Generating Answering Patterns from Factoid Arabic Questions

Essia Bessaies, Slim Mesfar, Henda Ben Ghezala*
Riadi Laboratory, *ENSI,
University of Manouba Tunisia
{essia.bessaies, slim.mesfar}@riadi.rnu.tn;
*henda.benghezala@ensi.rnu.tn

Abstract

This works deals with Arabic factoid Question Answering systems (QA). Commonly, the task of QA is divided into three phases: question analysis, answer pattern generation, and answer extraction. Each phase plays a crucial role in overall performance. In this paper, we focus on the two first phases: Question Analysis and Answer Pattern Generation. We used the NooJ platform which represents a valuable linguistic development environment. The first evaluations show that the actual results are encouraging and could be deployed for more types of questions other than factoid ones.

1 Introduction

In recent years, the medical domain has a high volume of electronic documents. Managing this large quantity of data makes the search of specific information complex and time consuming. This complexity is especially evident when we seek a short and precise answer to a human natural language question rather than a full list of documents and web pages. In this case, the user requirement could be a Question Answering (QA) system which represents a specialized area in the field of information retrieval.

The goal of a QA system is to provide inexperienced users with a flexible access to information allowing them to write a query in natural language and obtain not the documents which contain the answer, but its precise answer passage from input texts. There has been a lot of research in English as well as some European language QA systems. However, Arabic QA systems (Brini et al., 2009) could not match the pace due to some inherent difficulties with the language itself as well as due to lack of tools available to assist researchers. Therefore, the current project attempts to design and develop the modules of an Arabic QA system.

In this paper, we present a linguistic approach for analyzing medical questions and generating answer patterns from factoid Arabic questions.

In the first section, we give a short insight about Arabic NLP specificities followed by an overview of state-of-the-art developments. In section 3, we describe the generic architecture of the proposed QA system. Section 4 introduces our approach to the annotation of medical factoid questions and the generation of answering patterns.

In this section, we describe how we analyze a given question by means of the application of a cascade of morpho-syntactic grammars. The linguistic patterns depicted in these grammars allow us to annotate the question in order to extract its type (time, quantity …), its topic, as well as its focus. After examining the generation of response patterns from these extracted key words (Type, Topic and Focus), the results of our experiments are described in section 5.

2 Current Research

As explained in the introduction, QA systems present a good solution for textual information retrieval, knowledge sharing, and discovery. Current research deals with two challenging topics: the Arabic natural language processing in the medical domain and the second concerns QA systems.

2.1 The Arabic language

The Arabic language is a member of the Semitic language family and it is the most widely spoken one with almost 300 million first language speakers. The Arabic language has its own script (written from right to left) using a 28 letters alphabet (25 consonants and 3 long vowels) with allographic variants and diacritics which are used as short vowels

Proceedings of the Linguistic Resources for Automatic Natural Language Generation Workshop, pages 17–24, Santiago de Compostela, Spain, September 4, 2017. © 2017 Association for Computational Linguistics

except one diacritic which is used as a double consonant marker. The Arabic script does not support capitalization that could help researchers identify named entities, for example. Numbers, however, are written from left to right which presents a real challenge for Arabic text editors to handle words written from right to left and numbers from left to right.

2.2 Arabic QA systems

QA systems for Arabic are very few. Mainly, it is due to the lack of accessibility to linguistic resources, such as freely available lexical resources, corpora and basic NLP tools (tokenizers, morphological analyzers, etc.).

To our knowledge, there are only five research works on Arabic QA systems.

- QARAB (Hammo et al., 2002) is an Arabic QA system that that takes factoid Arabic questions and attempts to provide short answers. QARAB uses both information retrieval and natural language processing techniques.

- ArabiQA (Benajiba et al., 2007), which is fully oriented to the modern Arabic language, also answers factoid questions using Named Entity Recognition. However, this system is not yet completed.

- DefArabicQA (Trigui et al., 2010) provides short answers to Arabic natural language questions. This system provides effective and exact answers to definition questions expressed in Arabic from Web resources. DefArabicQA identifies candidate definitions by using a set of lexical patterns, filters these candidate definitions by using heuristic rules and ranks them by using a statistical approach. It only processes definition questions and does not include other types of question (When, How and Why).

- AQuASys (Bekhti and Alharbi, 2013) is composed of three modules: A question analysis module, a sentence filtering module and an answer extraction module. Special consideration has been given to improving the accuracy of the question analysis and the answer extraction scoring phases. These phases are crucial in terms of finding the correct answer. The recall rate is 97.5% and the precision rate is 66.25%.

- Yes/No Arabic Question Answering System (Kurdi, et al, 2014) is a formal model for a semantic based yes/no Arabic question answering system based on paragraph retrieval. The results are based on 20 documents. It shows a positive result of about 85% when we use entire documents. Besides, it gives 88% when we use only paragraphs. The system focuses only on yes/no questions and the corpus size is relatively small (20 documents).

After this investigation into QA systems, we aim to develop a QA system based on a linguistic approach. It uses NooJ's linguistic engine in order to formalize the automatic recognition rules for question analysis. The named entity recognizer (NER) is embedded in our QA system in order to annotate the factoid questions and associate them with the extracted named entities. For this purpose, we have adapted a rules based approach to recognize Arabic named entities. Furthermore, we aim to generate the potential answer patterns using the paraphrasing and transformation module in the NooJ platform (Silberztein, 2015).

3 Proposed Architecture for our QA System

From a general viewpoint, the design of a QA system (Figure 1) must take into account three phases:

Question analysis: This module performs a morphological analysis to determine the question class. A question class helps the system to classify the question type to provide a suitable answer. This module may also identify additional semantic features of the question like the topic and the focus.

Answer pattern generation: After analyzing the question and extracting the key words (e.g., focus and topic) from a given factoid question, the system generates response patterns from these extracted key words. Our approach automatically generates patterns using NooJ's linguistic engine.

Answer extraction: This module selects the most accurate answers among the phrases in a given corpus. The selection is based on the question analysis. The suggested answers are then given to the user as a response to his initial natural language query.

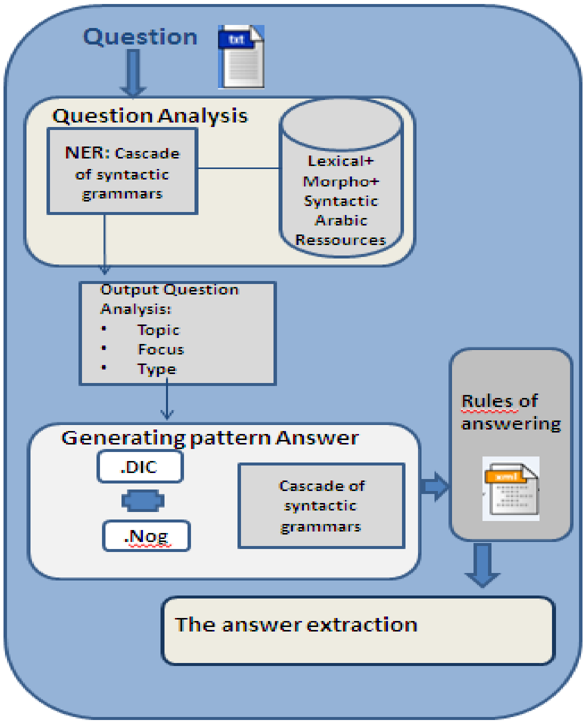

Question
txt
Question Analysis
NER: Cascade of syntactic grammars
Lexical+ Morpho+ Syntactic Arabic Ressources
Output Question Analysis:
• Topic
• Focus
• Type
Rules of answering
xml
Generating pattern Answer
.DIC
.Nog
Cascade of syntactic grammars
The answer extraction

Figure 1: Architecture for our QA System

In this paper, we explore only the first two phases.

4 Preconized processing approach

4.1 Question Analysis

Our approach uses the NooJ development platform. It allows us to build various required resources (dictionaries, morpho-syntactic grammars, etc.) and perform a linguistic analysis on a given corpus.

In addition to the linguistic resources in NooJ's Arabic module (Mesfar, 2010), we added specific properties in lexical entries for the needs of the current project. For instance, some additional information must be added to entries:

1. Nominal predicate information
   - **Npred =** Nominal predicate information
2. Synonyms
   - **Syn**=Synonyms of focus
3. Support verbs (that will be used to generate paraphrases)
   - **Sup**=Support verbs

Example

- اِكْتَشَفَ, V+FLX+NPred=اِكْتَشَاف +Syn1= أَوْجَدَ +Sup1=وَقَعَ +Sup2=تَمَّ +sup3=قَامَ بِـ

These enhanced lexical resources are used next within a cascade of morpho-syntactic grammars.

Named Entity Recognition:

We think that an integration of a Named Entity Recognition (NER) module will definitely boost system performance. It is also very important to point out that an NER is required as a tool for almost all the QA system components. Those NER systems allow extracting proper nouns as well as temporal and numeric expressions from raw text (Mesfar, 2007). In our case, we used our own NER system especially formulated for the Arabic medical domain. We have considered six proper names categories: organization, location, person, viruses, diseases, and treatment (Table 1).

Categories	**Definitions**	**Examples**
Organization	Names of corporattions, gov. entities or ONGs	بنك ألدم = blood bank
Location	Politically or geographically defined locations	مستشفى الأطفال = Children's Hospital
Person	Names of persons or families	طبيب النساء علي طارق = Gynecologist Tariq Ali
Viruses	Names of medical viruses	فيروس الروتا = Rotavirus
Diseases	Names of diseases, illness, sickness	مرض السرطان = Cancer
Treatment	Names of medical viruses	علاج طبيعى = Physiotherapist

Table 1: Named Entity Recognition

Automatic annotation of question using NooJ's syntactic grammars

Our approach focuses on the problem of finding document snippets that answer a particular category of fact-seeking questions or factoid questions, for example simple interrogative questions with a named entity (Timex, Numex or Enamex). The choice of factoid questions versus other types of questions is motivated by the following factors:

- A considerable percentage of the questions actually submitted to a search engine are factoid questions. Current search engines are only able to return links to full-length documents rather than brief document fragments that answer the user's question.
- The frequent occurrence of factoid questions in daily usage is confirmed by the composition of the question test sets in the QA track at TREC[1]. The percentage of questions that are factoid questions grew in TREC.
- Most recent approaches to open-domain QA use NER as a foundation for detecting candidate answers.

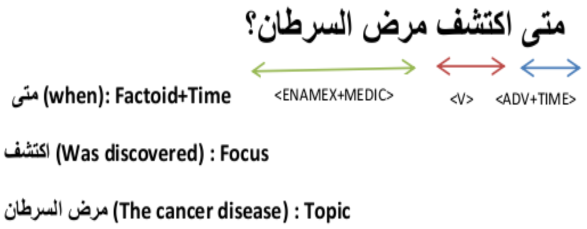
As far as current research is concerned, our QA module accepts, as input, only Arabic factoid questions. Then, in order to look for the best answer, it gives the maximum amount of information (syntactic, semantic, distributional, etc.) from the given question, such as the expected answer type, and the focus and topic of the question. This information will play an important role in the generation of potential answer patterns.

- **Type:** the type corresponds to the type of question (time, person, organization, etc.)
- **Topic:** the topic corresponds to the subject matter of the question.
- **Focus**: the focus corresponds to the specific property of the topic that the user is looking for.

The following example shows the detailed annotation of the identified parts of a question.

Example

- When was cancer discovered?



4.2 Generating answer patterns

Arabic sentences are usually complex and very long. This sets up obstacles for the extraction of a short and precise answer for a given question. Thus, we chose to generate answer patterns associated with the output of our question analysis. Our preliminary observations on sentence structure showed that a huge number of response structures could be extracted from Arabic texts.

These structures depend on author origins (native language, geographical zone of Arabic studies, etc). Hence, generating answer patterns could be an interesting alternative to identify the different structures and answer patterns (see Table 2).

The example provided in Table 2 consolidates the main objective of this project which consists in developing a context-sensitive and linguistically enhanced paraphrase generator. First, this generator uses the annotated sequences within the question analysis phase (recognized syntactic-semantic sequences, named entities, multi-words and other phrasal units). Then, it transforms them into semantically equivalent phrases, expressions, or sentences. This output produces potential answer patterns based on the original question's topic and focus.

For instance, these generated patterns will use the predicate noun, synonyms as well as the related support verbs shown in the annotation of focus or topic.

[1] Text Retrieval Conference is an ongoing series of workshops focusing on different types of information retrieval (IR), research areas, or tracks.

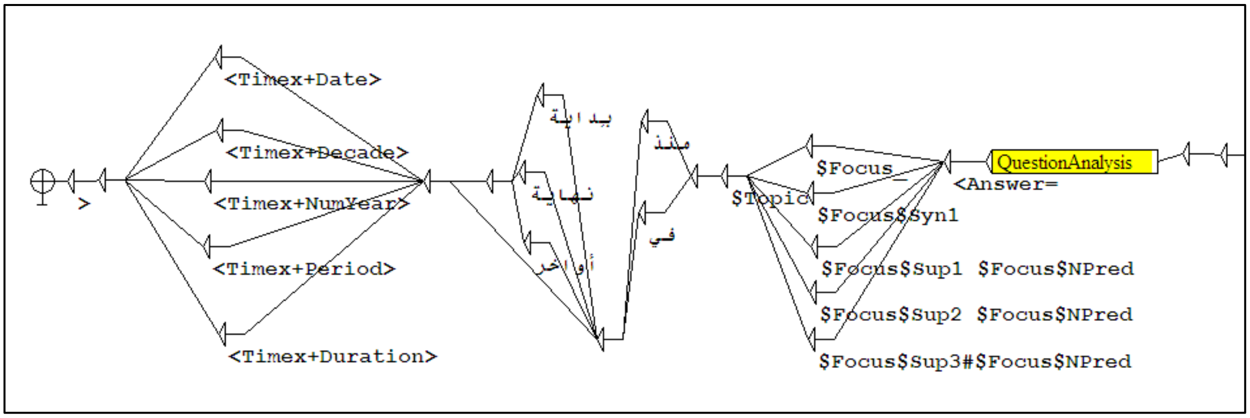


Figure 2: Sub-graph of answer pattern generation (syntactic grammar)

Question:	
متى وقع إكتشاف مرض السيدا ؟	When did the discovery of AIDS occur?

Some answer structures:	
وقع إكتشاف مرض السيدا في سنة 1981	The discovery of Aids occurred in 1981.
إكتشف مرض السيدا في سنة 1981	The discovery of Aids was in year 1981.
تم إكتشاف مرض السيدَا في سنة 1981	The discovery of aids was made in 1981
إكتشف مرض الإيدز منذ بداية الثمانينات	AIDS has been discovered since the early 80's
منذ سنة 1981 إكتشف مرض الإيدز	Since 1981, he has discovered AIDS
في بداية الثمانينات وقع إكتشاف مرض السيدا	In the early 80's the discovery of SIDA occurred
وقع إكتشاف مرض السيدا في الثمانينات	The discovery of SIDA occurred in the 1980s
تم إكتشاف مرض السيدا في الثمانينات	SIDA was discovered in the 80's
في سنة 1981إكتشف مرض السيدا	In 1981 he discovered Aids.
مرض السيدا يعتبر من الأمراض التي إكتشافها في بداية الثمانينات	Aids is one of the diseases that were discovered in the early 1980s.

Table 2: Example of generated example patterns

If we consider the previous example (Table 2), we can take advantage of the focus's annotation information (إِكْتَشَفَ – to discover) to generate:

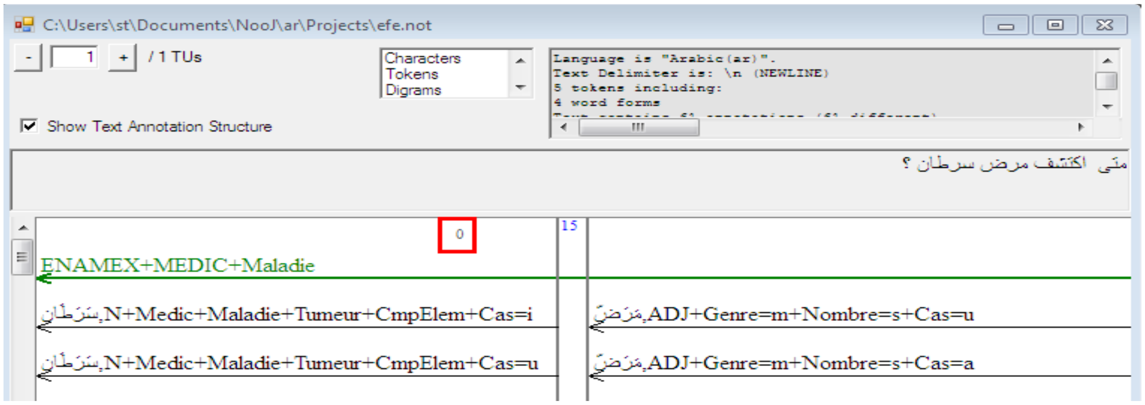
- إِكْتَشَافْ **(discovery)** : the nominal predicate
- أَوْجَدَ **(to find)** : a synonym
- وَقَعَ **(to occur) :** a support verb that can be used in conjunction with the predicate noun (discovery)

Based on the information associated with the different parts of the question, we generate answer patterns with respect to the potential phrase structures (nominal, verbal, prepositional, adverbial, active or passive phrases). In fact, this task takes into consideration the type of answer expected by the user, and this means that the answer extraction module will perform differently for each type of question.

The sub-graph illustrated in Figure 2 with NooJ's graphical editor, shows that the sub-graph called "Question Analysis" that stores the question parts in two variables: $Topic (contains the topic of our question) and $Focus (contains the special focus of the current question). Then, we proceed to the pattern generation where we use the related information (syntactic, semantic, distributional as well as synonymy and/or lexicon-grammar properties). In order to display the needed information, we build a combination of output patterns using the following variables:

- $Focus$Syn1
- $Focus$NPred
- $Focus$Sup

Finally, we add the potential combination of response parts (in the given example, we added Timex expressions)


C:\Users\st\Documents\NooJ\ar\Projects\efe.not
- 1 + / 1 TUs
Characters
Tokens
Digrams
☑ Show Text Annotation Structure

```
Language is "Arabic(ar)".
Text Delimiter is: \n (NEWLINE)
5 tokens including:
4 word forms
```

متى اكتشف مرض سرطان ؟

ENAMEX+MEDIC+Maladie

سَرَطان,N+Medic+Maladie+Tumeur+CmpElem+Cas=i

سَرَطان,N+Medic+Maladie+Tumeur+CmpElem+Cas=u

مَرَض,ADJ+Genre=m+Nombre=s+Cas=u

مَرَض,ADJ+Genre=m+Nombre=s+Cas=a

Figure 3: Text Annotation Structure

5 Experiments and Results

5.1 Question analysis

Named Entity Recognition (NER)

The ENAMEX+MEDIC grammar is launched automatically during the linguistic analysis, in order to annotate the sequences and expressions recognized by the transducers corresponding to the grammar launched (Figure 3).

To evaluate our NER local grammars, we analyzed our corpus to extract manually all named entities. Then, we compare the results of our system with those obtained by manual extraction. The application of our cascade of local grammars gives the results as shown in Table 3.

Precision	Recall	F-Measure
0.90	0.82	0.88

Table 3: NER grammar experiments

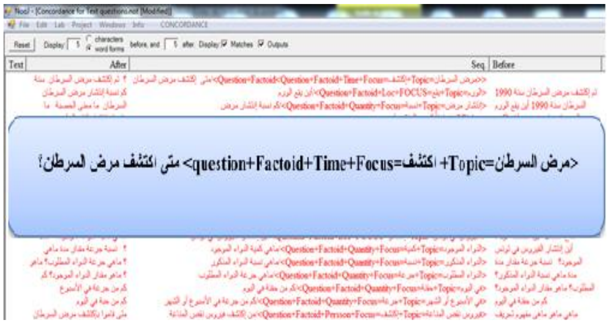
According to these results (Table 3), we obtain acceptable scores for named entities recognition. Our evaluation shows an F-measure of 0.88. This result is encouraging given the rate achieved by the systems participating in MUC[2].

Discussion

- Despite the problems described above, the techniques used seem to be adequate and display very encouraging recognition rates. Indeed, a minority of the rules may be sufficient to cover a large part of the patterns and ensure coverage. However, many other rules must be added to improve recall.

Automatic annotation of factoid question in standard Arabic

To evaluate our automatic annotation of questions using local grammars, we compare the results of our system with those obtained by manual extraction (Figure 4).


Reset
Display 5 ○ characters ● word forms before, and 5 after. Display ☑ Matches ☑ Outputs
Text
After
Seq
Before

<مرض السرطان=Topic+ اكتشف=question+Factoid+Time+Focus> متى اكتشف مرض المرطان؟

Figure 4: Annotation results of question analysis syntactic grammar (NooJ Grammar).

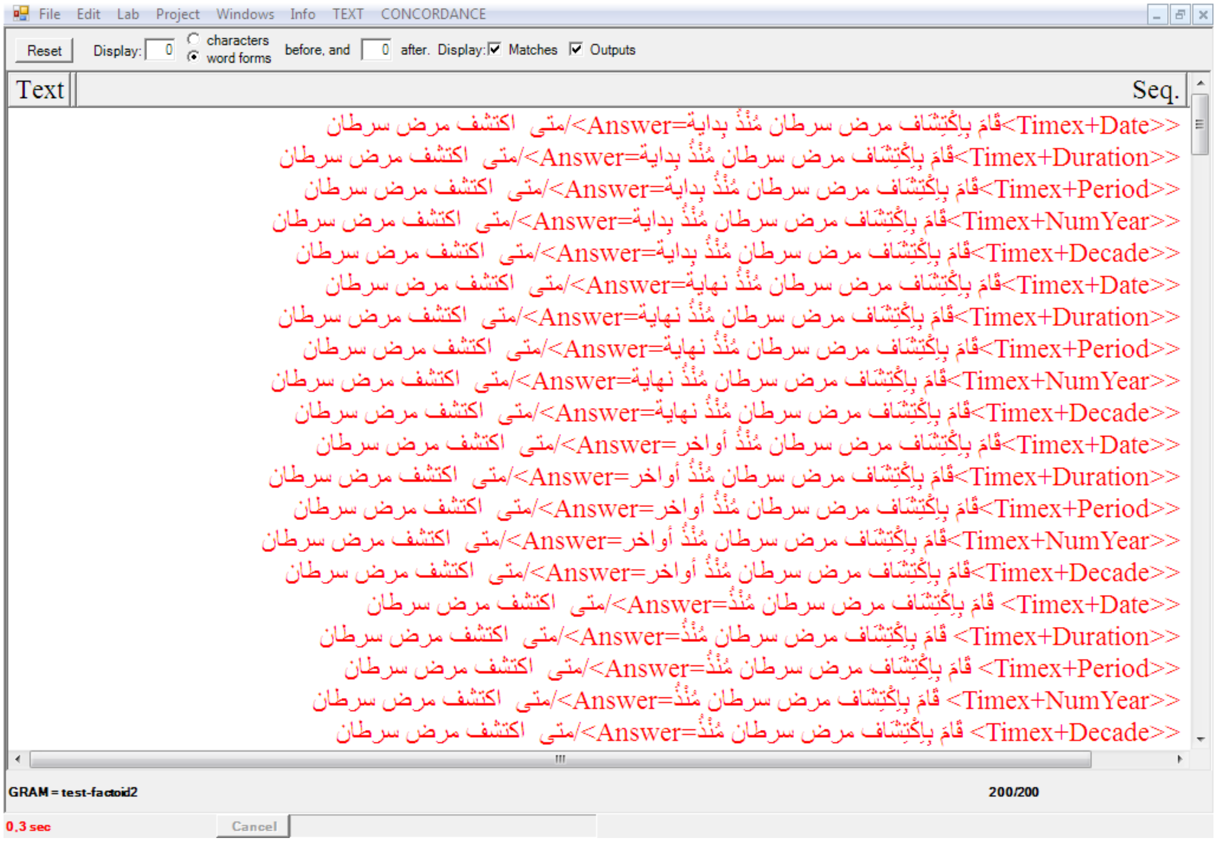
The application of our local grammar gives the result as shown in Table 4.

Precision	Recall	F-Measure
0.75	0.72	0.73

Table 4: Annotation of factoid question experiments

According to these results (Table 4), we obtain an acceptable annotation rate for the cascade of morpho-syntactic grammars. Our evaluation shows an F-measure of 0.73. We note that the rate of silence in the corpus is low, which is represented by the recall value 0.72.

[2] The Message Understanding Conferences.



```
<<Timex+Date>قَامَ بإكْتِشَاف مرض سرطان مُنْذُ بداية=Answer>/متى  اكتشف مرض سرطان
<<Timex+Duration>قَامَ بإكْتِشَاف مرض سرطان مُنْذُ بداية=Answer>/متى  اكتشف مرض سرطان
<<Timex+Period>قَامَ بإكْتِشَاف مرض سرطان مُنْذُ بداية=Answer>/متى  اكتشف مرض سرطان
<<Timex+NumYear>قَامَ بإكْتِشَاف مرض سرطان مُنْذُ بداية=Answer>/متى  اكتشف مرض سرطان
<<Timex+Decade>قَامَ بإكْتِشَاف مرض سرطان مُنْذُ بداية=Answer>/متى  اكتشف مرض سرطان
<<Timex+Date>قَامَ بإكْتِشَاف مرض سرطان مُنْذُ نهاية=Answer>/متى  اكتشف مرض سرطان
<<Timex+Duration>قَامَ بإكْتِشَاف مرض سرطان مُنْذُ نهاية=Answer>/متى  اكتشف مرض سرطان
<<Timex+Period>قَامَ بإكْتِشَاف مرض سرطان مُنْذُ نهاية=Answer>/متى  اكتشف مرض سرطان
<<Timex+NumYear>قَامَ بإكْتِشَاف مرض سرطان مُنْذُ نهاية=Answer>/متى  اكتشف مرض سرطان
<<Timex+Decade>قَامَ بإكْتِشَاف مرض سرطان مُنْذُ نهاية=Answer>/متى  اكتشف مرض سرطان
<<Timex+Date>قَامَ بإكْتِشَاف مرض سرطان مُنْذُ أواخر=Answer>/متى  اكتشف مرض سرطان
<<Timex+Duration>قَامَ بإكْتِشَاف مرض سرطان مُنْذُ أواخر=Answer>/متى  اكتشف مرض سرطان
<<Timex+Period>قَامَ بإكْتِشَاف مرض سرطان مُنْذُ أواخر=Answer>/متى  اكتشف مرض سرطان
<<Timex+NumYear>قَامَ بإكْتِشَاف مرض سرطان مُنْذُ أواخر=Answer>/متى  اكتشف مرض سرطان
<<Timex+Decade>قَامَ بإكْتِشَاف مرض سرطان مُنْذُ أواخر=Answer>/متى  اكتشف مرض سرطان
<<Timex+Date> قَامَ بإكْتِشَاف مرض سرطان مُنْذُ=Answer>/متى  اكتشف مرض سرطان
<<Timex+Duration> قَامَ بإكْتِشَاف مرض سرطان مُنْذُ=Answer>/متى  اكتشف مرض سرطان
<<Timex+Period> قَامَ بإكْتِشَاف مرض سرطان مُنْذُ=Answer>/متى  اكتشف مرض سرطان
<<Timex+NumYear> قَامَ بإكْتِشَاف مرض سرطان مُنْذُ=Answer>/متى  اكتشف مرض سرطان
<<Timex+Decade> قَامَ بإكْتِشَاف مرض سرطان مُنْذُ=Answer>/متى  اكتشف مرض سرطان
```

GRAM = test-factoid2

200/200

0,3 sec

Figure 5: Result of Generating answer patterns

This is due to the fact that this assessment is mainly based on the results of the NER module.

Discussion

Errors are often due to the complexity of user's questions or the absence of their structure in our system In fact, Arabic sentences are usually very long, which sets up obstacles for question analysis. Despite the problems described above, the developed method seems to be adequate and shows very encouraging extraction rates. However, other rules must be added to improve the result.

5.2 Generating answer patterns

For the already described example, we generate hundreds of answer pattern combinations (Figure 5). At this stage of our research, we can't deny that some further patterns are not yet covered by our grammar. This is due to the fact that this assessment is mainly based on the results of the question analysis module and the NER module.

Despite the problems described above, the developed system seems to be adequate and shows very encouraging extraction rates. However, other rules and other keywords (synonyms, supports verbs, etc.) have to be improved in our next steps.

6 Conclusion

Arabic QA systems could not match the pace due to some inherent difficulties with the language itself, as well as the lack of tools offered to support researchers. The task of our QA system can be divided into three phases: question analysis, answer pattern generation, and answer extraction. Each of these phases plays crucial roles in overall performance of the QA system. In this paper, we focused on the first two phases: question analysis and answer pattern generation.

In the near future, we aim to apply the generated patterns to a real corpus in order to deal with the answer extraction phase. We will consider such methods used in answer extraction including tools, evaluation, and corpus.

This will show the viability of the current research results and give real answers to end users. Finally, as a long term ambition, we intend to consider processing “why” and “how” question types.

References

Sman Bekhti and Maryam Alharbi. 2013. Aquasys: A question answering system for Arabic. In Proceedings of WSeas International Conference. *Recent Advances in Computer Engineering Series*, no. 12. WSEAS, pp 130-139

Yassine Benajiba, Paolo Rosso and Abdelouahid Lyhyaoui. 2007. Implementation of the ArabiQA Question Answering System’s components. In *Proceedings of the Workshop on Arabic Natural Language Processing, 2nd Information Communication Technologies Int. Symposium*, ICTIS-2007, Fez, Morroco, pp 3-5.

Wissal Brini, Mariem Ellouze, Slim Mesfar and Lamia Belguith. 2009. An Arabic question-answering system for factoid questions. In *Natural Language Processing and Knowledge Engineering*, 2009. NLP-KE 2009. International Conference on. IEEE, 2009, pp 1-7.

Bassam Hammou, Hani Abu-Salem and Steven Lytinen. 2002. QARAB: A Question answering system to support the ARABic language. In *Proceedings of the workshop on Computational approaches to Semitic languages*, ACL, Philadelphia, pp 55-65.

Heba Kurdi, Sara Alkhaider and Nada Alfaifi. 2014. Development and evaluation of a web based question answering system for Arabic language. In *Computer Science & Information Technology* (CS & IT), vol. 4, no. 2, pp 187-202.

Slim Mesfar. 2007. Named Entity Recognition for Arabic Using Syntactic Grammars. *NLDB 2007*, pp 305-316.

Slim Mesfar. 2010. Towards a Cascade of Morpho-syntactic Tools for Arabic Natural Language Processing. *CICLing* 2010, pp 150-162

Max Silberztein. 2015. *La formalisation des langues : l’approche de NooJ*. Londres: ISTE.

Omar Trigui, Lamia Hadrich Belguith and Paolo Rosso. 2010. DefArabicQA: Arabic Definition Question Answering System. In *Proceedings of the Workshop on Language Resources and Human Language Technologies for Semitic Languages*, 7th LREC, Valletta, Malta, pp 40-45.