An Arabic Question-Answering system for factoid questions

Wissal BRINI[1], Mariem ELLOUZE[1], Slim MESFAR[2], Lamia HADRICH BELGUITH[1]

[1]LARIS- MIRACL, FSEGS - University of Sfax, Tunisia
wissal.brini@voila.fr, mariem.Ellouze@planet.tn, l.belguith@fsegs.rnu.tn
[2]ISI, University of El-Manar, Tunisia
mesfarslim@yahoo.fr

In this paper, we propose an Arabic Question-Answering (Q-A) system called QASAL «Question -Answering system for Arabic Language». QASAL accepts as an input a natural language question written in Modern Standard Arabic (MSA) and generates as an output the most efficient and appropriate answer. The proposed system is composed of three modules: A question analysis module, a passage retrieval module and an answer extraction module. To process these three modules we use the NooJ Platform which represents a linguistic development environment.

Index Terms*—*Arabic language, factoid questions, Natural language processing, Question-Answering system.

I. Introduction

The amount of available information is becoming very huge, especially with the Web proliferation. The problem faced by the user is not the lack of documents or information but is the lack of time to find a short and precise answer among the variety of available documents. So, the information precision became of a great importance.

Information Retrieval (IR) systems are designed to retrieve the documents which are estimated to be relevant to the user's query. Search engines, for example, offer a lot of links toward Web pages, but are not able to provide an exact answer. . Therefore, these systems are unable to satisfy the users who are interested in obtaining a simple answer to a specific question [4]. Thus, a new need is emerged: the possibility of obtaining a unique, brief and concise answer. It is the main goal of Q-A systems, which aim to retrieve small pieces of texts that contain the answer to the question rather than the list of documents, traditionally returned by search engines.

We begin this paper with a short overview of Q-A systems. Then, we present the main steps of our Q-A system. After that, we present the implementation of our system using NooJ. Finally, we conclude this work and present some perspectives.

II. Related work to Q-A systems

A. For Latin languages

Q-A systems are relatively advanced for Latin languages. Indeed, many implementations in this field exist, among them, we cite:

QALC[1] [7]: is a Q-A system for English factoid questions in open domain. This system uses a syntactic and semantic analysis for each question. Nevertheless, it presents some errors due to incomplete syntactical rules.

QUANTUM[2] [12]: is a bilingual Q-A system that takes a natural language question as an input, looks for the answer in a 1-million document collection, and generates five answer suggestions of up to 50 characters each. The question analysis module is robust, but the bilingualism causes a drop in its performance.

Œdipe [3]: is a French Q-A system developed by the LIC2M which applies a set of morpho-syntactical patterns. However, it presents a minimalist approach concerned the used tools.

B. For Arabic languages

Q-A systems for Arabic language are very few because of many reasons. Mainly, it is due to the lack of accessibility to linguistic resources, such as corpora and basic NLP tools (Tokenizers, Morphological analyzers, etc.). Moreover, Arabic language has a very complex morphology (inflectional and derivational characteristics) and the current texts suffer from the scarcity of vowels inside as well as the absence of capitalization. These specificities Arabic language introduce many processing problems related to the word tokenization, the identification and categorization of named entities, etc.

To our knowledge, there are only three research works on Arabic Q-A systems:

AQAS [11]: is a knowledge-based Q-A system

[1] QALC: Question Answering program of the Language and Cognition group at LIMSI-CNRS.

[2] QUANTUM: Question Answering Technology of the University of Montréal.

978-1-4244-4538-7/09/$25.00 ©2009 IEEE

that extracts answers only from structured data and not from raw text (not structured text written in natural language); to our knowledge, no evaluation results has been published for AQAS system.

QARAB [9]: is an Arabic Q-A system that uses IR (Information Retrieval) and NLP (Natural Language Processing) techniques. QARAB system reached a precision of 97.3% and also a recall of 97.3%. The evaluation was done directly by four native Arabic speakers who presented 113 questions to the system and judged themselves the correctness of the answers. Note that such accuracy was not achieved in any other language in the Q-A state-of-the-art. We think that the obtained results could be reliable if a test-bed of questions in Arabic were provided in order to allow a comparison between different Q-A systems.

ArabiQA [5]: is an Arabic Q-A system that embeds an Arabic Named Entity Recognition (NER) system called ANERsys and it adapts the Java Information Retrieval System (JIRS) to extract passages from Arabic documents. Nevertheless, Question analysis and Answer extraction modules are not built yet [13].

III. FACTOID QUESTIONS

This paper focuses on the problem of finding document snippets that answer a particular category of facts-seeking questions, namely Factoid questions; simple interrogative sentences which await an answer related to a named entity. Examples of such questions are "When was Ibn Khaldoun born?", "Where is Djerba Island located?", "How much costs your wedding dress?" or "Who is the Tunisian President?".

The system which will be described in the next section will recognize the following set of question types (cf. Table 1):

TABLE I
FACTOID QUESTION TYPES TO BE PROCESSED

Question types	***Expected answer types***
من Who	شخص Person name
متى When	زمانTemporal expressions
أين Where	مكان Localization names or Organization names
كم How much	كميّة Numeric expressions

The choice of Factoid questions versus other types of questions is motivated by the following factors:

-A considerable percentage of the questions actually submitted on the Web using search engines are Factoid questions. Current search engine technology does little to support these questions because most search engines 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 Q-A track at TREC[3] (Text REtrieval Conference). The percentages of questions that are Factoid questions grew in TREC;

-Most recent approaches to open-domain Q-A use NER as core technology for detecting candidate answers [1].

In fact, the main task of the NER technology is to allow the identification and categorization of proper names as well as temporal and numeric expressions, in an open-domain text. NER systems proved to be very important for many tasks in NLP such as IR and Q/A tasks. First, this technology was introduced during the 6th Message Understanding Conference (MUC-6) which is a conference sponsored by the Defense Advanced Research Projects Agency (DARPA). According to this conference, we distinguish three types of entities to be recognized and categorized [8]:

-- ENAMEX: Detection and classification of proper names and acronyms. The classes considered in this sub-task are:

- Organization: named corporate, governmental, or other organizational entities such as "Bridgestone", "Mips" or "Language Computer Corporation".

- Person: named person or family such as "Mahatma Ghandi", "Marie Curie" or "Bill Clinton".

- Location: name of politically or geographically defined location (cities, provinces, countries, international regions, bodies of water, mountains, etc.) such "Morocco", "Italy" or "Spain".

-- TIMEX: Detection and classification of temporal expressions. The classes considered in this

[3] http://trec.nist.gov/

sub-task are:

- Date: complete or partial date expression such as "January 2008", "summer".
- Time: complete or partial expression of time of day such as "5 p.m.", "eleven o'clock" or "12h45 a.m.".

-- **NUMEX:** Detection and classification of numeric expressions monetary expressions and percentages. The classes considered in this sub-task are:

- Money: monetary expression such as "9,000 Euros", "million-dollar" or "$16,000".
- Percent: percentage such "5%", "20 pct" or "20.3%".

The problem of proper names identification is particularly difficult for Arabic language. It's due to some specific problems related to Arabic NER:

- *Non-Vocalisation :* It is due to a lack of short vowels in usual texts from which a high degree of ambiguity ensues. In theory, only the Koran, and children's books are fully vowelled; our automatic analysis allows parsing of fully vowelled, partially vowelled and unvowelled texts. Non-vocalisation can affect NER system when potential vocalizations can lead to different senses which can designate trigger words for two or more different NE type such as the case of unvowelled form "مؤسسة" [mowass'sat] that can accept, between others, the two vocalizations:

" مُؤَسَّسَة " [mowassasat – a company] => trigger word of an organization name.

" مُؤَسِّسَة " [mowassisat – a founder] => trigger word of a personal name.

- *Lack of capitalization:* The problem of identifying named entities is particularly difficult for Arabic, since names in the Arabic language do not start with capital letters, so we can not mark them in the text by looking at the first letter of the word [2].

- *Delimitation Problems:* They are related to the lack of information about unknown words with NEs, an antonomastic usage where proper names are substituted with a phrase or conversely as well as the presence of some homonyms[4] which increases ambiguity when trying to mark NE constituents [10] such as:

- "أَشْرَف" [achrafa] which can be a first name, an inflected verbal form meaning "he supervised", an elatif adjective which means "the most honorable", etc.
- "أَحْمَد" [ahmadu] which can be a first name, an inflected verbal form meaning "I thank", etc.

IV. QASAL Architecture

We propose QASAL system, a Q-A System for Arabic Language, which uses NLP techniques to process Arabic documents. QASAL processes factoid questions (i.e., simple interrogative sentences which relate to facts and generally expect a named entity as an answer).

QASAL has a pipeline architecture consisting of three components: Question analysis, Passage retrieval, and Answer extraction module (see FIG 1).

(i) Question analysis module: this module accepts, as input, any Arabic factoid question. Then, in order to look for the best answer, it gives the maximum amount of information from the concerned question, such as the expected answer type (used by the Answer extraction module), the question focus (i.e., the named entities appearing in the question, which can play an important role in the extraction of potential answers) and the list of important keywords (used by the passage retrieval module as a query).

The analysis of the question: متى استقلت تونس ؟ (When Tunisia became independent?) generates the following information :

- Expected answer type: Time,
- Focus of the question: تونس (Tunisia : a named entity),
- Keywords: استقلّت (Become independent: Verb).

(ii) Passage retrieval module: this second module is the core of the system. It retrieves the passages which are estimated as relevant to contain the expected answer. Indeed, text passages are an important intermediary between full documents and exact answers. They form a very natural unit of response for Q-A systems. Thus, the quality of a Q-A system heavily depends on the effectiveness of the second step of the pipeline: if the system fails to find any relevant documents for a question, further processing steps to extract an answer will inevitably fail too.

(iii) Answer extraction module: extracts the answer from the retrieved passage taking into account the constraint of the Question analysis module. The task has to take into consideration the type of answer expected by the user, and this means that the Answer extraction module should perform differently for each type of question.

The following figure shows the three steps of our system QASAL:

[4] A homonym is a word that has the same pronunciation and spelling as another word, but a different meaning.

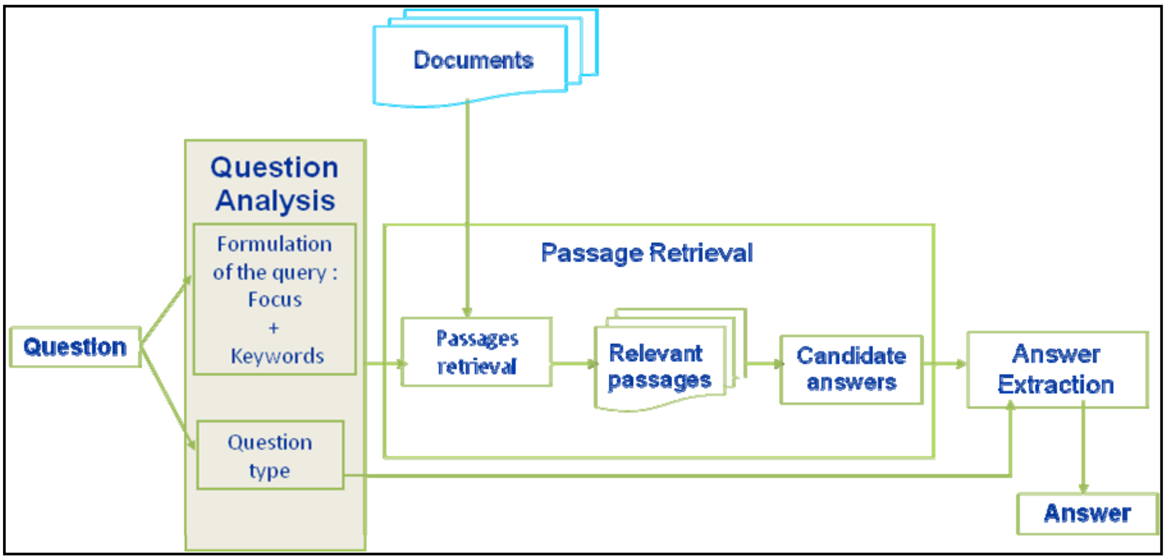


FIG. 1 QASAL architecture

V. QASAL IMPLEMETATION

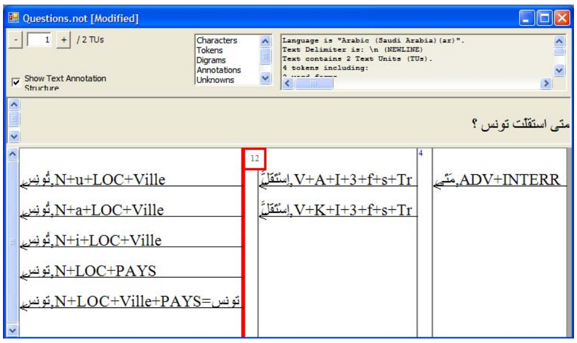
The implementation of QASAL system described below is based on the use of the NooJ’s linguistic engine. NooJ is a linguistic environment that includes large-coverage dictionaries and grammars, and parses corpora in real time. It includes tools to create and maintain large-coverage lexical resources, as well as morphological and syntactic grammars. Dictionaries and grammars are applied to texts in order to locate morphological, lexical and syntactic patterns and tag simple and compound words. Then, each recognized form is associated by the lookup algorithm of NooJ with a set of linguistic information: lemma, POS tag, gender and number, syntactic information (e.g. + Transitive), distributional information (e.g. + Loc), etc [10].

In the framework of our Q-A system, we use all developed linguistic resources included into the NooJ’s Arabic module[5] [14]. These resources include:

- an electronic dictionary, named "El-DicAr"[6], which links all the inflectional, morphological, and syntactic-semantic information to its list of lemmas,
- a morphological analyzer that identifies the component morphemes of the agglutinative forms using large coverage morphological grammars,
- a spell-checker that corrects the most frequent typographical errors,
- a named entity recognition tool which is based on a combination of the morphological analysis results and a set of rules described into local grammars,
- an automatic annotator which shows the output of application of all linguistic resources using the NooJ’s linguistic engine. These outputs are shown in FIG 2.
- as well as some tools for linguistic research and contextual exploration.

In this section, we will take advantage of all these linguistic resources to introduce the implementation of the three components of QASAL system:

(i) Question analysis: First, we apply the set of linguistic resources to the input question in order to annotate it. For example, FIG 2 shows the NooJ’s text annotation structure that gives the linguistic analysis of each word form in our sample question: ؟ متى استقلّت تونس (“When Tunisia became independent?”).



```
Language is "Arabic (Saudi Arabia)(ar)".
Text Delimiter is: \n (NEWLINE)
Text contains 2 Text Units (TUs).
4 tokens including:
```

متى استقلّت تونس ؟

تُونِس,N+u+LOC+Ville
تُونِس,N+a+LOC+Ville
تُونِس,N+i+LOC+Ville
تونس,N+LOC+PAYS
تونس,N+LOC+Ville+PAYS=تونس
إستقلّ,V+A+I+3+f+s+Tr
إستقلّ,V+K+I+3+f+s+Tr
متى,ADV+INTERR

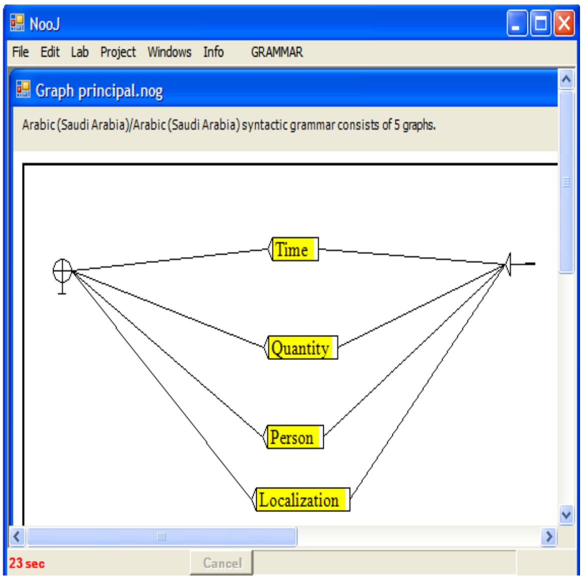
FIG. 2 Text Annotation Structure: متى استقلّت تونس ؟

[5] NooJ has already a dozen of modules including the French, English, Chinese, Hungarian, Arabic ones. (see http://www.nooj4nlp.net).

[6] « El-DicAr »: Electronic Dictionary for Arabic.

Next, we carry out a question analysis step based on some local grammars built as a set of Augmented Transition Networks (ATNs) using the NooJ's graph Editor. These grammars could translate each question into one or more Regular Expressions (RegEx) in order to represent its corresponding answer pattern(s).

In fact, the NooJ's local grammars are used to represent words sequences described by manually created rules, and then produced some kind of linguistic information such as type of the answer pattern(s). For this step, we notice that, in the present work, we are only dealing with factoid questions. Thus, the main graph (see FIG 3) of our question analysis grammar contains the four corresponding sub-graphs: Time, Quantity, Person and Localization. These sub-graphs process, respectively, questions starting with: When "متى ", How much "كم ", Who "من" and Where "أين".



Arabic (Saudi Arabia)/Arabic (Saudi Arabia) syntactic grammar consists of 5 graphs.

FIG. 3 NooJ syntactic grammar: Graph principal

For reading facility reasons, we give in the following figure a simplified version of the first question type: *Time questions*. In this grammar, we try to describe the different time based question, given as input, in order to produce the corresponding answer patterns. For example, if we consider the previous question متى استقلّت تونس ؟ ("When Tunisia became independent?") which have been already analyzed and annotated (cf. FIG 2), we can see that it follows the 3rd path in our grammar. This analysis will produce three different regular expressions (see FIG 4):

1. $V $N (<PREP> +<E>) <TIMEX>
2. <تَحَصَّل> $N (<PREP>+<E>) <اِسْتِقْلَال> (<PRON>+<E>) (<PREP>+<E>) <TIMEX>
3. <تَحَصَّل> $N (<PREP>+<E>) <PREF> <اِسْتِقْلَال> (<PREP> + <E>) <TIMEX>

Inside, these regular expressions, we notice that:

- $V: corresponds to the value of the verbal word form in the input question : استقلّت (became independent),
- $N : corresponds to the value of the nominal word form in the input question : تونس (Tunisia),
- (<PREP> + <E>) : represents an eventual presence of a preposition (<PREP>),
- (<PRON>+<E>) : represents an eventual presence of a pronoun,
- <TIMEX>: represents a call to any temporal expression already annotated in the answer text using the named entity recognition tool in NooJ.

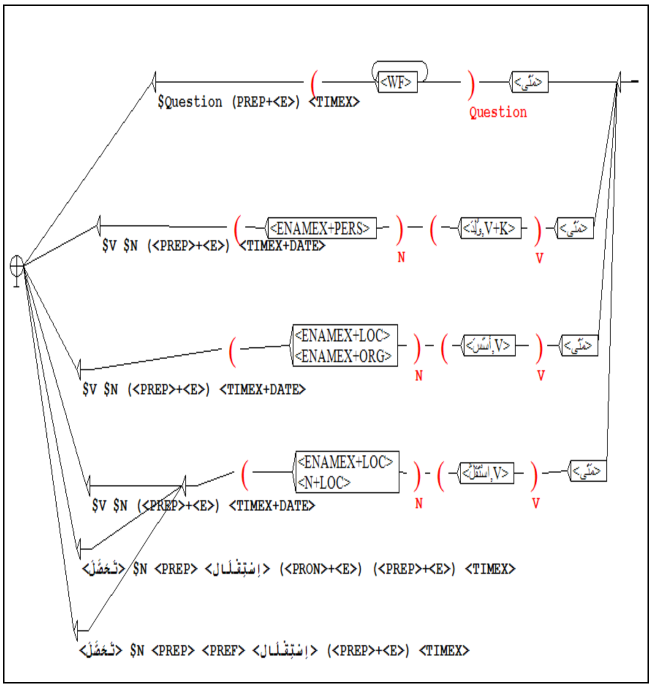
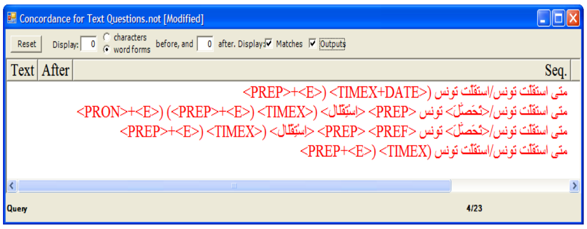

$Question (PREP+<E>) <TIMEX>
<WF>
<متى>
Question
$V $N (<PREP>+<E>) <TIMEX+DATE>
<ENAMEX+PERS>
N
<وُلِدَ,V+K>
V
<متى>
<ENAMEX+LOC>
<ENAMEX+ORG>
N
<أسس,V>
V
<متى>
$V $N (<PREP>+<E>) <TIMEX+DATE>
<ENAMEX+LOC>
<N+LOC>
N
<استقل,V>
V
<متى>
$V $N (<PREP>+<E>) <TIMEX+DATE>
<تَحَصَّل> $N <PREP> <إسْتِقْلَال> (<PRON>+<E>) (<PREP>+<E>) <TIMEX>
<تَحَصَّل> $N <PREP> <PREF> <إسْتِقْلَال> (<PREP>+<E>) <TIMEX>

FIG. 4 NooJ syntactic grammar (Time sub-graph)

Then, as shown in the following figure, the question analysis task allows the generation of all the potential answer pattern regular expression(s).



Concordance for Text Questions.not [Modified]

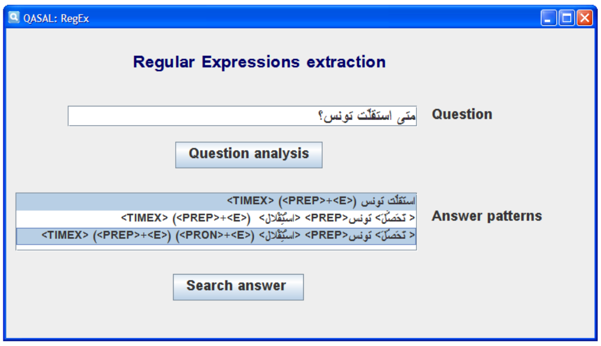
Reset Display: 0 ○ characters ● word forms before, and 0 after. Display: ☑ Matches ☑ Outputs

Text	After	Seq.
		متى استقلّت تونس/استقلّت تونس (<TIMEX+DATE> (<PREP>+<E>)
		متى استقلّت تونس/<تحصّلَ> تونس <PREP> <استِقلال> (<TIMEX> (<PREP>+<E>) (<PRON>+<E>)
		متى استقلّت تونس/<تحصّلَ> تونس <PREF> <PREP> <استِقلال> (<TIMEX> (<PREP>+<E>)
		متى استقلّت تونس/استقلّت تونس (TIMEX> (<PREP+<E>)

Query 4/23

FIG. 5 Concordance table for the factoid question : متى استقلّت تونس ؟

These regular expressions will be extracted automatically and shown in the corresponding list box in this application interface.



QASAL: RegEx

Regular Expressions extraction

Question: متى استقلّت تونس؟

Question analysis

Answer patterns:

استقلّت تونس (<PREP>+<E>) <TIMEX>

< تحصّلَ> تونس<PREP> <استِقلال> (<PREP>+<E>) <TIMEX>

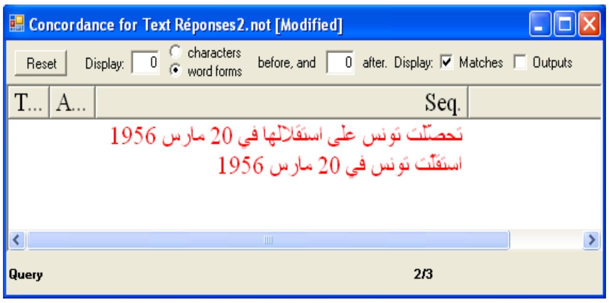
< تحصّلَ> تونس<PREP> <استِقلال> (<PRON>+<E>) (<PREP>+<E>) <TIMEX>

Search answer

FIG. 6 Interface of Regular Expressions extraction.

(ii) Passage retrieval: The first task of this step could be the selection of one or more regular expressions between those automatically generated. After that, these expressions are applied to the answer text in order to identify the potential answer(s).

The detected answers are displayed within a concordance table. The following figure shows the concordance table for answer to the factoid analyzed question: متى استقلت تونس؟ ("When Tunisia became independent?").



Concordance for Text Réponses2.not [Modified]

Reset Display: 0 ○ characters ● word forms before, and 0 after. Display: ☑ Matches ☐ Outputs

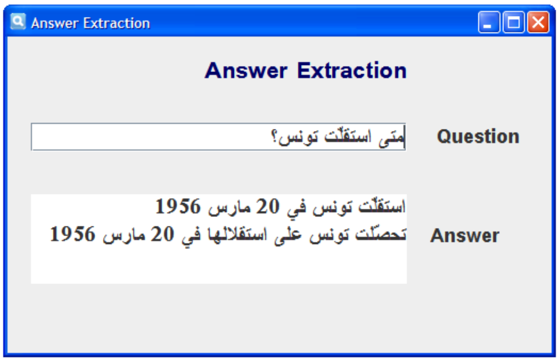
T...	A...	Seq.
		تحصّلت تونس على استقلالها في 20 مارس 1956
		استقلّت تونس في 20 مارس 1956

Query 2/3

FIG.7 Concordance table for the answer

(iii) Answer Extraction: this last step uses the displayed concordance table to automatically extract the answer of the input question.

The next figure shows the final result of our Q-A process.



Answer Extraction

Answer Extraction

Question: متى استقلّت تونس؟

Answer:

استقلّت تونس في 20 مارس 1956

تحصّلت تونس على استقلالها في 20 مارس 1956

FIG. 8 Answer Extraction for the factoid question:متى استقلّت تونس ؟

For the present work, we work in an ad-hoc situation. We consider that we have already stored all answers for the given questions. Thus, we are trying to deal a full coverage of almost one hundred questions. For the next step, we are planning to extend our application to allow the users asking for questions which were not already processed and described into analysis local grammars.

VI. Conclusion and Further Work

In this paper, we proposed an Arabic Question-Answering System called QASAL for factoid questions using NooJ local grammars. QASAL takes advantage of some linguistic techniques from IR and NLP to process a collection of Arabic text documents containing factoid questions as well as their different answers. The overall success of the system is limited to the amount of available tools developed for the Arabic language inside the NooJ's Arabic module.

As perspectives, we intend to extend our system for definition question and to use the Arabic WordNet [6] in order to allow a semantic query expansion.

References

[1] S. Abney, M.Collins and A. Singhal, 2000. "Answer extraction". In Proceedings of the 6th Applied Natural Language Processing Conference (ANLP-2000), 296–301.

[2] S. Abuleil, "Extracting Names From Arabic Text For Question-Answering Systems", Chicago State University.

[3] A. Balvet, M. Embarek, FERRET. O, "Minimalisme et question-réponse: le système OEdipe". TALN 2005, Dourdan, 6-10 juin 2005.

[4] Y. Benajiba, "Arabic Named Entity Recognition", Ph.D. dissertation, Polytechnical University of Valencia, Spain, 2009, 206p.

[5] Y. Benajiba, "Arabic Question Answering", Diploma of advanced studies, Spain : Polytechnical University of Valencia, 2007, 42 p.

[6] W. Black, S. Elkhateb, H. Rodriguez, M. Alkhalifa, P. Vossen, A. Pease and C. Fellbaum, (2006). "Introducing the Arabic WordNet Project". In Proceedings of the Third International WordNet Conference, Sojka, Choi, Fellbaum and Vossen eds.

[7] O. Ferret, B. Grau, M. Hurault-Plantet, "Finding an answer based on the recognition of the question focus". Actes de la conférence TREC-10, Gaithersburg MN.

[8] R. Grishman, B. Sundheim, Design of the MUC-6 Evaluation. In Proc. Of the 6th Conference on Message Understanding, pages 1-11, 1995.

[9] B. Hammo, H. Abu-Salem, S. Lytinen, "QARAB: A Question Answering System to Support the Arabic Language". In: Proc. Of the workshop on computational approaches to Semitic languages, ACL, pages 55-65, Philadelphia, 2002.

[10] S. Mesfar, "Named Entity Recognition for Arabic Using Syntactic Grammars". In: Natural Language Processing and Information Systems: Springer Berlin / Heidelberg, 2007, p. 305-316, ISBN 978-3-540-73350-8.

[11] F. A. Mohammed, K. Nasser, H. M. Harb, "A knowledge-based Arabic Question Answering System (AQAS)". IN: ACM SIGART Bulletin, pp. 21-33, 1993.

[12] L. Plamondon, "Le système de question-réponse QUANTUM". University of Montréal, 2002, 58 p.

[13] P. Rosso, Y. Benajiba, A. Lyhyaoui, "Towards an arabic Question Answering System". In Proceedings of SRO4, 2006.

[14] S. Mesfar, "Analyse morpho-syntaxique automatique et reconnaissance des entités nommées en Arabe standard", Ph.D. dissertation, University of Franche-Comté, November 2008, 236p.