Modifying a Natural Language Processing System for European Languages to Treat Arabic in Information Processing and Information Retrieval Applications

Gregory Grefenstette, Nasredine Semmar, Faïza Elkateb-Gara
Multilingual Multimedia Knowledge Engineering Laboratory (LIC2M)
Commissariat à l'Energie Atomique, Laboratoire d'Intégration des Systèmes et des Technologies (CEA LIST)
B.P. 6, 92265 Fontenay-aux-Roses Cedex, France
{gregory.grefenstette,nasredine.semmar,faiza.gara}@cea.fr

Abstract

The goal of many natural language processing platforms is to be able to someday correctly treat all languages. Each new language, especially one from a new language family, provokes some modification and design changes. Here we present the changes that we had to introduce into our platform designed for European languages in order to handle a Semitic language. Treatment of Arabic was successfully integrated into our cross language information retrieval system, which is visible online.

1 Introduction

When a natural language processing (NLP) system is created in a modular fashion, it can be relatively easy to extend treatment to new languages (Maynard, *et al.* 2003) depending on the depth and completeness desired. We present here lessons learned from the extension of our NLP system that was originally implemented for Romance and Germanic European[1] languages to a member of the Semitic language family, Arabic. Though our system was designed modularly, this new language posed new problems. We present our answers to these problems encountered in the creation of an Arabic processing system, and illustrate its integration into an online cross language information retrieval (CLIR) system dealing with documents written in Arabic, English French and Spanish.

2 The LIMA natural language processor

Our NLP system (Besançon *et al*., 2003), called LIMA[2], was built using a traditional architecture involving separate modules for

1. Morphological analysis:
    a. Tokenization (separating the input stream into a graph of words).
    b. Simple word lookup (search for words in a full form lexicon).
    c. Orthographical alternative lookup (looking for differently accented forms, alternative hyphenisation, concatenated words, abbreviation recognition), which might alter the original non-cyclic word graph by adding alternative paths.
    d. Idiomatic expressions recognizer (detecting and considering them as single words in the word graph).
    e. Unknown word analysis.
2. Part-of-Speech and Syntactic analysis:
    a. After the morphological analysis, which has augmented the original graph with as many nodes as there

[1] European languages from non indo-European families (Basque, Finnish and Hungarian) pose some of the same problems that Arabic does.

[2] LIMA stands for the LIC2M Multilingual Analyzer.

Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, pages 31–38,
Ann Arbor, June 2005.

are interpretations for the tokens, part-of-speech analysis using language models from a hand-tagged corpus reduces the number of possible readings of the input.
   b. Named entity recognizer.
   c. Recognition of nominal and verbal chains in the graph.
   d. Dependency relation extraction.
3. Information retrieval application:
   a. Subgraph indexing.
   b. Query reformulation (monolingual reformulation for paraphrases and synonymy; multilingual for cross language information retrieval).
   c. Retrieval scoring comparing partial matches on subgraphs and entities.

Our LIMA NLP system (Besançon *et al*., 2003) was first implemented for English, French, German and Spanish, with all data coded in UTF8. When we extended the system to Arabic, we found that a number of modifications had to be introduced. We detail these modifications in the next sections.

3 Changes specific to Semitic languages

Two new problems posed by Arabic (and common to most Semitic languages) that forced us to alter our NLP system are the problem of incomplete vowelization of printed texts[3] and the problem of agglutinative clitics. We discuss how these new problems influenced our lexical resources and language processing steps.

Lexical Resources

The first task for introducing a new language is to create the lexical resources for this language. Since Arabic presents agglutination of articles, prepositions and conjunctions at the beginning of words as well as pronouns at the end of words, and these phenomena were not treated in our existing European languages[4], we had to decide how this feature would be handled in the lexicon. Solutions to this problem have been proposed, ranging from generation and storage of all agglutinated words forms (Debili and Zouari, 1985) to the compilation of valid sequences of proclitics, words and enclitics into finite-state machines (Beesley, 1996). Our system had already addressed the problem of compounds for German in the following way: if an input word is not present in the dictionary, a compound-searching module returns all complete sequences of dictionary words (a list of possible compound joining "fogemorphemes" is passed to this module) as valid decompositions of the input word. Though theoretically this method could be used to treat Arabic clitics, we decided against using this existing process for two reasons:

1. Contrary to German, in which any noun may theoretically be the first element of a compound, Arabic clitics belong to a small closed set of articles, conjunctions, prepositions and pronouns. Allowing any word to appear at the beginning or end of an agglutinated word would generate unnecessary noise.
2. Storing all words with all possible clitics would multiply the size of lexicon proportionally to the number of legal possible combinations. We decided that this would take up too much space, though others have adopted this approach as mentioned above.

We decided to create three lexicons: two additional (small) lists of proclitic and enclitic combinations, and one large lexicon of full form[5] voweled words (with no clitics), the creation of the large lexicon from a set of lemmas using classic conjugation rules did not require any modification of the existing dictionary building and compilation component. Since our NLP system already possessed a mechanism for mapping unaccented words to accented entries, and we decided to use this existing

[3] Since the headwords of our monolingual and cross-lingual reference dictionaries for Arabic possess voweled entries, we hope to attain greater precision by treating this problem. An alternative but noisy approach (Larkey et al. 2002) is to reduce to unvoweled text throughout the NLP application.

[4] Spanish, of course, possesses enclitic pronouns for some verb forms but these were not adequately treated until the solution for Arabic was implemented in our system.

[5] Our dictionary making process generates all full form versions of non compound and unagglutinated words. These are e then compiled into a finite-state automaton. Every node corresponding to a full word is flagged, and an index corresponding to the automaton path points to the lexical data for that word.

mechanism for later matching of voweled and unvoweled versions of Arabic words in applications. Thus the only changes for lexical resources involve adding two small clitic lexicons.

Processing Steps: Morphological analysis

Going back to the NLP processing steps listed in section 2, we now discuss new processing changes needed for treating Arabic. Tokenization (1a) and simple word lookup (2a) of the tokenized strings in the dictionary were unchanged as LIMA was coded for UTF8. If the word was not found, an existing orthographical alternative lookup (1c) was also used without change (except for the addition of the language specific correspondence table between accented and unaccented characters) in order to find lexical entries for unvoweled or partially voweled words. Using this existing mechanism for treating the vowelization problem does not allow us to exploit partial vowelization as we explain in a later section.

At this point in the processing, a word that contains clitics will not have been found in the dictionary since we had decided not to include word forms including clitics. We introduced, here, a new processing step for Arabic: a clitic stemmer. This stemmer uses the following linguistic resources:

- The full form dictionary, containing for each word form its possible part-of-speech tags and linguistic features (gender, number, etc.). We currently have 5.4 million entries in this dictionary[6].
- The proclitic dictionary and the enclitic dictionary, having the same structure of the full form dictionary with voweled and unvoweled versions of each valid combination of clitics. There are 77 and 65 entries respectively in each dictionary.

The clitic stemmer proceeds as follows on tokens unrecognized after step 1c:

- Several vowel form normalizations are performed (ً ٌ ٍ ّ ُ َ are removed, آ إ أ are replaced by ا and final ؤ ئ ي or ة are replaced by ء ىء ى or ه).
- All clitic possibilities are computed by using proclitics and enclitics dictionaries.
- A radical, computed by removing these clitics, is checked against the full form lexicon. If it does not exist in the full form lexicon, re-write rules (such as those described in Darwish (2002)) are applied, and the altered form is checked against the full form dictionary. For example, consider the token وهواهم and the included clitics (و, هم), the computed radical هوا does not exist in the full form lexicon but after applying one of the dozen re-write rules, the modified radical هوى is found the dictionary and the input token is segmented into root and clitics as: وهواهم = و + هوى + هم.
- The compatibility of the morpho-syntactic tags of the three components (proclitic, radical, enclitic) is then checked. Only valid segmentations are kept and added into the word graph. Table 1 gives some examples of segmentations[7] of words in the sentence من جانبها أكدت وزارة الداخلية العراقية

Agglutinated word	Segmentations of the agglutinated word
ومن	ومن = و + من
جانبها	جانبها = جانب + ها
الداخلية	الداخلية = ال + داخلية الداخلية = [ا + ل] + داخلية
العراقية	العراقية = ال + عراقية العراقية = [ا + ل] + عراقية
المحافظات	المحافظات = ال + محافظات المحافظات = [ا + ل] + محافظات
للخطف	للخطف = [ل + ال] + خطف
الوزير	الوزير = ال + وزير الوزير = [ا + ل] + وزير
نفسه	نفسه = نفس + ه

Table 1: Segmentations of some agglutinated words.

Producing this new clitic stemmer for Arabic allowed us to correctly treat a similar (but previously ignored) phenomenon in Spanish in which verb forms can possess pronominal enclitics. For example, the imperative form of "give to me" is written as "dame", which corresponds to the radical "da" followed the enclitic "me". Once we implemented this clitic stemmer for Arabic, we created an en-

[6] If we generated all forms including appended clitics, we would generate an estimated 60 billion forms (Attia, 1999).

[7] For example, the agglutinated word الداخلية has two segmentations but only the segmentation: الداخلية = ال + داخلية will remain after POS tagging in step 2a

clitic dictionary for Spanish and then successfully used the same stemmer for this European language. At this point, the treatment resumes as with European languages. The detection of idiomatic[8] expressions (step 1d) is performed after clitic separation using rules associated with trigger words for each expression. Once a trigger is found, its left and right lexical contexts in the rule are then tested. The trigger must be an entry in the full form lexicon, but can be represented as either a surface form or a lemma form combined with its morpho-syntactic tag. Here we came across another problem specific to Semitic languages. Since Arabic lexicon entries are voweled and since input texts may be partially voweled or unvoweled, we are forced to only use lemma forms to describe Arabic idiomatic expressions rules with the existing mechanism, or else enter all the possible partial vowelizations for each word in an idiomatic expression. Since, at this point after step 1c, each recognized word is represented with all its possible voweled lemmas in the analysis graph, we developed 482 contiguous idiomatic voweled expression rules. For example one of the developed rules recognizes in the text كانون الثاني (January) as a whole and tags the expression as a being a month.

After idiomatic expression recognition, any nodes not yet recognized are assigned (in step 1e) default linguistic values based on features recognized during tokenization (e.g. presence of uppercase or numbers or special characters). Nothing was changed for this step of default value assignment in order to treat Arabic, but since Semitic languages do not have the capitalization clues that English and French have for recognizing proper and since Arabic proper names can often be decomposed into simple words (much like Chinese names), the current implementation of this step with our current lexical resources poses some problems.

For example, consider the following sentence:
فرانك لامبارد يحتفل بالتسجيل لتشلسي وزميله إيدور غوديانسن يشاركه الفرحة *Frank Lampard celebrates the score by Chelsea and his team mate Eidur Gudjohnsen shares his elation*. The name فرانك (Frank) is identified as such because it is found in the lexicon; the name لامبارد (Lampard) is not in the lexicon and incorrectly stemmed as لا + مبارد (plural of the noun مبرد (grater)); the name إيدور (Eidur) is incorrectly tagged as a verb; and غوديانسن (Gudjohnsen), which is not in the dictionary and for which the clitic stemmer does not produce any solutions receives the default tags adjective, noun, proper noun and verb, to be decided by the part-of-speech tagger. To improve this performance, we plan to enrich the Arabic lexicon with more proper names, using either name recognition (Maloney and Niv, 1998) or a back translation approach after name recognition in English texts (Al-Onaizan and Knight, 2002).

Processing Steps: Part-of-speech analysis

For the succeeding steps involving part-of-speech tagging, named entity recognition, division into nominal and verbal chains, and dependency extraction no changes were necessary for treating Arabic. After morphological analysis, as input to step 2a, part-of-speech tagging, we have the same type of word graph for Arabic text as for European text: each node is annotated with the surface form, a lemma and a part-of-speech in the graph. If a word is ambiguous, then more than one node appears in the graph for that word. Our part-of-speech tagging involves using a language model (bigrams and trigrams of grammatical tags) derived from hand-tagged text to eliminate unattested or rare sub paths in the graph of words representing a sentence. For Arabic, we created a hand-tagged corpus, and where then able to exploit the existing mechanism.

One space problem that has arisen in applying the existing processing designed for European languages comes from the problem of vowelization. With our previous European languages, it was extremely rare to have more than one possible lemmatization for a given pair: (surface form, grammatical part-of-speech tag)[9]. But, in Arabic this can be very common since an unvoweled string can correspond to many different words, some with the same part-of-speech but different lemmas. The effect of this previously unseen type of ambiguity on our data structures was to greatly increase the word graph size before and after part-of-speech tagging. Since each combination of (sur-

[8] An *idiom* in our system is a (possibly non-contiguous sequence) of known words that act as a single unit. For example, *made up* in *He made up the story on the spot*. Once an idiomatic expression is recognized the individual words nodes are joined into one node in the word graph.

[9] One example from French is the pair (*étaient*, finite-verb) that can correspond to the two lemmas: *être* and *étayer*.

face-form, part-of-speech-tag, and lemma) gives rise to a new node, the graph becomes larger, increasing the number of paths that all processing steps must explore. The solution to this for Arabic and other Semitic languages is simple, though we have not yet implemented it. We plan to modify our internal data structure so that each node will correspond to the surface form, a part-of-speech tag, and a set of lemmas: (surface-form, part-of-speech-tag, {lemmas}). The inclusion of a set of possible lemmas, rather than just one lemma, in a node will greatly reduce the number of nodes in the graph and speed processing time.

The next step in our NLP system, after part-of-speech tagging, is named entity recognition (Abuleil and Evans, 2004) using name triggers (e.g., President, lake, corporation, etc.). Beyond the problem mentioned above of distinguishing possible proper nouns, here we had an additional problem since our recognizer extracted the entity in its surface form. Since in Arabic, as in other Semitic languages, the input text is usually only partially voweled, this gave rise to many different forms (corresponding to different surface forms) for the same entity. This minor problem was solved by storing the fully voweled forms of the entities (for application such as information retrieval as shown below) rather than the surface form.

After named entity recognition, our methods of verbal and nominal chain recognition and dependency extraction did not require any modifications for Arabic. But since the sentence graphs, as mentioned above, are currently large, we have restricted the chains recognized to simple noun and verb chunks (Abney, 1991) rather than the more complex chains (Marsh, 1984) we recognize for European languages. Likewise, the only dependency relations that we extract for the moment are relations between nominal elements. We expect that the reduction in sentence graph once lemmas are all collected in the same word node will allow us to treat more complex dependency relations.

4 Integration in a CLIR application

The results of the NLP steps produce, for all languages we treat, a set of normalized lemmas, a set of named entities and a set of nominal compounds (as well as other dependency relations for some languages). These results can be used for any natural language processing application. For example, we have integrated LIMA as a front-end for a cross language information retrieval system. The inclusion of our Arabic language results into the information retrieval system did not necessitate any modifications to this system.

This information retrieval (IR) application involves three linguistic steps, as shown in section 2. First, in step 3a, subgraphs (compounds and their components) of the original sentence graph are stored. For example, the NLP analysis will recognize an English phrase such as "management of water resources" as a compound that the IR system will index. This phrase and its sub-elements are normalized and indexed (as well as simple words) in the following head-first normalized forms:

- management_water_resource
- resource_water
- management_resource

Parallel head-first structures are created for different languages, for example, the French "gestion des ressource en eau" generates:

- gestion_ressource_eau
- ressource_eau
- gestion_ressource.

The corresponding Arabic phrase: إدارة موارد المياه is likewise indexed with voweled forms:

- مَاء_مَـوْرِد_ إدَارَة
- مَاء_مَـوْرِد
- مَـوْرِد_ إدَارَة

When a question is posed to our cross language IR (CLIR) system it undergoes the same NLP treatment as in steps 1a to 3a. Then the query is reformulated using synonym dictionaries and translation dictionaries in step 3b. For Arabic, we have not yet acquired any monolingual synonym dictionaries, but we have purchased and modified cross-lingual transfer dictionaries between Arabic and English, Arabic and French, and Arabic and Spanish[10]. When a compound is found in a query, it is normalized and its sub elements are extracted as shown above. Using the reformulation dictionaries, variant versions of the compound are generated (monolingual, then cross-lingual versions) and at-

[10] Lindén and Piitulainen (2004) propose a method for extracting monolingual synonym lists from bilingual resources.

tested variants are retained as synonyms to the original compound[11] (Besançon *et al*., 2003). To integrate the Arabic version into our CLIR system, no modifications were necessary beyond acquiring and formatting the cross language reformulation dictionaries.

The final NLP step (3c) involving in our CLIR system involves ranking relevant documents. Contrary to a bag of word system, which uses only term frequency in queries and documents, our system (Besançon *et al*., 2003) returns documents in ranked weighted classes[12] whose weightings involve the presence of named entities, the completeness of the syntactic subgraphs matched, and the database frequencies of the words and subgraphs matched.

Example

An online version of our cross language retrieval system involving our Arabic processing is visible online at a third party site: http://alma.oieau.fr. This base contains 50 non-parallel documents about sustainable development for each of the following languages: English, Spanish, French and Arabic. The user can enter a query in natural language and specify the language to be used. In the example of the Figure 1, the user entered the query "إدارة موارد المياه" and selected Arabic as the language of the query.

Relevant documents are grouped into classes characterized by the same set of concepts (i.e., reformulated subgraphs) as the query contains. Figure 2 shows some classes corresponding to the query " إدارة موارد المياه". The query term إدارة_موارد_مياه is a term composed of three words: مياه, موارد and إدارة. This compounds, its derived variants and their sub elements are reformulated into English, French, and Spanish and submitted to indexed versions of documents in each of these languages (as well as against Arabic documents). The highest ranking classes (as seen in Figure 2 for this example) match the following elements:

Class	Query terms	Number of retrieved documents
1	إدارة_موارد_مياه	14
2	إدارة_موارد، موارد_مياه	18
3	مياه، إدارة_موارد	9

Terms of the query or the expansion of these terms which are found in the retrieved documents are highlighted as illustrated in Figures 2 and 3.

5 Conclusion

We have presented here an overview of our natural language processing system and its use in a CLIR setting. This article describes the changes that we had to implement to extend this system, which was initially implemented for treating European languages to the Semitic language, Arabic. Every new language possesses new problems for NLP systems, but treating a language from a new language family can severely test the original design. We found that the major problems we encountered in dealing with a language from the Semitic language family involved the problems of dealing with partially voweled or unvoweled text (two different problems), and of dealing with clitics. To treat the problem of clitics, we introduced two new lexicons and added an additional clitic stemming step at an appropriate place in our morphological analysis. For treating the problem of vowelization, we simply used existing methods for dealing with unaccented text, but this solution is not totally satisfactory for two reasons: we do not adequately exploit partially voweled text, and our data structures are not efficient for associating many different lemma (differing only in vowelization) with a single surface form. We are currently working on both these aspects in order to improve our treatment of Arabic. But the changes, that we describe here, involved in adding Arabic were not very extensive, and we able to integrate Arabic language treatment into a cross language information retrieval platform using one man-year of work after having created the lexicon and training corpus. A version of our CLIR is available online and illustrated in this article. We plan to more fully evaluate the performance of the CLIR system using the TREC 2001 and TREC 2002 in the coming year.

[11] This technique will only work with translations which have at least one subelement that is has a parallel between languages, but this is often the case for technical terms.

[12] This return to a mixed Boolean approach is found in current research on Question Answering systems (Tellex *et al.*, 2003). Our CLIR system resembles such systems, which return the passage in which the answer is found, since we highlight the most significant passages of each retrieved document.

References

Steven Abney. Parsing by Chunks. 1991. In R. C. Berwick, S. P. Abney, and C. Tenny, editors, *Principle-Based Parsing: Computation and Psycholinguistics*, Kluwer Academic Publishers, Boston.

Saleem Abuleil, Martha Evens. 2004. Named Entity Recognition and Classification for Text in Arabic. *IASSE 2004*, pp. 89-94

Mohamed Attia. 1999. A large-Scale Computational Processor of Arabic Morpholofy, and Applications. M.S. thesis in Computer Engineering, Cairo University, pp. 28-32.

Y. Al-Onaizan and K. Knight. 2002. Machine Transliteration of Names in Arabic Text. *Proc. of ACL Workshop on Computational Approaches to Semitic Languages*, pp. 400-408

Kenneth Beesley. 1996. Arabic Finite-State Morphological Analysis and Generation. *Proc. of COLING-96*, pp. 89-94.

Romaric Besançon, Gaël de Chalendar, Olivier Ferret, Christian Fluhr, Olivier Mesnard, and Hubert Naets. 2003. Concept-Based Searching and Merging for Multilingual Information Retrieval: First Experiments at CLEF 2003. *CLEF-2003*, pp. 174-184.

K. Darwish. 2002. Building a Shallow Arabic Morphological Analyzer in One Day. In *Proc. ACL-02*, pp. 47-54

Fathi Debili and Lotfi Zouari. 1985. Analyse morphologique de l'arabe écrit voyellé ou non fondée sur la construction automatique d'un dictionnaire arabe, *Cognitiva*, Paris.

Leah S. Larkey, Lisa Ballesteros, Margaret E. Connell. 2002. Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. *Proc. of SIGIR 2002*, pp. 275-282

Krister Lindén and Jussi Piitulainen. 2004. Discovering Synonyms and Other Related Words. *CompuTerm 2004*, Geneva, Switzerland, August 29.

John Maloney and Michael Niv. 1998. TAGARAB: A Fast, Accurate Arabic Name Recogniser Using High Precision Morphological Analysis. *Proc. of the Workshop on Computational Approaches to Semitic Languages*. Montreal, Canada. August.

Elain Marsh. 1984. A Computational Analysis of Complex Noun Phrases in Navy Messages. In *Proc. of COLING '84*, Stanford, pp. 505-508.

Diana Maynard, Valentin Tablan, Kalina Bontcheva, Hamish Cunningham. 2003. Rapid Customization of an Information Extraction System for a Surprise Language. *ACM Trans. Asian Lang. Inf. Process.* 2(3) pp. 295-300.

Stefanie Tellex, Boris Katz, Jimmy Lin, Gregory Marton, and Aaron Fernandes. 2003. Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering. *Proc. Of SIGIR 2003*, pp. 41-47

Figure 1: User interface for querying the database. The user can choose between English, French, Spanish and Arabic as input language. For best results, the query should be syntactically correct and not in telegraphic form.

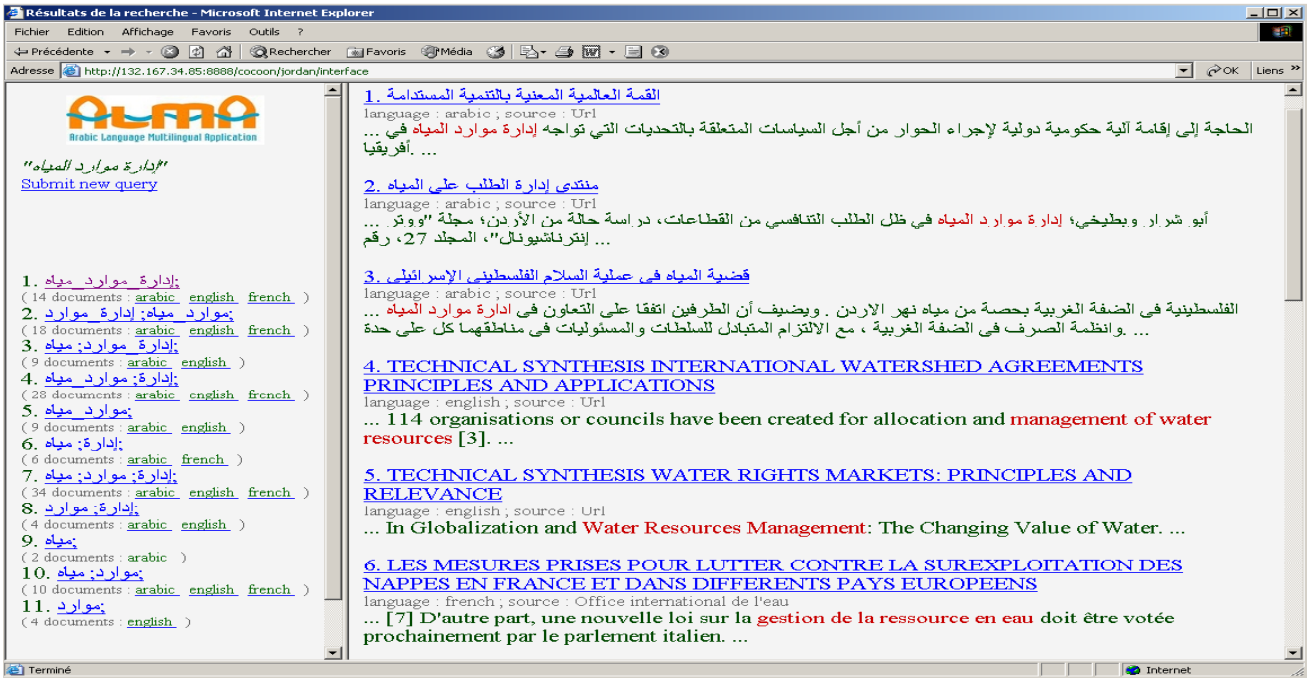

Figure 2: Search results user interface. Results can appear in many languages.

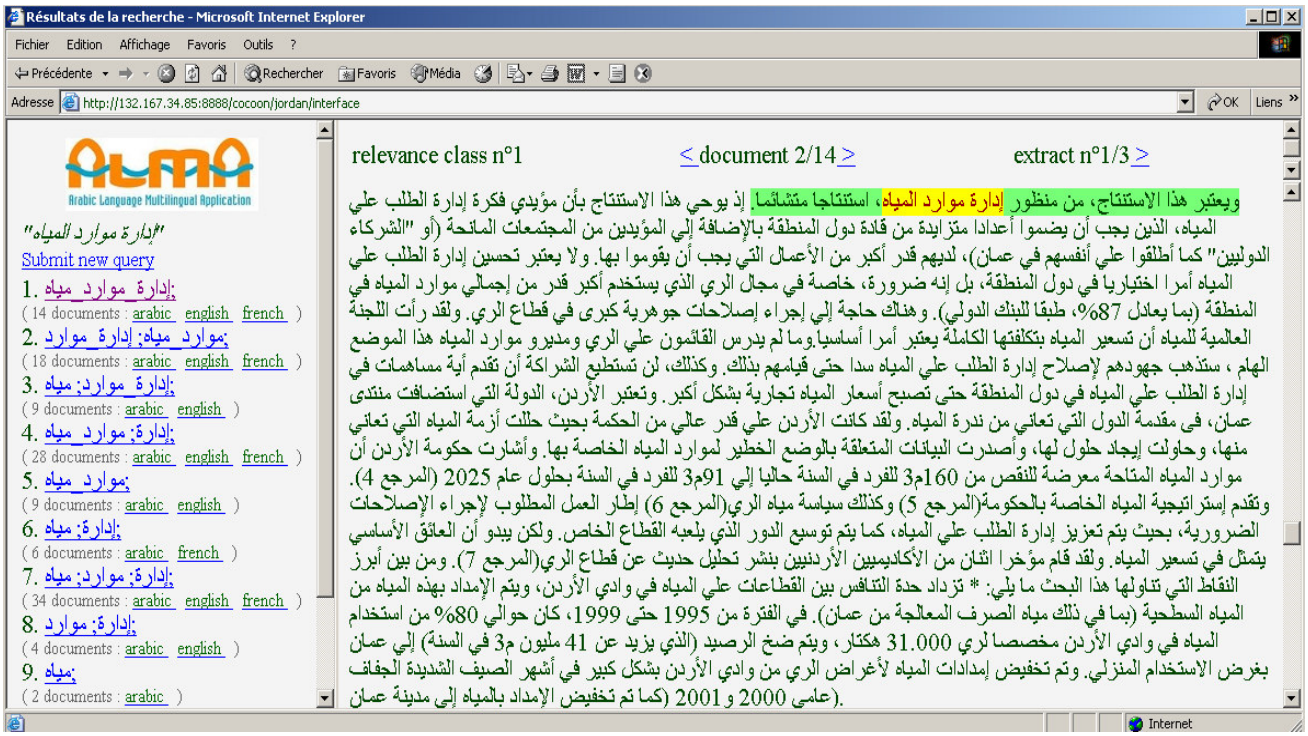

Figure 3: Highlighting query terms in retrieved documents.