A Rule-Induction Approach for Building an Arabic Language Interfaces to Databases

Hanane Bais
LAMIGEP, EMSI Marrakech,
Morocco
H.BAIS@emsi.ma

Mustapha Machkour
Department Computer Sciences, Ibn Zohr University,
Morocco
machkour@hotmail.com

Abstract: *In the field of Natural Language Interfaces for Databases (NLIDB), most of the solutions considered for translating natural language queries into database query language is based on linguistic operations. The application of these operations makes it possible to translate the natural language queries into an unambiguous logical interpretation. However, this task is extremely complex and requires excessive time. While nowadays emphasis is placed on the use of machine learning approaches to automate the operation of natural language processing systems. From this, the automation of the natural language queries translation process into a logical interpretation is interesting and remains a major challenge in the field NLIDB. Also, it can have a direct impact on reducing the complexity of the operation of NLIDB. In this study, we focused on applying a new approach to automate the operation of NLIDB. In this approach, we applied a supervised learning technique to induce rules that transform natural language queries into unambiguous expressions.*

Keywords: *Rule-induction, databases, machine learning, intelligent interfaces, arabic language processing.*

Received June 7, 2020; accepted October 10, 2021
https://doi.org/10.34028/iajit/20/1/6

1. Introduction

Natural Language Interface for the Database (NLIDB) [3] is one of the classic problems in the field of Natural Language Processing (NLP) [8]. The main aim of NLIDB is to improve the interaction between human and database by help non-expert users to extract data from the database without the need to have expertise in database query languages such as Structured Query Language (SQL) [1]. In this sense to access data stored in databases, users need to write a simple query in Natural language [1, 9]. Until now, most of the NLIDB uses linguistic operations to translate natural language queries into unambiguous logical interpretation [14, 12]. However, using linguistic expertise for developing the NLP system is still an extremely difficult and complex task.

A promising approach to automate the constructing of NLP system is to use machine-learning approach. For that, the automatic mapping of natural language queries into an unambiguous logical interpretation is a major and interesting challenge in the field of computer linguistics. In this paper, we discuss the problem of translating Arabic Natural Language Queries (ANLQ) into Database Query (DBQ). This is done by the induction of rules that automatically transform ANLQ into XML Logical Query (XLQ) expressed in Extensible Markup Language (XML) form.

To inducing the transfer rules (ANLQ-XLQ rules), our system trains a parallel corpus of ANLQs paired with their logical queries. The main advantage of this approach is that it allows our system to learn rules that help it to significantly reduce execution time and simplify the translation process from ANLQs into XLQ. This is attained while maintaining the performance achieved in linguistics operations.

The rest of this paper is organized as follows. We begin with a brief description of our proposed approach. Then, the operating process of this approach is presented. Then, the results of the experiments carried out are discussed with a comparison with the results obtained by the application of linguistic operations. Finally, we conclude with a conclusion.

2. Related Works

Research in the field of NLDBI has appeared in the late sixties and early seventies [6]. Since then, most of the created NLIDB use linguistic expertise to translate NLQs into a logical interpretation [5, 6]. However, the use of the Linguistic approach usually requires time. Also, the hand-crafted parser suffers from problems with robustness [15].

The application of A Rule-Induction Approach (RIA) for the development of NLP systems is not a recent subject [11]. RIA has been used in many areas of NLP, such as machine translation [13], Information Extraction [10], spoken language understanding [4] and Sentence Reduction [7]. Regarding the application of Rule-induction in the construction of NLIDB there a few contribution.

The first one is the Chill system [16]. Chill is a parser acquisition system that constructs an automate natural language interface for database queries. Chill processes a parser acquisition as the learning of search-control rules within a logic program representing a shift-reduce parser. It applies techniques from inductive logic programming to learn relational control knowledge. Starting with a general framework for constructing a suitable logical form, Chill can train on a corpus including a set of example pairs (sentences paired with their database queries). Then it induced parsers that map subsequent user query directly into executable queries. In the second work, Lappoon proposes a method that uses a machine learning approach to semi-automate the construction of NLIDB systems [14]. He applied a "meta" inductive logic programming learning approach that joins the strengths of different learners, to induce a semantic parser. This method performs better than using a single learner.

3. Methods

The proposed architecture is composed of two parts. The first one is the induction rules module. The second part is the translation based induced rules. This architecture is shown in Figure 1.

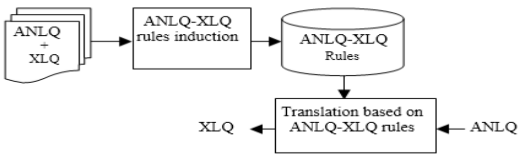


Figure 1. Rule-induction approach.

The main advantage of this approach is that it provides a logarithmic growth of the rules generated over the execution of the system which ensures its consistency. This is proven by the experiments.

4. ANLQ-XLQ Rules Induction

In this part, we discuss the learning process used to induce the ANLQ-XLQ rules from pairs of ANLQ and XLQ examples. Figure 2 illustrates this process.

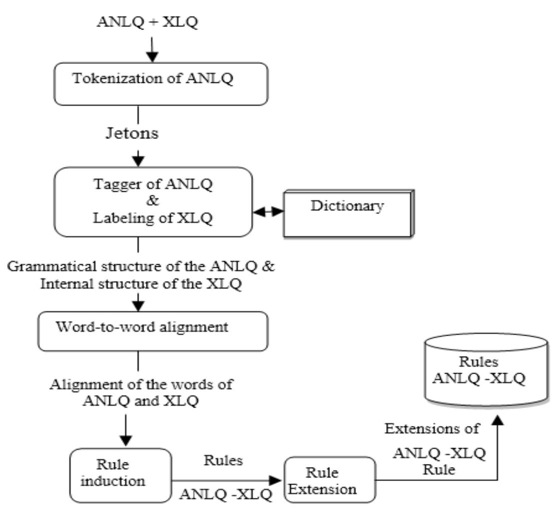


Figure 2. The process used to induce the ANLQ-XLQ rules.

As previously discussed, the ANLQ-XLQ rule induction process takes ANLQ coupled with XLQ as input and produces the ANLQ-XLQ rules as output. In this process the system uses many approaches of Arabic language processing [2]. The algorithm that explains the operating principle of this process is described as follows:

Algorithm rules _ induction

Inputs:
ANLQ Queries in Natural Language
XLQ its xml logical query
Outputs:
Rule set RL = $\{(R_I),\ 1 \leq i \leq m\}$
Start
Divide ANLQ into a set of $W = \{(t_j\colon GC_j),\ 1\leq j \leq m\}$
where GC_j *where* GC_j *is the grammatical category of the token* t_j*;*
Divide XLQ into a set of fractions $F = \{(f_k, E_k),\ 1\leq k\leq l\}$
where E_k *is the label of the part* f_k*;*
For each token $(t_j\colon GC_j) \in W$
For each fraction $(fk\colon Ek) \in F$
If $t_j = F_k$
Replace t_j *by W[j] en* $(t_j\colon GC_j)$*;*
Replace F_k *by W[j] en* $(F_k\colon E_k)$*;*
End if
End For
End For
Create a new rule R_I*:*
$$\sum_{j=0}^{n} (W[j] : GCj) \rightarrow \sum_{k=0}^{l} (W[j] : Ek);$$
$RL = \{R_I \cup \text{Extensions of the } R_I\}$*;*
Return RL;
End rules_ induction

As shown by the algorithm above, the process of induction of ANLQ-XLQ rules comes in five steps. The first step is to partition the ANLQ into elementary units to simplify its complexity and process tokens rather than an entire sentence. The following example shows the partitioning result of the ANLQ:

اوجد اسماء الطلاب الذين معدلاتهم = 14

اوجد اسماء الطلاب الذين معدلاتهم = 14

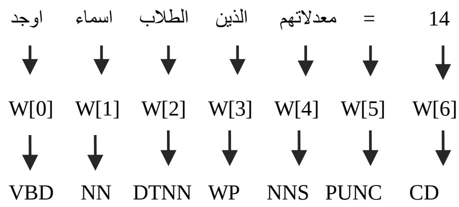
W[0] W[1] W[2] W[3] W[4] W[5] W[6]

During the second step, our system applies automatic tagging to detect the grammatical structure of ANLQ and the labeling of the XLQ to know its internal structure. This step is important before the introduction of the ANLQ -XLQ rules since it helps to recognize the functional structure of the ANLQ and XLQ.

For the labeling of XLQ, we have developed a specific method to label it according to the position of the words it contains. As an example of labels, we propose:

- R-S-O1 is the first object in the 'SELECT' fraction of the query.
- AT-1-R-S-O1 is the first attribute of the first object in the 'SELECT' part of the query.

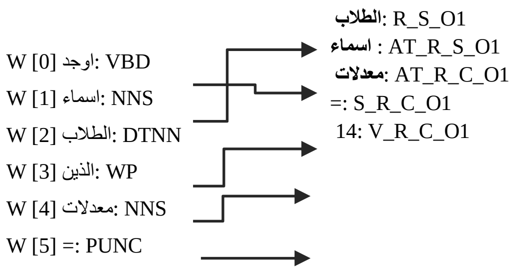
The following figure shows the grammatical structure of the ANLQ of the preceding example.



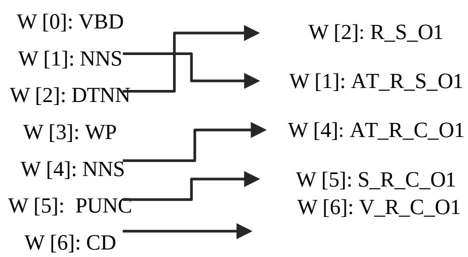
The next exemple displays the result of the labeling of the XLQ corresponding to the ANLQ:

<Request>
<Select>
<Object>
<Name> **الطلاب**</Name> → الطلاب: R_S_O1
<Attribute>
<Name> **اسماء** </Name> → اسماء: AT_R_S_O1
</Attribute>
</Object>
</Select>
<Cond>
<Object>
<Attribute>
<Name> **معدلات**</Name> → معدلات: AT_R_C_O1
</Attribute>
</Object>
<Symbol> = </Symbol> → =: S_R_C_O1
<Value> **14** </Value> → 14: V_R_C_O1
</Cond>
</Request>

The third step is Word-to-word alignment. In this step, the result of ANLQ's automatic tagging and XLQ Labeling are used by our system to identify the equivalent ANLQ and XLQ words. In the next example, we display the word alignment result of ANLQ and XLQ.



For the induction of the ANLQ-XLQ rule, our system applies unification between the sequences of the ANLQ and the XLQ sequence. ANLQ is represented by its grammatical structure and the XLQ is represented by the returned result of the Labeling function. The ANLQ-XLQ rule maps the grammatical structure of the ANLQ to the XLQ structure as shown in the following model.



The actual induction of the ANLQ-XAQ rule is carried out in the fourth step. The rule ANLQ-XLQ induced from the previous example is as follows:

W[0]:VBD+ W[1]:NN+ W[2]: DTNN+ W[3]:WP+ W[4]:NNS+ W[5]: PUNC+ W[6]:CD
→
W [2]:R_S_O1+W [1]:AT_1_R_S_O1+W [4]:AT_1_R_C_O1+W [5]: S_R_C_O1+W [6]:V_R_C_O1

Finally, the fifth step is to increase the number of ANLQ that our system can translate directly into XLQ, we use a method to create extensions of the induced ANLQ-XLQ rule. This method makes it possible to add new rules to the newly induced one without having to deal with other examples that produce them. Extension rules can be represented as an instance of induced rules. The extension of the induced ANLQ-XLQ rules is based on the out-of-context grammar below:

VBW → VB | VBG | VBD | VBN | ε
NNW → NN | NNS | NNP | NNPS
JJW→ JJ | JJR | JJS | ε
WPW → WP | WDT | WPS | WRB | ε
PRW → PRP | PRP$ |ε
RB → RB | RBR| RBS| ε
DTW → DT | ε

The box below shows examples of extension of ANLQ-XLQ rule induced by the preceding example:

- *Extension* 1:

W[0]:VBD+ W[1]:NN+ W[2]: DTNN+ W[3]:WP+ W[4]:NNS+ W[5]: PUNC + W[6]:CD
→
W[2]:R_S_O1+W[1]:AT_1_R_S_O1+W[4]:AT_1_R_C_O1 +W[5]:S_R_C_O1+W[6]:V_R_C_1

- *Extension* 2:

W[0]:VBG+ W[1]:NNS+ W[2:DTNN+ W[3]:WP+ W[4]:NNS+ W[5]: PUNC + W[6]:CD
→
W[2]:R_S_O1+W[1]:AT_1_R_S_O1+W[4]:AT_1_R_C_O1 +W[5]:S_R_C_O1+W[6]:V_R_C_1

- *Extension* 3:

W[0]:VBD+ W[1]:NN+ W[2]: NN+ W[3]:WP$+ W[4]:NNS+ W[5]:PUNC+ W[6]:CD
→
W[2]:R_S_O1+W[1]:AT_1_R_S_O1+W[4]:AT_1_R_C_O1 +W[5]:S_R_C_O1+W[6]:V_R_C_1

- *Extension* 4:

W[0]:VBD+ W[1]:NNS+ W[2]: DTNN+ W[3]:WP+ W[4]:NN+ W[5]:PUNC + W[6]:CD
→
*W[2]:R_S_O1+W[1]:AT_1_R_S_O1+W[4]:AT_1_R_C_O1 +W[5]:S_R_C_O1+W[6]:V*_R_C1

5. Translation based on ANLQ-XLQ Rules

The translation process based on ANLQ-XLQ rules consists of four main steps. First, we divide the ANLQ into tokens. Then, we use the automatic tagging to know the grammatical function of each of these tokens. Then our system tries to find the rule ANLQ-XLQ which corresponds to the grammatical structure of ANLQ from the base of the rules ANLQ-XLQ previously induced. To do this, he looks for the rule ANLQ-XLQ whose left part has the same grammatical structure as that of ANLQ submitted. Finally, the system applies the appropriate ANLQ-XLQ rule to produce the XLQ. The sequence of these different steps is described in Figure 3.

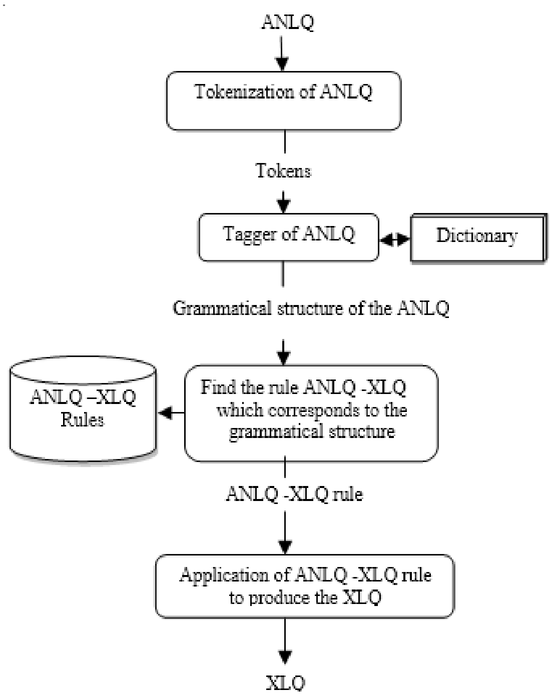
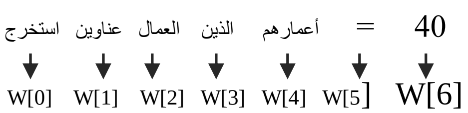


Figure 3. Translation process based on ANLQ -XLQ rules.

The example below shows the result of applying the ANLQ-XLQ rule induced in the previous section on the ANLQ “استخرج عناوين العمال الذين اعمارهم = 40” in order to transform it into an XLQ.

- *Step* 1: Tokenization of ANLQ



- *Step* 2: Tagger of ANLQ

W[0]: استخرج: VBD + W[1]: عناوين: NN +W[2]: العمال: DTNN +W[3]: الذين: WP +W[4]: أعمار: NN+W[5]: =: PUNC + W[6]: 40: CD

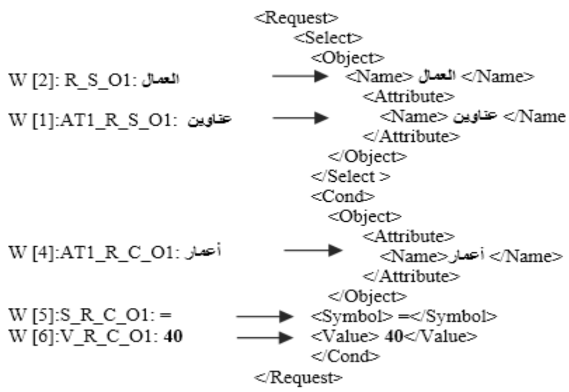
- *Step* 3: search for the rule ANLQ -XLQ whose left part has the following grammatical structure:

W [0]: VBD + W[1]: NN+W[2]: DTNN +W[3]:WP+W[4]: NN+W[5]: PUNC+ W[6]: CD

The proper ANLQ-XLQ rule is:

W[0]: VBD + W[1]: NN+W[2]: DTNN +W[3]:WP+W[4]: NN+W[5]: PUNC+ W[6]: CD
→
W[2]:R_S_O1+w[1]:AT1_R_S_O1+W[4]:AT1_R_C_O1+w[5]:S_R_C_O1+w[6]:V_R_C_O1

- *Step* 3: The application of the ANLQ-XLQ rule to produce the XLQ



6. Results Presentation

Interface in Figure 4 displays the rules induced from a corpus of example of ANLQ paired with their XLQ.

Figure 4. Rules induced from a corpus of example of ANLQ paired with their XLQ.

Interface in Figure 5 represents the translation based on ANLQ-XLQ rules.

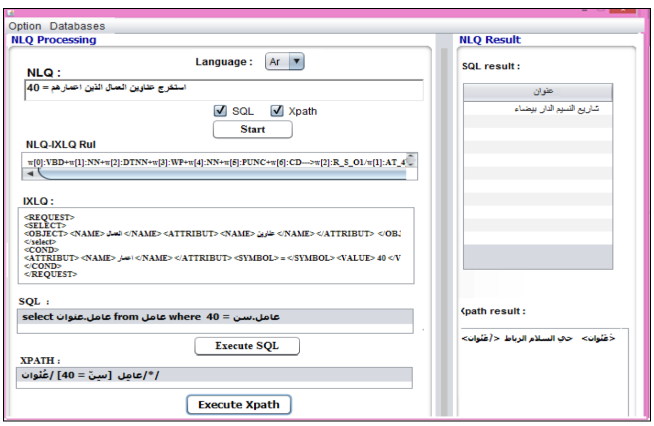


Figure 5. Translation Based on ANLQ-XLQ rules.

Table 1. Examples of rules induced by our system.

ANLQ	Induced Rules	Nb of extensions
استخرج لي جميع الموظفين	w[0]:VBD+w[1]:NNP+w[2]:NOUN+ w[3]:DTNNS→w[3]:R_S_OA1	7 règles
اظهر اسماء الطلاب	w[0]:VBD+w[1]:NN+w[2]:DTNN → w[2]:R_S_O1/w[1]:AT_1_R_S_O1	31 règles
ابحث عن أعمار و عناوين المدرسين	w[0]:VBP+w[1]:IN+w[2]:NN+w[3]: CC+w[4]:NN+w[5]:DTNNS→w[5]: R_S_O1 /w[2]:AT_1_R_S_O1/w[4]:AT_2_R_ S_O1	127 règles
جميع أعمار و عناوين الموظفين و العمال	w[0]:NOUN+w[1]:NN+w[2]:CC+w[3]:NN+ w[4]:DTNNS+w[5]:CC+w[6]:DTNN → w[4]:R_S_O1/w[1]:AT_1_R_S_O1/w [3]:AT_2_R_S_O1/w[6]:R_S_O2 /w[1]:AT_1_R_S_O2/w[3]:AT_2_R_ S_O2	255 règles
ما هي رواتب الموظفين الذين أعمارهم تزيد عن 30	w[0]:WP+w[1]:PRP+w[2]:NN+w[3]: DTNNS+w[4]:WP+w[5]:VBD+w[6]: PUNC+w[7]:CD→w[3]:R_S_O1/w[2]:AT_1_R_S_O1/w[5]:AT_1_R_C_O 1 /w[6]:S_R_C_O1/w[7]:V_1_R_C_O1	287 règles
أوجد أعلى سن للموظفين الذين رواتبهم > 5000	w[0]:VBD+w[1]:VBD+w[2]:NN+w[3]:NNS+w[4]:WP w[5]:NN+w[6]:PUNC+w[7]:CD→ w[3]:R_S_O1/w[1]:AG_AT_1_R_S_ O1/w[2]:A_1_R_S_O1/w[5]:AT_1_R _C_O1/w[6]:S_R_C_O1/w[7]:V_1_R _C_O1	767 rules
اوجد اسم المدرس الذي يدرس مادة ' الجبر'	w[0]:VBD+w[1]:NN+w[2]:DTNN+w [3]:WP+w[4]:VBP+w[5]:NN+w[6]:P UNC+w[7]:DTNN+w[8]:PUNC → w[2]:R_S_O1/w[1]:AT_1_R_S_O1/w [5] :AT_1_R_C_O1 /w[7]:V_1_R_C_O1	971 règles
جميع اسماء و أعمار الموظفين و العمال الذين أعمارهم > 40	w[0]:NN+w[1]:NN+w[2]:CC+w[3]:N N+w[4]:DTNNS+w[5]:CC+w[6]:DT NN+w[7]:WP+w[8]:VBD+w[9]:PUN C+w[10]:CD → w[4]:R_S_O1/w[1]:AT_1_R_S_O1/w [3]:AT_2_R_S_O1/w[8]:AT_2_R_S_ O1/w[6]:R_S_O2/w[1]:AT_1_R_S_O 2/w[3]:AT_2_R_S_O2/w[8]:AT_2_R _S_O2/w[3]:AT_1_R_C_O1/w[8]:A T_1_R_C_O1/w[9]:S_R_C_O1/w[10]:V_1_R_C_O1	1537 règles
معلومات الموظفين و العمال الذين اسمهم ' محمد ' أو' أحمد'	w[0]:NNS+w[1]:DTNNS+w[2]:CC+ w[3]:DTNN+w[4]:WP+w[5]:NN+w[6]:PUNC+w[7]:NNP+w[8]:PUNC+w [9]:NNP+w[10]:PUNC+ w[11]:NN+w[12]:PUNC → w[1]:R_S_O1/w[0]:AT_1_R_S_O1/w [3]:R_S_O2/w[0]:AT_1_R_S_O2/w[5]:AT_1_R_C_O1/w[7]:V_1_R_C_O 1/w[11]:V_1_R_C_O1	3071 règles
أحسب عدد الموظفين و العمال ذوي اعلى راتب	w[0]:VBD+w[1]:NN+w[2]:DTNNS+ w[3]:CC+w[4]:DTNN+w[5]:NN+w[6]:JJ+w[7]:NN → w[2]:R_S_O1/w[1]:AT_1_R_S_O1/w [4]:R_S_O2/w[1]: AT_1_R_S_O2 /w[6]:AG_AT_1_R_C_O1/w[7]:AT_ 1_R_C_O1	6143 règles
اظهر رواتب الموظفين و العمال الذين اسمهم ' محمد ' و راتبهم > 2000	w[0]:NN+w[1]:DTNNS+w[2]:CC+w [3]:DTNN+w[4]:WP+w[5]:NN+w[6] :PUNC+w[7]:NNP+w[8]:PUNC+w[9]:CC+w[10]:NN+w[11]:NN+w[12]:C D→w[1]: R_S_O1/w[0]:AT_1_R_S_O1/w[3]:R _S_O2/w[0]:AT_1_R_S_O2/w[5]:AT _1_R_C_O1/w[7]:V_1_R_C_O1/w[1 0]:AT_1_R_C_O1/w[11]:S_R_C_O1/ w[12]:V_1_R_C_O1	12287 règles

Table 1 show examples of rules induced by our system.

We tested the proposed method using 1000 ANLQs. During this experiment we tried to make a comparison between the translation based on the ANLQ–XLQ rules and that based on linguistic operations. The performances obtained were satisfactory.

Indeed, in the first test, we present the number of ANLQs for which our system generates a response in the case of translation based on the ANLQ–XLQ rules and that based on linguistic operations. The results obtained by this test are presented in Figure 6.

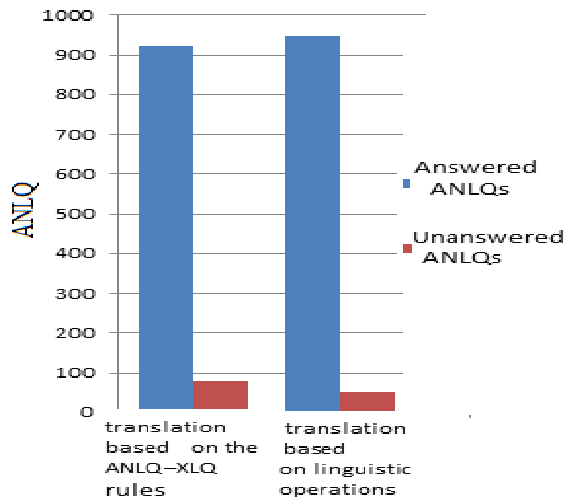


Figure 6. Answered\Unanswered ANLQs.

On examining Figure 6, we note that the performance of the translation based on linguistics operations is almost similar to the translation based on ANLQ -XLQ rules. 92.1% of the ANLQs generated responses during translation based on language operations and 94.7% of these ANLQs were answered in the case of the use of the ANLQ-XLQ rules.

Secondly, we are interested in testing the performance of the proposed method according to the correctly generated queries when using language operations and the ANLQ-XLQ rules. The results obtained are showed in Figure 7 as follows:

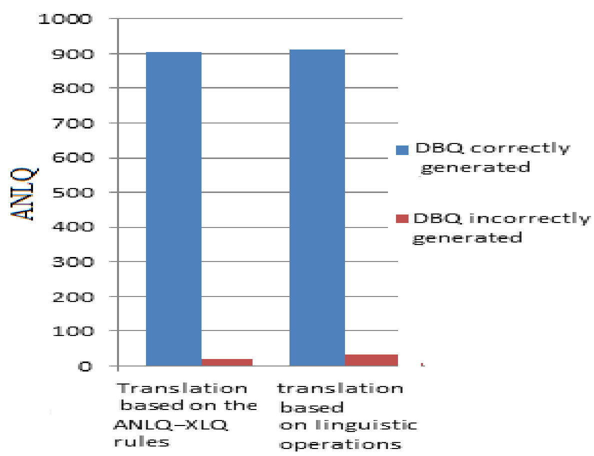


Figure 7. DBQ correctly/incorrectly generated.

We have shown in Figure 7 that 96.4% of the ANLQs generated DBQ queries syntactically correctly during translation based on linguistic operations and 97.83% when translating based on ANLQ-XLQ rules. In Figure 8, the DBQ requests match / do not match the ANLQs are presented.

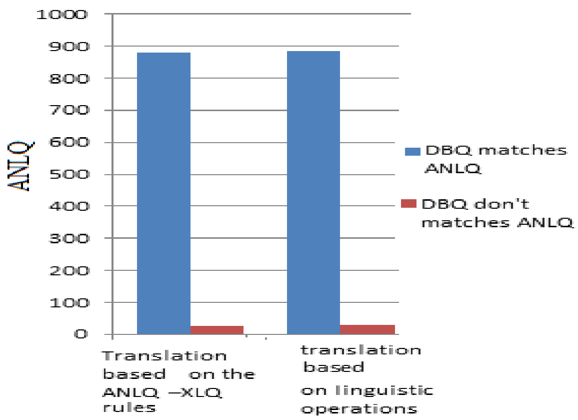


Figure 8. DBQ matches\ don't matches ANLQ.

From Figure 8, we show that 97.23% of DBQs match ANLQs are produced by A Translation based on the ANLQ–XLQ rules and 96.71% by translation based on linguistic operations

Figure 9 illustrates the response time improvement result when using the ANLQ -XLQ rules. These ANLQs are run on a laptop with a Core i3 CPU and 4GB memory.

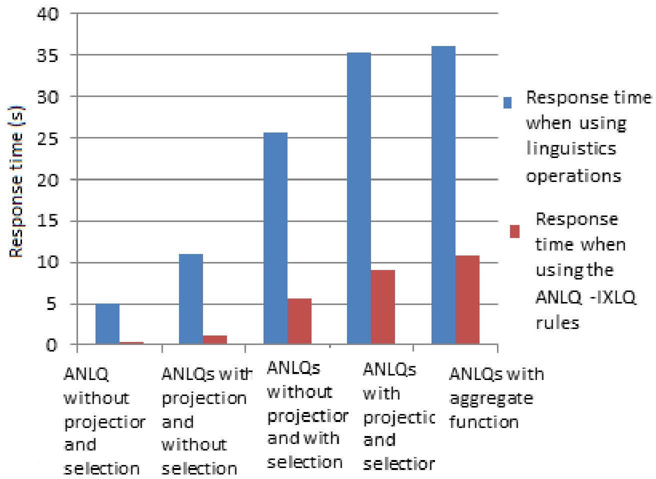


Figure 9. Response time improvement result when using the ANLQ -XLQ rules.

As shown in Figure 9, the response time has been decreased, and can reach in some cases 91.07%, when using the ANLQ -XLQ rules compared to that of linguistics operations.

To be able to test the influence of computer resources on the performance of the system we have carried out experiments on the improvement of response time between two machines: a laptop with a CPU Core i3 (2.40 GHZ) a memory of 4GB and another desktop with a Core i5 CPU (3.20 GHZ) and a 4GB memory. The results obtained which are illustrated in Figure 10 show that there are significant improvements.

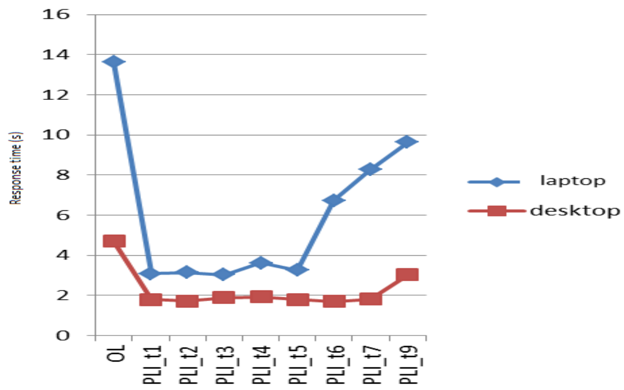


Figure 10. Influence of computer resources on the performance of system.

To prove the consistency of using ANLQ-XLQ rules and to avoid memory overflow and large CPU occupancy, we examined the number of rules generated when processing 100 ANLQs. Figure 11 shows the results of this review.

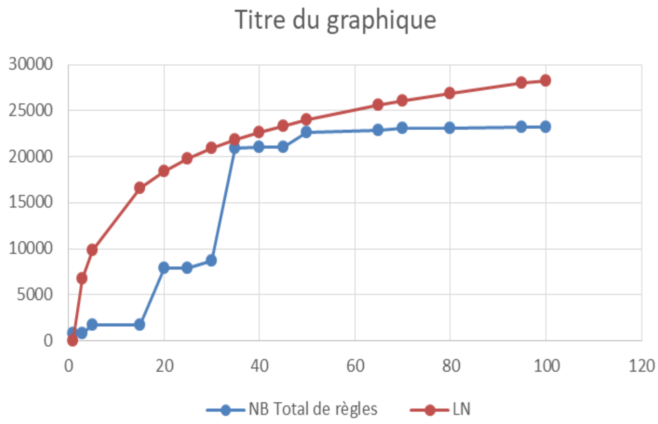


Figure 11. Logarithmic growth during the generation of ANLQ-XLQ rules.

The graph in Figure 11 shows that the system offers logarithmic growth during the generation of ANLQ-XLQ rules. This degrades newly created ANLQ-XLQ rules after a limited set of trainings, which ensures its consistency.

7. Conclusions

In this paper, we have proposed a rules induction approach to induce transfer rules that map the ANLQ to XLQ without using language operations. To learn these rules, our system exploits the examples of the ANLQs associated with their XLQs. The results of the experiment show that such an approach is promising since it reduces the time well and simplifies the procedure necessary to translate the ANLQs having a grammatical structure similar to that of the ANLQs that have already been processed. This is done while maintaining the consistency of the system.

The possible future direction is to continue extending the capabilities of proposed interface to learn queries in other languages and increase the ability of our system to interface other database models

References

[1] Akerkar R. and Joshi M., "Natural Language Interface Using Shallow Parsing," *International Journal of Computer Science Applications*, vol. 5, no. 3, pp. 7090, 2008.

[2] Al Tamimi A., Jaradat M., Al-Jarrah N., and Ghanem S., "AARI: Automatic Arabic Readability Index," *The International Arab Journal of Information Technology*, vol. 11, no. 4, pp. 370-378, 2014.

[3] Androutsopoulos I., Ritchie G., Thanisch P., "Natural Language Interfaces to Databases an Introduction," *Natural Language Engineering*, 1995.

[4] Athanasopoulou G., Klasinas I., Georgiladakis S., Iosif E., and Potamianos A., "Using Lexical, Syntactic And Semantic Features for Non-Terminal Grammar Rule Induction in Spoken Dialogue Systems," *in Proceedings IEEE Spoken Language Technology Workshop*, South Lake Tahoe, pp. 596-601, 2014.

[5] Bais H., Machkour M., Lahcen K., "A Model of a Generic Natural Language Interface for Querying Database," *International Journal of Intelligent Systems and Applications*, vol. 8, no. 2, pp. 35-44, 2016.

[6] Bais H., Machkour M., Lahcen K., "Querying Database Using A Universal Natural Language Interface Based On Machine Learning," *in Proceedings International Conference on Information Technology for Organizations Development*, Fez, pp. 16, 2016.

[7] Cordeiro J., Dias G., and Brazdil P., "Rule Induction for Sentence Reduction," *in Proceedings Portuguese Conference on Artificial Intelligence*, Azores, pp. 528-539, 2013.

[8] Landhäußer M., Weigelt S., and Tichy W., "NLCI: a Natural Language Command Interpreter," *Automated Software Engineering*, vol. 24, no 4, pp. 839-861, 2017.

[9] Llopis M. Ferr_andez A., "How to Make A Natural Language Interface to Query Databases Accessible to Everyone: An Example," *Computer Standards Interfaces*, vol. 35, no. 5, pp. 470-481, 2013.

[10] Nagesh A., Ramakrishnan G., Chiticariu L., Krishnamurthy R., Dharkar A., and Bhattacharyya P., "Towards Efficient Named-Entity Rule Induction for Customizability," *in Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics*, Jeju Island, pp. 128-138, 2012.

[11] Popescu A., Armanasu A., Etzioni O., Ko D., and Yates A., "Modern Natural Language Interfaces to Databases: Composing Statistical Parsing with Semantic Tractability," *in Proceedings of the 20th international conference on Computational Linguistics. Association for Computational Linguistics*, Geneva Switzerland, pp. 141, 2004.

[12] Rao G., Chanchal A., Chaudry S., Nikita K., Patel S., "Natural language Query Processing Using Semantic Grammar," *International Journal on Computer Science and Engineering*, vol. 2, no. 2, pp. 219-223, 2010.

[13] Shaalan K. and Hossny A., "Automatic Rule Induction in Arabic to English Machine Translation Framework," *Challenges for Arabic Machine Translation*, vol. 9, no. 2012, pp. 135, 2012.

[14] Shah A., Pareek J., Patel H., and Panchal N., "NLKBIDB-Natural Language and Keyword Based Interface to Database," *in Proceedings International Conference on Advances in Computing, Communications and Informatics*, Mysore, pp. 1569-1576, 2013.

[15] Tang L., "Using a Machine Learning Approach for Building Natural Language Interfaces for Databases: Application of Advanced Techniques in Inductive Logic Programming," *Journal of Computer Science, Informatics Electrical Enginnering*, vol. 2, no. 1, 2008.

[16] Zelle J. Mooney R., "Learning to Parse Database Queries Using Induc Programming," *Proceedings of the 30th National Conference on Artificial Intelligence*, Portland Oregon, pp. 1050-1055, 1996.

Hanane Bais received her PhD degree in computer science from the the Ibn Zoher University, Agadir, Morocco in 2013 from Departments of Computer Science, Faculty of Science, University Ibn Zohr, Agadir, Morocco. She is currently a Ph.D. candidate of the Ibn Zoher University, Agadir, Morocco. Her research interests include DataBase system, natural language processing and artificial intelligence.

Mustapha Machkour is a full professor of higher education, department of computer Sciences, Head of the Intelligent Computing Models and Knowledge Engineering (M3IC) team, Ibn Zohr University, Agadir, Morocco.
Member of Laboratory of Computer Systems and Vision, Faculty of Science, Ibn Zohr University, Agadir Morocco. Current research interests include Natural Language Processing, DataBase System, logic and artificial intelligence, Image processing.