View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Waseda University Repository

Enhancing Root Extractors Using Light Stemmers

Mahmoud El-Defrawy
College of Computing
and Information Technology
AAST, Alexandria, Egypt
eldefrawy.mahmoud@yahoo.com

Yasser El-Sonbaty
College of Computing
and Information Technology
AAST, Alexandria, Egypt
yasser@aast.edu

Nahla A. Belal
College of Computing
and Information Technology
AAST, Alexandria, Egypt
nahlabelal@aast.edu

Abstract

The rise of Natural Language Processing (NLP) opened new possibilities for various applications that were not applicable before. A morphological-rich language such as Arabic introduces a set of features, such as roots, that would assist the progress of NLP. Many tools were developed to capture the process of root extraction *(stemming)*. Stemmers have improved many NLP tasks without explicit knowledge about its stemming accuracy. In this paper, a study is conducted to evaluate various Arabic stemmers. The study is done as a series of comparisons using a manually annotated dataset, which shows the efficiency of Arabic stemmers, and points out potential improvements to existing stemmers. The paper also presents enhanced root extractors by using light stemmers as a preprocessing phase.

1 Introduction

Natural Languages (NLs) are the medium that allow two or more parties to communicate and interact. Linguistics have captured NLs as a set of sophisticated rules that describe the usage of a language.

The merger between linguistics and computer science began to formalize into Natural Language Processing (NLP) in mid 1950s (Nadkarni et al., 2011). Machine Translation (MT) (Hutchins, 2004) was one of the first tasks of NLP. MT takes one language as an input then predicts the output in another language. Linguistic complexity limited the development of MT, and other NLP tasks (Nadkarni et al., 2011).

There exists various NLP tasks. For example, Text Summarization (Jing and McKeown, 2000)(Nenkova, 2005), Part of Speech Tagging (POST) (Habash et al., 2009), word segmentation (Monroe et al., 2014), sentiment analysis (Oraby et al., 2013b)(Oraby et al., 2013a), and many more tasks. Each task has a specific goal which can be achieved by utilizing another NLP task. For example, sentiment analysis utilizes stemming algorithms (Oraby et al., 2013a). Many NLP tasks are a part of more complex tasks.

Text plays a central role in NLP and can be found in different forms, such as simple text or extracted rom images (Fathalla et al., 2007), this increases text resources and hence increases the need for more concise text forms.

Stemming analysis is essential for many complex tasks. Stemming is a way of reducing a given word into a concise representation while preserving most of its linguistic features (Ryding, 2005). Arabic language is highly supportive for stemming analysis. Arabic language is a derivative language, where words are constructed from basic forms called roots (Ryding, 2005). Stemming for the Arabic language is the process of deriving back the root of a given word. Some stemmers derive multiple roots for a single word, hence, various techniques were used to disambiguate multiple roots. For example, utilizing a words context was used in the technique Context-Based Arabic Stemmer , CBAS, proposed in (El-Defrawy et al., 2015). Arabic stemmers are utilized for many tasks, such as sentiment analysis (Saleh and El-Sonbaty, 2007)(Oraby et al., 2013b)(Oraby et al., 2013a), question answering (Ezzeldin et al.,

29th Pacific Asia Conference on Language, Information and Computation: Posters, pages 157 - 166
Shanghai, China, October 30 - November 1, 2015

Copyright 2015 by Mahmoud El-Defrawy, Yasser El-Sonbaty and Nahla Belal

2013), and Information Retrieval (IR) (Aljlayl and Frieder, 2002a)(Larkey et al., 2002)(Taghva et al., 2005).

In this paper, a study is conducted to analyze and compare different Arabic stemmers from different perspectives, using a manually annotated dataset. Moreover, the paper presents an enhanced version of root extractors using light stemmers for preprocessing. The paper is organized as follows, section 2 presents a concise introduction of Arabic morphology, which gives the basic intuition of Arabic stemming analysis. Section 3 discusses various techniques and strategies used to develop Arabic stemmers, followed by a detailed comparison and evaluation of existing Arabic stemmers in section 4. Finally, a conclusion is presented in section 5.

2 Background

Understanding the linguistic theory about derivational analysis provides intuitive reasoning behind different choices Arabic stemmers would do, and highlights their capabilities, strengths, and weaknesses. This section outlines the theory behind Arabic morphology and the main challenges associated with it. Arabic morphology is the study of a words construction, a new word generated from a root (Ryding, 2005). A new word is generated by changing its root. For example, the word ناجح (nāǧḥ, means ”Successful”) is generated from root ن ج ح (nūn ǧym ḥāʾ, means ”Success”) by adding ا (ʾlf) in the middle.

Arabic morphology uses a set of templates which are called patterns. Patterns are accurately defining possible changes to a root to generate a word. Pattern is a sequence of letters that captures the structure of the new word (Ryding, 2005). There are two types of letters that constitute the pattern. The first is a generic set of letters ف (fāʾ) ع (ʿyn) ل (lām) that represent a roots letters. The second type is augmented letters, which represents possible additions. Augmented letters are represented by themselves in the pattern, such as the pattern فاعل (fāʿl,means ”Actor of the verb”) which used to generate the word ناجح (nāǧḥ, means ”Successful”), the augmented letter ا (ʾlf) is represented by itself in the pattern. There are ten letters which can be used as augmented letters. It has been collected in the word سألتمونيها (sʾltmūnyhā). The root-pattern system (Ryding, 2005) starts by substituting a roots letter into the patterns generic letters, where a new word is generated. There are some cases where some additional modifications are required, commonly due to grammatical rules and letters compatibility, which is not captured neither in the root nor the pattern.

2.1 Vocalization and Mutation

Vocalization is a words letter transformed from one form to another, mostly due to grammatical or phonological rules. Vocalization defines the rules of handling weak letters, and Hamza (ء, Arabic letter) in different situations. For example, the root ق و ل (qāf wāw lām, means ”Saying”) transforms to the word قال (qāl, means ”Said”) depending on the tense of the sentence, where the weak letter و (wāw) is transformed into the weak letter ا (ʾlf). Similarity, mutation follows a similar behavior but for a different reason. For example, the word إضتراب (ʾḍtrāb, means ”Disturbance”) transforms to إضطراب (ʾḍṭrāb, means ”Disturbance”), due to phonological incompatibility between ض (ḍāḍ) and ت (tāʾ) which results ت (tāʾ) being transformed to ط (ṭāʾ). Vocalization and mutation are common challenges that face constructing Arabic stemmers.

2.2 Prefixes and Suffixes Addition

Prefixes and suffixes addition (Ryding, 2005) is a categorization to a type of augmented letters. It describes the augmented letters additions to the front or to the end of the root. Patterns can be defined to represent such additions. However, some letters can be added to the front or the end which are not part of augmented letters. For example, the word كتابك (ktābk, means ”your book”), the letter ك (kāf) at the end is added as an indication to ownership. It is not part of the word itself and it is also not part of augmented letters. Defining new patterns with prefixes and suffixes attached not only would increase the number of patterns, but it will also break the augmented letters rule, which is preferable to define in a separate process to avoid breaking the Arabic linguistic model of having ten augmented letters سألتمونيها (sʾltmūnyhā).

2.3 Stopwords

Arabic language defines a set of words that have a special meaning, such في (fy, means ”In”) and من (mn, means ”From”). Such words do not follow root-pattern substitution and commonly have some static forms (Ryding, 2005). It is part of Arabic morphology to identify such words and skip it.

2.4 Diacritics

Diacritics (Ryding, 2005) are part of Arabic words semantics. It encapsulates a set of invaluable features capturing grammatical, morphological, and phonological information. Diacritics are annotations on individual letters of a word, but it is optional (Pasha et al., 2014). Most of the Arabic readers depend on their intuition to capture such information. Many Arabic resources do not include Diacritics, such as newspapers and non-linguistic books, which creates another challenge for Stemming algorithms.

Stemming is the reverse process of derivational analysis, where for a given word a root needs to be extracted. For Arabic speakers, stemming is fairly simple even with missing information. They are capable of deducing the correct root. But, for computational devices, it is a highly complex process. Even with a complete representation of Arabic morphology, the missing information of input, such as diacritics, may lead to a set of possible results. The challenges presented above enforce scientists to make different assumptions when constructing stemmers to find an appropriate balance between correct and incorrect results.

3 Related Work

Various stemmers were developed to utilize Arabic morphological features. Each stemmer developed some mechanism to extract these features. In this section, we explore major stemmers, and their techniques.

3.1 Khoja Stemmer

Khoja stemmer (Khoja and Garside, 1999) starts by removing diacritics, punctuation, and non-characters of the input word. The word then follows a set of predefined paths, such as a decision tree. The paths are initially based on the words length then a series of prefixes and suffixes removals are defined. The resulting word gets matched with a set of predefined patterns. The matching process is highly complex, since it involves an additional set of linguistic rules. Finally, the extracted root gets validated against a set roots dictionary, then the process is terminated if the root is correct. In case the extracted root is incorrect, the stemmer continues searching for other root possibilities. The process is terminated when it reaches the first correct root, or after exhaustive search without finding a root, and it is then marked as an un-stemmed word. The number of used patterns is relatively small, indicating that Khoja stemmer is intensively dependent on prefixes and suffixes removal. Khoja stemmer is one of the closest simulations to the manual root extraction. The decisions made by Khoja stemmer are static, where it has a linguistic justification for each decision. But, it does not capture the dynamics of the language and it does not explore all linguistic possibilities. Khoja stemmer turns out to be a powerful tool (AlSughaiyer and AlKharashi, 2004). However, it does not involve other important cases, such as mutation, and the complexity of its decisions makes it challenging to update.

3.2 Sebawi Stemmer

Sebawi (Darwish, 2002) uses a different approach to build an Arabic stemmer. It utilizes a set of word-root pairs to deduce Arabic patterns, prefixes, and suffixes. The knowledge of the word and root makes it possible to segment the word into three parts, prefix, suffix, and stem (infix). The stems characters are then aligned with roots characters to formulate a pattern. The deduced patterns vary from linguistically defined patters (Darwish, 2002). For example, the word مكتوب (mktūb, means ”Written”) when aligned with its root ك ت ب (kāf tāʾ bāʾ, ”Writing”) would result م (mym) as prefix and pattern فعول (fʿūl) instead of the actual pattern مفعول (mfʿūl). Sebawi (Darwish, 2002) keeps track of prefixes, suffixes, and deduced pattern counts, which will be used in the stemming analysis. In the root extraction process, an input word is entered, the stemmer searches for possible matches in the deduced patterns, when it matches prefix, suffix, and pattern, a root is extracted. However, there is a potential that the input word would match two or more patterns; Sebawi utilizes the frequencies computed from the pattern deduction to associate a score with each possible match based on the conditional probability of prefix, suffix, and deduced pattern. Finally, the resulting roots are compared to an Arabic root dictionary to validate their existence (Darwish, 2002). The deduction of pattern removes the need for manually enumerating them. However, the deduced patterns are different from the linguistic patterns, deduced patterns introduce new patterns not previously used and in many cases deduced frequencies will not reflect the actual linguistic frequencies.

3.3 Light10 Stemmer

Light stemming is a less complex version of stemming analysis (AlSughaiyer and AlKharashi, 2004). Light stemmers are more concerned with removing the prefix and suffix of a word (AlSughaiyer and AlKharashi, 2004). Aljlayl and Frieder (2002b) construct a light stemmer to show that light stemming has a higher potential than root extraction with respect to Information Retrieval (IR). Larkey et al. (2002) conducted a similar study by constructing a set of light stemmers and comparing them with Khoja stemmer. Both types of stemming analysis showed improvement in IR (Larkey et al., 2002). However, the Light10 outperforms various stemmers in IR and it is widely used in IR (Larkey et al., 2007). Light10 is a fast and straightforward algorithm. It starts by removing punctuation, diacritics, and non-Arabic letters. It mainly normalizes the Hamza with all of its variations to ا (’lf). Then it starts by removing prefixes according to a set of constraints.

3.4 ISRI Stemmer

ISRI stemmer (Taghva et al., 2005) is another simulation for the linguistic process similar to Khoja stemmer (Khoja and Garside, 1999). It starts by normalizing the input word, removing diacritics, and non-related Arabic characters. The key in normalization is unifying the different forms of Hamza to ا (’lf) which differs from Khoja stemmer (Khoja and Garside, 1999). The normalized word then follows a series of decisions to remove possible prefixes that is three, or less characters, and then map it to a group of patterns according to its length. ISRI searches for possible matches within a groups patterns, if there is no match; it starts by removing possible suffixes. The stemming process should be stopped when the remaining length of the input word is three or less characters. Another key difference from Khoja stemmer is that ISRI does not validate roots against any type of dictionaries. ISRI is more oriented towards finding the minimal representation of an input word which can be used for information retrieval. The lack of dictionary has some side effects, such as the extracted roots are not necessarily correct, the root could be a meaningless set of characters. Roots would be unreliable for further processing, specially for linguistic based tasks.

3.5 Tashaphyne Stemmer

Tashaphyne is a light weight Arabic stemmer (Zerrouki, 2010). It is uses similar approach to ISRI stemmer. Since, it searches for the minimum representation of an Arabic word (Zerrouki, 2010). But, It is not as greedy as ISRI stemmer. It starts by removing non-related letters in the root extraction process, such as diacritics. It uses two lists of prefixes and suffixes to segment a given word. Tashaphyne provides both a light stem or a root to the input word.

3.6 ElixirFM Morphological Analyzer

ElixirFM (Smrž, 2007) is a functional morphological analyzer that utilizes syntactic features to distinguish a words sense (Smrž, 2007). Arabic Grammar and Morphology are highly correlated (Ryding, 2005). Many of the prefixes and suffixes additions have grammatical justification, which contributes to the formulation of patterns, such as pronoun additions. ElixirFM uses such correlation to improve the root extraction process; it uses Prague Arabic Dependency Treebank (PADT) (Smrz et al., 2008) to provide annotated syntactic features associated with Buckwalter stem dictionary (Buckwalter, 2002) for additional morphological knowledge. ElixirFM also handles many cases, such as mutation, using orthographical and phonological rules. The ElixirFM generates all possible roots and associates all deduced features (reasons) to distinguish word senses. It also provides additional options, such as inflecting words in various forms. ElixirFM provides various levels of analysis, such as resolving words with or without tokenization.

3.7 MADAMIRA Morphlogical Analyzer

MADAMIRA (Pasha et al., 2014) is a morphological analyzer that provides a set of valuable features including stemming. MADAMIRA (Pasha et al., 2014) is composed of two sub-tools, MADA (Habash et al., 2009) and AMIRA (Diab et al., 2007). MADA annotates the input word with every possible morphological feature, such as diacritics and lemma (Habash et al., 2009). MADA is capable of predicting 19 morphological features by using 14 distinct Support Vector Machine (SVM) and N-gram language model to predicte the other 5 features (Habash et al., 2009). AMIRA (Diab et al., 2007) includes a word Tokenizer, POST, and Base Phrase Chucker (BPC), where some tasks intersect with MADA. AMIRA uses a machine learning approach (SVM) for its predictions. AMIRA analysis is not as deep as MADA with respect to the intersected tasks which makes AMIRA relatively faster (Pasha et al., 2014). The merger extends both tools (Pasha et al., 2014). It is a dynamic tool that provides a set of valuable features to other tasks, such as Machine Translation (MT) and Named Entity Recognition (NER) (Pasha et al., 2014). MADAMIRA provides a light stemming analysis feature where it removes prefix and suffix from a word. It is a powerful tool that captures underlying data dynamics. However, it is dependent on the data quality and the learning features.

4 Stemmers Evaluation and Enhancement

Arabic stemmers have been evaluated using a standard IR test, due to the lack of existing stemmed datasets (Smirnov, 2008). In this section, the stemmers are evaluated using a manually stemmed dataset. In addition, an enhancement for the root extractors discussed in the Related Work section is obtained by using light stemmers as a preprocessing step to root extractors. The results of the enhancement are shown in tables 2, 3, and 4.

4.1 Evaluation Dataset

A set of 29 manually annotated documents were used for stemmers evaluation. The dataset is part of International Corpus of Arabic (ICA) (Alansary et al., 2007). ICA is a collection of Arabic documents obtained from various resources such as newspapers, magazines, and books (Alansary et al., 2007). ICA was collected and annotated to give a complete representation of the Arabic language to be used in Arabic NLP research (Alansary et al., 2007). The 29 documents contain $10,302$ tokens. Only $8,941$ words are Arabic words, while the remaining are tokens, such as ”/T” (beginning of a title). Only $6,323$ words have associated roots and $3,629$ unique word-root pairs of the $10,302$ tokens. Every word has a various set of features associated with it for evaluating the discussed stemmers, such as stem and root. This makes the dataset an ideal reference for evaluating the introduced stemmers. However, some features were left blank because the words do not have the associated feature, such as stopwords, as shown in Figure 1. The dataset will be used to conduct a series of comparisons to evaluate Arabic stemmers from various perspectives.

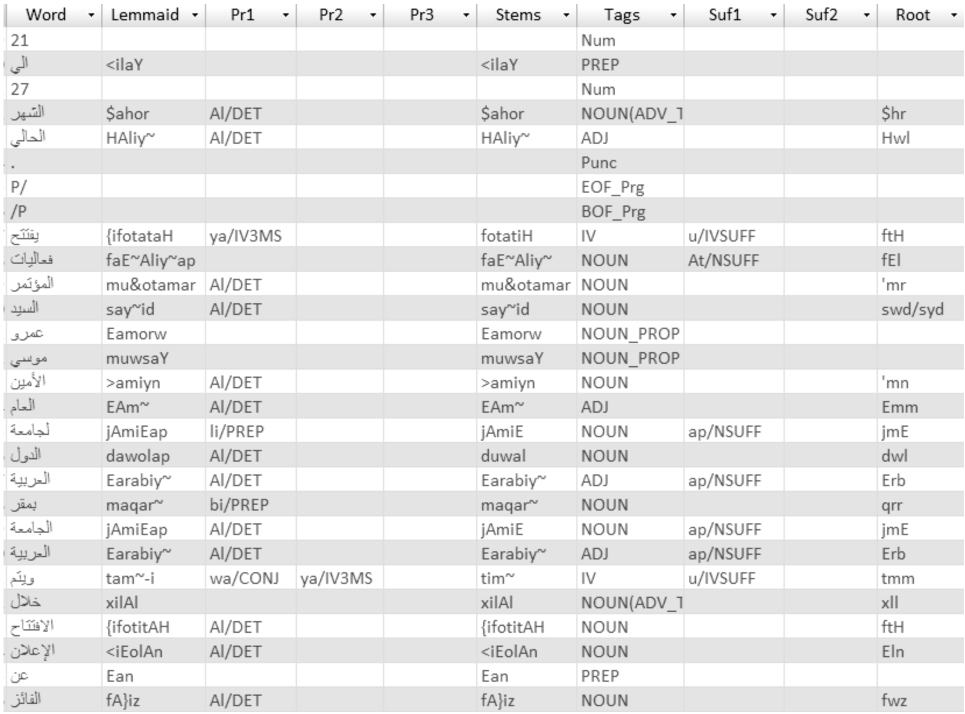
4.2 Evaluation Criteria

Arabic roots and stems provide a valuable set of characteristics that are useful for many computational tasks (Aljlayl and Frieder, 2002a)(Oraby et al., 2013a)(Ezzeldin et al., 2015). Various tasks require different perspectives such as Information Retrieval would use roots for grouping, and other may use linguistic features of roots.

The linguistic accuracy provides a representative measure for the efficiency of the stemmer in linguistic based tasks. Linguistic accuracy is computed as the ratio between the number of correctly stemmed words and the number of the input words. On the other hand, roots can be used as a word’s label, which can group linguistically similar words. Another set of measures were used to measure the macro and micro classification capabilities of the roots. The difference between macro and micro is that the size of class if reflected on the micro measure where the the macro measure treats classes equally regardless of class’s size. The following set of equations are used to provide macro and micro classification measurements (Manning et al., 2008):

4.3 Evaluation Results

The stem and root features of the evaluation data set allow to investigate the two types of stemming algorithms, light stemming and root extraction. The light stemmers that are used in the experiment are Light10



Word	Lemmaid	Pr1	Pr2	Pr3	Stems	Tags	Suf1	Suf2	Root
21						Num			
الى	<ilaY				<ilaY	PREP			
27						Num			
الشهر	$ahor	Al/DET			$ahor	NOUN(ADV_T			$hr
الحالي	HAliy~	Al/DET			HAliy~	ADJ			Hwl
.						Punc			
P/						EOF_Prg			
/P						BOF_Prg			
يفتتح	{ifotataH	ya/IV3MS			fotatiH	IV	u/IVSUFF		ftH
فعاليات	faE~Aliy~ap				faE~Aliy~	NOUN	At/NSUFF		fEl
المؤتمر	mu&otamar	Al/DET			mu&otamar	NOUN			'mr
السيد	say~id	Al/DET			say~id	NOUN			swd/syd
عمرو	Eamorw				Eamorw	NOUN_PROP			
موسي	muwsaY				muwsaY	NOUN_PROP			
الأمين	>amiyn	Al/DET			>amiyn	NOUN			'mn
العام	EAm~	Al/DET			EAm~	ADJ			Emm
لجامعة	jAmiEap	li/PREP			jAmiE	NOUN	ap/NSUFF		jmE
الدول	dawolap	Al/DET			duwal	NOUN			dwl
العربية	Earabiy~	Al/DET			Earabiy~	ADJ	ap/NSUFF		Erb
بمقر	maqar~	bi/PREP			maqar~	NOUN			qrr
الجامعة	jAmiEap	Al/DET			jAmiE	NOUN	ap/NSUFF		jmE
العربية	Earabiy~	Al/DET			Earabiy~	ADJ	ap/NSUFF		Erb
ويتم	tam~-i	wa/CONJ	ya/IV3MS		tim~	IV	u/IVSUFF		tmm
خلال	xilAl				xilAl	NOUN(ADV_T			xll
الافتتاح	{ifotitAH	Al/DET			{ifotitAH	NOUN			ftH
الإعلان	<iEolAn	Al/DET			<iEolAn	NOUN			Eln
عن	Ean				Ean	PREP			
الفائز	fA}iz	Al/DET			fA}iz	NOUN			fwz

Figure 1: Evaluation Dataset.

$$Accuracy_{\ macro} = \frac{\sum_{i=1}^{n} |X_i \cap Y_i|}{\sum_{i=1}^{n} |X_i \cup Y_i|} \qquad Accuracy_{\ micro} = \frac{1}{n} \sum_{i=1}^{n} \frac{|X_i \cap Y_i|}{|X_i \cup Y_i|} \tag{1}$$

$$Precision_{\ macro} = \frac{\sum_{i=1}^{n} |X_i \cap Y_i|}{\sum_{i=1}^{n} |X_i|} \qquad Precision_{\ micro} = \frac{1}{n} \sum_{i=1}^{n} \frac{|X_i \cap Y_i|}{|X_i|} \tag{2}$$

$$Recall_{\ macro} = \frac{\sum_{i=1}^{n} |X_i \cap Y_i|}{\sum_{i=1}^{n} |Y_i|} \qquad Recall_{\ micro} = \frac{1}{n} \sum_{i=1}^{n} \frac{|X_i \cap Y_i|}{|Y_i|} \tag{3}$$

$$F_{1\ macro} = \frac{\sum_{i=1}^{n} |X_i \cap Y_i|}{\sum_{i=1}^{n} |X_i| + |Y_i|} \qquad F_{1\ micro} = \frac{1}{n} \sum_{i=1}^{n} \frac{|X_i \cap Y_i|}{|X_i| + |Y_i|} \tag{4}$$

Where:

n is the number of candidate roots.
X is the set of candidate roots.
X_i is an indvidual candiate root.

And

Y is the set of (semantically) correct roots.
Y_i is an indvidual (semantically) correct root.

Table 1: Light Stemmer Linguistic Accuracy

Light Stemmer	linguistic Accuracy
MADAMIRA (MADA)	91.73%
Light10	47.83%

Table 2: Stemmers Lingustic Accuracy

Stemmer	linguistic Accuracy	linguistic Coverage
Khoja	72.1%	72.1%
MADA + Khoja	72.1%	72.1%
ISRI	14.2%	14.2%
MADA + ISRI	16.91%	16.91%
Tashaphyne (TASH)	30.3%	30.3%
MADA + TASH	38.23%	38.23%
ElixirFM	NA	98.15%

Table 3: Stemmers Macro Classification Statistics

Stemmer	Accuracy	Precision	Recall	F_1
Khoja	57.53%	57.53%	59.59%	58.55%
MADA + Khoja	57.53%	57.53%	59.59%	58.55%
ISRI	10.43%	10.43%	10.49%	10.46%
MADA + ISRI	15.40%	15.40%	15.60%	15.50%
TASH	25.07%	25.07%	25.15%	25.11%
MADA + TASH	41.79%	41.79%	41.85%	41.82%

Table 4: Stemmers Micro Classification Statistics

Stemmer	Accuracy	Precision	Recall	F_1
Khoja	71.42%	95.38%	73.98%	83.33%
MADA + Khoja	71.42%	95.38%	73.98%	83.33%
ISRI	14.20%	97.34%	14.25%	24.87%
MADA + ISRI	17.25%	96.59%	17.36%	29.43%
TASH	30.42%	99.45%	30.47%	46.11%
MADA + TASH	39.61%	99.86%	39.62%	56.74%

and MADAMIRA stemmers, while the root extractors stemmers are Khoja, ISRI, and Tashaphyne. The experiments also investigate combining the two types of stemming algorithms, where light stemming is used as preprocessing for root extractors. It combines MADAMIRA stemmer with Khoja, ISRI, and Tashaphyne stemmers. Only unique words in the dataset having an associated root feature are used in the test.

Table 1 shows the improvement of the light stemming algorithm MADAMIRA over Light10 stemmer. MADAMIRA gives an accuracy of 91.73% with roughly 44% accuracy improvement over Light10. Using the MADAMIRA light stemmer as a pre-processing phase before root extraction using Khoja, ISRI, and Tashaphanye stemmers improves the accuracy of root extraction.

Table 2 shows the linguistic accuracy of Khoja, ISRI, and Tashaphanye stemmers in standalone mode and when preceded by MADAMIRA preprocessing. There is a substantial difference between Khoja stemmer and the other two, ISRI and Tashaphyne, with at least 40% linguistic accuracy gap. This is due to the usage of a roots dictionary by Khoja. But when adding MADAMIRA as a preprocessing phase, there is a noticeable improvement in ISRI and Tashaphanye by roughly 2% and 8%, respectively. There is no effect of using MADAMIRA with Khoja, this is due to the robust segmentation of Khoja and the existence of dictionary validation. The ElixirFM morphological trees were not sufficient to disambiguate the generated roots. However, it provides a valuable set of features and substantial root converge which can be used for further analysis. The usage of MADAMIRA is also reflected on the classification and clustering measures. As noticed, the increase of linguistic accuracy increases related measures, namely, classification and clustering. Table 2 also shows the effectiveness of generating possible roots of ElixirFM. However, the syntactic strategy for distinguishing words' senses is not completely effective in producing only one root.

Classification has a distinctive property, that is grouping similar words. By comparing Tables 3 and 4, there is a noticeable increase in clustering measures over classification. This due to the fact that size of classes is being reflected to the micro classification measure where it is ignored with macro classification measure.

The performance of stemming algorithms can be noticeably improved by applying some minor changes, such as normalization processes. For example, changing the form of Hamaz (ء, an Arabic letter). Such changes would not affect only linguistic based task but also related non-linguistic tasks.

5 Conclusion

Stemmers are employed in various tasks, such as information retrieval (IR) (Aljlayl and Frieder, 2002a)(Larkey et al., 2002) and sentiment analysis (Oraby et al., 2013b). Stemmers achieve a noticeable improvement in related NLP tasks (Oraby et al., 2013a). However, the evaluation of stemmers does not explicitly show the stemming efficiency (Smirnov, 2008). In this paper, direct evalaution was used to study the behaviour of Arabic stemmers. The paper investigates two types of stemming algorithms, namely, light stemmers and root extractors. The light stemmers studied were MADAMIRA (Pasha et al., 2014) and Light10 (Larkey et al., 2007). And, the root extractors studied were Khoja (Khoja and Garside, 1999), ISRI (Taghva et al., 2005), and Tashaphyne (Zerrouki, 2010). The measures used to compare the stemmers were the linguistic accuracy and coverage, in addition to macro and micro classification measures. The results obtained show that the increase of linguistic accuracy increases the effectivness in other tasks(Oraby et al., 2013b)(Ezzeldin et al., 2015).

This study and IR's results (Taghva et al., 2005) show that low linguistic accuracy in stemming algorithms does not necessarily affect efficiency of a stemmer in information retrieval, possibly due to the presence frequently correct events (extracted roots). For example, ISRI stemmer has an accuracy of 14.2%, but performs efficiently and shows competitive result with Khoja in IR (Taghva et al., 2005). The study shows another set of possible improvements, which is using light stemmers as preprocessing for the root extraction task. Different studies show that light stemming has a higher potential for improving IR than root extraction (Larkey et al., 2002)(Taghva et al., 2005). Using light stemming associated with root extraction methods would build a complete hierarchical representation of Arabic words, in addition, light stemming improves the performance of other stemmers. The study conducted and the results obtained show the correlation between linguistic accuracy and other measures, the increase in linguistic accuracy increases other related mesures. The existence of multiple Arabic stemmers adds richness to the stemming analysis task. Each of the discussed stemmers has its own strengths and weaknesses, where the weaknesses could be reduced by combining multiple stemmers in effective ways.

References

Sameh Alansary, Magdy Nagi, and Noha Adly. 2007. Building an international corpus of arabic (ica): progress of compilation stage.

Mohammed Aljlayl and Ophir Frieder. 2002a. On arabic search: improving the retrieval effectiveness via a light stemming approach. pages 340–347

Mohammed Aljlayl and Ophir Frieder. 2002b. On arabic search: improving the retrieval effectiveness via a light stemming approach. pages 340–347

Imad A AlSughaiyer and Ibrahim A AlKharashi. 2004. Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology*, 55(3):189–213

Tim Buckwalter. 2002. Buckwalter Arabic morphological analyzer version 1.0.

Kareem Darwish. 2002. Building a shallow arabic morphological analyzer in one day. pages 1–8.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2007. Automated methods for processing arabic text: from tokenization to base phrase chunking. *Arabic Computational Morphology: Knowledge-based and Empirical Methods. Kluwer/Springer*.

Mahmoud El-Defrawy, Yasser El-Sonbaty, and Nahla A Belal. 2015. Cbas: Context based arabic stemmer. *International Journal on Natural Language Computing (IJNLC)*.

Ahmed Magdy Ezzeldin, Mohamed Hamed Kholief, and Yasser El-Sonbaty. 2013. Alqasim: Arabic language question answer selection in machines. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, volume 8138, pages 100–103 Springer.

Ahmed Magdy Ezzeldin, Yasser El-Sonbaty, and Mohamed Hamed Kholief. 2015. Exploring the effects of root expansion, sentence splitting and ontology on arabic answer selection. *Natural Language Processing and Cognitive Science: Proceedings 2014*, page 273

Radwa Fathalla, Yasser El-Sonbaty, and Mohamed A Ismail. 2007. Extraction of arabic words form complex color image. In *9th IEEE International Conference on Document Analysis and Recognition (ICDAR 2007)*, pages 1223–1227, Brazil, 23-26 September. IEEE.

Nizar Habash, Owen Rambow, and Ryan Roth. 2009. Mada+ tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization. In *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt*, pages 102–109.

John Hutchins. 2004. The first public demonstration of machine translation: the georgetown-ibm system, 7th january 1954.

Hongyan Jing and Kathleen R McKeown. 2000. Cut and paste based text summarization. pages 178–185.

Shereen Khoja and Roger Garside. 1999. Stemming arabic text. *Lancaster, UK, Computing Department, Lancaster University*.

Leah S Larkey, Lisa Ballesteros, and Margaret E Connell. 2002. Improving stemming for arabic information retrieval: light stemming and co-occurrence analysis. pages 275–282

Leah S Larkey, Lisa Ballesteros, and Margaret E Connell. 2007. Light stemming for arabic information retrieval. In *Arabic computational morphology*, pages 221–243 1402060459. Springer.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

Will Monroe, Spence Green, and Christopher D Manning. 2014. Word segmentation of informal arabic with domain adaptation. *ACL, Short Papers*.

Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. 2011. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551

Ani Nenkova. 2005. Automatic text summarization of newswire: Lessons learned from the document understanding conference. 5:1436–1441.

Shereen Oraby, Yasser El-Sonbaty, and Mohamad Abou El-Nasr. 2013a. Finding opinion strength using rule-based parsing for arabic sentiment analysis. In *Advances in Soft Computing and Its Applications*, volume 8266, pages 509–520

Shereen M Oraby, Yasser El-Sonbaty, and Mohamad Abou El-Nasr. 2013b. Exploring the effects of word roots for arabic sentiment analysis. In *Conference on Natural Language Processing*, Nagoya, Japan, October.

Arfath Pasha, Mohamed Al-Badrashiny, Ahmed El Kholy, Ramy Eskander, Mona Diab, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *In Proceedings of the 9th International Conference on Language Resources and Evaluation, Reykjavik, Iceland*.

Karin C 2005. *A reference grammar of modern standard Arabic*. Cambridge University Press.

Sherine Nagi Saleh and Yasser El-Sonbaty. 2007. A feature selection algorithm with redundancy reduction for text classification. In *Computer and information sciences, 2007. iscis 2007. 22nd international symposium on*, pages 1–6

Ilia Smirnov. 2008. Overview of stemming algorithms. *Mechanical Translation*.

Otakar Smrz, Viktor Bielický, Iveta Kourilová, Jakub Krácmar, Jan Hajic, and Petr Zemánek. 2008. Prague arabic dependency treebank: A word on the million words. In *Proceedings of the Workshop on Arabic and Local Languages (LREC 2008), Marrakech, Morocco*, pages 16–23.

Otakar Smrž. 2007. Elixirfm: implementation of functional arabic morphology. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, pages 1–8. Association for Computational Linguistics.

Kazem Taghva, Rania Elkhoury, and Jeffrey S Coombs. 2005. Arabic stemming without a root dictionary. pages 152–157.

Taha Zerrouki. 2010. Tashaphyne, arabic light stemmer/segment. http://tashaphyne.sourceforge.net.