NEURAL-BASED ARABIC MORPHOLOGICAL ANALYZER

A THESIS SUBMITTED TO
THE SCHOOL OF COMPUTING

BY

TEGUH IKHLAS RAMADHAN

2301191004

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF
MASTER OF INFORMATICS
IN
THE SCHOOL OF COMPUTING

TELKOM UNIVERSITY
2021

APPROVAL PAGE

Approval of the School of Computing of Telkom University
I certify that this thesis satisfies all the requirements as a thesis for the degree of Master Informatics.
Date Jun 24 , 2021 (*the date can be set manully)

(Dana Sulistyo Kusumo, Ph.D.)
Head of Master Informatics
This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Informatics.

Date Jun 24 , 2021

(Supervisor's name) (Co-Supervisor's name)
Supervisor Co-Supervisor

Examining Committee Members.
Date Jun 24 , 2021

(Jury's name) (Chairperson of the jury) :

(Jury's name) (jury's member) :

(Jury's name) (jury's member) :

ABSTRACT

Al-Qur'an is a source and guidance for Muslims. It is a document that is 15 centuries ago and well written in the Arabic language. Many Muslims have to learn Arabic in addition to knowing the meaning of it. One of the most critical aspects of the Arabic language is morphology and identifying the word's morphological description. It is called by morphological analysis task. This task is essential because from the morphological aspect of a word, can know the different form of a word, and from that, it can know the meaning of it. The Gonzales paper has successfully created a model to identify morphological features (MSD) of Arabic word verb only. This study will focus on adding some other Arabic type of word, which is a noun. Trying to use the current state-of-the-art approach method is neural-based with the recurrent neural network (RNN). RNN can capture more information about the sequence of sub-word like prefix, infix, root, and suffix to make a better msd identifier. The input is a single Arabic word Going through pattern extraction, subword vectorizing, verb form identification, pronoun and type of word identification, and finally MSD identification process to see the result. This model successfully identify MSD with 99% accuracy and 97% F1 - score.

Keywords: morphosyntactic description, recurrent neural network, Arabic word classification , sub-word vectorizing

ABSTRAK

Al-Qur'an adalah sumber dan petunjuk bagi umat Islam. Ini adalah dokumen yang berusia 15 abad yang lalu dan ditulis dengan baik dalam bahasa Arab. Banyak Muslim harus belajar bahasa Arab selain mengetahui artinya. Salah satu aspek yang paling penting dari bahasa Arab adalah morfologi dan mengidentifikasi deskripsi morfologi kata. Makalah Gonzales telah berhasil membuat model untuk mengidentifikasi fitur morfologi (MSD) kata kerja bahasa Arab. Penelitian ini akan fokus pada penambahan beberapa jenis kata Arab lainnya, yaitu kata benda. Mencoba menggunakan metode pendekatan state-of-the-art saat ini berbasis saraf dengan jaringan saraf berulang (RNN). RNN dapat menangkap lebih banyak informasi tentang urutan sub-kata seperti prefiks, infiks, root, dan sufiks untuk membuat pengidentifikasi msd yang lebih baik. Input berupa satu kata Arab Melalui ekstraksi pola, vektorisasi subkata, identifikasi bentuk kata kerja, identifikasi kata ganti dan jenis kata, dan terakhir proses identifikasi MSD untuk melihat hasilnya. Model ini berhasil mengidentifikasi MSD dengan akurasi 99%.

Kata kunci: morphosyntactic description, recurrent neural network, Arabic word classification , sub-word vectorizing

CONTENTS

APPROVAL ii

ABSTRACT iii

ABSTRAK iv

CONTENTS v

LIST OF TABLES vii

LIST OF FIGURES ix

LIST OF TERMS x

1 INDTRODUCTION 1

1.1 Rationale . . . 1
1.2 Theoretical Framework . . . 2
1.3 Conceptual Framework/Paradigm . . . 3
1.4 Statement of the Problem . . . 3
1.5 Objective and Hypotheses . . . 4
1.6 Assumption . . . 4
1.7 Scope and Delimitation . . . 4
1.8 Significance of the Study . . . 4

2 REVIEW OF LITERATURE AND STUDIES 5

2.1 Related Literatures . . . 5
2.1.1 Finite-state morphological analysis and generation of Arabic at Xerox Research: Status and plans in 2001 . . . 5
2.1.2 A computational lexeme-based treatment of Arabic morphology . . . 5
2.1.3 Standard arabic morphological analyzer (SAMA) . . . 5
2.1.4 MAGEAD: a morphological analyzer and generator for the Arabic dialects . . . 5
2.1.5 A syllable-based account of Arabic morphology . . . 6
2.1.6 Elixirfm: implementation of functional arabic morphology . . . 6
2.1.7 A lexicon of Arabic verbs constructed on the basis of Semitic taxonomy and using finite-state transducers . . . 6
2.1.8 Arabic Word Generation and Modelling for Spell Checking . . . 6
2.1.9 Jabalín: a Comprehensive Computational Model of Modern Standard Arabic Verbal Morphology Based on Traditional Arabic Prosody 6

2.1.10 Rule Based Pattern Type of Verb Identification Algorithm for The Holy Qur'an . . . 7
2.2 Related Studies . . . 7
2.2.1 Recurrent Neural Network . . . 7
2.2.2 Syntax (*Nahwu*) . . . 8
2.2.3 Morphology (*Sharaf*) . . . 8
2.2.4 Verb (*fiil*) . . . 9
2.2.5 Arabic Buckwalter Transliteration . . . 11
2.2.6 Pattern Table . . . 11

3 RESEARCH METHODOLOGY 13
3.1 Research Design . . . 13
3.1.1 Flowchart . . . 13
3.1.2 Data Preprocessing . . . 14
3.1.3 Build RNN Model . . . 16
3.1.4 Testing . . . 20
3.2 Population/Sampling . . . 22
3.2.1 Data Source . . . 22
3.2.2 Data Generation . . . 22
3.2.3 Experiment Scenario . . . 22

4 IMPLEMENTATION , AND RESULT ANALYSIS 24
4.1 System Implementation . . . 24
4.1.1 Data Preprocessing . . . 24
4.1.2 Build RNN Model . . . 25
4.1.3 MSD Identification . . . 27
4.2 Jabalin Implementation . . . 29
4.2.1 Jabalin Online Interface . . . 29
4.2.2 Jabalin MSD TAG Mapping . . . 29
4.2.3 Web Crawling Get the Result . . . 30
4.3 Result Analysis . . . 30

5 CONCLUSION AND RECOMMENDATIONS 32
5.1 Conclusions . . . 32
5.2 Future Work . . . 32

BIBLIOGRAPHY 33

Appendices 35

A MISCELLANEOUS 37

LIST OF TABLES

1.1 Input Example from 3 different languages 2
1.2 Arabic side of MSDs 3
1.3 Dictionary dependency problem example that the two type of word must be known before both of it 3

2.1 Every Arabic word both *fi'il* and *isim* here must be attached to one of these 14 pronouns 9
2.2 Verb forms 10
2.3 Buckwalter Examples 11

3.1 Raw data from corpus.quran.com, location means the address of each every word in the Qur'an, form is its Arabic word on Buckwalter form, TAG is POS TAG of its word, and features is its MSD (verb form, root, lemma, gen, num etc. 15
3.2 The format of the pattern will be extracted from each Arabic word tested. In the form of a prefix, suffix, and diacritics infix pattern beside the prefix and suffix, which is the differentiator of each type of word, dhomir, and wazan 17
3.3 Examples of prefixes and suffixes exist. The number of prefixes and suffixes available for all types of words is more than this. It is just an example of a prefix and suffix that is in the form I form 17
3.4 The example of sub word vectorizing 18
3.5 RNN layer detail 19

4.1 The result of preprocessing process from Quran corpus data, Position is verse, chapter and word position in the Qur'an, word, and root is in the Buckwalter form. Tag, *Wazan* and other features in the part of making msd gold later on. 25
4.2 This table results from generating processes from Quran corpus data with the help of a pattern table. The type of word and *dhamir* or pronoun is getting from the pattern table to help the following process in the system. MSD already in the proper format after this process 26
4.3 The result of pattern extraction and subword vectorizing process 27
4.4 The number of data after splitting the dataset 27
4.5 Testing result 27
4.6 The details of result accuracy base on the MSD 28
4.7 MSD tag that produce by Jabalin system and translate into this system, the position starts from left to right 30
4.8 Result Jabalin system per all of the MSD 31

A.1 Mapping the Arabic features into the MSD 37

LIST OF FIGURES

2.1 Standard structure of RNN that process sequence of input [12] 7
2.2 The Arabic pattern table 11
2.3 The Arabic pattern table consists of prefix, infix, suffix, and root. The root will be added with the Arabic word later. 12

3.1 Major flowchart of the entire system 13
3.2 Breakdowned flowchart that show the entire process of the system 14
3.3 Data preprocessing process which consists of preprocessing and generating process. 15
3.4 Build RNN model process which consists of pattern extraction, word vectorizing, split data , training and testing processes 16
3.5 The simple design of recurrent neural network 19
3.6 The testing process is to test the model and get the final output which is morphosyntactic description (MSD) 20

4.1 The Jabalin online interface for Arabic morphological analyzer 29

LIST OF TERMS

Terms	Definition
Sharaf	Arabic morphological aspect.
Fi'il	Mostly verb in Arabic language.
Isim	Mostly noun in Arabic language
Huruf	Particle in Arabic
Tashrif	Changing the form of word to other form
Lughawi	Inflection
Istilahy	Derivation
Dhomir	Pronoun
Wazan	Verb form
MSD	Morphological features or morphosyntactic description
RNN	Recurrent neural network
Nahwu	Arabic syntactic aspect
Buckwalter	Arabic transliteration
Mufrad	Singular
Muthanna	Dual
Jamak	Plural
Mudhakar	Masculine
Muannath	Feminine
Fi'il Madhi	Perfect verb
Fi 'il Mudhari	Imperfect verb
Fi'il Amr	Imperative verb
Shahih	Regular verb
Mutal	Irregular verb
Tsulatsy	Triliteral form
Ruba'iy	Quadriliteral form

CHAPTER 1

INDTRODUCTION

This chapter includes the following subtopics, namely: (1) Rationale; (2) Theoretical Framework; (3) Conceptual Framework/Paradigm; (4) Statement of the problem; (5) Hypothesis (Optional); (6) Assumption (Optional); (7) Scope and Delimitation; and (8) Importance of the study.

1.1 Rationale

Indonesia is a country where most followers of the religion are Islam. About 232 million people are Muslims. Muslims have a holy book, the Qur'an. A written document or text that is more or less 15 centuries ago in which Muslims are obliged to do what is ordered and stay away from what is prohibited written in the Qur'an. Al-Qur'an is written in Arabic so that many Indonesian people who learn Arabic to be able to understand the contents of the Qur'an. Processing Arabic becomes necessary to analyze the Qur'an further or make tools to facilitate learning Arabic. Before doing so back to basics of Linguistics is a morphological analysis..

In Arabic, the science of morphology is called *Sharaf*. *Sharaf* is the basis of Arabic and is also called the science of tools to understand sentences in Arabic and the key to opening a repository of Islamic knowledge. An important part of morphological is the formation of sentences or classes of words. Different from other languages, for example, in Indonesian, there are 13 classes of words. In Arabic, there are only 3, i.e., *isim* (noun), *fi'il* (verb) and *huruf* (particle). Not that other words are omitted, it's just examples such as adjectives, adverbs, pronouns that belong to the *isim* group. While prepositions, conjunctions, question words enter into *huruf*. So the three elements above are the root or core of all the word classes that exist[16]. Whereas all verbs include *fi'il* but not all *fi'il* are verbs. Several adjectives fall into the *fi'il* category [3].

In Sharaf (morphology), the most important thing is to regulate or focus on changing the form of words to other words or commonly referred to as *tashrif* [3]. The pattern of word formation or commonly called *wazan* is created. *Tashrif* divided into two, namely *tashrif lughawi* (inflection) and *tashrif istilahy* (derivation) [23]. In Arabic, the changes in word types include types (*fi'il* / *isim*), *dhomir* / pronoun and *wazan* or patterns of the formation of these words. The changes and patterns of each word to another influence the semantics or meaning of the word. Therefore it would be better if there is a system or program that can identify the word both from inflection and its derivation for morphological

analysis of the Qur'an.

The Gonzales paper [21], a reference paper from this study, has conducted a morphological analysis of Arabic. Jabalin application that can be accessed on the site `http://elvira.lllf.uam.es/jabalin/analizarForma.php` has been able to identify and generate the infected word and its derivation successfully. The paper only focuses on one element of the word that is *fi'il* or verb. That study explicitly said that the future work is to work on morphological analysis of *isim* or nouns. Therefore, this research focus on the identification model. In addition to only being able to be identified, it can also identify *isim*. But here *isim* is limited to the only isim whose root is derived from the verb.

In the application of the annotation of the Qur'an that is currently an identification of the types of words and *dhomir* (pronoun), but there is no pattern / *wazan*, which is where this pattern is essential to know the meaning and or changes in other words of the word. The Gonzales paper [21] focuses more on *fi'il* or verbs, while in this study the *isim* or noun is added which is the future work in the paper. From these two things, this research focuses on identifying patterns and also identifying *isim*. It's just that *isim* in this study focus on the *isim* whose roots come from *fi'il*.

The current state of the art for Arabic morphological analysis is rule-based. However, this study will be using a neural-based approach or using the deep learning method. The method, in this case, is a recurrent neural network because in the current morphological task paper, which is SIGMORPHON 2018 [10] mainly using neural-based, and the result is good. Try to implement that in this study to help to make a morphological analyzer model.

1.2 Theoretical Framework

This system input will be the Arabic word, and the output is its MSD, or morphosyntactic description [30], or morphological features [19]. On the Tabel 1.1 is the example of

Table 1.1: Input Example from 3 different languages

No	Language	Input (Word)	Output (MSD)
1	English	Run	pos=V,mood=IND,tense=PST,per=3,num=SG
2	Indonesia	makan	pos=V,voice=ACT
3	Arabic	يَعْلَمُوْنَ	pos=V,mood=IND,aspect=IPFV,voice=ACT, GEN=M,PER=3,NUM=PL,verb-form=Iia

input and msd from 3 different common languages, and the Arabic got more MSDs, so that makes the Arabic word is interesting to study. Although on the Arabic side, the MSD can

be classified into three namely types of word, pronoun(*dhamir*), and verb form (*wazan*), and will be discussed more later on (See on Table 1.2).

Table 1.2: Arabic side of MSDs

No	Arabic	MSDs
1	Type of words (madhi,mudhari,amr,nahyi)	pos={V/N},mood={IND/IMP}, aspect={IPFV/PFV},voice={ACT/PASS}
2	Dhamir (pronouns)	GEN={M/F},PER={1/2/3}, NUM={SG/DU/PL}
3	Wazan	verb-form={1-22}

1.3 Conceptual Framework/Paradigm

The previous recent study is Gonzales paper [21] there is no Noun tag identification. On the recent morphological world contest, Sigmorphon 2018 [10] there is no verb form or *wazan* identification which *wazan* is important on the Arabic word itself according to book [3]. So in this study is for focussing on fulfilling weaknesses from these two related studies, which are adding noun tag and add verb form identification and wrap it to msd form.

Table 1.3: Dictionary dependency problem example that the two type of word must be known before both of it

Fiil Madhi	Fiil Mudhari	Wazan
ضَرَبَ	يَضْرِبُ	Iai
ضَرَبَ	x	?
يَضْرِبُ	x	?

To identify the verb form, especially on form Iau till Iii (form I), the two types of word *fiil madhi* and *fiil mudhari* must be known before for both of it to know its verb form 1.3. To know the type of word of the word, you must see the Arabic dictionary, but the resource or the API is limited, so it must identify its verb form without using the dictionary. It is important because form Iau and Iii is the most common verb form on Arabic according to the Holy Qur'an.

1.4 Statement of the Problem

According to the problem explanation and the weaknesses from the previous study, the problem is how to identify the verb form without a dictionary and how to add a noun to the data and wrap it into the msd form.

1.5 Objective and Hypotheses

The purpose of this study is the same with aim morphological analysis of the Arabic language. The objective is to establish the formalization of the words in Arabic itself. Try to add some other data type of word, which is a noun, to the dataset. Using a deep learning approach (recurrent neural network) hope can help identify the MSD of the Arabic word better. The recurrent neural network can capture the information of the sub of the word. Neural-based also can eliminate the dictionary dependency problem. It is just read from all the data and see the sub-word sequence connection and mapping to the verb form to see the result.

1.6 Assumption

Using deep learning approach (neural) can identify verb form without the help of dictionary. Using deep learning approach can make the model to identify the MSD of the word more better than rule-based approach.

1.7 Scope and Delimitation

The Arabic word in this study follow the rules, namely :

1. One Arabic with full of diacritics (*harakat*)
2. The Arabic word must be in the normal form not affected by the sentences rule (*nahwu*)
3. Active word only
4. The Arabic word roots do not contain weak letter which are ا, و,ي
5. The noun word is noun which derived from a verb

1.8 Significance of the Study

With this study can help

CHAPTER 2

REVIEW OF LITERATURE AND STUDIES

This chapter discusses the literature studies and theories of this study. It is divided into two sections: (1) the related literatures; and (2) the related studies.

2.1 Related Literatures

2.1.1 Finite-state morphological analysis and generation of Arabic at Xerox Research: Status and plans in 2001

Research conducted by Beesley et.al. in 2001 tried to make a morphological analysis and generation of words in Arabic. By using the finite-state machine method with the programming languages perl, lexc and twolc. And the extracted features are the root and pattern of the word itself. Incoming inputs are prefixes, patterns, and suffixes. Can produce 72,000,000 abstracts words [6].

2.1.2 A computational lexeme-based treatment of Arabic morphology

Research conducted by CavalliSforza et.al. in 2001 tried to make a morphological analysis of Arabic with the morphe method. Built with the Lisp programming language and the rule method used is lexeme based. With the input program, Lexeme and some of the word features [29].

2.1.3 Standard arabic morphological analyzer (SAMA)

Research undertaken by Buckwalter et.al. from 2004 to 2010 was to make an application called SAMA or short for standard Arabic morphological analyzer. It is an Arabic morphological analyzer with rule-based concatenative stem-based. Built using the Perl programming language and its input in the form of prefixes, suffixes, Arabic language stems, and compatibility tables. Covering large-scale Arabic grammar and using Buckwalter transliteration as well. The SAME application is open source and the results are 62 % accuracy rate for Quranic texts and 70 % for newspaper texts[13].

2.1.4 MAGEAD: a morphological analyzer and generator for the Arabic dialects

The research was undertaken by Habash et al. in 2010 the Magead application was a morphological analyzer for Arabic dialects. By using the finite-state transducer method and taking lexeme based on root and word patterns as its linguistic model. With input

that is root, pattern, vocalization, and affixes. Covering large-scale Arabic open-source grammars. This model has produced 94.9 % precision. [15]

2.1.5 A syllable-based account of Arabic morphology

Research conducted by Cahill et.al. from 2007 to 2010 was to make a morphological analyzer for Arabic discussion with the rule-based syllable method. By using the DATR programming language and input in the form of root, pattern, and inflow vowel. Cover Arabic grammar partially and use SAMPA transliteration.[9]

2.1.6 Elixirfm: implementation of functional arabic morphology

This research carried out by Smrz and Bielicky is making Elixirfm application which aims to implement morphology in Arabic functionally. Built using the haskell and perl programming languages. Using root and word patterns as linguistic models. Covers large-scale Arabic grammar. Using arabtex and buckwalter as transliteration. [27]

2.1.7 A lexicon of Arabic verbs constructed on the basis of Semitic taxonomy and using finite-state transducers

Research conducted by Neme in 2011 was to develop a special verbal Morphological analysis using FST. By using root and defined patterns as a linguistic model. The input of this program is 15,400 verbs, 31 root classes, and 460 reflected classes. With this model successfully produced 99.9 % lexical coverage [24]

2.1.8 Arabic Word Generation and Modelling for Spell Checking

Research conducted by Attia et.al. this is building an application called AraComlex which is a morphological analyzer application to fix SAME [13]. The method used is finite-state transducers and the lexc programming language. Use lexeme-based as a linguistic model. Inputs to the program are lemma, pattern, root and lemma lookup to the database. Cover large scale Arabic grammar. And this application managed to get the precision of 98.2 % at a recall of 100 % [26].

2.1.9 Jabalín: a Comprehensive Computational Model of Modern Standard Arabic Verbal Morphology Based on Traditional Arabic Prosody

Research conducted by Gonzalez et.al. is a reference paper in this study and tries to build an Arabic morphological analyzer and generation with Jabalin application. Using root and pattern as its linguistic model. It was built using python and input, which was entered the same as in ElixirFM [27] only it had been normalized. And produce 99.27 % of accuracy. In the future work this paper advocates toward nominal morphology [21].

2.1.10 Rule Based Pattern Type of Verb Identification Algorithm for The Holy Qur'an

The author's previous research is still on a small scale, namely to build a model of verb identification in Arabic conducted on the Qur'an using rule-based. Built using the java programming language and its input is a verb in the Qur'an. Use the root and word patterns as its linguistic model. Has identified 3 attributes namely the type of words, pronouns and wazan with an accuracy of 96.48 % [25]

2.2 Related Studies

2.2.1 Recurrent Neural Network

Recurrent neural network or RNN is one of the deep learning methods which can read and process sequential data like text [4]. Unlike other deep learning methods such as multilayer perceptron or backpropagation, there are no linkages between the input. In RNN, every sequential of inputs are matter and have the sequence information from each input or data.

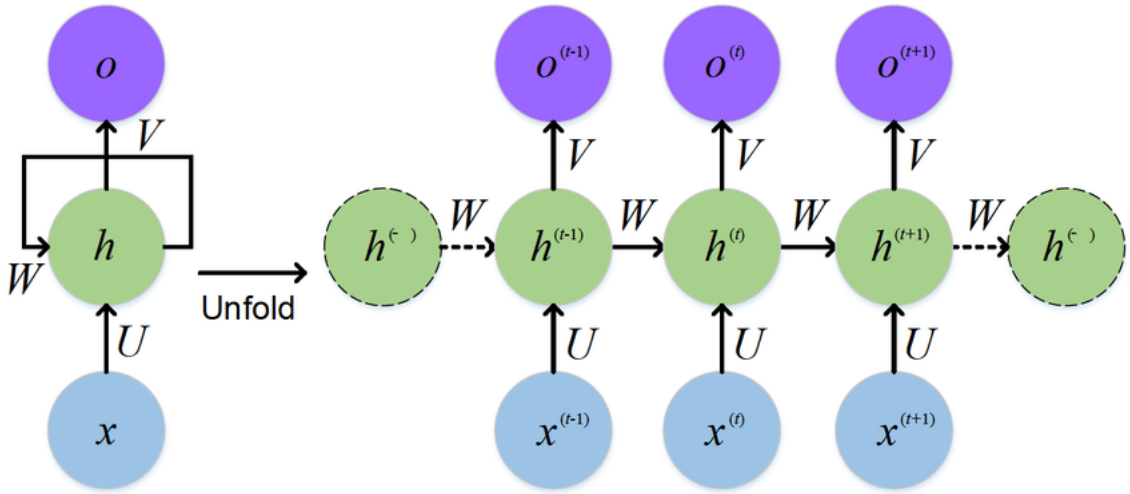

Figure 2.1: Standard structure of RNN that process sequence of input [12]

RNN resolve the problem especially in text mining task which are :

1. Handle variable-length sequence
2. Track long term dependencies
3. Maintain information about order
4. Share paramteres about the sequence

2.2.2 Syntax (*Nahwu*)

Nahwu is a branch of Arabic language which discusses how to compile sentences in accordance with the rules of Arabic, both related with the word in a sentence or word condition (final gift and form) in a sentence [2]. *Nahwu* in linguistics means syntax which is the set of rules, principles, and processes that govern the structure of sentences in a given language, usually including word order [8].

2.2.3 Morphology (*Sharaf*)

Morphology is a science that explains the procedure for changing a word from one forms to other forms to produce different meanings [3]. Morphology in linguistics means which is rules specify how new words and word forms are formed and function as redundancy rules with respect to existing complex words in the lexicon [7].
In morphology aspect in Arabic (*sharaf*), there are discussions closely related to this paper, namely :

- Sentence-forming Element
  The sentence forming elements in Arabic are 3, namely *isim*, *fiil* and *huruf*. *Isim* is a noun, *fiil* is a verb and *huruf* are one Arabic letter that has meaning [3].

- Pronouns (*dhamir*)
  In Arabic, there are many pronouns that can be seen in Table 2.1 [3]. Pronouns based on the number is divided into three namely singular (*mufrad*), dual (*muthanna*) and plural (*jamak*). By gender type, there are gentle (*mudhakar*) and feminine (*muannath*). Based on the type of pronoun there is a third, second and first pronoun [20].

  pronouns based on the number is divided into three namely *mufrad* one person, *muthanna* two people and plural three people or more. By type, there are *mudhakar* (gentle) and *muannath* (feminine). Based on the type of pronoun there is a third, second and first pronoun [20].

- Verb Forms (*wazan*)
  Verb pattern or *wazan* is a standard formula, where each verb will later enter one of the verb pattern [3]. On the Table 2.2[1]. there are six verb pattern of verb pattern which is distinguished by its perfect verb and imperfect verb pattern. With more spesific by its second diacritics on perfect verb and third diacritics on imperfect verb.

- Word form change (*tashrif*)
  tashrif is the change in words from the original form (verb) to other forms [3]

[1] with diacritics o : *sukun*, a : *fatha*, u : *dhamma*

Table 2.1: Every Arabic word both *fi'il* and *isim* here must be attached to one of these 14 pronouns

No	Mean	Dhamir	Amount	Gender	Pronoun type
1	He	هُوَ	Mufrad (singular)	Mudzakkar (masculine)	Third Person
2	They both	هُمَا	Mutsanna (2 person)		
3	They	هُمْ	jamakk (Plural)		
4	She	هِيَ	Mufrad (singular)	Muannats (feminine)	
5	They both	هُمَا	Mutsanna (2 person)		
6	They	هُنَّ	jamakk (Plural)		
7	You	أَنْتَ	Mufrad (singular)	Mudzakkar (masculine)	Second Person
8	You both	أَنْتُمَا	Mutsanna (2 person)		
9	You all	أَنْتُمْ	jamakk (Plural)		
10	You	أَنْتِ	Mufrad (singular)	Muannats (feminine)	
11	You both	أَنْتُمَا	Mutsanna (2 person)		
12	You all	أَنْتُنَّ	jamakk (Plural)		
13	I	أَنَا	Mufrad (singular)	Mudzakkar & Muannats	First Person
14	We	نَحْنُ	jamakk (Plural)		

2.2.4 Verb (*fiil*)

Verb or *fiil* in Arabic language has the meaning of action [3]. The type of verb is divided into three there are, perfect verb (*fiil madhi*), verb for the past that has the meaning already do something; imperfect verb (*fiil mudhari*), verb that it is doing or the action that will come after (future); and imperative verb (*fiil amr*), verb for command.

- Perfect verb (*fiil madhi*), verb for the past that has the meaning already do something.
- Imperfect verb (*fiil mudhari*), verb that it is doing or the action that will come after (future).
- Imperative verb (*fiil amr*), verb for command.

When viewed from the constituent letters also verb in Arabic is divided into two types, namely [3] :

- Regular/*Sahih* Verb
  Is verb whose constituent letters are free from the letters ا, و and ي. *sahih* is divided

Table 2.2: Verb forms

No	Arab	Form	Wazan
1	فَعَلَ-يَفْعُلُ	Iau / Triliteral	1 - tsulatsy mujarrod
2	فَعَلَ-يَفْعِلُ	Iai / Triliteral	2 - tsulatsy mujarrod
3	فَعَلَ-يَفْعَلُ	Iaa / Triliteral	3 - tsulatsy mujarrod
4	فَعِلَ-يَفْعَلُ	Iia / Triliteral	4 - tsulatsy mujarrod
5	فَعُلَ-يَفْعُلُ	Iuu / Triliteral	5 - tsulatsy mujarrod
6	فَعِلَ-يَفْعِلُ	Iii / Triliteral	6 - tsulatsy mujarrod
7	فَعَّلَ	II / Triliteral	فَعَّلَ biharfin - tsulatsy mazid
8	فَاعَلَ	III / Triliteral	فَاعَلَ biharfin - tsulatsy mazid
9	أَفْعَلَ	IV / Triliteral	أَفْعَلَ biharfin - tsulatsy mazid
10	تَفَعَّلَ	V / Triliteral	تَفَعَّلَ biharfayn - tsulatsy mazid
11	تَفَاعَلَ	VI / Triliteral	تَفَاعَلَ biharfayn - tsulatsy mazid
12	اِنْفَعَلَ	VII / Triliteral	اِنْفَعَلَ biharfayn - tsulatsy mazid
13	اِفْتَعَلَ	VIII / Triliteral	اِفْتَعَلَ biharfayn - tsulatsy mazid
14	اِفْعَلَّ	IX / Triliteral	اِفْعَلَّ biharfayn - tsulatsy mazid
15	اِسْتَفْعَلَ	X / Triliteral	اِسْتَفْعَلَ bitsalasaty ahrufin - tsulatsy mazid
16	اِفْعَالَّ	XI / Triliteral	اِفْعَالَّ bitsalasaty ahrufin - tsulatsy mazid
17	اِفْعَوْعَلَ	XII / Triliteral	اِفْعَوْعَلَ bitsalasaty ahrufin - tsulatsy mazid
18	اِفْعَوَّلَ	XIII / Triliteral	اِفْعَوَّلَ bitsalasaty ahrufin - tsulatsy mazid
19	فَعْلَلَ	I / Quadriliteral	فَعْلَلَ - ruba'iy mujarrod
20	تَفَعْلَلَ	II / Quadriliteral	تَفَعْلَلَ ziyadah biharfin - ruba'iy mazid
21	اِفْعَنْلَلَ	III / Quadriliteral	اِفْعَنْلَلَ ziyadah biharfayn - ruba'iy mazid
22	إِفْعَلَلَّ	IV / Quadriliteral	إِفْعَلَلَّ ziyadah biharfayn - ruba'iy mazid

into three, namely regular verb type 1 (*salim*), regular verb type 2 (*mahmuz*, and regular verb type 3 (*mudhaaf*) [17].

- Irregular/*mutal* Verb
  Is verb whose constituent letters contain at least one of ا, و and ي.

2.2.5 Arabic Buckwalter Transliteration

Buckwalter is a transliteration of Arabic which is quite popular and developed by Tim Buckwalter [14]. This transliteration has been widely used in the field of natural language processing especially for Arabic [1]. This research uses Buckwalter transliteration to create a program which later can be run on the programming language. Table 2.3 shows some examples of Arabic and Buckwalter forms.

Table 2.3: Buckwalter Examples

No	Arab	Buckwalter
1	فَتَبَسَّمَ	fatabas~ama
2	ضَاحِكًا	DaAHikFA
3	نِعْمَتَكَ	niEomataka
4	قَوْلِهَا	qawolihaA
5	أَوْزِعْنِي	>awoziEoniy

2.2.6 Pattern Table

Many process in this study relies on a pattern table. Pattern table is the Arabic rule sources that will map the pattern of the word into its type of word and pronoun [3]. The pattern table is also used to know the changes from every word. It is called derivation from a column to another column, and from row to another row, it is called inflection. This is the characteristics of the pattern table :

فِعل النهي	فِعل الامر	اسمُ مَفعُوْل	اسمُ فاعِل	مَصدَر	فِعل مُضارِع	فِعل مَاضِ	
		ma | o uwo N |	| aA i N |		ya | o u u |	| a a a |	هوَ
		ma | o uwo a | Ani / ma | o uwo a | yoni	| aA i a | Ani / | aA i a | yoni		ya | o u a | Ani	| a a a | A	هُمَا
		ma | o uwo u | wona / ma | o uwo i | yona	| aA i u | wona / | aA i i | yona		ya | o u u | wona	| a a u | woA	هُمْ
		ma | o uwo a | pN	| aA i a | pN		ta | o u u |	| a a a | to	هِيَ
		ma | o uwo a | taAni / ma | o uwo a | tayoni	| aA i a | taAni / | aA i a | tayoni		ta | o u a | Ani	| a a a | taA	هُمَا
		ma | o uwo a | AtN	| aA i a | AtN		ya | o u o | na	| a a o | na	هُنَ
laAta | o u o |	>u | o u o |				ta | o u u |	| a a o | ta	انتَ
laAta | o u a | A	>u | o u a | A				ta | o u a | Ani	| a a o | tumaA	انتُمَا
laAta | o u u | woA	>u | o u u | woA				ta | o u u | wona	| a a o | tumo	انتُمْ
laAta | o u i | yo	>u | o u i | yo				ta | o u i | yona	| a a o | ti	انتِ
laAta | o u a | A	>u | o u a | A				ta | o u a | Ani	| a a o | tumaA	انتُمَا
laAta | o u o | na	>u | o u o | na				ta | o u o | na	| a a o | tun~a	انتُنَ
					>a | o u u |	| a a o | tu	انَا
					na | o u u |	| a a o | naA	نحَنُ

Figure 2.2: The Arabic pattern table

1. The column heading shows its type of word and the row heading shows its pronoun of the pattern
2. The contents of each cell in this table are the pattern of the Arabic word, which are prefix, infix, and suffix converted in the Buckwalter form in this study.
3. The infix is consists of the sequence of roots, and *harakat* turns with each other. The root is blank in the pattern table because it will depend on the Arabic word itself.
4. One verb form is a different set of pattern tables. All of 22 total of verb form in Arabic it has different pattern table. Some of the verb form patterns have the same pattern, so that will be one of the challenges in this study.
5. There are triliteral (*tsulatsy*) verb form and quadriliteral (*ruba'iy* verb form. Triliteral is a verb form with three roots, and Quadriliteral is a verb form with four roots.

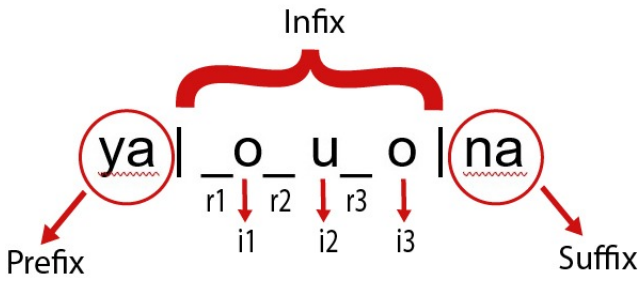

Figure 2.3: The Arabic pattern table consists of prefix, infix, suffix, and root. The root will be added with the Arabic word later.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Research Design

The appropriate research design should be specified and described (including requirement, modeling and detailed description of system/product/method development.

3.1.1 Flowchart

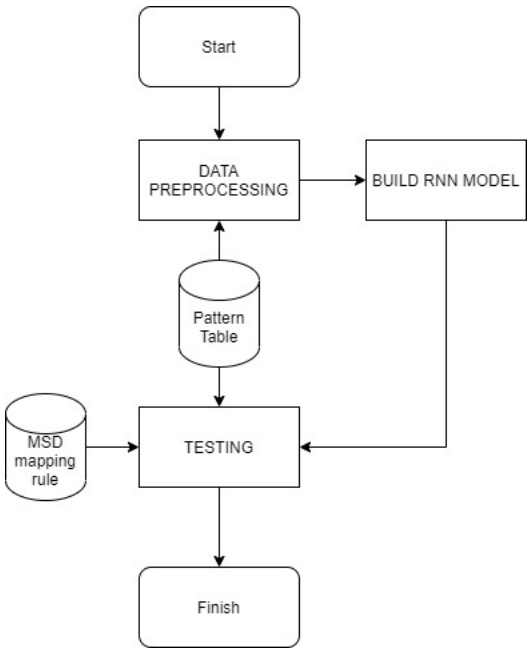

Figure 3.1: Major flowchart of the entire system

The flowchart in Figure 3.1 shows the flow of the entire system in general. First is the data preprocessing process. In this study, the data will be obtained from http://corpus.quran.com [11] which is still raw data and must be doing some preprocessing afterward. To do the data preprocessing, will get help from the pattern table is the table from Arabic linguistic book [3] that can help to generate more data to build the data training for the following process. The second is the build RNN model process. This process will focus more on building an RNN model with architecture, training, and test data. Build the RNN model is necessary because this model will be used to determine the verb form later on. The last process is the testing process that will test the data that has been build before to see its output which is msd, and evaluate the model that has been creating. All of the process will be explaining more later on in this chapter and the flowchart will be breakdowned to

see every detail for each of every process in this system in Figure 3.2.

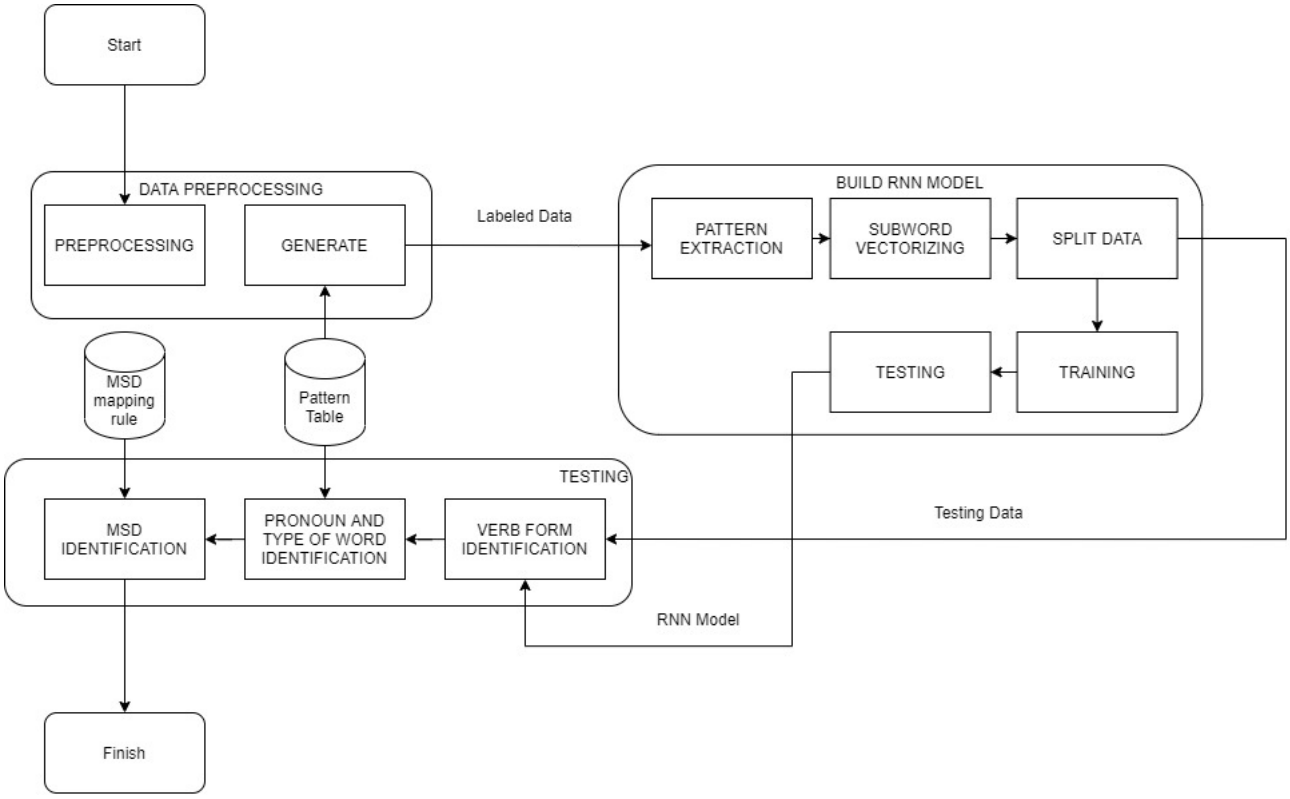

Figure 3.2: Breakdowned flowchart that show the entire process of the system

3.1.2 Data Preprocessing

The Arabic sources that will be used in this study are come from the website http://corpus.quran.com [11] because this data contains the Arabic, which has been labeled before with POS, verb form, and other morphological feature (msd). The raw data look like in Table 3.1. The Arabic word is in the Buckwalter Arabic format, and the POS TAG is not only V and N, so it must be some preprocessing before getting the correct data first for this study purposes. To get more Arabic word, the data which has been processed before will be generated with the help of pattern table. It will create more data with all of the possible inflicted and derived word base on Arabic linguistic (Figure 3.3).

Preprocessing

In this process will be preprocessing the data so the data will fit with this study limitation. The process will be done as follow :

1. Get only V and N TAG

2. Concatenate the prefix and suffix to form the full Arabic word

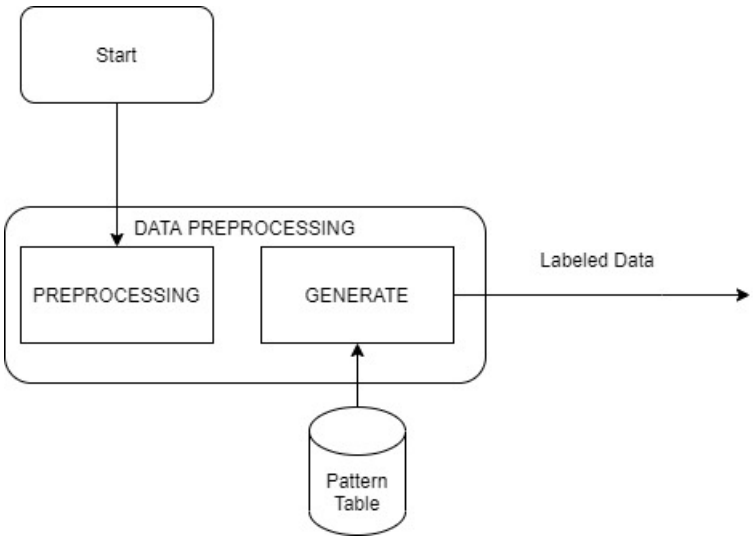

Figure 3.3: Data preprocessing process which consists of preprocessing and generating process.

Table 3.1: Raw data from corpus.quran.com, location means the address of each every word in the Qur'an, form is its Arabic word on Buckwalter form, TAG is POS TAG of its word, and features is its MSD (verb form, root, lemma, gen, num etc.

LOCATION	FORM	TAG	FEATURES
(1:1:1:1)	bi	P	PREFIX/bi+
(1:1:1:2)	somi	N	STEM/POS:N/LEM:{som/ROOT:smw/M/GEN
(1:1:2:1)	{ll∼ahi	PN	STEM/POS:PN/LEM:{ll∼ah/ROOT:Alh/GEN
(1:1:3:1)	{l	DET	PREFIX/Al+
(1:1:3:2)	r∼aHoma‘ni	ADJ	STEM/POS:ADJ/LEM:r∼aHoma‘n/ROOT:rHm/MS/GEN
(1:1:4:1)	{l	DET	PREFIX/Al+
(1:1:4:2)	r∼aHiymi	ADJ	STEM/POS:ADJ/LEM:r∼aHiym/ROOT:rHm/MS/GEN
(1:2:1:1)	{lo	DET	PREFIX/Al+
(1:2:1:2)	Hamodu	N	STEM/POS:N/LEM:Hamod/ROOT:Hmd/M/NOM
(1:2:2:1)	li	P	PREFIX/l:P+
(1:2:2:2)	l∼ahi	PN	STEM/POS:PN/LEM:{ll∼ah/ROOT:Alh/GEN
(1:2:3:1)	rab∼i	N	STEM/POS:N/LEM:rab∼/ROOT:rbb/M/GEN
(1:2:4:1)	{lo	DET	PREFIX/Al+

3. Avoid the word that its roots weak letter which are ا, و,ي

4. Get only Active voice

5. Get the feature which are Arabic in both of buckwalter and Arabic version, root, verb form / *wazan*, type of word, pronoun / *dhamir* and the msd to make a gold standard of data.

Generate

The generating process relies on a pattern table. The process will generate all of the possible inflected and derived the word from the source Arabic word. This generating process aims to make the data bigger to feed into the RNN later on.

The output of this process is the correct and labeled data which will be the reference for the following process, which is to build an RNN model.

3.1.3 Build RNN Model

After getting the labeled data from the previous process, the data will be used to make the RNN model purpose to identify an essential morphological feature, which is verb form or *wazan*. Figure 3.4 there will be five processes from the pattern extraction process going through the testing process sequentially.

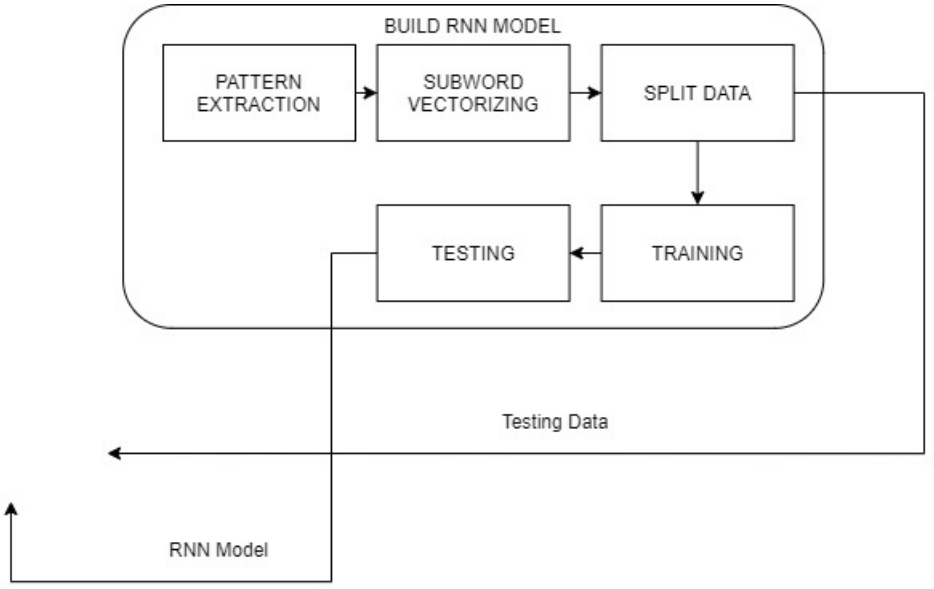

Figure 3.4: Build RNN model process which consists of pattern extraction, word vectorizing, split data , training and testing processes

Pattern Extraction

In this study, the pattern of a word is essential to determine the type of word. An Arabic word in this study uses the Buckwalter translation. It takes it from the prefix, suffix, diacritics, and Arabic letters between the prefix and suffix. The pattern extraction using the same method as common Arabic stemmer that usually used for certain other Arabic natural language processing task [5] (look Table 3.2).

At this stage, it is extracting or obtaining a pattern as seen in Table 3.2 by separating the center pattern by checking (string matching) with each prefix and suffix. Example

Table 3.2: The format of the pattern will be extracted from each Arabic word tested. In the form of a prefix, suffix, and diacritics infix pattern beside the prefix and suffix, which is the differentiator of each type of word, dhomir, and wazan

Pattern			Arab
Prefix	Infix	Suffix	
mu	a a o a a	taAni	مُتَدَحرِجَتَانِ

prefix and suffix Table 3.3 [1].

Table 3.3: Examples of prefixes and suffixes exist. The number of prefixes and suffixes available for all types of words is more than this. It is just an example of a prefix and suffix that is in the form I form

Prefixes		Suffixes	
Buckwalter	Arabic	Buckwalter	Arabic
ya	يَ	A	ا
ta	تَ	woA	وْا
>a	أَ	to	تْ
na	نَ	taA	تَا
		na	نَ
		tumaA	تُمَا
		tumo	تُمْ
		ti	تِ
		tun∼a	تُنَّ
		tu	تُ
		Ani	انِ
		wona	وْنَ
		yona	يْنَ
		naA	نَا

The suffix prefix extracts the pattern of the Arabic word and then infix (Algorithm 1). By stem the Arabic word according to the list of a prefix, infix list, suffix list, and the list of Arabic letters, which is a consonant letter. The output of this process is a prefix, infix, suffix, and the root of the Arabic word. This process also provides the reference for the word vectorizing process (Figure 2.3).

[1] Prefixes and suffixes are still many. This is only one example in verb form I

Algorithm 1 Get the pattern from one Arabic word

```
function PATTERN EXTRACTION(word)
    prepare
        prefixList                                  ▷ list of possible prefix
        infixList                                   ▷ list of possible infix
        suffixList                                  ▷ list of possible suffix
        arabicLetterList                            ▷ list of arabic letter in buckwalter
    end prepare
    lemma, suffix = suffixExtractor(suffixList,word)          ▷ extract the suffix
    prefix , lemma2 = prefixExtractor(prefixList,lemma)       ▷ extract the prefix
    infix, root = infixExtractor(arabicLetterList,lemma2)     ▷ extract the infix and root
    return prefix, infix, suffix, root
end function
```

Subword Vectorizing

After the word going to the pattern extraction process, there will be going to the subword vectorizing process. This study will be seeing the sequence between sub of the word, so the sub of the word (in this case is a pattern) must be vectorized so that it can be processed into the RNN model because RNN can only accept vector for its input. The encoding method that will be using is label encoding [22], because the prefix, infix, suffix, and the Arabic letter are categorical and have the list from the previous process.

All of the possible prefix, suffix, infix, and root will be store in the dictionary and have the label in it to the reference for its vector. For example, in the Table 3.4 is an example of sub word vectorizing with the word يَعْلَمُوْنَ. The number of every subword is there in the dictionary that has been built before. The dictionary number is 113 in total and with all possible prefix, suffix, infix, and root.

Table 3.4: The example of sub word vectorizing

Input	yaEolamuwona /يَعْلَمُوْنَ							
TAG	P	r1	i1	r2	i2	r3	i3	S
Buckwalter	ya	E	o	l	a	m	u	wona
Arabic	يَ	ع	*sukkun*	ل	*fetha*	م	*demma*	وْنَ
Digit Value	35	18	7	23	14	24	12	8
Digit Number	1	2	3	4	5	6	7	8

Split Dataset

At this stage, the data has vector attributes for each Arabic word. In this process, the data must split into data training and data testing. The data will be split using the ratio 8:2, 8 for data training and 2 for data testing. Data training will be used to train the model, and data testing will be used to evaluate the model that has been trained before.

Training

The purposes of training process is to build model to determine the verb form or *wazan*. The model will use RNN as a method and take the input from previous process which are prefix, infix, suffix and root (see Table 2.2). The output of the model will be the verb form or *wazan* there are 15 of total *wazan* in this study.

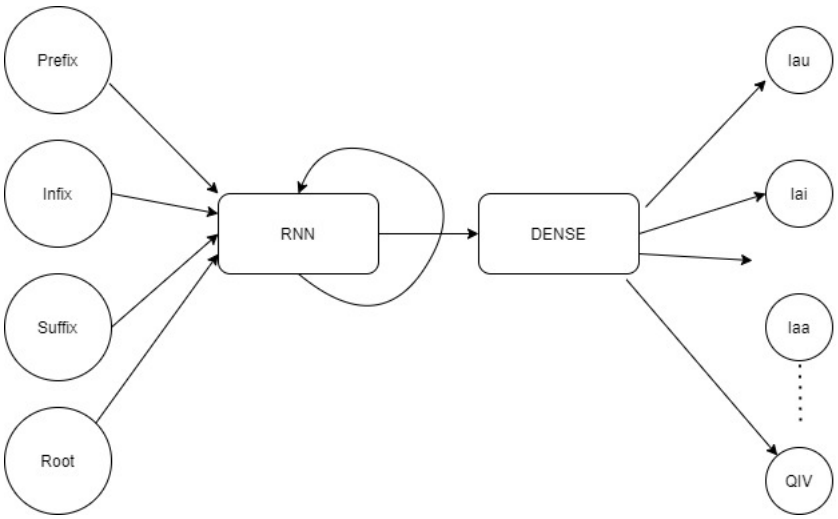

Figure 3.5: The simple design of recurrent neural network

The detail of the layer that will be used for this model, namely (see Table 3.5)

Table 3.5: RNN layer detail

No	**Layer (type)**	**Output Shape**
1	Embedding	(None, 12, 16)
2	Spatial Dropout	(None, 12, 16)
3	Simple RNN	(None, 128)
4	Dense	(None, 15)

1. Embedding
   The embedding layer is the Keras layer that will convert the subword vector into matrices for each integer in the value of the vector.

2. Spatial Dropout
   The spatial dropout layer is the Keras layer that will prevent overfitting.

3. Simple RNN
   The simple RNN layer is the RNN layer with a vanilla / simple structure that does not have the memory in it or LSTM.

4. Dense

The dense layer is the final layer which maps the output with a probability distribution and takes the highest probability to be the final output.

The implementation of the system or model in this study uses python programming language with tensorflow keras library and Adam optimizer [18]. This model runs in Lenovo laptop with specifications as follows [31].

1. Processor intel core i7 gen 11
2. 16 GB RAM
3. Intel Irisxe graphics

Testing

This process will test the model using a dataset that has been split before to see the accuracy, precission, recall and f1 score that is resulting from the training process [28].

3.1.4 Testing

This process is the final process to finally get the system's actual output, which is MSD. This process will be getting the RNN model built before and the testing dataset from the previous process. There are three subprocesses in this process: verb form identification, pronoun and type of word identification, and MSD identification.

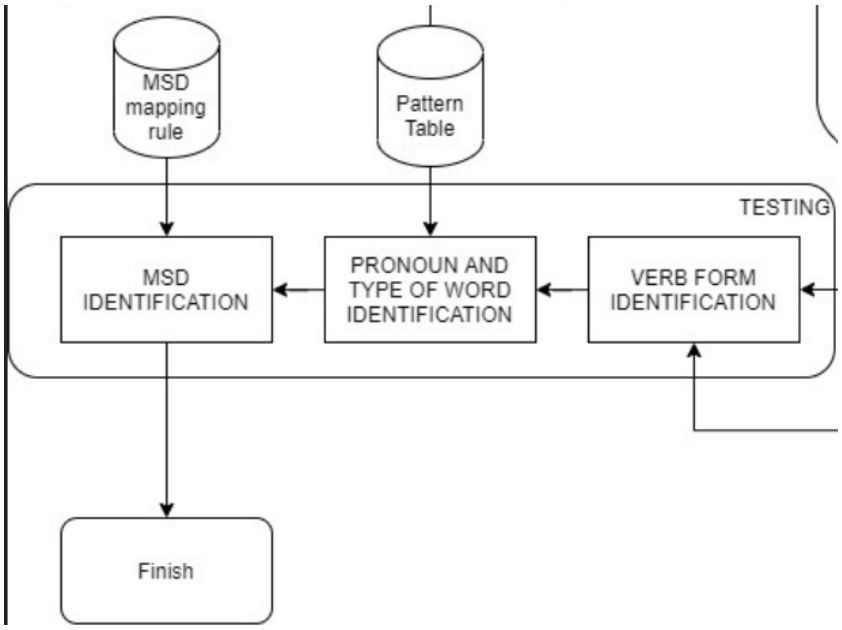

Figure 3.6: The testing process is to test the model and get the final output which is morphosyntactic description (MSD)

Verb Form Identification

Verb form identification is the first subprocess in the testing process. The verb form identification is the same as the RNN build model process because it gets the input as a vector of prefix, suffix, infix, and root, and the output is the verb form. The verb form or *wazan* is essential because it will be the reference table in the pattern table 2.2 to identify other morphology descriptions, for example, pronoun information and type of word.

The input of the process is data testing vectors and going to rnn model directly without training process. The output will be the predicted verb form itself.

Pronoun and Type of Word Identification

Algorithm 2 Get a pronoun and type of word from one Arabic word

```
1: function PROUNOUN AND TYPE OF WORD IDENTIFICATION(pattern, verbForm)
2:     prepare
3:         patternTable                          ▷ Prepare the pattern table data
4:     end prepare
5:     table = patternTable.getTableByVerbForm(verbForm)
6:     if (table.isPatternExist(pattern))
7:         pronoun = table.getPronoun(pattern)                  ▷ Get rows head
8:         typeOfWord = table.getTypeOfWord(pattern)            ▷ Get columns head
9:     else
10:        pronoun, typeOfWord = null
11:    end if
12:    return pronoun, typeOfWord
13: end function
```

Pattern table is all patterns in all types of words that exist in the type of words in Arabic. All existing patterns consist of the suffix and infix prefixes. It was formed in a table with columns representing word types and lines describing pronouns (look Figure 2.2 [2]). This pattern table will later become a reference to get the type of words and pronouns - also verb forms for some types of words [3]. The number of word patterns is counted for each word type, pronoun patterns, and verb form patterns.

Fi'il and *isim* derived from *fi'il* in Arabic are always attached to the pronoun [23] and its type of word. After identifying, the identification of *dhamir* becomes necessary because every *fi'il* or *isim* pattern which derived from *fi'il* always has pronouns. In Arabic, the pronouns are different from Indonesian and English. There are 14 types of pronouns in Arabic. See Table 2.1 for more details. The pronouns and their type of word getting from the pattern table get from the previous result. The process output is the verb form, and it automatically tells which pattern table should refer to get the pronoun and type of word from that word by look up the pattern table itself.

[2] Examples of pattern tables in verb form I

[3] Some verb forms must require a dictionary to determine them

MSD Identification

MSD identification process is the final process of the entire system resulting in the final output, which is msd or morphosyntactic description. This process will take the output of the previous process, which is its type of word and pronoun. From that particular output, map the type of word and pronoun into the MSD [3]. From Table A.1 can see all of the complete mapping rules, base on the Arabic book rule.

The Algorithm 3 is simply just get the msd from the rules and append it so can get all of possible msd of the word.

Algorithm 3 Get complete morphological features

```
1:  function MSD IDENTIFICATION(pronoun, typeOfWord)
2:      prepare
3:          msdMappingRule                    ▷ Prepare the msd mapping rule
4:          msd                                          ▷ initiate empty msd
5:      end prepare
6:      if(pronoun and typeOfWord not null)
7:          msd.add(msdMappingRule.getMsdFromPronoun(pronoun))
8:          msd.add(msdMappingRule.getMsdFromTypeOfWord(typeOfWord))
9:      else
10:         msd = null
11:     end if
12:     return msd
13: end function
```

3.2 Population/Sampling

3.2.1 Data Source

The Arabic data used in this study comes from http://corpus.quran.com [11]. The data will be filtered with specific criteria which will cover the scope conditions.

3.2.2 Data Generation

After getting the data from http://corpus.quran.com, it will be generated to get all of the possible inflected and derivated words. The help of pattern tables will generate with a different prefix, suffix, and infix but with the same root. So the data will bigger than before. Big data is one of the keys to success on the deep learning method, increasing the model's accuracy.

3.2.3 Experiment Scenario

The purposes of this experiment is to know how well the RNN can or the neural method can identify the morphologycal aspect of Arabic word rather than using rule based method.

After implementing the method on the flow chart Figure 3.1. The result will be compare with previous work Jabalin Application [21] which is using rule based method.

CHAPTER 4

IMPLEMENTATION , AND RESULT ANALYSIS

4.1 System Implementation

This section will explain the implementation result of all of the processes already explained in chapter 3, based on the flowchart in Figure 3.1 and Figure 3.2.

4.1.1 Data Preprocessing

This first process will be more focused on the Qur'an corpus raw data to get the Arabic word that matches with systems criteria. This implementation has two subprocesses which are preprocessing and generate processes.

Preprocessing

The primary purpose of this process is to get the main dataset for this system and make the gold standard of msd. The Qur'an corpus data has the msd information for each word, so that will be the system gold msd.

After the do the preprocessing, get 1778 unique words and just 15 different wazan or verb forms (see Table 4.1). All Arabic words are on Buckwalter form but will be changed into Arabic form later on in the implementation. In this process, already get the morphology features such as POS, *wazan* of verb form, and other features. The other features sample is NUM, GEN, aspect, and PER; for example, PERF, 2MS its means aspect perfective, PER = 2, NUM = Singular, and GEN = masculine. The primary purpose of this process is to get the main dataset for this system and make the gold standard of msd. The Qur'an corpus data has the msd information for each word, so that will be the system gold msd.

Generate

This process aims to expand the data to get a possible word with a different prefix, infix, and suffix with the help of a pattern table. This process also makes the actual gold standard for msd with the basic format used later in this system. After doing the word generate process, successfully get more words from 1772 words into 30936 unique words. (see Table 4.2). This total word is enough because mainly all of the other morphological research using mainly dozens of thousands of data.

Table 4.1: The result of preprocessing process from Quran corpus data, Position is verse, chapter and word position in the Qur'an, word, and root is in the Buckwalter form. Tag, *Wazan* and other features in the part of making msd gold later on.

POSITION	WORD	TAG	WAZAN	ROOT	Other Features/ MSD
(1:7:3:1)	>anoEamota	V	(IV) tsulatsy	nEm	PERF, 2MS
(2:5:8:2)	mufoliHuwona	N	(IV) tsulatsy	flH	MP
(1:5:2:1)	naEobudu	V	Iau	Ebd	IMPF, 1P
(2:3:7:1)	razaqona	V	Iau	rzq	PERF, 1P
(2:6:3:1)	kafaruwA@	V	Iau	kfr	PERF, 3MP
(2:7:1:1)	xatama	V	Iai	xtm	PERF, 3MS
(2:9:6:1)	yaxodaEuwna	V	Iaa	xdE	IMPF, 3MP
(2:9:10:1)	ya$oEuruwna	V	Iau	$Er	IMPF, 3MP
(2:10:12:1)	yako*ibuwna	V	Iai	k*b	IMPF, 3MP

4.1.2 Build RNN Model

The data that has been collected from the previous process is large enough, which is good to build an RNN out of it. The data is 30936 unique words, along with all of the other attributes attached to it. In this process, we only need the word and the *wazan* or verb form only because this model will be used as a verb form identification model. This process aims to get a pattern and subword vector from each word to be processed to build an RNN model to identify the correct verb form. After that, the data must be split into training and testing to evaluate the model's result using accuracy, precision, recall, and f1 score.

Pattern Extraction and Subword Vectorizing

After do the implentation pattern extraction using Algorithm 1 and subword vectorizing such as like the example of Table 3.4, adding 4 more columns in the data which are patterm, vector, and *wazan* index (see Table 4.3. The *wazan* index is number representation of *wazan* or verb form.

Split Dataset

This process is splitting the data into training and testing. See the Table 4.4 is the proportion and the total of the data that has been splitting.

Training and Testing

The model will be trained with 50 epochs and an Adam optimizer. After doing the training, it gets 99% accuracy into the data training itself. On the testing process, the

Table 4.2: This table results from generating processes from Quran corpus data with the help of a pattern table. The type of word and *dhamir* or pronoun is getting from the pattern table to help the following process in the system. MSD already in the proper format after this process

Arabic	Buckwalter	Root	Wazan	Type of Word	Dhamir	MSD
مَظْلُوْمٌ	maZoluwomN	Zlm/ ظلم	Iai	اِسِمْ مَفْعُوْلْ	هُوَ	{"POS": "N", "NUM": "SG", "GEN": "MASC", "Verb Form": "Iai"}
ظَالِمٌ	ZaAlimN	Zlm/ ظلم	Iai	اِسِمْ فَاعِلْ	هُوَ	{"POS": "N", "NUM": "SG", "GEN": "MASC", "Verb Form": "Iai"}
يَظْلِمُ	yaZolimu	Zlm/ ظلم	Iai	فِعِلْ مُضَارِعْ	هُوَ	{"POS": "V", "aspect": "IPFV", "tense": "PRS/FUT", "PER": "3", "NUM": "SG", "GEN": "MASC", "Verb Form": "Iai"}
ظَلَمَ	Zalama	Zlm/ ظلم	Iai	فِعِلْ مَاضِ	هُوَ	{"POS": "V", "aspect": "PFV", "tense": "PST", "PER": "3", "NUM": "SG", "GEN": "MASC", "Verb Form": "Iai"}
مَظْلُوْمَانِ	maZoluwomaAni	Zlm/ ظلم	Iai	اِسِمْ مَفْعُوْلْ	هُمَا	{"POS": "N", "NUM": "DU", "Verb Form": "Iai"}
مَظْلُوْمَيْنِ	maZoluwomayoni	Zlm/ ظلم	Iai	اِسِمْ مَفْعُوْلْ	هُمَا	{"POS": "N", "NUM": "DU", "Verb Form": "Iai"}
ظَالِمَانِ	ZaAlimaAni	Zlm/ ظلم	Iai	اِسِمْ فَاعِلْ	هُمَا	{"POS": "N", "NUM": "DU", "Verb Form": "Iai"}
ظَالِمَيْنِ	ZaAlimayoni	Zlm/ ظلم	Iai	اِسِمْ فَاعِلْ	هُمَا	{"POS": "N", "NUM": "DU", "Verb Form": "Iai"}
يَظْلِمَانِ	yaZolimaAni	Zlm/ ظلم	Iai	فِعِلْ مُضَارِعْ	هُمَا	{"POS": "V", "aspect": "IPFV", "tense": "PRS/FUT", "PER": "3", "NUM": "DU", "Verb Form": "Iai"}

Table 4.3: The result of pattern extraction and subword vectorizing process

Arabic	Pattern	Vector	Wazan	WazanIndex
مَظْلُوْمٌ	ma-ouwoN-	29, 17.0, 7.0, 23.0, 9.0, 24.0, 8.0, 1	Iai	10
ظَالِمٌ	-aAiN-	1, 17.0, 4.0, 23.0, 15.0, 24.0, 8.0, 1	Iai	10
يَظْلِمُ	ya-oiu-	35, 17.0, 7.0, 23.0, 15.0, 24.0, 12.0, 1	Iai	10
ظَلَمَ	-aaa-	1, 17.0, 5.0, 23.0, 14.0, 24.0, 9.0, 1	Iai	10
مَظْلُوْمَانِ	ma-ouwoa-Ani	29, 17.0, 7.0, 23.0, 9.0, 24.0, 9.0, 16	Iai	10
مَظْلُوْمَيْنِ	ma-ouwoa-yoni	29, 17.0, 7.0, 23.0, 9.0, 24.0, 9.0, 7	Iai	10
ظَالِمَانِ	-aAia-Ani	1, 17.0, 4.0, 23.0, 15.0, 24.0, 9.0, 16	Iai	10
ظَالِمَيْنِ	-aAia-yoni	1, 17.0, 4.0, 23.0, 15.0, 24.0, 9.0, 7	Iai	10
يَظْلِمَانِ	ya-oia-Ani	35, 17.0, 7.0, 23.0, 15.0, 24.0, 9.0, 16	Iai	10
ظَلَمَا	-aaa-A	1, 17.0, 5.0, 23.0, 14.0, 24.0, 9.0, 34	Iai	10

Table 4.4: The number of data after splitting the dataset

No	Type	Count	Portion
1	Training data	24748	80 %
2	Testing data	6188	20 %

model will be tested with 6188 words in the data test. After doing the testing, it gets the same result as training which is 99 % of accuracy and details as follow in Table 4.5.

The result is satisfactory enough to identify 99% of the data testing even though with the 6183 total data.

Table 4.5: Testing result

No	Evaluation	Result
1	Accuracy	99%
2	Precision	99%
3	Recall	96%
4	F1-score	97%

4.1.3 MSD Identification

The testing process is the final process of the entire system in this study. The purpose of this process is to get the final output which is msd. The process will be going through verb form identification, type of word and pronoun identification, and then msd identification.

The data will be used for data testing and going through the verb form identification process first on the verb form identification using RNN model to identify its verb form

or *wazan*. After that, the data will be going through the type of word and pronoun identification with the help of a pattern table using Algorithm 2 to find the correct table. Finally, after getting the type of word and pronoun using Algorithm 3 and the help of Table A.1 to produce msd.

After going through the process of msd identification, the result is 99 % accuracy, which is the same as building the RNN model from before. In the Tabel 4.6 can see the accuracy of all the msd is in the testing dataset. Overall is over than 95% accuracy even there is 100%, the only verb form Iii with 73% accuracy. This result is excellent enough because so close to 100%.

Table 4.6: The details of result accuracy base on the MSD

No	MSD Tag	MSD Value	Accuracy	Identified Total	Total Data
1	POS	V	98%	4018	4069
2	POS	N	99%	2115	2119
3	Aspect	IPFV	98%	1316	1335
4	Aspect	PFV	99%	1569	1573
5	Tense	PRS/FUT	98%	1316	1335
6	Tense	PST	99%	1569	1573
7	PER	1	97%	494	505
8	PER	2	98%	2324	2359
9	PER	3	99%	1200	1205
10	NUM	PL	99%	2340	2353
11	NUM	SG	98%	2056	2079
12	NUM	DU	98%	1737	1756
13	GEN	MASC	99%	2147	2160
14	GEN	FEM	99%	1755	1767
15	Verb Form	Iaa	99%	593	595
16	Verb Form	Iau	99%	1097	1098
17	Verb Form	Iia	99%	741	746
18	Verb Form	Iai	99%	766	771
19	Verb Form	Iii	73%	19	26
20	Verb Form	Iuu	99%	88	89
21	Verb Form	II	100%	737	737
22	Verb Form	III	100%	140	140
23	Verb Form	IV	100%	1184	1184
24	Verb Form	V	100%	376	376
25	Verb Form	VI	92%	35	38
26	Verb Form	VII	100%	58	58
27	Verb Form	IX	100%	55	55
28	Verb Form	X	100%	237	237
29	Verb Form	Q1	100%	38	38
30	Mood	IMP	97%	1133	1161

4.2 Jabalin Implementation

This study aims to get a better result and add more kinds of data to the dataset. In this study, the problem comes from Jabalin system [21] that cannot identify the Arabic noun. To prove this system is better than Jabalin, they must also compare the Jabalin system with this system. The idea is to implement the testing dataset into the Jabalin and see how the Jabalin result compares to this study system result.

4.2.1 Jabalin Online Interface

Jabalin system has not only the model but also the online application for the user to use (Figure 4.1. Users can access Jabalin online interface in here `http://elvira.lllf.uam.es/jabalin/analizarForma.php`. The idea to compare the result is using python web crawling to send the Arabic words and get the msd result.

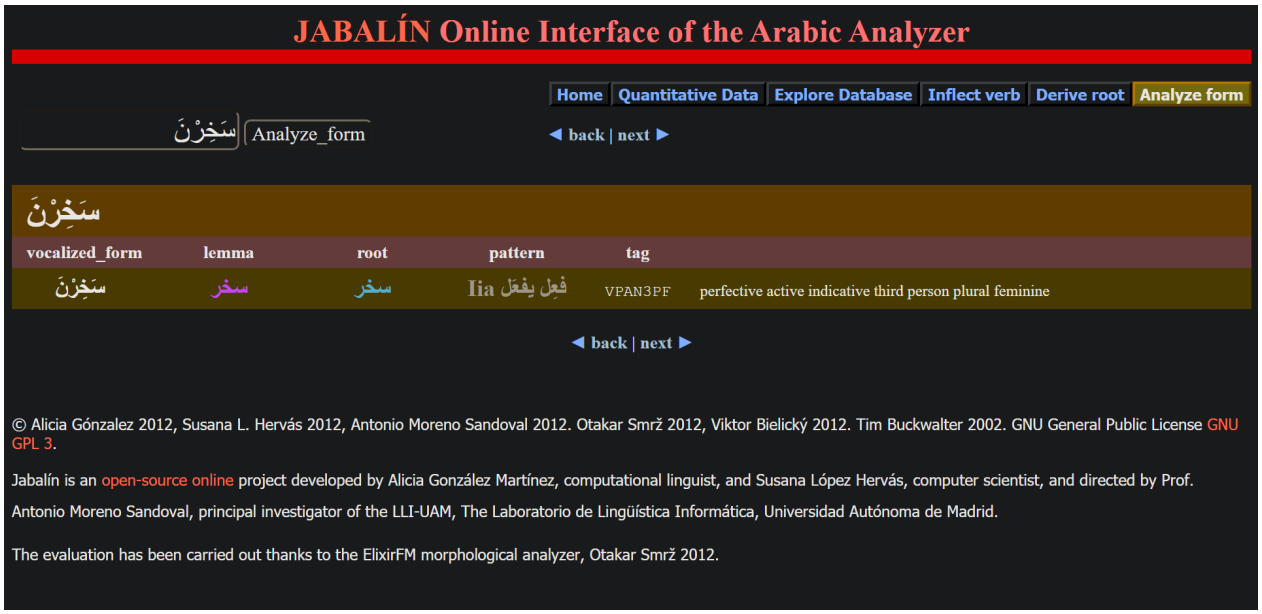

Figure 4.1: The Jabalin online interface for Arabic morphological analyzer

4.2.2 Jabalin MSD TAG Mapping

Before the crawler can get the msd result of each word in the testing data, the Jabalin system produces a different msd format. The Jabalin msd format must be translated or mapped to this system msd format. Table 4.7 show the sequence of a letter in the Jabalin system and the msd meaning of it. The msd is translated so the system can process the result like in the Table 4.6.

Table 4.7: MSD tag that produce by Jabalin system and translate into this system, the position starts from left to right

Position	Jabalin Tag	Example Value	Meaning	System Tag	System Value
1	POS	V	Verb	POS	V
2	Aspect	I	Imperfect	Aspect	IPFV
3	Voice	A	Active	-	-
4	Mood	I	Imperative	Mood	IMP
5	PER	3	Third Person	PER	3
6	NUM	P	Plural	NUM	PL
7	GEN	F	Feminine	GEN	FEM

4.2.3 Web Crawling Get the Result

The data testing shall be used to test Jabalin performance with the help of python web crawling. The testing is so slow because the data is so large and depends on an internet connection also. The result is get around 40 % with the detail in Table 4.8.

The result is low because there is data that cannot handle with the Jabalin system, which is the noun tag of Arabic. Can be seen in Table 4.8 the POS N is 0% accuracy and mood IMP is 0%. The best performance is identifying the first person type of Arabic, which is 94% of accuracy.

4.3 Result Analysis

It can be seen the result of both systems in Table 4.8 and Table 4.6. This system model is better than the Jabalin system in this testing dataset. This system gets 99% accuracy as a whole, and the Jabalin gets just 40% accuracy. Comparing the result per MSD between this system and the Jabalin, this system can identify the noun successfully, which is the limitation of the Jabalin system. The Jabalin system cannot identify POS nouns and mood IMP at all. However, the result is still better if compared with this study system. Look at the stats of accuracy per msd in both of the table results. The Jabalin system cannot win each MSD accuracy in the table.

The neural-based method, which in this study case is a recurrent neural network, can capture the information about the sequence of prefix, suffix, infix, and root in the Arabic word and help the model identify its morphological features. Not only can solve dictionary dependency problem and can identify noun but also make the result is better than Jabalin system with this study testing data. Even in this case, the recurrent neural network is just using a simple RNN model does not use LSTM or any memory added model into the deep learning structure. It is because the sequence of the sub-word is not too long. It is just 12 maximum lengths of a vector.

Table 4.8: Result Jabalin system per all of the MSD

No	MSD Tag	MSD Value	Accuracy	Identified Total	Total Data
1	POS	V	59%	2427	4069
2	POS	N	0%	0	2119
3	Aspect	IPFV	68%	913	1335
4	Aspect	PFV	39%	628	1573
5	Tense	PRS/FUT	68%	913	1335
6	Tense	PST	87%	1371	1573
7	PER	1	94%	475	505
8	PER	2	44%	1051	2359
9	PER	3	74%	901	1205
10	NUM	PL	35%	841	2353
11	NUM	SG	48%	998	2079
12	NUM	DU	33%	588	1756
13	GEN	MASC	19%	422	2160
14	GEN	FEM	4%	79	1767
15	Verb Form	Iaa	35%	211	595
16	Verb Form	Iau	35%	393	1098
17	Verb Form	Iia	34%	256	746
18	Verb Form	Iai	33%	256	771
19	Verb Form	Iii	3%	8	26
20	Verb Form	Iuu	7%	63	89
21	Verb Form	II	41%	303	737
22	Verb Form	III	41%	58	140
23	Verb Form	IV	40%	479	1184
24	Verb Form	V	40%	154	376
25	Verb Form	VI	47%	18	38
26	Verb Form	VII	31%	18	58
27	Verb Form	IX	27%	15	55
28	Verb Form	X	37%	89	237
29	Verb Form	Q1	34%	13	38
30	Mood	IMP	0%	0	1161

The RNN model or verb form identification process seems to one of the critical successes of the identifier. Because in that process (Figure 3.4) successfully build the RNN with 99% and the accuracy msd is also 99%. If the verb form is successfully identified, the word will be directed into the correct table. The msd must be right because that word is in the correct table to see the type of word and pronoun to identify the MSD.

The extensive dataset is also the critical success of the identifier. With the help of a pattern table, it can generate all possible inflicted and derived words with the same root. The model already got information of the root in the word that can be a reference to know the verb form.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusions

This study trying to make a morphological analyzer with neural-based approach. The input is one single Arabic word and the output is morphological features (MSD) of the word. The neural-based with just simple recurrent neural network method successfully identifies msd with the noun type of Arabic and solves dictionary dependency problems. The Arabic word has information of the sequence of sub-word, which are prefix, suffix, root, and suffix, to help identify the verb form. RNN is excellent for capturing that information. Not only be able to identify the requirement, but the model also makes better accuracy than the Jabalin system using its own data testing words.

5.2 Future Work

This system successfully identified the MSD with noun type of Arabic, but this system still has many limitations. For example, this system using nouns, but the noun with derived from verb only. If the noun is not derived from a verb, the system cannot identify it. Also, this system cannot process *mu'tal* verb, passive voice, and not all the verb form covered. In the following work, hope can handle such kind of that data, and with the help of neural-based, because of the result have seen in this study is good enough

BIBLIOGRAPHY

[1] G. n. N. Abdelhadi Soudi, Antal van den Bosch. *Arabic Computational Morphology*. Text, Speech and Language Technology. Springer, 2007. ISBN 9781402060458.

[2] U. R. Abu Razin. *Ilmu Nahwu Untuk Pemula*. Cetakan III. Pustaka BISA, 2017. URL `https://dewisaputri.files.wordpress.com/2016/02/ebook-ilmu-nahwu-untuk-pemula.pdf`.

[3] U. R. Abu Razin. *Ilmu Sharaf Untuk Pemula*. Cetakan III. Pustaka BISA, 2017. URL `https://dewisaputri.files.wordpress.com/2016/02/ebook-ilmu-nahwu-untuk-pemula.pdf`.

[4] C. C. Aggarwal et al. Neural networks and deep learning. *Springer*, 10:978–3, 2018.

[5] M. N. Al-Kabi, S. A. Kazakzeh, B. M. A. Ata, S. A. Al-Rababah, and I. M. Alsmadi. A novel root based arabic stemmer. *Journal of King Saud University-Computer and Information Sciences*, 27(2):94–103, 2015.

[6] K. R. Beesley. Finite-state morphological analysis and generation of arabic at xerox research: Status and plans in 2001. In *ACL Workshop on Arabic Language Processing: Status and Perspective*, volume 1, pages 1–8, 2001.

[7] G. Booij. *The grammar of words: An introduction to linguistic morphology*. Oxford University Press, 2012.

[8] E. K. Brown and J. E. Miller. *Concise encyclopedia of syntactic theories*. Pergamon Press, 1996.

[9] L. Cahill. A syllable-based account of arabic morphology. In *Arabic Computational Morphology*, pages 45–66. Springer, 2007.

[10] R. Cotterell, C. Kirov, J. Sylak-Glassman, G. Walther, E. Vylomova, A. D. McCarthy, K. Kann, S. Mielke, G. Nicolai, M. Silfverberg, et al. The conll–sigmorphon 2018 shared task: Universal morphological reinflection. *arXiv preprint arXiv:1810.07125*, 2018.

[11] K. Dukes. The quranic arabic corpus, 2017. URL `http://corpus.quran.com/`.

[12] W. Feng, N. Guan, Y. Li, X. Zhang, and Z. Luo. Audio visual speech recognition with multimodal recurrent neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 681–688. IEEE, 2017.

[13] D. Graff, M. Maamouri, B. Bouziri, S. Krouna, S. Kulick, and T. Buckwalter. Standard arabic morphological analyzer (sama). *Linguistic Data Consortium LDC2009E73*, 2010.

[14] N. Habash. *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2010. ISBN 9781598297966. URL `https://books.google.co.id/books?id=nZtdAQAAQBAJ`.

[15] N. Habash and O. Rambow. Magead: a morphological analyzer and generator for the arabic dialects. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 681–688, 2006.

[16] S. Hidayatullah. *Cakrawala Linguistik Arab*. Grasindo, Jakarta Indonesia, 2017.

[17] Z. Karimatanisak et al. *Fi'il Shohih dalam Kitab Al-Akhlaq Lil Banaat Jilid 2 (Analisis Morfologis)*. PhD thesis, Universitas Negeri Semarang, 2015.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] C. Kirov, R. Cotterell, J. Sylak-Glassman, G. Walther, E. Vylomova, P. Xia, M. Faruqui, S. Mielke, A. D. McCarthy, S. Kübler, et al. Unimorph 2.0: universal morphology. *arXiv preprint arXiv:1810.11101*, 2018.

[20] D. C. Kurniawan, N. Nababan, and R. Santosa. Dhomir on the moses story in al qur'an suraa at-thaha. In *Proceeding of International Conference on Art, Language, and Culture*, volume 2, pages 243–248, 2017.

[21] A. G. Martínez, S. L. Hervás, D. Samy, C. G. Arques, and A. M. Sandoval. Jabalín: a comprehensive computational model of modern standard arabic verbal morphology based on traditional arabic prosody. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 35–52. Springer, 2013.

[22] A. Mottini and R. Acuna-Agost. Relative label encoding for the prediction of airline passenger nationality. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 671–676. IEEE, 2016.

[23] S. Nasution. *Pengantar Linguistik Bahasa Arab*. 1. Lisan Arabi, Sidoarjo East Java Indonesia, 3 edition, 2 2017. ISBN 978-602-70113-8-0.

[24] A. A. Neme. A lexicon of arabic verbs constructed on the basis of semitic taxonomy and using finite-state transducers. 2011.

[25] T. I. Ramadhan, M. A. Bijaksana, and A. F. Huda. Rule based pattern type of verb identification algorithm for the holy qur'an. *Procedia Computer Science*, 157:337–344, 2019.

[26] K. F. Shaalan, M. Attia, P. Pecina, Y. Samih, and J. van Genabith. Arabic word generation and modelling for spell checking. In *LREC*, pages 719–725, 2012.

[27] O. Smrž. 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, 2007.

[28] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009.

[29] A. Soudi, V. Cavalli-Sforza, and A. Jamari. A computational lexeme-based treatment of arabic morphology. In *Proceedings of the Arabic Natural Language Processing Workshop, Conference of the Association for Computational Linguistics (ACL 2001)*, pages 50–57, 2001.

[30] E. Vylomova, J. White, E. Salesky, S. J. Mielke, S. Wu, E. Ponti, R. H. Maudslay, R. Zmigrod, J. Valvoda, S. Toldova, et al. Sigmorphon 2020 shared task 0: Typologically diverse morphological inflection. *arXiv preprint arXiv:2006.11572*, 2020.

[31] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola. *Dive into Deep Learning*. 2020. `https://d2l.ai`.

Appendices

APPENDIX A

MISCELLANEOUS

Table A.1: Mapping the Arabic features into the MSD

No	Arabic Features	Type	MSD
1	فِعِلْ مَاضِ	Type of Word	- POS:V - aspect:PFV - tense:PST
2	فِعِلْ مُضَارِعْ	Type of Word	- POS:V - aspect:IPFV - tense:PRS/FUT
3	مَصْدَرْ	Type of Word	- POS:N
4	اِسِمْ فَاعِلْ	Type of Word	- POS:N
5	اِسِمْ مَفْعُوْلْ	Type of Word	- POS: N
6	فِعِلْ الأَمرْ	Type of Word	- POS: V - mood: IMP
7	فِعِلْ النَهْيِ	Type of Word	- POS: V - mood: IMP
8	هُوَ	Pronoun	- PER:3 - NUM:SG - GEN:MASC
9	هُمَا	Pronoun	- PER:3 - NUM:DU
10	هُمْ	Pronoun	- PER:3 - NUM:PL - GEN:MASC
11	هِيَ	Pronoun	- PER:3 - NUM:SG - GEN:FEM
12	هُنَّ	Pronoun	- PER:3 - NUM:PL - GEN:FEM

13	أَنْتَ	Pronoun	- PER:2 - NUM:SG - GEN:MASC
14	أَنْتُمْ	Pronoun	- PER:2 - NUM:PL - GEN:MASC
15	أَنْتِ	Pronoun	- PER:2 - NUM:SG - GEN:FEM
16	أَنْتُمَا	Pronoun	- PER:2 - NUM:DU
17	أَنْتُنَّ	Pronoun	- PER:2 - NUM:PL - GEN:FEM
18	أَنَا	Pronoun	- PER:1 - NUM:SG
19	نَحْنُ	Pronoun	- PER:1 - NUM:SG