Journal of King Saud University – Computer and Information Sciences (2016) **28**, 157–169

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Rational kernels for Arabic Root Extraction and Text Classification☆

Attia Nehar [a,*], **Djelloul Ziadi** [b], **Hadda Cherroun** [a]

[a] *Laboratoire d'informatique et Mathématiques, Université A.T. Laghouat, Algeria*
[b] *Laboratoire LITIS – EA 4108, Normandie Université, Rouen, France*

Received 9 April 2015; revised 8 November 2015; accepted 8 November 2015
Available online 19 December 2015

KEYWORDS

N-gram;
Arabic;
Classification;
Rational kernels;
Automata;
Transducers

Abstract In this paper, we address the problems of Arabic Text Classification and root extraction using transducers and rational kernels. We introduce a new root extraction approach on the basis of the use of Arabic patterns (Pattern Based Stemmer). Transducers are used to model these patterns and root extraction is done without relying on any dictionary. Using transducers for extracting roots, documents are transformed into finite state transducers. This document representation allows us to use and explore rational kernels as a framework for Arabic Text Classification. Root extraction experiments are conducted on three word collections and yield 75.6% of accuracy. Classification experiments are done on the Saudi Press Agency dataset and N-gram kernels are tested with different values of *N*. Accuracy and F1 report 90.79% and 62.93% respectively. These results show that our approach, when compared with other approaches, is promising specially in terms of accuracy and F1.

1. Introduction

Text Classification (TC) is a machine learning-based task. It aims to automatically sort a set of documents into one or more classes from a predetermined set (Sebastiani and Ricerche, 2002). Applications of TC include many domains, such as article indexing, Web information searching, mail spam detection, and even automatic assessment systems.

In this work, we enhance the root extraction technique, introduced by authors in a previous paper (Nehar et al., 2012), and we asses its performance in the context of Arabic Text Classification (ATC). Indeed, root extraction method introduced in Nehar et al. (2012) gives a set of possible roots. Our new root extraction approach chooses the best root based on a statistical study of character occurrences in the Arabic roots corpus علي حلمى موسى (1978). Experiment and comparison are conducted to assess performances against standard stemmers. Our root extraction technique transforms documents into finite state transducers. Then, rational kernels (Cortes et al., 2004), which

☆ This work is supported by the MESRS – Algeria under project 8/U03/7015.
* Corresponding author.
E-mail addresses: a.nehar@mail.lagh-univ.dz (A. Nehar), djelloul.ziadi@univ-rouen.fr (D. Ziadi), h.cherroun@mail.lagh-univ.dz (H. Cherroun).
Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2015.11.004

are language/task independent methods, are used as a framework to do ATC. This allows the use of different distance measures or kernels, like N-grams kernels, with the aim of identifying the suitable value for N.

The rest of this paper is organized as follows. Section 4 presents the two main types of stemming, namely: light stemming and root extraction (or heavy stemming) techniques. In Section 5, we recall some notions on weighted transducers and rational kernels. We present, in Section 6 our new root extraction approach, then we explain how to use rational kernels as a framework for ATC. Experiments and results are reported and interpreted in Section 7.

2. Related work

Due to the increased availability of Arabic documents in digital form and the complexity of the Arabic language, Arabic Text Classification (ATC) has increasingly begun to receive attention. Significant work has been conducted to improve performance of ATC systems (El Kourdi et al., 2004; Syiam et al., 2006; Duwairi, 2007; Althubaity et al., 2008; Hadi et al., 2008; Mesleh, 2008; Gharib et al., 2009; Kanaan et al., 2009; Khreisat, 2009; Alsaleem, 2011; Hadni et al., 2013; Hmeidi et al., 2014).

In general, an ATC system consists of three steps:

1. *Preprocessing step*: text is normalized by removing diacritics, punctuation marks, stopwords, special characters, numbers and all non-letter characters.
2. *Features extraction*: text is transformed into a vectorial form by extracting a set of features. For example, (Khreisat, 2009) performed features extraction by using N-gram technique, while (Syiam et al., 2006) relied on stemming. In addition, terms weighting and feature reduction techniques could be used to enhance performance.
3. *Learning step*: in this step, the goal is to teach the system how to classify Arabic text documents. Many supervised algorithms were used: Support Vector Machines (Alsaleem, 2011; Gharib et al., 2009; Mesleh, 2008), K-Nearest Neighbours (Hadi et al., 2008; Syiam et al., 2006, Naive Bayes Alsaleem, 2011; El Kourdi et al., 2004; Hadi et al., 2008). Most techniques rely on similarity measures over extracted features to determine whether two documents are similar.

In the first step, elements that add nothing to the meaning of documents are removed from the text. The second step aims to represent documents in a vectorial form by extracting a set of features from the documents. The Bag of Words (BOW) was by far the simplest way to represent documents in a vectorial form. All the words are used to index a vector representing occurrences of words in a document. Many improvements of the BOW were proposed to enhance ATC systems, including feature selection, dimension reduction, terms weighting and stemming. The BOW is a word level representation. In Khreisat (2009), N-gram technique with a character level is used. Stemming is used to reduce dimensionality. Several stemming techniques are proposed (Buckwalter, 2004; Al-Nashashibi et al., 2010). Authors of Khoja and Garside (1999) designed a dictionary based root extraction technique that reports good results, but the dictionary have to be kept up-to-date. The root extraction technique developed in Al-Serhan et al. (2003) gets the three-letter roots for Arabic words without relying on any language resources such as pattern files or roots dictionary.

In Arabic language, surface words could be classified to root family classes, i.e, words that are derived from the same root but do not have the same sense. Reducing semantically distinct words to the same root can lead to classification performance decrease. In order to avoid this, light stemming is applied in TC systems (Aljlayl and Frieder, 2002). Light stemming consists of removing a small set of prefixes and/or suffixes, without trying to consider infixes, or detecting patterns. Due to this strategy, light stemming results in a large number of features compared with root extraction.

In the third step, most algorithms depend on distance metrics to evaluate the similarity (or dissimilarity) between documents using feature vectors. The quality of the classification system is related to the used distance measure.

3. Challenges and linguistic issues in ATC and root extraction

Arabic Text Classification faces many challenges. The first important challenge is related to Arabic morphological analysis, which is a crucial tool for ATC systems. Indeed, the process of Text Classification depends on the content of documents, a massive number of features can lead to poor performance in terms of accuracy and time. Arabic is lexically a very rich language, important number of surface words can be generated from one stem or root. Treating all surface words will end up with a very large number of features, one solution is to use morphological tasks, like stemming and root extraction. The second challenge concerns the semantical level of Arabic language, Text Classification is sensitive to expressions meaning. The morphological richness and orthographic ambiguity, due to optional diacritization, can lead to a large number of homographs and homonyms (Habash, 2010). Synonyms are also widespread in Arabic language. The third challenge is the lack of publicly available free Arabic corpora for evaluating ATC systems. Much work was done on manually obtained datasets. This lack should be fulfilled over time with standard and benchmark corpora. In the next paragraphs we will give more details about these challenges.

Morphological analysis is the study of internal word structure (Habash, 2010). Morphologically, the Arabic language is the most complicated and rich language. Many words can be formed using the same root, a few patterns, and a few affixes. One of the challenges in a root extraction is that words in Modern Standard Arabic are free of diacritics, which makes them more ambiguous. For instance, the two words (كَتَبَ, He wrote) and (كُتُبٌ, Books) are originated from the same root but has different meanings when vocalized. Furthermore, unvowelled words can lead to more important ambiguity. Lets take the two words: (يَنَعَتْ, Ripen) and (يَنْعِتُ, Qualify). These words originated from two different roots (ينع and نعت respectively) though have the same orthography.

Multiple affixes and clitics can appear in a word, due to agglutinative nature of Arabic, sometimes giving word-forms

is translated to a whole sentence in English. For example, the Arabic word (أنلزمكموها) is translated to (Shall we compel you to accept it).

Roots with geminate or weak radicals need specific rules when analyzed. For the first type of roots, one of the doubled letters is removed in the word-form, the root recovery algorithm needs to handle this case. For example, the word (شدّت, She pulls) is generated from the root (ش د د). For the latter type of roots, things are more complicated. Weak root radicals (و ي) change into a vowel or are deleted depending on their vocalic environment (Habash, 2010). The main difficulty when extracting roots is deciding whether a word is generated from weak roots. The word (التنمية, The development) is generated from the root (م و ن), we can notice that the letter (و) doesn't appear in the word.

Spelling mistakes are common in the Arabic documents. Obvious errors can be handled easily by modern analysis tools. However, non obvious errors can cause ATC systems to be less effective. This type of errors can be hard to identify if the misspelled word happens to be a valid word. For example, the misspelled word (أقصد, I mean) for which the first letter أ was deleted, remains a valid Arabic word (قصد, Go to).

Broken plurals, broken feminine and many other peculiarities of Arabic language are complicating factors of morphological analysis.

4. Stemming approaches

As part of ATC systems, stemming is applied to reduce dimensionality of the feature vectors. Root extraction (commonly called heavy stemming) transforms each surface Arabic word in the document, into its root. However, light stemming, reduces word by removing prefixes and suffixes.

4.1. Root extraction

There are several root extraction techniques used as a part of ATC systems. It can be grouped into two types: (i) *Root extraction using a dictionary*, where the dictionary of Arabic word roots is needed. (ii) *Root extraction without dictionary*, where roots are determined without relying on any pattern or root files.

One of the earliest works is due to Khoja and Garside (1999). This heavy stemmer attempts to locate and remove the longest prefix and longest suffix from a word, then checks a list of verb and noun patterns to determine whether the remainder could be a known root with a known pattern applied. This tool relies on many linguistic resources including a list of all diacritic and punctuation characters, definite articles and stop words. This heavy stemmer reports good results, but the dictionary has to be maintained. Taghva et al. (2005) implement a root extraction method that shares many features with the Khoja stemmer except for the use of root dictionary. Al-Serhan et al. (2003), developed a statistical based algorithm, it extracts the three-letter roots for Arabic words without needing any additional linguistic resources such as patterns or root pattern files. Word roots are extracted by assigning ranks and weights to letters that compose a word. Consonants were assigned a weight of zero and different weights were assigned to all affixes that can be formed from the letters grouped in the word (سألتمونيها). The algorithm chooses the three letters having the least products (weight × rank) as root letters. Weights and ranks are assigned to letters using a little bit information on language (Al-Serhan et al., 2003). In other works (Ghwanmeh et al., 2009; Harmanani et al., 2006; Momani and Faraj, 2007), a rule-based approach was used. For instance, (Harmanani et al., 2006) proposed a method in which roots are extracted based on a set of language dependent rules that are interpreted by a rule engine. More recent works tend to use all available resources (roots file, patterns file and rules) to extract roots (Hadni et al., 2013; Yaseen and Hmeidi, 2014; Al-Kabi et al., 2015). This tendency to use all available resources indicates that root extraction is yet a challenging task.

4.2. Light stemming

In Arabic language, most surface words are built from roots and by applying patterns on these roots to derive words. Even they share the same root (س د ر), the two words: (مدرسة) which means *school* and (دراسة) which means *a study*, do not have the same meaning. Thus, stemming can affect the meaning of words. Light stemming (Aljlayl and Frieder, 2002; Darwish, 2002) attempts to improve the TC performance while conserving the words meaning. In Aljlayl and Frieder (2002), the basis of the algorithm involves several rounds through the text, that attempt to find and remove the most recurrent prefixes and suffixes from a word. Other works focus on the repercussion of light stemming on the efficacy on information retrieval (IR) (Kanaan et al., 2008; Larkey et al., 2007). They conclude that light stemming has a positive effect on Arabic IR. However, the main drawback is that light stemming leads to a lot of features. In this work, we focus on the effect of root extraction on Arabic Text Classification systems.

5. Weighted transducers and rational kernels

Before we describe our ATC and stemming framework, let's give in what follows, some preliminaries on Weighted transducers and rational kernels.

Transducers are a generalized form of finite automata. Indeed, each transition is assigned with an output etiquette (label) in addition to the familiar input etiquette. Output etiquettes are assembled along a path to form an output sequence as with input etiquettes. *Weighted transducers* are finite-state transducers with transitions carrying some weight in addition to the input and output etiquettes. The weight of a pair of input and output strings (x, y) is computed by summing the weights of the paths labeled with (x, y). The following gives definition of weighted transducers (Berstel, 1979; Cortes et al., 2007).

Definition 5.1. A weighted finite-state transducer T over the semiring $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ is given by:

$T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$ where Σ is a finite input alphabet, Δ is a finite output alphabet, Q is a finite set of states, $I \subseteq Q$ the set of initial states, $F \subseteq Q$ the set of final states, $E \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times (\Delta \cup \{\epsilon\}) \times \mathbb{K} \times Q$ a finite set of transitions, $\lambda : I \to \mathbb{K}$ the initial weight function, and $\rho : F \to \mathbb{K}$ the final weight function

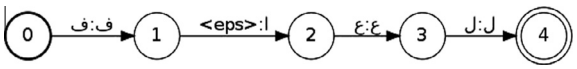

Fig. 1 Example of a transducer.

Table 1 Measures for the 3-letter root ش ر ك and built words.

Measures	مفاعلة	فاعل	الفعالة	يفاعل	يتفاعل
Words	مشاركة	شارك	الشراكة	يشارك	يتشارك

For a path π in a transducer, $p[\pi]$ denotes the origin state of that path, $n[\pi]$ its destination state and $w[\pi]$ gives the summation of the weights of its arcs. We denote by $P(I,x,y,F)$, the set of paths from the start states I to the final states F labeled with input string x and output string y. A transducer T is *regulated* if the output weight affected by T to any pair of input-output strings (x,y) given by:

$$[\![T]\!](x,y) = \oplus_{\pi \in P(I,x,y,F)} \lambda(p[\pi]) \otimes w[\pi] \otimes \rho[n[\pi]] \quad (1)$$

is well-defined in $\mathbb{K}$. If $P(I,x,y,F) = \varnothing$ then $[\![T]\!](x,y) = \bar{0}$. Fig. 1 shows an example of a simple transducer, with an input string x : فاعل and an output string y : فعل. The sole possible path in this transducer is the singular set : $P(\{0\},x,y,\{4\})$.

Regulated weighted transducers are closed under the following operations called rational operations:

- The *sum* (or *union*) of two weighted transducers T_1 and T_2 is defined by:

$$\forall (x,y) \in \Sigma^* \times \Sigma^*, [\![T_1 \oplus T_2]\!](x,y) = [\![T_1]\!](x,y) \oplus [\![T_2]\!](x,y) \quad (2)$$

- The *product* (or *concatenation*) of two weighted transducers T_1 and T_2 is defined by:

$$\forall (x,y) \in \Sigma^* \times \Sigma^*, [\![T_1 \otimes T_2]\!](x,y) = \oplus_{x=x_1x_2, y=y_1y_2} [\![T_1]\!](x_1,y_1) \otimes [\![T_2]\!](x_2,y_2) \quad (3)$$

- The composition of two weighted transducers T_1 and T_2 with matching input and output alphabets Σ, is a weighted transducer denoted by $T_1 \circ T_2$ when the sum:

$$[\![T_1 \circ T_2]\!](x,y) = \oplus_{z \ in \Sigma^*} [\![T_1]\!](x,z) \otimes [\![T_2]\!](z,y) \quad (4)$$

is well-defined in $\mathbb{K}$ for all $x,y \in \Sigma^*$.

Rational kernels are a general family of kernels, based on weighted transducers, that extend kernel methods to the analysis of variable-length sequences or more generally weighted automata. Let X and Y be non-empty sets. A function $K : X \times Y \to \mathbb{R}$ is said to be a kernel over $X \times Y$. Cortes et al. (2004) give a formal definition for rational kernels:

Definition 5.2. A kernel K over $\Sigma^* \times \Delta^*$ is said to be rational if there exists a weighted transducer $T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$ over the semiring $\mathbb{K}$ and a function $\varphi : \mathbb{K} \to \mathbb{R}$ such that for all $x \in \Sigma^*$ and $y \in \Delta^*$:

$$K(x,y) = \varphi([\![T]\!](x,y)) \quad (5)$$

K is then said to be defined by the pair (φ, T).

6. Framework for Arabic Root Extraction and TC

In the next section we clarify how to employ transducers to extract roots. First, Arabic patterns, prefixes and suffixes are modeled by simple transducers, then, a root extraction transducer is constructed using these simple ones by applying rational operations like concatenation, union and composition. Then, we show how to use rational kernels as a framework to do ATC.

6.1. Extracting roots by transducers

Arabic language, which is a Semitic language, is unlike other languages in many aspects, such as syntactical, morphological and semantic aspects. It is a very inflectional language and one of the key properties is that words, for the most part, are created from roots by following certain patterns and adding prefixes and suffixes. For instance, the Arabic word الشراكة (Partnership) is built from the three-letters root شرك and using the pattern فعال, then prefix ال and suffix ة (which is used to denote female gender) are added. This results in the measure الفعالة (see Table 1). Notice here that the letter ف denotes the first letter of the three-letter root, ع denotes the second one and ل denotes the third one.

Measures are used to construct a root extraction transducer. Fig. 1 shows the example of the measure فاعل.. This transducer (T_{m1}) will be employed to get the three-letter root of any surface Arabic term fitting with this measure, through the use of composition operation (4). We denote by T_{term}, the transducer that maps any term to itself, i.e., the sole possible path is the unique path given by: $P(\{0\}, term, term, \{i\})$ (Fig. 2 shows transducer associated to the Arabic word المدرسة).

When composing two transducers, the result is a transducer as well.

$$(T_{word} \circ T_{m1})(term,y) = \sum_{z \in \Sigma^*} T_{term}(term,z) \cdot T_{m1}(z,y)$$

Since the only possible string matching z is $z = term$, we conclude that:

$$(T_{term} \circ T_{m1})(term,y) = T_{term}(term,term) \cdot T_{m1}(term,y)$$

As we have $T_{term}(term,term) = 1$, so:

$$(T_{term} \circ T_{m1})(term,y) = T_{m1}(term,y)$$

If *term* fits with the measure, the resulting transducer will give the root y associated to *term*.

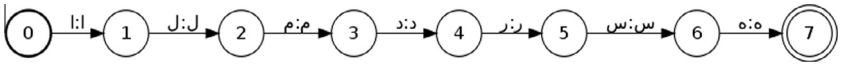

Fig. 2 Transducer corresponding to the word المدرسة (school).

Table 2 Examples of noun patterns.

Noun Patterns				
3-letters	4-letters	5-letters	6-letters	7-letters
فعل	فاعل	مفاعل	متفاعل	استفعال
	فعول	مفتعل	مفعوعل	افعيلال
	مفعل	مفتعل	مستفعل	افتعالة

Table 3 Examples of verb patterns.

Verb Patterns		
3-letters	4-letters	3-letters +1
فعل	فعلل	فاعل
3-letters +2	3-letters +3	4-letters +1
افتعل	استفعل	تفعلل
انفعل	افعوعل	افعنلل
تفاعل		

In Arabic language, there are 4 verb prefixes (ت ي ا ن), 12 noun prefixes (ا، ال، ب، ت، س، ف، لل، ل، ي، و، ن، م) and more than 20 suffixes: (تما، كما، ان، ها، وا، تم، كم، تن، كن، نا، تا، ما، ون، ين، هن، هم، ته، تى، نى، ن، ك، ه، ة، ت، ا، اتت، ي). When taking into account diacritics, the number of patterns can exceed three thousand (to our knowledge). As we don't consider diacritics in our work, patterns are greatly less (not more than two hundred), and many of which are not employed in the context of Modern Standard Arabic (MSA). Indeed, the patterns (فَعَلَ، فَعُلَ، فُعَلٌ، فَعِلٌ) will result in only one pattern (فعل) after removing diacritics. For illustration, Tables 2 and 3 show examples of noun and verb patterns.

We will follow the following steps, to build the stemming transducer, which will enable us to consider all measures:

1. Construct all noun prefixes (resp. verb prefixes) transducer;
2. Construct all noun patterns (resp. verb patterns) transducer;
3. Construct all noun suffixes (resp. verb suffixes) transducer;
4. Concatenate noun transducers (resp. verb transducers) obtained in 1, 2 and 3.
5. Sum the two transducers obtained in step 4.

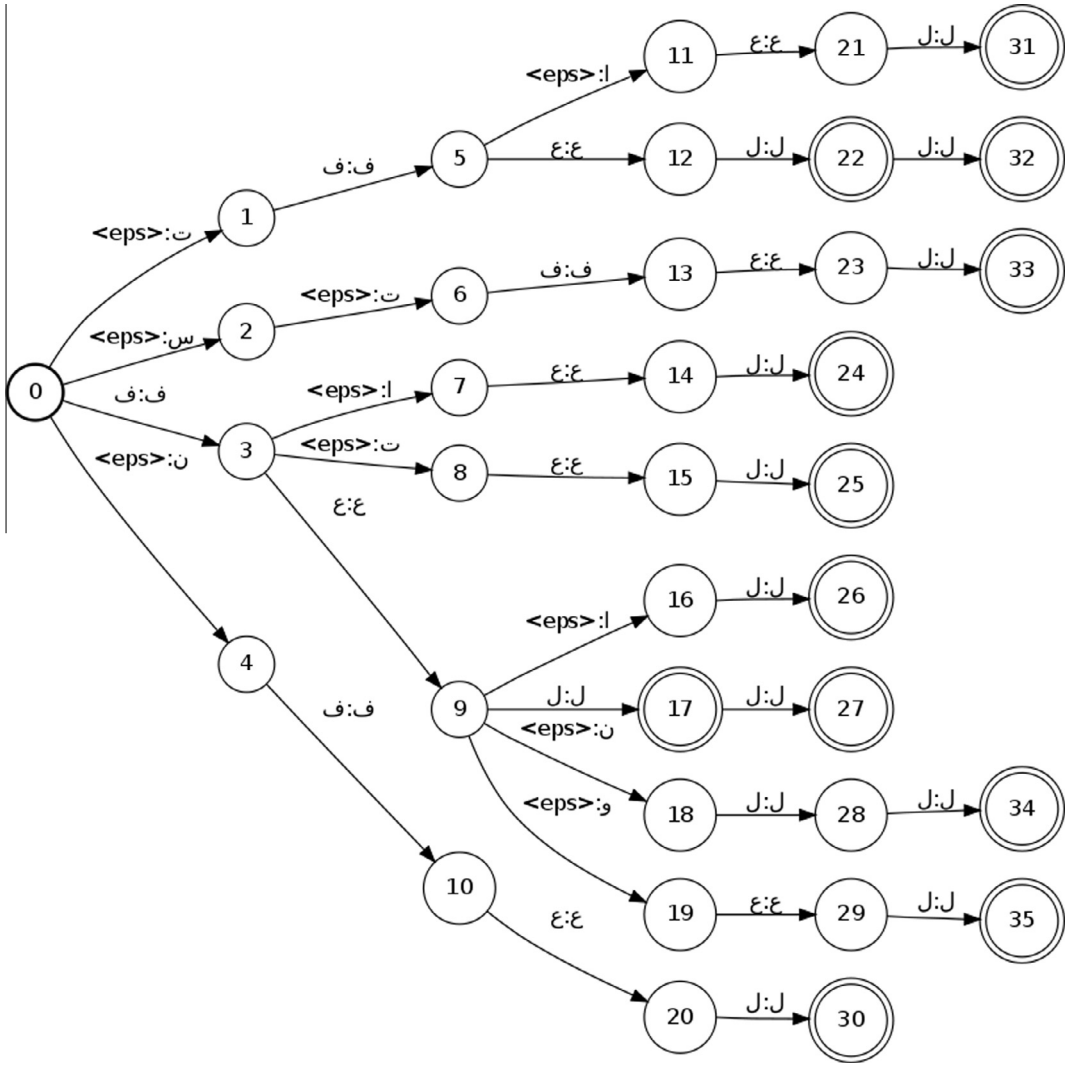

Fig. 3 Transducer of verb patterns.

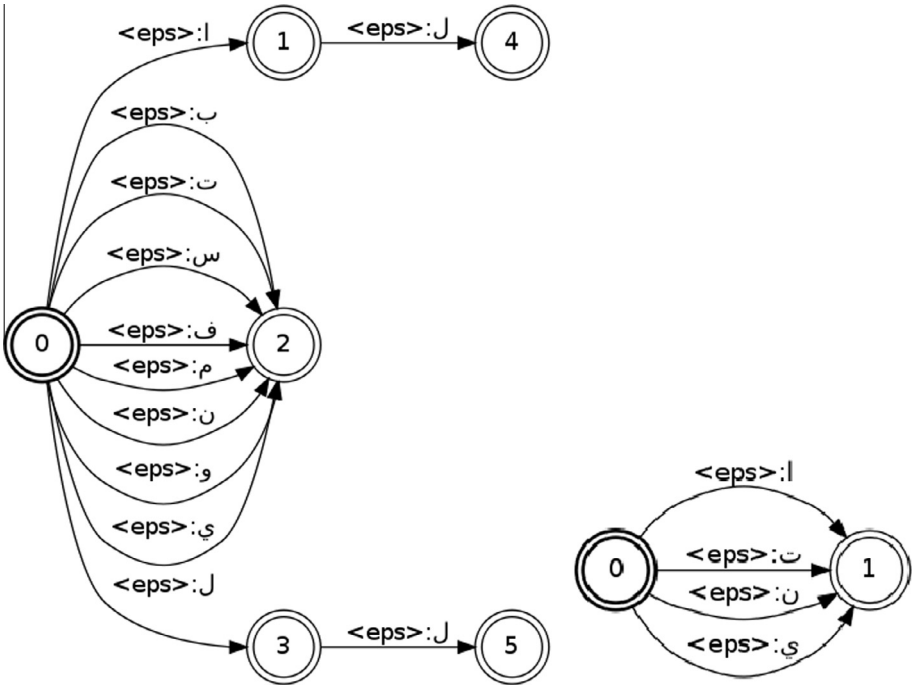

Fig. 4 Transducer of noun prefixes (top) and verb prefixes (bottom).

Steps one and three are quite similar. A transducer for each prefix (resp. suffix) is built, then, the union of these transducers is formed. The resulting transducer represents the prefixes (resp. suffixes) transducer (see Figs. 4 and 5). In step 2, we build one transducer per noun pattern. The chosen noun patterns are the most used ones in the context of MSA. Then, the union of these transducers gives the transducer of all noun patterns. Transducer of all verb patterns is built in the same manner (Fig. 3). In the fourth step, transducers from previous steps are concatenated. The final transducer is attained by the sum operation of the two transducers from the fourth step.

The resulting transducer $T_{stemmer}$ is so large that it cannot be represented graphically, it includes more than 400 states. This transducer can extract the root of any grammatically correct Arabic word, i.e, a word that fits with some Arabic measure. In addition, it can give us a morphological knowledge about the word, since we do not perform any optimization operation on this transducer (minimization or determination). We can take advantage of this information to enhance performance of classification system. However, the resulting stemmer will not be able to correctly stem Arabic words having a weak consonant root. This kind of words could be handled with the use of phonological rules, which is not supported in this work.

How to deal with non-determinism?

The composition of $T_{stemmer}$ with any given word transducer T_{word} gives a transducer which may include many paths, hence, many possible roots. Indeed, an Arabic word could match with more than one measure at the same time. Let's take the word انتصر (win). This Arabic word matches with, at least, two measures: انفعل and افتعل giving the roots تصر and نصر respectively. Thus, the use of $T_{stemmer}$ leads to a set of one or more possible stems. The correct stem belongs to the set of possible stems. To cope with this situation, root extraction transducer must be weighted. Many schemes are possible. We use a bigram window probabilities technique to affect a score to a given root. The technique is based on a statistical study of letter frequencies in the Arabic roots corpus علي حلمى موسى (1978). This corpus contains more than 10 thousand three letter roots. The score is affected to a given root by calculating the probability of letter occurrences in different positions. Let $s = c_1c_2c_3$ be a three letter root. $Score(s)$ is calculated by:

$$Score(s) = P_1(c_1, c_2) \times P_2(c_2, c_3)$$

where $P_1(c_1, c_2)$ is the probability to have the letter c_2 in the second position preceded by c_1, and $P_2(c_2, c_3)$ is the probability to have the letter c_3 in the third position preceded by c_2. Thus we consider the correct root is the one that has the best score $s_{best} : best = Arg(Max\{Score(s)\ s \in \{\text{Possibleroots}\}\})$.

One may be wondering why patterns are more important than dictionaries. This is for two reasons. First, dictionaries are generally very large and need to be maintained. Second, patterns are fixed and limited. They represent the creative energy of Arabic language. Indeed, patterns fit most current Arabic words and they are allowed to create all future words needed for civilization and scientific terminology purposes (ابن الحسن العلمي، ادريس, 2011).

6.2. Rational kernels for Arabic Text Classification

Our ATC system is structured as follows:

1. Preprocessing step.
2. Feature extraction: we feed the previous transducer ($T_{stemmer}$) by words of the documents obtained from step 1. This will produce a transducer per word. The concatenation of these transducers will represent the document in the next step.

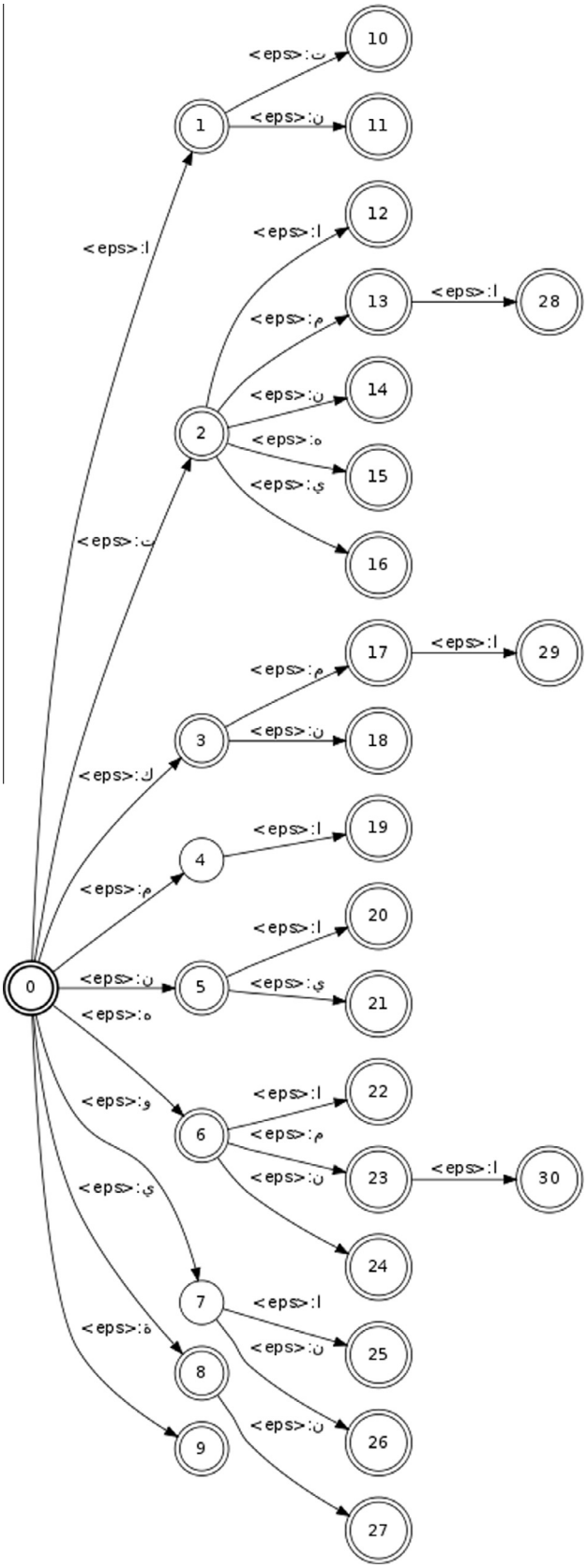

Fig. 5 Transducer of noun and verb suffixes.

3. Learning step: Rational kernels are used to measure distance between documents (Cortes et al., 2004; Cortes et al., 2007), and SVM is used to do classification.

Consider a set of documents $S = \{d_1, d_2, \ldots, d_N\}$, a document d_i consists of a sequence of words: $w_1^i w_2^i \ldots w_m^i$. Applying our root extraction transducer on each word of d_i and right concatenate results will transform this document into a finite state transducer T_{d_i}. Transducers obtained from the whole set of documents will be packaged into an archive file (far) to be treated by the learning algorithm (Fig. 9). String kernels, which are kernels defined over pairs of string, can be extended to transducers. They are typically represented by weighted finite-state transducers, to measure distance between documents. Fig. 6 shows an example of string kernels (for sake of simplicity, we take N-grams with $N = 2$ and alphabet $\Sigma = \{a, b\}$) represented by weighted finite-state transducer $T_{2-\text{grams}}$. Let T_{d_i}, T_{d_j} be two transducers representing documents d_i and d_j respectively. Similarity between these documents is calculated based on bi-gram kernel by:

$$K(d_i, d_j) = \varphi(T_{d_i} \circ T_{2-\text{grams}} \circ T_{d_j}) \quad (6)$$

where φ is a function that computes the sum of weights of all accepting paths of $(T_{d_i} \circ T_{2-\text{grams}} \circ T_{d_j})$, and $\circ$ is the composition operation (see (4)).

7. Experimental results and discussion

7.1. Experiment settings

Transducers are created and manipulated using the OpenFst library (Allauzen et al., 2007), which is an open source library for constructing, combining, optimizing, and searching weighted finite-state transducers. OpenKernel, which is a library used to create, combine and apply kernels for machine learning applications, will be used to accelerate experiments. The next batch reports the main commands of OpenFst and OpenKernel libraries used to implement our classification system.

```
1 fstcompose word.fst model.fst result.fst
2 fstconcate doc.fst result.fst doc.fst
3 farcreate data.list data.far
4 klngram -order=3 -sigma=29 data.far 3gram.kar
5 svm-train -k openkernel -K 2gram.kar cul.train
cul.train.2gram.mdl
6 svm-predict cul.test cul.train.2gram.mdl cul.
test.2gram.pred
```

To extract the root of each word in the document, we iterate on these words using the OpenFst command *fstcompose* (line 1), where *word.fst* is a linear finite state transducer with identical input and output symbols, which represents a word, and *model.fst* is our ponderated stemming transducer. The resulting transducer *result.fst* represents the best root. Resulting transducers are right concatenated to a finite state transducer (*doc.fst*), representing the entire document, using the

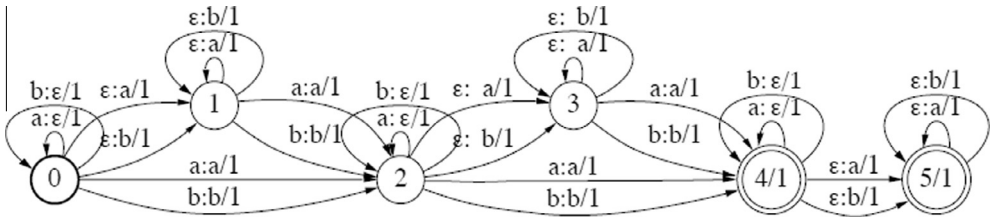

Fig. 6 Bigram kernel for alphabet $\Sigma = \{a, b\}$.

Table 4 Gold Standard details.

Corpus	# Words
Gold1	679
Gold2	844
Gold3	1000

Table 5 Accuracy of heavy stemmers.

Corpus	Khoja stemmer %	Our stemmer %	Al-Serhan stemmer %
Gold1	82.77	71.68	51.40
Gold2	85.55	74.82	49.64
Gold3	87.60	80.30	56.40
Average	85.30	75.60	52.48

OpenFst command *fstconcate* (line 2). The set of finite state transducers (FSTs) obtained so far is then packaged in a FST archive (Far) using the OpenKernel command *farcreate* (line 3), where *data.list* contains the list of all FST documents, one file per line, and *data.far* is the FST archive (Far).

Many kernels could be created using OpenKernel library. The N-gram kernels could be created using the command *klngram*. For instance, 3-gram kernels is created in line 4, where the first argument *–order* specifies the size of the N-grams, and the second argument *–sigma* specifies the alphabet size, epsilon not included (Arabic alphabet size is 28). The first parameter is the FST archive (*data.far*) and the second one (*3gram.kar*) is the resulting kernel archive.

OpenKernel library includes a plugin for the LibSVM implementation (Chang and Lin, 2011). This enables us to do training, predicting and scoring on our dataset. Training command creates a model on the training set (line 5), where the first argument -k specifies the kernel format, the second one (-K) specifies the N-gram kernel archive. The first parameter specifies a correctly classified subset of the training set, the second parameter is the resulting model. In this command, *cul.train* contains a correctly labeled sub set of training documents belonging to Cultural class. Having a model, we can use it to classify documents of the testing dataset with the command *svm-predict* (line 6), where the first parameter specifies a correctly classified subset of the testing set, the second parameter is the resulting model from the previous command. The last parameter is the result of prediction using the model.

7.2. Root extraction results

To check the performances of our root extraction technique, experiments were performed on three word collections. The first one (Gold1) is a sample taken from the Corpus of Contemporary Arabic (Sawalha, 2011). The two others (Gold2 and Gold3) are house built sets. All words of these sets were annotated by hand with the correct root. Roots have been confirmed by Arabic Language experts in Arabic Language. The

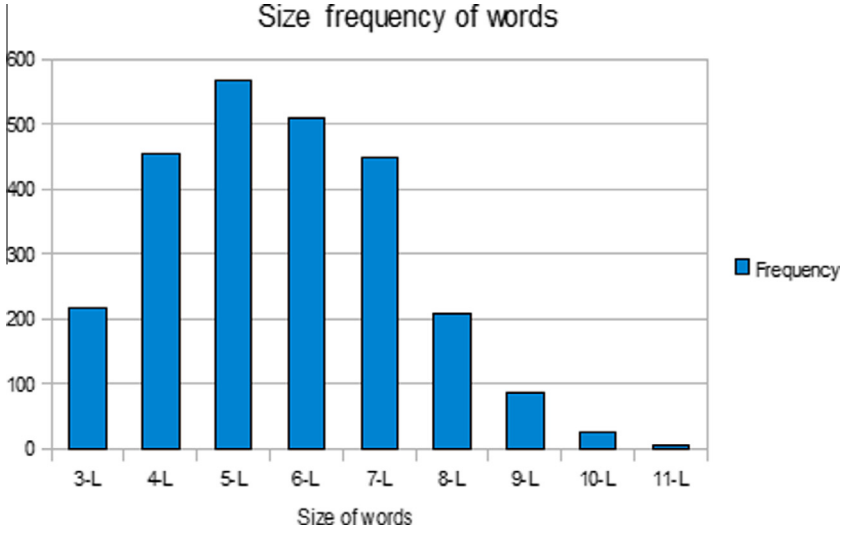

Fig. 7 Frequency distribution of words based on the length.

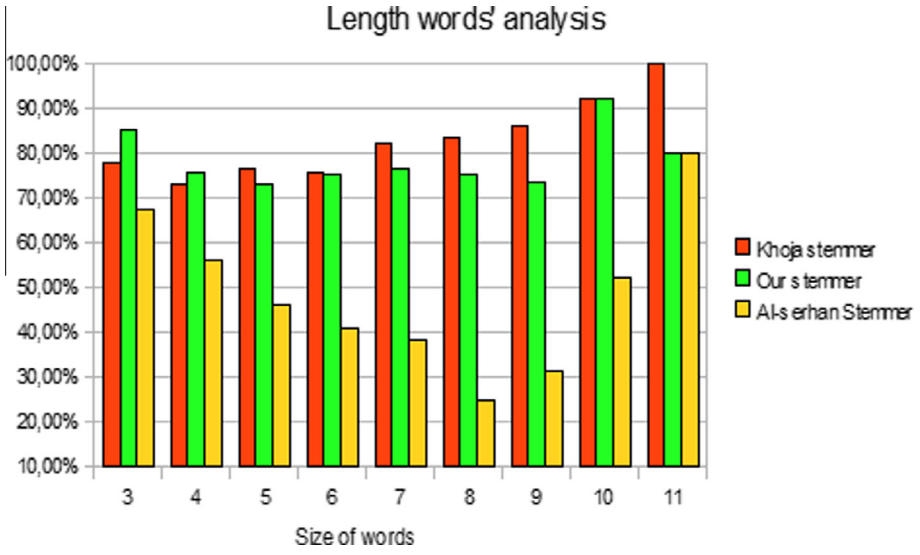

Fig. 8 Accuracy of stemmers by words' length categories.

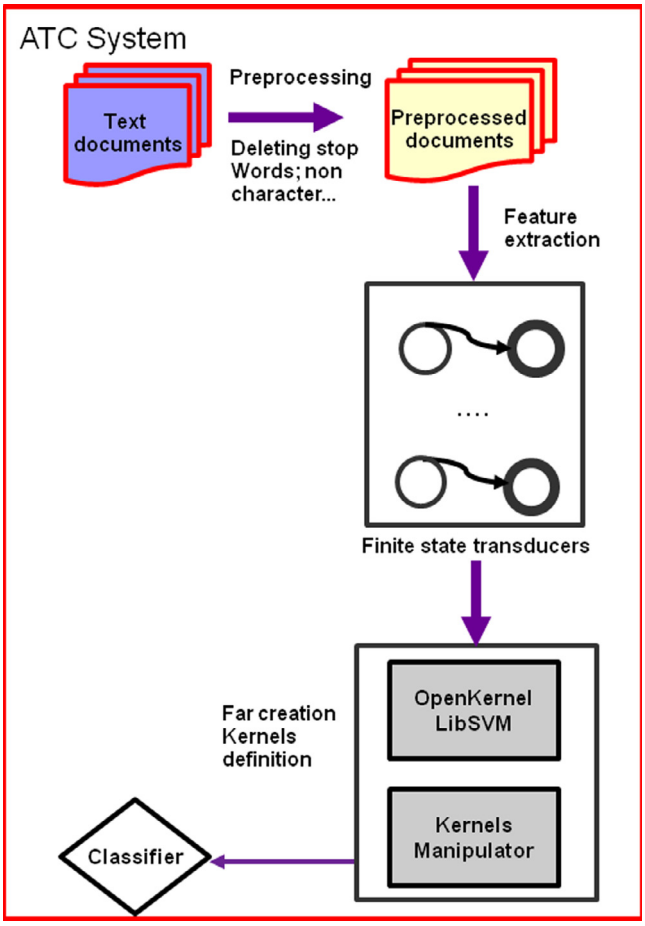

Fig. 9 Components of Arabic Text Classification System.

three sets are picked randomly from different topics, including politics, culture, sport and news. Table 4 gives an overview of these three collections. We give for each gold, the number of words (# words). Table 5 reports the accuracy of our technique on the three sets of words.

Experimental results show the effectiveness of our approach of root extraction. Results on different corpora are stable and the best score is achieved with the biggest corpus (Gold3). Results of our root extraction tool are sandwiched between Khoja and Al-Serhan stemmer results. This can be explained by the fact that Khoja's stemmer is a dictionary based tool, which makes it language-dependent. Al-Serhan stemmer is an unsupervised one. It uses a little bit information about the language. Our tool is semi-supervised. It uses a language knowledge -patterns- but only in the construction stage. Patterns are fixed and do not change.

For a deeper analysis, we report results of stemmers for each category of words based on word length. Fig. 7 gives the frequency of each category for the overall corpus. Worthwhile to note that words with length 4, 5, 6 or 7 represent 78% of the corpora size. Fig. 8 shows, in bar charts, the prediction accuracy of the three stemmers based on word length. First, we can notice that our stemmer and Khoja stemmer outperform Al-serhan stemmer for all categories. Compared to Khoja stemmer, results show that our stemmer gives better prediction accuracy for 3-L and 4-L categories. It is competitive with Khoja stemmer for 5-L, 6-L and 10-L. For the 7-L, 8-L and 9-L categories, Khoja stemmer performs better than our stemmer. This reveals that we need to analyze the erroneous roots given by our stemmer for these categories. Table 6 shows examples of incorrect results, from our stemmer, that belongs to different categories of length.

Discussion

Errors could be classified into four classes. Consider the three first examples in Table 6. This kind of errors occur when one of the radicals of the correct root is a weak consonant و، ي. Indeed, the correct root of the word (United, "المتحدة") is ("وحد"), with a weak radical at the first position ("و"). Weak root radicals change into a vowel or are deleted, depending on their vocalic environment. There are several rules with different conditions. These rules are not supported in our stemmer. The second class of errors is related to quadrilateral roots

Table 6 Examples of incorrect roots by our stemmer.

Input Arabic word (with correct root)	Number of characters	Our stemmer Output	Khoja stemmer Output	Serhan stemmer Output
(It seems, "يبدو","بدا")	4	بدو	بدا	بدو
(United, "المتحدة","وحد")	7	حده	حدد	محد
(His resignation, "استقالته","قيل")	8	سقل	قول	سقل
(The racial, "العنصري","عنصر")	7	-	عنصر	عنص
(The military, "العسكرية","عسكر")	8	-	عسكر	عكر
(Foreign, "الخارجية","خرج")	8	-	خرج	خرج
(Humanity, "الانسانية","انس")	9	-	انس	نسن
(With subscriptions, "بالاشتراكات","شرك")	11	-	شرك	بشر
(The response, "الرد","ردد")	4	لرد	ردد	لرد
(Duties, "مهام","مهم")	4	هام	هوم	هام

Table 7 SPA corpus details.

Categories	Training texts	Testing texts	Total
Culture	201	57	258
Economics	200	50	250
Social	203	55	258
Politics	200	50	250
General	205	50	255
Sports	205	50	255
	1214	312	1526

(i.e. roots with four consonants). For the words (The racial, "المتحدة") and (The military, "العسكرية"), our stemmer fails to get the correct roots because we did not consider patterns formed by quadrilateral roots. For the third class, considering examples (Foreign, "الخارجية"), (Humanity, "الانسانية") and (With subscription, "بالاشتراكات"), we can easily notice that our stemmer fails also to get the correct roots for words having compound prefixes/suffixes (i.e. more than one prefix/suffix). In the first and second examples, suffixes ("ي" and "ة" giving "ية") are used. In the third example, the prefixes ("ب" and "ال"giving "بال") are used. This could be explained by the fact that we do composition only once between prefixes, patterns and suffixes transducers (See Section 6.1). The last class of errors is related to roots with geminate radicals. The two last examples in Table 6, (The response, "الرد") and (Duties, "مهام"), illustrate the cases where the last radical is deleted when using particular templates. Our stemmer could not extract correct roots in such case.

7.3. ATC results

Experiments are performed on the Saudi Press Agency (SPA) dataset (Althubaity et al., 2008) for training and testing the ATC system. As detailed in Table 7, this dataset contains 1526 text documents belonging to one of the six categories: culture, economic, social, general, politics and sport. As mentioned earlier, stop words, non Arabic letters, symbols and digits were removed. We have used a stratified sampling with

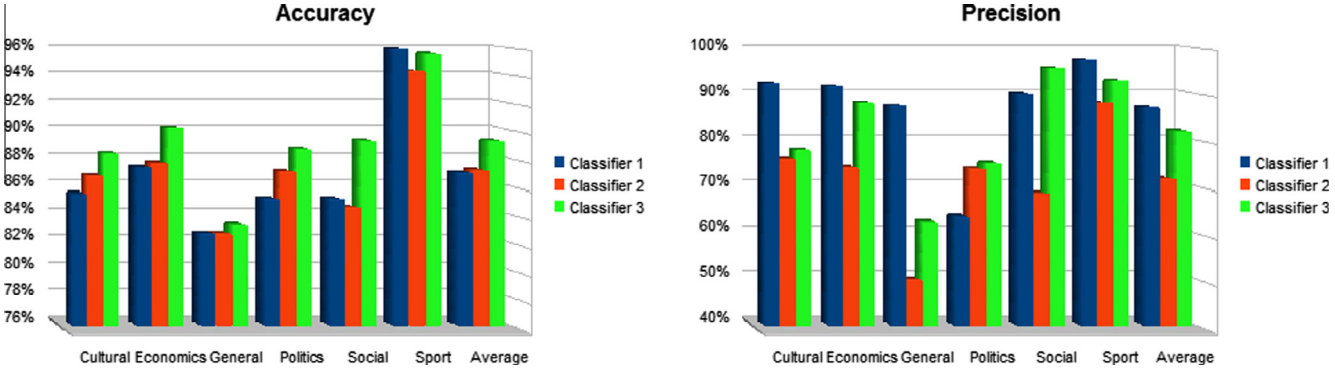

Fig. 10 Accuracy and precision of SVM Classification using 2-gram kernel.

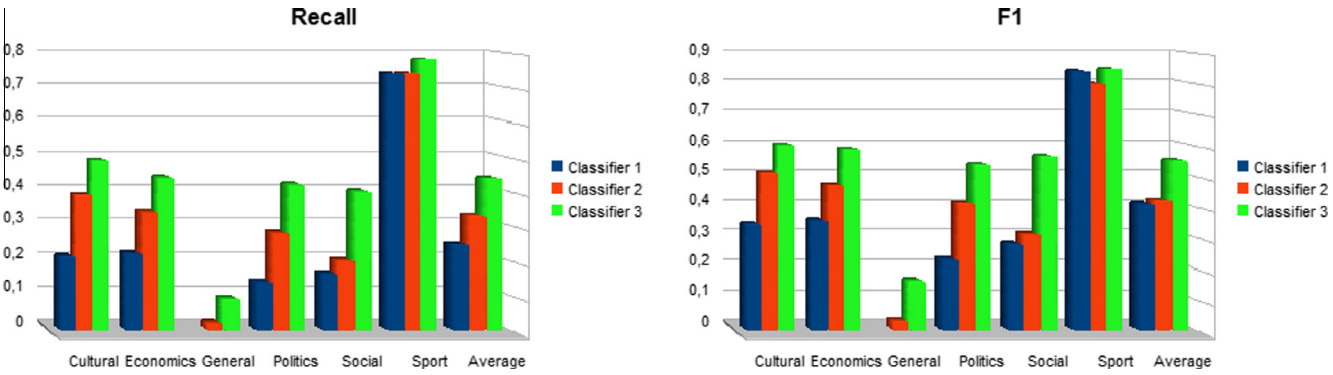

Fig. 11 Recall and F1 of SVM Classification using 2-gram kernel.

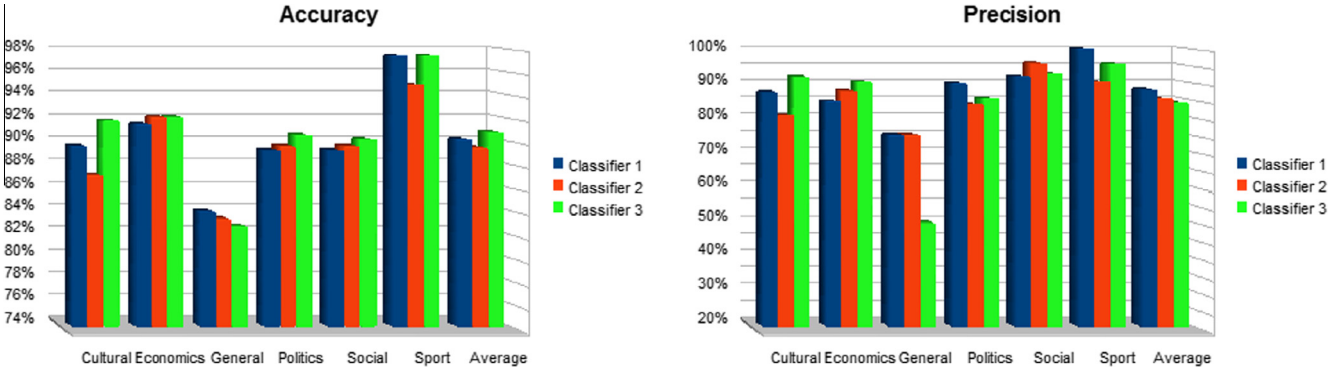

Fig. 12 Accuracy and precision of SVM Classification using 3-gram Kernel.

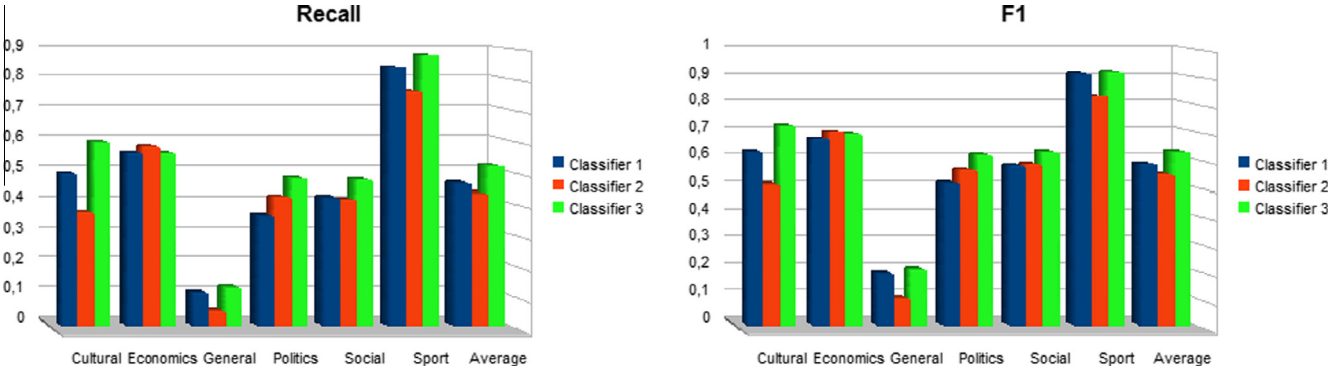

Fig. 13 Recall and F1 of SVM Classification using 3-gram Kernel.

80% of documents for training the classifier and 20% for testing. Learning is done using LibSVM implementation (Chang and Lin, 2011), included in Openkernel, with three N-gram kernels ($N = 2, 3, 4$). Since we want to show the effect of root extraction, we report results of the three classifier versions; without root extraction (Classifier 1), with Al-Serhan heavy stemmer (Classifier 2) and with our heavy stemmer (Classifier 3), in terms of accuracy, precision, recall and F1. In Figs. 10, 12 and 14, we report results in terms of accuracy and precision for the three classifiers with the three kernels (bigrams, 3-grams and 4-grams). Figs. 11, 13 and 15 give results in terms of recall and F1 for the same classifiers.

Discussion

Concerning the quality of classification, Fig. 16 shows that best results were reached with 3-gram kernel for accuracy, recall and F1 measures. This can be explained by the fact that over than 80% of Arabic words are built from 3-letter roots.

For the 3-gram kernel, we measure the effect of root extraction on classification. For most classes, root extraction enhances results in terms of accuracy, Recall and F1 (see Figs. 12 and 13). However, for precision, root extraction affects negatively performances (see Fig. 12).

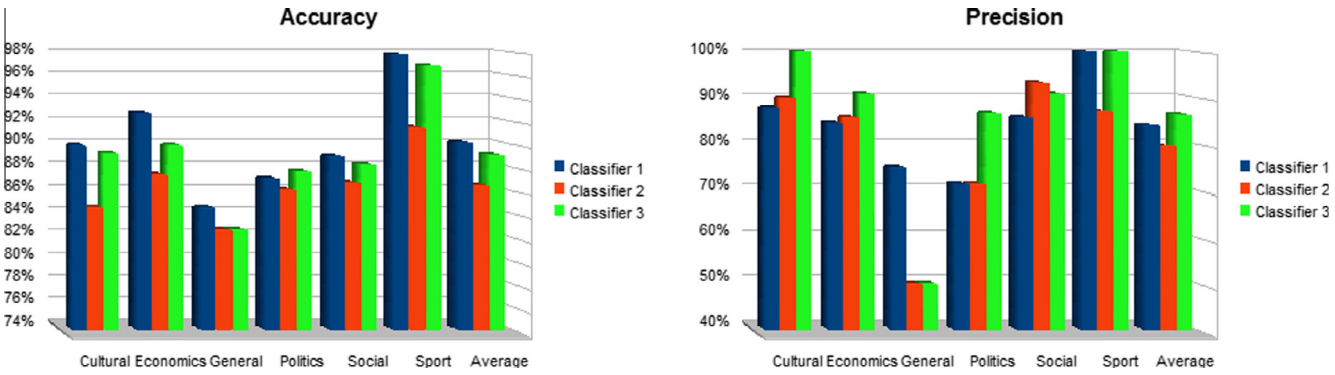

Fig. 14 Accuracy and precision of SVM Classification using 4-gram Kernel.

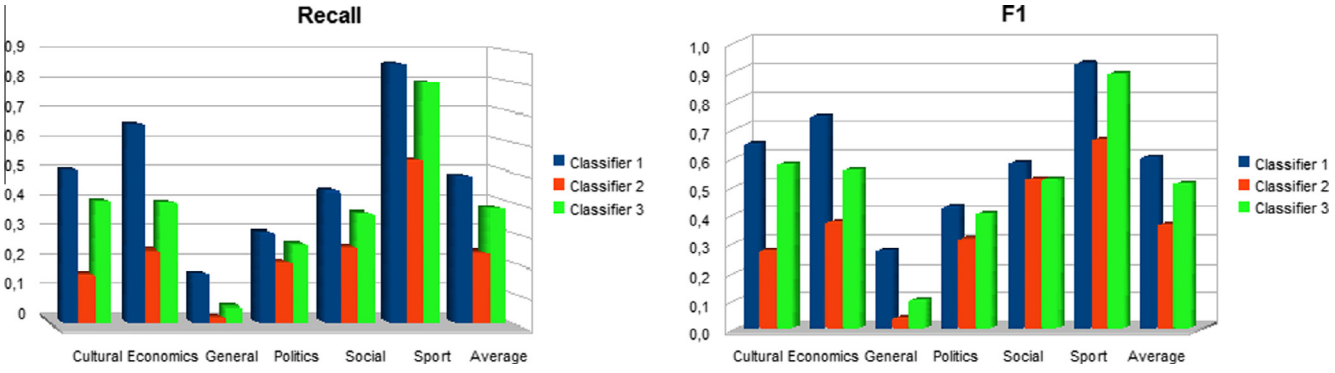

Fig. 15 Recall and F1 of SVM Classification using 4-gram Kernel.

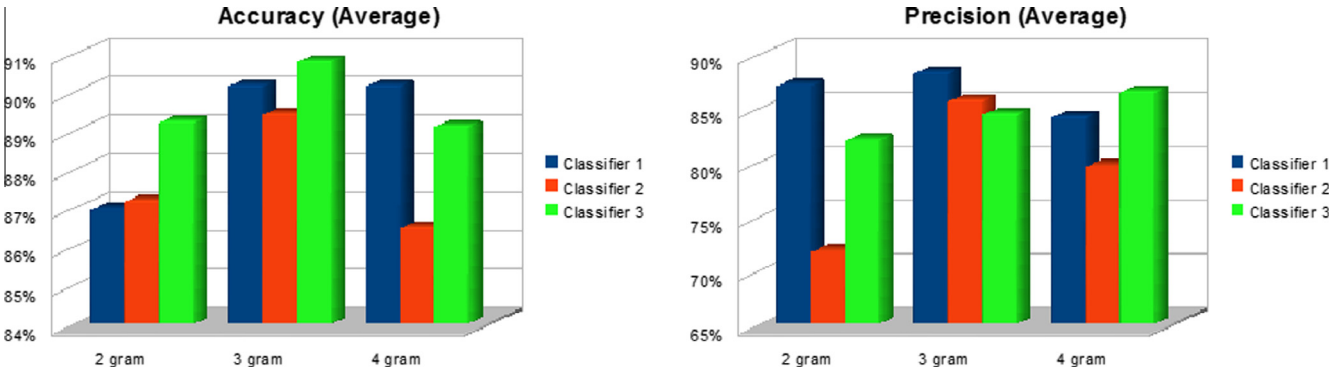

Fig. 16 Accuracy and precision averages using Bigram, 3-gram and 4-gram Kernels.

One can argue the best scores observed by sport class by the fact that it uses a specific vocabulary. Poor results are reported for the General class. This is expected given the used words in this kind of documents which are generic. At last, our classifier surpasses other classifiers in most cases.

8. Conclusion

In this work, we introduced a new framework for Arabic word root extraction and Text Classification. It is based on the use of transducers for heavy stemming, and rational kernels for measuring distance between documents. First, our root extraction method uses transducers for modeling Arabic patterns. Second, rational kernels are used to measure similarity between documents. Investigation and analysis of this framework in the context of Arabic Text Classification show that root extraction improves the quality of classifiers in terms of accuracy, recall and F1. But it slightly decreases the precision. 3-gram based classifiers reached the best results. Like that of Al-Serhan, our approach of root extraction does not rely on dictionary, and it gives better results.

In future work, we will focus on the effect of light stemming on the ATC. Other kernels, like words level grams and gappy word grams, will be investigated. For the root extraction, four-letter root will be considered and specific cases where weak consonant appears in roots will be addressed.

References

Al-Kabi, M.N., Kazakzeh, S.A., Ata, B.M.A., Al-Rababah, S.A., Alsmadi, I.M., 2015. A novel root based Arabic stemmer. J. King Saud Univ. – Comput. Inf. Sci. 27 (2), 94–103.

Al-Nashashibi, M., Neagu, D., Yaghi, A., 2010. Stemming techniques for arabic words: a comparative study. In: Computer Technology and Development (ICCTD), pp. 270–276.

Al-Serhan, H., Shalabi, R.A., Kannan, G., 2003. New approach for extracting arabic roots. In: Proceedings of The 2003 Arab Conf. on Infor. Technology, Alexandria, Egypt, pp. 42–59

Aljlayl, M., Frieder, O., 2002. On arabic search: improving the retrieval effectiveness via light stemming approach. In: ACM Eleventh Conference on Infor. and Knowledge Management, pp. 340–347.

Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., Mohri, M., 2007. OpenFst: a general and efficient weighted finite-state transducer library. In: Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007), vol. 4783 of LNCS. Springer, pp. 11–23, http://www.openfst.org.

Alsaleem, S., 2011. Automated arabic text categorization using SVM and NB. Int. Arab J. e-Technol. 2 (2), 124–128.

Althubaity, A., Almuhareb, A., Alharbi, S., Al-Rajeh, A., Khorsheed, M., 2008. KACST arabic text classification project: overview and preliminary results. In: Proceedings of The 9th IBIMA conference on Information Management in Modern Organizations.

Berstel, J., 1979. Transductions and Context-free Languages. Teubner Studienbücher, Stuttgart.

Buckwalter, T., 2004. Buckwalter Arabic Morphological Analyzer Version 2.0. Linguistic Data Consortium (LDC). University of Pennsylvania, Philadelphia, PA, USA.

Chang, C.-C., Lin, C.-J., 2011. LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. 2, 1–27.

Cortes, C., Haffner, P., Mohri, M., 2004. Rational kernels: theory and algorithms. J. Mach. Learn. Res. 5, 1035–1062.

Cortes, C., Kontorovich, L., Mohri, M., 2007. Learning languages with rational kernels. In: Proceedings of the 20th Annual Conference on Learning Theory, COLT'07. Springer-Verlag, Berlin, pp. 349–364.

Darwish, K., 2002. Building a shallow arabic morphological analyzer in one day. In: Proceedings of the ACL-02 Workshop on Computational Approaches to semitic languages. Association for Computational Linguistics, pp. 1–8.

Duwairi, R.M., 2007. Arabic text categorization. Int. Arab J. Inf. Technol. 4 (2), 125–132.

El Kourdi, M., Bensaid, A., Rachidi, T.-E., 2004. Automatic arabic document categorization based on the naive bayes algorithm. In: Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, Semitic '04, pp. 51–58.

Gharib, T., Habib, M., Fayed, Z., 2009. Arabic text classification using support vector machines. Int. J. Comput. App. 16 (4), 192–199.

Ghwanmeh, S., Kanaan, G., Al-Shalabi, R., Rabab'ah, S., 2009. Enhanced algorithm for extracting the root of arabic words. In: Computer Graphics, Imaging and Visualization, 2009. CGIV'09. Sixth International Conference on. IEEE, pp. 388–391.

Habash, N., 2010. Introduction to Arabic Natural Language Processing. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Hadi, W., Thabtah, F., ALHawari, S., Ababneh, J., 2008. Naive Bayesian and k-nearest neighbour to categorize arabic text data. In: Proceedings of the European Simulation and Modelling Conference. Le Havre, France, pp. 196–200.

Hadni, M., Ouatik, S.A., Lachkar, A., 2013. Effective arabic stemmer based hybrid approach for arabic text categorization. Int. J. Data Min. Knowl. Manage. Process (IJDKP) 3.

Harmanani, H.M., Keirouz, W., Raheel, S., 2006. A rule-based extensible stemmer for information retrieval with application to arabic. Int. Arab J. Inf. Technol. 3 (3), 265–272.

Hmeidi, I., Al-Ayyoub, M., Abdulla, N.A., Almodawar, A.A., Abooraig, R., Mahyoub, N.A., 2014. Automatic arabic text categorization: a comprehensive comparative study. J. Inf. Sci., 1–11

Kanaan, G., Al-Shalabi, R., Ababneh, M., Al-Nobani, A., 2008. Building an effective rule-based light stemmer for arabic language to improve search effectiveness. In: International Conference on Innovations in Information Technology, 2008. IIT 2008. IEEE, pp. 312–316.

Kanaan, G., Al-Shalabi, R., Ghwanmeh, S., Al-Ma'adeed, H., 2009. A comparison of text-classification techniques applied to arabic text. J. Am. Soc. Inf. Sci. Technol. 60 (9), 1836–1844.

Khoja, S., Garside, R., 1999. Stemming arabic text. Technical report, Computing Department, Lancaster University.

Khreisat, L., 2009. A machine learning approach for Arabic text classification using N-gram frequency statistics. J. Informatrics 3 (1), 72–77.

Larkey, L.S., Ballesteros, L., Connell, M.E., 2007. Light stemming for arabic information retrieval. In: Arabic Computational Morphology. Springer, pp. 221–243.

Mesleh, A., 2008. Support vector machines based arabic language text classification system: feature selection comparative study. In: Sobh, T. (Ed.), Adv. Comput. Inf. Sci. Eng.. Springer, Netherlands, pp. 11–16.

Momani, M., Faraj, J., 2007. A novel algorithm to extract tri-literal arabic roots. In: International Conference on Computer Systems and Applications, 2007. AICCSA'07. IEEE/ACS. IEEE, pp. 309–315.

Nehar, A., Ziadi, D., Cherroun, H., Guellouma, Y., 2012. An efficient stemming for Arabic Text Classification. In: International Conference on Innovations in Information Technology (IIT), pp. 328–332.

Sawalha, M., 2011. Open-source Resources and Standards for Arabic Word Structure Analysis. PhD, University of Leeds, Leeds.

Sebastiani, F., Ricerche, C.N.D., 2002. Machine learning in automated text categorization. ACM Comput. Surv. 34, 1–47.

Syiam, M., Fayed, Z., Habib, M., 2006. An intelligent system for arabic text categorization. Int. J. Intell. Comput. Inf. Sci. 6 (1), 1–19.

Taghva, K., Elkhoury, R., Coombs, J.S., 2005. Arabic stemming without a root dictionary. In: ITCC (1), pp. 152–157.

Yaseen, Q., Hmeidi, I., 2014. Extracting the roots of arabic words without removing affixes. J. Inf. Sci. 40 (3), 376–385.