

    5/24/2005 1:53:28 PM   Message In : +962795060648,□□□□ موديل□□□ للبيع سيارة بيجو

    5/24/2005 1:55:25 PM   90050

    5/24/2005 1:55:25 PM   Message In : +962795012563,مطلوب باص بريجو مديل الفين

    5/24/2005 1:55:28 PM   90050

    5/24/2005 1:55:28 PM   Message In : +962795030435, سياره دايو نوبيرا موديل 97 لون سلفر جمرك حديث فل أوبشن
جير أوتامتيك فحص كامل السعر النهائي دينار6400

    5/24/2005 1:56:54 PM   90050

    5/24/2005 1:56:54 PM   Message In : +962795060805,مطلوب قلاب دايهاتسو اقساط مديل الثمينات

    5/24/2005 1:57:53 PM   90050

    5/24/2005 1:57:53 PM   Message In : +962795102223, كازيه للبيع ع شارعين رئيسيين مقابل شركة الدخان.ناعور
المطلوب □□□□□□□.د

    5/24/2005 1:58:52 PM   90050

    5/24/2005 1:58:52 PM   Message In : +962795219537,فله

    5/24/2005 1:59:21 PM   90050
```

5/24/2005 1:59:21 PM Message In : +962795161007,94 للبيع دايو ليمنز موديل

5/24/2005 1:59:23 PM 90050

5/24/2005 1:59:23 PM Message In : +962795018664, مطلوب سياره موديل 90اواكثر ولا يشترط نوع السياره بدون دفعه اولي وعلىاقساط 250شهرى

5/24/2005 1:59:52 PM 90050

5/24/2005 1:59:52 PM Message In : +962795115381,فيلا للبيع..خلدا 500متر أرض 340 م بناء

5/24/2005 2:00:51 PM 90050

5/24/2005 2:00:51 PM Message In : +962795053965,:-)ابو مصعب :فيات 132 موديل 79 بحاله جيده

5/24/2005 2:00:51 PM 90050

5/24/2005 2:00:51 PM Message In : +962795184639, .يوجد قطع اراضي للبيع مطله ومشرفه على سد الملك طلال للمراجعه □□□□□□□□□

5/24/2005 2:01:19 PM 90050

5/24/2005 2:01:19 PM Message In : +962795205638,سياره ألماني أو ياباني شحن حموله أربعه طون

5/24/2005 2:01:19 PM 90050

5/24/2005 2:01:19 PM Message In : +962795136638, باص تويتا □□ فان للبيع .المراجعه على هاتف □□□□□□□□□□□

5/24/2005 2:01:51 PM 90050

5/24/2005 2:01:51 PM Message In : +962795242602, لبيع،طلب سيارة،عقار ارسل الاعلان المبوب ل90050 وتلقى الرد فورا ب30قرش

5/24/2005 2:02:19 PM 90050

5/24/2005 2:02:19 PM Message In : +962795243613,طلب سياره

5/24/2005 2:02:21 PM 90050

5/24/2005 2:02:21 PM Message In : +962795118813, منزل في العقبه في منطقه الثالثه مكون من شقتين سوبر ديلوكس مفروش أو فارغ المساحه 225م.للبيع فارغ 60الف مفروش 80الف

5/24/2005 2:04:24 PM 90050

5/24/2005 2:04:24 PM Message In : +962795250566,طلب سياره تويوتا ستارلت

5/24/2005 2:04:24 PM 90050

5/24/2005 2:04:24 PM Message In : +962795198864, محل تجاري مساحه 65 م يصلح لكافت الاعمال الاشرفيه تقاطع سميه سعر مغري

5/24/2005 2:04:25 PM 90050

5/24/2005 2:04:25 PM Message In : +962795096608,سياره-تويوتا كورونا 7 8 سنتر بور كندشن بحالة جيدة السعر 2600 نهاءئ

5/24/2005 2:04:25 PM 90050

5/24/2005 2:04:25 PM Message In : +962795060387,للبيع ارض في المزار الجنوبي قرب مدرسه جعفر

5/24/2005 2:04:28 PM 90050

5/24/2005 2:04:28 PM Message In : +962795077756, مطلوب عارضات ازياء لشركه ازياء بدبي واخصائيات تجميل ومساج هاتف 0097150 5389945

5/24/2005 2:04:55 PM 90050

5/24/2005 2:04:55 PM Message In : +962795204561,90050بيع،،عقار ارض ل

5/24/2005 2:04:55 PM 90050

5/24/2005 2:04:55 PM Message In : +962795241705,تويوتا كريسدا موديل 200 غير مجمركه

5/24/2005 2:05:23 PM 90050

5/24/2005 2:05:23 PM Message In : +962795242990,للبيع. مقهى في الزرقاء الجديده في شارع سته وثلثون

5/24/2005 2:05:53 PM 90050

5/24/2005 2:05:53 PM Message In : +962795076130, للبيع سيارة دايو اسبيرو موديل 92 فل عدا الجير بحالة جيدة السعر 3800

5/24/2005 2:06:23 PM 90050

5/24/2005 2:06:23 PM Message In : +962795243275, سياره هونداى اكسنت 96 فل ابشن للبيع اقساط او كاش0795243275

5/24/2005 2:06:25 PM 90050

5/24/2005 2:06:25 PM Message In : +962795071066,بيع، سيارة، مرسيدس 87 بطه كافه الاضافات كحلي فحص كامل. هوندا جيب crv 99/ فل اوبشن فحص كامل 0795071066

5/24/2005 2:07:22 PM 90050

5/24/2005 2:07:22 PM Message In : +962795216316,للبيع مرسيدس s300
موديل92
غير مجمركة

```
5/24/2005 2:08:21 PM   90050

5/24/2005 2:08:21 PM   Message In : +962795190970, للبيع بكب ايسوزو فحص كامل موديل الفين واربعه جميع
الاضافات

5/24/2005 2:09:20 PM   90050

5/24/2005 2:09:20 PM   Message In : +962795320379,سياره دايو نبيرا مديل 2000 سعر 8500

5/24/2005 2:09:50 PM   90050

5/24/2005 2:09:50 PM   Message In : +962795293182,مركز مساج

5/24/2005 2:10:48 PM   90050

5/24/2005 2:10:48 PM   Message In : +962795345522,طلب سيارة

5/24/2005 2:10:49 PM   90050

5/24/2005 2:10:49 PM   Message In : +962795364186,الاعلان المبوب

5/24/2005 2:10:49 PM   90050

5/24/2005 2:10:49 PM   Message In : +962795363696,للبيع سياره هوندا سيفك 98

5/24/2005 2:11:19 PM   90050

5/24/2005 2:11:19 PM   Message In : +962795374240,مطلوب سكرتيره

5/24/2005 2:11:21 PM   90050

5/24/2005 2:11:21 PM   Message In : +962795095785, بيع مزرعه 7 دونمات مشجره زيتون وعنب عمرها 13 سنه بها
بئر إرتوازي 6إنش . سياج حرجي .غرفه+بركس 20%6 في الازرق الرتامه

5/24/2005 2:11:48 PM   90050

5/24/2005 2:11:48 PM   Message In : +962795249774, لبيع،طلب سيارة،عقار ارسل الاعلان المبوب ل90050 وتلقى
الرد فورا ب30قرش

5/24/2005 2:12:17 PM   90050

5/24/2005 2:12:17 PM   Message In : +962795311522, أرض للبيع في حي نزال  □□□م تصلح لبناء عمارة موقع
ممتاز الخدمات قريبة.

5/24/2005 2:12:49 PM   90050

5/24/2005 2:12:49 PM   Message In : +962795407746, لبيع،طلب سيارة،عقار ارسل الاعلان المبوب ل90050 وتلقى
الرد فورا ب30قرش

5/24/2005 2:12:49 PM   90050
```

```
5/24/2005 2:12:49 PM   Message In : +962795015566,Used Mini cooper 2002 wanted

5/24/2005 2:13:17 PM   90050

5/24/2005 2:13:17 PM   Message In : +962795389877,اطلب تكسي اقساط

5/24/2005 2:13:17 PM   90050

5/24/2005 2:13:17 PM   Message In : +962795121447,انابدي باص كيا   كيف البيع

5/24/2005 2:13:47 PM   90050

5/24/2005 2:13:47 PM   Message In : +962795411508,اودي 1994

5/24/2005 2:14:15 PM   90050

5/24/2005 2:14:15 PM   Message In : +962795407070,مطلوب قطعة ارض في ام السماق او خلدا او تلاع العلي

5/24/2005 2:14:16 PM   90050

5/24/2005 2:14:16 PM   Message In : +962795280784,قرش ونص م 98

5/24/2005 2:14:45 PM   90050

5/24/2005 2:14:45 PM   Message In : +962795357772,هوندا سيفك 97 فحص كامل لون خمري كندشن سنتر بور للبيع

5/24/2005 2:15:15 PM   90050

5/24/2005 2:15:15 PM   Message In : +962795403012,مطلوب سياره مرسيدس كوبيه 200 موديل 90

5/24/2005 2:15:16 PM   90050

5/24/2005 2:15:16 PM   Message In : +962795468602,لبيع،طلب سيارة،عقار ارسل الاعلان المبوب ل90050 وتلقى
الرد فورا ب30قرش

5/24/2005 2:15:16 PM   90050

5/24/2005 2:15:16 PM   Message In : +962795408763,1280 متر مع بيت طابقين 300 متر في ابو ديس القدس

5/24/2005 2:15:17 PM   90050

5/24/2005 2:15:17 PM   Message In : +962795396088,سياره.مرسيدس.300...موديل.1980..فل.مطلوب.7000.دينار

5/24/2005 2:16:16 PM   90050

5/24/2005 2:16:16 PM   Message In : +962795180177,مزرعة للبيع في المفرق  28 دلم مزروعة زيتون 400 عمر
الزيتون 25 سنة بسعر 5000  الدلم قابل للتفاوض 0 وهي على شارع بغداد شرق جامعة ال البيت 4 كيلو   0 صاحب المزرعة ظافر القاضي
```

```
5/24/2005 2:16:44 PM   90050

5/24/2005 2:16:44 PM   Message In : +962795484359,سياره شكودا 94للبيع عندي

5/24/2005 2:18:12 PM   90050

5/24/2005 2:18:12 PM   Message In : +962795443664, مطلوب سيارة مرسيدس كوبيه .ماتور 2000 سس.موديل 92
لغاية 95

5/24/2005 2:18:44 PM   90050

5/24/2005 2:18:44 PM   Message In : +962795409302, ارض تجاري قرب جامعة مؤتة للبيع . السعر النهائي 50 دينار
المتر.

5/24/2005 2:19:12 PM   90050

5/24/2005 2:19:12 PM   Message In : +962795531691,Car

5/24/2005 2:19:12 PM   90050

5/24/2005 2:19:12 PM   Message In : +962795475987,What d mineng dish.

5/24/2005 2:19:41 PM   90050

5/24/2005 2:19:41 PM   Message In : +962795440017,ارض بجانب اثار ام الرصاص  □□□م□ اسعر  □□□□□

5/24/2005 2:19:41 PM   90050

5/24/2005 2:19:41 PM   Message In : +962795412191,سياره اوبل اوميغا للبيع م 2000 فحص جميع الاضافات

5/24/2005 2:20:10 PM   90050

5/24/2005 2:20:10 PM   Message In : +962795408763,520  متر في اسكان سلطه المياه 2 الدبه حوض 3 قبل ابو
السوس البيادر

5/24/2005 2:20:11 PM   90050

5/24/2005 2:20:11 PM   Message In : +962795568493, لبيع طلب سيارة عقار ارسل الاعلان المبوب ل90050 وتلقى
الرد فورا ب30قرش

5/24/2005 2:20:11 PM   90050

5/24/2005 2:20:11 PM   Message In : +962795515120,

5/24/2005 2:20:12 PM   90050

5/24/2005 2:20:12 PM   Message In : +962795282001,طلب سياره

5/24/2005 2:20:42 PM   90050
```

5/24/2005 2:20:42 PM Message In : +962795313472, لبيع مرسيدس 200 -72-بحاله جيده -بور سنتر-انذار-1700 دينار

5/24/2005 2:21:09 PM 90050

5/24/2005 2:21:09 PM Message In : +962795472818, بدي سياره دفعه الف دينار والباقي اقساط بقيمه حوالي الفين دينار

5/24/2005 2:21:10 PM 90050

5/24/2005 2:21:10 PM Message In : +962795562722,مطلوب سياره أوبل أوميغا 95 بسلفه 2000 واقساط شهريه

5/24/2005 2:21:12 PM 90050

5/24/2005 2:21:12 PM Message In : +962795564440,سياره

5/24/2005 2:21:40 PM 90050

5/24/2005 2:21:40 PM Message In : +962795583112,عقار مساحه .130م.

5/24/2005 2:22:41 PM 90050

5/24/2005 2:22:41 PM Message In : +962795526595, يوجد لدينا سياره متسوبيشي باجيرو مديل 88 بسعر 6 ألاف دينار ألمراجعه مازن تلفون 0795526595

5/24/2005 2:23:08 PM 90050

5/24/2005 2:23:08 PM Message In : +962795572966,عندي بيت طابقين في الكرك للبيع وبيت في عمان جبل التاج

5/24/2005 2:23:08 PM 90050

5/24/2005 2:23:08 PM Message In : +962795477458,سياره.موديل.1970.:.مرسيدس.فحص.مرخصه.حديث.للبيع.بسعر □□□□دينار

5/24/2005 2:23:39 PM 90050

5/24/2005 2:23:39 PM Message In : +962795412191,مطلوب سياره بطه م 85 الي 88 سعر طري مش فحص

5/24/2005 2:24:08 PM 90050

5/24/2005 2:24:08 PM Message In : +962795628297,بيجو 206 موديل 2002 للبيع لون أبيض

5/24/2005 2:24:10 PM 90050

5/24/2005 2:24:10 PM Message In : +962795233276, للبيع بيت في ضاحية الامير هاشم طبربور من طابقين كل طابق 100م الارض نصف دنم ت:0795233276

5/24/2005 2:25:06 PM 90050

```
5/24/2005 2:25:06 PM   Message In : +962795617892,

5/24/2005 2:25:08 PM   90050

5/24/2005 2:25:08 PM   Message In : +962795326516,لدي محل تجاري في العقبة ارغب ببيعه. في شارع الاوقاف. ش رئيسي طول المحل 17م ومؤجر الآن .

5/24/2005 2:25:38 PM   90050

5/24/2005 2:25:38 PM   Message In : +962795459645,بدي سيارة اقساط بدون دفعة شو قسط وكم سعرها

5/24/2005 2:25:39 PM   90050

5/24/2005 2:25:39 PM   Message In : +962795555422,مطلوب سياره أمريكي بحاله ممتازه موديل 85-80

5/24/2005 2:25:39 PM   90050

5/24/2005 2:25:39 PM   Message In : +962795550237,مطلوب شراء سياره أوبل فكترا □□ او □□□□فل أوبشن

5/24/2005 2:25:40 PM   90050

5/24/2005 2:25:40 PM   Message In : +962795630673,باص كيا بريجو 1995

5/24/2005 2:25:41 PM   90050

5/24/2005 2:25:41 PM   Message In : +962795581985,مطلوب أرض أو شقه بسعر معقول في العقبه
للمراجعه
0795581985

5/24/2005 2:26:36 PM   90050

5/24/2005 2:26:36 PM   Message In : +962795423254, سياره موديل ثلاث وسبعون اربعه وعشرين صهريج نضح
فحص كامل

5/24/2005 2:26:37 PM   90050

5/24/2005 2:26:37 PM   Message In : +962795653655,سيارة دايو نوبرا 97 فل ما عدا الجير السعر 4750 دينار

5/24/2005 2:26:37 PM   90050

5/24/2005 2:26:37 PM   Message In : +962795665232, أبيع سياره فكترا 39

5/24/2005 2:26:38 PM   90050

5/24/2005 2:26:38 PM   Message In : +962795349993,بدي اتجوز

5/24/2005 2:27:07 PM   90050

5/24/2005 2:27:07 PM   Message In : +962795344433,بيع سياره بي أم 520 مديل 96 فل
```

5/24/2005 2:27:07 PM 90050

5/24/2005 2:27:07 PM Message In : +962795691820, لبيع،طلب سيارة،عقار ارسل الاعلان المبوب ل90050 وتلقى الرد فورا ب30قرش

5/24/2005 2:28:04 PM 90050

5/24/2005 2:28:04 PM Message In : +962795663349, أطلب سياره سعرها كاش 5000دينار بدفعه 500دينار و200 شهريا

5/24/2005 2:28:07 PM 90050

5/24/2005 2:28:07 PM Message In : +962795556778, لبيع،طلب سيارة،عقار ارسل الاعلان المبوب ل90050 وتلقى الرد فورا ب30قرش saab vector 2004 for sale

5/24/2005 2:28:36 PM 90050

5/24/2005 2:28:36 PM Message In : +962795239308, للبيع محل خلويات مع الديكور والبضاعه اوبدون الموقع فن الهاشمي الشمالي

5/24/2005 2:28:36 PM 90050

5/24/2005 2:28:36 PM Message In : +962795519012,93 بيع سياره تاتا بكب ثلاجة م

5/24/2005 2:29:05 PM 90050

5/24/2005 2:29:05 PM Message In : +962795562152, ارض طريق المطار في ام العمد 10دنم مزروعه زيتون وبها فيلا ريفيه بسعر 270الف دينار لكامل المزرعه تلفون0795562152

APPENDIX M
A SAMPLE LOG OF CATS RESPONSES

```
  5/24/2005 1:52:39 PM   Out Message: +962795049358,  *mercedes
795510100:200
795967002:180
795444474:200
795594258:200
796623456:560
962796474470
796925603:uno
795145032:200
795555241:280
795400445:200

   5/24/2005 1:52:39 PM   Out Message: +962795016664,  Your advertisement has been received.

   5/24/2005 1:53:39 PM   Out Message: +962795060879,  Your advertisement has been received.

   5/24/2005 1:53:39 PM   Out Message: +962795060648,  Your advertisement has been received.

   5/24/2005 1:55:40 PM   Out Message: +962795012563,  *hyundai
0795568896
795088534
0795937707
*mitsubishi
0795444349
*nissan
0795601711:urvan
0796581408
0795086753
0795601711
*kia
0795171077

   5/24/2005 1:55:40 PM   Out Message: +962795030435,  Your advertisement has been received.

   5/24/2005 1:57:40 PM   Out Message: +962795060805,  Your advertisement has been received.

   5/24/2005 1:59:40 PM   Out Message: +962795161007,  Your advertisement has been received.


   5/24/2005 2:00:41 PM   Out Message: +962795115381,  Your advertisement has been received.

   5/24/2005 2:01:41 PM   Out Message: +962795053965,  Your advertisement has been received.
   5/24/2005 2:01:41 PM   Out Message: +962795184639,  Your advertisement has been received.

   5/24/2005 2:01:41 PM   Out Message: +962795205638,  Your advertisement has been received.

   5/24/2005 2:01:41 PM   Out Message: +962795136638,  Your advertisement has been received.



   5/24/2005 2:02:43 PM   Out Message: +962795243613,  *skoda
795170866:felicia
796885599:felicia
795791339:felicia
795533222
795786353:felicia
795012569
795900567:fabia
795904901:fabia
*nissan

   5/24/2005 2:02:43 PM   Out Message: +962795118813,  Your advertisement has been received.

   5/24/2005 2:04:44 PM   Out Message: +962795250566,  *toyota
795271778:corolla
795139513:corolla
796782260:corolla
795066516
962795715719:cresseda
795888398
795546408
0795490206:celica
795299982:celica

   5/24/2005 2:04:44 PM   Out Message: +962795198864,  Your advertisement has been received.

   5/24/2005 2:04:44 PM   Out Message: +962795096608,  Your advertisement has been received.

   5/24/2005 2:04:44 PM   Out Message: +962795060387,  Your advertisement has been received.

   5/24/2005 2:04:45 PM   Out Message: +962795077756,  *المسيطبة،الفرش،دلة
795538175
*الكرسي
796470607
0795304047
796499433
*ما حص
796499433
*عصفور
796745852
```

*طريق المطار
0796804416

5/24/2005 2:05:45 PM Out Message: +962795204561, Your advertisement has been received.

5/24/2005 2:06:45 PM Out Message: +962795076130, Your advertisement has been received.

5/24/2005 2:06:45 PM Out Message: +962795243275, Your advertisement has been received.

5/24/2005 2:06:45 PM Out Message: +962795071066, Your advertisement has been received.

5/24/2005 2:07:46 PM Out Message: +962795216316, Your advertisement has been received.

5/24/2005 2:08:46 PM Out Message: +962795190970, Your advertisement has been received.

5/24/2005 2:09:46 PM Out Message: +962795320379, Your advertisement has been received.

5/24/2005 2:11:47 PM Out Message: +962795345522, *skoda
795170866:felicia
796885599:felicia
795791339:felicia
795533222
795786353:felicia
795012569
795900567:fabia
795904901:fabia
*nissan

5/24/2005 2:11:47 PM Out Message: +962795363696, Your advertisement has been received.

5/24/2005 2:12:48 PM Out Message: +962795311522, Your advertisement has been received.

5/24/2005 2:13:49 PM Out Message: +962795407746, *skoda
795170866:felicia
796885599:felicia
795791339:felicia
795533222
795786353:felicia
795012569
795900567:fabia
795904901:fabia
*nissan

5/24/2005 2:13:49 PM Out Message: +962795121447, *kia
0795171077

5/24/2005 2:14:49 PM Out Message: +962795407070, *تلاع العلي

```
795519623
795663274
796770392
0795619940
795354215
795660247
795096485
796996498
795510070
0795771656
0796709709

   5/24/2005 2:14:49 PM   Out Message: +962795280784,  Your advertisement has been received.

   5/24/2005 2:14:50 PM   Out Message: +962795357772,  Your advertisement has been received.

   5/24/2005 2:15:50 PM   Out Message: +962795403012,  *mercedes
795280784:200
795510100:200
795444474:200
795594258:200
795145032:200
795400445:200
795642424:200
795167818:200
795040777:200
795424834:200

   5/24/2005 2:15:51 PM   Out Message: +962795468602,  *skoda
795170866:felicia
796885599:felicia
795791339:felicia
795533222
795786353:felicia
795012569
795900567:fabia
795904901:fabia
*nissan

   5/24/2005 2:15:51 PM   Out Message: +962795408763,  Your advertisement has been received.

   5/24/2005 2:15:51 PM   Out Message: +962795396088,  Your advertisement has been received.

   5/24/2005 2:16:51 PM   Out Message: +962795180177,  Your advertisement has been received.

   5/24/2005 2:16:51 PM   Out Message: +962795484359,  Your advertisement has been received.

   5/24/2005 2:18:52 PM   Out Message: +962795443664,  *mercedes
795071066:87
```

```
795016664:230
795396088:300
795280784:200
795216316:300
795510100:200
795967002:180
795444474:200
795594258:200
796623456:560

   5/24/2005 2:18:52 PM   Out Message: +962795409302,  Your advertisement has been received.

   5/24/2005 2:19:52 PM   Out Message: +962795440017,  Your advertisement has been received.

   5/24/2005 2:19:52 PM   Out Message: +962795412191,  Your advertisement has been received.

   5/24/2005 2:20:52 PM   Out Message: +962795408763,  Your advertisement has been received.


   5/24/2005 2:20:54 PM   Out Message: +962795282001,  *skoda
795484359
795170866:felicia
796885599:felicia
795791339:felicia
795533222
795786353:felicia
795012569
795900567:fabia
795904901:fabia
*nissan

   5/24/2005 2:20:54 PM   Out Message: +962795313472,  Your advertisement has been received.


   5/24/2005 2:21:55 PM   Out Message: +962795562722,  *opel
795412191:omega
795682213:omega
795996005:omega
795928857:omega
795061726:omega
796870777:omega

   5/24/2005 2:22:55 PM   Out Message: +962795526595,  Your advertisement has been received.

   5/24/2005 2:23:55 PM   Out Message: +962795572966,  Your advertisement has been received.

   5/24/2005 2:23:55 PM   Out Message: +962795477458,  Your advertisement has been received.

   5/24/2005 2:23:55 PM   Out Message: +962795412191,  *mercedes
```

```
795313472:200
795280784:200
795510100:200
795444474:200
795594258:200
795145032:200
795400445:200
795642424:200
795167818:200
795040777:200

   5/24/2005 2:24:56 PM   Out Message: +962795628297,  Your advertisement has been received.

   5/24/2005 2:24:56 PM   Out Message: +962795233276,  Your advertisement has been received.


   5/24/2005 2:25:57 PM   Out Message: +962795555422,  *jeep
795519996
795953855
795221044:cherokee
795751609:grand
795726070:land cruiser
795869544:grandcherokee
796768174:grandcherokee
795512202

   5/24/2005 2:25:57 PM   Out Message: +962795550237,  *opel
795931363:vectra
795200188:vectra
796500681:vectra
795445424:vectra
795062184:vectra
795904901:vectra
795787766:vectra
795650954:vectra

   5/24/2005 2:25:57 PM   Out Message: +962795630673,  Your advertisement has been received.

   5/24/2005 2:25:57 PM   Out Message: +962795581985,  *aqaba
795996677
*محطة وقود
795591395
*القطرانة
796303103
795035738
*طريق عمان
795652838
*أبو الحصاني
795412116
```

```
*بريك

   5/24/2005 2:26:57 PM   Out Message: +962795653655,  Your advertisement has been received.

   5/24/2005 2:27:57 PM   Out Message: +962795344433,  Your advertisement has been received.


   5/24/2005 2:28:59 PM   Out Message: +962795239308,  Your advertisement has been received.

   5/24/2005 2:28:59 PM   Out Message: +962795519012,  Your advertisement has been received.

   5/24/2005 2:30:00 PM   Out Message: +962795562152,  Your advertisement has been received.

   5/24/2005 2:30:00 PM   Out Message: +962795698333,  *طبربور
795137654
796160501
796363378
796618729
796361178
795823475
795552143
0795572000
0795706688
796792123
795337730


   5/24/2005 2:31:01 PM   Out Message: +962795716406,  Your advertisement has been received.

   5/24/2005 2:31:01 PM   Out Message: +962795498533,  Your advertisement has been received.

   5/24/2005 2:32:01 PM   Out Message: 962795737735,  Your advertisement has been received.

   5/24/2005 2:33:01 PM   Out Message: +962795634602,  Your advertisement has been received.

   5/24/2005 2:33:01 PM   Out Message: +962795519012,  Your advertisement has been received.

   5/24/2005 2:34:01 PM   Out Message: +962795057986,  Your advertisement has been received.

   5/24/2005 2:34:01 PM   Out Message: +962795646643,  *mitsubishi
795130389:galant
795590838:galant
795903775:galant
796916911:galant
795904890:galant
795996050:galant
0795864448:galant
0795904890:galant
```

```
   5/24/2005 2:34:01 PM   Out Message: +962795731441,  Your advertisement has been received.

   5/24/2005 2:34:01 PM   Out Message: +962795828287,  Your advertisement has been received.

   5/24/2005 2:34:02 PM   Out Message: +962795788203,  *jeep
795519996
795953855
795221044:cherokee
795751609:grand
795726070:land cruiser
795869544:grandcherokee
796768174:grandcherokee
795512202

   5/24/2005 2:34:02 PM   Out Message: +962795731129,  *skoda
795484359
795170866:felicia
796885599:felicia
795791339:felicia
795533222
795786353:felicia
795012569
795900567:fabia
795904901:fabia
*nissan

   5/24/2005 2:35:03 PM   Out Message: +962795778848,  *hyundai
0795568896
795088534
0795937707
*mitsubishi
0795444349
*nissan
0795601711:urvan
0796581408
0795086753
0795601711
*kia
795136638
795630673

   5/24/2005 2:35:03 PM   Out Message: +962795818865,  Your advertisement has been received.

   5/24/2005 2:36:03 PM   Out Message: +962795553412,  Your advertisement has been received.

   5/24/2005 2:36:03 PM   Out Message: +962795806991,  Your advertisement has been received.

   5/24/2005 2:36:03 PM   Out Message: +962795702867,  Your advertisement has been received.
   5/24/2005 2:37:03 PM   Out Message: +962795838019,  Your advertisement has been received.

   5/24/2005 2:37:03 PM   Out Message: +962796491625,  Your advertisement has been received.

   5/24/2005 2:37:03 PM   Out Message: +962795840331,  Your advertisement has been received.

   5/24/2005 2:40:04 PM   Out Message: +962795795733,  Your advertisement has been received.

   5/24/2005 2:40:04 PM   Out Message: +962795783008,  *nissan
795250440:sunny
795088851:sunny
795118558:sunny
795602676:sunny
0796332525:sunny
796189600:sunny
795982785:sunny
795965390:sunny

   5/24/2005 2:40:05 PM   Out Message: +962795921035,  *skoda
795484359
795170866:felicia
796885599:felicia
795791339:felicia
795533222
795786353:felicia
795012569
795900567:fabia
795904901:fabia
*nissan

   5/24/2005 2:41:05 PM   Out Message: +962795700407,  Your advertisement has been received.

   5/24/2005 2:41:05 PM   Out Message: +962795870047,  لا يوجد معلومات حاليا فور توفرها سيقوم النظام ببعثها
اتوماتيكيا

   5/24/2005 2:42:05 PM   Out Message: +962795817430,  Your advertisement has been received.

   5/24/2005 2:42:05 PM   Out Message: +962795154718,  Your advertisement has been received.

   5/24/2005 2:42:05 PM   Out Message: +962795974040,  Your advertisement has been received.

   5/24/2005 2:43:06 PM   Out Message: +962795614826,  *mitsubishi
795828287:lancer
795526595:pajero
795331727:lancer
795415020:lancer
795130389:galant
796708687:lancer
```

```
795590838:galant
795903775:galant

   5/24/2005 2:44:07 PM   Out Message: +962795966267,  *skoda
795484359
795170866:felicia
796885599:felicia
795791339:felicia
795533222
795786353:felicia
795012569
795900567:fabia
795904901:fabia
*nissan

   5/24/2005 2:44:07 PM   Out Message: +962795129113,  Your advertisement has been received.

   5/24/2005 2:44:07 PM   Out Message: 962795792818,  Your advertisement has been received.

   5/24/2005 2:45:08 PM   Out Message: +962796508005,  *skoda
795484359
795170866:felicia
796885599:felicia
795791339:felicia
795533222
795786353:felicia
795012569
795900567:fabia
795904901:fabia
*nissan

   5/24/2005 2:45:09 PM   Out Message: +962795728442,  Your advertisement has been received.

   5/24/2005 2:45:09 PM   Out Message: +962796338908,  Your advertisement has been received.

   5/24/2005 2:45:10 PM   Out Message: +962796528809,  *skoda
795484359
795170866:felicia
796885599:felicia
795791339:felicia
795533222
795786353:felicia
795012569
795900567:fabia
795904901:fabia
*nissan

   5/24/2005 2:46:10 PM   Out Message: +962795850706,  *nissan
795250440:sunny
```

```
795088851:sunny
795118558:sunny
795602676:sunny
0796332525:sunny
796189600:sunny
795982785:sunny
795965390:sunny

   5/24/2005 2:46:10 PM   Out Message: +962796338908,  Your advertisement has been received.

   5/24/2005 2:46:12 PM   Out Message: +962795585403,  *جبل الحسين،ب795230601
795219046
795453837
795684674
795659917
795051153
795821236
796861068
795316997
795137654
796785049

   5/24/2005 2:47:13 PM   Out Message: +962795821655,  Your advertisement has been received.

   5/24/2005 2:47:13 PM   Out Message: +962795517822,  *mercedes
795313472:200
795280784:200
795510100:200
795444474:200
795594258:200
795145032:200
795400445:200
795642424:200
795167818:200
795040777:200

   5/24/2005 2:47:14 PM   Out Message: +962795861630,  *skoda
795484359
795170866:felicia
796885599:felicia
795791339:felicia
795533222
795786353:felicia
795012569
795900567:fabia
795904901:fabia
*nissan
```