Methods for Handling Spontaneous E-commerce Arabic SMS: CATS, an Operational Proof of Concept

Maher Daoud and Christian Boitet

***Abstract*—The purpose of this paper 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. To analyze the sublanguage, we first used a web based corpus to build the basic system. 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. Response generation is based on semantic matching ("looking for" and "sell" posts) and reasoning and is able to handle "no answer situations". CATS is currently deployed in Jordan by Fastlink (the largest mobile operator). Testing the content extraction component with a real noisy free texts shows a 90% F-measure.**

***Index Terms*—Spontaneous NL interface, SMS services, sublanguages, content extraction, classified ads, Arabic processing.**

I. Introduction

A natural language interface accepts users' inputs in natural language interacting with typically retrieval systems, which then results in appropriate responses to the commands or query statements. Hence, a natural language (NL) interface should be able to transform unrestrained natural language statements into proper actions for the system.

This type of unrestricted NL interface is an interesting choice because, if it could be built, it would offer many advantages. Firstly, it does not involve any learning and training, because its syntax 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: handling linguistic knowledge, and handling domain knowledge.

The study of the current scene shows that 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 restrictedness of the domain. Although most e-commerce activities are domain-specific, we did not yet find any e-commerce operational system offering an interface based on a restricted but natural *sublanguage*.

NL-based systems have the reputations of high development cost and low quality. Our goal in this paper is to show that the most important factor in building NL-based systems is the selection of adequate methods for the development, regardless of the targeted language, in terms of richness of resources, 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 describing the articles/goods they would like to sell or to search for, 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 a 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 paper is divided into three parts. The first describes the current scene concerning our assumptions and our proposed solution. In this part, we describe the main requirements of the proposed system, its main components, and its internal and external data specifications.

Manuscript received May 2, 2008. Manuscript accepted for publication June 18, 2008.

Daoud Maher Daoud is with Amman University, PoBox 141009, Zip Code 11814, Amman Jordan (Daoud@batelco.jo, Daoud.Daoud@imag.fr)

Christian Boitet is with GETALP, LIG, Université Joseph Fourier, 385 rue de la Bibliothèque, BP n° 53, 38041 Grenoble, cedex 9, France (e-mail: Christian.Boitet@imag.fr)

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

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

I. The Scene and Problems of Current Approaches

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

For example, the failure of MKBEEM [1] [2] to provide full spontaneous NL interface is due 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 providing 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 [3]. Those dependency trees are then processed and mapped into semantic representations, which are finally transformed into CARIN (an ontological representation). 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 analyses input texts syntactically [4, 5]. 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.

- As for MIETTA [6], it is also a multilingual system. However, it avoided the use of full natural language interface and only was used form filling interface and keywords processing.
- Similarly, TREE [7] avoided the use of full natural language interfaces and used form filling to interact with users in different languages.
- The HappyAssistant [8] prototype used a very limited NL processing for noun phrases only to provide NLDI.
- CASA [9] had also a form filling interaction style with keywords-based processing.
- Finally, GOOGLE SMS is 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 the inferring.

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 [10]. 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 paper will show that it is necessary and possible to build (multilingual) NL-based e-commerce systems for limited domains with mixed sublanguage and content-oriented methods.

II. A Corpus-based 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 in 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 [11]. However, systems such as CATS require deeper text understanding methods [12].

A. *The Scarcity of Data*

The shortage of data is one of the main obstacles in developing natural language systems. It is not easy to collect corpuses for restricted domain, especially if they must come from a very private medium of communication such as SMS.

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.

B. *Choice of a Web-based E-commerce Corpus*

In [13], it is found in experiment with different domains, that the best parsing performance was obtained for the same domain (religion, romance and love stories, etc.), followed by the same class (fiction or non-fiction), and the worst was obtained 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.

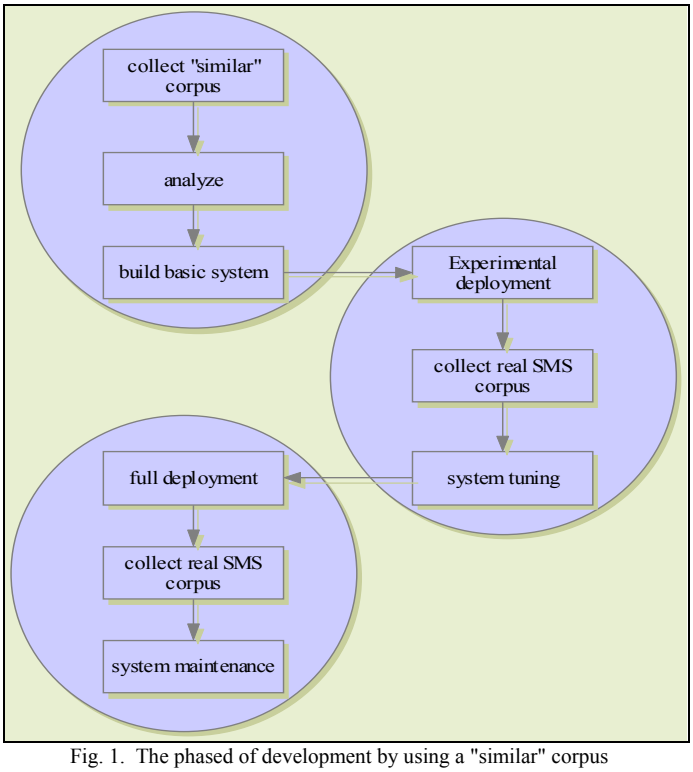

Fig. 1. The phased of development by using a "similar" corpus

As shown in figure 1, we can distinguish between 3 phases in a corpus-based development:

- Design phase and basic implementation: 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. Consequently, 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.

III. Sublanguage Analysis

It is noticeable that in restricted domains of knowledge, among certain groups of people and in particular types of texts, people have their own way of encoding their thoughts. Such restrictions can be said to reduce the degree of lexical and syntactic variation in text [14]. These specific languages are called either sublanguages or restricted or specialized languages.

As presented in figure 2, the analysis of the linguistic aspects and features of a sublanguage is needed to specify the sublanguage grammar (with the incorporation of the domain knowledge). Then general linguistic knowledge and sublanguage grammar can be used to determine the best NL technique to use. Similarly, the sublanguage grammar and the domain knowledge are both indispensable in selecting the best content representation.

A. *Typology of SMS-based Task-oriented Sublanguages*

To measure the lexical complexity of SMS-based classified ads sublanguage, we use the type-token ratio (TTR). This ratio increases with the lexical complexity and richness of the text and decreases if more words repeat themselves and the lexical complexity is lower. We calculated the TTR for different corpuses for the sake of comparison.

We measure the language complexity by the length of the sentence in words. 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 fragmentary is its style.

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

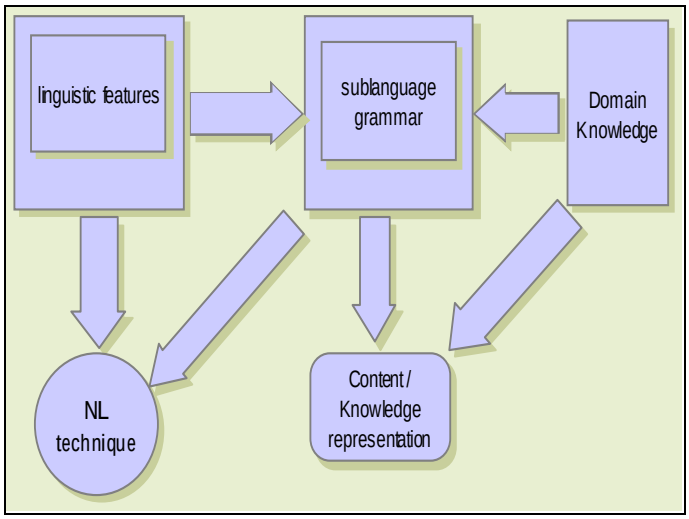

Fig. 2. NL development using sublanguage study

B. *General Corpus Statistics*

The SMS-based corpus consists of posts from Cars and Real Estate domains collected during a limited experimental period of CATS operation.

TABLE I
EXAMINED SMS-BASED CORPUS

Domain	Number of sentences	Sentence average length (words)	Types	Tokens	TTR
Cars	771	9	1181	5875	.201
Real Estate	641	12.5	1441	6182	.233

As it is shown in table I, 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. In other words, the user needs a lesser amount of words to encode his thoughts in the Cars domain than in the Real Estate domain.

When we compare SMS-based posts with Web-based posts, we find that the first are generally smaller than the second.

The findings also show that the least TTR value was for Cars at 0.201, then for Real Estate at 0.233.

The TTR values of Web-based posts 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 [15], suggesting a more topical diversity than that found in classified ads.

Additionally, the top 50 most frequent used words percentage in SMS-based Cars and Real Estate are 53.77%, 45.76% respectively. These findings suggest that as we move from Cars to Real Estate, the percentage of function words (such as prepositions) 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 Real Estate domain.

C. *Lexical Characteristics*

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).

We observe that some words in the Cars and Real Estate domains 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.

In the Cars domain, named entities are references to Car Makes and Models. In the Real Estate domain, they are references to Locations. The study of the corpus of classified ads shows that Named Entities consist of one or more words. 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.

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, number 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 and/or followed by unit words. 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"

There are many variations of spelling of the Arabic text in the studied corpus. For example, people write the Alef letter "ا", or with Hamza (ء) over it "أ" or under it "إ.". Also, we find 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. After some Arabic letters, people tend to wrongly insert a space, or to (also wrongly) omit it (e.g., "أبـو بكـر" or "أبوبكـــر" {Abu-Baker}).

The inconsistency of the Arabic spelling of transliterated proper nouns is also detected in the classified ads text where many of the proper names (car make and model as an example) are transliterated from other languages.

D. Syntactic Characteristics

The studied posts can have different syntactic structures caused by different word orders and grouping patterns of their constituents.

In some posts, we find that some constituents are not present because they do not interest the poster or are irrelevant for him, in cases 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 supposed 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.

E. Semantic Characteristics

We have shown that the syntactic structure for different posts which express the same information can vary enormously.

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".

The use of generalization in the query is also presented in the studied corpus. For example, the use of a generalization concept for searching is quite frequent 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.

F. The Main Outcome of Sublanguage Analysis

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 the following post:

"*Opel Astra station color red (power sunroof Center Electrical windows and mirrors check for sale*"

Similarly, techniques used for semi-structured text relying on position, layout and format of text are bound to fail on the given data.

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 structures for the description of the SMS.

This approach of describing sentences semantically achieves better results than using a pure syntactic description. It is also part of our engineering methodology, which allows semantic knowledge to be easily included in the system [16].

The study suggests also the need for a lexical lookup able to handle spelling variations as well as to store a concepts hierarchy.

Because of the information structure attached to this sublanguage, it is also necessary to have a content representation able to model the post, and to normalize the knowledge in a post regardless of its original surface structure.

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:

للبيـــع مـــنزل مســـتقل فـــي خلـــدا "For sale an independent house in Khalda" and that somebody sends the following query:

مطلـوب فيـلا فـي غـرب عمـان "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 matches 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.

IV. The CATS Architecture

The CATS 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 any pre-specified layout, recognizes the various relevant bits of information, 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.

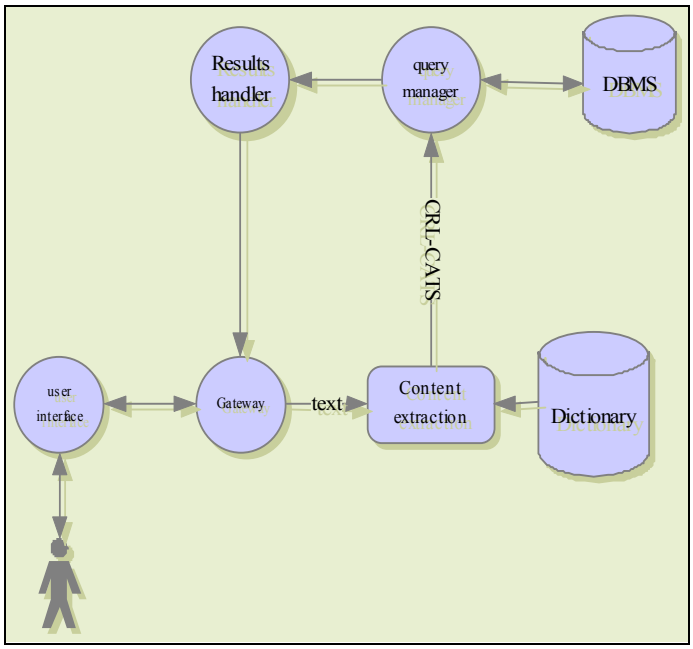

Fig. 3. Overall architecture of the CATS system

A. Overall Architecture

The overall structure of the CATS 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 then 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.

B. 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.

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.

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, feature, 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)

Here is a CRL-CATS expression encoding one classified ad post contains one or more CRL-CATS statements.[1]

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

[1] The labels ':00', ':06', ':0C', ':0U', etc. are identifiers associated by the DeCo engine to the "nodes" of the graphical representation, while the symbols 'sal', 'mak', etc. are labels on the arcs, created by the *grammar*.

For example, consider the following "sell" post:

للبيـــع ســـياره هونـــدا موديـــل 1997 جـــير اوتوماتيـــك مكيـــف ســـنتر بســـــعر 7750 دينـــار.

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]

In the above example, *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 (CATS word) *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.

V. Content Extraction in Arabic

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

1. 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.

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.

A. CE CATS Structure

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.

We have chosen to write our CE in EnCo [17] 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.

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.

B. Structure of the 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.

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 different word senses [18].

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.

C. 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.

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.

A Real Estate post 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.

For example, identifying relations between the MDO and the property values is 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. This is achieved by the DeCo rule:

<{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;

VI. The QA Component: Database Design, Semantic Matching and Response Generations

CE handled mismatches at 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 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 a light-weight inference storage that has other features such as reliability and concurrency?

A. Basic Implementation

During the past two decades, relational databases have been developed to a level that 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.

In relational database systems, data objects are normally stored using a horizontal scheme [19]. 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. Generally, CRL-CATS expressions are the source of the columns.

Additionally, the DB has to be designed to identify related concepts and to contain 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".

B. Implementing Semantic Matching

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 to ensure 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
.....						

Fig. 4. Scenario 2 implementation

As shown in figure 4, the system inserts values for "make" and "country", 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 with a higher performance.

C. Storing "Sell" Posts

For a "sell" post, the extracted information from the CRL-CATS and from other sources is passed to a stored procedure to generate the insert SQL statement.

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 prespecified columns to facilitate further semantic matching.

D. Processing of "Looking for" Posts

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.

E. *No Answer Situations*

We 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 [20]. 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'
```

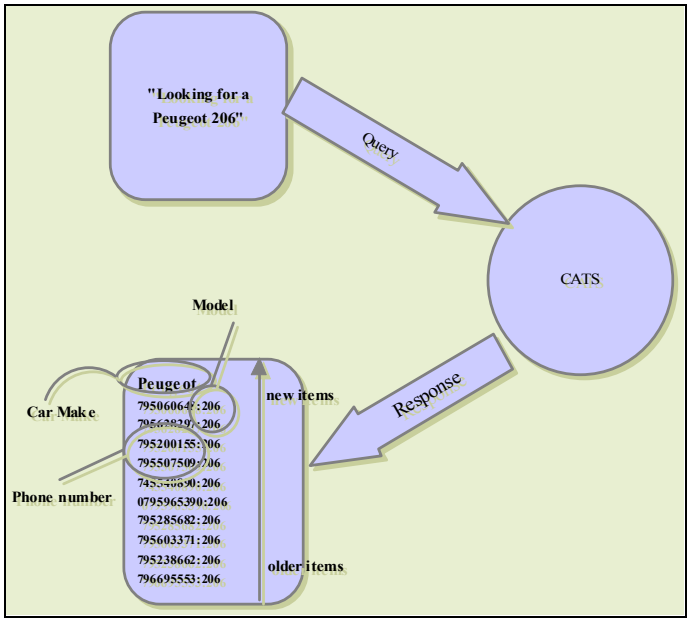

Fig. 5. Example of response in Cars

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 the "or" operator connecting these conditions.

In cases where no answer is found, even with this relaxation, the query is marked as *unanswered* by setting its SendFlag in the main table. A service (an agent) will periodically check at predefined time intervals the availability of any answer. As soon as an answer is found, it is then sent to the poster.

F. *Generating Responses*

Given the length constraint put by SMS, we used a tabular form to display the results as shown in figure 5.

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.

VII. Operational Experimentation, Evaluations and Discussion

A. *Status of the System*

This service is currently available in Jordan, where thousands of people have already used it to sell or buy 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.

B. *Evaluation*

Because the CATS system is targeting end users, we performed an end-to-end evaluation of the system by surveying users directly. We first explained the system to a sample of around 200 users from different backgrounds, and then asked 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. [21] 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 [22, 23]:

$$precision = \frac{\text{number of correct entities identified by the system}}{\text{number of entities identified by the system}}$$

$$recall = \frac{\text{number of correct entities identified by the system}}{\text{number of entities identified by a human}}$$

$$F - Measure = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

C. 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 posts received 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 II
RESULTS FOR THE TWO DOMAINS

Precision	92.7
Recall	87.2
F-measure	90

Table II shows the precision, recall and F-measures values for the Cars and Real Estate Domains.

We also remark that "looking for" posts show higher F-measure that "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 entities have lower F-measure than textual entities, suggesting that numerical entities are harder to detect or to identify correctly.

D. Assessment

In general, the results indicate that our content extractor performs well in identifying different parts of information. 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) [24].

For further assessment of our system, we compared our results with other more recent systems that use English: Phoebus [25], SimpleTagger [26], and AmilCare [27]. 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 [25] in two domains: hotel postings and comics books.

For the *price* entity CATS, scored a 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.

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 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. It is much better in comparison 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 [28].

VIII. CONCLUSION

We have shown in this paper, by surveying some e-commerce systems, that none of them handles spontaneous users' requests online. 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. 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.

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. Alight-weight ontology was implemented in the QA

We have shown that CATS is not like other experimental NL systems, because it was designed from the beginning to be a *production system*.

CATS is currently deployed in Jordan by the largest mobile operator (Fastlink) after passing intensive testing by its services. 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 methods to build similar systems. We can also explore other techniques 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.

We also plan to port CATS to other domains and other languages. Furthermore, CATS is being investigated for multilinguality by exploring different approaches the localization of similar applications. This work is part of research currently conducted at the GETALP group of LIG.

REFERENCES

[1] MKBEEM, "(web site)," 2005.

[2] J. Heinecke and F. Toumani, "A Natural Language Mediation System for E-Commerce applications: an ontology-based approach," presented at Workshop on Human Language Technology for the Semantic Web and Web Services,2nd International Semantic Web Conference, Sanibel Island, Florida, 2003.

[3] A. Lehtola, Y. KÄPYLÄ, C. BOUNSAYTHIP, and M. TALLGREN, "Multilingual and Ontological Product Cataloguing Tool – User Experiences," presented at eChallenges-2003 - Building the Knowledge Economy: Issues, Applications, Case Studies., Bologna, Italy, 2003.

[4] A. Lehtola, C. Bounsaythip, and J. Tenni, "Controlled Language Technology in Multilingual User Interfaces," presented at 4th ERCIM Workshop on "User Interfaces for All" Special Theme "Towards an Accessible Web", Stockholm, Sweden, 1998.

[5] A. Lehtola, J. Tenni, C. Bounsaythip, and a. K. Jaaranen, "WEBTRAN: A Controlled Language Machine Translation System for Building Multilingual Services on Internet ", vol. 2006, 1999.

[6] P. Buitelaar, K. Netter, and F. Xu, "Integrating Different Strategies for Cross-Language Information Retrieval in the MIETTA Project," presented at TWLT14, Enschede, Netherlands, 1998.

[7] H. Somers, B. Black, J. Nivre, T. Lager, A. Multari, L. Gilardoni, J. Ellman, and A. Rogers, "Multilingual Generation and Summarization of Job Adverts: the TREE Project," presented at Fifth Conference on Applied Natural Language Processing, Washington, DC, 1997.

[8] J. Chai, V. Horvath, N. Nicolov, M. Stys, N. Kambhatla, W. Zadrozny, and P. Melville, "Natural Language Assistant - A Dialog System for Online Product Recommendation," AI Magazine, vol. 23, pp. 63–75, 2002.

[9] X. Gao and L. Sterling, "Classified Advertisement Search Agent (CASA): a knowledge-based information agent for searching semi-structured text." presented at Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, 1998.

[10] R. I. Kittredge:, "Sublanguages," American Journal of Computational Linguistics, vol. 8, pp. 79-84, 1982.

[11] F. Benamara, "Cooperative Question Answering in Restricted Domains: the WEBCOOP Experiment," presented at ACL'04 Workshop on Question Answering in Restricted Domains, Barcelona, 2004.

[12] D. Moldovan, C. Clark, S. Harabagiu, and S. Maiorano, "COGEX: A Logic Prover for Question Answering," presented at HLT-NAACL 2003, Edmonton, 2003.

[13] S. Sekine, "The Domain Dependence of Parsing", presented at Applied Natural Language Processing (ANLP'97), Washington D.C., USA, 1997.

[14] J. Lehrberger, "Automatic Translation and the Concept of Sublanguage," in Sublanguage: Studies of Language in Restricted Semantic Domains, R. Kittredge and J. Lehrberger, Eds. Berlin & New York: Walter de Gruyter, 1982, pp. 81-106.

[15] A. Goweder and A. De Roeck, "Assessment of a significant Arabic corpus," presented at Arabic NLP Workshop at ACL/EACL 2001, Toulouse, France, 2001

.