2010 Fifth International Conference on Internet and Web Applications and Services

An Improved Structured and Progressive Electronic Dictionary for the Arabic Language: iSPEDAL

Mohammad HAJJAR
Institute University of Technology
Lebanese University
Lebanon
m_hajjar@ul.edu.lb

Abd El Salam AL HAJJAR, Khaldoun ZREIK
Paragraph Laboratory
University of Paris 8- Vincennes- Saint-Denis, France
abdsalamhajjar@hotmail.com, zreik@univ-paris8.fr

Patrick GALLINARI
Laboratory for informatics of Paris 6
Pierre and Marie Curie University
Paris, France
patrick.gallinari@lip6.fr

***Abstract*—In this article, we propose an improved structured and progressive electronic dictionary for the Arabic language (iSPEDAL) which can be presented in the form of a relational database or in the form of an XML document which can be easily exploitable using suitable query languages. Indeed, many Arabic dictionaries are found but are not structured and not directly exploitable since they are in flat textual files form. iSPEDAL doesn't contain any duplicated data (roots, prefixes, suffixes, the infixes, the patterns and the derived words). Moreover, for a given word, it provides links to its root, to their associated affixes, and to its patterns. iSPEDAL is supplied automatically from one or several traditional textual dictionaries and is enriched permanently with any Arabic textual corpus using system that we built. This system is composed of a Parser, a Selector, a Classifier, an Extractor, a Comparator, an Analyzer, and a Validator. The Parser allows the transformation of a textual source (dictionary or textual corpus) into a set of words. The Selector determines if a word is new or already exists in iSPEDAL. The Classifier allows to classify a given word and to add it to iSPEDAL as a root or as a derived word. The Extractor uses the Arabic extraction method to deduce the root of all words arriving to this component without their root or any indication about their root. The Comparator permits to avoid duplication of roots, affixes or patterns in iSPEDAL. The Analyzer allows the extraction of the affixes and the pattern from a derived word and of its root. The Validator can validate the information (word, root, patterns, and affixes) before adding to iSPEDAL database. This dictionary can be used to evaluate the information extraction methods from an Arabic document, given that; the vocabulary of the Arabic language is essentially built from the roots.**

Arabic Language, Corpus, Dictionary, Information Extraction, Root.

I. INTRODUCTION

The performance of information retrieval systems in Arabic still very problematic for several reasons [1], [3], [6]. One of the main reasons is that the Arabic language vocabulary is basically constructed from the roots. Indeed, the Arabic language has about seven thousands separate roots. An Arabic word is derived from its root by adding prefix, infix, or suffix based on one pattern [2], [3], [8]. The information extraction methods from the Arabic document proceed in a reverse order by extracting the root from the word. In this regard, several methods have been proposed [1], [4], [6], [8], [11], [14], [15], [17], [21], [23], [24]. These methods are either based on the morphological characteristics of the Arabic language or on statistical approaches. To evaluate these methods, we developed an evaluation system and we built a structured corpus limited to twenty roots and two thousand words [25]. In this structured corpus, each word is associated to its root. To validate these results, these methods must be evaluated on a larger similar corpus, such as a structured dictionary.

In this article, we propose an improved version of the structured and progressive electronic dictionary for the Arabic language SPEDAL [29] the iSPEDAL. This improved dictionary is also a relational database [18], [19], [22] which contains the roots, the affixes, the patterns, and the words with their links.

In the next section, we present the state of the art; Section 3 presents the architecture of the system with their components and the results. Section 4 describes the conclusion and the future work.

II. STATE OF THE ART

The actual available dictionaries are not directly exploitable and not structured. First, they are not exploitable because they are in flat textual files form [26]. Second, the structure of a conventional dictionary is different from the needed structure where the key is the root and the derived words are attached to this key. Yet, the key in a conventional dictionary is also the root, but is attached to this root, its mean, the derived words and their means. In that case, these added means are not attached to this key, but to another one, and by this, they induce a noises data. For these reasons, we previously proposed the structured and progressive electronic dictionary for Arabic language SPEDAL [29]. This dictionary is presented in the form of an easily exploitable relational database using an appropriate query language. It contains the roots, the prefixes, the suffixes, the infixes, and the patterns, as well as information provided by a standard dictionary [2]. In addition, for a given word, it provides links with its root, associated affixes, and its pattern.

After the implementation and evaluation of the SPEDAL dictionary [29], we observe that the data in the dictionary need cleaning and filtration, to avoid the noise, the conflict, and the duplication. The noise in the data is due to the entrance of some wrong words to the SPEDAL database, where these words do not exist in the Arabic vocabulary, and to create incorrect relations between some words and roots, that deduced affixes and patterns not existing in the Arabic grammar. The conflict appears when we observe some words that are entry to SPEDAL as root, but in reality it is not root, it is a word derived from another root, and vice versa. The duplication demonstrates when we try to find a word in the SPEDAL, in some case, we detect that it is existed more than one in the database, in each one, it has a different root, affixes, and patterns To resolve these problems in the previous dictionary, we propose this improved system that enriches and updates this dictionary [29], and divide the database into two parts, the first one contains the cleaned and confirmed data, and the other one contains the pending data. To achieve this goal, we ameliorate the architecture of the old system by modifying certain components and by adding new components as Selector, Extractor, and Validator. The Selector determines if a word is new or already existed in

978-0-7695-4022-1/10 $26.00 © 2010 IEEE
DOI 10.1109/ICIW.2010.80

iSPEDAL, it tests if a word already exists in the database, before the analysis. The Extractor takes all words which not having a root, and determines the root of these words, it uses the Arabic extraction method to deduce the root of these words. The Validator can validate the information (word, root, patterns, and affixes), based on the standard Arabic resources (set of Arabic patterns, prefixes, infixes, and suffixes) [25], [28]. If this information is valid, the Validator adds it to iSPEDAL database. that If this information is not valid, (does not exist in the Arabic grammar), the Validator considers the related word as pending and needs special review to validate this information and inserts the word into the pending part in iSPEDAL database.

III. ARCHITECTURE

Figure 1 shows the architecture of our new system for supporting and enriching iSPEDAL automatically from multiple textual classic dictionaries and Arabic text. This system is composed of several modules that are the Parser, the Selector, the Classifier, the Extractor, the Comparator, the Analyzer, and the Validator. The Parser allows converting a source text (text corpus or dictionary) of a set of words. The Selector determines if a word is new or already exists in iSPEDAL. The Classifier can classify a given word, and present it as a root or as a derivative word. The Extractor uses the Arabic extraction method to deduce the root of all words arriving to this component without their root. The Comparator allows avoiding duplication at all levels in iSPEDAL. The Analyzer can extract the affixes and the pattern from a derived word and its root. The Validator can validate the information (word, root, patterns, and affixes) before adding to iSPEDAL.

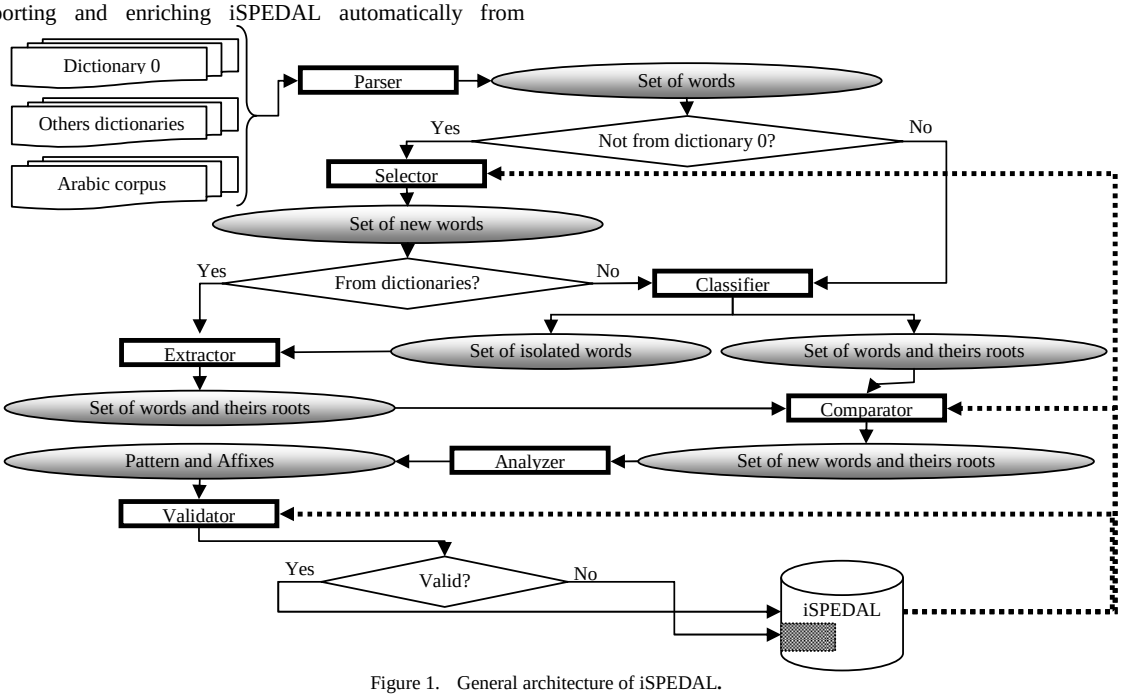

Figure 1. General architecture of iSPEDAL.

A. The Parser

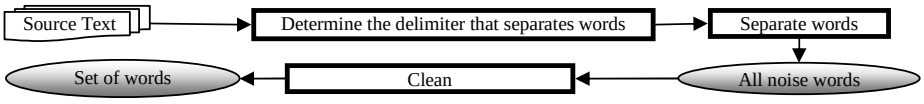

Figure 2. The Parser.

The Parser is the entry point of our system. The objective of this component is to transform a text document into a set of words. It well proceeds on dictionary in flat text form or any corpus text. Parsing a source text is done in several stages. The first is to determine the delimiters between words that can be spaces, symbols or other according to the input source. The second step is to provide a first set of words from the document source. The last step in this module is used to clean the obtained words. To do this, several steps are needed. The first is to eliminate the non-characters, numbers and symbols from each word, the second is used to remove the stop words, the last is used to extract the short words (إلى, من,...) to be added to iSPEDAL. So, the output of the Parser is a set of words which may be the roots or the derived words.

The output of Parser is passed into conditional phase to decide if the Parser inputs are from initial dictionary (dictionary 0), from other dictionary, or from any Arabic corpus. If the Parser inputs are from dictionary 0, the Parser output presents an input to the Selector, otherwise its presents an input to Classifier.

B. The Selector

Our target does not duplicate the data in the iSPEDAL dictionary, for that, we need to test any inputs if they are already existing in our database, before to analyze and insert it. The Selector plays this role, it takes the set of word out from the Parser, in condition that the data arrived from the source, other than the dictionary 0 (other dictionaries or Arabic corpus). The Selector decides if the word exists in

iSPEDAL or not, by access into database of the iSPEDAL. If the word exists, the Selector ignores it, to make a set of new word not existing in the iSPEDAL. The set of new output words from the Selector phase passes into conditional phase, to determine if the arriving inputs from the dictionaries or from Arabic corpus, In the case of a dictionary, this set passes directly to the Classifier, in other case, they passes to Extractor.

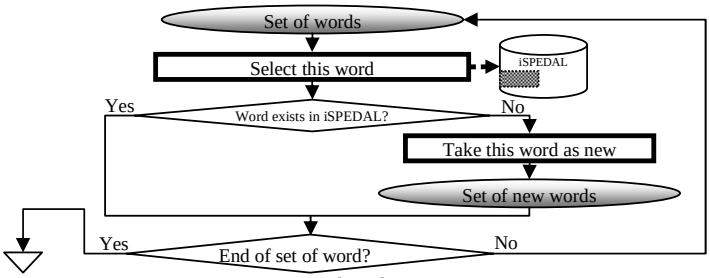

Figure 3. The Selector.

C. The Classifier

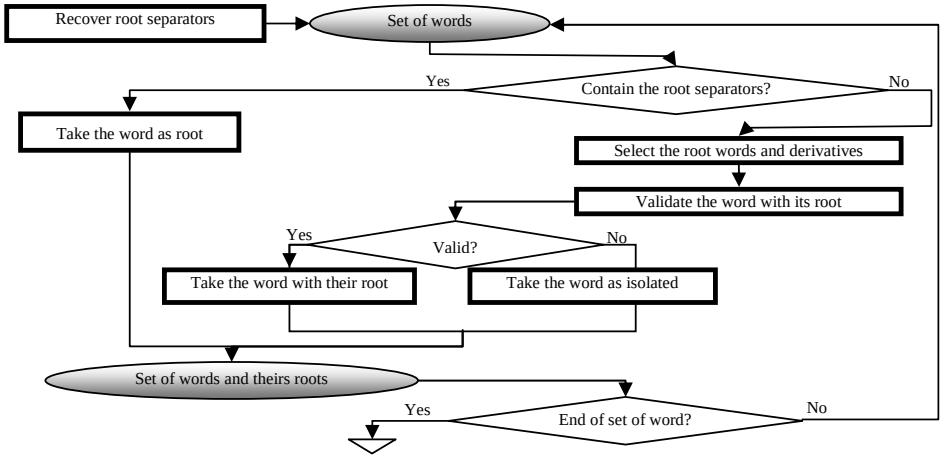

Figure 4. The Classifier

The inputs of this component can be provided either by the Parser (if the data source is dictionairy0) or by the Selector (if the data source is other dictionaries). The objective of this component is to classify a word, and decide if it is a root or not. Three classes are possible: roots, words derived from a root, and isolated word. Therefore, if it is root, it adds to the set of word, where the word and the root are the same. Otherwise, it determines their root, and added to the set of derived words and establishes the link with this root. If it does not have a root, it is isolated, and in this case no link was established with the roots.

The key issue to be addressed in this component is: how to determine if a word is a root? To answer this question, several cases are possible (Figure 4).

In the case of a dictionary, the roots are supported by special separators and the words, which are located after the root and before the next root, are derived from the first. But, must validate a word with the proposal root, because we can in the dictionary, some words existing after a root and not derived from it. This type of words are not considered in iSPEDAL. For example, in the case of Arabic dictionary Al Lesan [26], [27], each root is preceded by the symbol "@" and followed by the symbol ":" (Figure 4), the words that are after the root and before the next root its derived from the first. In this example, all words that are located between the two roots أكل and غرب are derived from the root أكل. To validate if all the words between two root are derived from the first must compare the characters of the word with the characters of the probable root, because in Arabic language the word deriving from a root, must contain all the root characters. In this case, the word تقول, which is not validated by comparison with the root أكل, is considered in iSPEDAL as isolated word.

D. The Extractor

The role of this component is very important, because it takes as input all words, not having a root, and it determines the root of these words, these words are arrived from Arabic corpus, the corpus is a journal, articles… that gives a set of word without an indication of the roots for these words, or these words are arrived from dictionary but the Classifier cannot determine the root for each word.

The origin of the Extractor is an Arabic root extraction method root of an Arabic word [24]. To use a method must choose a method has a high level of performance. Based on previous evaluation study of Arabic root extraction methods [25], we found that the method named "Stemming Arabic without a root dictionary", have high performance in extraction of the Arabic root.

The method "Stemming Arabic without a root dictionary", is based on the elimination of several sets of diacritics and affixes and the application of several patterns that have already defined [8].

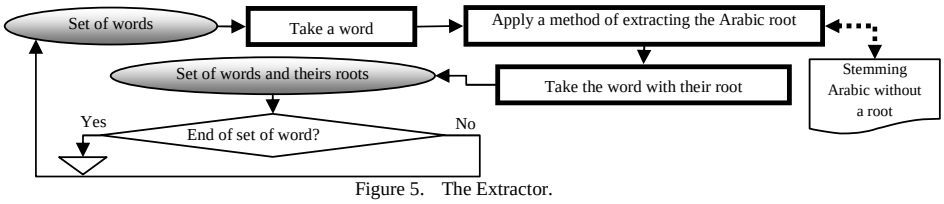

Figure 5. The Extractor.

E. The Comparators

The Comparator allows avoiding duplication in iSPEDAL at the words, roots, prefixes, infixes, suffixes, and patterns. This feature is only applied in the process of enrichment, to really enrich our dictionary and not add duplicates at all levels. Thus, the role of the Comparator is to filter the words before adding them to iSPEDAL. The Comparator receives as input a set of word pair (word, root) provided by the Classifier or by the Extractor, for each couple it will check if the word exists in iSPEDAL, when yes, it checks if the word exist under the same root, and also checks the existing of the root in iSPEDAL, if the word not exist in iSPEDAL, the Comparator will takes this word and their root as new couple [2], [3]. This new couple went to the Analyzer to extract the affixes and the pattern.

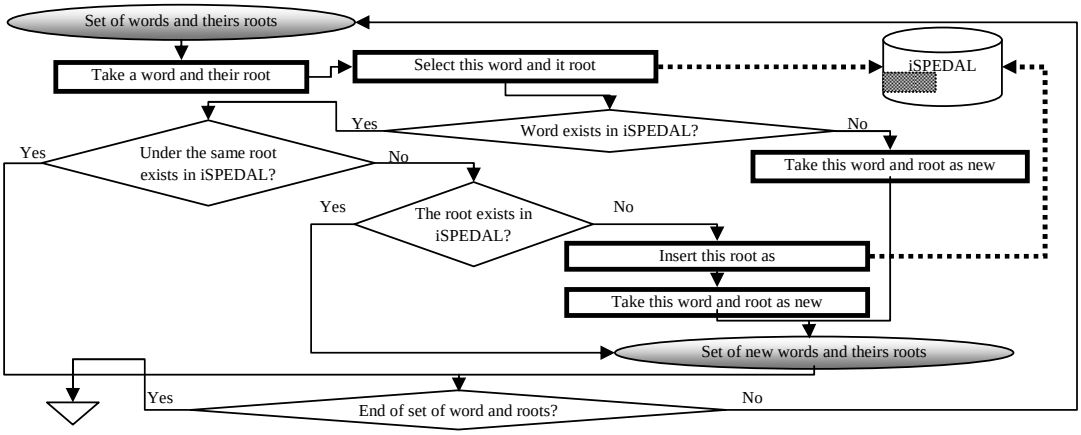

Figure 6. The Comparator.

F. The Analyzer

In general, an Arabic word is derived from its root by adding affixes (prefix, infix, or suffix) based on specific pattern. The Analyzer takes as input a couple (derived word, Root) and produces as output prefixes, suffixes, infixes, and the pattern according to the derived word. To do this, we begin by identifying the positions of the root letters in the derived word. The next step is to determine the letters that are in the derived word and not part of the root. Thus, the letters that precede the first letter of the root, if they exist, in the derived word are the prefixes. Similarly, the letters following the last letter of the root, if they exist, in the derived word are suffixes. In later letters, which are located between the first and last letter of the root, if they exist in the derived word and do not form part of the root are the infixes. The next step is to deduce the pattern in the derived word; the pattern is inferred by the positions of the root letters in the derived word. The first step permit to remove the suffixes, the second permit to remove the prefixes if they do not belong to the set { س, ا, م, ت}, the third step is to transform the letters after the prefixes {س, ا, م, ت} if they exist in the root of the order which permutes the first letter with"ف", the second with "ع" and the third with "ل". The infixes are reproduced as they are.

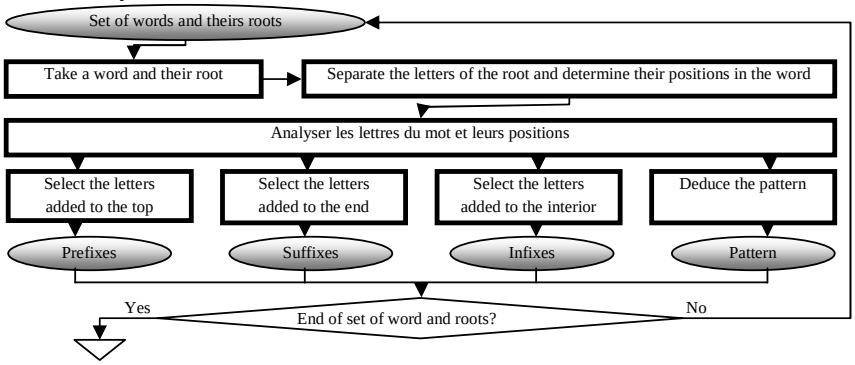

Figure 7. The Analyzer.

G. The Validator

The Validator presents a filter to avoid the noise in the iSPEDAL database, based on the standard Arabic resources (set of Arabic patterns, prefixes, infixes, and suffixes) [25], [28], that used in Arabic grammar. The Validator takes the Analyzer output (patterns, prefixes, infixes, and suffixes), and begin to test if these elements exit in the Arabic resources. It starts with the patterns, if the Analyzer gives a pattern that does not exist in the Arabic grammar; the Validator considers the related word pending and needs special review to validate the specific root. But if the pattern is correct, the Validator must also view the others elements (prefixes, infixes, and suffixes) in the same methodology. If the Validator attacks an affix that does not exist in Arabic grammar, it also considers the related word pending, and inserts the word into the pending part in iSPEDAL database. If the Validator knows that all the Analyzer outputs (patterns, prefixes, infixes, and suffixes) are valid, i.e. exist in the Arabic grammar, it inserts the related word and its root into the iSPEDAL database, with flag (mark) confirms that this data is correct.

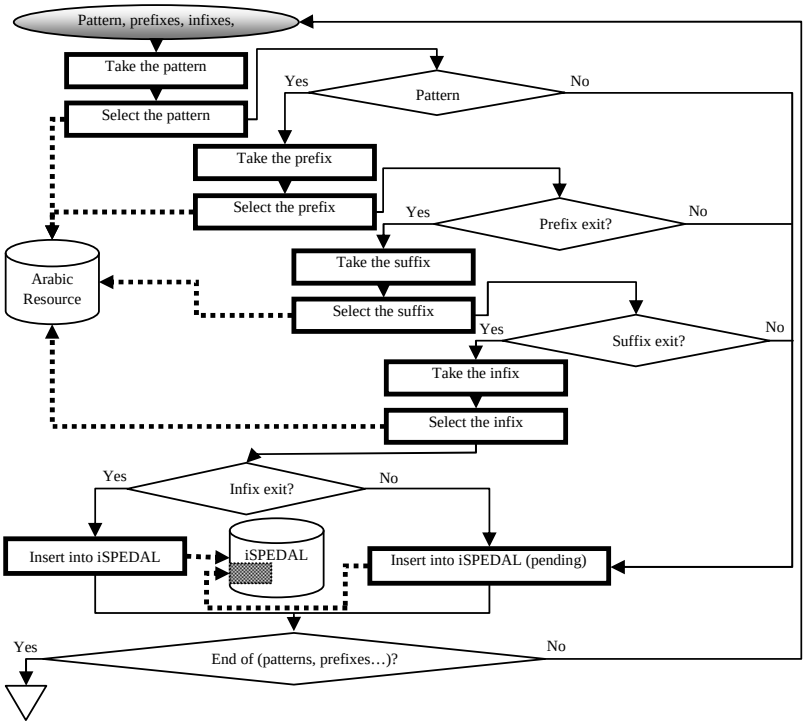

Figure 8. The Validator.

H. The Results

The main result of this work is an improved structured and progressive electronic dictionary for Arabic language (iSPEDAL). iSPEDAL mainly contains roots, prefixes, suffixes, infixes, patterns, and derived words. In addition, for a given word, it provides links to its roots, with associated affixes, and with its possible pattern.

A second result is a system which allows feed iSPEDAL automatically from one or more textual dictionaries and is continuously enriched it from any Arabic corpus. The components of this system are: a Parser, a Selector, a Classifier, an Extractor, a Comparator, an Analyzer, and a Validator. The Parser allows converting a source text (text corpus or dictionary) of a set of words. The Selector determines if a word is new or already exists in iSPEDAL. The Classifier can classify a given word, and present it as a root or as a derived word. The Extractor uses the Arabic extraction method to deduce the root of all words that are arrived to this component without their root or any indication about their root. The Comparator avoids duplication at all levels in iSPEDAL. The Analyzer can extract the affixes and the pattern from a derived word according to their root. And the Validator can validate the information (word, root, patterns, and affixes) before adding to iSPEDAL.

IV. CONCLUSION AND PRESPECTIVE

In this article, we have presented an improved version of the structured and progressive electronic dictionary for the Arabic language (iSPEDAL) [31]. This new dictionary may be presented in the form of a relational database or an XML document easily exploitable using the appropriate query languages. This new dictionary doesn't contain any duplicated data, where these data are roots, patterns, prefixes, suffixes, and the infixes, in addition the information provided by a classical dictionary. In addition, it provides links between a given word with its root, associated affixes, and its possible pattern [2], [3], [6].

Our new version of automatic system can supply and enrich iSPEDAL, with clean and useful Arabic data, from one or more dictionaries and classical Arabic corpus, to achieve a structured dictionary contain all the Arabic words used in the world, with theirs correct elements (root, pattern, and affixes).

This dictionary can be used to evaluate the methods for extracting the information from an Arabic document [25].

Then, this dictionary will help to improve the existing methods of extracting the information from Arabic document. The originality of our dictionary is that it is scalable, also contributing to the Arabic language evolution.

The next step is to establish to iSPEDAL a semantic dimension by adding semantic relations between the words. To create the semantic relationships between the words we need to exploit the characteristics of the conventional dictionaries. In general, a classical dictionary provides words with their synonyms. These synonyms may be words or roots. So we use the Classifier and the Analyzer to determine the semantic relations at the words and roots. Consequently, to determine the semantic relationships between two words we should go through their roots.

ACKNOWLEDGMENT

This work has been done as a part of the following projects: "Arabic Web Intelligence" supported by the Lebanese National Centre of Scientific Research (CNRSL), "Merit Program" supported by Islamic Development Bank (IDB), and "Recherche d'information multimédia multilingue Arabe" supported by the CEDRE franco-lebanese comity.

REFERENCES

[1] W. A. George and J. Boreham, The use of an association measure based on character structure to identify semantically related pairs of words and document titles, Information Storage and Retrieval, Vol. 10, 1974, pp. 253-260.

[2] Al Kharashi, A Web Search Engine for Indexing, Searching and Publishing Arabic Bibliographic Databases, 1999.

[3] A. Chen, A. Chen, and F. Gey, Building an Arabic stemmer for information retrieval, TREC-11 conference, 2002.

[4] K. Darwish, Building a Shallow Arabic Morphological Analyzer in One Day. The ACL-02 Workshop on Computational Approaches to Semitic Languages, Philadelphia, USA, 2002.

[5] L. S. Larkey, L. Ballesteros, and M. E. Connel, Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis, Proc. of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, 2002, pp. 275 – 282.

[6] H. Suleiman Mustafa, Character contiguity in N-gram based word matching: the case for Arabic text searching. Information Processing and Management.41 (4), 2004, pp. 819-827.

[7] Kanaan, R. Al-Shalabi, J. Jaarn, M. Al-Kabi, A. Hasnah, *A New Stemming Algorithm to Extract Quadri-Literal Arabic Roots*, 2004.

[8] K. Taghva, R. Elkoury, and J. Coombs, Arabic Stemming without a root dictionary, International Conference on Information Technology: Coding and Computing (ITCC'05) – Vol. I, 2005, pp. 152-157.

[9] Al Ameed, S. Al Ketbi, A. Al Kaabi, K. Al Shebli, N. Al Shamsi, N. Al Nuaimi, and S. Al Muhairi, Arabic Light Stemmer: A new Enhanced Approach , The Second International Conference on Innovations in Information Technology (IIT'05), 2005.

[10] L. Larkey, L. Ballesteros, and M. Connell, Light Stemming for Arabic IR, Arabic Computational Morphology: Knowledge-based and Empirical Methods, A. Soudi, A. Van Bosch, and G. Neumann Editors. Kluwer/Springer's series on Text, Speech, and Language Technology, 2005.

[11] F .Douzidia and G. Lapalme, Un système de résumé de textes en arabe, 2ème Congrès International sur l'Ingénierie de l'Arabe et l'Ingénierie de la langue, Alger, 2005.

[12] Y. Kadri and J. Nie, Effective Stemming for Arabic Information Retrieval, proceedings of the Challenge of Arabic for NLP/ MT Conference, Londres, Royaume-Uni, 2006.

[13] L. Khreisat, Arabic Text Classification Using N-gram Frequency Statistics A Comparative Study, The 2006 International Conference on Data Mining Part of the 2006 World Congress in Computer Sciences DMIN, 2006, pp. 78-82.

[14] F. Ahmed and A. Nürnberger, N-grams Conflation Approach for Arabic, ACM SIGIR Conference, Amsterdam, 2007.

[15] M. El-Halees, Arabic Text Classification Using Maximum Entropy, The Islamic University Journal (Series of Natural Studies and Engineering) Vol. 15, No.1, 2007, pp. 157-167.

[16] Khemakhem, B. Gargouri, A. Abdelwahed, G. Francopoulo, Modélisation des paradigmes de flexion des verbes arabes selon la norme LMF - ISO 24613, Traitement Automatique des Langues Naturelles, Toulouse, France, 2007.

[17] M. Ben Abderrahmen, B. Gargouri, M Jmaiel, LMF-QL: A graphical Tool to Query LMF databases, Third Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, Poznań, Poland, 2007.

[18] E. Norbert, Arabic Language Support in SQL Server, Microsoft corporation, SQL Server Technical Article, http://msdn.microsoft.com/en-us/library/cc295829 (SQL.90).aspx, 2008.

[19] C-A. Comes, L-D. Savu, I-O Spatacean, B. Stefan, and A. Avram, Universal Symbolic Translator for Procedural Language over SQL, 7th WSEAS Int. Conf. on Applied Computer & Applied Computational Science (ACACOS '08), Hangzhou, Chine, 2008.

[20] J. Micher, C.Voss, Buckwalter-based Lookup Tool as Language Resource for Arabic Language Learners Software Engineering, Testing, and Quality Assurance for Natural Language Processing, USA, 2008, pp. 66–67.

[21] M. Sinane, M. Rammal, and K. Zreik, Arabic documents classification using N-gram, Conférence ICHSL6, Toulouse, 2008.

[22] F. Baccar, A. Khemakhem, B. Gargouri, K. Haddar, and A. Ben Hamadou, Modélisation normalisée LMF des dictionnaires électroniques éditoriaux de l'arabe, TALN 2008, Avignon, France, 2008.

[23] Francopoulo, M. George, Language resource management − Lexical markup framework (LMF), ISO/TC 37/SC 4 Rev.15, 2008.

[24] Al Hajjar, M. Hajjar, K. Zreik, Classification of Arabic Information Extraction methods, 2nd International Conference on Arabic Language Resources and Tools, Le Caire, Egypte, 21-23 Avril 2009.

[25] A. Al Hajjar, M. Hajjar, and K. Zreik, A new system for evaluation of Arabic root extraction methods, The Fifth International Conference on Internet and Web Applications and Services. ICIW, Barcelona, Spain, 2010.

[26] Ibn Manzour, Lisan Al-Arab. www.muhaddith.org, 2009.

[27] Sakher, Lexicons: Lisan Al-Arab, Al Qamous Al Mouhit, Al Wasit, Al Mouhit, Mouhit Al Mouhit, Al Ghani, Taj Al Arous, Najaat Al Raed, http://lexicons.sakhr.com, 2009.

[28] Khoja, S., AND Garside, R. Stemming Arabic text. Computing Department, Lancaster University, Lancaster, www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps. 1999.

[29] A. Al Hajjar, M. Hajjar, K. Zreik, Un nouveau dictionnaire électronique structuré et évolutif pour la langue arabe, In Patrimoine 3.0 : Actes du douzième colloque international sur le document électronique (CIDE.12). Ed. Europia, Paris, 2009.