Effective Unsupervised Arabic Word Stemming: Towards an Unsupervised Radicals Extraction

Ahmed Khorsi
Department of Computer Science, Al-Imam Mohammed Ibn Saud Islamic University,
Kingdom of Saudi Arabia

Abstract: *This paper presents a new totally unsupervised and 90% effective stemming approach for classical Arabic. This stemming is meant to be a preparatory step to an unsupervised root (i.e., radicals) extraction. As a learning input, our stemming system requires no linguistic knowledge but a plain classical Arabic text. Once the learning input analyzed, our stemming system is able to extract the strongest segment of a given length, namely the stem. We start by a definition of the targeted stem, then, we show how our system performs about 90% true positives after a leaning of less than 15000 words. Unlike the other unsupervised approaches, ours does not suppose the perfectness of the input text and deals efficiently with the eventual (practically very frequent) misspellings. The test corpus we have used is an ultimate reference in the classical Arabic and its labeling has been rigorously done by a team of experts.*

Keywords: *Computational morphology, machine learning, natural language processing, classical arabic, semitic languages.*

Received September 7, 2010; accepted October 24, 2010

1. Introduction

In Text Information Extraction (TIE) tasks, a system should be able to bind words derived from the same linguistic origin to each other. For instance, an Information Retrieval (IR) system should be able to find a relevant document even if it contains only variants of the query's words. Searching with singular form should not prevent the matching of documents with only plural forms and so on.

One common technique used to deal with the word's form variants is stemming which aims at reducing all variants to a string common to all variants. Let be w and w' two words in a human language L over an alphabet A. Let $MD(w, w')$ be a measure of the difference between w's and w''s meanings. The traditional stemming attempts to approximate an ideal function R:

$$R: L \rightarrow A^*$$

Such that:

$$R(w) = R(w') \Rightarrow MD(w, w') < \varepsilon$$

where ε is relatively a negligible difference. More intuitively, stemming is reducing a word's original form to another one in such a way that the reduction of all variants of the same word results in the same reduced form [2] and the stemming of any word that does not belong to the variants of the original word results in a different reduced form. For instance, the word 'readings' may be reduced to 'read' since the ending 'ings' does not affect the meaning so much. Words that are not variants of the word 'readings' could not be reduced to the same substring 'read' (e.g., 'bread'). The intent behind is: if the words of some text are all stemmed, as well as the query's ones, seeking the query's stems will ensure the finding of all words in the text with close forms and meanings to the words in the query, even if the query's words do not match exactly those in the text [15]. Still, this will more likely match documents with lesser actual relevance. Actually, it has been recognized that stemming improves recall while it may decrease the precision [24]. In the concatenative languages [16, 22] such as English, words are generally in the formprefixes.stem.suffixes and the problem is to find frontiers (i.e., the starting letter of stem and suffix). In more complex morphologies such as Arabic [11, 12], the stem is not the actual unified carrier of the word's and variants' broad meaning. This is due to the internal changes that affect the word with no substantial effect on the meaning. For instance, pruning the verb 'يكتب' ([jaktubu]: is writing) [1] from the prefix (Arabic is a right to left language) 'يـ' ([j]: the 28th Arabic letter), will not allow the straight matching of the noun 'كاتب' ([ka:tˤb]: writer).

Verb: يـ كـ تـ بـ
Noun: كـ ا تـ بـ

That says: an efficient meaning matching cannot rely on matching only strings resulting from stripping out the common prefixes and suffixes from the sought word. In Arabic, the actual invariant kernel that holds the broad meaning of the word is the root (radicals) [3].

- *Definition 1:* In Arabic, word radicals (root) are letters from which all variants of the word are

derived. Radicals may be of either three or four letters [23].

An unsupervised extraction of the root seems to require a preliminary step that narrows the original form of the word keeping only the shortest substring that encompasses all root letters [18]. This is actually a shrinking of the noisy appendages which is nothing but a kind of stemming. In section three, we go through the known works on unsupervised stem extraction. Many works have been published on unsupervised morphology processing and Arabic stemming, but here, we pay our attention to Arabic able unsupervised approaches. In section four, we detail our algorithm and the principles it is based on. Section five is a discussion of the test results. We finalize by a conclusion and some improvement options.

2. Known Related Works

The oldest published work on unsupervised word segmentation may well be due to Zelig Harris. He suggested in the 50s a process to break down a phonetic text into morphemes [10] using only the successor's numbering. The principle is to count the number of the observed successors and/or predecessor at each position in the word. Obviously, the count will depend on the observed corpus. Harris used a plain dictionary. Positions exceeding a defined threshold *k* of successors are marked as fragments' ends.

- *Example 1:* Let be the word 'writing'. Figure 1 shows the variation of the successors counts observed in a spelling dictionary. The curve rises for the prefix 'writ' which actually points a frontier. Figure 2 shows the variation of the successors counts for the Arabic word 'والكتاب' ([walkita:b]: and the book) observed in a spelling dictionary. The number of successors of the common prefix 'وال' ([wal]: and the) is a peak.

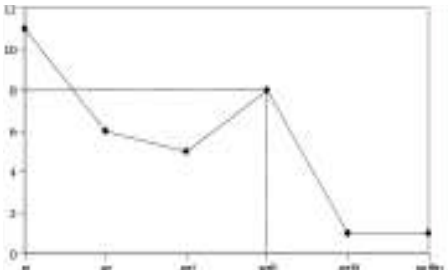

Figure 1. Successors counts variation for the word 'writing'.

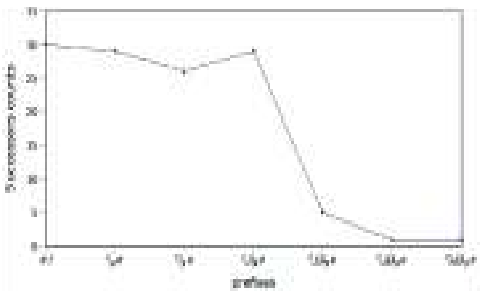

Figure 2. Successors counts variation for the word 'والكتاب'.

Hafer and Weiss [7] adapted Harris's technique to the segmentation of words and tested 15 different setups of four measures:

1. *Successors and Predecessors Threshold:* Same as Harris's.
2. *Peaks vs. Plateaus:* Every peak or plateau is considered as a segmentation position.
3. *Complete Word:* A word prefix is considered to be a segment if and only if it matches a complete word in the corpus.
4. *Entropy:* Let $w=a_0...a_{n-1}$ be a word formed of n letters a_is. Let w_i be the prefix $a_0...a_{i-1}$ and $cnt(w_i)$ the number of words in the corpus sharing the prefix w_i. The word's entropy at a position i is computed as follows:

$$H(w_i) = \sum_{l \in A} -\frac{ctn(w_i.l)}{ctn(w_i)} \log \frac{ctn(w_i.l)}{ctn(w_i)} \quad (1)$$

where l is a letter over an alphabet A and $w_i.l$ the string resulting from appending l to the end of w_i. Any entropy value exceeding a threshold is considered as cutoff point.

Xu and Croft [24] suggested the use a co-occurrence-based measure to build a stemmer or refine an existing one. Their basic hypothesis is "that the word forms that should be conflated for a given corpus will co-occur in documents from that corpus". The same approach has been tested on Arabic [12] and results shown that the co-occurrence approach alone is not so effective for Arabic as it is for English and Spanish. In [19] a statistical translation, an English stemmer and a parallel corpus have been used to build an unsupervised Arabic stemmer. Though, the building itself is supervised. Lee *et al.* [14] used a manually segmented corpus (suffixes, prefixes dictionary) to build a stemmer. Their stemmer attempts to carry out all possible segmentations following the template prefix*.stem.suffix* then checks them against the manually built corpus. A more simplistic approach, often said light stemming is to strip unconditionally out all common prefixes and suffixes [4, 13]. This seems to perform acceptable results in information retrieval. This may be due to the fact observed and used in Xu-Croft's discussed above. Of course, other works attempted to adapt classical approaches such as rule based to Arabic [9]. Other unsupervised morphology approaches have been investigated in the other languages [6, 20]. [8] is a survey on unsupervised methods for concatenative (this is not the case for Arabic) languages.

3. Stemming Approach

First, let us define the stem that our system has to extract.

- *Definition 2:* Let be a word $w=a_1...a_m$ and $root=r_1...r_l$ the root which w is derived from.

stem=$s_1...s_k$ is the stem of *w* if and only if *stem* is the shortest substring of *w* that encompasses all letters of *root*.

- *Example 2:* Let be *w*='الكتاب' (['alkita:b]: the book) Root='ك ت ب' ([k t b]), Stem='كتاب' ([kita:b]: a book).

The stem in this definition is not forcibly common to all variants of the word due to internal inflections. Although, it allows the removal of the common prefixes and suffixes (e.g., 'ال' ['al]). In the example above 'الكتاب' (['alkita:b]: the book) is the singular form of the plural 'الكتب' (['alkutub]: the books). According to the same definition, the stem of the word 'الكتب' is 'كتب' ≠'كتاب'. Recall that our stemming is a step toward an effective extraction of the root not a normalization of the words variants. This means we are about removing all extra letters that do not affect significantly the meaning from the beginning and the end of the word. The light stemming is not suitable for such purpose since it removes all common prefixes and suffixes even if the removed part is not actually a prefix but a core substring. For instance, a light stemmer will wrongly consider 'وال' ([wal]) in the word 'والديه' ([wa:lrdajhi]: his parents) as a prefix.

3.1. Corpus Acquisition

To test our approach we used the Qur'an's [17] text for two main reasons:

1. It is an ultimate reference of the classical Arabic.
2. A group of experts have built manually a database of all Qur'an words mapped to their stems, affixes and most of them to their roots. The resulting database has been thoroughly reviewed.

3.2. Preprocessing and Collations

Once the diacritic symbols removed, the text is processed word by word where all variants of the letter 'ء' (Hamza): 'أ' (['a]), 'ؤ' (['u]), 'ئ' ([I']), 'إ' (['I]) are reduced to the form 'ء' and the compact form 'آ' (['a:]) split into 'اء' (['a:]). Notice that these transformations do alter neither the pronunciation nor the meaning. Then 'ة' ([t]) is replaced by 'ت' ([t]). Finally, conventional character '#' is added to the beginning and the end of the word.

Algorithm 1: Construction of the N-Grams Corpus

1. Foreach letter in w do
2. Replace 'آ' with 'اء'
3. Replace 'أ', 'إ', 'ؤ', 'ئ' with 'ء'
4. Replace 'ة' with 'ت'
5. w← '#'.w.'#'
6. For s←1 to |w|-1 do
7. For e←s+1 to |w| do
8. If $w_{s..e}$∈N-Grams then
9. Increment freq($w_{s..e}$)
10. Else
11. Add $w_{s..e}$ to N-Grams
12. Freq($w_{s..e}$)← 1

The run of this algorithm should carry out:

- *N-Grams:* The set of all distinct n-grams acquired from the plain text mapped to:
- *Freq:* The number of their occurrences.

3.3. Stem Extraction

The only knowledge we will be using is that extracted from the input plain text: N-Grams and freq. Our basic idea is that: "a letter is irremovable from the beginning or the end of a word if the remaining substring depends relatively on this letter".

3.3.1. Dependency Measure

Let be $w_{s+1..s+l}$ (i.e., w_{s+1} ...w_{s+l}) a substring of *l* letters extracted form the word *w*. We would like to measure how dependent is $w_{s+1..s+(l-1)}$ on w_{s+l}. Instead of basing our measure on letters variations as do Harris's like approaches, we evaluate the conditional probability over the letter's probability. Let us call forward dependency *fd*:

$$fd(w_{s+1...s+l}) = \frac{P(w_{s+l} \mid w_{s+1...s+(l-1)})}{P(w_{s+l})} \quad (2)$$

This measure of dependency is not new and has been used to measure the events dependencies [5]. All terms in the formulae above could be computed using *N-Grams* and *freq* since:

$$P(w_{s+l} \mid w_{s+1...s+(l-1)}) = \frac{P(w_{s+1...s+l})}{P(w_{s+1...s+(l-1)})} \quad (3)$$

Where:

$$P(w_{i+1...j}) = \frac{freq(w_{i+1...j})}{\sum freq(A^{(j-i)})} \quad (4)$$

where $A^{(j-i)}$ denotes the n-grams of the length *j-i* ($0 \le i \le j \le |w|$). The Backward Dependency *bd* is a measure of how dependent is $w_{s+2...s+l}$ on the first letter w_{s+1}.

$$bd(w_{s+1...s+l}) = \frac{P(w_{s+1} \mid w_{s+2...s+l})}{P(w_{s+1})} \quad (5)$$

Example 3: Let be the word 'الكتاب':

$$fd(\text{'كتاب '}) = \frac{P(\text{'الكتا'} \mid \text{'ب '})}{P(\text{'ب'})} \quad (6)$$

and

$$bd(\text{'كتاب '}) = \frac{P(\text{'الكتا'} \mid \text{'تاب '})}{P(\text{'تاب '})} \quad (7)$$

We come to the algorithm that finds the extractable segment of *l* ($l \le |w|$) letters which strongly depends on its leading and ending letters.

Algorithm 2: Extraction of a Segment of the length l

1. *Foreach letter in w do*
2. *Replace 'أ' with 'ء'*
3. *Replace 'أ', 'إ', 'ؤ', 'ئ' with 'ء'*
4. *Replace 'ة' with 'ت'*
5. *w← '#'.w.'#'*
6. *For s← 2 to |w|- (l-1) do*
7. *If $fd(w_{s...s+(l-1)}) < fd(w_{(s-1)...(s-1)+(l-1)})$ then*
8. *Add $w_{s...s+(l-1)}$ to FS*
9. *For s←|w|-(l-1)-1 to 1 do*
10. *If $bd(w_{s...s+(l-1)}) < fd(w_{(s+1)...(s+1)+(l-1)})$ then*
11. *Add $w_{s...s+(l-1)}$ to BS*
12. *Seg←u ∈ (FS∩BS): $fd(u)+bd(u)=max_{v \in FS \cap BS} fd(v)+bd(v)$*

- *Example 4:* For the word 'الكتاب', we would like to find the extractable 4 letters segment. Figure 3 shows the variations in forward and backward dependencies. Two clearly distinguishable peaks appear in both the forward and backward dependencies graphs on the segment 'ال' (['al]). This means that this segment is the most inseparable from its leading and ending letters. Actually, this is the actual 4 letters stem see section 3.

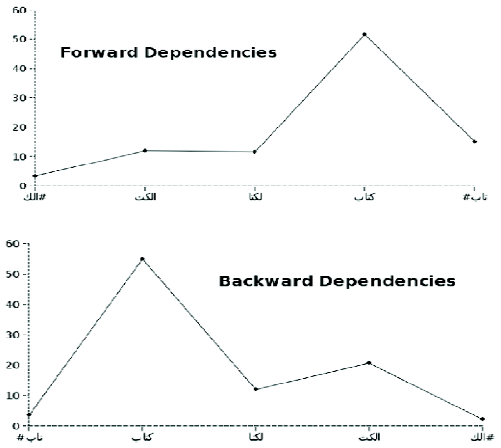

Figure 3. Dependencies variations for 4-grams extracted from the word 'الكتاب'.

4. Tests and Results

4.1. Test Set

During a prior investigation, a group of experts have manually processed the text of the Holy Quran. The resulting corpus holds exactly 14011 traditional distinct Arabic words mapped to their stems and roots (i.e., radicals). Below is a sample.

Table 1. Morphological analysis database structure.

Sora Number	2
Verse Number	2
Word Number	2
Diacretized Word	الكِتَابُ
Undiacretized Word	الكتاب
Prefix	ال
Stem	كتاب
Suffix	
Root	كتب

We used this corpus for two main advantages:

1. Stems and roots where manually extracted and rigorously reviewed. This means that our baseline is a pure human expert decision.
2. We targeted pure classical Arabic and the text our approach is being tested on is the ultimate reference for the language. Moreover, we could not be able to find a corpus that targets the old classical Arabic. All of the available corpora to our knowledge are for Modern Arabic Vocabulary (MAV).

From this corpus we need only the undiacritized word form, its root and stem. We report hereafter the performance of our approach on two objectives mentioned in the definition at the beginning of section 3:

1. Capability of extracting a segment that gathers all root's letters.
2. Minimality of the extracted segment satisfying the first objective (Definition 2).

We have measured three values:

- *ES:* Number of times our system has been able to extract a segment.
- *MS:* Number of times our system has been able to extract the minimal segment (Definition 2).
- *SS:* Number of times our system has been able to extract a segment of the same length or shorter than the manually selected stem. Of course, our approach may result in a stem shorter than that the human expert chooses. For instance, the human expert would judge that the stem of the word 'كلمة' is itself. According to our definition (Definition 2), the stem is 'كلم'. Recall that our approach is not the final word normalization, but groundwork for radical extraction.

Table 2 and Figure 4 show the results for three different collation setups:

1. Long vowels ('ا','و','ي') are deleted before the comparison.
2. All long vowels normalized to 'ا'.
3. No additional transformation is performed on long vowels.

Indeed, most of the changes that happen in the radicals when a word is derived are: deletion, insertion or substitution of a long vowel (i.e., و [w], ا or ى [a:], ي [j]) [25].

- *Example 4:* Deriving respectively the words 'كاتب' ([ka:tˌb]: writer), 'كتيب' ([kutajˌb]: small book) and 'كوتب' ([ku:tˌba]: a writing has been addressed to him) from the radicals 'ك ت ب' is insertion of the long vowel 'ا' ([a:]) in the first word, the long vowel 'ي' ([j]) in the second one and the long vowel 'و' ([w]) in the last one.

Radicals	ك		ت	ب
Word	ك	ا	ت	ب
Radicals	ك	ت		ب
Word	ك	ت	ج	ب
Radicals	ك		ت	ب
Word	ك	و	ت	ب

Deriving the verb 'سار' ([saːra]: has walked at the beginning of the night) from the radicals 'س ي ر' is a substitution of the long vowel 'ي' by the long vowel 'ا'. And the verb 'سر' ([sɪr]: walk [Imperative]) is derived by deleting the long vowel 'ي'.

Radicals	س	ي	ر
Word	س	ا	ر
Radicals	س	ي	ر
Word	س		ر

Table 2. Test results.

Collation Setup	Value	Count	Percentage Over 13893 Words
Long Vowels Deleted	ES	12846	92.46%
	MS	9409	67.72%
	SS	10120	72.84%
Long Vowels Normalized to 'ا'	ES	12318	88.66%
	MS	8317	59.86%
	SS	9182	66.09%
No Additional Normalization	ES	10497	75.56%
	MS	7329	52.75%
	SS	7916	56.98%

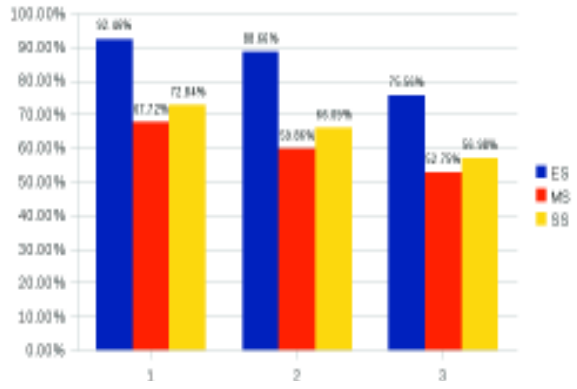

Figure 4. Test results.

4.2. Discussion

The higher ES, MS and SS are achieved when the long vowels و [w], ا or ى [a:], ي [j] are left out in the comparison; which was expected (Example 4). Actually, this setup prevents mismatches due to long vowels changes discussed earlier. First setup's ES, MS and SS go over the third setup's ES, MS and SS by about 15% (respectively 16.9, 14.97 and 15.86). This gives to conclude that the choice of the Arabic collation has a significant effect on the matching. In most investigations we have seen, the collation side is vaguely handled and never argued. In all setups, ES is clearly higher than MS and SS. Algorithm 2 may fail to carry out a result when $BS \cap BS = \varnothing$. *ES* is a measure of its successfulness in finding out a result even when it is not the minimal one. MS is the lowest rate, thing which would be expected since it is the most restrictive measure. Although, this is the measure of how successful our system was in carrying out the targeted stem (see Definition 2). Notice that for the three experiments setups, this measure is above 50% which supports the effectiveness of the dependency measure we used (see section 3.3.1). MS is lesser than SS for two main reasons:

1. Our definition of the stem is different from the common one.
2. Our corpus is far from being exhaustive. In fact, when a word form is dominant and the corpus misses the other variants of the same word, the dependency of the actual stem on the extra letters carried by the dominating form is high and those extra letters are considered by the system as core letters. For instance, a corpus which does not include the other variants of the word 'مقعد' ([maq?ad]: seat) will make the system learn that the segment 'قعد' [q?ad] (which is the actual stem) cannot be separated from the letter 'مـ' [m].

This led us to measure how successful was the system to carry out a stem shorter than the human expert chosen one. In the best case, it reached 72%. Here also, we could not say that the stem our system chosen was a better solution, but the targeted stem in our approach is different from that targeted by a human judgment.

5. Conclusions

We have presented a totally unsupervised approach for the extraction of morphological segments from classical Arabic words. This extraction is intended to be a preprocessing step to the unsupervised extraction of radicals. After a learning of relatively a small traditional Arabic text (less than 15000 words), with a totally unsupervised processing where the system is not fed with any kind of structured linguistic knowledge, results are very satisfactory. However, one of the improvements tracks to explore is the learning corpus. We mean by the learning corpus, the plain text used to extract the n-gram frequencies. One option is to use a huge dictionary once and build an index of frequencies. This may cover a wide range of words variants. A second improvement path is to explore the n-gram collation tuning. For instance, in many words, a long vowel is a core letter and its deletion results in a wrong dependency value and may lead to a confusion with other words. Adding some n-gram restrictions to the collation convention seems to avoid a part of the problem. A third perspective is to assess the system with fully diacretized text. In our case, this is possible since the Quran's text is fully and meticulously diacretized. However, the resulting dependencies may

be usable only with fully diacretized words. The next step is the unsupervised radicals (root) extraction.

References

[1] International Phonetic Association, *Handbook of the International Phonetic Association: a Guide to the use of the International Phonetic Alphabet*, Cambridge University Press, 1999.

[2] Baeza-yates R., "Challenges in the Interaction of Information Retrieval and Natural Language Processing," *in Proceedings of Computational Linguistics and Intelligent Text Processing*, Berlin, pp. 445-456, 2004.

[3] Chejne A., *The Arabic Language: It's Role in History*, University of Minnesota Press, 1969.

[4] Chen A. and Gey F., "Building an Arabic Stemmer for Information Retrieval," *in Proceedings of the 11th Text Retrieval Conference (TREC)*, Gaithersburg, pp. 631-639, 2002.

[5] Falk R. and Bar-Hillel M., "Probabilistic Dependence between Events," *The Two-Year College Mathematics Journal*, vol. 14, no. 3, pp. 240-247, 1983.

[6] Goldsmith J. "Unsupervised Learning of the Morphology of a Natural Language," *Computational Linguistics*, vol. 27, no. 2, pp. 153-198, 2001.

[7] Hafer M. and Weiss S., "Word segmentation by letter successor varieties," *Information Storage and Retrieval*, vol. 10, no. 11-12, pp. 371-385, 1974.

[8] Hammarström H. "Unsupervised Learning of Morphology: Survey, Model, Algorithm and Experiments," *Thesis for the Degree of Licentiate of Engineering*, Chalmers University of Technology and G¨oteborg University, 2007.

[9] Harmanani H., Keirouz W., and Raheel S., "A Rule-Based Extensible Stemmer for Information Retrieval with Application to Arabic," *The International Arab Journal of Information Technology*, vol. 3, no. 3, pp. 265-272, 2006.

Computational Linguistics, Japan, vol. 1, pp. 391-398, 2003.

[20] Snover M., Jarosz G., and Brent M., "Unsupervised Learning of Morphology using a Novel Directed Search Algorithm: Taking the First Step," *in Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, vol. 6, pp. 11-20, 2002.

[21] Soudi A. and Bosch A., "Arabic Computational Morphology: Knowledge-Based and Empirical Methods," Humanities, Social Sciences and Law, Arabic Computational Morphology, Springer, vol. 38, pp. 3-14, 2007.

[10] Harris Z., "From Phoneme to Morpheme," *Language, Linguistic Society of America*, vol. 31, no. 2, pp. 190-222, 1955.

[11] Jurafsky D. and Martin J., *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice Hall, New York, 2000.

[12] Larkey S., Ballesteros L., and Connell E., "Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-Occurrence Analysis," *in Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 275-282, 2002.

[13] Larkey L., Ballesteros L. and Connell M., "Light Stemming for Arabic Information Retrieval," *in Proceedings of the Arabic Computational Morphology*, Springer-Netherlands, pp. 221-243, 2007.

[14] Lee Y., Papineni K., Roukos S., Emam O., and Hassan H., "Language Model Based Arabic Word Segmentation," *in Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, Japan, vol. 1, pp. 399-406, 2003.

[15] Manning C., Raghavan P., and Schütze H., *Introduction to Information Retrieval*, Cambridge University Press, New York, 2008.

[16] Menn L., *Non-Fluent Aphasia in a Multilingual World*, John Benjamins Publishing Company, 1955.

[17] Qur'an ed. "Quran Meanings Translation," King Fahd Complex for Printing the Holy Qur'an, 2010.

[18] Roeck A. and Al-Fares W., "A Morphologically Sensitive Clustering Algorithm for Identifying Arabic Roots," *in Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, Hong Kong, pp. 199-206, 2000.

[19] Rogati M., McCarley S., and Yang Y., "Unsupervised Learning of Arabic Stemming using a Parallel Corpus," *in Proceedings of the 41st Annual Meeting on Association for*

[22] Sproat W., "Morphology and Computation," MIT Press, 1992.

[23] Versteegh K., *The Arabic language*, Edinburgh University Press, 2001.

[24] Xu J. and Croft W., "Corpus-Based Stemming using Cooccurrence of Word Variants," *ACM Transactions on Information Systems*, vol. 16, no. 1, pp. 61-81, 1998.

[25] جني إ., "التصريف الملوكي" دار الفكر العربى للطباعة والنشر, 1998.

Ahmed Khorsi is an assistant professor at the college of computer and information system in Al-Imam Mohammed Ibn Saud Islamic University since 2008. He obtained his computer science Master's degree in 2002 and PhD degree in the same filed in 2007 from Algeria. During his research career, has been interested in automata theory, knowledge representation, machine learning and computer security. He was one of the main contributors to Qur'an project funded by King Abdulaziz City of Science and Technology. He is working on Arabic NLP as his main field of research. He has advanced skills in key technologies such as databases and programming and is especially interested in open source systems.

Effective Unsupervised Arabic Word Stemming: Towards an Unsupervised Radicals Extraction. A Khorsi. Int. Arab J. Inf. Technol. 9 (6), 571-577, 2012. 5. 2012. A Two-Level Plagiarism Detection System for Arabic Documents. A Khorsi, H Cherroun, D Schwab. Cybernetics and Information Technologies 20, 2018.Â Towards Hybridization of Knowledge Representation and Machine Learning. A Khorsi. Computing and Informatics 26 (2), 123-147, 2012. 2. 2012. 2L-APD: A Two-Level Plagiarism Detection System for Arabic Documents. A Khorsi, H Cherroun, D Schwab. Cybernetics and Information Technologies 18 (1), 124-138, 2018. 1. 2018. Unsupervised Affix Identification Approach using Probabilistic Dependence. A Alsheddi, A Khorsi. Unsupervised keyphrase extraction has a series of advantages over supervised methods. Super-vised keyphrase extraction always requires the ex-istence of a (large) annotated corpus of both doc-uments and their manually selected keyphrases to train on - a very strong requirement in most cases. Supervised methods also perform poorly outside of the domain represented by the training corpus - a big issue, considering that the domain of new documents may not be known at all. Unsupervised keyphrase extraction addresses such information-constrained situations in one of two ways: (a) by relying on in-c