Towards an Arabic Web-based Information Retrieval System (ARABIRS): Stemming to Indexing

El Younoussi Yacine
National School of Applied Sciences Tetouan- University of Abdelmalek Essaadi
M'hannech 2, PO Box 2222, Tetouan, Morocco.

ABSTRACT
Arabic, the mother tongue of over 300 million people around the world, is known as one of the most difficult languages in Automatic Natural Language processing (NLP) in general and information retrieval in particular. Hence, Arabic cannot trust any web information retrieval system as reliable and relevant as Google search engine.

In this context, we dared to focus all our researches to implement an Arabic web-based information retrieval system entitled ARABIRS (ARABic Information Retrieval System). Therefore, to launch such a process so long and hard, we will start with Indexing as one of the crucial steps of the system.

General Terms
Natural Languages Processing , Arabic Language Processing, Information Retrieval.

Keywords
Arabic Information Retrieval System, Arabic Language Indexing, Arabic Language Stemming.

1. INTRODUCTION
All the issues raised about information retrieval (IR) are looking for an answer to the famous problem of relevance. In other words, all the Information Retrieval systems (IRS) would like to find out the best and effective way to retrieve the relevant and only the relevant information to respond their user's queries.

As part of the web retrieval information, the problem of relevance is further complicated because, on the one hand, the web content exponential increase, and secondly the nature of certain languages like Arabic recognized by its richness and morphological complexity.

Text documents in natural language in general and Arabic in particular, require a special processing and analysis to change an unstructured text representation (natural) to a structured one, usually called Index, which contains the most significant terms (called descriptors) in the collection of documents.

The user express usually his request as a set of natural language key words. Most of Information Retrieval Systems match documents and queries by establishing a link between index terms and the keywords of user query, hence the problem of relevance. Indeed, users ignore completely the index content; therefore, they formulate queries whose their contents are inconsistent with the terms of index. For example, if there is the keyword كاتب (writer or author) of a user query, the IRS must recover all the documents containing the morphological variants of this term such as كتب (he wrote or books or it was written), كتاب (book, writers or authors), etc. This problem should be solved by Stemming processing.

In fact, stemming is the word's roots or stems extraction process. In the case of Arabic Language, there is definitely, up to now, no reference stemmer that can apply this process so effectively than English or French language stemmer.

In addition, a document can possibly be relevant towards a user query even if they don't share the same terms. For example, a document about "UNIX", may be relevant for a user query on "Operating Systems" even if that expression wouldn't be in the document. Then, in order that the Information Retrieval System will be able to do this kind of matching, he must apply one of the queries reformulation (queries expansion) techniques, which add new terms to the initial user query.

In this paper we propose an approach for non-vowelized (without diacritics) Arabic language Stemming. Our aim is to increase performance of Arabic text documents indexing. It is a process beginning of setting up a new Information Retrieval System dedicated totally and especially to Arabic language entitled ARABIRS (ARABic Information Retrieval System).

2. INFORMATION RETRIEVAL
2.1 Information Retrieval Process
Following a user query, the goal of any Information Retrieval System is to find only the relevant results from a set of documents in the corpus. The information retrieval process is shown in Figure 1.

The information retrieval process typically involves three main steps:

- Indexing: it's applied to the documents and the user query.
- Document-Query Matching: it looks for relevant documents
- Query Expansion: it is applied in order to increasing the relevance rate of the IRS. It may add new terms to the original user query.

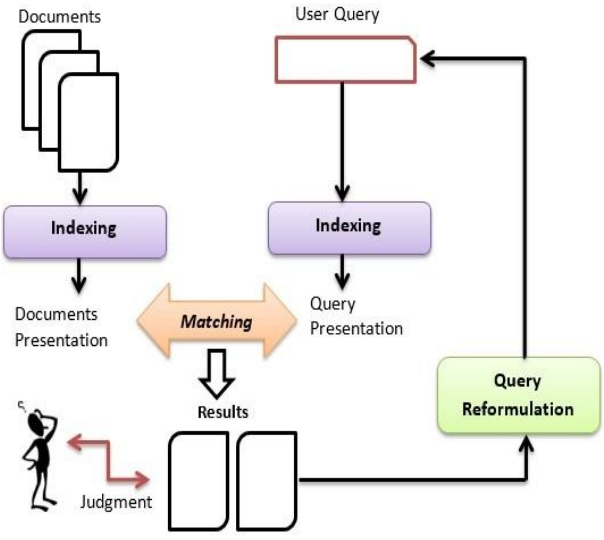

Fig 1: Information Retrieval Process

2.2 Indexing

Indexing is to find out discriminant terms (called concepts) in a document or a user query in order to pass from an unstructured text representation (in natural language) to a structured one usually called Index. Indexing is one of the very useful and crucial steps in an information retrieval process. Each Index descriptor must take a weight representing its degree of importance.

Document-Query Matching, in other words, relevance documents retrieving for a user query, is based on the index entries (descriptors). The construction step of this structure is among the very important steps in the information retrieval process.

Index descriptors have generally not the same importance, so to make a difference, weighting process is applied. This process assigns a weight to each term in the index.

In a document or a collection of documents, we certainly have many words that belong to the same morphological family, especially in the case of a high flexional language like Arabic. Indeed, the Arabic words كتاب (book), الكتاب (the book), الكتب (the books), الكاتب (the author), etc. belong all to the same morphological base, the root كتب, so we have to assign all of these words to the same entry in an index structure which could be the root كتب. The extraction roots process is called Stemming that can significantly reduce the size of the Index.

2.3 Stemming

Stemming is the process of reducing a given word to its basic form, which can be either a stem, or a root.

The *stemming* aims to bring together all the morphological variants of a given word around its stem or root, and then comes the importance of the stemming in the indexing process.

The Stemming is several decades older. At the beginning it was oriented to the English language, thus, Julie Beth Lovins has published the first stemming algorithm in 1968 [1]. Then, in July 1980, another algorithm has been published. It is the *Martin Porter* algorithme [2]. Both algorithms have marked English stemming domain. This success is due to the simplicity of these algorithms that refer to a suffix stripping approach, which removes word suffixes by applying some context rules that indicate the conditions under which a suffix will be deleted [1] [2].

The suffix stripping method cannot give good results in the case of Arabic language because of the richness and complexity of its morphological and syntactic properties [3] [4]. Arabic is known as one of the most difficult languages to control in the automatic Natural Language Processing (NLP).

Accordingly, there are three main approaches for Arabic stemming:

1. Light stemming: the light stemming is stripping prefixes and/or suffixes process in order to generate a word stem, without worrying about infixes or recognize word pattern to find the root. Several light stemming algorithms have been cited in the literature, such as Kareem Darwish [5], Aitao Chen [6] and Leah Larkey [7] algorithms.
2. Morphological Analysis : this approach remove all additive word letters (prefixes, suffixes, proclitics and enclitics) that have been added during the word generation process (flexion or derivation). In some cases, it will add a new letter or replace a letter by another one in order to find out the root or roots of a given word. There are some several approach of Arabic morphological analysis like Karim Darwish [5], shereen Khoja [8] and Tim Buckwalter [9] algorithms.
3. Statistic Stemming : Statistical tools can contribute to the development of an independent language Stemming approach. Indeed, similarity calculations may be useful to identify allomorphs of a given word. Such approaches usually involve n-gram technics. We can mention the Mayfield algorithm [10] based on 6-gram, which works for many languages including Arabic.

2.4 What Stemming for Indexing ?

We have seen in the 2.3 previous section that the different stemming techniques lead to either a stem or a root. Now the question arises is, what morphological basis should we choose for indexing, the root or stem?

In ARABIRS, our IRS, we have opted for the root to be the morphological basis of our stemming algorithm, because we have previously mentioned that one of the stemming advantages in the indexing process is to reduce the size of the index, so if we assume that the stemmer returns the stem as a result, then we will not be able to reach this goal. Take for example the following Arabic words list كتابات (writings), كاتبة (secretary) et يكتبون (they write), if we then apply a Stemming for all words, there will be two different cases:

- If Stemming algorithm returns the root, then the result will be the same for all the three words: the root كتب. Therefore, we will have one index entry for all these words.
- If Stemming algorithm returns the stem, then we will have three different results: كتاب for the word كتابات, كاتب for كاتبة and كتب for the word يكتبون, accordingly we will have three different index entries, each one corresponds to a different word.

Then, we can conclude that the stem cannot reduce the index size.

3. AN ARABIC WEB-BASED IRS: ARABIRS

3.1 Presentation

ARABIRS is a web Information Retrieval System dedicated to Arabic language. This system is thought to be the gate number 1 of all Arabs to access the Arabic content on the web.

You should know that during the past decade, the Arabic e-content on the web has recognized a dramatic increase. According to the latest results from Internet World Stats[1] (December 31, 2013), the Arabic language is ranked fourth among the most used languages on the Internet with a growth of 5296.6% (table 1).

According to these remarkable figures, we decided to set up an IRS, worthy of this language and its content.

Table 1. Top Ten Languages Used in the Web - December 31, 2013 (Number of Internet Users by Language)[2]

[1] Internet World Stats: www. internetworldstats.com

[2] Source: www. internetworldstats.com/ stats7.htm

Top Ten languages in the internet	Users by Language (in million)	Users Growth in Internet (2000 - 2013)	Population (in million)
English	800.6	468.8%	1370.977
Chinese	649.4	1910.3%	1392.320
Spanish	222.4	1507.4%	439.320
Arabic	135.6	5296.6%	367.465

3.2 General architecture of ARABIRS

ARABIRS is a modular system. It consists of two main modules:

- Analysis module (see Figure 2): it takes care of all the necessary treatments before the search for information. It consists of four sub-modules that take in charge:
  - Tokenization (Segmentation): tokenize the text into multiple tokens.
  - Preprocessing: apply some preprocessing operations on words (tokens) such as removing stop words.
  - Stemming : extract roots from words
  - Indexing: for creating ARABIRS Index.
- Information retrieval Module: this module is responsible for searching relevant documents towards the user query. It consists of two sub-modules responsible for:
  - Document-Query matching: the mapping between the relevant documents and the user query.
  - Queries Reformulation (or extension): reformulating the user query by adding new words in order to increase the relevance of the results.

3.3 Our Arabic Stemming Approach

Stemming is a decisive treatment during the indexing process. Indeed, after the tokenization, all the tokens will be passed to the stemming module to find their roots.

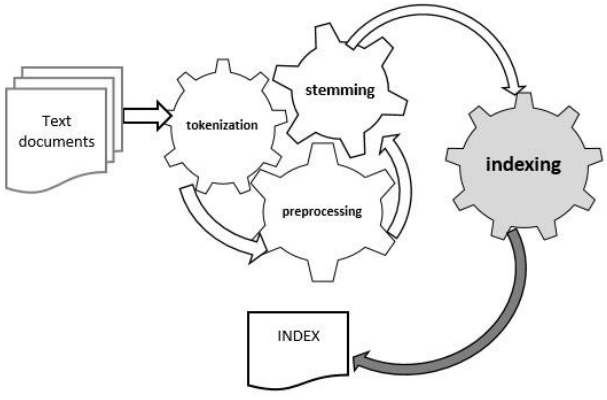

Fig 2: ARABIRS Indexing Process

In this work, we deal with non-vowelized texts written in modern standard Arabic language (MSA) taught in schools and is used by the media and in official speeches. Texts without diacritics are incredibly difficult in Natural Language Processing (NLP) and especially in Arabic Stemming. Actually, diacritics may wipe off grammatical and semantic ambiguity.

We take for example the non-vowelized word كتب, in this case we cannot decide if that is the verb كَتَبَ (write) or the plural name كُتُبٌ (books). Accordingly, we will suppose that an Arabic word may have several roots (see Table 2).

Table 2. Five possible roots for the word ايمان [5]

pronunciation	English translation	root
إيمان	faith	أمن
أيّمان	Two poor people	أيّم
أَيَمأَنُ	We will he give support	مأن
أَيْمانٌ	convenant	يمن
أَيَمأَنَّ	Will they (feminine) point to	يمأ

3.3.1 Stemming Process

We tried to develop a Stemming approach based on finite state automata. As in Figure 3, this approach is based on four axes: normalization, lexicon, light Stemming and morphological analysis.

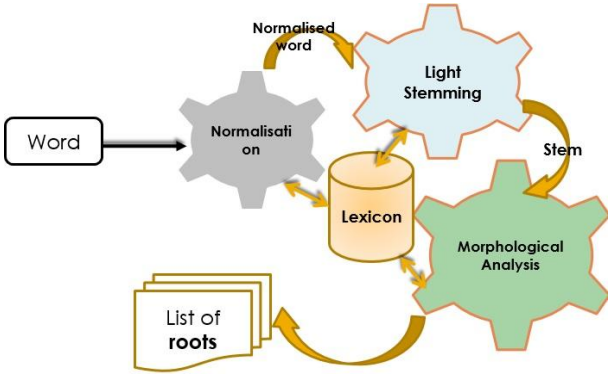

Fig 3: Our approach Stemming Process

- Normalization: Because the Arabic script may change from country to another, we conduct a series of graphical changes to standardize the Arabic text:
  - Remove the Arabic diacritics : ِ ٍ ً ٌ ّ ْ
  - Transform the initial letters أ, إ, آ and ٱ to ا.
  - Transform respectively the letters ى and ة to ي and ه
- Lexicon: the lexicon is crucial for the our stemming approach. Our lexicon consists of several dictionaries:
  - Roots dictionary: it contains about 10 000 Arabic roots extracted from the dictionary Lissano Arabic Al-Arab (لسان العرب)
  - Affixes dictionary: this approach assumes that a maximum Arabic word consists of a root, prefix, proclitic, suffix and enclitic. That is why we adopt the following affixes dictionaries: proclitics, prefixes, suffixes and enclitic.

    Note that proclitics can combine to each other to form compounds proclitics, this combination is managed by compatibility rules.

- Patterns dictionary: patterns in Arabic language is a kind of template that is applied to the roots to generate lexemes. Then, to extract a word root we have to know its corresponding pattern. The dictionary contains 16 Arabic language patterns such as فعال, فعول, فاعل, etc.
- Stop words dictionnary: it contains non-discriminative words during the document-query matching operation, such as: prepositions (في, على) quantifiers (كل, بعض) etc.

- Light Stemming: its goal is to remove some affixes to find the word stem. Our approach assumes that the length of the stem cannot be less than 5 letters. In addition, before removing an affix we need to verify its compatibility with other potential word affixes.
- Morphological analysis: after light stemming, we will have a stem that will be submitted to the morphological analysis module to extract all the possible roots.

  We've mentioned before, that we can find the root of an Arabic word, if we know its corresponding pattern. So, our approach is based on finite state automata that represent the Arabic patterns. We can find the root of a word if it happens to reach a final state of the automaton.

3.3.2 Evaluation

After we implemented this Arabic stemming approach, we have passed to the evaluation step.

For this purpose, we constructed a corpus of 50 Arabic text documents. All these documents have collected from an Arabic news website called "News Archive[3]" (أرشيف الأخبار) in which, subjects are of different types: sports, politics, economics, etc.

After segmentation step, we had 8860 tokens, including 580 stop words (it remains 8280 words). Then we have applied our stemming approach and have found the following results (see Figure 4):

- 71.14% of success.
- 28.86% of failure.

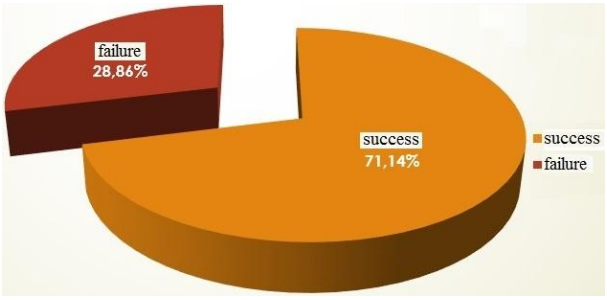

Fig 4: the evaluation results of our stemming approach applied on 8280 words.

Although the success and failure rates depend on the nature of the corpus texts, the results reflect a pretty good performance and effectiveness for the Arabic Stemming approach.

[3] http://www.newsarchiver.com/

4. TOWARDS A LEXICON BASED STEMMER

4.1 Processus Général

After evaluating our stemming approach presented in the previous section, we found that it shows some disadvantages, particularly in relation with the doubled and the so-called weak roots that contain one or more of the three letters و low (waw) أ (hamzah) ي (yaa) or have a double radical.

We say that an Arabic root is doubled where the two last radicals are identical. Example: مدد

The weak roots can be classified into four classes: *first weak*, *second weak* and *third weak*.

- First-weak roots (Assimilated): they have one of the two weak letters و or ي as their first radical root. Example: (وقف، يبس)
- Second-weak roots (Hollow): they have a weak letter in the middle of the root. Example: بيع, قول
- Third-weak roots (Defective): when the ending of the root is a weak letter. Example: رمي

In order to achieve a reliable and relevant Arabic stemming approach to our ARABIRS IRS we tried to automatically create a lexicon that could gather all inflections and derivations of the Arabic language.

Then, we propose a Stemming approach based on a lexicon of stems. The creation of this lexicon involves calculating **Levenshtein** similarity. Indeed, the Arabic dictionary Lissano Al-Arab (لسان العرب) contains about 10 000 roots and each one is defined by a text rich enough of stems. So, our idea comes from that here. Why not use these definitions to extract all the stems of any dictionary entry (root), which will be used later in an Arabic stemming processing.

Therefore, before applying the process of figure 3, we will look for the word supposed to stemmed, in the stems lexicon. If we find it, we will retrieve all the corresponding roots and then we will apply the process of figure 3 to find other possible roots.

4.2 Create a Stems Lexicon based on Levenshtein Similarity

The Levenshtein distance or edit distance is the minimum number of edit operations (insertions, deletions and substitutions) to change a given string into another. The Levenshtein distance between two strings S_1 and S_2 can be converted into a similarity measure [11] (between 0.0 and 1.0) using:

$$sim_{ld}(s_1, s_2) = 1 - \frac{dist_{ld}(s_1, s_2)}{\max(|s_1|, |s_2|)}$$

$dist_{ld}(|S_1|,|S_2|)$ is the actual Levenshtein distance which returns 0 if S_1 and S_2 are the same or a positive number of edits if they are different. |S| is the length of a string S.

This similarity measure is greater as the number of differences between the two strings is little.

Due to this Levenshtein similarity measure, we have created a lexicon containing all the roots of Lissano Al-Arab (لسان العرب) dictionary, which everyone gives access to a group of stems representing the derivations and flexions of this root.

All The steps of this approach are shown schematically in Figure 5.

After the text segmentation of the root definition, we treat word by word. If the word size is > 5, a light Stemming is applied to remove all possible affixes.

When the word size is ≤5, we calculate the Levenshtein similarity of the stem towards its potential root.

If the value returned is less than a threshold s, we assume that the number of differences between the two words is inacceptable. Otherwise, we add the stem to the lexicon.

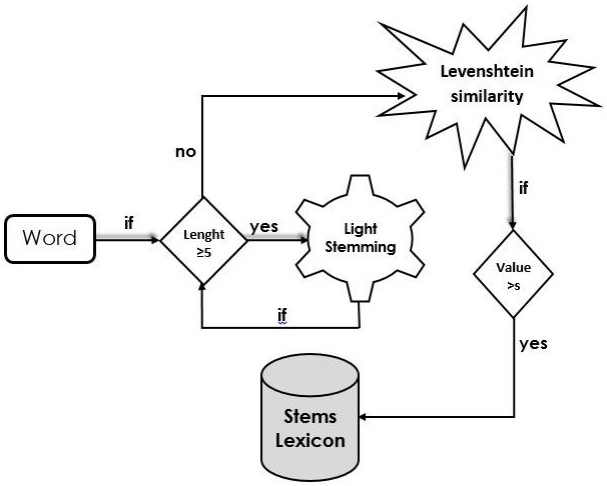

Fig 5: steps of stems lexicon creation

Knowing that the size of all Lissano-Al-Arab dictionary roots is between 3 and 4 radicals, then the stems of 5 characters, may undergo more than two edit operations to become a root, which can result in 0.6 similarity value . In some special cases, such as the doubled roots (مضعف), we can have up to 4 edit operations, which means 0.4 similarity value. Therefore, we set the S threshold value to 0.4.

Arabic has a right-to-left connected script that uses 28 letters, which change shape depending on their positions in words. Weak roots could have some orthographic features, such as the radical و (waw) of the root قول (say) which becomes ا (Alif) in the lexeme قال (he said) and will be removed from the lexeme قل (say).

Given these orthographic characteristics, we will adopt the following normalizations:

- All orthographic forms of the letter Hamza (أ, ئ, ء, ؤ) are considered similar when calculating the Levenshtein distance.
- When one of the two weak letters ي (Yaa) or و (waw) are found in the middle of the root, and the lexeme contains, in the middle, the weak letter ا (Alif), so in this case the letter ا is considered similar to the other two letters ي and و
- If we find the weak letter ي (Yaa) at the root ending, and the lexeme contains the terminal letter ى (Alif Maksoura), then the two letters are considered similar.

4.3 Test and Results:

To test our approach to building a stems lexicon based on light stemming and Levenshtein similarity measure, we manually built a test table that consists of 20 Arabic roots; including 10 normal roots and 10 weak roots.

After that, we have applied a manual stems extraction from each root definition. The number of stems is between 20 and 40 stems.

We then applied our automatic stems extraction approach on the definitions of all the test roots.

For both roots, regular and weak, we calculated the precision and recall values and then we have deduced the 11 points (from 0.0 to 1.0 with a 0.1 step) mean precision.

For regular roots, we had the following results:

Table 3. Regular roots Recall-Precision values

Recall	0,0	0,1	0,2	0,3	0,4	
Precision	1,00	0,95	0,92	0,90	0,87	
Recall	0,5	0,6	0,7	0,8	0,9	1,0
Precision	0,84	0,84	0,81	0,77	0,75	0,74

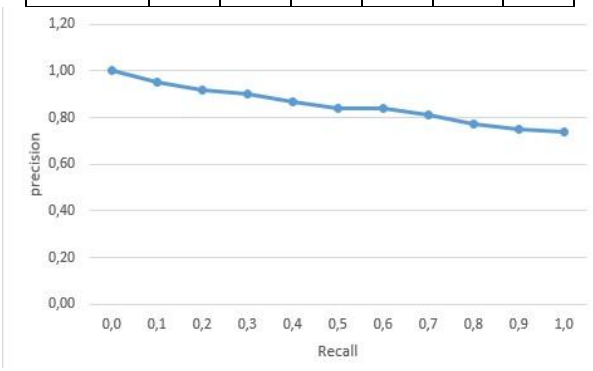

Fig 6: Regular roots recall-precision curve

For weak roots, we had the following results:

Table 4. Weak roots Recall-Precision values

Recall	0,0	0,1	0,2	0,3	0,4	
Precision	1,00	0,90	0,85	0,82	0,78	
Recall	0,5	0,6	0,7	0,8	0,9	1,0
Precision	0,77	0,74	0,73	0,71	0,67	0,64

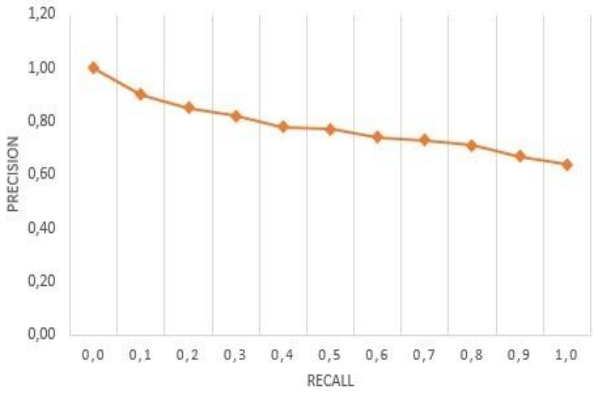

Fig 7: Weak roots recall-precision curve

To compare the two results, we have gathered the two curves, regular roots and weak roots, and we got the following curve:

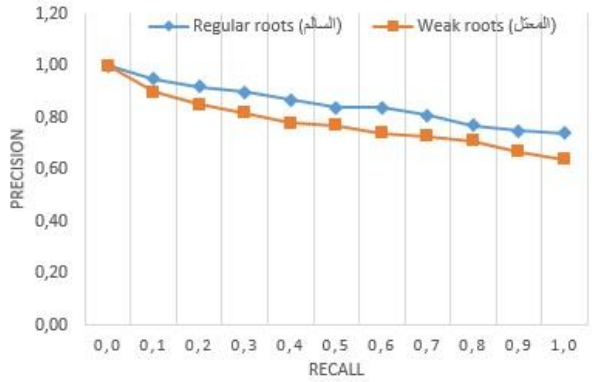

Fig 8: Recall-Precision curve comparison of Regular and Weak roots

It can be seen that our approach has shown a relative effectiveness towards the automatic extraction of stems from LISSANO AL ARAB (لسان العرب) Arabic dictionary.

In fact, we managed to achieve around 75% rate precision for regular roots, and 64% for weak roots.

We found that the failures is mainly due to the existence of some stems in the definition of the root. These stems share two or three radicals with the root, even if they are not part of its morphological variations. For example, after processing the definition of the root أسد, we found the lexeme إسم among its stems list. Indeed, the stem إسم share two radicals with de root أسد, therefore the Levenshtein similarity measure gives 0.66 value that is greater than 0.4 threshold value. That's how the lexeme إسم will be one of the stems list of the root أسد.

We have also noted that there is a relevance variety between regular and weak roots extraction. In fact, the regular roots extraction is more relevant than the weak roots. We have already mentioned that weak roots contain weak letters (و, ي and ا) that may change during the flexion or derivation process, they could even be deleted.

Take the example of وفي root, among its flexions:

- The stem يفي: the letter و was replaced by the letter ي;
- The stem وفت: the letter ي was replaced by ت;
- The stem ف: is the imperative form of وفي root, where the two letters و and ي were deleted.

5. CONCLUSION ET PERSPECTIVES

The stemming is a paramount processing and of great importance in an information retrieval process of Arabic language. In this paper, we have presented a non-vowelized Modern Stantard Arabic Language stemming approach based on light stemming and stems lexicon. The automatic construction of this lexicon is based on the Arabic dictionary LISSANO AL ARAB (لسان العرب) and Levenshtein similarity measure. This stems extraction approach showed a respectable relevance rate despite some failure cases due particularly to weak roots. We hope that our stemming approach helps to create a reduced and accurate Arabic index involving an increase of results relevance of our Arabic Information Retrieval System ARABIRS.

We are currently focusing on the comparison of the Levenshtein distance with other distance measurement in order to reach a high relevance rate for creating the lexicon of stems.

6. REFERENCES

[1] Julie Beth Lovins Ding, W. and Marchionini, G. 1997. A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[2] Porter, M.F. 1980. An algorithm for suffix stripping. *Program*, Vol. 14 No.3, pp. 130-137

[3] Aljlayl, M., and Frieder, O. 2002. On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach.

[4] Larkey, L. S., Ballesteros, L. and Connell M. 2002. Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis.

[5] Darwish, K. 2003. Probabilistic methods for searching OCRdegraded Arabic text. Unpublished Ph.D. thesis, University of Maryland,USA.

[6] Chen, A., and Gey, F. 2002. Building an Arabic stemmer for information retrieval. In TREC 2002. Gaithersburg: NIST, pp 631-639.

[7] Leah, L., Lisa, B. and Margaret E. C. 2007. Light stemming for Arabic information retrieval. Arabic Computational Morphology, Springer Netherlands,

[8] Khoja, S. 1999. Stemming Arabic Text.

[9] Farghaly, A. and Shaalan, K. 2009. Arabic Natural Language Processing: Challenges and Solutions. Volume 8 , Issue 4 (December 2009), Article No: 14.

[10] Mayfield, J., McNamee, P., Costello, C., Piatko, C., and Banerjee, A. JHU/APL at TREC 2001: Experiments in filtering and in Arabic, video, and web retrieval. In TREC 2001. Gaithersburg: NIST, 2001.

[11] Peter, C., 2006.A Comparison of Personal Name Matching: Techniques and Practical Issues. Data Mining Workshops, 2006. ICDM Workshops 2006