Minimal Resources for Arabic Parsing: an Interactive Method for the Construction of Evolutive Automata

Claude Audebert, Christian Gaubert [*], André Jaccarini

Maison méditerranéenne des sciences de l'homme (MMSH)
5 rue du Château de l'Horloge BP 647 13094 Aix-en-Provence, France
claude.audebert@gmail.com, jaccarini@mmsh.univ-aix.fr
[*] Institut français d'archéologie orientale du Caire (IFAO)
37 el Cheikh Aly Yousef Str., Cairo, Egypt
cgaubert@ifao.egnet.net

Abstract

We present scenarii showing the interactive construction of operators. Some grammars and their progressive refinements through the "feedback" method are given as an example: a kernel of grammars for retrieving quotations, a grammar reflecting a current syntactic operator. We also recall previously developed morphological parsers. These grammars are designed as Finite State Automata, part of them made deterministic for better performance, using the *Sarfiyya* software developed on purpose that allows many operations on FSA. Purely algorithmic, this approach uses minimal resources, is rather independent from lexicons, gives to the tool words a prominent place and bases parsing on surface structures. On the theoretical level, it aims at putting forward the specificity of Arabic language that allows to work without a lexicon (as a limit case) due to the high level of grammaticalization in this language. This work is thus of interest to the linguist who looks for the good balance between lexicon and grammar as well as to the specialist in cognitive sciences (duality between data and programs). On the practical level, this work aims at establishing a coherent methodology for the creation of multipurpose searching operators.

Information retrieval and reported discourse

As far as IR is concerned, quotations and reported discourse are undoubtedly of great use to exploit large corpora. Their automatic retrieval raises a good number of questions in Arabic corpora. We have chosen this topic and linked it to related grammars involved in solving the IR of reported discourse. Namely the syntactic structures of this type of discourse, involving specific and fundamental operators; second, a specific structure often linked with this type of discourse and showing a judgmental point of view of the speaker (the "*min al-*" structure). These grammars are as we can see underlain by syntax and they represent respectively a solution to various types of questions: information retrieval, syntax and morphology.

This paper will deal essentially in presenting scenarii showing interactive construction of operators.

In many cases it is more appropriate to speak of indirect reported discourse. This holds especially true for newspapers which will be our main source for this study. Since indirect speech is much more frequent in these papers than direct reporting of what someone said in another context. Besides, quotations and indirect speech, if they share certain elements, do not entail the same syntactic structures, as we will see below.

Reported discourse involves a speaker, the discourse he is said to have uttered and someone who reports it, usually in given circumstances. As we can see, there are numerous marks/external/material signs that can help detect an indirect reported discourse. For instance *proper names*, whether that of the author of the discourse or the one who reports it. Both of them can be accompanied by their *title* or quality or position. For instance:

أعلن الدكتور مختار خطاب وزير قطاع الأعمال العام انه يجرى حاليا تنفيذ الاجراءات النهائية لبيع

Dr. Mukhtar Khattab, Minister of Public Works, declared that the final procedures for the selling of ... are now carried out

Unfortunately, proper names in general are a source of difficulties and have to be put in a lexicon. In the type of grammars we designed, they can be analyzed as a common noun if they have an Arabic root, or as a silence, if they are foreign. In this case, however, the silence can be very useful to reveal a proper name be it of a person, or a country.

Titles also can be put in a lexicon and be used to delineate the parts of discourse quoted.

As for the circumstances of the discourse or that of the report, they can be exploited towards our goal.

One of the issues we have to address also is punctuation which is deficient and cannot be depended upon for its lack of unification/uniformity/homogenization. Quotation marks which could be of great use for pinpointing quotations are not dependable. Fortunately, indirect discourse does not entail the use of quotes.

From our line of action, the reader will gather that we adopted a **surface approach**, which guided the design of our grammars. By this we mean a number of things : 1- that these grammars are based on the morphology of the Arabic noun, *without* the introduction of a *lexicon*, 2- that consequently they account for the rules of morphology in a minimal approach. 3- We chose to represent these rules by *automata*. Because, thanks to automata, a remarkable conciseness of Arabic morphological data representations is made possible, and that, conversely it reflects the very nature of Arabic. On the methodological level, this approach is different from other contemporary approaches: purely *algorithmic*, it uses *minimal resources*, is *independent from lexicons*, gives the tool words a prominent role and bases parsing on *surface* structures.

The most prominent character to reveal *indirect reported discourse* is the **syntactic features** it involves. To summarize: the type of *verbs* introducing first, the *conjunctio*n they govern and finally the *preposition* they are construed with.

Reported discourse is introduced by *specific verbs* which are generally called in classical Arabic grammars declarative verbs. 1- They fall into two classes depending on their syntactic construction: introduced by ʾ*inna* or ʾ*anna*. In fact the first group is only represented by one verb construed with ʾ*inna* : the verb *qāla* meaning to say, tell ex *qāla lī* : he told me. All the other verbs take ʾ*anna*. This fact allows for a two branches automaton.

2- the second point to consider is the preposition governed by each verb. Some take ʿ*an*: ʿ*abbara* ʿ*an*, some take *bi* : *a□bara-hu bi* ; some ʾ*ilā* : ʾ*ašāra* ʾ*ilā* ; etc.

We have designed the automaton for each preposition.

Another approach could have been chosen that of a two branches automaton regardless of the prepositions the various verbs govern.

The *variety of approaches* accounts for the fact that grammars are not definitely set but can be considered as a particular viewpoint on the language. On the other hand one must consider their adequacy to a given aim.

3- Speaking of verbs means handling their *conjugations* in all tenses and modes. For our present purpose we will note that in most cases, the reported discourse is in the past and generally the third person singular such as in « during the meeting, the prime minister declared so and so or that the needs of the country grew... ».

But provision must be made for the other possible cases. We already have implemented the verbs conjugations at all modes and persons (Gaubert, 2001) but for our present purpose we have chosen a different approach, in order to avoid noise and because the aim is not the verbs conjugations itself but simply detecting declarative verbs.

4 - Another matter has also to be solved: the vowels of the tool word written in Arabic script *alif nūn* and which can stand for four tool words of the highest level i.e. ʾ*inna*, introducing a nominal sentence, and here preceded by *qāla* ; ʾ*anna*, introducing a subordinate nominal sentence and preceded by all the declarative verbs mentioned above ; ʾ*in*, introducing a conditional sentence ; ʾ*an* introducing a subordinate conjunctive clause.

The various syntactic structures involved and their high level, call for a disambiguization of *alīf nūn*. We have implemented this disambiguization according to various approaches since (Jaccarini 1998, Gaubert 2001) and raised the general question of the disambiguization of all ambiguous tool words (Audebert, Jaccarini 1988). As a matter of fact the tool words which we called *tokens* are at the basis of our syntactic approach based on a *top down* exploration of tokens (tool words) because they structure sentences. And also since we put ourselves in the decoding perspective linked to backtrack as a disambiguating process.

This approach had naturally led us to deal with transducers whose theory was developed by Schützenberger.

5- We will have to look further into another possible structure obtained by the erasing/ removal of ʾ*anna* and the clause it governs and its replacement by a *ma□dar* or verbal noun:

ex. أشار إلى أنها وصلت أمس

He indicated that she arrived yesterday

Transformed into:

أشار الى وصولها

He indicated her arrival yesterday

With this type of structure, the presence and the role of the preposition following the declarative verb is essential.

6- to end the various questions raised by the grammars linked to reported discourse, we must take into consideration the case when the verb is ambiguous due to the fact that it *does not introduce a quotation or a reported discourse* although it displays on the surface, all the necessary syntactic features mentioned above.

Ex : وهذا كله يوضح أن الدخول فى استثمارات

And all this makes it clear that the entrance in…

ويؤكد ذلك أن بعض الصناعات

And all this certifies that some industries…

The other case of ambiguity is illustrated by a very large class of verbs which share with the verbs introducing quotations or reported discourse the same syntactic structure but do not introduce such a discourse, like ʿ*alima*, ʿ*arafa* etc. This case calls certainly for a lexicon of the verbs likely to introduce reported discourse.

Sarfiyya, a FSA recursive parser based application dedicated to Arabic

The *Sarfiyya* software, now written in Java, includes a specifically designed Finite State Automata (FSA) parser and a set of classical and less classical tools for FSA manipulation, many of them being graphically driven trough a GUI interface (figure 1).

The parser, which is still under development, uses letter and FSA transitions, in other words it's a recursive FSA parser. It includes several enhancements:

- It is able to parse categorized FSA, meaning that each transition holds a morpho-syntactic category information, like conjugation, article presence, etc.;
- Fragments of recognized sentences can be displayed or hidden by choosing some special categories;
- It uses deterministic pre-parsing whenever it is possible to improve parsing performance: a deterministic FSA accepting the same language as its non-deterministic source is computed once and used as an acceptor. Since deterministic FSA parse strings in a linear time, if an input sentence is refused, we don't need to parse it with the fully categorized non-deterministic FSA.
- It also uses some word-based optimizations;
- It has post-treatment parsing feature for dealing with morphology, based on root control and other micro-lexicon resources. This part was discussed in (Gaubert, 2001).

A debugger giving a complete trace of the path of each solution trough the automata is a powerful tool for grammar development.

This parser is not yet designed for transducers. The automata structure is implemented with Java tables.

Sarfiyya contains a specially designed regular expression interpreter and builder. Thus the FSA can be specified by raw regular expressions, and then refined by adjusting the different categories wanted. These expressions can use FSAs among a library of previous designed, optimized and documented FSAs.

In order to be able to constitute a library of FSA and to re-use some of them, we also made provision of the possibility to compose the FSAs one with another, that means to parse the output of a first FSA by a second one, the first one acting like a kind of sieve in some respect (piping). This feature is currently being extended to the

ability of composing FSA with other filters, not necessarily pure FTS-based ones, like statistical modules.

Designing a quotation and reported speech retrieval grammar

A first grammar is an attempt to detect and catch excerpts form phrases containing quotation of reported speech. The general skeleton of this automaton can be described by the *qāla … ʾinna* construction and more precisely by the following simplified regular expression, applied to every phrase of a text:

Phrase-qala = words (ε+ف+و) cit-qala words إن (Post+ε) words end

+ is for disjunction; . is for concatenation and can be eluded; * is the Kleen star symbol; ε is the epsilon transition; s is the space character. A is the complete Arabic alphabet from *hamza* and its variants to *yāʾ*.

For the ease of reading of these expression we did simplified the combinations of Arabic characters, spaces and punctuations into a single sub-automata named "words". This automaton may be a variant of the following regular expression:

words = (A A* (s+،) (s+،) *)*

"cit-qala" represents different production of the *qāla* verb, using only the third person, present or past. These productions are compiled in a dictionary automaton recognizing these forms and only them.

cit-qala = تقول+يقول+قالوا+قالت+قال+ etc.

Post is the automaton representing all the postfixed personal pronouns. Therefore the fragment *inna* (Post+e): means « followed by Post or not ». Taking into account the very frequent use of *alif* instead of *alif hamza ta*□*t* (*alif*$_h$) for *ʾinna*, we will use (*alif*+*alif*$_h$) nun to catch both of them.

Let's define the active part of the Phrase-qala automaton as a function of a triplet {verb, preposition, conjunction}

cit (verbs, preposition, conjunction) = verbs words* preposition conjunction (Post+ε)

cit1 = cit (cit-qala, ε, إن) (figure 2)

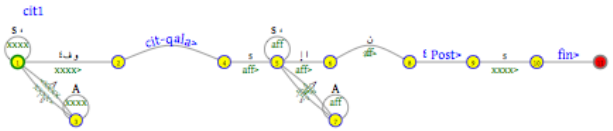

figure 2 : the *cit1* automaton

Other governments can be defined as:

cit2 = cit (cit-dir, ε, أن) where cit-dir contains the relevant forms of أضاف, أعلن, أوضح, أكد, ...

cit3 = cit (cit-bi, ب, أن) , cit-bi : وصف, صرح

cit4 = cit (cit-ʾalā, s على, أن) , cit-ʾalā : أكد, علق

cit5 = cit (cit-ʾan, s عن, أن) , cit-ʾan : أعرب, عبر, أخبر

cit6 = cit (cit-ilā, s إلى, أن) , cit-ilā : أشار, أضاف

To handle the general case of quotations and reported speech, one would naturally want these six automata to be merged together into a single multi-government automaton:

cit = words (ε+ف+و) (Σcit$_i$) (Post+ε) words end

Some factorization can then be achieved with the final ن for all the automata and the preceding s for cit4 to cit6 (figure 3).

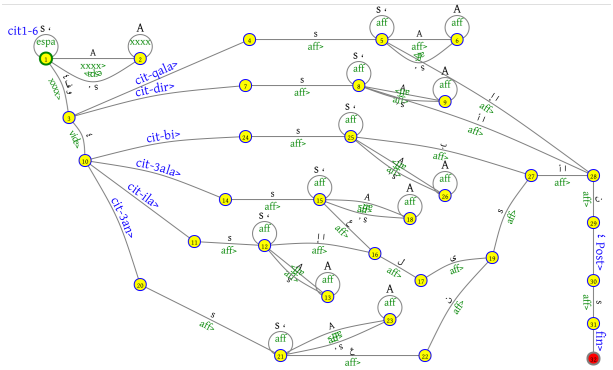

figure 3: the complete *cit* automaton

Computing the deterministic FSA?

The algorithm use for this computation is the classical one described by Aho et al. (1986), adapted to the recursive FSA: each FSA transition is recursively replaced by the letter-based transitions it contains. We call this calculus the DET function.

The 32 states long cit automata obtained is highly non-deterministic: it contains ε-transition at its very beginning, and then fans out widely in order to detect the various verbs introducing the quotations. It is therefore not surprising to observe that the deterministic automata computed from cit contains 9800 states, and that the computation itself lasts ... more than 4 hours and half on a 2.4 GHz CPU! The time grows exponentially with the non-deterministic nature of the given input automata. For instance, if we subtract the cit6 branch from cit (cit1-5), we get a 4200 states deterministic automata within 18 minutes, and subtracting again cit5 (cit1-4), we get a 1900 states DET(cit) within 7 minutes, etc.

Thought this calculus only needs to be performed once and then be saved, which we did through serialization, it would certainly deserve programming improvements, using optimized structures like hash tables, sorted trees, etc.

It is therefore a better solution in our case to parse the corpus with each of the six 6 cit automata, pre-parsed by the deterministic version of each of them, rather than calculating the big one which appears to be far too ambiguous for the present version of our system. In this case, the computation of the deterministic version of cit4 for instance lasts 0.3 s, and the sum of all the six components lasts less than 1.5 seconds. For each of these grammars (and for the large cit and cit1-5), the DET pre-process leads to save more than 90% of the total time.

Testing this grammar

We will use a small corpus of 20 various articles taken from al-Ahram (2001), for a total of 19 000 words. It is a sample of a larger corpus of more than 100 000 words that we use for deeper testing. This is a raw corpus in the sense that no preparation was made in order to align punctuations, spaces and no segmentation of phrases was achieved.

Running with an average of 1000 words by second, the cit automata finds 48 quotations in this corpus, half of them being introduced by *qāla* and another 40% using the direct construction introduced by:

أضاف، أوضح،أكد

But this first analysis also reveals a number of noises and limitations. The first one is the ambiguity of the long sentences. Many of them, beginning with a real quotation, continue with other related sentences, some of them being also quotations. Some of these sentences also contain the ʿ*anna* conjunction, and for that reason they belong to the solutions computed by the parser; in some cases this can lead to five or six noisy solutions. The following measures operating at different levels could help to solve this problem:

1- to introduce a sub automaton between the verb and the conjunction that would reject any conjunction itself, by creating an acceptor of words suite containing the conjunction and using its negation. This technique of the complement is frequently used at the character level (see the [^a-z] pattern of the *grep* regular expression engine) The problem here is that the deterministic version of the automata is only possible at the character level and exclude FSA transition. DET(cit) becomes impossible to compute, leading to performance troubles.

2- Split up the long sentences into overlapping sequences containing only one conjunction ʿ*anna* or ʿ*inna* and consider them as the input of the cit grammar in place of the complete sentence. This would constitute a kind of preparation filter that operates in a linear time.

3- Branch from our parser a non-greedy version, that would always keep the shortest transition path it computes when parsing a circular part of the automata (consider *words*).

4- Fix a hard limit to the size of the words suite that may be parsed between two states: this would require handling augmentations (transducer's tests and actions). The longest distance we've found between the verb and the conjunction was 16 words, containing a proper name with its two detailed functions and the circumstances where he uttered the speech.

Solution 2, 3 and 4 require testing and can be used together.

Another point to take into consideration – and closely related to the noise problem mentioned - is the possibility to have multi conjunction governed by one verb. Ex:

وقال زعيف إن الفتوى مازالت قائمة، وأن تدمير التماثيل أمر مؤكد لا رجعة فيه

ʿ*Anna* is also used here after *qāla* in a faulty way: we should *realistically* add cit-qala to the cit2 automata! To address the multiple conjunctions problem, we could add a loop *wa* transition coming back to ʿ*anna* or ʿ*inna*; this must be checked along with the previous mentioned solutions.

The "*min al-*" Grammar

This grammar displays a very high structural level and illustrates another aspect of the token *alif nūn*, under both forms: ʾ*anna* and ʾ*an* and the case of a high structural token.

The regular expression which specifies it is:

words (ف+و+e) من s ال A* s ان s end

It results into a 15 states FSA which compiles in a 29 deterministic DET(minel). Our corpus only contains a dozen of them, with the same proportion in the larger corpus, with the most frequent ones:

من الضرورى أنْ
من المنتظر أنْ
من الممكن أنْ

We can notice here again that some noise can be introduced when an accidental *min* belonging to a GN structure is one word close to *alif nūn*.

فالسؤال طلب من التلاميذ أن يتخيلوا

A solution would be to inventory the mostly used words for a valid "*min al-*" structure and compile it in a dictionary FSA; this would limit the noisy cases but won't certainly totally avoid them.

We also note a very frequent use of *laysa* and *yabdū* before this scheme.

Revisiting Morphology

Much have been said about arabic morphology and the use of automata since the works of Koskenniemi (1983), Beesley (1996) and recently Mefsar (2008). None of these approaches, as far as we know, avoid the use of a lexicon. Illustrating the power of automata, we showed that the famous study published by D. Cohen (1970) can be specified by a 6-states non deterministic automata. A quotient - or skeleton - language can even be defined and holds all the characteristics of a semi-natural language (Audebert, Jaccarini 1994, Jaccarini 1997).

This language can be obtained by reducing all the Arabic roots to one unique representative. Our hypothesis is that the grammaticalness of Arabic sentences is little influenced by root permutation. Arab grammarians had foreseen this phenomenon by choosing a unique paradigmatic root to represent all Arabic patterns and to organize their dictionary giving the primacy to the root.

We define the general morphological system, including its irregularities, as a transduction of the basic system. This transduction is in fact the formalization of what linguists call regularity postulate.

Another possibility to model some phenomenon in arabic conjugation is to define new categories, derived from the standard ones, and to associate them with micro-lexicon data; the output of a FSA containing these categories can be filtered with deterministic procedures. The automata are progressively developped and tuned in order to measure the influence of each transformation. This approach developed in (Gaubert 2001) covers about 98% of verbal and nominal morphology with an unavoidable but measurable noise.

We now plan to integrate and interact the previously exposed operators with these morphological automata, either directly of using the piping feature of Sarfiyya.

Kawâkib, a demonstration web site

A first version of an interactive demonstration web site named *Kawâkib* will soon be available at http://www.ifao.egnet.net/kawakib. It uses some of the grammars exposed here and combines Java and Javascript to offer interactive experience to the users.

It includes several features as the most used roots of a text, tool word and reported speech detection; the interface will be available in English, French and Arabic.

Primary designed for educational use, this tool can also be use for information retrieval purpose as well as text indexing.

Conclusion

By choosing these few grammars, we have tried to prove the necessity of the feedback method. Through *Sarfiyya* which is the kernel of a general processor of automata and which contains basic linguistic resources, we have also tried to prove the necessity of establishing a scientific and formalized method to *evaluate grammars* themselves rather than simply the results obtained. We should notice that *ambiguity*, which is inescapable, is not due to absence of the use a dictionary/lexicon. It varies according to the assigned aims and should be organized into a hierarchy. The implementation of augmentations in some basic transition networks can offer a powerful tool for solving some phrase-scope issues. Overall, automata can be considered as declarative and help clarify the transition from the declarative to the operational.

Bibliographical references

Aho, Sethi, Ullmann (1986), Compilateurs. Principes, techniques et outils. French edition 1991. InterEdition.

Audebert C, Jaccarini A. (1986) À la recherche du Ḥa*bar, outils en vue de* l'établissement d'un programme d'enseignement assisté par ordinateur, Annales islamologiques 22, Institut français d'archéologie orientale du Caire.

Audebert C, Jaccarini A. (1988). De la reconnaissance des mots outils et des tokens. Annales islamologiques 24, Institut français d'archéologie orientale du Caire.

Audebert C, Jaccarini A. (1994). Méthode de variation de la grammaire et algorithme morphologique. Bulletin d'études orientales XLVI. Damascus.

Beesley, Kenneth R. (1996). Arabic Finite-State Morphological Analysis And Generation. COLING.

Cohen, D. (1970) Essai d'une analyse automatique de l'arabe. In: David Cohen. Etudes de linguistique sémitique et arabe. Paris:Mouton, pp. 49-78.

Gaubert Chr., (2001). Stratégies et règles pour un traitement automatique minimal de l'arabe. Thèse de doctorat. Département d'arabe, Université d'Aix-en Provence.

Jaccarini A., (1997). Grammaires modulaires de l'arabe. Thèse de doctorat. Université de Paris-Sorbonne.

Koskenniemi K. (1983). Two-level Morphology. A General Computational Model for Word-Form Recognition and Production. Department of General Linguistics. University of Helsinki.

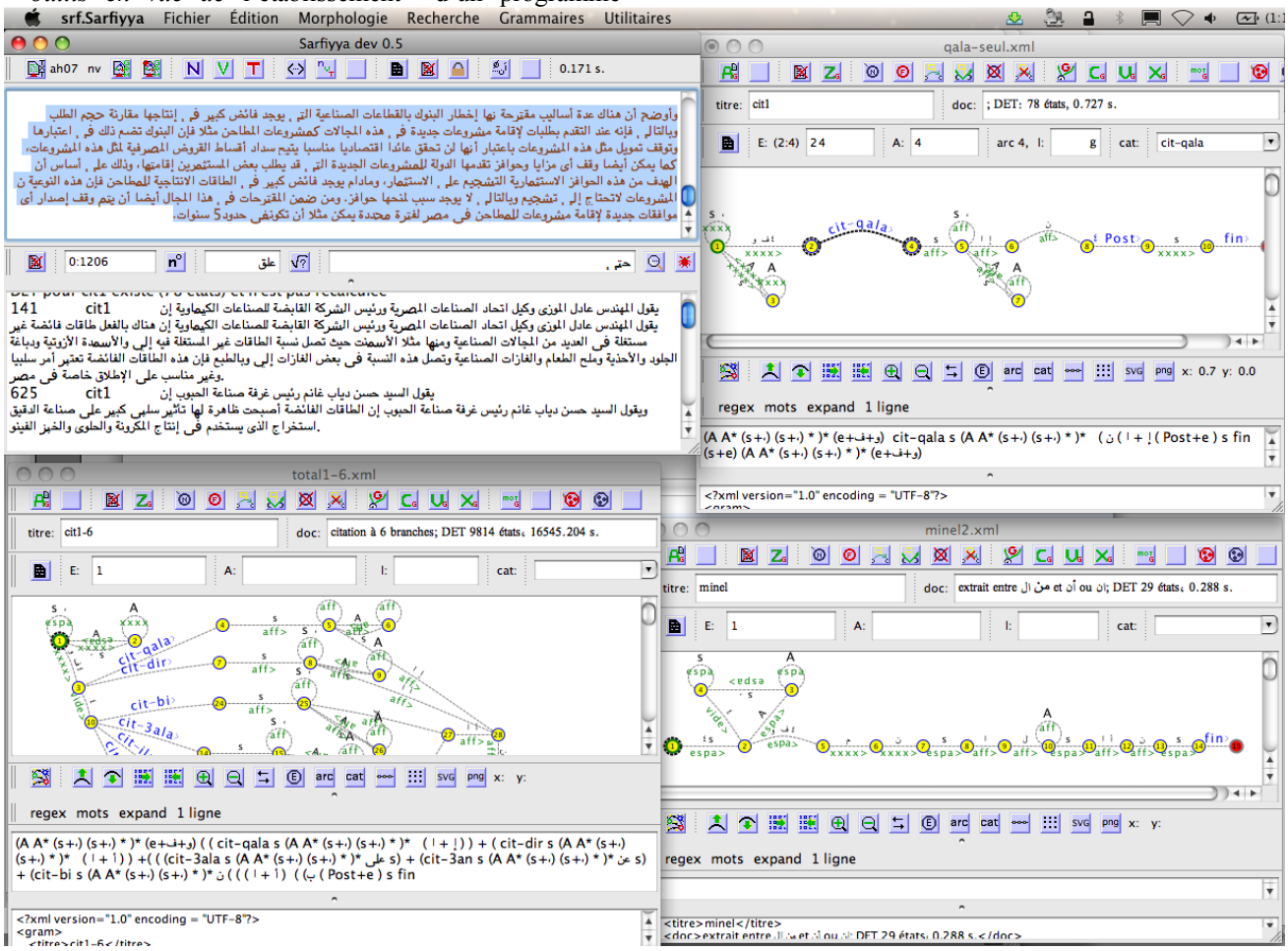

figure 1 : *Sarfiyya*