An Arabic Language Interface to Databases Using a Morphologically-based Lexicon, Language Indicators and POS Tagging

Khaleel Al-Rababah
University of Bahrain

Safwan Shatnawi
University of Bahrain

Abstract

In this paper, we propose an Arabic Natural Language Interface to Databases (ANLIDB), The ANLIDB can respond to ill-formed questions submitted by users by bringing those questions to syntactically accepted questions. The ANLIDB also implements algorithms for extracting significant single and multiple phrases from Arabic natural language questions submitted to the database and then constructing and executing SQL questions. In this paper, we are dealing with Arabic language questions. An Arabic natural language (ANL) question is accepted as an input and then outputs all possible relations and its corresponding attributes. Arabic morphological, ontological, and syntactical analyses were applied in this paper. A lexicon derived from the database was created, and a simple part-of-speech (PoS) was implemented as well. The system shows high rates of success in identifying relations, correct mapping of attributes, and constructing and executing SQL statements.

Keywords: Morphology, Arabic Language, Syntactic, Lexicon, Natural Language, Finite State Automata, Stem.

1. Introduction

Organizations today use relational database management systems (RDBMSs) to hold huge volume of information. The process of retrieving information from such databases by filling in a form and setting too many parameters might not be the perfect way for computer- inexperienced employees to get the right information needed [2]. Natural language database interface would allow users to access data in a database by issuing questions in a natural language. The submitted natural language question is analyzed into a representation using lexical, syntactical, semantic, and linguistic knowledge. Automatic recognition of significant terms found in a natural language question plays a significant role in Natural Language applications such as machine learning, natural language interface to databases (NLIDB), and translation. The main objective of this paper is to respond to even ill-formed Arabic Natural Language questions, the response relies on bringing those questions to syntactically correct questions.

Arabic language is considered to be a Semitic language with richness in morphology. [9]. An Arabic word is classified as a particle, noun, or verb. Arabic language consists of 28 characters. Arabic is different from English in which nouns do not start with capital letters, which makes it a challenging task to recognize and extract proper nouns from text. Suffixes, prefixes and infixes play a role in creating different patterns of the same noun. Also, there are specific suffixes if added to a term; it could change the word from masculine to feminine. Also most written Arabic text suffers from the absence of diacritics which creates ambiguity and as a result complex morphological rules need to be applied to identify target nouns. Arabic language is known to be a high inflectional and derivational language which makes the morphological analysis very complicated [6].

This paper is organized as follows: In section 3 we presented related works, in section 4 domain description was given, section 5 gives a detailed description of system architecture, section 6 related to experiments and results, while section 7 is the conclusion and future work.

2. Related Work

Wissal Brini et al. [5] outlined a group of problems of identifications of proper nouns. The first problem is the lack of vocalization which can result in ambiguity. The second identified problem is lack of capitalization which makes it hard to identify named entities. The last problem is the delimitation problem which related to the lack of information about unknown words and the presence of homonyms which can increase ambiguity.

Mohammad Moinul et al. [10] stated that most common natural language questions submitted to a database can be identified by a small number of formulated structures. These structures can be outlined by finding delimiters of words that is important for formulating the question in a form like SQL (Structured Query Language).

Safwan and Rajeh [12] developed a generic, dynamic, and domain independent English language interface to database. A data synonym tree was developed which is based on a database under study, also they implemented a syntax state table in order to extract the SQL artifacts from a natural question submitted by a user.

Riyad et al. [6] used set of keywords and special verbs to identify Arabic nouns; the keywords are used to mark name phrases that might contain certain proper noun then a process is used to extract those nouns.

Mohammed et al. [8] presented an Arabic Retrieval System using Native language instead of using SQL. The paper shows a method for extracting tables and attributes from an Arabic natural language question; the method is basically based on generating all combinations of stems suffixes and prefixes of a given token and then search for the prefix in the lexicons of prefixes, search for stem in lexicon of stem, and then finally search suffixes in the lexicon of suffixes. Finally translate the match into Buckwalter code. The translated word is then mapped to a relation or an attribute.

Faraj A. El-Mouadib et al. [13] demonstrated an Augmented Transition Network technique to extract nouns from a sentence; first it checks that tokens structures is in allowable grammatical structure using a parser based on Context-Free Grammer and then extract the phrases from parser nodes.

[9] Showed that a database is made up of three types of elements: relations, attributes, and values. Also, it outlined that each element is distinct and unique. Also it noted that many natural language questions specify a free-standing values, where the attribute is implicit. Finally it specifies that each token matches a unique database element.

[14] highlighted the possible problems that could happen when translating a question noun into its correct relation or attribute; those problems correspond to M-o-1 and 1-to-N mappings between linguistic expressions and both relations and attributes.

3. Domain Description

The following ERD shows the domain under study

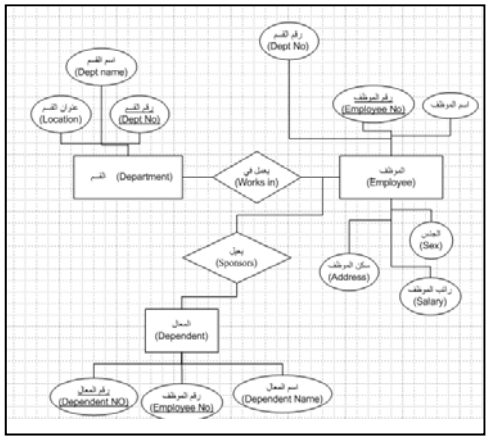


Figure 1. ERD Diagram

4. System Architecture

The system will receive a Arabic question in a natural form and then carrying out the processing as depicted in figure 2.

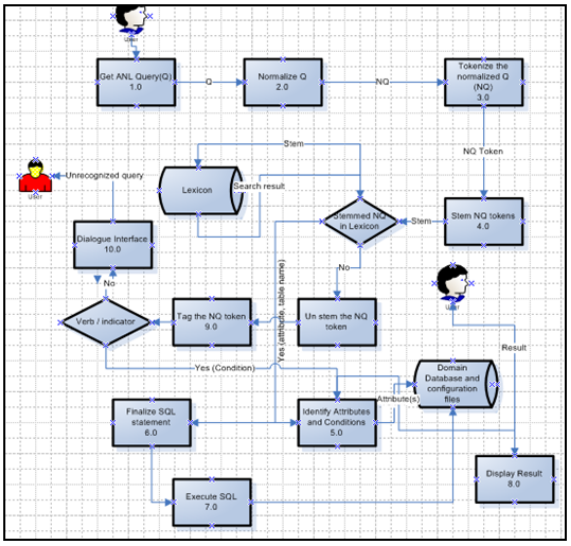


Figure 2. System Architecture

The system is composed of the following main components:

A. Lexicon construction

The lexicon contains all ANL words related to the domain under study; these words describe the domain; they are obtained by analyzing the domain's database schema elements which could be in a form of relations (tables), attributes, or relationships among relations. The lexicon is considered to be the backbone for the proposed system. Two approaches are used to construct the lexicon: pattern-morphology approach and morphologically-based stemming. The two approaches were used to shed light on the effect of using different lexicon construction methods on the accuracy and performance of ANLI systems. The following will describe the approaches used to construct the lexicon.

In pattern-morphology approach, each word that contributes to the description of the database artifacts is selected and stored in the lexicon along with all possible morphological patterns, also we store all words produced by adding prefixes and suffixes to those word's patterns as a result each word used to describe the domain will generate hundreds of lexicon words; table1 contains example of one word used and sub set of the generated lexicon's words; in addition to that for each word we find all possible synonyms, with each synonym we applied the

process previously mentioned as in table1, table 2 shows some synonyms for one of the words.

Table 1. Database artifact and possible lexicon's words

Word	Pattern	Suffix	prefix	Generated word
سكن	فاعل	-	-	ساكن
	فاعل	هـ	-	ساكنه
	فاعل	تهـ	-	ساكنته
	فاعل	هم	م	مساكنهم
	فعل	-	ي	يسكن
	فعل	-	تـ	تسكن

Table 2. Database artifact and possible synonyms

Word	Synonyms
سكن	مكان, عنوان, إقامة
موظف	عامل, أجير, مستخدم

In morphologically-based stemming lexicon approach, each database artifact is stemmed; we used Shereen Khoja Stemmer which uses both rule-based and database search approaches to stem Arabic words, the stemmed word is stored in the lexicon, also all synonyms of the artifact are stemmed and stored in the lexicon as well, as a result the search space will be reduced by using stemmed words.

B. The relation matcher

The component is responsible for finding relations according to the following logic:

```
int RelationFinder (string token )
{
for ( i=0; i < lexicon_size;i++)
    If (token == lexicon(i) )
Num_of_Relations++
return Num_of_Relations ;
}
```

Figure 3. Relation finder

If the Num_of_Relations = 0 then we still need check for attributes as we will see later, but if number of relations and attributes =0 then you prompt the user to enter another ANL question. For example, the ANL question "أعطيني أعلى راتب" which means "Give me the maximum salary", here no explicit relation found in the question but still the attribute "راتب" indicates which relation does it belong to which in our case it belongs to Employee relation

Example1: العاملين وأسماءهم وعناوينهم وأرقامهم ورواتبهم is a submitted ANL (Employees and their names and their addresses and their numbers and their salaries) Each token will be matched against the entries in the lexicon, the following shows part of the lexicon where it shows different morphological forms of Table الموظف(The Employee) as well as synonyms for الموظف

Table 3. Lexicon

Morphological forms of الموظف (The Employee)	Synonyms for الموظف(The Employee) and sample of the variations of each synonym
"الموظف", "موظفين", "الموظفين", "الموظف", "الموظفون", "موظف", "الموظفات", "موظفه", "موظفة", "الموظفه", "الموظفة",	"العاملات", "العاملين", "عمال", "العامل", "عامل", "صاحب", "مستخدمه", "مستخدم", "مديرات", "مدراء", "مدير", "أفرد", "شخص", "مسؤولين", "مسؤول", "مستخدمون", "أفراد", "اجير", "النسوه", "", "النساء", "الرجال", "المرأه", "الرجل", "", "", ""

Since Arabic language supports free order ANL question, the question in Example1 can be submitted in different orders, one form are shown below: أعطيني أسماء وأرقام وعناوين الموظفين ورواتبهم. (Give names and numbers and addresses for employees and their salaries)

C. Attributes matcher

This component identifies attributes. Identifying attributes can be a challenging task due to the following reasons:

Challenges of identifying attributes

- A user can enter ANL in a free order, the following gives two equivalent ANL questions that can mapped to the same database elements but with different order, Example1: الموظف واسمه وعنوانه ورقمه means Employee and his name and his address and his number.

  Example 2: اسم الموظف وعنوان الموظف ورقم الموظف which employee name and employee address and employee number. The character "و" spelled "wa" in Arabic and means "and" in English as in "واسمه" it serves as a conjunction pronoun, and one of the possessive pronouns such as "ه" or "h" in English as in " رقمه" which means “his number” whereas "رقم without “h” at the end means just رقم “number” in English

- The ambiguity in identifying attributes as shows in the following example: أعطيني اسم الموظف عنوانه رقم القسم رقم الموظف رقم المعال here the token "رقم" occurred three times; since this token can be found in all of the three tables in the database which poses a challenge to map it correctly to the correct relation.

Cases for identifying attributes

Case 1: Mapping attributes to one relation found in the ANL submitted question.
Here, one relation is identified, the following example illustrates this:
"أعطني معلومات عن الاجيرين الذين رواتبهم أكبر من 100"
which means "Give information about employees whom their salary greater than 100"
The token "الاجيرين", which is a synonym to "الموظفين", will be mapped to "الموظف" relation.
Another example, أعطني معلومات عن موظف رقمه 1000 وعنوانه واسمه وراتبه (Give me information about employee his number is 1000 and his address and his name and his salary. Token موظف mapped to الموظف table, token رقمه mapped to رقم الموظف, عنوانه to سكن, اسمه to اسم الموظف, and راتبه to راتب الموظف.

Case 2: More than one relation is presented in the ANL question along with more than one attributes. Since relations might have the same attribute names and because of the ill-formed submitted ANL questions, we need to solve any ambiguity in mapping the attribute(s) to the right relations.

The paper used the following logic to map tokens to attributes in the correct relation. Of course, this logic applied after identifying the relations.

```
for ( int i=0; i < token_count;i++)
{
for (int j=0; j< lexicon_size;j++)
      if (token[i] == lexicon[j])
            attrib_counter++;
if (attrib_counter > 1)
if     (token[i]     starts     with
conjunction   و   (wa)   &&   token[i]
ends with a possessive pronoun)
relation                              =
most_recently_identified_releation;
else
relation = closeness_relation_rule;
}
```

Figure 4. Attribute finder

The following two examples demonstrate this logic
Example: given the following ANL أعطيني رقم المستخدم عنوانه واسمه وأسم القسم
الموظف and القسم relations are both identified.

Next, we need to identify attributes, we start with token رقم (number) . Searching the lexicon, it appears that this token exist in the tables: Department, Employee, Depdendent, According to the logic, we need to test whether starts with و (wa) or ends with ه (h) , but since the token neither starts with و (wa) nor ends with possessive pronoun, so we use the closeness neighboring rule ;so a result the token رقم is mapped to relation الموظف as رقم الموظف.

The remaining attributes are mapped as follows:
Token عنوانه is mapped to Employee relation, according to closeness rule, it is mapped to "الموظف" Employee relation.
The token واسمه has the same closeness to both relations, but because the token starts with the conjunction "و" which serves as a continuation marker of more attributes and more importantly the possessive pronoun ه (h) which indicates that this attribute belongs to the same relation as the previous attribute; so the this attribute will be mapped to "الموظف" relation.

The last token ورقم (wa number), as it starts with conjunction "و", but it doesn't have the possessive pronoun ه (h) at the end; so again the closeness neighboring rule is applied and as a result it is mapped to the القسم (department) relation as رقم "القسم". In the next example we will extend the number of relations to be three. أعطيني اسم الاجير وراتبه ورقم العال ورقم القسم. All attributes will be mapped to the correct relations according to the rules laid out in the logic above.

Case 3: Identifying implicit attributes
There might be situations where no explicit attributes exist in a submitted ANL question.
أعطني معلومات عن الموظف 100
No tokens can explicitly be mapped to attributes. The only alternative is to try to identify attributes through the values which comes after relations. So, the number 100 represent a value of an attribute in الموظف , the question is which attribute of this relation will be mapped to? We suggest to have two defaults attributes for each relation, one for text data type and one for number data value. For الموظف relation , we assign رقم الموظف as representative default attribute with number data type and اسم الموظف as a default as an attribute with text value. So, 100 will represent رقم الموظف attribute. The ANL question أعطني معلومات عن الموظف 100 will be expanded to أعطني معلومات عن الموظف الذي رقمه 100 .

D. Condition identification and SQL construction and execution

Several techniques are being approached in this paper to identify the SQL statement parts such as POS and relative pronouns. The POS will primarily be used in identifying verbs; this will serve two purposes: the first purpose will be as a marker that separates the main SQL condition clause from the from clause, while the second purpose will be identifying possible relationships in the ANL question.

- **Tagging**

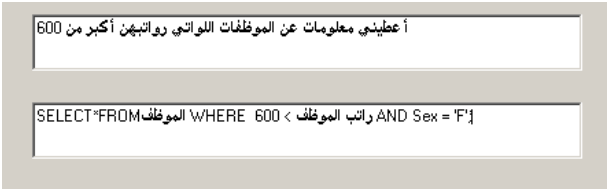
In order to carry out the tagging, a lexicon of all possible Arabic language patterns are collected, a sample of those patterns are shown below:
"فعل", "يفعلون", "ستفعلن", "ففعلت", "ففعلن", "فيفعلان", "وستفعلان", "تفعلن", "فاعلن", "ستفعلان", "فسيفعلون", "وستفعلن", "ففعلتا".
Tagging is the process of assigning labels for Arabic words; in order to tag a word the algorithm finds all patterns of lexicon that has the same length as the token. For each pattern found in before, find the number of identical letters between the token and the pattern and then locate the pattern which has the maximum number of identical letters with the token. The pattern will indicate the token's tag. [reference 0]
For example, "" the token "فكتبن" will have the pattern "ففعلن".
Another example, the verb "يعملون" in the ANL question,
"أعطيني معلومات عن الموظفين الذين يعملون في قسم المحاسبة" will serve as "work" relationship between the "Employee" and "Department" tables.

- **Relative pronouns**

Relative pronouns can identify nouns besides also it can serve as a delimiter which divide the ANL question into two parts: the first part is the SQL type along with the relations and required fields; in the second part, which comes after those indicators is the condition part . Examples of those pronouns are : "الذي", "التي", "الذين" , "اللواتي","اللذان"which correspond to who, whom in English . Nouns, if exist, directly precede those pronouns as shown in the following two examples:

1. أعطيني معلومات عن الموظفين الذين يعملون في قسم المحاسبة which is equivalent in English to "Give me information about male employees who work in Accounting department". It is obvious the existence of the pronoun "الذين" in the sentence which comes right after the word "الموظفين".

2. The ANL question أعطني معلومات عن الموظفات اللواتي رواتبهن أكبر من 600" which is equivalent in English to "Give me information about female employees whom their salary is greater than 600. The existence of pronoun "اللواتي" serve two things: an identifier for the noun Employees "الموظفات" and this noun is female.



- **Condition Construction**

Once SQL condition part is identified, the next step is constructing the condition. To construct the condition, condition operators will be identified, table 4 shows the possible operators along with their synonyms and Figure 4 shows different scenarios for constructing such conditions.

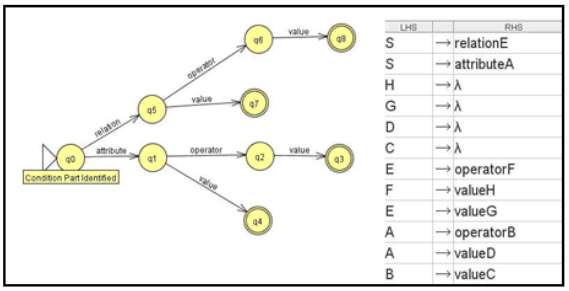


Figure 4. Finite State Automata for condition construction and its grammar

Condition operators might be a single phrase operator or a multi-phrase operator. The following table shows possible condition operators that may occur in any Arabic Natural Language question.

Table 4. Relational Operator

Meaning	Operator
تزيد (feminine)	greater >
يزيد (masculine)	greater >
يزيد عن	greater than >
أكبر من	greater than >
أقل من	less than <
أعلى من	>
تزيد عن	greater than >
لا يزيد عن	< less than
لا تزيد عن	<= less than
أقل أو يساوي	<= less than
اقل او يساوي	<=
أكبر أو يساوي	>= greater than or equal
اعلى او يساوي	>= greater than or equal
لا يقل عن (not less than) masculine	>

As shown in Table 4, "يساوي" is a single phrase operator, "لا يساوي" is a two-phrase operator, and "لا يزيد عن" represent a three-phrase operator.

Besides, we identified semantic interpretations for most operators, for example the operator أكبر أو "

"لا تقل عن", "لا يقل عن", "أعلى او يساوي" could be يساوي " عن" .

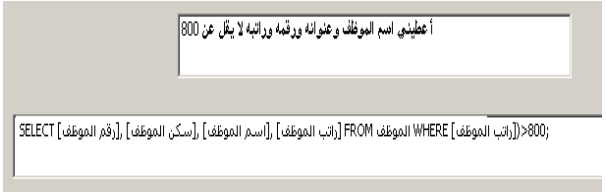
Locating a value or a string condition that comes after an attribute

Here we search for conditions around explicit attributes; the value that comes after a condition is either a string or a number. The logic, of course, will verify that the token comes after the attribute token is not an attribute or a relation. The logic will also ignore the following words that might occur in any ANL question and proceed searching for a condition operator. Those words are shown in the following array named .Ignore. Ignore() = [,"في", "ذا", "ذو", "ذي", "فيه", "به", "من", "ذوو", "فيها", "بها" "فيهم", "بهم", "", "فيه", "به", "من"]

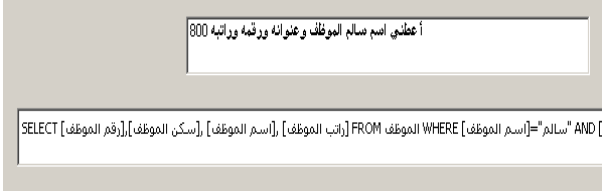
Here we discuss many possibilities:

a. It is possible that an attribute is directly followed by one of the following [,"الذي", "الذين", "اللذان" "اللواتي", "التي"], The following example explains this:
أعطيني رواتب الموظفين الذين تزيد رواتبهم عن 1000, the phrase " رواتب الموظفين " will be mapped to the database field "راتب الموظف", the phase "الذين" indicates the beginning of a condition which means "WHERE.". The condition is followed by phrase "تزيد" this phrase indicates the existence of an operators in the Table 4, but checking the following token it was different from it was expected to be "عن" ; our logic handles this and brings the ANL question to the form:
أعطيني رواتب الموظفين الذين رواتبهم تزيد عن 1000

b. The condition operator is preceded by an attribute and followed by a value.
Example: أعطني اسم الموظف وعنوانه ورقمه وراتبه لا يقل عن 800
Also, The following shows the ANL question and its constructed SQL statement:

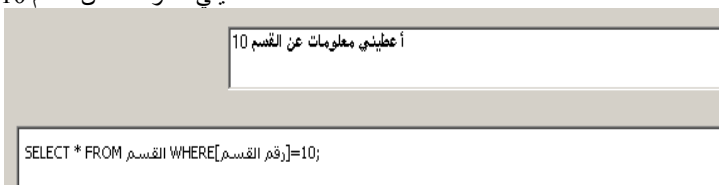


c. The third case is where the attribute is directly followed by a value with a hidden condition operator. In this case, the hidden operator will be considered as equal "= ". The following example shows such a case:
Example: أعطني اسم سالم الموظف وعنوانه ورقمه وراتبه 800



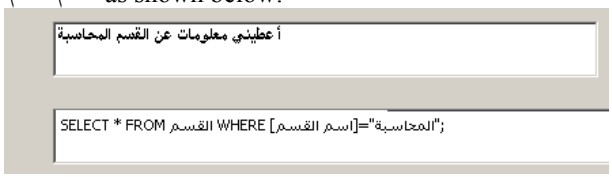
As you notice the example represents an ill-formed ANL question where it exhibits syntactic problem where the correct syntax for it is:
أعطني اسم الموظف سالم وعنوانه ورقمه وراتبه 800

d. This case discusses the possibility that no attribute is explicitly mentioned in the ANL question. We handle this case by searching for values that follow an identified relation, the following example shows such a case :
أعطيني معلومات عن القسم 10

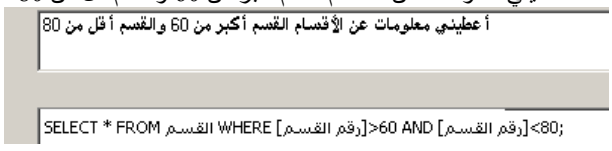


Here we need to decide to what field does the value belong, since no field was mentioned, we suggest two default fields for each relations, one for field with number value and the other for a text value. For "الموظف" relation we select two fields: "رقم الموظف" attribute as it is of type number and "اسم الموظف" as it is of type text. So the value 10 was assigned to "رقم الموظف" attribute.

Assume the question was أعطيني معلومات عن القسم المحاسبة then the value "المحاسبة" will be assigned to "اسم القسم" as shown below:

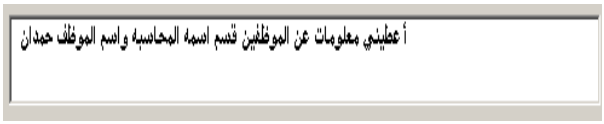


Case 4: A relation is followed by an operator then a value. The following example shows this case beside it is an example of ill-formed question:
أعطيني معلومات عن الأقسام القسم أكبر من 60 والقسم أقل من 80



As mentioned above, our system handles complex, ill-formed, and free-order ANL questions. The following question represents an ill-formed ANL question:
أعطيني معلومات عن الموظفين قسم اسمه المحاسبه واسم الموظف حمدان

Processing the question, we brought it to a syntactically correct form:
أعطيني معلومات عن الموظفين ذوو الاسم حمدان ويعملون في قسم المحاسبه

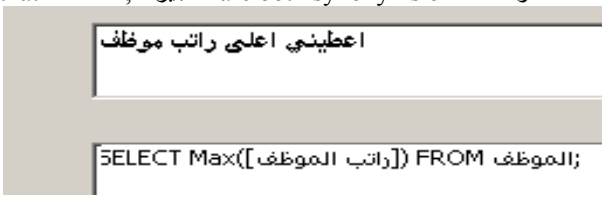



```
SELECT [اسم الموظف], [رقم الموظف], [سكن الموظف] FROM القسم INNER JOIN الموظف ON [رقم القسم].القسم = [رقم القسم].الموظف
WHERE [اسم الموظف]="حمدان" AND [اسم القسم]="المحاسبة";
```

Another example which can hold ambiguity, " أعطيني رواتب الموظفين التي تزيد عن 50" , here just one attribute "راتب الموظف" was identified in the question and no other attribute was identified in the where clause which was recognized by the existence of the phrase "التي", the sentence in English is "Give me Employee's salaries which exceeds 50". Ambiguity here rises because it is not totally clear to what 50 refers to, but since one attribute is mentioned in the ANL question then the ANL question was brought to a syntactically correct question as in the following: أعطيني الموظفين الذين رواتبهم تزيد عن 50 which in English is "Give the Employees with salary greater than 50".

Aggregate Functions

Aggregate function phrases are recognized through phrases containing words like "مجموع"which means sum. For example, "معدل , "متوسط" are both refer to Average. "اعلى", "اكبر", "اكثر" are all mean Maximum. Both "اقل" and "ادنى" mean Minimum
For example, the ANL questions:
All the following ANL statements, "أعطيني اكثر راتب موظف", "أعطيني اكبر راتب موظف", "أعطيني اعلى راتب موظف", "اعطيني اعلى راتب اجير", "أعطيني اكبر راتب عامل موظف", "" means "Give me the maximum salary of an employee", are all equivalent ANL statements which give the same SQL statement, SELECT MAX(راتب الموظف)FROM الموظف. As it was mentioned before that " عامل ", " اجير " are both synonyms of " موظف "



5. Experiments and Results

In the following, we will demonstrate cases where generation of SQL statements were wrong because of inaccurately identifying either attributes, relations ,or conditions followed by table 5 that demonstrates this through forming 75 ANL questions.
The following example shows where the system (stem based approach) failed to extract the right stem, the result was a construction of a wrong SQL statement.

"أعطني وظيفة الموظف سالم" which means "Give me the job of employee salem", the stemmer will take the first letter "و" as prefix so when the stemmer strip it the remaining characters in the word "ظيفة" will generate a wrong stem. Another example, the token "عنوان" which means "Address", the problem that could arise is that the last two characters "ان" will be considered as suffix which indicates dual whereas they are original characters and losing them will produce a new word which has a different meaning from the original and as result will produce a wrong stem.

If a specific and known domain is to be used then adding new rules to the stemmer can overcome such problems, but if we are looking for portability and the system to be applied to any domain that might dramatically degrade the accuracy of the system.
Compared with morphologically- based lexicon, this approach has a smaller lexicon search space. When it comes to accuracy, the accuracy of finding the right relations and its corresponding attributes depends on the accuracy of extracting the correct stem. By removing additives (prefixes, suffixes, infixes) we increase ambiguity. Moreover, stemmers may generate wrong stems.

Table 5. Proposed System Evaluation

	Stem based lexicon			Morphological basic stem		
SQL Part	Identified	Unidentified	Why failed	Identified	Unidentified	Why failed
Attributes	60	15	Stem_A_1 Stem_A_2	67	8	Morph_A_1 Morph_A_2
Relations	68	7	Stem_A_3	70	5	Morph_A_3.
Conditions	62	13	Stem_A_4	68	7	Morph_A_4

Where:

Stem_A_1.Generation of wrong stems when removing original characters of a token as a result of removing prefixes, suffixes
Stem_A_2. Removing the possessive pronouns during stemming which confuses the closeness rule.
Stem_A_3. System ignore some tokens that are found to be synonyms to one of the domain relations due to the fact our lexicon doesn't include all synonyms of a database artifact
Stem_A_4.Stemming a value can generate a new value which as a result constructs a different condition
Morph_A_1. A synonym of an attribute was not in the lexicon

Morph_A_2. Attributes are wrongly assigned to the wrong relation do to ill-formed and free order questions and the result the closeness rule failed to correctly assign it.

Morph_A_3. System ignores some tokens that are found to be synonyms to one of the domain relations due to the fact that our lexicon doesn't include all synonyms of a database artifact.

Morph_A_4. Some tokens were identified as attributes instead of values which prevented the construction of a condition

There are situations where a token can be interpreted as an attribute instead of a value, for example, "أعطيني معلومات عن الموظف راتب حمدان"; here, the token "راتب", which means "Salary" in English, can take the meaning of an attribute name or a value in Arabic. So, according to our domain, it could mean Employee name or it could be an attribute name in the Employee relation.

Removing suffixes will complicate the process of identifying relations since the possessive pronouns will be removed during stemming as a result the closeness rule may not correctly capture the right relation for the attribute. Sometimes these suffixes add implicit condition to the SQL condition part for example

"استرجع معلومات الموظفات" stemming "الموظفات" produce "موظف" which will result in retrieving the information of all employee, while the query asks for female employees only, this since the suffix "ات" is added to the female words in the Arabic language – which implicitly add "where sex = "female" to the SQL condition part

Results in Figure 5 shows that the system performs much better with "Morphological basic stem" than using "Stem based" approach. Also, it shows that, using the two approaches finding and matching the correct attributes is more critical than identifying the right relations.

Moreover, constructing the wrong condition in both approaches can dramatically degrade the system performance.

The system performs very well executing free order ANL questions. The system efficiently handles the case of implicit attributes it checks that the value comes after a relation will not be a relation or an attribute.

6. Conclusion and Future Work

This paper presented a technique to translate ANL questions submitted to databases of specific domain into SQL statements by employees of little experience in using SQL statements. The system uses a lexicon extracted from a domain under study. Morphological, syntactical and ontological Arabic language characteristics are all used in the system .Certain language markers are served as separators to locate nouns and also to locate parts of SQL statement. Also, a simple POS is created to recognize verbs which can serve as a relationship indicator between relations found around the verb.

The system addresses and handles the following language characteristics:

- The multiplicity of the syntactic interpretation of the same word. Reducing different forms of an element to a canonical form which helps reducing complexity. Handling implicit and explicit attributes and associates those with the right relation. Recognizing markers and separators to recognize nouns and verbs. Handling the free order and ill-formed ANL questions
- The system shows higher percentages of success in constructing and executing SQL statements.

7. Acknowledgment

Thanks goes to deanship of scientific research in University of Bahrain for supporting the work in this paper.

8. References

[1] Emad Al-Shawakfa et al., A comparison study of some Arabic root finding algorithms, Journal of the American society for information science and technology, 2010.

[2] Processing of natural language queries to a relational database, M. Samsonova, A pisarev and M. Blagov, Bioinformatics 19(Suppl. 1) Oxford University Press 2003

[3] Khaleel Al-Rababah, Abdulmahdi Al-Rababah, Safwan shatnawi, Identifying Significant Single Phrases in Submitted Free - Order Arabic Natural Language Questions , IEEE 2011, London, UK, in press.

[4] Daniel Jurafsky, James H. Martin, Speech and Language Processing, Pearson International Edition, 2009, ISBN-10: 0-13-504196-1

[5] Wissal Brini et al, An Arabic Question-Answering system for factoid questions, 2009 IEEE International Conference on Natural Processing and Knowledge Engineering (IEEE NLP-KE '09) Special Session On Arabic Language Processing.

[6] Riyad Al-Shalabi et al, Proper Noun Extracting Algorithm for Arabic Language, International Conference on IT to Celebrate S. Charmonman's 72nd Bithday March 2009, Thailand.

[7] Hasan Al-Serhan and Aladdin Ayesh, A trilateral word roots extraction using neural network for Arabic, 1-4244-0272/7/06. IEEE 2006.

[8] Mohammed Otair, Raid Al-Sardi, Salah Al-Gialain, 978-1-4244-2624-9/08, 2008 IEEE.

[9] Towards a theory of Natural Language Interfaces to Databases, Ana-Maria Popescu, Oren Etzioni, Henry Kautz, IUI ‘ 03 Miami, Florida USA, ACM 1-58113-586-6/03/0001.

[10] Isolating Significant phrases in common natural Language queries to databases, Mohammad Moinul Hoque, Md. Shahriar, S.M. Abdullah Al-Mamun, IEEE, 2008, 1-4244-2136-7/08.

[11] Eiman Tamah, Improving Arabic document categorization: Introducing Local stem, 978-1-4244-8136-1/10, IEEE 201.

[12] Safwan shatnawi, Rajeh Khamis, An approach for developing natural language interface to databases using data synonyms tree and syntax state table,

[13] Faraj A.El-Mouadib et al, Generic interactive natural language interface to databases, international journal of computers, Issue 3, volume 3, 2009.

[14] In-Su Kang et a;, Lightweight natural language databases interface, NLDB 2004, lncs 3136, pp. 76-88, 2004 Springer-Verlag Berlin Heidelberg 2004.

[15] Shihadeh Alqrainy, Hasan Muaidi AlSerhan, and Aladdin Ayesh, Patteren-based Algorithm for Part-of-Speech Tagging Arabic Text, IEEE 2008.

