Enhanced Arabic Root-Based Lemmatizer

ليميتيزر محسن للجذور العربية

Prepared by
Halah Atta

Supervised by
Dr. Ahmed Al-Hmouz

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of master's in computer science

Department of Computer Science
Faculty of Information Technology
Middle East University
Jun, 2020

Authorization

I, **Halah Atta** authorize Middle East University to provide an electronic copy of my thesis to the libraries, organizations, or bodies and institutions concerned in research and scientific studies upon request.

Name: Halah Atta.

Date: 06 / 07 / 2020.

Signature: 

Thesis Committee Decision

This is to certify that the thesis title "**Enhanced Arabic Root-Based Lemmatizer**" was successfully defended and provide on 15/6/2020.

Examination Committee Members:

Name	Position	Workplace	Signature
Dr. Ahmed Al-Hmouz	Supervisor	Middle East University	.....
Dr. Ahmad Alzou'bi	Internal Examiner	Middle East University	.....
Dr. Fayez AL-Shrouf	External Examiner	Isra University	.....

Acknowledgment

Special thanks to **my family** and everyone who encouraged me and supported me during my studying period and in all stages of my life to complete it to the fullest.

I would like to express my deepest gratitude to **Dr. Ahmed Al-Hmouz** whose support, devotion, recommendations, patience, encouragement have led me to achieve this work.

The Researcher

Halah Atta

Dedication

To my Mother Soul

To my **beloved mother** whose loved the knowledge and implant it loves in my heart, whose supported, encouraged and taught me for you I continue the road.

To my **dear father,** your presence in my life is my strength source and resolute of my success.

To my **precious family,** brothers, sisters and their family and all my beloved ones.

I dedicate this thesis

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Full Description
ARBLS	Arabic Rule-Based Light Stemmer
CA	Classical Arabic
IE	Information Extraction
IR	Information Retrieval
ISRI	Information Science Research Institute's
MSA	Modern Standard Arabic Language
NER	Name Entity Recognition
NLP	Natural Language Processing
QA	Question Answering
TC	Text Classification
TS	Text Segmentation
TSm	Text Summarization
ARLSTem	A Novel Robust Arabic Light Stemmer
ISRI	Information Science Research Institute

Enhanced Arabic Root-Based Lemmatizer

Prepared by:

Halah Atta

Supervised by:

Dr. Ahmed Al-Hmouz

Abstract

Generating meaningful information is a big task for any natural language processing application, the need to differentiate between original words and affixes in the Arabic language is important but complex in nature, stemmer and lemmatizer are the most needed components in the Arabic language processing Application. As the fundamental functionality of stemming and lemmatizing is removing what is called word morphology into a common root or base.

In this thesis, we propose a new rule-based lemmatizer, which aims to enhance the use of natural language processing applications for the Arabic language by implementing well-defined rules which result in finding the word lemma without using a dictionary. Our proposed model called "T'assel lemmatizer", is the first lemmatizer which exploit the most frequent extra letters in the word based on priorities established according to the extra letters groups.

The dataset used is a set of proverbs in the standard Arabic language contains 480 proverbs and consists of 2,493 words including 1637 unique words, the accuracy of T'assel lemmatizer was 74.11%.

Keywords: Natural language processing, stemming, Lemmatization, Arabic language, Rule-Based, Arabic morphological language.

ليميتيزر محسن للجذور العربية

اعداد: هالة عطا

اشراف: الدكتور احمد الحموز

الملخص

يحتاج أي تطبيق معالجة لغة طبيعية اعاده صياغة الكلمات وتحويلها الى بيانات ذات معنى، كما ان الحاجة إلى التمييز بين الكلمات الأصلية ومشتقاتها في اللغة العربية ضروره ولكنها معقده في طبيعتها، لإن Stemmer و lemmatizer هما أكثر المكونات المطلوبة في تطبيق معالجة اللغة العربية. فالوظيفة الأساسية Stemmer و lemmatizer هي اعادة الكلمات ومشتقاتها إلى مصدرها المجرد من الزيادات اللغوية و صياغة قاعدة مشتركة.

في هذه الأطروحة، نقترح أداة lemmatizer جديدة بالاستناد إلى قواعد الصرف في اللغة العربية، لغايات تطوير استخدام تطبيقات معالجة اللغة الطبيعية للغة العربية من خلال استخدام الية جديدة لتطبيق قواعد محددة جيدًا تؤدي إلى العثور على جذر الكلمة الصحيحة في اللغة العربية دون استخدام قاموس بالرجوع الى اولويات معده ضمن النموذج المقترح.

وان نموذجنا المقترح المسمى "T’assel lemmatizer" ، هو أول lemmatizer يستغل الأحرف الإضافية الأكثر تكرارًا في لكلمة لتحديد القواعد الأولى بالتطبيق وفقًا لمجموعات الأحرف الإضافية الأكثر تكرارًا . قاعدة البيانات المستخدمة هي مجموعة من الأمثال في اللغة العربية الفصحى تحتوي على 480 مثال عربي وتتكون من 2،493 كلمة منها 1637 كلمة فريدة ، باجراء التقييم على النموذج المقترح اظهرت النتائج ان دقة "T’assel lemmatizer" هي 74.11% .

الكلمات المفتاحية: معالجة اللغة الطبيعية ، الجذعية ، الميتة ، اللغة العربية ، قواعد صرف في اللغة العربية

Chapter One

Introductions

1.1 Overview

Stemming can be defined as the normalizing of the word into their originating word to a single form, usually called as a stem (Hammo, 2009; Chen & Gey,2002). Until now performing stemming in the Arabic language is very difficult because of the complex morphological impact of the Arabic language. This will totally change the stemming results and have an effect by the utmost impact (Marwan, 2004).

Lemmatizer is another module for natural language processing, which returns dictionary-form or the base-form of the word which is called lemma. The lemmatizer perform linguistic rules more than the stemmer in order to keep the meaning of the word. The purpose of stemming and lemmatizing is reducing the extensional forms of each word into a common root or base (Prathibha & Padma, 2015).

There are numerous attempts that have been conducted for the English Natural Language Processing (NLP). On the other hand, Arabic NLP attracts many researchers to apply those models to get significant results (Khoja & Garside, 1999; Larkey, Ballesteros & Connell, 2002; Buckwalter, 2002; Glybovets, 2015). The English language is considered a global language of communication. Recently, as the needs increased for Arabic text processing, a novel and effective stemmer or lemmatizer for the Arabic language is essential (Nwesri, 2008).

There are resources available for Arab natives to study online. These resources could be news, basic information or anything that is written in Arabic. This creates the need to process the Arabic text information in a structured and meaningful manner.

There are basically two types of Arabic language formats that are used for formal and informal communication: Classical Arabic (CA) and Modern Standard Arabic (MSA). The Arabic language is used by the literature scholars and in popular epics like the Quran, literature text, stories, poems, etc. Each region has its own dialect. As well as the Arabic language itself is a dialect for other Arab nations (Zaidan & Callison-Burch, 2014; Glybovets, 2015; Nwesri, 2008).

The Arabic language is very complex, but rich with morphological word structure that very hard to decode. So, preprocessing the Arabic text is very important but a tedious task to perform. Generally speaking, preprocessing of any language includes finding the stem or lemma of a word. The Arabic language does not have any fixed or regular structure to perform. This will lead to the change in the word when the stemmer and lemmatizer tries to find the stem or lemma (Otair, 2013; El-Sadany & Hashish, 1989; Aljlayl & Frieder, 2002; Khoja, Garside & Gerry, 2001).

Various models are built to overcome the stemming problems, such as morphological analysis, light stemming, N-grams, statistical-based stemming, parallel corpora (collections). The existing techniques of Arabic language stemmer have low efficiency in term of finding the base-root (Mustafa, Eldeen, Bani-Ahmad & Elfaki, 2017; Hadni, Ouatik & Lachkar, 2013).

The purpose of stemming and lemmatizing process is to find and match the words with its identical stem or lemma, even if it is invalid, it will match it. The very basic objective of the stemming (lemmatization) process is to derive the stem from the word by removing the postfix and prefix (Khoja & Garside, 1999; Larkey, Ballesteros & Connell, 2002; Buckwalter, 2002; Taghva, Elkhoury & Coombs, 2005;Sembok & Ata, 2013;Al-Omari, Abuata & Al-Kabi, 2013).

Most of the existing studies used a dictionary to help in the stemming process, and all of them used the rules without a specific order. This work proposes a Rule-based model to use the Arabic morphological to define a group of rules, the rules will be executed based on groups of priorities of the extra letters, and without using the dictionary.

This chapter delivers a general introduction about the Arabic stemmer in section 1.1. In section 1.2 is the research problems, aims and objectives in section 1.3, in section 1.4 the contribution and the significance of the research, in section 1.5 is the study scope and limitations and the thesis outline in section 1.6.

1.2 Problem Statement

Nowadays, the internet is a wide source to extract the information from in any format. Almost anything could be found on the internet from a basic dataset to a complex database. Generating meaningful information is a big task for any NLP application. The need for any NLP based applications require having a very accurate lemmatizer. The NLP will help to automate the process of segmentation and stemming etc. Name Entity Recognition (NER) is one of the concepts which require a base-form word to be extracted from (Naili, Chaibi & Ghezala, 2019).

To create such a tool, the need to differentiate between the original words and affixes in the Arabic language is important but complex in nature. The challenge of determining the original Arabic word in one word and affix in the other word arises very frequently (Alansary, Nagi & Adly, 2007; Al-Fedaghi & Al-Anzi, 1989; Al-Harbi et al., 2008).

Numerous algorithms are developed for stemming and Lemmatizing. Some are used for morphological analysis mechanisms that can achieve morphologically related forms, combined under the same stem or lemma (Fautsch & Savoy, 2009; Kammoun, Belguith & Hamadou, 2010). Three stemming approaches are defined as the stem base stemmer, the light stemmer, and the statistical stemmer (Syiam, Fayed & Habib, 2006; Dahab, Ibrahim & Al-Mutawa, 2015).

The recent big growth of the Arabic internet content and the evolution of technology interactions has raised up the need for more effective lemmatization techniques for the Arabic language, this motived us to find an effective method for determine how to use the Arabic language rules to create an effective lemmatizer.

1.3 Aim and Objectives

As the number of researches conducted on the Arabic language for stemming and lemmatizing increases, there is no sign of increased accuracy. This will not produce any significant impact in term of any NLP based applications (Naili, Chaibi & Ghezala, 2019).

The proposed model aims to enhance the Arabic language stemmers and lemmatizer by implementing the rules that fulfill the morphological nature of the Arabic language word. As there are no standard sets of rules existing, each lemma should be

present in the Arabic dictionary. Whereas the stem does not need to be a valid dictionary word. To reach this aim, the following objectives are posed.

1. To improve the extracted rules from the Arabic language to increases the model performance.
2. To provide a method for arranging the implementation of those rules for the lemmatizing process.

1.4 Contribution and Significance of Research

The Arabic language stemming and lemmatizing is one of the major issues for the researchers. Yet there is a remarkable research gap while processing the documents. The purpose of the research is to improve the extracted rules from the Arabic language and find a method for arranging the implementation of those rules to increases the accuracy of a stemmer (lemmatizer) for better extraction of stems (lemma). We implement the model without using the dictionary.

By improving the Rules and the methods of executed those rules are in order of priority of the extra letters groups. Helped in reducing the ambiguity in the Arabic language and helped in solving some of irregular plurals problems.

1.5 Study Scope and Limitations

With the enhancement in the stemmer's development for the information extraction with utmost accuracy, the need for accuracy originates. The scope of the research is to develop a stemmer (lemmatizer) technique that resolves the problems of the research gap. The stemming and Lemmatizing process is used in the NLP based application such as

Information Retrieval (IR), Question Answering (QA), Information Extraction (IE), Text Segmentation (TS), Text Classification (TC), Text Summarization (TSm), Name Entity Recognition (NER) and Topic Identification (Beesley, 1996; Feldman & Sanger, 2007).

As the proposed model is an Arabic Lemmatizer (stemmer), this work is limited only for the text written in the Arabic language.

1.6 Thesis Outline

The rest of this thesis is organized as follows:

Chapter Two discusses the literature review on Arabic Stemming attempts and its shortcomings.

Chapter Three discusses in detail the proposed model.

Chapter Four discusses the results obtained from the proposed model.

Chapter Five presents the conclusion and future work.

Chapter Two

Background and Literature Review

2.1 Background

Stemming and lemmatizing techniques are applied as a pre-processing step to derive the original word from its multiple forms. Understanding the Arabic language stemming and lemmatizing problem, multiple kinds of research have been conducted.

As Arabic is one of the semitic languages, it has complex morphology, which is different from the east Asian and the European languages in term of morphology and semantical (Otair, 2013).

Arabic language alphabets consist of 28 letters that start from right to left. The struggle in the Arabic alphabets is in the form of the letter that will be different according to its linking to the previous or the following letters within the word. For example, the letter Sheen (ش) at the beginning of the word looks like (شـ)and in the middle, it looks like(ـشـ) and at the end of the word (ـش) (Hasanuzzaman, 2013).

There are three long vowels YAA(ي), WAW(و) and ALIF(ا)that take different forms based on their location in a word. There is shorthand symbols for such words and presents as (Hamza ء) or short vowels (damahُ, fathaَ, kasraِ) and Shadda (ّ) means the extra letter "repeat the letter". Although, they aid in clearing the ambiguity in Arabic which also gives a word the wanted meaning in short vowels (diacritics). Diacritics are

usually ignored by modern writers and are expected to understand the missing diacritics based on the knowledge of the language (Hasanuzzaman, 2013).

Arabic derivational system consists of independent roots of 10,000 words because it considered as a root-based language. Arabic word roots consist of bare-root verb form that is tri-literals, quad-literals or pent-literals in form (Mustafa, 2013).

By using well-known morphological patterns (Grammatically) the words are made by expanding the root with affixes (Alserhan & Ayesh, 2006). In the Arabic language, the affixes are prefixes that are attached at the beginning of the words. Postfixes (suffixes) are added at the end. Infixes (morphemes) are found in the middle of the word. Affixes can be categorized into three major types: Prefixes, Suffixes and Postfixes that can be connected to words (Froud, Bensliman, Lachkar & Ouatik, 2010).

The Arabic language has a very difficult type of irregular plurals that are recognized as broken plurals. An Arabic language plural does not follow the morphological rules. Broken plurals establish 10% of Arabic texts and 41% of plurals (Mustafa, 2013).

These complexities affect the preprocessing of Arabic texts and make it very different from the preprocessing of the other languages. With that said, it affects the stemming and lemmatizing process particularly and makes it harder and difficult to implement. Many efforts have been done in this regard, even though some improvements are still needed. So, the proposed model tries to state some rules that satisfy the nature of the morphology of the Arabic language along with including each lemma present in the Arabic dictionary.

As discussed before, the purpose of stemming and lemmatizing is the same as removing the extensional forms of each word into a common root or base. The difference is that lemmatizing returns dictionary-form of the word and apply linguistic rules more than the stemmer, which tries to keep the meaning of the word.

2.2 Literature Review

Khoja et al. (1999) proposed Arabic word root extraction method by removing the prefixes and suffixes, the root words is extracted. This will generate the problem of over-stemming when the model by mistake detects some semantically wrong words for a single root. There is also the miss-stemming problem when the model tries to remove the affixes that are actually the part of words.

Buckwalter et al. (2002) proposed a novel morphological analyzer which generates every possible stem for the text. The proposed model manually constructed the table based on three groups' i.e. prefix, suffix, possible stem and also generates the valid blend of every pair consisted of prefix/ suffix and stem, prefix/stem pairs and stem/suffix pair. These pairs are recorded in the truth table form that will help in the root detection. Basically, the proposed model divides the word into three possible parts i.e. prefix, stem, and suffix.

After a generation of tables, the matching process is performed to the probable combination of tables for prefix, stem, and suffix. This process will generate the input word. It is a very tedious task for the stemmer to detect the root if is not in the table. The proposed model doesn't deal with irregular plurals.

Larkey et al. (2002) proposed a novel stemming technique in which some selected prefixes and suffixes are truncated in a word that will help to generate the stem. The

proposed model also removes the sentences that are found in prefix and suffix very often. The proposed model has its own merits and demerits i.e. irregular plural and also without the prior knowledge of linguistic rules it removes the affixes.

Taghva et al (2005) proposed a stemmer named “the Information Science Research Institute’s (ISRI) stemmer”, that uses the root as the base so that it will use it with the Khoja stemmer (1999) approach. The proposed model does not own a root-dictionary for lookup. If the model does not find the root of the word, then the ISRI stemmer will normalize that particular word and return the word into their normalized form. The ISRI stemmer removes the affixes and diacritical marks and also can’t deal with irregular plural. There are very high chances of increased word ambiguity in the root-based approach because of multiple meanings for the same word that is generated from the identical root. And to improve this kind of root-based stemmer’s drawbacks, light stemming is used in the Arabic language.

Harmanani, H. M et al. (2005) proposed a novel approach for stemming in Arabic language based on language-dependent rules and a rule engine is used to interpret it. The proposed model at first determines the word types i.e. it's a verb or a noun. This will help the engine to check if the word rhymes with the existed pattern for stem extraction. The proposed model discusses the effectiveness of the rule-based stemmer on indexing and stated that the proposed model enhances the indexing speed by 75% and after indexing 19 documents precision can reach 100%.

Larkey et al. (2007) concluded that there are numerous stemmers developed but with the same number of stems after modifying the small chunk of words like letter “و” from light2, light3, and light8 and light10. It does not deal with irregular plural, infixes or

patterns and removes the word if the remainder is equal to or more than 3 without any knowledge of linguistic functions (Larkey, Ballesteros & Connell, 2002).

Al-Omari et al. (2013) proposed a unique model that uses the mathematical basis and letter to generate the relation between them so that the root and the stems can be found. The proposed model does not use any pattern for the root extraction or stem extraction in an Arabic word. The proposed model applied the mathematical operations on the Arabic text to match and produce a relation between them.

Sembok & Ata (2013) proposed a stemmer that uses the rule-based stemmer which categorizes the rules into basically three categories i.e. prefixes, suffixes, and recoding. Another approach used by this proposed model is generating the possible root that is generated for a given word. To confirm the accuracy of the proposed stemmer, a root word dictionary is used. This process led to a pattern generator and intensification module that handles multiple letters. The recoding module analyses the very performance of tests that are based on recall precision measurement. This is done between this stemmer and Al-Omari's algorithm (Al-Omari, Abuata & Al-Kabi, 2013).

Kreaa et al. (2014) proposed a stemmer which is an integration of two different approaches (light Stemmer and Lookup Tables). This model removes the word affixes by using the diacritical marks and set of affixes. This will overcome the problem of over-stemming (Khoja & Garside, 1999;Larkey, Ballesteros & Connell, 2002; Larkey, Ballesteros & Connell, 2007;Buckwalter, 2002;Taghva, Elkhoury & Coombs, 2005).

Al-Kabi, M. N et al. (2015) proposed a new Arabic stemmer that has both light and heavy stemmer features. The proposed model uses the three-phase strategy for the experimental design. The first phase is to remove the Arabic affixes. The second phase is used to identify the pattern of the verb in each Arabic word. While the third phase is used to enhance the proposed model for Arabic roots identification. Experimental analysis shows 75.03% accuracy of the proposed algorithm comparative to another Arabic stemmer that used the same dataset. The proposed model uses the Arabic WordNet tables (Black et al.,2006) to ensure the accuracy of the stems that are extracted by the proposed algorithm.

Al-Lahhamet et al. (2017) proposed the CondLight stemmer which is based on the Light10 stemmer. The proposed model added the extra prefixes and suffixes in the existing stemmer used by the light10 stemmer (Larkey, 2002). The list of added prefix in the Light10 stemmer is (ف ,ي ,س ,ل ,ت ,ب) and the list of suffixes are (هن, هم, وا). The proposed model uses the conditional sets to remove them. These conditions are generated from properties of morphological of the Arabic text. The proposed model shows the 5% retrieval as compared to the existing Light10 stemmer model. This also concluded that the addition of some condition in a stemmer increases their accuracy.

Al-Lahham et al. (2018) and M Mustafa et al. (2019) attempted to improve the accuracy of stemmers by two proposed models. The first is a light stemmer also called Extended-Light focuses on improving the quality by removing the problem of under-stemming, based on Light10 stemmer. The second is linguistic stemmer which states that the words that are available and used in the Arabic language are rhythmic in some patterns

according to the rules. ExtendedS and LightStem shows the improved effectiveness consistently as compared to the light10.

Awwad et al. (2019) proposed a stemmer algorithm for the Arabic language that consists of three stages. The first stage constructed a hash table for each affix and the stop word. The second stage uses the three-stripping function applied on affixes. The final stage consists of 24 recursive calls performed by the stripping engine for producing the stem results. The result generated by the proposed stemmer has 24 potential stems for the chosen word. This will solve the previous stemmer problems and improve the model accuracy for the Arabic language.

Zamani et al. (2019) proposed a model that was used to calculate the stemming on Quran with a 95% accuracy. The proposed stemmer has some errors while producing the root words if checked manually. To overcome this problem, al-Quran dictionary is used to justify and analyze the results of each stemming that is conducted on Quran by Khoja stemmer.

2.3 Summary

A literature review of the existing attempts concludes that to extract the stems or lemmas in a text or sentence, there are no standard sets of rules existing.

In the NLP for Arabic language, numbers of algorithms are proposed for implementing stemming and lemmatizing proposed to get the stem or lemma of a word. Table 2.1 summarizes the most novel, effective and relevant to our proposed model. It

covers the algorithms with their findings and short-comings and the dataset used to perform the experimental analysis.

Table 2.1.stemmer for Arabic language summarizes.

Study	Approach	Data set	Technique	Main finding	Short come
Khoja, & Garside, 1999	Root - based		Removing the longest prefixes and suffixes, and then try to determine the root using a dictionary of root words. "for each word stemming it makes two comparing for each word stemming; one against a set of predefined patterns and the other to check that the extracted root is in the dictionary or not".	when the letters deleted during the derivational process of words, it has the skill to detect it	there is over-stemming when the stemmer group some semantically different words into a lone root and there is miss-stemming when the algorithm removes some affixes that are parts of words, creates ambiguity in IR and also the root dictionary need to be upkeeps
Buckwalter, 2002	Root - based	The data contains of three Arabic-English lexicon files: prefixes, suffixes, and stems	Manually constructed dictionaries, they produce dictionaries listing the Arabic suffixes and prefixes and stems/roots.	The algorithm divided the input word into three sub-strings (potential prefix ,stems and suffix) with all its possibilities. it produced according to the pre-constructed tables, then a matching it for each possible combination of prefix, stem, and suffix that could produce the input word.	Could not stem the word if it is not including in the stem table and It could not handle irregular plural.

Study	Approach	Data set	Technique	Main finding	Short come
Larkey, Ballesteros, & Connell, 2002	light stemming	Use standard TREC data to evaluate their performance for IR in Arabic	choice some prefixes and suffixes to be shortened from the words and generate the stems, it also tries to remove the strings that were often found as prefixes or suffixes would be found as affixes at the start or end of a word.	display that light stemming has a higher potential than root extraction for Information Retrieval (IR).	Its disadvantages that it does not work with irregular plural and it eliminates the affixes without any previous knowing in linguistic rules.
Taghva, Elkhoury, & Coombs, 2005	Root - based	document retrieval tasks are done on the Arabic Trec-2001 collection.	The algorithm work similar to the Khoja stemmer to extracts the roots but without a root dictionary	In the ISRI stemmer the case of a word cannot be rooted its returns a normalized form	It cannot deal with broken plurals and generates ambiguity in IR The lack of a dictionary has some effects; the root could be a meaningless set of characters.
Larkey, Ballesteros, & Connell, 2007	light stemming	Use standard TREC data to evaluate their performance for IR in Arabic.	the same steps with small changing like Remove the letter "و" from light2, light3, and light8 And light10 if the rest of the word is three or more letters long it remove any confirmed letters if this leaves two or more letters.	Found that light stemming did not certainly have been able to perform better using morphological analysis.	It does not work with irregular plural and without any previous knowing in linguistic rules it removes the affixes also It could not handle the infixes or patterns.

Study	Approach	Data set	Technique	Main finding	Short come
Al-Omari, Abuata, & Al-Kabi, 2013	light stemming	tests on an Arabic dataset consisting of 6,412 Arabic	Rule-Based Light Stemmer , build on some relations between letters and mathematical rules (ARBLS).	They did a comparison of the effectiveness of the new stemmer with the Khoja stemmer. The results shows that the new stemmer is more effective than the others	Incapability to handle Arabic authoritative words, which represent a command to a second person, especially those imperative consisting of two letters.
Sembok & Ata. 2013	a rule-based approach	consists of 6236 documents of the Quran collection, the number of queries used in the experiment is 36.	The rules are classified into groups: prefixes, suffixes, and recoding. Also, templates of roots are used to generate possible roots for the word. And a dictionary of roots used to confirm the rightness of the root nominate	The analysis of performance-tested based on recall-precision measurement between this stemmer and Al-Omari's algorithm and The algorithm performance better than Al-Omari's algorithm.	Their analysis proposes that most of the errors are due to the order in using the rules when applied in the stemming.
Kreaa, Abdel Hamid, Ahmad & KassemKabalan, 2014	a hybrid approach that integrates two methods: Light Stemmer and Looks in Tables.	uses Arabic WordNet tables to confirm the accuracy, using the Arabic Trec 2001 collection.	It removes the affixes letter by letter depending on diacritical marks and affix sets to try to beat the Over Stemming problem. And the broken plural.	The algorithm uses a set of rules to define if a clear series of the letter is part of the original word or not, with that it helps to solve some ambiguity problem	low accuracy on handling the broken plural forms

Study	Approach	Data set	Technique	Main finding	Short come
Al-Lahham, Matarneh & Hasan, 2018	light stemming	TREC2002 collection.	is based on Light10 stemmer, they add extra prefixes to the list in Light10 stemmer, which is: (س , ل , ت, ب ف , ي ,) and new suffixes (هن , هم, وا), they use set of conditions to remove these prefixes and suffixes.	After testing the proposes conditional light stemmer retrieval effectiveness, it increases about 5% enhancement of retrieval	Did not test the suggest stemmer against the correctness of the output words after affixes removal.
Mustafa, Aldeen, Zidan, Ahmed & Eltigani, 2019	proposes two different stemmers The first is a light stemmer the second is a linguistic stemmer.	TREC 2001 Arabic corpus	It was driven from the understanding of Arabic morphology, which is that words in Arabic are frequently rhymed into diverse patterns given to some different rules.	the proposed two stemmers (ExtendedS and LingStem) are steadily do better than light 10	but they could not solve the problem with a word when they are semantically related to each other's.
Awwad, 2019	approach using the affixes removal.	with 2000 Arabic words, but with four diverse subject documents	They propose a stemmer generally involves of three phases, first a hash table is produce for each affix type and stop word, Then the second phase, three affix stripping functions are defined. Finally, 24 recursive calls are prepared in the stripping engine to produce stem results.	They found that the most effective stripping orders are when begin by removing prefixes then removing infixes, and finally removing suffixes, with result 86% of correctness.	low accuracy on handling the broken plural forms

Chapter Three

Methodology and the Proposed Model

3.1 Introduction

In this chapter we will present the phases of our proposed lemmatizer, Section 3.2 introduce the main phases of the T'assel lemmatizer model and Section 3.3 illustrates the model and shows an example of the process, then summary in section 3.4.

3.2 Methodology

The proposed model uses the new set of rules for stemming and lemmatizing so that the model will overcome the existing problem of ambiguity related to existed stemmers and lemmatizer. The proposed model enhances the overall stemming and lemmatizing process performance of the model. There are a number of steps included in NLP processing of the text such as normalization, tokenization and stemming (lemmatizing) to derive the clean and accurate stem or lemma.

3.2.1 Tokenization

The process of tokenization removes the additional components such as Arabic stop words like (على) ,(في), white spaces in a sentence. Tokenization is in charge of outlining the word boundaries. This will help the lemmatizer to process the words more accurately and effectively. Choosing the Arabic stop words has its own preposition, pronouns and function words. Yet, there are no precise lists of stop words exist that can be accepted widely.

3.2.2 Normalization

In the Arabic language, there are 10 characters used as extra characters that need to be distinguished separately. These 10 characters create the overhead for stemmers and lemmatizer. These characters are our primary characters based on their location in a word.

For example, many shaped letters (Letter Hamza) (ا,أ,آ,إ) and Alef maksoura (ى), which is another form for Alef, (sometimes mixed while writing it with the letter yaa (ي)). TaaMarbota (ة) sometimes mixed with ha (ه). Hamza: (ئ ,ؤ ,ء) can be used interchangeably that depends on the sentence and the word (Darwish, Magdy& Mourad, 2012;Darwish, Magdy & Mourad, 2012).

3.2.3 Rule-based Implementation

Sembok & Ata (2013) stated that most of the errors in their work are due to the order in using the rules when Implementing the stemmer. To overcome this shortcoming, we divided the extra into groups of priorities based on the frequency of use as shown in Table 3.1, Since the extra letters are combined in the word (saltomineeha),"سألتمونيها ").

Table 3.1: The extra letters groups of priorities

Group 1 Most Frequently	Group2 less Frequently	Group3 least Frequently
(Alf ,"ا ")	(Hamzaa ," ء "),	(Seein,"س "),
(Waaw," و ")	(Meem ," م ")	(Haa ," ه")
(Yaa," ي ")	(Noon ," ن ")	(Laam," ل)
	(Taa," ت ")	

When the extra letter detects in the word, we Implement this letter rules to determine if this letter extra or not, if the letter is extra we remove the letter, if not extra we don't remove the letter, we repeat this workflow until finding the lemma of the word. Figure 3.1 explain the overall overview of the lemmatizing process**.** More about the rules of our proposed model in section 4.3.3 in chapter 4.

3.3 Proposed model

Arabic words have many forms or version of each word form. Even though, the existed problem of inappropriate content of stop word list and size, there exists an agreement that confirms that removal of stop words from Arabic text improves retrieval accuracy (Nwesri, 2008).

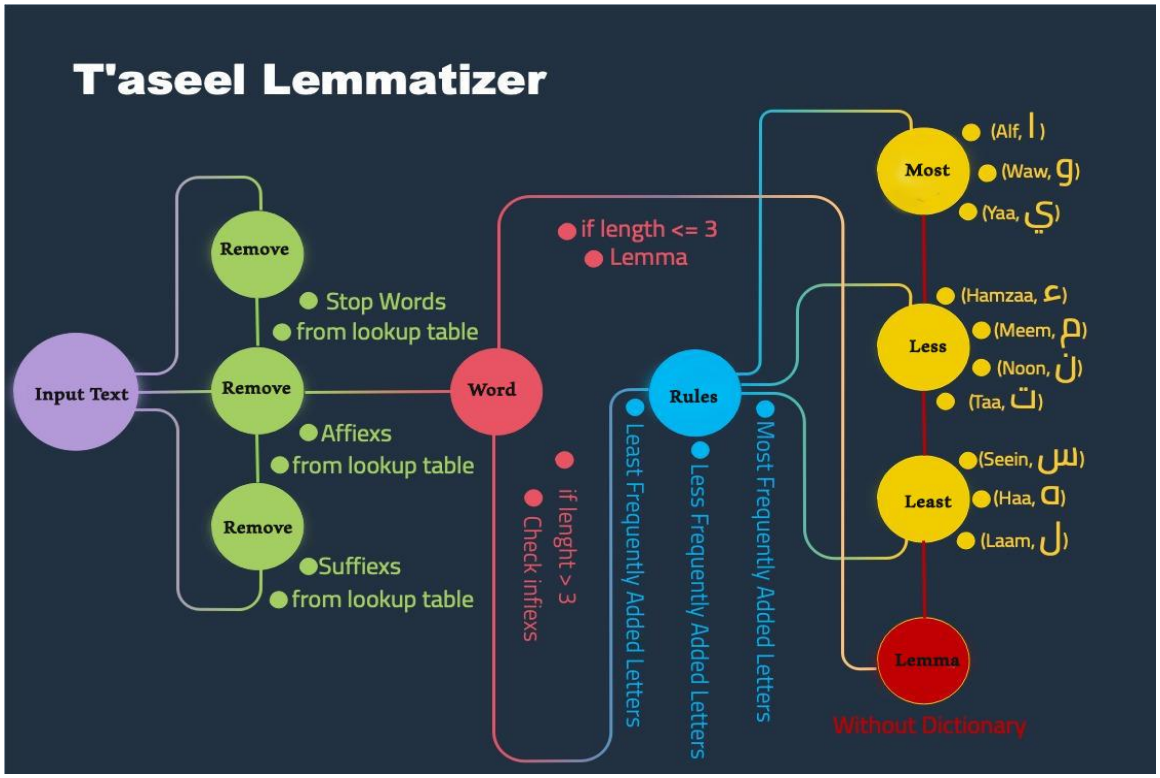


Figure 3.1 phases of the proposed model.

The following are the steps that describe the methodology of the proposed system.

1. The database is a set of proverbs in the Standard Arabic language.
2. Remove stop word
3. As a part of word normalization, we will remove the affix (postfix and prefix) according to a specified lookup table when the length of word five or more letters.
4. Look for the extra letters in the word but based on priorities established according to the groups shown in Table 3.1.
5. Implement the extra letter rule to extract the lemma
6. A lemma is when there is a word in which there are no extra letters.
7. The intensity is the addition of a repeating character (Shadaa, "ّ").
8. The feature of our work is dealing with the whole word and dividing the word into letters.
9. One of our goals is to have each lemma be present in the Arabic dictionary.
10. As none of the ancient Arabic linguists mentioned that there are predefined position of the extra letter (affixes) that exist always in the beginning or end of the word (determined prefixes or postfixes). The results are evaluated by comparing them with the results of an expert in the Arabic language.

As an example the proverb: " وَجَدْتُ النَّاسَ إِنْ قارَضْتَهُمْ قارَضُوْكَ"

1.Toknization: The sentences will be tokenized into words

- Token as: "وَجَدْتُ" " النَّاسَ" " إِنْ" " قارَضْتَهُمْ" "قارَضُوْكَ"

- After remove the stop word the output will be as:

"وَجَدْتُ" " الْنَّاسَ" " قارَضْتَهُمْ" "قارَضُوْكَ"

2. Normalization: Remove diacritics

- The output will be as: "وجدت" "النّاس" " قارضتهم " " قارضوك"
- After remove the affix the output will be as: " وجدت" "نّاس" " قارضت " " قارض"

3. Implemented the rules according to the groups of priorities for extra letters as shown in Table 3.1, example: To extract the lemma from the word " قارضت "

a. Divided to the Arabic letters :(Gaf, "ق"), (Alef,"ا"),(Raa,"ر"),(Dtaad,"ض"),(Taa, "ت").

b. According to the priorities in the process of looking for the extra letters, the priority will be (Alef," ا"),then (Taa,"ت")

c. First, it searches for the rule letter (Alef,"ا "): The rule of (Alf,"ا") is if the letter (Alf,"ا") is second position and before and after is three original letters then letter (Alf,"ا") is extra letter we remove it.

- Then the word output is: " قرضت ".

a. Then searches for the rule letter (Taa,"ت"): The letter (Taa,"ت") if the letter (Taa,"ت") came in last position in the word and the three letter before is original letter then the letter (Taa,"ت") is extra we remove it.

- Then output lemma is the word: " قرض ".

3.4 Summary

A literature review of the existing attempts concludes that to extract the stems or lemmas in a text or sentence, there are no standard sets of rules existing. The proposed model used a better understanding of the Arabic language morphological operation to improve Arabic text stemming (without using a dictionary). The new lemmatizer called a T'assel lemmatizer were introduced in this chapter.

The choice of naming our model a lemmatizer is that our goal is that each output lemma should be present in the Arabic dictionary. The Stem does not need to be always a valid dictionary word, while a lemma is always a valid dictionary word.

The rules to extract the lemmas in Arabic text is defined using morphological books الممتع بالتصريف (1987, الاشبيلي)and المغني (1997, الحنبلي). The proposed model uses the suggestion table lookup that has the excess words (stop words) and each excess letter has defined their own rules so that extraction of lemmas can be smoothened in a word for finding the lemmas.

Chapter Four

Experimental Results and Discussion

4.1 Introduction

The previous chapter declared the methodological of a proposed T'assel lemmatizer. This chapter will present a description of the proposed lemmatizer. The introduction is in section 4.1, In Section 4.2 description of the applied dataset, Section 4.3 illustrates the experiments setup and configurations and the implementation of the proposed lemmatizer, in section 4.4 is model performance evaluation, the result is in section 4.5, section 4.6 is the discussion and then the summary in section 4.7.

4.2 Dataset Specifications

The dataset is a set of proverbs written in the Standard Arabic language, contain 480 proverbs with 2,493 words from it 1637 unique words, the words have advance levels of expressions and many of word is Uncommon words.

The proverbs were collected from proverbs Arabic book (معجم الامثال العربي (1992, صيني وعبد العزيز وسليمان)), the proverbs speak about different subjects and have a lot of variety words and the words are of diverse lengths. A sample of the dataset is shown in the Figure 4.1.

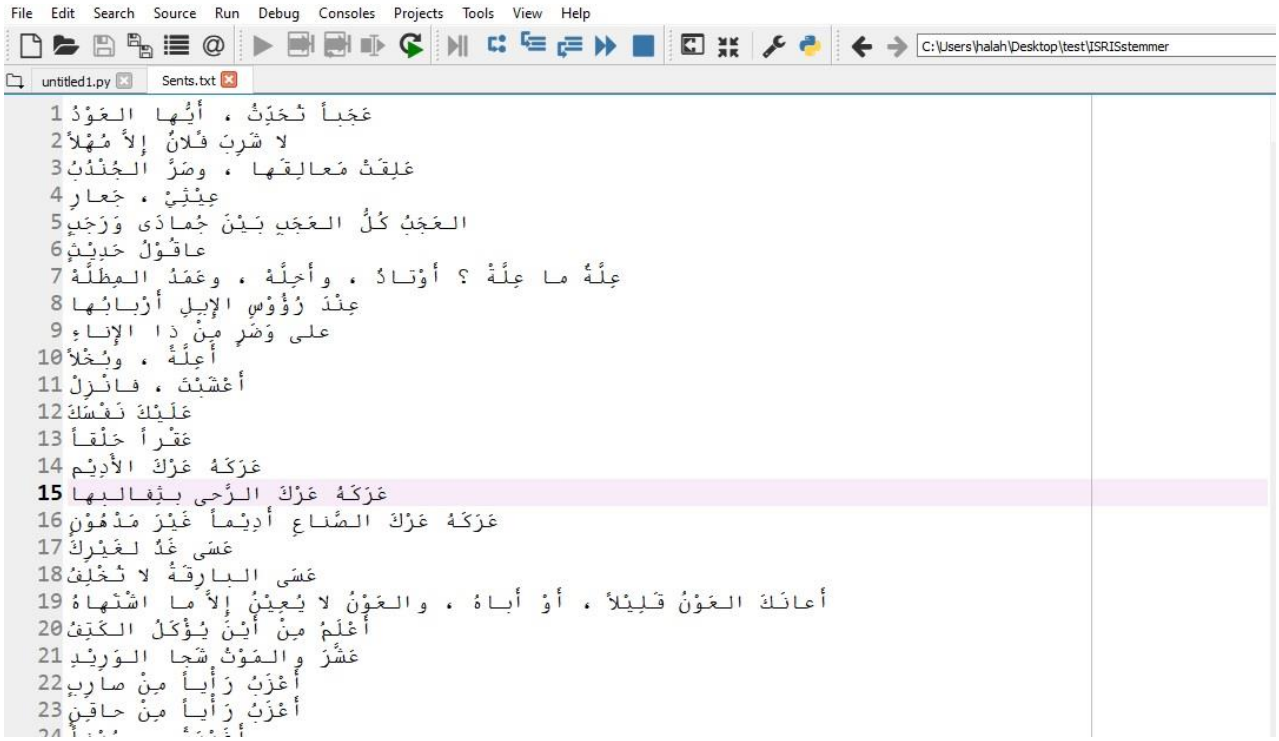


Figure 4.1 sample of the dataset

4.3 Experimental Procedure

The main component of our lemmatization approach is a rule-based analyzer, the rules build on two factors: the **length of the word** and the **position** of the letter in the word. And the methods of executed those rules are in order of priority of the extra letters groups.

The proposed model uses two suggestion lookup tables to remove the stop words sample shown in Table 4.1 and the affix (postfix and prefix) sample shown in Table 4.2, next is the explaining of T'assel lemmatizer procedure in detail.

4.3.1 Experiments Setup

python programming language has a good supporting library for NLP based processing application so we used python language programming for the implementation

of the T'assel lemmatizer. The experiments were executed on spyder (python 3.7), Spyder is an open-source cross-platform for scientific programming in the Python language.

The libraries used in our experiment are The RE library (Regular expression operations), from __future__ we used unicode_literals, we used nltk for the tokenization process. requires that all words have Unicode types. the hardware environments used in this experiment are as follows:

- 4 Intel(R) core(TM) i5-3210 M CPU @ 2.50GHz.
- 4.00GB RAM
- 64 bit-operating system x 64 processor

4.3.2 Encoding

There is a diverse design used to encoded the texts yet UTF-8 encoding (Unicode 6.3 Character Code Charts) is the most common use in the World Wide Web is the reason we used the UTF-8 encoding in our experimental.

4.3.3 Tokenization and Normalization

The process of tokenization removes the additional components such as Arabic stop words and white spaces in a sentence as well as split the text to words, stop words work as a syntactic function and don't convey any meaning or reference to the subject.

we implemented the process by using a lookup table, if the word matching any of word in stop word lookup table the word will be removed from the candidate word to lemmatize. Table 4.1 show samples of stop words used in T'assel lemmatizer.

Table 4.1 Samples of stop words used in T'assel lemmatizer

Stop Word	UTF-8 Encoding
أيا	\u0623\u064a\u062
أل‌ج	\u0623\u062c\u0644
إذن	\u0625\u0630\u0646
إنْ	\u0625\u0650\u0646\u0652
أن	\u0623\u064e\u0646
إنّ	\u0625\u0650\u0646\u0651
أنّ	\u0623\u064e\u0646\u0651
أم	\u0623\u064e\u0645
أل	\u0623\u0644
أمَا	\u0623\u064e\u0645\u064e\u0627
أمّا	\u0623\u064e\u0645\u0651\u0627
إمّا	\u0625\u0650\u0645\u0651\u0627
أو	\u0623\u0648
أَلَا	\u0623\u064e\u0644\u064e\u0627
إلّا	\u0625\u0644\u0651\u0627
أَلّا	\u0623\u064e\u0644\u0651\u0627

In the normalization process inside T'assel lemmatizer we

- Removed punctuation
- Removed diacritics (such as: Fatha, Kasra, Dama, without removing the Shada and sokon)
- As a part of normalizing the word we removed the affix (postfix and prefix) according to a specified lookup table when the length of word five or more letters,

the lookup table was created with the most common and regulars affix. Table 4.2 shows some of the removed Affix in T'assel lemmatizer.

- In our lemmatizer, the word is written in Standard Arabic language we used it as is without replacing any letter with others.

Table 4.2 Samples of the removed Affix in T'assel lemmatizer

Removed prefix		**Removed postfix**	
Prefix	UTF-8 Encoding	Postfix	UTF-8 Encoding
أسا	\u0623\u0633\u0627	تما	\u062a\u0645\u0627
أست	\u0623\u0633\u062a	تمو	\u062a\u0645\u0648
أسن	\u0623\u0633\u0646	تن	\u062a\u0646
أسي	\u0623\u0633\u064a	ناك	\u0646\u0627\u0643
أفن	\u0623\u0641\u0646	ناكم	\u0646\u0627\u0643\u0645
أفي	\u0623\u0641\u064a	ناكما	\u0646\u0627\u0643\u0645\u0627
ألت	\u0623\u0644\u062a	ناكن	\u0646\u0627\u0643\u0646
ألي	\u0623\u0644\u064a	ناه	\u0646\u0627\u0647
أوت	\u0623\u0648\u062a	ناها	\u0646\u0627\u0647\u0627
سأ	\u0633\u0623	ناهم	\u0646\u0627\u0647\u0645
ست	\u0633\u062a	ناهما	\u0646\u0627\u0647\u0645\u0627
سي	\u0633\u064a	ناهن	\u0646\u0627\u0647\u0646
فسأ	\u0641\u0633\u0623	وا	\u0648\u0627
فست	\u0641\u0633\u062a	يه	\u064A\u0647
فسي	\u0641\u0633\u064a	نه	\u0646\u0647

4.3.4 The Rules

In the previous chapter we present a method to extract the lemmas of word by using rules based on the understanding of the Arabic language morphological, the rules build on two factors: the **length of the word** and the **position** of the letter in the word.

So these Rules determined when we can remove some letters from a word (cases to remove) and where they must not be removed to create the lemma, below examples of some used rules.

- The letter (Noon," ن "): if the letter (Noon," ن ") came after the letter (Alf,"ا") and the letter (Alf,"ا") was after three original letters then the letter (Noon," ن ") is extra so we can remove it.Example word (عمدان),(فرحان).

- The letter (Meem,"ي") if the letter (Meem,"ي") came second position in the word and the letter before and after is original letter then the letter (Meem,"ي") is extra we remove it .and if it came in beginning and after came three original letter then it also extra letter we remove it, and if the letter (Meem,"ي") came in third position and before and after is original letter then it also extra letter we remove it.

- The letter (Hamzaa," ء"): if The letter (Hamzaa," ء") came in the beginning and after came two original letters then the letter (Hamzaa," ء") is not extra letter (don't remove) example: "أخذ ", but if then the letter (Hamzaa," ء") came in the beginning

and after came three original letters then the letter (Hamzaa," ء") is extra letter so we remove it. Example word " أكرم ".

- The letter (Meem,"م"): if the letter (Meem,"م") came in the beginning and After came three original letters then the letter (Meem,"م"): is extra so we can remove it. Example words (معمل).

We build the rules as function for each letter with secretin length for example: to remove The letter (Yaa, "ي") for the word with length of **four letter** we build the fowling function.

If letter (Meem,"ي") in beginning and after is three letter not in EL

If letter (Meem,"ي") in second position and after and before is letter not in EL

If letter (Meem,"ي") in third position and before and after is letter not in EL

Where **EL** is the set of the extra letter, an example for implementation the function of the letter (Meem,"ي") rules above are the words: "شريف" ,"سيطر" ,"يفعل".

The main objective of this step is to use the Arabic morphological to define a group of rules to use it in the stemming (lemmatizer) model, we create the rules and executed it by grouping the extra letters to groups of priorities to execute.

4.4 Model Performance Evaluation

In this section, defines the evaluation equation used to measure the performance of T'assel lemmatizer model.

True lemma (TL): The number of correctly identified lemma by the model.

False lemma (FL): The number of wrongly identified lemma by the model.

Total of words (TW): The total numbers of candidate words to lemmatize from the dataset.

Accuracy (AC): Calculates the proportion of correctly identified lemmas.

Fail (FC): Calculates the proportion of wrongly identified lemmas.

Accuracy (AC) $= \frac{\text{True lemma}}{\text{Total of words}} = \frac{\text{TL}}{\text{TW}}$ ……………………………………..….4.1

Fail (FC) = $\frac{\text{False lemma}}{\text{Total of words}} = \frac{\text{FL}}{\text{TW}}$ ………………………………….………….......4.2

To evaluate the results of our proposed lemmatizer based on the lengths of the words in our dataset we will use the following equations.

True lemma-length (TLs): The number of correctly identified lemma by the model according to specific words lengths.

False lemma- length (FLs): The number of wrongly identified lemma by the model according to specific words lengths.

Total of words- length (TWs): The total numbers of candidate words to lemmatize from the dataset according to specific words lengths.

Accuracy-length (ACs): Calculates the proportion of correctly identified lemmas according to specific words lengths.

Fail -length (FCs): Calculates the proportion of wrongly identified lemmas according to specific words lengths.

Accuracy (ACs) $= \frac{\text{True lemma}-\text{length}}{\text{Total of words}-\text{length}} = \frac{\text{TLs}}{\text{TWs}}$ ……...…..….4.3

Fail (FCs) = $\frac{\text{False lemma}-\text{length}}{\text{Total of words}-\text{length}} = \frac{\text{FLs}}{\text{TWs}}$ ……..…….…..…..4.4

4.5 Results

In this section, the result of our lemmatizer will present, and comparing our proposed Lemmatizer result with ARLSTem Stemmer (Abainia, Ouamour& Sayoud, 2017) result and ISRI Stemmer (Taghva et al., 2005). In the next section, analyzing and discussing the results based on the lengths of the words in our dataset.

To evaluate the performance of the experiment conducted on our dataset, we evaluate our achieved result by using the evaluation equation in section 4.4 and compare our model result and result with ARLSTem Stemmer (Abainia et al., 2017) and ISRI Stemmer (Taghva et al., 2005). Samples of comparison result shown in Table 4.4.

To evaluate our achieved result, we used equation (4.1) and equation (4.2) from section 4.4, where TL= 1002 and TW =1352 as follows:

Accuracy (AC) = $\frac{\text{TL}}{\text{TW}} = \frac{1002}{1352} = 0.7411$ so the accuracy percentage is 74.11% .

Fail (FC) = $\frac{\text{FL}}{\text{TW}} = \frac{350}{1352} = 0.25$ so the fail percentage is 25% .

To evaluate ARLSTem Stemmer (Abainia et al., 2017) and ISRI Stemmer (Taghva et al., 2005) conducted on our dataset we used the equations (4.1) and (4.2) from

section 4.4. The accuracy result of ARLSTem Stemmer (Abainia et al., 2017) was 54% and the accuracy result of ISRI Stemmer (Taghva et al., 2005) was 72.2%.

Table 4.3 Samples of comparison between T'assel lemmatizer result and other Stemmers

Arabic word	Length of word	T'assel lemmatizer Our result	ARLSTem Stemmer result	ISRI Stemmer result
النّازِلِيْن	9	نزل	نازل	نزل
النّوافِلا	9	نفل	نوافلا	نفل
'لَطَمَتْنِيْ	6	لطم	لطمتني	لطم
أثْكِلِيْه	6	ثكل	ثكل	ثكل
بالِياً	5	بلي	بالي	بال
كالفاخِرَةِ	8	فخر	فاخر	فخر
لَقِيْتُهُ	5	لقي	لقي	لقت
نافِعاً	5	نفع	نافع	نفع
حَمَّلْتُكَ	6	حمل	حمل	حمل
مَحْمَلِكَ	5	حمل	محمل	حمل
'مُقْبِلَةً	5	قبل	مقبل	قبل
الحَوالِبُ	7	حلب	حوالب	حلب
تَدَعَنَّ	5	دع	تدع	تدع
تُحْسَبُوا	6	حسب	تحسب	حسب
قارَضُوْكَ	6	قرض	قارضو	قارضو
المَجْزُوْزِيْنَ	9	جزز	مجزوز	جزز

As shown in Table4.4, our proposed model helped in reduce some ambiguity problems by fairly handling the broken plural as in the words النَّوافِلا and الحَوالِبُ.

The results of the comparison show that the ARLSTem Stemmer (Abainia et al., 2017) many times miss extracting the correct stem of the word, and could produce a completely new word that doesn't exist in the Arabic language and it didn't address the

broken plural forms. The results of the ISRI Stemmer (Taghva et al., 2005) were better than ARLSTem Stemmer (Abainia et al., 2017) .

As the accuracy rate of lemmatizer (stemmer) depends on an aspect such as the type of stemming approach and the component of the dataset used in the implementation of the lemmatizer (stemmer), the result will change due to the difference of the type of the dataset that used to test and the different approach used.

Table 4.5 shows a sample of errors in our proposed lemmatizer. Where the inverted letters were the major of ours lemmatizer errors.

Table 4.4 Samples of errors in our proposed lemmatizer

Arabic word	Length of word	T'assel lemmatizer Error result	The lemma
'برُغائِها	7	رغئ	رغا
افْتَد	5	فتد	فدى
'هَفْوَةٌ	5	هفو	هفا
بالمَشْرَفِيَّةِ	10	مشرفي	شرف
رُؤُوْسِ	4	رؤوس	رأس
المُنْتَعِلُ	8	تعل	نعل
لأرِيَنَّكَ	7	لأر	رأى
يَأتَمِرُ	5	أتر	امر
'المَمْطَوْرُ	7	طرر	مطر
المُؤْتَمَنُ	8	مؤتم	امن
لَلبَهِيْتَةِ	8	بهيت	بهت
أرْجاؤُه	6	رجؤ	أرجى

4.6 Discussion

In this section the discuss results of our proposed lemmatizer based on the lengths of the words in our dataset. The experiment is split into four different divisions.

Our dataset has a number of words length of four- letters (632), length of five-letters (398), length of six- letters (198), and length of seven- letters and above (124). To evaluate our model performance based on the lengths of the words in our dataset we used the equations (4.3) and (4.4) from section 4.4 for four divisions.

Example: Evaluating the result when the number of correctly identified lemma by the model according to four- letter lengths was 506 words (TLs = 506) and TWs = 632, using equations (4.3) and (4.4) from section 4.4 for the four divisions as follows.

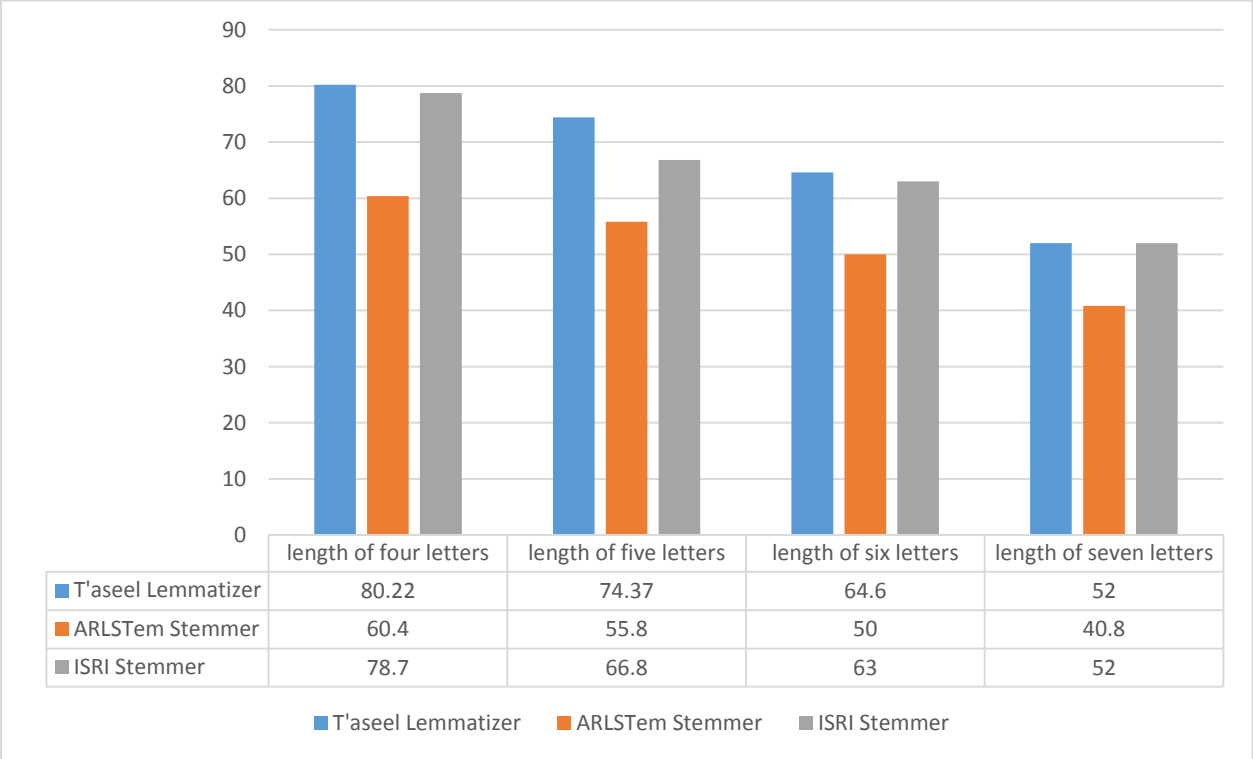
Accuracy (ACs) $= \frac{\text{TLs}}{\text{TWs}} = \frac{506}{632} = 80.22\%$

Fail (FCs) $= \frac{\text{FLs}}{\text{TWs}} = \frac{126}{632} = 19.7\ \%$

After applying our proposed T'assel lemmatizer the results indicate that when the length of the word is four -letters the accuracy was 80.22% with fail 19.7% when the length of the word is five -letters the accuracy was 74.37% with fail 25.62%, and when the length of the word was six -letters the accuracy was 64.6% with fail 35.3%, as well when the length of word seven and above letters the accuracy was 52% with fail 48%.

To evaluate ARLSTem Stemmer (Abainia et al., 2017) and ISRI Stemmer (Taghva et al., 2005) based on the lengths of the words in our dataset we used the equations (4.3) and (4.4) from section 4.4 for four divisions.

The summary of our proposed lemmatizer (T'assel lemmatizer) result accuracy based on the lengths of the words and ARLSTem Stemmer (Abainia et al., 2017) and ISRI Stemmer (Taghva et al., 2005) showed in Figure 4.2.



4.2 The accuracy results comparison between T'assel lemmatizer and other Stemmers based on lengths of words

4.7 Summery

In this chapter, the experiment has been conducting with an analysis of results, we compared our result with other stemmers. The result in the implementation shows that our proposed model helped in reduce some ambiguity problems by fairly handling the broken plural and perform better than ARLSTem Stemmer (Abainia et al., 2017) , ISRI Stemmer's (Taghva et al., 2005) performance was the nearest to our result. The next chapter provides a research conclusion and future work.

Chapter Five

Conclusion and Future Work

5.1 Conclusion

In this thesis, a developed and evaluated a new Arabic lemmatizer proposed, T’assel lemmatizer, the core of the T’assel lemmatizer is to Implement the rules of letters to extract the lemmas based on priorities established according to the groups of the extra letters.

The plan here is to keep the meanings of words after reducing the extensional forms of each word into a common root or base (lemma), the different extensional forms of a word can convey altered meanings, our proposed model helped in reduce some ambiguity problems by handling the broken plural.

The result of the implementation of the proposed T’assel lemmatizer indicates that it has an accuracy of lemma when the length of the words is four are 80.22% and accuracy when five letters is 74.37%, still there is needed work on the word with length more than 6 letters, and with overall accuracy of 74.11%. T’assel lemmatizer performance was better than ARLSTem Stemmer (Abainia et al., 2017), and slightly better than ISRI Stemmer's performance.

5.2 Future Work

A number of improvements could be applied to our proposed lemmatizer could be considering in the future as follows:

1) Modify the rules to improving in control removing the extra letter in our proposed lemmatizer.

2) Merge our proposed lemmatizer with approach using Arabic WordNet (Kreaa et al., 2014) to create hybrid lemmatizer with higher accuracy.

3) Make our dataset to be as a benchmark on other stemmers.

REFERENCE

- Abainia, K., Ouamour, S., & Sayoud, H. (2017). A novel robust Arabic light stemmer. Journal of Experimental & Theoretical Artificial Intelligence, 29 (3), 557-573. DOI: 10.1080/0952813X.2016.1212100

- Alansary, S., Nagi, M., &Adly, N. (2007). Building an International Corpus of Arabic (ICA): progress of compilation stage. In 7th international conference on language engineering, Egypt, 352–360. Retrieved 10 10,2019, from http://www.bibalex.org/unl/Attachements/Paper/Building%20an%20International%20Corpus%20of%20Arabic%20(ICA).pdf

- Al-Fedaghi, S., & Al-Anzi, F. (1989). A new algorithm to generate Arabic root-pattern forms. In proceedings of the 11th national Computer Conference and Exhibition. 391-400. Dhahran, Saudi Arabia: King Fahd University of Petroleum and Minerals. Retrieved 10 10,2019, from https://arxiv.org/ftp/arxiv/papers/1203/1203.3584.pdf

- Al-Harbi, S., Almuhareb, A., Al-Thubaity, A., Khorsheed, M. S., & Al-Rajeh, A. (2008). Automatic Arabic text classification. In Proceedings of The 9th International Conference on the Statistical Analysis of Textual Data .77-83.JADT .Retrieved 10 10, 2019, from https://eprints.soton.ac.uk/272254/

- Aljlayl, M., Frieder, O., & Frieder, O. (2002). On Arabic search: improving the retrieval effectiveness via a light stemming approach. In Proceedings of the eleventh

international conference on Information and knowledge management. 340-347. CIKM. doi: 10.1145/584792.584848

- 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. Journal of King Saud University-Computer and Information Sciences, 27(2), 94-103.doi: 10.1016/j.jksuci.2014.04.001

- Al-Lahham, Y. A., Matarneh, K., & Hasan, M. (2018). Conditional Arabic light stemmer: condlight. Int. Arab J. Inf. Technol., 15(3A), 559-564. Retrieved 10 10 ,2019, from https://www.academia.edu/40470732/Conditional_Arabic_Light_Stemmer_CondLight

- Al-Omari, A., Abuata, B., & Al-Kabi, M. (2013). Building and benchmarking new heavy/light Arabic stemmer. In The 4th International conference on Information and Communication systems (ICICS'13).17-22. Retrieved 10 10.2019, from https://www.researchgate.net/publication/265091746_Building_and_Benchmarking_New_HeavyLight_Arabic_Stemmer_The_fourth_International_Conference_on_Information_and_Communication_Systems

- Alserhan, H. M., &Ayesh, A. S. (2006). An application of neural network for extracting Arabic word roots. In Proceedings of the 10th WSEAS international conference on Computers, 10, 646–650. ICCOMP'06. doi:10.5555/1981848.1981967

- Awwad, S. (2019). Arabic Word stemming Based on Pattern Affixes Removal. In Proceedings of the 10th International Conference on Information and Communication Systems (ICICS). 1-6. IEEE. doi: 10.1109/IACS.2019.8809169.

- Beesley, K. R. (1996). Arabic finite-state morphological analysis and generation. In Proceedings of the 16th conference on Computational linguistics. 1. 89-94. Association for Computational Linguistics. doi: 10.3115/992628.992647

- Black, W., Fellbaum, C., Alkhalifa, M., Elkateb, S., Pease, A., Rodriguez, H., & Vossen, P. T. J. M. (2006). Introducing the Arabic WordNet project. Proceedings of the 3rd Global Wordnet Conference. 295-299. In P. Sojka, K-S. Choi, C. Fellbaum, & P. T. J. M. Vossen (Eds.), Retrieved 10 10.2019, from https://research.vu.nl/en/publications/introducing-the-arabic-wordnet-project

- Buckwalter, T. (2002). Buckwalter Arabic morphological analyzer version 1.0. Philadelphia: Linguistic Data Consortium. Retrieved 10 10.2019, from https://catalog.ldc.upenn.edu/LDC2002L49

- Chen, A., &Gey, F. (2002). Building an Arabic stemmer for information retrieval. In TREC Vol. 2002. 631-639. Retrieved 10 10.2019, from https://trec.nist.gov/pubs/trec11/papers/ucalberkeley.chen.pdf

- Dahab, M. Y., Ibrahim, A., & Al-Mutawa, R. (2015). A comparative study on Arabic stemmers. International Journal of Computer Applications, 125(8). 38-47.doi: 10.5120/ijca2015906129

- Darwish, K., Magdy, W., & Mourad, A. (2012). Language processing for Arabic microblog retrieval. In Proceedings of the 21st ACM international conference on

Information and knowledge management.2427-2430 .ACM .doi: 10.1145/2396761.2398658

- El-Sadany, T. A., & Hashish, M. A. (1989). An Arabic morphological system. IBM Systems Journal, 28(4), 600-612.doi: 10.1147/sj.284.0600

- Fautsch, C., & Savoy, J. (2009). Algorithmic stemmers or morphological analysis? An evaluation. Journal of the American Society for Information Science and Technology, 60(8), 1616-1624.doi:10.1002/asi.21093

- Feldman, R., & Sanger, J. (2007). The text mining handbook: advanced approaches in analyzing unstructured data. Cambridge university press. doi: 10.1017/CBO9780511546914

- Froud, H., Benslimane, R., Lachkar, A., &Ouatik, S. A. (2010). Stemming and similarity measures for Arabic Documents Clustering. In 2010 5th International Symposium On I/V Communications and Mobile Network. Rabat, 1-4. IEEE. doi: 10.1109/ISVC.2010.5656417.

- Glybovets, A. (2015). Arabic natural language processing. 177(9). 34-37. Retrieved 10 10.2019, from http://www.irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?I21DBN=LINK&P21DBN=UJRN&Z21ID=&S21REF=10&S21CNR=20&S21STN=1&S21FMT=ASP_meta&C21COM=S&2_S21P03=FILA=&2_S21STR=NaUKMAkn_2015_177_9

- Hadni, M., Ouatik, S. A., &Lachkar, A. (2013). Effective Arabic stemmer-based hybrid approach for Arabic text categorization. International Journal of Data Mining & Knowledge Management Process, 3(4), 1-14. doi : 10.5121/ijdkp.2013.3401

- Hammo, B. H. (2009). Towards enhancing retrieval effectiveness of search engines for diacritisized Arabic documents. Information retrieval, 12(3), 300-323.doi: 10.1007/s10791-008-9081-9

- Harmanani, H. M., Behrouz, W. T., &Raheel, S. (2006). A rule-based extensible stemmer for information retrieval with application to Arabic. The International Arab Journal of information Technology 3 (3) .9-15, Retrieved 8 12, 2019, from https://iajit.org/index.php?option=com_content&task=view&id=197&Itemid=268

- Hasanuzzaman, H. (2013). Arabic language: Characteristics and importance. The Echo. A Journal of Humanities & Social Science, 1(3). 11-16. Retrieved 8 12, 2019, from https://www.academia.edu/3236727/Pratidhwani_the_Echo

- Kammoun, N. C., Belguith, L. H., &Hamadou, A. B. (2010). The MORPH2 new version: A robust morphological analyzer for Arabic texts. In JADT 2010: 10th International Conference on Statistical Analysis of Textual Data.9-11. Retrieved 8 12, 2019, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.652.3634&rep=rep1&type=pdf

- Khoja, S., & Garside, R. (1999). Stemming Arabic text. Lancaster University Computing Department.23-28 Retrieved 8 12, 2019, from https://www.scirp.org/(S(i43dyn45teexjx455qlt3d2q))/reference/ReferencesPapers.aspx?ReferenceID=2004190

- Khoja, S., Garside, R., & Gerry, K. (2001). An Arabic tagset for the morphosyntactic tagging of Arabic, corpus linguistics. Lancaster University, 17-24. Retrieved 8 12, 2019, from https://www.yumpu.com/en/document/view/34382149/an-arabic-tagset-for-the-morphosyntactic-tagging-of-arabic

- Kreaa, Abdel Hamid, Ahmad S. Ahmad, and Kassem Kabalan. (2014).Arabic words stemming approach using Arabic WordNet. International Journal of Data Mining & Knowledge Management Process ,4(6), 1-14.doi: 10.5121/ijdkp.2014.4601

- Larkey, L. S., Ballesteros, L., & Connell, M. E. (2002). 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. 2.275-282. ACM.doi: 10.1145/564376.564425

- Larkey, L. S., Ballesteros, L., & Connell, M. E. (2007). Light stemming for Arabic information retrieval. In Arabic computational morphology ,3, 221-243. Springer. doi: 10.1007/978-1-4020-6046-5_12

- Marwan B., (2004). Arabic Language Processing in Information Systems,1-7.Springer.doi: 10.1598/RT.61.1.2

- Mustafa, M., Eldeen, A. S., Bani-Ahmad, S., & Elfaki, A. O. (2017). A comparative survey on Arabic stemming: approaches and challenges. Computer Science & Communications. Intelligent Information Management, 9(2), 39-67.doi: 10.4236/iim.2017.92003

- Mustafa, A. M. (2013). Mixed-Language Arabic- English Information Retrieval. (Thesis). University of Cape Town ,Faculty of Science ,Department of Computer Science. Retrieved 8 12, 2019, from http://hdl.handle.net/11427/6421

- Mustafa, M., Aldeen, A. S., Zidan, M. E., Ahmed, R. E., &Eltigani, Y. (2019). Developing Two Different Novel Techniques for Arabic Text Stemming. Intelligent Information Management, 11(1), 1-23.doi: 10.4236/iim.2019.111001

- Naili, M., Chaibi, A. H., &Ghezala, H. H. B. (2019). Comparative Study of Arabic Stemming Algorithms for Topic Identification. Procedia Computer Science, 159, 794-802.doi: 10.1016/j.procs.2019.09.238

- Nwesri, A. (2008). Effective retrieval techniques for Arabic text. (Thesis). RMIT University. School of Computer Science and Information Technology, Australia. Retrieved 8 12, 2019, from https://researchbank.rmit.edu.au/eserv/rmit:6764/Nwersi.pdf

- Otair, M. A. (2013). Comparative analysis of Arabic stemming algorithms. International Journal of Managing Information Technology, 5(2), 1-13.doi: 10.5121/ijmit.2013.5201

- Porter, M. F. (1980). An algorithm for suffix stripping. program, Program: electronic library and information systems. 14(3), 130-137.doi: 10.1108/eb046814

- Prathibha, R. J., & Padma, M. C. (2015). Design of rule based lemmatizer for Kannada inflectional words. In *2015* International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT*)* . 264-269. IEEE.doi: 10.1109/ERECT.2015.7499024

- Sembok, T. M. T., & Ata, B. A. (2013). Arabic word stemming algorithms and retrieval effectiveness. In Proceedings of the World Congress on Engineering ,3, 3-5. Retrieved 8 12, 2019, from http://www.iaeng.org/publication/WCE2013/WCE2013_pp1577-1582.pdf

- Syiam, M. M., Fayed, Z. T., & Habib, M. B. (2006). An intelligent system for Arabic text categorization. International Journal of Intelligent Computing and Information Sciences, 6(1), 1-19. Retrieved 8 12, 2019, from https://www.researchgate.net/publication/48340463_An_Intelligent_System_For_Arabic_Text_Categorization

- Taghva, K., Elkhoury, R., & Coombs, J. (2005). Arabic stemming without a root dictionary. In International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II .1. 152-157. IEEE.doi: 10.1109/ITCC.2005.90

- Zaidan, O. F., &Callison-Burch, C. (2014). Arabic dialect identification. Computational Linguistics, 40(1), 171-202.doi: 1.1162/COLI_a_00169

- Zamani, F. E., Umam, K., Azis, W. D. I., &Abdillah, W. S. (2019). Analysis and implementation of computer-based system development of stemming algorithm for finding Arabic root word. In Journal of Physics: Conference Series, 1402(6), 30-36. IOP Publishing. doi:10.1088/1742-6596/1402/6/066030

المراجع العربية

- الاشبيلي, ابن عصفور(1987). **الممتع في التصريف** .ط 1, بيروت : دار المعرفة للطباعة والنشر
- الحنبلي , موفق الدين (1997) .**المغني** .ط3 , الرياض : دار عالم الكتب للطباعة والنشر
- صيني، محمود، وعبد العزيز، ناصف، وسليمان، ومصطفى (1992).**معجم الامثال العربية**. ط1. بيروت :مكتبة لبنان