ON FEATURE AUGMENTATION FOR SEMANTIC ARGUMENT CLASSIFICATION OF THE QURAN ENGLISH TRANSLATION USING SUPPORT VECTOR MACHINE

A THESIS SUBMITTED TO
THE SCHOOL OF COMPUTING

BY
DINA KHAIRA BATUBARA
2301140026

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

TELKOM UNIVERSITY
2017

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 _____

_____

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 _____

_____ _____

M. Arif Bijaksana, Ph.D
Supervisor

Prof. Dr. Adiwijaya
Co-Supervisor

Examining Committee Members,
(Dr. Arief Fathul Huda) (Chairperson of the jury) : _____

(Kemas Muslim Lhaksmana, Ph.D) (Jurys member) : _____

(Dana Sulistyo Kusumo, Ph.D) (Jurys member) : _____

SELF DECLARATION AGAINST PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Date/Month/Year:

Name, Last Name: **Dina Khaira Batubara**

Signature : _____

Date/Month/Year:

Name, Last Name of first Supervisor: **M. Arif Bijaksana, Ph. D**

Signature : _____

Name, Last Name of second Supervisor: **Prof. Dr. Adiwijaya**

Signature : _____

ABSTRACT

Semantic Argument Classification is the process of analyzing the sentence to investigate the pattern of WHO did WHAT to WHOM, WHEN, WHERE, WHY, HOW, from a structured text data. Research on the classification of semantic arguments requires data that has been labeled semantically in large numbers, which is called corpus. In the preliminary research, two types of corpus have been built, namely FrameNet and Propbank, both are from news domain or news genre. Because building a corpus is costly and time-consuming, recently many studies have used FrameNet and Propbank corpus as training data to conduct semantic argument classification research on new domains without the need to build a new corpus for those new domains.

This thesis will perform a research related to semantic argument classification on a new domain that is Quran English Translation by utilizing Propbank corpus as training data. The Quran English translation is a translation of the original Arabic Quran. Hence the composition of grammar and the sentence structure, English-Quran is still influenced by the original languages, namely Arabic. In its original language, Arabic, the holy Quran has a significant difference from the newswire domain, being closer to poetic language, more creative linguistic expression, and has many variations of vocabulary and sentence structure.

Previous studies have proven that there is a significant decrease in performance when classifying semantic arguments on different domain between the training and the testing data. The main problem is when there is a new argument that found in the testing data but not found in the training data. To recognize this new argument in training data, one solution is by extending the argument features in the training data to accommodate the new features of the new argument. This thesis proposes four new features to improve the baseline system performance.

By using SVM Linear, the experiment has proven that the performance of semantic argument classification on Quran data using Propbank Corpus as training data can be improved by augmenting the proposed features to the baseline system with some combination option. When tested on auto labeled data, the augmentation of PTO+SP features to the baseline system can improve the accuracy by 1.25% and improve F-1 score by 1.30%. When tested on hand-labeled data, the augmentation of combination PO+PTO features to the baseline system can improve the accuracy by 0.47% and improve F-1 score by 0.40%.

Keywords: semantic argument classification, semantic role labeling, shallow semantic parsing, classification algorithm, Support Vector Machine classifier.

ABSTRAK

Klasifikasi argumen semantik adalah proses menganalisa kalimat untuk menyelidiki pola WHO did WHAT to WHOM, WHEN, WHERE, WHY dan HOW dari struktur data teks. Penelitian terkait klasifikasi argumen semantik memerlukan data yang telah diberi label semantik dalam jumlah yang besar, yang disebut korpus. Pada penelitian awal, telah dibangun dua jenis korpus yaitu FrameNet dan Propbank, keduanya merupakan domain atau aliran berita. Karena membangun korpus membutuhkan biaya yang besar dan waktu yang lama, maka beberapa tahun terakhir telah banyak penelitian yang memanfaatkan korpus FrameNet dan Propbank sebagai data pelatihan untuk melakukan penelitian klasifikasi argumen semantik pada domain yang baru tanpa perlu membangun korpus untuk domain baru tersebut.

Thesis ini akan melakukan penelitian terkait klasifikasi argumen semantik pada domain baru yaitu translasi AlQuran dalam bahasa Inggris dengan memanfaatkan korpus Propbank sebagai data latih. AlQuran dalam bahasa Inggris adalah alih bahasa dari AlQuran asli berbahasa Arab. Oleh karena itu dalam tata bahasa dan struktur kalimat, bahasa dalam translasi Quran berbahasa Inggris masih dipengaruhi oleh bahasa aslinya, yaitu bahasa Arab. Di dalamnya bahasa aslinya yaitu bahasa Arab, AlQuran memiliki perbedaan yang signifikan dari domain berita, lebih dekat dengan bahasa puitis, memiliki ekspresi linguistik yang lebih kreatif, dan memiliki banyak variasi kosa kata dan struktur kalimat.

Peneitian terdahulu telah membuktikan bahwa terjadi penurunan performansi secara signifikan ketika melakukan klasifikasi argumen semantik pada domain yang berbeda antara data latih dan data uji. Masalah utamanya adalah karena ditemukan argumen baru yang terdapat pada data uji namun tidak ditemukan dalam data latih. Untuk mengenali argumen baru ini dalam data latih, salah satu solusinya adalah dengan memperluas fitur argumen dalam data latih untuk mengakomodasi fitur baru dari argumen baru tersebut. Thesis ini mengusulkan penambahan empat fitur baru pada sistem baseline untuk meningkatkan kinerja sistem.

Dengan menggunakan SVM Linear, percobaan telah membuktikan bahwa performansi klasifikasi argumen semantik pada domain Quran menggunakan data Propbank sebagai data training dapat ditingkatkan dengan penambahan fitur yang diusulkan pada sistem baseline dengan dengan beberapa pilihan kombinasi. Ketika diuji pada auto labeled data, penambahan fitur PTO+SP dapat meningkatkan akurasi sebesar 1.25% dan F-Measure sebesar 1.30%. Ketika diuji pada hand-labeled data, penambahan fitur PO+PTO dapat meningkatkan akurasi sebesar 0.47% dan F-Measure sebesar 0.40%.

Keywords: semantic argument classification, semantic role labeling, shallow semantic parsing, classification algorithm, Support Vector Machine classifier.

DEDICATION

With all the praise due to Allah SWT, I dedicate this study to the teachers of my life, my beloved parents Alm. H. Nazaruddin Batubara, and Hj. Sundariwaty Siregar, who have taught me many things and always pray for me. My partner in life, Bayu Irawan, thanks for the understanding and the full support when I was finishing this study. My kids, Farah Khaulah Salsabila, M. Arham Rasydan Abdurrahman, Salman Faris Musyaffa and Shafiya Naira Imanina. Thanks for your understanding and have been praying for me ceaselessly. I apologize for being unable to care all of you because I have to finish this study immediately. I wish you will be a smart children and can achieve what has been dreamed of in the future.

ACKNOWLEDGEMENT

Bismi-llāhi rraḥmāni rraḥīmi. In the name of Allah, the Most Gracious, the Most Merciful. Al-ḥamdu lillāh, all the praises due to Allah, for His blessing so that I can finish this thesis. Firstly, I would like to express my sincere gratitude to PT. Telkom Indonesia for the scholarship program and the amazing support during completion this study. I would like to express my sincere gratitude and appreciation to Mr. M. Arif Bijaksana Ph.D. and Prof. Adiwijaya for the continuous guidance and support throughout the research and writing the thesis. High appreciation is also dedicated to Mr. Dana S. Kusumo Ph.D. as The Head of Graduate School of Informatics Engineering and the whole team in Telkom University Graduate School of Informatics Engineering for the support and assistance during completion of my study. My sincere thanks also to the whole team in PT. Telkom Riau Daratan, especially for Mr. Deddy Riswandy and Mr. Ilham from Access Maintenance & Optima, Mr. Joni Hardi and Mr. Binsar Simanjuntak from Network Area, for the full support and give me opportunities to finish the scholarship program. And the last but not least, my sincere thanks to my fellow students in PJJ Telkom of 2014 for all the discussions and the fun we had in the last three years, especially for my housemates Rizkiana Amalia, for being a friend of encouragement and discussion partners, for Mba Eka Pura Hartati and Dewi Riyanti for being very helpful friends, for motivating each other and for sharing the moments during completion our study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name : Batubara, Dina Khaira

Nationality : Indonesia

Place, Date of Birth : Pematangsiantar, 4 Oktober 1981

Marital Status : Married

Phone : +62 8115359000

Email : khairadina@gmail.com

Education Degree	**Institution**	**Year of Graduation**
Master	Telkom University	2017
Bachelor	STMIK-AMIK Riau	2007
High School	SMK TELKOM Sandhy Putra Medan	1999
Junior High School	SMPN 1 Pematangsiantar	1996

CHAPTER 1

INTRODUCTION

1.1. Rationale

Semantic Argument Classification is the process of analyzing the sentence to investigate the pattern of WHO did WHAT to WHOM, WHEN, WHERE, WHY, HOW, from a structure text. Semantic argument classification is also referred to as semantic role labeling (SRL) process. This process will extract information from a sentence or text data. Originally to build a good performance SRL system for a domain is done by semantic labeling process on a large data manually. The representation of SRL has many benefits in natural language processing (NLP) application and has proven to be useful for questions and answers, information extraction, machine translation, and so on.

Research on semantic argument classification requires data that has been labeled semantically in large numbers, called corpus. In the preliminary research, two types of corpus have been built, namely FrameNet [1] and Propbank [2], both are from news domain or news genre. Because building a corpus is costly and time-consuming, recently many studies have used FrameNet and Propbank corpus as training data to conduct semantic argument classification research on new domains such as domain literature, biomedical abstract, spoken of data, social media data, etc without the need to build a new corpus for those new domains [1] [2].

Those studies were done by adaptation of data on a new domain or test data to source domain or training data through the model formation, feature development, utilizing machine learning, etc [3]. This adaptation process requires a large amount of training data that has been labeled in large quantities, and little data has been labeled for target data. This problem is classified as a supervised problem. There are also

studies that do not require data that has been labeled on data testing, which is classified as an unsupervised problem.

Previous studies have proven that there is a significant decrease in performance when classifying semantic arguments on different domain between the training and the testing data. Despite existing SRL system which has been tested and worked well on a sentence from the same domain, it showed a sharp decline in performance when tested on a different domain [3].

Table 1. Performance of Propbank Data Trained on Intra and Quran Domain

	Propbank	Tested on Quran	
	Tested Intra Domain	Auto Labeled	Hand Labeled
Accuracy	98.30	81.92	87.40
Precision	98.30	83.20	89.30
Recall	98.30	81.90	87.40
F-Measure	98.30	82.10	88.10

The main problem on the argument classification task with different domain is when there is a new argument that found in the testing data but not found in the training data. In terms of sentence structure, there are some verses of the Quran that have a different pattern with the sentence news text, which allows the emergence of new arguments. To recognize the new argument in training data, extending the argument features in the training data to accommodate the new features of the new argument becomes one of the solutions.

This thesis will perform a research related to semantic argument classification on a new domain that is Quran English Translation. The Quran English translation is a translation of the original Arabic Quran. Hence the composition of grammar and the sentence structure, English-Quran is still influenced by the original languages, namely Arabic. In its original language, Arabic, the holy Quran has a significant difference from the newswire domain, being closer to poetic language, more creative linguistic expression, and has many variations of vocabulary and sentence structure.

For a case study, this thesis selected the verses with predicate “say” as the samples. The basic of choosing this keyword because “say” is one of the many

emerging predicates that called as much as 1722 times. In addition, compared to the other predicate, a sentence with predicate phrase “say” has a variety of more complex patterns. So it would be better if it is used as a reference.

Table 2. Some Differences Between Propbank and Quran Sentences With Predicate "say"

Argument	Propbank	Quran
Arg0	I, You, They, We, He, She, It	those who remained behind rejoiced in their staying [at home]
	John, Mary, The man, The cat	those who disbelieve in allah and his messengers and wish to discriminate between allah and his messengers
	The beautiful woman, The green house	those upon whom the word will have come into effect
		the one who was freed and remembered after a time
		who were given a portion of the scripture who believe in superstition and false objects of worship
		the eminent among his people who disbelieved and denied the meeting of the Hereafter while We had given them luxury in the worldly life
		a believing man from the family of Pharaoh who concealed his faith
ArgM-TMP	in March, on Sunday, at twelve, in 1982	when their eyes are turned toward the companions of the fire
	today, yesterday, last week, last year	when the punishment descended upon them
		when the hypocrites and those in whose hearts was disease
		when he wanted to strike the one who was an enemy to both of them
Arg2	to Anni, to him, to you	to the one whom he knew would go free
	john said to mary youre an idiot	for those who disbelieve
		to one who gives you a greeting of peace
		to those who associated others with allah
		to those who had wronged taste the punishment of eternity
		to the one on whom Allah bestowed favor and you bestowed favor
Arg1	what you have to understand is that philly literally stinks ?	who revealed the scripture that moses brought as light and guidance to the people ?
		who has forbidden the adornment of allah which he has produced for his servants and the good lawful things of provision?
		who provides for you from the heaven and the earth ?
		When we have become dust as well as our forefathers, will we indeed be brought out [of the graves]?
		When is [the fulfillment of] this promise, if you should be truthful?
Predicate	A lorillard spokeswoman said this is an old story .	Said Pharaoh, "And what is the Lord of the worlds?"
		Say, "He is Allah who is one ".

Information :

ARG0 : Argument-0, the subject/who said

ARG1 : Argument-1, the object/the utterance

ARG2 : Argument-2, the patient/to whom said

ARGM-TMP : Argument-Temporal, explaining about time, circumstances at the time of the event

Motivated by novel domain adaptation approach, this research will attempts improve the baseline system by the augmented four novel features to deal with the new argument on testing data. The system requires training a large of text labeled in the training data and a little of text labeled in the testing data. This research will use the Propbank corpus data as training data and the English translation of Quran as testing data. The feature augmentation performs an adaptation from the Quran domain to the Propbank domain. The adaptation process by detecting the important features contained in the Quran domain. By taking advantage of existing corpora from newswire domain will significantly reduce system development costs for Quran domain.

1.2. Theoretical Framework

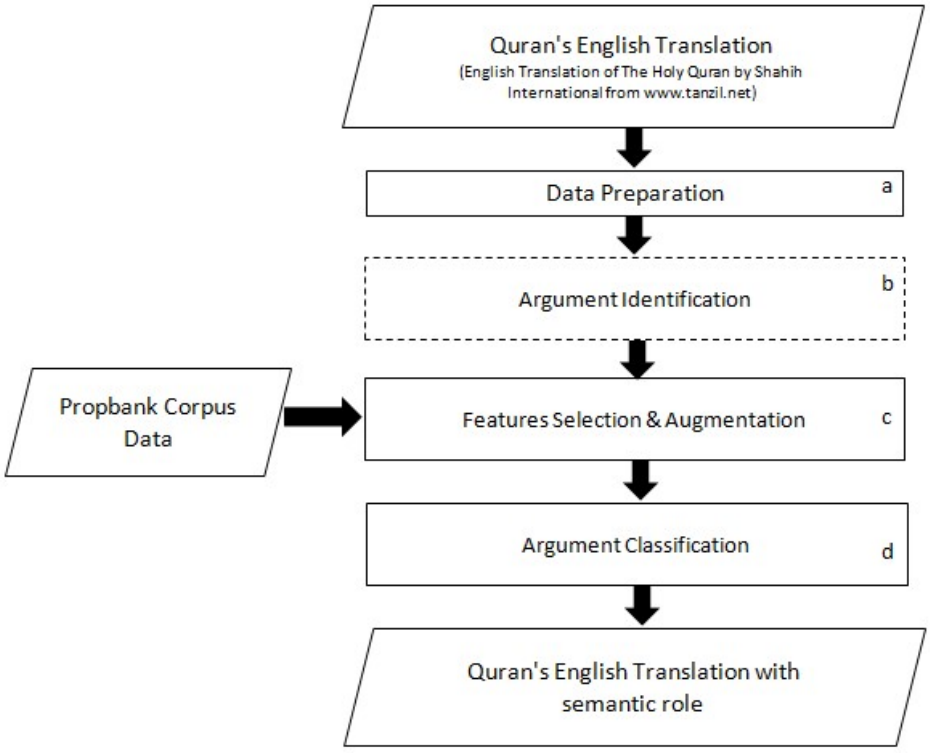

Figure 1. Theoretical Framework

Some theories in Figure 1 as a reference in this thesis is as follows:

a. Data Preparation

The training data are the PropBank corpus and the test data are the English translation of the Holy Qur'an. In data preparation process, the XML file for the English translation of the Qur'an in according to the rules of Propbank's frameset XML file is constructed [4]. Some information namely the sentence, the predicate and the arguments will be extracted from the XML file.

For example, sentence "[Satan] said, "I am better than him You created me from fire and created him from clay", it can be extracted that the predicate is "said", and the argument is "Satan" as argument 0 (ARG0/subject) and "I am better than him You created me from fire and created him from clay" as argument 1 (ARG1/object /the utterance).

b. Argument Identification

The first step in semantic argument classification is identify all arguments in a sentence based on the predicate. Argument identification conducts by parsing the sentence into a syntactic parse tree, both for training and testing data. The parse tree decompose words in a sentence into part-of-speech tags and nodes with the syntactic category.

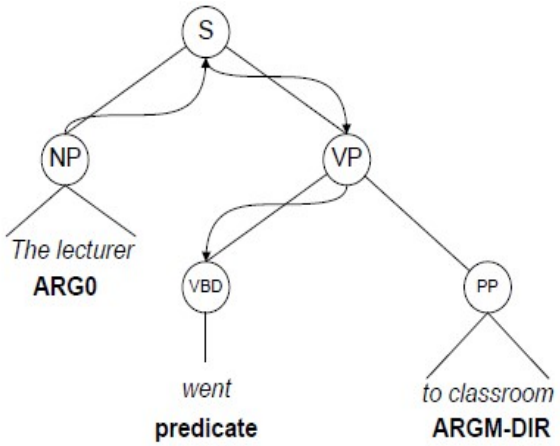

Figure 2. Parse Tree

Argument identification is the process to find the boundaries for all the arguments on a sentence. Argument identification is the process of determining which nodes are in a parse tree that is an argument or not, including their boundaries. For example from Figure 2, argument identification is to identify if "The lecturer", "went" and "to classroom" are arguments. In this thesis, this process will be done with two scenarios:

1. Identifying arguments on test data automatically using practNLPTools[1]. In this thesis, this type of data is called as auto labeled data.
2. Identifying arguments on test data manually by the author. In this thesis, this type of data is called as hand-labeled data or manually labeled data.

For testing the feasibility of practNLPTools to identify arguments in this thesis, the manually validation process is carried out by matching the result between auto labeled data and hand-labeled data then the accuracy is calcuated. Because the data are abundant, with consideration of data variation, this study selects four chapters from the Qur'an as a sample. The four chapters are Al-Baqarah, Al-Maidah, Yusuf, and Al-Qashash. These four chapters contain 947 arguments or 36.90% of all arguments in the Quranic data. Validation results obtained are as follows:

Table 3. Accuracy of Quran's Argument Identification Process by practNLPTools (%)

Row Labels	ARG0	ARG1	ARG2	ARG3	ARGM ADV	ARGM DIS	ARGM MOD	ARGM NEG	ARGM PNC	ARGM TMP	rel	Accuracy
Yusuf	100.00	95.83	25.00			100.00				15.38	100.00	92.77
Al-Baqarah	88.42	72.50	33.33	100.00	100.00	100.00	100.00	100.00	100.00	25.00	99.17	82.32
Al-Maidah	86.67	75.44	66.67		100.00	100.00	100.00	100.00		25.00	94.74	84.44
Al-Qashash	95.12	84.09	25.00			100.00	100.00			20.00	100.00	86.75
Accuracy	92.46	80.55	37.93	100.00	100.00	100.00	100.00	100.00	100.00	20.93	98.64	86.03

From the validation results in Table 3, it can be seen that the overall process of identification argument is good for it has the accuracy of 86.03%. Therefore it can be stated that practNLPTools is qualified to be a tool in identifying the argument in this study.

For ARG0, it is very good because it achieves 92,46% while for ARG1 is 80,55%. The accuracy of these two arguments still can be increased when tested on

[1] https://pypi.python.org/pypi/practNLPTools/1.0

more data. The accuracy is quite low in the ARG2 and ARGM-TMP argument namely > 50%, this is because there are a significant differences between Propbank and Quran on these two arguments (see Table 2).

c. Features Selection and Augmentation

In this thesis, the features used are divided into three categories; the basic features, additional features, and proposed features. The basic features are a set of features that are commonly used for semantic argument classification research. While additional features are a set of features that are added to the system to increase the performance. The basic features and additional features are used by adopting from previous research [4] [5]. Proposed features are a set of novel features developed from basic and additional features and augmented to the baseline system to deal with the new argument on the testing data. These features are expected to improve the performance of the system.

d. Argument Classification

The final step is the argument classification. It is the process to determine the semantic roles for all argument nodes. In the example above, the first identified node "The lecturer" labeled as ARG0, node "went" as predicate and node "to classroom" as ARGM-DIR. For the semantic role in general, ARG0 refers to the agent and ARG1 refers to the theme of the predicate. While for the semantic role from ARG2-5, there are no common meaning that remains consistent in the different predicate.

1.3. Conceptual Framework

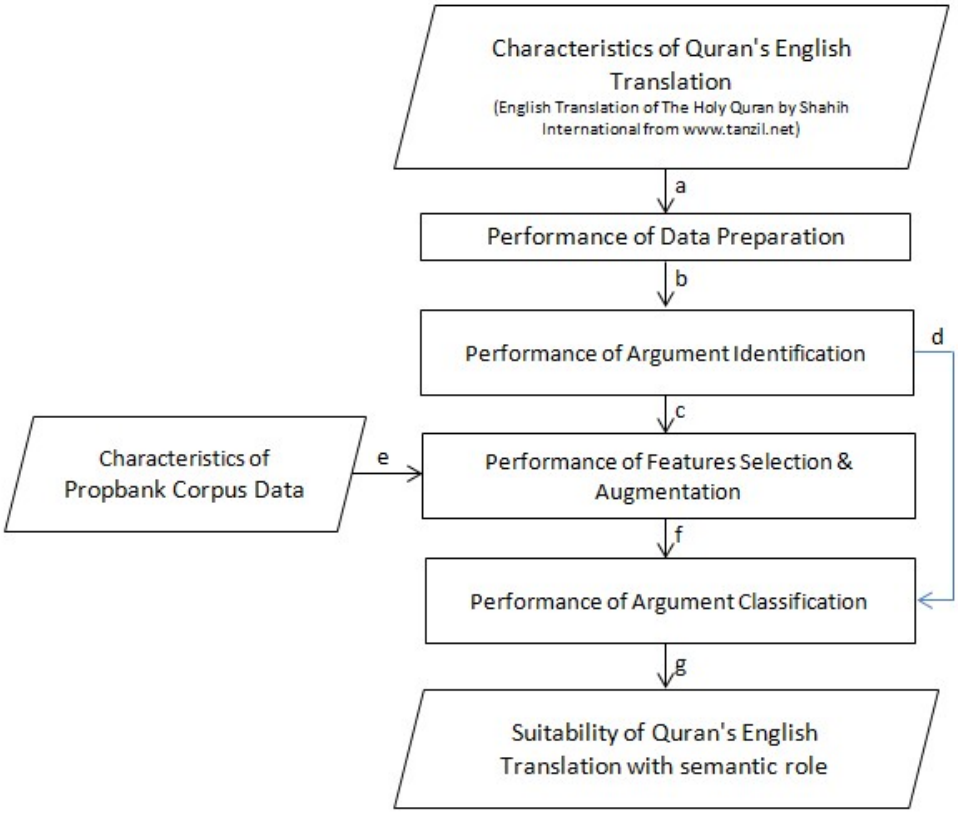

Figure 3. Linkage Between Variables in Conceptual Framework

a. Characteristics of Quran's English Translation and Performance of Data Preparation Process.

The number of sentences (verse) can affect the level of performance of the data preparation process. More sentences mean the data preparation process will be longer.

b. The performance of Data Preparation and Performance of Argument Identification Process.

Accuracy in sentences tokenization process in data preparation will affect the accuracy of argument identification process.

c. The performance of Argument Identification and Performance of Features Selection & Augmentation Process.

The number of arguments can affect the level of performance of features selection and augmentation process. More variety of arguments will produce a good data representation. The accuracy of the argument identification process will affect the accuracy of feature extraction generated.

d. The performance of Argument Identification and Performance of Argument Classification.

The accuracy of the argument identification process will affect the accuracy of argument classification process. Error in the process of identification of the argument will produce an error for the next process that is the classification.

e. Characteristics of Propbank Corpus Data and Performance of Features Extraction & Augmentation Process.

The number of arguments can affect the level of performance of features selection and augmentation process. More variety of arguments will produce a good data representation.

f. The performance of Features Selection & Augmentation and Performance of Argument Classification.

The accuracy of features selection & augmentation process will affect the accuracy of argument classification process. The suitability of the selected features with the characteristics of the data will produce a good accuracy of the classification process.

g. The performance of Argument Classification and Suitability of Quran's English Translation with the semantic role.

The performance of Argument Classification process will result in good suitability in the process of classification of semantic roles in Quran data.

1.4. Statement Of The Problem

Statement of the problem can be stated as follows:

- How to improve the performance of semantic argument classification on Quran domain especially on ARG0, ARG1, ARG2, and ARGM-TMP using Propbank as

training data by augmented four new features to deal with the new argument on Quran domain?

1.5. Objectives and Hypothesis

According to the problem statement, the objective of this thesis is to improve the performance of semantic argument classification on Quran domain by minimizing the mismatches of an argument between Propbank and Quran domain. The process is carried out by augmenting four new features to deal with a new argument on Quran data.

Some previous studies showed the decline in performance when testing is conducted on a different domain [3]. The main problem on the argument classification task with different domains is when there are a new arguments found in the testing data but they are not found in the training data. As a reference for this study, the hypotheses of this thesis is as follows:

- The four new features augmented to deal with the new argument on Quran domain data will minimize the mismatch between Propbank and Quran domain data especially on ARG0, ARG1, ARG2, and ARGM-TMP that ultimately will improve the performance of semantic argument classification on Quran domain.

1.6. Assumption

In this thesis, there are some assumptions as follows:

1. This thesis focuses on improving the performance of argument classification, excluding the process of argument identification. Therefore the process of argument identification is assumed to have been done before.
2. From validation process on practNLPTools's performance, it can be seen that the overall process of identification argument is quite good with the accuracy of 86.03%. Therefore it can be assumed that practNLPTools is qualified to be a tool in identifying the argument in this study.

3. The focus of this thesis is to improve the performance of four arguments i.e. ARG0, ARG1, ARG2, and ARGM-TMP, which have many discriminative features between Propbank and Quran and achieved poor performance when they are tested on Quran domain data. Therefore it can be assumed that the other arguments already have a good performance.

1.7. Scope And Delimitation

The scope of this thesis as follows:

1. This thesis focuses on improving the performance of argument classification, excluding the process of argument identification. Therefore the process of argument identification is carried out manually and uses the existing Semantic Role Labeling system.
2. The target domain data are the English Quran's translation Quran by Ministry of Religious Affairs downloaded through the website Tanzil – Quran Navigator[2] with the predicate "say" which is labeled refer to Propbank annotation rule.
3. The focus of this thesis is on improving the performance of four arguments i.e. ARG0, ARG1, ARG2, and ARGM-TMP, which have many discriminative features between Propbank and Quran and have poor performance when they are tested on Quran domain data.

1.8. Importance Of The Study

The importance of this study is as one of the preliminary research that implementing semantic argument classification or semantic role labeling (SRL) system in the Quran domain. Nowadays, the study on SRL system in Quran Domain is still very limited, so hopefully, this research can give a positive contribution in SRL system in general and especially in the study of the Quran domain.

[2] www.tanzil.net

CHAPTER 2

REVIEW OF LITERATURE AND STUDIES

2.1. Related Literatures

2.1.1. Text Mining

In general, text mining can be defined as an activity to extract a collection of documents (corpus) to obtain useful information by using a variety of analysis tools. Similarly, with data mining that has the aim to extract information from data, text mining also has the same goal which is done through an interesting search pattern. However, there are fundamental differences between text mining and data mining in source data. Sources data used in text mining are only a textual data which does not have a structure like the data in general. Text mining itself has similarities to data mining, they are equally dependent on preprocessing routine, pattern discovery algorithms and the presentation layer elements are used to improve search answer sets [6].

Text mining is the process of detecting and extracting information from unstructured text. The process includes information retrieval and lexical analysis. The ultimate goal is to transform unstructured text into data that can be analyzed using analytical methods.

2.1.2. Semantic Argument Classification

An argument is a statement, reasons or facts or also referred as a noun phrase to form propositions together with the predicate. This can also be interpreted that an argument is a piece of information that complements a predicate in a sentence which is usually a subject, direct object, and indirect object or referred as the words that follow or complement the predicate of a sentence. While semantics is the part of

language structure which is related to the meaning of the phrase or the meaning of a language structure [7]. Therefore it can be defined that the important task of a semantic argument in a sentence is to answer questions such as what, who, when, where, how, why and how.

Figure 3 below presents the simplest semantic role labeling algorithm. Although there are a large number of algorithms, it basically uses the steps in this algorithm.

Most algorithms begin with the analysis of the earliest semantic role (Simmons, 1973), starting with parsing using a wide-coverage parser to decide parse to the input string. Parse then get through the parse tree to find the predicate word. For each of these predicates, the algorithm verifies each node in the parse tree and determines the semantic role (if any) that is played for this predicate. For a supervised classification, given a set of labeled training data like PropBank, the feature vector is extracted for each node, using baseline and some additional feature templates. A 1-of-N classifier is then trained to estimate a semantic role for each constituent Given these features, where N is the number of potential semantic roles plus an extra NONE role for non-role constituents. Most standard classification algorithms have been used (logistic regression, SVM, etc).

```
function SEMANTICROLELABEL(words) returns labeled tree

    parse ← PARSE(words)
    for each predicate in parse do
        for each node in parse do
            featurevector ← EXTRACTFEATURES(node, predicate, parse)
            CLASSIFYNODE(node, featurevector, parse)
```

Figure 4. A generic semantic-role-labeling algorithm

2.1.3. Corpus

The corpus is a collection of data in the form of text documents used in the case study text mining. Today, two kinds of corpus that can be used for developing

semantic argument annotation data are available; they are FrameNet and PropBank. Now, it is widely found a new corpus coming from the application FrameNet and PropBank. Some of the corpora are also used in many application domains adaptations such MiPACQ (multisource Integrated Platform for Clinical Question Answering), BioProp, and NomBank (Pradhan et al., 2005). In this research, the corpus is used as training data is PropBank. It is used because the annotation process between one label with another has an independent labeling. This is to simplify the classification process. While the corpus is used as the testing data is Quran translation in English compiled by the author refer to PropBank annotation rule.

2.1.3.1. Propbank

PropBank (Proposition Bank) was built in 2002 at the University of Pennsylvania by Martha Palmer and Paul Kingsbury [8]. PropBank uses semantic representations on a practical approach by adding information of predicate-argument, or semantic role label with Penn Treebank syntactic structure. PropBank itself is a collection of XML files that each file represents the verb which there are few examples of sentences and arguments that have been labeled.

Based on PropBank annotation rule, the argument of the verb labeled sequentially from ARG0 to ARG4, where ARG0 is the proto-agent (usually the subject of a transitive verb) and ARG1 are proto-patient (usually a direct object), and others. In addition to the core argument (ARG1-ARG4), there is also an additional argument that is marked as ARGMs. Examples of ARGMs is ARGM-LOC (indicate location), ARGM-DIR (indicates direction), ARGM-PRP (implying the destination), and so on. Table 4 List of core argument on PropBank shows a list of core arguments contained in PropBank and Table 5 List of Adjunct Arguments on PropBank shows a list of additional arguments contained in PropBank [9].

Table 4. List of Core Arguments on Propbank **[9]**

Tag	Description
ARG0	Agent, Operator
ARG1	Thing, Operated
ARG2	Explicit Patient
ARG4	Explicit Argument
ARG5	Explicit Instrument

Table 5. List of Adjunct Arguments on Propbank **[9]**

Tag	Description	Example
ARGM-LOC	Locative	The museum, in Westborough, Mass
ARGM-TMP	Temporal	Now, by next summer
ARGM-MNR	Manner	Heavily, clearly, at a rapid rate
ARGM-DIR	Direction	To market, to Bangkok
ARGM-CAU	Cause	In response to the ruling
ARGM-DIS	Discourse	For example, in part, Similarity
ARGM-EXT	Extent	At $38.375,50 points
ARGM-PRP	Purpose	To pay for the plant
ARGM-NEG	Negation	Not
ARGM-MOD	Modal	Can, might, should, will
ARGM-REC	Reciprocals	Each other
ARGM-PRD	Secondary Predication	To become a teacher
ARGM	Bare ARGM	With a police escort
ARGM-ADV	Adverbials	(none of the above)

2.1.3.2. English Translation of The Holy Quran

In its original Arabic, referring to [10], there are some uniqueness of the Quran sentence compared to the general sentence, among others:

1. Arabic Verbs.

In general, classical Arabic follows Verb-Subject-Object (VSO) order. The majority of Arabic verbs are trilateral, which can be derived to 15 different forms. Each derivation signifies some semantic variation over the original form.

2. The Quran Linguistic Style.

2.1. Literal vs Technical Sense Of Word

The Quran borrows an Arabic word and specializes it to indicate a technical term. For example, the word "jannah" meaning literally "a garden", but as a technical term in the Quran whenever this word is used to refer "the paradise".

2.2. Grammatical Shift

The Quran often draws the attention of the reader by shifting grammatical agreement is a statement. For example in verse [3:133], *"when you are in the ships and they sail with them with a fair breeze"*. The mode changed from "you" to "they" and "them" moving from the second person to the third person.

2.3. Verbs associated with different proposition

The Quran exhibits many examples where a certain verb is associated with a preposition which is unusual to the verb, but common with a different verb. For example the verb *"khala"* that means 'be alone'. This verb is usually followed by the preposition 'with' like 'John was alone with Mary'. However in verse [2:14], the Quran choose to use the preposition 'to', which sounds unusual to say, 'John was alone to Mary'.

2.4. Metaphors and Figurative

The Quran uses heavily metaphors and figurative. Verse [9:14] used the verb "shine", but the Arabic verb *ishtala* means "to flare" and shows the analogy of 'old age symptom by many gray hair' wit a 'fire burning a bush'.

2.5. Metonymy

In many verses the Quran uses metonymy. In [12:82] the Arabic verse literally means 'ask the town' which means (and was translated so) 'ask the people who live in the town'.

2.6. Imperative vs non-Imperative

Arabic verbs are classified into the past, present and imperative. thus, in Arabic, the imperative structure can be understood from the type of the verb used.

The Quran English translation is a translation of the original Arabic Quran. Hence the composition of grammar and the sentence structure, English-Quran is still influenced by the original languages, namely Arabic. In this thesis, English Translation of The Holy Quran is used as the testing data. The data is compiled into an XML file with the same structure as the corpus PropBank.

2.1.4. PractNLPTools

Practical Natural Language Processing Tools for Humans or practNLPTools is a pythonic library over SENNA and Stanford Dependency Extractor [11]. This research proposes a unified neural network architecture and learning algorithm that can be applied to various natural language processing tasks including part-of-speech tagging, chunking, named entity recognition, and semantic role labeling.

This thesis uses practNLPTools to perform argument identification on Quran domain data. The argument identification process generates the Qur'an data that has been labeled semantically and in this study is referred as auto labeled data.

2.1.5. Data Preparation

One important task in text mining is in preparing the data. It is because of no existing structured text data, therefore it is necessary a process transform the data into a structured space-vector model. The necessary steps are generally known as data preprocessing process. Some data preprocessing steps commonly used are:

1. Selection: Decides of the text that will be processed (sentences, paragraphs, and so on).
2. Tokenization: create a token of a text sentence into discrete words.
3. Stopwords: deletion of the words that are considered unimportant or will affect the data processing such as; a, the, of, and others.
4. Stemming: elimination of prefixes and suffixes to change a word into its basic form.

In this thesis, the selected text as the data are sentences that are derived from PropBank and English Quran's translation. Furthermore, the sentence will be converted into a parse tree with the help of a parser for information extraction.

2.1.6. Features

The commonly NLP task is to label the words. As well as SRL aims at delivering a semantic role to a syntactic constituent of a sentence. Traditional NLP approach is by extract a set of manually designed features from a sentence which is then fed to a standard classification algorithm, such as the Support Vector Machine (SVM), often with a linear kernel. The choice of features is a completely empirical process, mainly based first on linguistic intuition, and then trial and error, and the feature selection is task dependent, implying additional research for each new NLP task [11]. Complex tasks such as SRL suppose a large number of potentially complicated features (e.g., taken from a parse tree).

2.1.6.1. SRL Basic Features

Features commonly used by SRL system are called SRL basic features introduced by Gildea and Jurafsky [4]. Basic features are a set of features that are used for labeling the semantic argument.

1. Predicate: Lemma predicate is used as a feature. For example in a sentence "The Lecturer went to classroom", the predicate of the sentence is: "went".
2. Path: Syntactic path passing through parse tree from parse constituent towards predicate classified. Figure 4 illustrates the tree with NP path NP↑S↓VP↑VBD. ↑ presenting the upward movement in the tree and ↓ presenting downward movement in the tree.

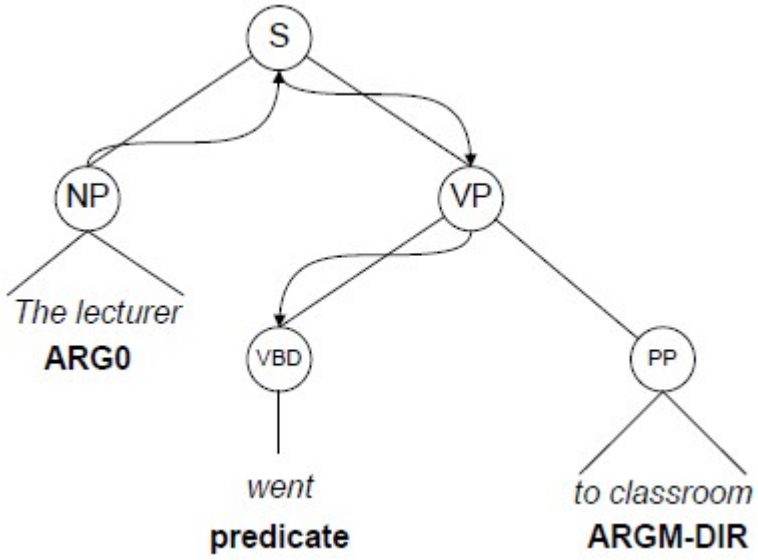

Figure 5. Illustration of Baseline Features **[4]**

3. Phrase Type: Syntax category of correspondence phrase is based on a semantic argument. Example: (NP, VP, S, etc.). Example: Phrase type of the phrase "The lecturer" of the sentence "The lecturer went to classroom" in figure 4 is NP.
4. Position: Features binary identification are based on the position of the phrase whether before or after the predicate. This feature is highly correlated with grammatical function because usually, the subject will appear before the verb or after the object. The Position usually represented in binary like 'L' or 'R' (left or right) or 'before' and 'after'.
5. Voice: The feature determines the predicate in a sentence whether active or passive predicate. The difference between active and passive verbs play an important role in the relationship between semantic roles and functions of grammar because the direct object of active verbs usually has a semantic relationship with the subject of passive verbs.
6. Head Word: The keyword of a phrase is calculated based on the table Head Word compiled by Magerman (1994) and modified by Collins (1999). Head words in a noun phrase can be used to specify the limits of choice of the semantic role.

7. Sub-Categorization: This feature is a phrase structure which expands the parent node of a predicate in a parse tree. For example in figure 4 illustration of Sub-Categorization features from predicate "went" is VP → VBD-PP.

Therefore, the basic features extracted from the sentence "The Lecturer went to classroom" are as follows:

Table 6. Examples of Baseline Features Extraction

Pr	**Vo**	**Sc**	**Pt**	**Hw**	**Pa**	**Po**	**Ar**
went	active	VP: VBD_PP	NP	lecturer	NP↑S↓VP↓VBD	before	ARG1
went	active	VP: VBD_PP	PP	to	PP↑VP↓VBD	after	ARG-DIR

Information :

Pr : Predicate
Vo : Voice
Sc : Sub-Categorization
Pt : Phrase Type
Hw : Head Word
Pa : Path
Po : Position
Ar : Argument

2.1.6.2. Additional Features

Throughout the study of the SRL system, some additional features have been developed after SRL basic features introduced by Gildea and Jurafsky. Pradhan et al. [12] use these basic features and designs some additional features, i.e. the part-of-speech tag of the headword, the predicted named entity class of the argument, features providing word sense disambiguation for the verb (they append 25 variants of 12 new feature types as a whole). They use the Propbank data that released on Feb 2004 adn SVM as the classier. It is close to the state-of-the-art in performance.

Xue et al [13] proposed some new additional features to perform an SRL task on by using the Propbank data released in April 2004, they tested the system with maximum entropy classifier and achieved very comparable result with [12].

In Toutanova et al [14] an SRL model over Propbank that effectively exploits the semantic argument frame as a joint structure, is presented. It incorporates strong dependencies within a comprehensive statistical joint model with a rich set of features over multiple argument phrases.

Yang et al. [15] propose some new features to improve SRL performance. The key idea of their work is to make a group of similar arguments activate one feature and another group of similar arguments activate another feature. The experiment conducted on Chinese and English Propbank.

This research uses some additional features that are expected to improve the performance of the classification. The additional features used in this thesis are:

1. Constituent Order

Constituent Order is related to the first/last word/POS in the constituent argument. However, this feature is designed for distinguishing arguments and non-arguments. [5] use a version of this feature. This features calculate the position of each constituent relative to the predicate, which is support the position of its proximity to the predicate [9].

2. Argument Order

Argument Order is introduced by [16], this feature is an integer that indicates the position of constituents in order of argument to the verbs. It is calculated after the initial phase constituents are classified as an argument or a non-argument. Because this feature does not use syntax parse tree, this can help create a strong semantic role labeling without being influenced the error parser [9].

3. Syntactic Frame

Syntactic Frame is introduced by [13]. This is a feature to complete the path and Sub-cat features. This feature refers syntactic predicates and NP as "pivots" and other elements are defined in relation to them. It describes a sequential pattern of noun phrases and predicate in a sentence. As example for constituent "classrooms" in Figure 2-1, the syntactic frame is np_v_NP, while "the lecturer" syntactic frame is

NP_v_np. The current constituent is expressed in capital letters on the syntactic frames produced, but can also be generalized to other terms such as CUR, X, etc that declare the constituent's position.

4. Noun Head of PP

Noun Head of PP is introduced by [12]. When the argument verb is a prepositional phrase (PP), head of the word is a preposition. This can often be a reliable indicator of the semantic role (e.g. in, across, and toward generally indicate the location), some prepositions can be used in various ways, and meaning can be determined by the object of the preposition in this case a noun [9]. For example, in March indicates time, while in Indonesia indicates location. So to figure 2-1, at "The lecturer" Noun Head PP produced is null because the lecturer is not a PP but NP. While for the head to classroom noun PP produced is "to".

5. First/Last Word/POS In Constituent

First/Last Word/POS In Constituent is introduced by [12]. This feature takes the first/last word/POS (Part-Of-Speech) in constituent no matter what the type is. This feature is obtained in from general way so that it is free from parser errors and applies to all types of its compilers.

Then the additional features obtained from the example sentence "The Lecturer went to classroom" are as follows:

Table 7. Examples of Additional Features Extraction

Nh	**Fw**	**Lw**	**Fp**	**Lp**	**Sf**	**Co**	**Ao**
-	the	lecturer	DT	NP	NP_v_np	1	0
to	to	classroom	PP	NP	np_v_NP	1	1

Information :

Nh : Noun Head of PP
Fw : First Word In Constituent
Lw : Last Word In Constituent

Fp : First POS In Constituent
Lp : Last POS In Constituent
Sf : Syntactic Frame
Co : Constituent Order
Ao : Argument Order

2.1.7. Feature Selection/Evaluation

Features or attribute selection/evaluation is a process of selecting or evaluating the most relevant attribute on the entire data with predictive modeling problems are being worked on. Attribute subset selection is mainly an optimization problem, which involves searching the space of possible feature subsets to select the one that is optimal or nearly optimal with respect to the performance measures accuracy, complexity etc. of the application [17].

The problem of Feature Selection can be defined as the process of selecting the best subset of features that describe the hypothesis at least as well as the original set (John, Kohavi, & Pfleger, 1994).

$$F' \in F$$

where F is the set of original 'n' features and F' is the output by a feature selector with m features.

There are three general classes of feature selection algorithms: filter methods, wrapper methods and embedded methods.

1. Filter Methods: This method is applying a statistical measure to assign a score to each feature. The features are sorted by rank scores then selected to be stored or removed from the database. This method is often univariate and considering these features independently or in connection with the dependent variable. Some examples of filter methods are the Chi-squared test, information gain and correlation coefficient scores.
2. Wrapper Methods: Wrapper methods consider the selection of a set of features as a search problem. This method constructs a number of different combinations of features, then evaluated and compared with other

combinations. A predictive model is used to evaluate the combination of features and set the value based on the accuracy of the model. The search process may be methodical as the best-first search, it possibly stochastic such as random hill-climbing algorithm, or may use heuristics, such as forward and backward to add and remove features. The wrapper method's example is the recursive feature elimination algorithm.

3. Embedded Methods: Embedded method learn about the best contribute features to the accuracy of the model while the model is being created. The most common type of embedded feature selection methods is regularization method. Regularization methods are also called penalization methods that introduce additional constraints into the optimization of a predictive algorithm (such as a regression algorithm) that bias the model toward lower complexity (fewer coefficients). Examples of regularization algorithms are the LASSO, Elastic Net, and Ridge Regression.

This thesis will use Filter Method for evaluating the most relevant attribute on entire data. The most relevant features selected will be developed into new features that expected will accommodate the features of a new argument. The proposed features will construct by detecting the most important features from Quran domain. The process is by finding out the attributes or features that have a high value of correlation with the class. For this process will use Gain Ratio attribute evaluation. Gain Ratio Attribute Evaluator evaluate the worth of an attribute by measuring the gain ratio with respect to the class. The selected features will be developed into a new feature and will be added to the training and testing data sets.

$$GainR(class, attribute) = \frac{H(class) - H(class|attribute)}{H(attribute)}$$

To select a feature set for the experiment, this thesis uses Wrapper Method, that select basic and additional features as in Table 6 and Table 7 then will be evaluated and compared with proposed features combinations.

2.1.8. Classifier

This thesis uses Support Vector Machine (SVM) as a classifier. SVM classification concept is trying to find a hyperplane (decision boundary lines) that separates the two best classes. The basic idea of SVM is to seek the maximum limit hyperplane as illustrated in the following figure:

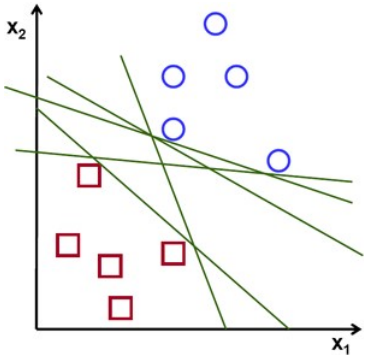

Figure 6. Hyperplane Options Are Possible

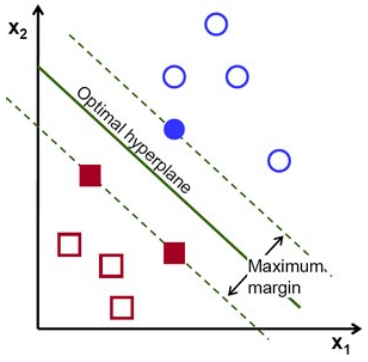

Figure 7. Hyperplane With Maximum Margin

Figure 6 shows the possible selection hyperplane to classify existing data sets. While Figure 7 shows the hyperplane with maximum margin among options that allow. Although it could also be used hyperplane arbitrary, hyperplane with maximum margin will give a better generalization in classification method [18].

2.1.8.1. SVM Linear

In SVM Linear, each training data express in the form (x, y) with i = 1,2,3, ..., N, and $x_i = \{x_{i1}, x_{i2}, ..., x_{iq}\}^T$ are attributes (features) for the training data set all i. While $y_i \in \{-1, +1\}$ declared label.

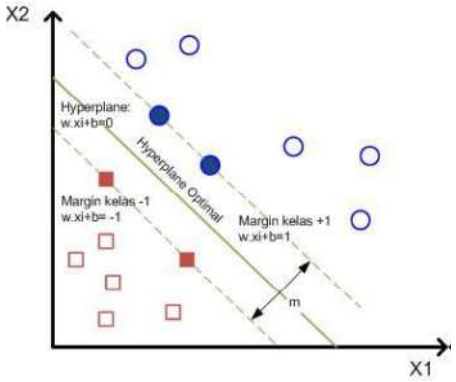

Figure 8. Hyperplane Margin

Figure 8 shows the linear hyperplane in SVM classification. Margin hyperplane is denoted as :

$$w.x_i + b = 0 \quad \text{..... (1)}$$

w and b are the parameters of the model. $w.x_i$ are an inner-product between w and x_i.

The data x_i that goes into -1 class is the data that satisfies the inequality as follows :

$$w.x_i + b < -1 \quad \text{..... (2)}$$

While the data xi that enter into + 1 class is data that satisfies the following inequality:

$$w.x_i + b > +1 \quad \text{..... (3)}$$

According to the figure, if there are data that are in -1 class (eg x_1) at hyperplane will satisfy the equation 1. For data in -1 class denoted :

$w.x_1 + b = 0$ ..... (4)

While for x + 1 class (eg x_2) will satisfy the equation :

$w.x_2 + b = 0$ ..... (5)

The difference from equation (5) and (4) then obtained equation :

$w.(x_2-x_1) = 0$ ..... (6)

x_2-x_1 is a parallel vector in hyperplane position and directed from x_1 to x_2.

Because the inner product value is 0, the direction of w should be perpendicular to the hyperplane as shown in the figure. By providing the label -1 for the first class and +1 for the second class, then the predictive of all the testing data using the formula [18]:

$$y = \begin{cases} +1, \text{jika } w.z + b > 0 \\ -1, \text{ jika } w.z + b < 0 \end{cases}$$ ..... (7)

According to the figure, the hyperplane for -1 class (dashed line) is the data on support vector that satisfies the equation :

$w.x_1 + b = -1$ ..... (8)

and for +1 class (dashed line) is the data that satisfies the equation :

$w.x_2 + b = +1$ ..... (9)

Therefore the margin can be calculated by the difference between (9) and (8) equation :

$w.(x_2-x_1) = 2$ ..... (10)

Margin hyperplane is obtained from the distance between the two hyperlinks from both classes. So the notation above can be summarized as :

$$\|w\| \; x \; d = 2 \; atau \; d = \frac{2}{\|w\|}$$ ..... (11)

2.1.8.2. Multiclass Support Vector Machine

For classification process, SVM is only able to perform a binary classification (two classes). While for the case study of semantic argument classification uses more than two classes. There are three approaches to address the problem of SVM with the multiclass case. The approach includes the one-vs-one, one-vs-all, and the error

correcting code output [19]. This thesis uses one-vs-all (OVA) approach in which this approach decomposition is done for multiclass problem into N binary class problem. For each $y_i \in Y$, formed binary settlement where ∀ vector which is owned by y_i class is regarded as a positive sample, and the other is considered as negative sample for separate one class with another. With the result that will form a number N of binary SVM [18].

The test vectors are classified by combining the results of all binary classifier, usually using a voting of all predicted results binary classifier. When the class gets the most votes, then label the class will be given to the test vectors. For OVA approach, if the sample is positive then classified these samples to get the vote, while if it is negative then all classes except positive will receive a vote. Suppose there are multiclass problems, $Y = \{y_1, y_2, y_3\}$. In the OVA approach test vectors to be predicted as (+, -, -). This means that the positive test vectors predicted when y_1 is used as a positive grade, and vice versa when y_2 and y_3 are used as a positive grade [18]

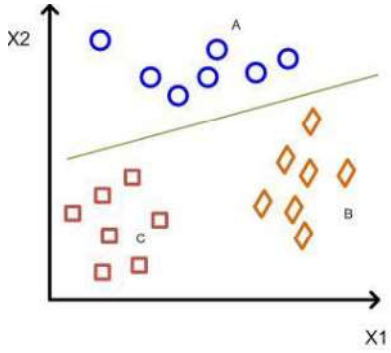

Figure 9.Multiclass SVM Prediction of A to Class B and C

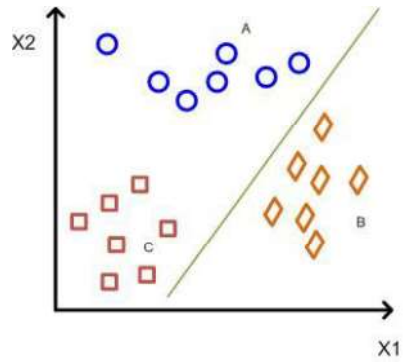

Figure 10. Multiclass SVM Prediction of B to Class A and C

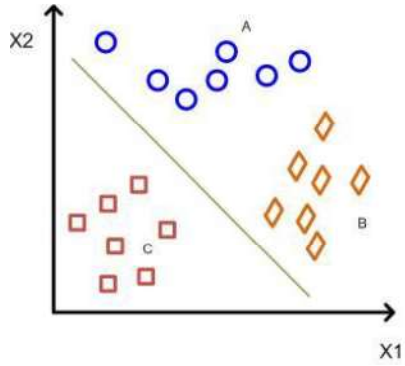

Figure 11. Multiclass SVM Prediction of C to Class A and B

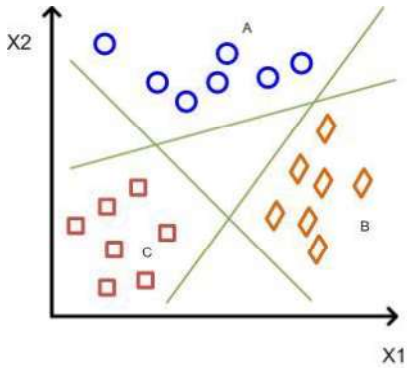

Figure 12. Multiclass SVM The Results Combination of A, B, and C

2.1.9. Weka

Weka (Waikato Environment for Knowledge Analysis)[3] is an open source workbench that supports a wide range of activities of practitioners in applying machine learning. Weka contains the implementation of algorithms for classification, clustering and association rules, together with the use of interfaces and other visualization utility for data exploration and evaluation algorithms. Some of the features of the Weka, among others:

1. Preprocessing data
2. Classification
3. Cluster analysis
4. Association analysis
5. Data Visualization
6. Attribute Selection

2.1.10. Comparison With Other System

When a system is compared, the quality of a system cannot be learned if it is trained with different data. Because the author has not found a similar study using the Qur'an domain as test data, system comparison in this thesis is not by comparing the system built with the existing system. As an analytical material, this thesis compare the performance of the classification of semantic arguments on the Quran domain using baseline system and the performance using the system after added with proposed features.

2.2. Related Studies

The annotated corpuses FrameNet and Propbank are an initial research that have greatly driven the development of Semantic Role Labeling. [4] is first introduced SRL system automatically based on statistical classifier trained on hand-

[3] http://www.cs.waikato.ac.nz/ml/weka/

annotated corpora FrameNet. In their initial work, they use gold or auto parse syntax tree as input and then extracting various lexical and syntactic features to identify the semantic role for a given predicate.

After [4], there has been a lot of progress produced on automatic semantic role labeling. Progress can be linked to better modeling techniques, more relevant features and in small sizes, clean annotations and machine learning models.

Based on the basic model, [20] build integer linear programming architecture where a dependency relationship between the arguments is veiled in the constraint conditions. [14] proposes a joint model to explore the relationship of all the arguments of the same predicate.

For features and machine learning models, [4] as initial research uses basic features on Table 6 and using maximum entropy classifier. [21] results in two systems using decision tree classifier. The first system uses the same features with [4]. Then they show the improved performance of another system which uses some additional features. The state-of-the-art reported by Pradhan et al [12] where a wide range of novel features, including features extracted from named entities, verb clusters, verb sense, temporal cue word, dynamic context, etc are tested with an SVM classifier. And thereafter, most studies in SRL use SVM and maximum entropy classifier. [22] [13] [2] [1].

In features engineering, terms of finding the proper syntactic and semantic knowledge, the SRL researchers investigated features based on two formalisms, namely constituency grammar and dependency grammar. The SRL systems constructed since 2000 has found a variety of features that constituency grammar provides for. The initial Basic Features introduced by [4], and a number of additional features were later introduced by [12] [13]. It has been a long history that SRL systems have tried to use the dependence among semantic arguments, such as [4] [13] [23] [9] [24] [15].

Research on semantic argument classification requires data that has been labeled semantically in large numbers, which called corpus. In the preliminary research, two types of corpus have been built, namely FrameNet [25] and Propbank [8], both are from news domain or news genre. Because building a corpus is costly and time-consuming, recently many studies have used FrameNet and Propbank corpus as training data to conduct semantic argument classification research on the new domain without the need to build a new corpus for those new domains [1] [2].

In the Quran domain, there were some efforts to build the Quran corpus. [26] present preliminary work on the creation of a unique Arabic proposition repository for Quranic Arabic. They annotated the semantic roles for the 50 most frequent verbs in the Quranic Arabic Dependency Treebank (QATB) [27]. [10] present an initial research task for building a lexical database of the verb valences in the Arabic Quran using FrameNet frames. They studied the context of verbs in the Arabic Quran and compare them with matching frames and frame evoking verbs in the English FrameNet. These two studies focus on building corpus using the rules of Propbank and FrameNet. Both did not report any experiments to test the performance of the built corpus. While in the same domain that is Bible, the author has not found a similar research, either related to the preparation of corpus for semantic labeling as well as research related to the classification of a semantic argument. Research related to semantic research on Bible generally discusses the linguistic context and meaning of words.

In the sentence patterns, there are some similarities and differences in Propbank and Quran's translation. This supports the idea that the Propbank data is reusable for Quran domain, without having to build a corpus Quran thoroughly. By augmented some appropriate features to minimize the mismatch between Propbank and Quran domain data will improve the performance of semantic argument classification on Quran domain.

CHAPTER 3

RESEARCH METHODOLOGY

3.1. Research Design

3.1.1. System/Product/Method implementation

The system built on this final project has some device specifications and needs. Here is the functionality specification used in system development.

3.1.1.1. Hardware Specification

Hardware specification to support this thesis for building and processing the thesis data are as follows:

1. Processor Inter(R) Core(TM) i3-4005U CPU @ 1.70GHz
2. Minimum 8 GB RAM
3. Minimum 600 GB HD
4. Minimum 6 GB Heap Size Java™ Configuration

3.1.1.2. Software Specification

Software specification to support this thesis for building and processing the thesis data are as follows:

1. Windows Pro 10 x64 Bit Operating System
2. Java™ Standard Edition V7
3. JDK 1.8 Version
4. NetBeans IDE 8.0.2
5. Weka 3.6.13 © 2015
6. PractNLPTools 1.0
7. Notepad ++
8. Microsoft Excel 2010

3.1.2. Experiment Scenario

3.1.2.1. Baseline Architecture

The general process of the baseline system is as follows:

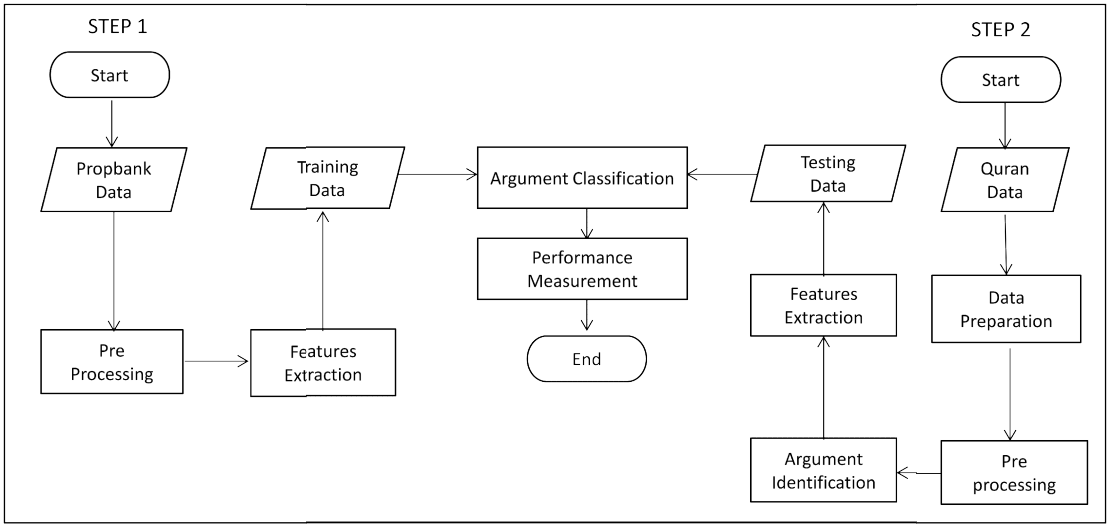

Figure 13. Baseline Architecture

3.1.2.2. Experiment Design For Proposed Solution

The general process of the proposed solution as follows:

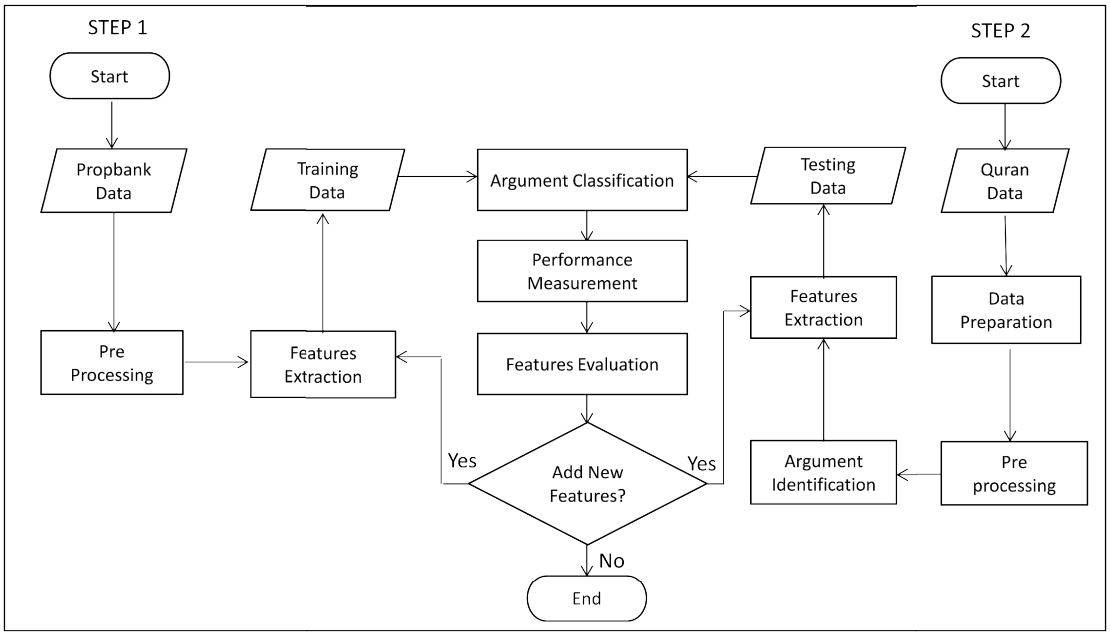

Figure 14. Experiment Scenario

Data used from the Propbank corpus and the English translation of the Quran were formed into a PropBank model. The system built in this thesis was able perform the feature extraction process of semantic argument classification of an English sentence. The extractor in this thesis played a role in producing quality features in the classification process. The steps undertook generally consist of feature extraction process and classification of a semantic argument.

Figure 12 is the general stage of the system developed in this thesis, with the following explanation:

3.1.2.3. Argument Identification

As mentioned in the assumption on the previous chapter, the study focuses on argument classification process. Therefore, the system proposed in this research focused on preparing and performing the argument classification step. As explains in the limitations of this research, the process of argument identification was assumed to have been done or using the data that has been through the argument identification step.

3.1.2.3.1. Propbank Corpus

The Propbank data used in this research was the data contained in Propbank frameset version 1.7[4] in XML format. This Propbank frameset consisted of 7261 predicates and its variation in XML files which generated as many as 24.865 arguments or constituent. Each XML file consisted of one to three sample sentences based on predicates that had been completed with the arguments and predicates contained in the sentence.

3.1.2.3.2. Quran Translation

The Qur'an translation was unlabeled data. Because this thesis used the Supervised Learning method, the unlabeled Quran's translation data should be labeled first. For need analysis, two scenarios for the labeling process were applied. The first scenario used the existing automatic semantic role labeler, in this thesis using

[4] http://propbank.github.io/

practNLPTools 1.0. The result of this scenario was named Quran Auto Labeled Data. The second scenario used hand-labeled data namely manually labeling the data by the author refer to Propbank annotation rule [4]. The result was named Quran Manual Labeled or Hand Labeled Data. This two types of data were the testing data in this thesis. This process generated labeled Qur'an translation data stored in XML format, according to the frameset of Propbank data.

```
<example>
<text>
His [devil] companion will say Our Lord I did not make him transgress but he [himself] was in extreme error
</text>
<arg n="0">His [ devil ] companion</arg>
<arg n="M" f="MOD">will</arg>
<rel>say</rel>
<arg n="1">Our Lord I did not make him transgress but he [ himself ] was in extreme error</arg>
</example>
<example>
<text>
[Allah] will say Do not dispute before Me while I had already presented to you the warning
</text>
<arg n="0">Allah</arg>
<arg n="M" f="MOD">will</arg>
<rel>say</rel>
<arg n="1">Do not dispute before Me while I had already presented to you the warning</arg>
</example>
```

Figure 15. Quran Translation XML File

3.1.2.4. Preprocessing

Preprocessing was the stage of extracting information in the text. This process aimed to select the text structure that were used as input in accordance with the needs and structures expected by the system. The input at this stage was an XML file of PropBank data and Quran translation data that had been structured and labeled. The necessary argument information was taken from the already labeled data. This process generated data used on the system being tested.

```
<example name="unstated goal or source">
<inflection aspect="ns" form="full" person="ns" tense="past" voice="active"/>
<text>
    William Gates and Paul Allen in 1975 developed an early
    language-housekeeper system for PCs, and Gates became an industry
    billionaire six years after IBM adapted one of these versions in
    1981.
</text>
        <arg n="0">IBM</arg>
        <rel>adapted</rel>
        <arg n="1">one of these versions</arg>
        <arg f="TMP" n="M">in 1981</arg>
</example>
```

Figure 16. Propbank XML File

Figure 16 is an example of extractable data, where sentences, predicates, and arguments are taken. A sentence is between the <text> and </text> tag, the predicate is between the <rel> and </rel> tag and the arguments are between the <arg> and </arg> tags. For predicate "adapt" in sentence "William Gates and Paul Allen in 1975 developed an early language-housekeeper system for PCs, and Gates became an industry billionaire six years after IBM adapted one of these versions in 1981", are obtained some arguments; "IBM" as ARG0, "one of these versions" as ARG1 and "in 1981" as ARGM-TMP. This argument information extraction generates a new sentence that will be used for the next process, i.e. "IBM adapted one of these versions in 1981".

3.1.2.5. Baseline System

This research takes a critical view of the features used in the semantic role tagging literature. Previous research has shown that different features are required for different subtasks [13]. Developing features that capture the right information is essential to advance the latest analysis in the semantic field.

As mentioned in point 2.1.12, it is necessary for each research to establish a baseline system as a benchmark in performance appraisal for the proposed system. In this study, the baseline system is a system built using the basic features and additional features described above [12].

Table 8. Baseline System Features

Basic Features **[4]**	Additional Features **[5] [13]**
Predicate	Noun Head of PP
Phrase Type	Syntactic Frame
Path	First Word In Constituent
Position	Last Word In Constituent
Voice	First POS In Constituent
Headword	Last Word In Constituent
Sub-Categorization	Constituent Order
	Argument Order

State-of-the-art SRL systems consist of several stages: producing a parse tree, identifying which parse tree nodes represent the arguments of a given verb, and finally classifying these nodes to compute the corresponding SRL tags [11]. This requires extracting the various basic features from the parse tree or syntactic tree and feeding them into statistical models. The process of producing a parse tree or a syntactic tree of a sentence is called parsing. The following is an example of a parse tree and syntactic tree for the sentence generated from the sentence above:

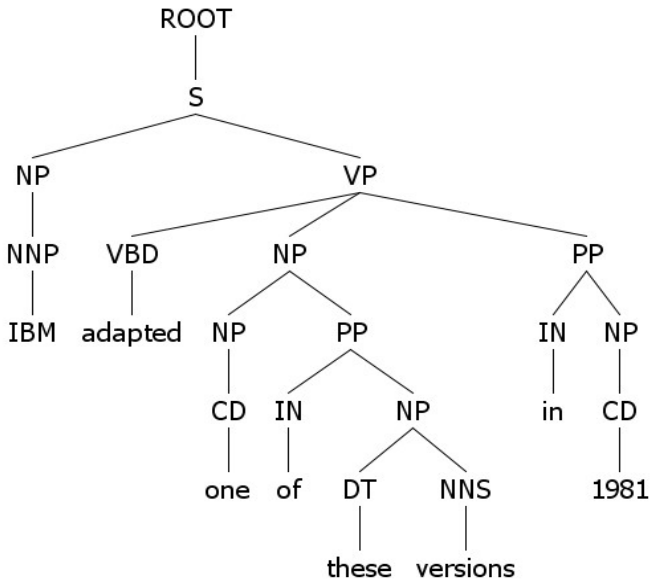

Figure 17. Parse Tree Form a Sentence

```
==============
Syntactic Tree
==============
(ROOT
  (S
    (NP (NNP IBM))
    (VP (VBD adapted)
      (NP
        (NP (CD one))
        (PP (IN of)
          (NP (DT these) (NNS versions))))
      (PP (IN in)
        (NP (CD 1981))))))
```

Figure 18. Syntactic Tree From a Sentence

To perform the parsing process, this research used the Stanford Parser[5]. Features were extracted from the parse tree, syntactic tree and dependencies list that were formed. The extracted feature was a reference to a phrase whether it was classified into the list of available arguments (ARG0, ARG1, ..., ARGM).

There were three categories of features used in this research, namely the baseline, additional features and proposed features. Basic Features are some of the features that were first discovered and generally used for semantic argument classification process, while additional features are additional features that will be used in the classification of a semantic argument that is considered to increase the performance.

Basic Features [4] consist of Predicate, Path, Phrase Type, Position, Voice, Headword, and Sub-Categorization. While the additional features are Noun Head of PP, First Word in Constituent, Last Word in Constituent, First POS in Constituent, Last POS Constituent, Syntactic Frame, Constituent Order and Argument Order.

[5] https://nlp.stanford.edu/software/lex-parser.shtml

3.1.2.5.1.Basic Features Extraction

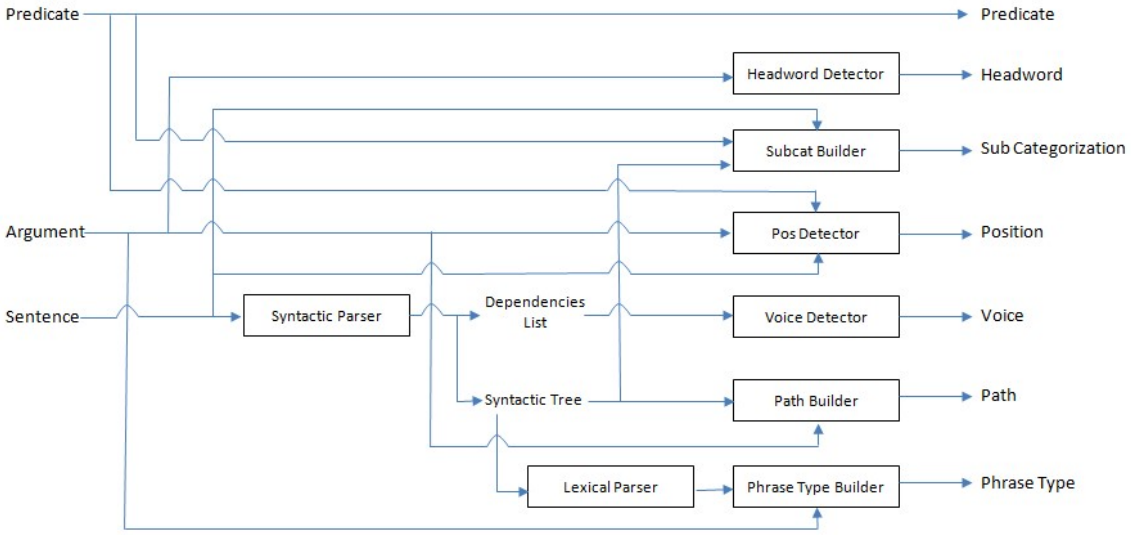

Figure 19. Basic Features Architecture

Figure 17 above illustrates the input and flowing system for performing basic feature extraction. A predicate is a feature that is directly obtained by extracting the tag <rel> from the PropBank's frameset XML file. To extract Headword feature, the required input parameter is the argument/constituent. To build the Sub-Categorization feature required predicate, argument, and sentence that have been obtained its syntactic tree. To detect the position required three parameters namely sentence, argument, and predicate. To detect the Voice feature it takes a sentence as an input parameter. While the Path feature requires a sentence that was going to be converted into a syntactic tree form with its arguments. And finally building the Phrase Type feature requires arguments and syntax tree sentences.

The following are a description of the feature extraction steps in the Basic Features.

1. Predicate

The predicate feature extraction process was done by retrieving information between the <rel> and </rel> tags of the XML file.

2. Path

The path feature was taken based on the parse tree shown in Figure 3-3. The path feature from the sentence are:

- the argument "IBM": NP↑S↓VP↑VBD
- the argument "one of these versions": NP↑VP↓VBD
- the argument "in 1981": PP↑VP↓VBD

3. Phrase Type

The Phrase Type feature was obtained by taking Part of Speech from the argument that the label will be identified. For the sentence " IBM adapted one of these versions in 1981" has Phrase Type as follows: NP, NP, PP.

4. Position

The Position represents the location of the constituent/argument to be identified. Position feature was obtained by splitting into sentences based on the predicate. If the argument was in the first sub-sentence or before the predicate, it was represented as "left". If it was in the second sub-sentence or after the predicate, it was represented as "right".

5. Voice

Voice was obtained by performing the process of dependency parsing. In the dependencies list, there are two types of subjects: nsubj or nsubjpass. For type njsubj then voice sentence is "active", while for nsubjpass then voice sentence is "passive". In Figure 3-7 below, it can be seen that adapted has nsubj dependencies relation with IBM, so the sentence has predicate as active voice.

```
Dependencies
============
root(ROOT-0, adapted-2)
nsubj(adapted-2, IBM-1)
dobj(adapted-2, one-3)
case(versions-6, of-4)
det(versions-6, these-5)
nmod:of(one-3, versions-6)
case(1981-8, in-7)
nmod:in(adapted-2, 1981-8)
```

Figure 20. Dependencies List

6. Headword

The headword feature of the sentence was taken from the keyword of an argument taken from the syntactic tree that is formed. For example, the headword of arguments from the above sentence are:

- Argument "IBM": they
- Argument " one of these versions ": versions
- Argument "in 1981": 1981.

7. Sub-Categorization

Sub-Categorization feature was obtained from a phrase structure which expanded the parent node of a predicate in a parse tree. For example in figure 3-4, an illustration of Sub-Categorization features from predicate "adapted" is VP → VBD-NP-PP.

3.1.2.5.2.Additional Features Extraction

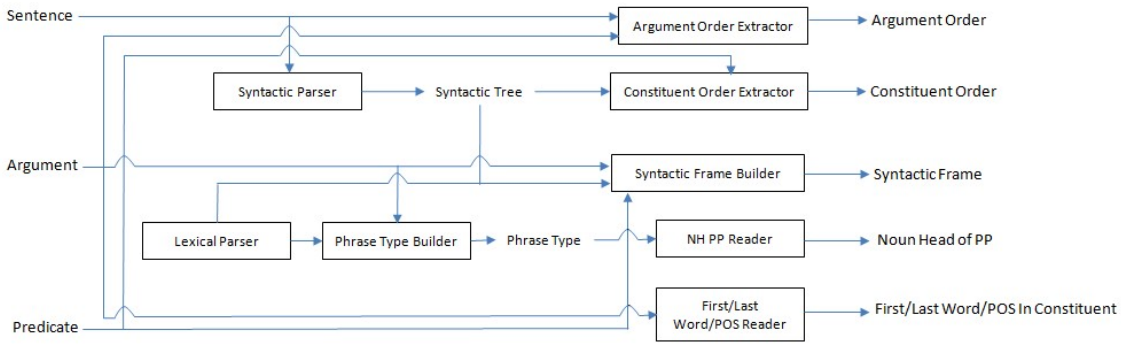

Figure 21. Additional Features Architecture

Figure 19 illustrates the flow for additional features. For the Argument Order feature requires sentences and arguments as input parameters. The Constituent Order Features require syntactic trees and predicates. The Syntactic Frame feature requires three parameters: argument, syntax tree and predicate. The Noun Head of PP feature requires the phrase type and argument as an input parameter, this feature is embedded in the result of phrase type baseline feature. The First/Last Word/POS in Constituent features is not tied to any features and is not even related to the syntax of sentence structure, it only requires a list of arguments present in the sentence as input parameters.

The following are a description of the feature extraction steps in the additional features.

1. Noun Head of PP (NH)

The Noun Head of PP feature depends on the Phrase Type of the baseline feature, where the PP type of Phrase Type will be searched for its Head Noun on the proposition.

2. First Word in Constituent (FW)

This feature extracts the first word of the extracted constituent, the extracting phase of this feature can be done after the sentence is split into several constituents.

3. Last Word in Constituent (LW)

This feature extracts the last word of the extracted constituent, the extracting phase of this feature can be done after the sentence is split into several constituents.

4. First POS in Constituent (FP)

This feature extracts the first Part-Of-Speech (POS) of the extracted constituent, the extracting phase of this feature can be done after the sentence is split into several constituents.

5. Last POS in Constituent (LP)

This feature extracts the last Part-Of-Speech (POS) of the extracted constituent, the extracting phase of this feature can be done after the sentence is split into several constituents.

6. Syntactic Frame (SF)

The Syntactic Frame feature relies on the Phrase Type generated from the baseline feature because it points to the predicate and NP as the syntactic reference defining the other elements in relation to them.

7. Constituent Order (CO)

Basically, the CO feature is include in the path feature of the baseline feature, where it calculates the relative position or distance to the predicate for each extracted constituent. Therefore, in the development of this feature, a slight modification of CO is calculated relative to the predicate position, changed to the relative calculation of the node with part-of-speech VP and S. This change improves the performance of standard CO although not significant. So the CO produced for the sentence is 1,1,1 according to the distance of each constituent at the Phrase Type level of the syntactic tree.

8. Argument Order (AO)

This feature is a feature not related to syntactic trees or other Baseline Feature. This feature only sees the sequence of arguments that have been obtained from the sentence. For example, the above sentence is divided into 3 constituents, so for each constituent get the Argument Order value in the order of constituent in the sentence 1,2,3 and so on (if divided into more than 3 constituent).

Finally, the dataset obtained from the feature extraction is a data set that has attributes of Basic Features, additional features, sentence indexes, extracted sentences, and constituents.

3.1.2.6. Argument Classification

The feature extracted dataset that had been formed were used as data for the classification. The classification were conducted using Weka. Before classifying, there were some preparation processes that should be carried out.

1. CSV to ARFF

Extraction data originally in CSV format were changed to ARFF format. This process was done by inputting CSV data to Weka.

2. Instance Filtering

The filtering process here changes from the class argument attribute of the previous string type to the nominal type.

3. Attribute Remove

Attribute remove aimed to reduce the attributes that will affect the performance of the classification results.

4. Attribute Merge

The merge attribute was performed by combining the content attributes of each attribute contained in each dataset in ARFF format by eliminating duplicate values. The result of the merge attribute used as a new attribute on both datasets that had been formed.

In this research, to select the classifier, an initial experiment was conducted on baseline system by two commonly used classifiers, Naive Bayes, and linear SVM. The results obtained are as follows:

Table 9. Comparison of Naive Bayes and SVM Linear Performance

	Tested on Propbank		Tested on Quran		Tested on Quran	
			Naïve Bayes		SVM Linear	
	Naïve Bayes	SVM Linear	Auto Labeled	Hand Labeled	Auto Labeled	Hand Labeled
Accuracy	76.21	98.30	69.29	66.78	81.92	87.40
Precision	74.40	98.30	81.30	82.40	83.20	89.30
Recall	76.20	98.30	69.30	66.40	81.90	87.40
F-Measure	75.30	98.30	72.80	72.00	82.10	88.10

Table 9 shows the performance of each classifier when for Propbank and Quran data. The best performance for these data is obtained by using SVM linear. Therefore, this research will use SVM as a classifier. This method is included in the top 10 rankings in data mining algorithms [28] and has shown good performance in text classification task, where data in large dimensions represented using sparse feature vectors [29]. SVM also works well on the set of high dimensional data [18]. For an experiment on this research will use libSVM [22] with linear kernel, degree 3, tolerance of the termination criterion, e = 0.001 and cost per unit violation of the margin is C = 1.0. The type of SVM that will be used is a standard algorithm (C-SVC) for classification.

3.1.2.7. Baseline System Performance

The following Table 10 shows the performance of classification results using baseline system when tested on auto labeled and hand-labeled data. Performance is indicated by the value of accuracy, precision, recall, and F-1. From the table, it can be seen that the performance of the system increased significantly when added additional features.

Table 10. Baseline System Performance

Features	Auto Labeled Data				Hand Labeled Data			
	A	P	R	F1	A	P	R	F1
Basic	77.12	79.10	77.10	77.80	76.47	79.90	76.50	77.70
Basic + Additional	81.92	83.20	81.90	82.10	87.40	89.30	87.40	88.10

In detail for the four arguments which become the focus of research on this thesis, the performance of the baseline system are as follows:

Table 11. ARG0, ARG1, ARG2 and ARGM-TMP Baseline System Performance

Arguments	Auto Labeled Data		Hand Labeled Data	
	Basic	Basic + Additional	Basic	Basic + Additional
ARG0	91.37	94.06	88.40	92.17
ARG1	77.64	78.79	79.92	86.41
ARG2	51.85	85.19	45.16	78.71
ARGM-TMP	27.66	77.66	14.05	77.27
ALL	77.12	81.92	76.47	87.40

Such as the performance of all arguments, the performance of these four arguments also increased when additional features were added to the system. From the results of this initial experiment was concluded that the performance of the system can be improved by adding some features that corresponding with the type of data.

3.1.2.8. System Improvement

The main problem on the argument classification task with different domains is when there is a new argument found in the testing data but not found in the training data. To recognize the new argument in training data, extending the argument features in the training data to accommodate the new features of the new argument becomes one of the solutions.

For example is the word "the monitor". In the news domain corpus like Propbank, "monitor" is tagged as a verb. However on the domain of computer hardware corpus, "monitor" is tagged as a noun. Therefore the argument "monitor/noun" is a new argument on the news domain corpus. Similarly, "monitor/verb" becomes a new argument on the corpus of computer hardware domains.

Table 12. Example of new argument on Quran

Argument	Propbank	Quran
who	ARG1	ARG0 / ARG1
those	ARG1	ARG0/ARG1
when	TMP	TMP / ARG1
TO + DT	these	those
Allah	Unseen	ARG0/ARG1
Moses, Pharaoh etc	Unseen	ARG0/ARG1
Exalted	Unseen	ARG1

Based on the problems stated above, feature extraction is the crucial point in this research. The task of SRL is usually handled as a supervised problem. Therefore, a series of features is critical to the performance of the SRL system. Many works [13, 30] has studied what features are discriminative for semantic role labeling assignments.

Therefore, this thesis proposes four new features that augmented to baseline system to improve the system performance. By adding these proposed features, the performance of ARG0, ARG1, ARG2, and ARGM-TMP were expected to increase further.

The idea of forming a new feature was done through detection of the most important features from Quran domain. The features evaluation process performed from features extracted data. The process was by finding out the attributes or features that have a high value of correlation with the class. This process performed by Weka, using Gain Ratio Attribute Evaluator and search method Ranker. The result obtained was the features rank based on the value of the gain ratio.

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 16 Arg):
        Gain Ratio feature evaluator

Ranked attributes:
 0.6845    4 Position
 0.4826    3 Phrase Type
 0.4561    8 NH_PP
 0.3436    2 Path
 0.3098   11 1St POS
 0.3033   12 Last POS
 0.2723    9 1St Word
 0.2531    6 Head Word
 0.2428   13 Syntactic Frame
 0.2364   10 Last Word
 0.2336   14 Cons Order
 0.2245   15 Arg Order
 0.0827    7 Subcategorization
 0.0553    1 Predicate
 0.0234    5 Voice

Selected attributes: 4,3,8,2,11,12,9,6,13,10,14,15,7,1,5 : 15
```

Figure 22. Feature Evaluation

Based on the results of feature evaluation process in Figure 22, the proposed features performed by developing the Position, Phrase Type, Path, 1st POS and Last POS features. For Noun Head of PP feature was not included because this feature does not work thoroughly, just for preposition type phrases.

The following are the proposed features to improve the system performance:

1. Position Order (PO)

PO Position Order built from position and path features. PO represented the position of an argument from the predicate. This feature was the development of feature position in basic features. The feature position of basic feature only specify the position of the argument against the predicate, i.e. before or after. But in this

feature, the distance is also mentioned. For example, one position on the left, formulated into "left_1".

PO was expected to improve system performance by detecting new argument through the position in detail. For example, for the "who" argument. Table 12 shows that in training data, the "who" argument is tagged only as ARG1. While in the Quran data other than as ARG1, the "who" argument is also widely tagged as ARG0. So to recognize it, it was marked by the characteristics of PO. For ARG0 the phrase type order in the tag as "left_0". If as ARG1 is tagged as "right_0" or "right_1", depending on its position. Similarly for other new arguments.

2. Phrase Type Order (PTO)

PTO built from phrase type and position features. PTO represented the phrase type and position of an argument or constituency in one package. The purpose of the position on this feature is like a position on the basic features, namely the position of the argument against the predicate. PTO format example is "NP_left". PTO was expected to improve system performance by detecting new argument through phrase type and its position in one package. For example, for the "who" argument.

Table 12 shows that in training data, the "who" argument is tagged only as ARG1. While in the Quran data other than as ARG1, the "who" argument is also widely tagged as ARG0. So to recognize it, it was marked by the characteristics of PTO. For ARG0 the phrase type order in the tag as "WHNP_left". If as ARG1 is tagged as "SBAR_right". Similarly for other new arguments.

3. Second POS In Constituent (SP)

SP as well as the first/last POS in constituent, but took the POS of the second word of the constituent. The consideration of proposing this feature was to detect new arguments as well as in Table 12, "to those" not present in training data. But in training data, there is "to these". Since "to those" and "to these" have the same POS as "TO + DT", the system can recognize "to those" through its POS "TO + DT".

4. Second Word In Constituent (SW)

SW as well as the first/last word in constituent, but took the second word of the constituent. The consideration of proposing this feature was to detect new arguments as well as in Table 12. For example, for the "those" argument, it can be seen that in training data, the "those" argument is tagged only as ARG1. While in the Quran data other than as ARG1, the "those" argument is also widely tagged as ARG0. It can be recognized by the second word. For ARG0 the second word is "who" and for ARG1 the second word usually varies.

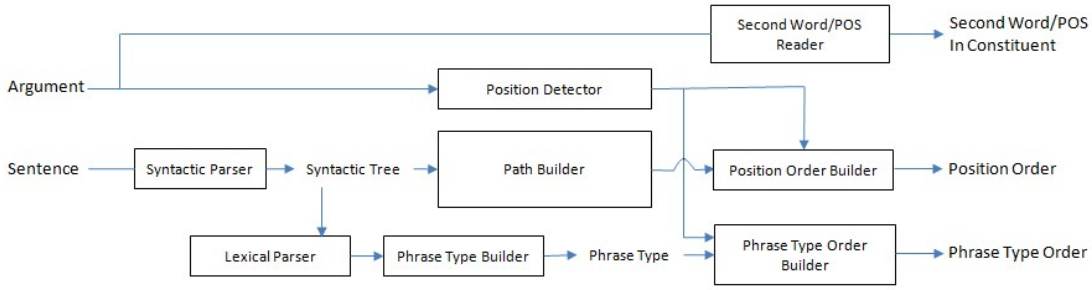

Figure 23. Proposed Features Architecture

3.1.2.9. Evaluation

After the classification process, it is necessary to evaluate the results of the experiment. The evaluation process was conducted to determine the performance of the model formed in the classification process. Evaluation was carried out by recording the accuracy, precision, recall, and f-score generated. Recording results used as a comparison with the baseline system.

3.2. Population / Sampling

As the training data, this research used Propbank frameset that consisted of 7261 predicates and its variation in XML files which generated as many as 24.865 arguments or constituent. And as the testing data is the English translation of Quran with the predicate "say" which was labeled refer to Propbank annotation rule.

3.3. Instruments and Data Collection

The data used for the experiment in this research were:

1. As the training data, this thesis used Propbank data that contained in frameset Propbank version 1.7[6] in XML format. Each XML file consisted of one to three sample sentences based on predicates that had been completed with the arguments and predicates contained in the sentence.
2. As the testing data, this thesis used English Quran's translation Quran by Ministry of Religious Affairs downloaded through the website Tanzil – Quran Navigator[7].

Both of the data above had been downloaded and stored in a separate file, so it can be read and analyzed further.

3.4. Tools for Data Analysis

The general parameter used in previous studies to measure the performance of the SRL system is accuracy. Accuracy is Accuracy is the percentage of data that classified correctly compared to the total amount of data.

$$Accuracy = \frac{True\ Prediction}{Amount\ of\ Data}\ x\ 100\% \quad \text{..... (12)}$$

Besides accuracy, one of the measurement methods that is used to measure the performance of the classification system is confusion matrix. To evaluate a classification system confusion matrix dividing the binary classification into two classes. For more details, see the following table:

Table 13. Confusion Matrix

		Prediction Results Class	
		Positive	**Negative**
Original Class	**Positive**	True Positive (TP)	False Negative
			Error Type II
	Negative	False Positive	True Negative (TN)
		Error Type I	

[6] http://propbank.github.io/
[7] www.tanzil.net

Example from the sentence "She drinks":

- True Positive: the word "drinks" is correctly identified as a verb.
- False Positive: the word "She" is misidentified as a verb.
- True Negative: the word "She" not identified as a verb.
- False Negative: the word 'Drink' is not identified as a verb.

Precision is the amount of data that is truly positive (the number of positive data identified correctly as positive) divided by the amount of data that are recognized as positive, while recall is the amount of data to true positives divided by the amount of data that is actually positive (true positive + true negative). Written in the form of the equation becomes:

$$Precision = \frac{TP}{TP+FP} \quad \text{..... (13)}$$

$$Recall = \frac{TP}{TP+FN} \quad \text{..... (14)}$$

In classification process, precision value = 1 in Class C means that all data labeled class C indeed come from class C (but do not say there is no data of class C are not labeled correctly). While the recall = 1, it means that all the data of the class C-labeled class C (but does not rule out also that data wrongly labeled as class C). Precision and Recall usually have a reverse trade-off relationship. If you want precision ride normally be paid by reducing the recall, and vice versa. The value of recall and precision can be combined in one metric that is F-measure (Rijsbergen, 1979) [18]. F-measure is the average harmonic weights between precision and recall.

$$F = 2\,x\,\frac{Precision\ x\ Recall}{Precision+Recall} \quad \text{..... (15)}$$

CHAPTER 4

PRESENTATION, ANALYSIS, AND INTERPRETATION OF DATA

4.1. Presentation of Data

4.1.1. Training Data

The Propbank data used in this research were the data contained in frameset Propbank in XML format. Each XML file consisted of one to three sample sentences based on predicates completed with the arguments and predicates contained in the sentence. This data originally consisted of 27.629 arguments. Due to resource limitations in running the experiment, resample technique using Weka to filter data was conducted and it only took 90% data that consisted of 24,865 arguments with 7261 predicates These data consisted of sentences, constituents or arguments, and argument labels.

4.1.2. Testing Data

The testing data were two kinds of the Qur'an translation labeled data. The first was the Quran Auto Labeled Data and the second was Quran Manual Labeled or Hand Labeled. The Quran domain data were Qurans verses with the predicate "say". For auto labeled data, there were 2566 arguments and for hand-labeled data, there were 3374 arguments. These two types of data stored in XML format, according to the frameset of Propbank data. Similar to training data, this data consisted of sentences, constituents or arguments, and argument labels.

Table 14. Presentation of Training and Testing Data

Sentence	Constituent	Arg Label
gen. noriega abandon his command for a comfortable exile	gen. noriega	0
gen. noriega abandon his command for a comfortable exile	his command	1
gen. noriega abandon his command for a comfortable exile	for a comfortable exile	2
he abandoned himself to the very worst	he	0
he abandoned himself to the very worst	himself	1
he abandoned himself to the very worst	to the very worst	2
if he seem to frown you then abase yourselves as miserable wretches	if he seem to frown	M ADV
if he seem to frown you then abase yourselves as miserable wretches	you	0
if he seem to frown you then abase yourselves as miserable wretches	then	M TMP
if he seem to frown you then abase yourselves as miserable wretches	yourselves	1
if he seem to frown you then abase yourselves as miserable wretches	as miserable wretches	M PRD
they degrading him by making him report to a supervisor	they	0
they degrading him by making him report to a supervisor	him	1
they degrading him by making him report to a supervisor	by making him report to a supervisor	M MNR

4.1.3. Features Extracted Data

Feature extracted data were data that had been through the argument identification and feature extraction process. These data consisted of the features of the argument. This features extracted data were data ready for the classification process.

Table 15. Presentation of Features Extracted Data

Predicate	Path	Phrase Type	Position	Voice	Head Word	Subcategorization	NH_PP	1St Word	Last Word	1St POS	Last POS	Syntactic Frame	Cons Order	Arg Order	2nd Word	2nd POS	PT Order	Posit Order
abide	CDâ†'NPâ†'FRAG	NP	left	null	trace1	VP-VB-PP	null	trace1	trace1	'NN '	'NN '	np_v_np	0	1	null	null	NP_left	left_0
abide	PPâ†'VPâ†"VB	PP	right	null	ceasefire	VP-VB-PP	by	by	ceasefire	'IN '	'NN '	null	1	2	the	'DT '	PP_right	right_1
abolish	NPâ†'Sâ†"VPâ†"VPâ†"VB	NP	left	'aktif '	government	VP-VB-S	null	the	government	'DT '	'NN '	CUR_v_np_np_np_np	1	1	government	'NN '	NP_left	left_1
abolish	MDâ†'VPâ†"VPâ†"VB	MD	left	'aktif '	would	VP-VB-S	null	would	would	'MD '	'MD '	null	0	2	null	null	MD_left	left_0
abolish	Sâ†'VPâ†"VB	S	right	'aktif '	share	VP-VB-S	null	its	motor	'PRP$ '	'NN '	null	1	3	golden	'JJ '	S_right	right_1
abominated	NPâ†'Sâ†"VPâ†"VBD	NP	left	'aktif '	they	VP-VBD-NP	null	they	they	'PRP '	'PRP '	CUR_v_np_np	1	1	null	null	NP_left	left_1
abominated	NPâ†'VPâ†"VBD	NP	right	'aktif '	idea	VP-VBD-NP	null	the	monarchy	'DT '	'NN '	np_v_CUR_np	1	2	very	'RB '	NP_right	right_1
aborted	NPâ†'Sâ†"VPâ†"VBN	NP	left	'aktif '	she	VP-ADVP-VBN	null	she	she	'PRP '	'PRP '	CUR_v	1	1	null	null	NP_left	left_1
aborted	ADVPâ†'VPâ†"VBN	ADVP	left	'aktif '	completely	VP-ADVP-VBN	null	completely	completely	'RB '	'RB '	null	1	2	null	null	ADVP_left	left_1
abound	NPâ†'Sâ†"VPâ†"VBP	NP	left	'aktif '	sellers	VP-VBP	null	sellers	sellers	'NNS '	'NNS '	CUR_v	1	1	null	null	NP_left	left_1
abounds	NPâ†'Sâ†"VPâ†"VBZ	NP	left	'aktif '	room	VP-VBZ-PP	null	johns	room	'NNS '	'NN '	CUR_v_np	1	1	room	'NN '	NP_left	left_1

4.2. Analysis of The Data

Here are the results of the experiments conducted in accordance with the experiment design in the previous chapter. From the experiment, by adding four new features separately to baseline system has proven can improve the performance of the system.

Table 16. Performance of Proposed Features Separately

Features	Auto Labeled		Hand Labeled	
	ACC	F1	ACC	F1
Baseline (B)	81.92	82.10	86.72	87.60
B+SP	83.13	**83.40**	**87.46**	**88.20**
B+SW	81.96	82.10	86.93	87.80
B+PO	82.23	82.50	87.20	88.00
B+PTO	**82.54**	82.70	87.40	88.10

By some combination option of the proposed features, the following are the result of an enhancement in accuracy and F-1 score for each ARG0, ARG1, ARG2, ARGM-TMP and overall arguments.

Table 17. The Enhancement of System Performance With Proposed Features

ARGUMENT	AUTO LABELED DATA				HAND LABELED DATA			
	ACCURACY	FEATURES	F-MEASURE	FEATURES	ACCURACY	FEATURES	F-MEASURE	FEATURES
ARG0	0.67	B+SW	1.30	B+PO+SP	1.10	B+PTO+SW	0.00	-
ARG1	3.28	B+PO+SP	2.20	B+PO+SP	0.27	B+PO	1.10	B+PO+PTO+SP
ARG2	0.00	-	1.20	B+PTO	5.81	B+PTO+SP	1.90	B+PO+PTO
ARGM-TMP	6.38	B+PO+PTO	7.00	B+PTO+SP	7.44	B+PTO	4.50	B+PTO
ALL ARG	1.48	B+PO+SP	1.60	B+PO+SP	0.47	B+PO+PTO	0.40	B+PO+PTO

4.2.1. Argument Performance

The following are the performance of the classification results for ARG0, ARG1, ARG2, and ARGM-TMP using the proposed features.

4.2.1.1. ARG0

The following are the performance of the classification results for ARG0 using the proposed features.

Table 18. The performance of ARG0 With The Proposed Features.

FEATURES	Auto Labeled		Hand Labeled	
	ACC	F1	ACC	F1
Baseline (B)	94.06	91.50	92.17	**91.90**
B+SP	94.39	**92.70**	**93.19**	91.80
B+SW	**94.73**	91.10	91.99	91.20
B+PO	93.83	91.80	91.99	91.80
B+PTO	93.83	92.00	91.99	**91.90**
B+SP+SW	**94.62**	90.90	93.00	91.20
B+PO+SP	94.17	**92.80**	91.71	91.60
B+PO+SW	**94.62**	91.00	**93.19**	91.40
B+PO+PTO	93.83	92.20	91.99	**91.90**
B+PTO+SP	93.61	**92.50**	91.80	**91.80**
B+PTO+SW	**94.51**	91.00	**93.28**	91.40
B+PO+SP+SW	94.28	91.10	92.91	91.10
B+PO+PTO+SP	93.83	**92.40**	91.71	**91.80**
B+PO+PTO+SW	**94.39**	91.30	93.09	91.20
B+PTO+SP+SW	94.17	91.10	93.00	91.10
ALL FEATURES	94.17	91.00	92.82	90.80

For ARG0 when tested on auto labeled data, Table 18 shows that the highest accuracy was obtained by adding SW features, increased of 0.67% from the baseline system. And for F-1, the highest was obtained by adding PO+SP features, increased of 1.30% from the baseline features. When tested on hand-labeled data, the highest accuracy was obtained by adding PTO+SW features, increased of 1.10% from the baseline features, and for an F-1 score there was no improvement from the baseline, but the addition of PTO and PO+PTO features was obtained the same F-1 value as the baseline, that was 91.90%.

4.2.1.2. ARG1

The following are the performance of the classification results for ARG1 using the proposed features.

Table 19. Performance of ARG1 With The Proposed Features

FEATURES	Auto Labeled		Hand Labeled	
	ACC	F1	ACC	F1
Baseline (B)	78.79	83.70	86.41	89.10
B+SP	81.19	85.40	84.45	88.90
B+SW	78.35	83.60	85.73	88.40
B+PO	79.68	84.30	**86.68**	**89.20**
B+PTO	79.86	84.30	86.14	89.00
B+SP+SW	78.26	83.50	84.18	88.30
B+PO+SP	**82.08**	**85.90**	85.60	88.70
B+PO+SW	78.70	83.90	84.92	88.70
B+PO+PTO	80.21	84.50	**86.48**	**89.20**
B+PTO+SP	81.37	85.20	85.67	88.90
B+PTO+SW	78.26	83.40	84.58	88.60
B+PO+SP+SW	78.44	83.60	84.18	88.30
B+PO+PTO+SP	**81.46**	**85.50**	**86.34**	**89.30**
B+PO+PTO+SW	79.33	84.00	84.79	88.60
B+PTO+SP+SW	77.91	83.10	83.57	88.00
ALL FEATURES	78.17	83.30	83.84	88.20

For ARG0 when tested on auto labeled data, Table 19 shows that the highest accuracy was obtained by adding PO+SP features, increased of 3.28% from the baseline features. And for F-1, the highest was obtained also by adding PO+SP features, increased of 2.20% from the baseline features. When tested on hand-labeled data the highest accuracy was obtained just by adding PO feature, increased of 0.27% from the baseline features. And F-1 score increased to 1.10% by adding PO+PTO+SP features.

4.2.1.3. ARG2

The following are the performance of the classification results for ARG2 using the proposed features.

Table 20. The Performance of ARG2 With The Proposed Features

FEATURES	Auto Labeled		Hand Labeled	
	ACC	F1	ACC	F1
Baseline (B)	**85.19**	52.30	78.71	74.80
B+SP	**85.19**	50.50	80.00	75.40
B+SW	81.48	50.60	81.94	75.40
B+PO	**85.19**	52.30	79.35	76.20
B+PTO	**85.19**	**53.50**	80.00	75.20
B+SP+SW	79.63	48.90	81.29	75.90
B+PO+SP	**85.19**	50.80	81.94	74.90
B+PO+SW	81.48	51.20	81.29	76.40
B+PO+PTO	**85.19**	52.90	80.65	**76.70**
B+PTO+SP	**85.19**	49.50	**84.52**	75.70
B+PTO+SW	79.63	49.40	80.00	75.20
B+PO+SP+SW	81.48	48.40	81.94	76.00
B+PO+PTO+SP	**85.19**	49.50	**83.87**	75.80
B+PO+PTO+SW	77.78	49.40	80.00	75.20
B+PTO+SP+SW	79.63	46.70	81.94	74.10
ALL FEATURES	79.63	46.20	81.29	74.30

For ARG2 when tested on auto labeled data , Table 20 shows that there is no improvement from the baseline for the accuracy. The addition of SP, PO, PTO, PO+SP, PO+PTO, PTO+SP, PO+PTO+SP features was obtained the same accuracy value as the baseline, that is 85.19%. And the highest F-1 score was obtained by adding PTO feature, increase of 1.20% from the baseline. When tested on hand-labeled data the highest accuracy was obtained by adding PTO+SP features that increased significantly of 5.81% from the baseline and F-1 score increased of 1.90% by adding PO+PTO features.

4.2.1.4. ARGM-TMP

The following are the performance of the classification results for ARGM-TMP using the proposed features.

Table 21. The Performance of ARGM-TMP With The Proposed Features

FEATURES	Auto Labeled		Hand Labeled	
	ACC	F1	ACC	F1
Baseline (B)	77.66	71.60	77.27	80.60
B+SP	76.60	75.40	73.55	79.70
B+SW	77.66	74.90	76.03	78.40
B+PO	77.66	71.60	73.55	78.10
B+PTO	82.98	73.20	**84.71**	**85.10**
B+SP+SW	75.53	75.10	72.31	78.30
B+PO+SP	75.53	74.70	76.03	79.80
B+PO+SW	77.66	75.30	73.14	78.50
B+PO+PTO	**84.04**	73.50	**83.47**	**84.30**
B+PTO+SP	81.91	**78.60**	79.75	82.30
B+PTO+SW	82.98	77.20	**83.06**	**84.80**
B+PO+SP+SW	76.60	75.40	71.07	77.50
B+PO+PTO+SP	80.85	77.60	78.51	81.70
B+PO+PTO+SW	79.79	75.40	77.69	81.60
B+PTO+SP+SW	81.91	77.80	78.10	82.20
ALL FEATURES	80.85	77.90	76.03	80.50

For ARGM-TMP when tested on auto labeled data , Table 21 shows that the highest accuracy was obtained by adding PO+PTO features, increased significantly of 6.38% from the baseline. And the highest F-1 score was obtained by adding PO+SP features, also increased significantly by 7% from the baseline. When tested on hand-labeled data the highest accuracy was obtained just by adding PTO feature, increased significantly of 7.44% from the baseline, and F-1 score increased 4.50% also by adding PTO features.

4.2.2. Feature Performance

The following are the features performance of the proposed system results for all arguments.

Table 22. The Features Performance on All Argument With The Proposed Features when tested on auto labeled data (%)

No	Features	Accuracy	Precision	Recall	F-Measure
1	Baseline+PO+SP	83.40	84.60	83.40	83.70
2	Baseline+PTO+SP	83.16	84.30	83.20	83.40
3	Baseline+SP	83.13	84.50	83.10	83.40
4	Baseline+PO+PTO+SP	83.13	84.30	83.10	83.40
5	Baseline+PO+PTO	82.74	83.70	82.70	82.80
6	Baseline+PTO	82.54	83.60	82.50	82.70
7	Baseline+PO	82.23	83.60	82.20	82.50
8	Baseline+PO+PTO+SW	82.15	83.20	82.20	82.30
9	Baseline+PO+SW	82.07	83.20	82.10	82.30
10	Baseline+PTO+SW	82.00	83.00	82.00	82.10
11	Baseline+SW	81.96	83.10	82.00	82.10
12	**Baseline**	**81.92**	**83.20**	**81.90**	**82.10**
13	Baseline+PO+SP+SW	81.76	83.10	81.80	82.00
14	Baseline+SP+SW	81.72	83.00	81.70	82.00
15	Baseline+PTO+SP+SW	81.68	83.10	81.70	81.90
16	ALL	81.64	83.20	81.60	81.90

When tested on auto labeled data, Table 22 shows that the highest accuracy for all arguments was obtained by adding PO+SP features, that was 83.40%, increased of 1.48% from the baseline. And the highest Precision, Recall, and F-1 were obtained also by adding PO+SP features, respectively increased of 1,40%, 1,50% and 1,60% from the baseline.

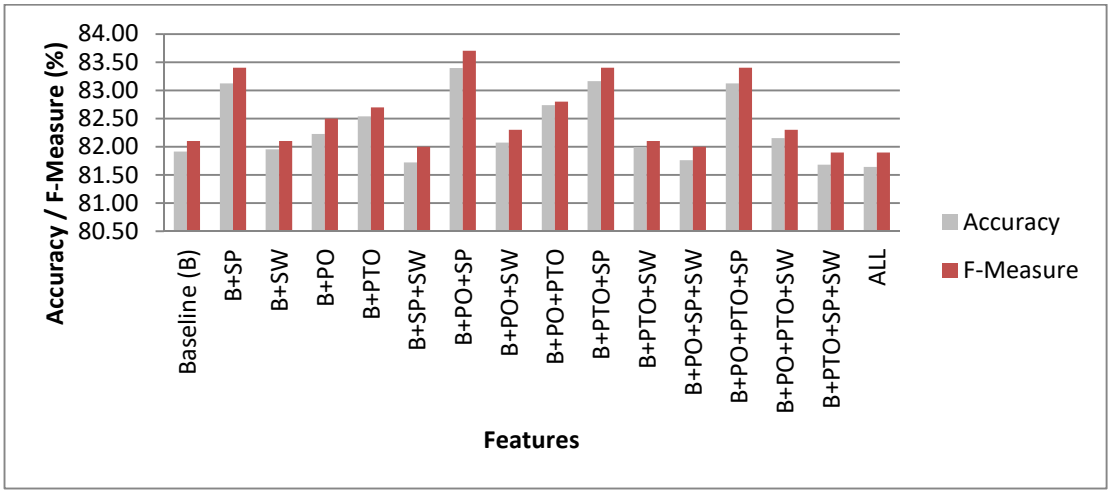

Figure 24. The Features Performance of The Proposed Features when tested on auto labeled data (%)

While when tested on hand-labeled data, the following are the features performance of the proposed features results for all arguments.

Table 23. Features Performance on All Argument With The Proposed Features when tested on hand-labeled data (%)

No	Feature	Accuracy	Precision	Recall	F-Measure
1	B+PO+PTO	87.88	89.60	87.90	88.50
2	B+PTO	87.79	89.40	87.80	88.40
3	B+PO+PTO+SP	87.55	89.70	87.60	88.30
4	B+PTO+SP	87.46	89.50	87.50	88.20
5	B+PTO+SW	87.43	89.40	87.40	88.10
6	**Baseline (B)**	**87.40**	**89.30**	**87.40**	**88.10**
7	B+SP	87.20	89.40	87.20	88.00
8	B+PO	87.17	89.20	87.20	88.00
9	B+PO+SP	87.02	89.20	87.00	87.90
10	B+PO+PTO+SW	86.99	89.10	87.00	87.80
11	B+PO+SW	86.93	89.30	86.90	87.80
12	B+PTO+SP+SW	86.72	89.10	86.70	87.60
13	B+SW	86.69	89.00	86.70	87.60
14	B+SP+SW	86.54	89.20	86.50	87.60
15	ALL	86.48	88.90	86.50	87.40
16	B+PO+SP+SW	86.43	89.10	86.40	87.40

When tested on hand-labeled data, Table 23 shows that the highest accuracy for all arguments was obtained by adding PO+PTO features, that was 87.88%, increased of 0.47% from the baseline. And the highest precision, recall, and F-1 were obtained also by adding PO+PTO features, respectively increased of 0,30%, 0,50% and 0,40% from the baseline.

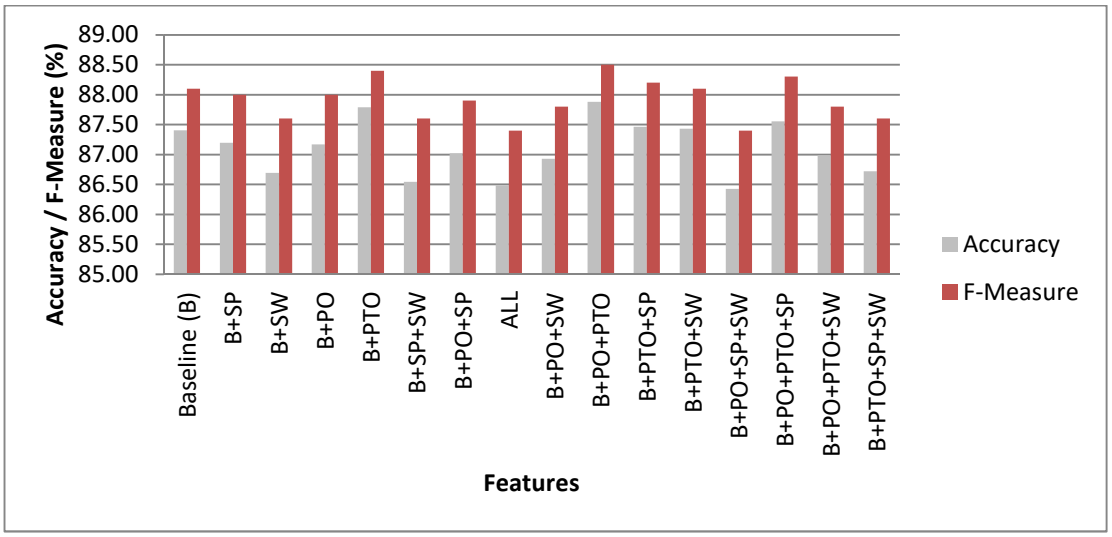

Figure 25. The Features Performance of The Proposed System when tested on hand-labeled data (%)

4.2.3. Correlation

The correlations between ARG0, ARG1, ARG2, ARGM-TMP and all argument performance and the proposed features are as follows:

4.2.3.1. Tested on Auto Labeled Data

When tested on auto labeled data, the correlation between proposed features and the performance of arguments are as follows:

Table 24. The Correlation Between Proposed Features and Arguments Performance when tested on Auto Labeled Data

FEATURES	ACCURACY					F-MEASURE				
	ARG0	ARG1	ARG2	ARGM-TMP	ALL	ARG0	ARG1	ARG2	ARGM-TMP	ALL
Baseline (B)	94.06	78.79	85.19	77.66	81.92	91.50	83.70	52.30	71.60	82.10
B+SP	94.39	81.19	85.19	76.60	83.13	92.70	85.40	50.50	75.40	83.40
B+SW	94.73	78.35	81.48	77.66	81.96	91.10	83.60	50.60	74.90	82.10
B+PO	93.83	79.68	85.19	77.66	82.23	91.80	84.30	52.30	71.60	82.50
B+PTO	93.83	79.86	85.19	82.98	82.54	92.00	84.30	53.50	73.20	82.70
B+SP+SW	94.62	78.26	79.63	75.53	81.72	90.90	83.50	48.90	75.10	82.00
B+PO+SP	94.17	82.08	85.19	75.53	83.40	92.80	85.90	50.80	74.70	83.70
B+PO+SW	94.62	78.70	81.48	77.66	82.07	91.00	83.90	51.20	75.30	82.30
B+PO+PTO	93.83	80.21	85.19	84.04	82.74	92.20	84.50	52.90	73.50	82.80
B+PTO+SP	93.61	81.37	85.19	81.91	83.16	92.50	85.20	49.50	78.60	83.40
B+PTO+SW	94.51	78.26	79.63	82.98	82.00	91.00	83.40	49.40	77.20	82.10
B+PO+SP+SW	94.28	78.44	81.48	76.60	81.76	91.10	83.60	48.40	75.40	82.00
B+PO+PTO+SP	93.83	81.46	85.19	80.85	83.13	92.40	85.50	49.50	77.60	83.40
B+PO+PTO+SW	94.39	79.33	77.78	79.79	82.15	91.30	84.00	49.40	75.40	82.30
B+PTO+SP+SW	94.17	77.91	79.63	81.91	81.68	91.10	83.10	46.70	77.80	81.90
ALL FEATURES	94.17	78.17	79.63	80.85	81.64	91.00	83.30	46.20	77.90	81.90

Table 24 shows that there is a correlation of the proposed feature for the improvement of performance for ARG0, ARG1, ARG2, and ARGM-TMP separately, and for overall arguments when tested on auto labeled data.

1. The accuracy of ARG0 has a positive correlation with the SW feature. By adding the SW feature and its combination (SW, SP +SW, PO+SW, PTO+SW) to the system has proven to improve the accuracy of ARG0. While the F-1 score of ARG0 has a positive correlation with the SP, PO and PTO features. By adding these features and its combination (SP, PO+SP, PTO+SP, PO+PTO+SP) to the system have proven to improve the F-1 score of ARG0.
2. The accuracy and F-1 score of ARG1 has a positive correlation with the SP, PO and PTO features. By added these features and its combination (SP, PO+SP, PTO+SP, PO+PTO+SP) to the system has proven to improve the accuracy and F-1 score of ARG1.
3. The accuracy of ARG2 has a negative correlation with the SW feature. By adding the SW feature and its combination (SW, SP +SW, PO+SW, PTO+SW, PO+PTO+SW, PTO+SP+SW) to the system have proven degraded the accuracy of ARG2. While the F-1 score of ARG2 has a positive correlation with the PO and PTO features. By adding these features and its combination (PO+PTO, PTO, PO) to the system has proven to improve the F-1 score of ARG2.

4. The accuracy of ARGM-TMP has a positive correlation with the PTO feature. By added this feature and its combination (PTO, PTO+SW, PTO+SP, PO+PTO, PO+PTO+SW, PO+PTOSW) to the system has proven to improve the accuracy of ARGM-TMP. While the F-1 score of ARGM-TMP has a positive correlation with the PTO and SP features. By adding these features and its combination (PTO+SP, PO+PTO+SP, PTO+SP+SW) to the system has proven to improve the F-1 score of ARGM-TMP.
5. The accuracy and the F-1 score of all arguments have a positive correlation with the SP feature. By added this feature and its combination with PO and PTO (SP, PO+SP, PTO+SP, PO+PTO+SP) to the system has proven to improve the accuracy of all arguments.

4.2.3.2. Tested On Hand Labeled Data

When tested on hand-labeled data, the correlation between proposed features and the performance of arguments are as follows:

Table 25. The Correlation Between Proposed Features and Arguments Performance When Tested on Hand Labeled Data

FEATURES	ACCURACY					F-MEASURE				
	ARG0	ARG1	ARG2	ARGM-TMP	ALL	ARG0	ARG1	ARG2	ARGM-TMP	ALL
Baseline (B)	92.17	86.41	78.71	77.27	87.40	88.10	89.10	74.80	80.60	88.10
B+SP	91.99	85.73	81.94	76.03	87.20	88.00	88.90	75.40	79.70	88.00
B+SW	93.19	84.45	80.00	73.55	86.69	87.60	88.40	75.40	78.40	87.60
B+PO	91.99	86.68	79.35	73.55	87.17	88.00	89.20	76.20	78.10	88.00
B+PTO	91.99	86.14	80.00	84.71	87.79	88.40	89.00	75.20	85.10	88.40
B+SP+SW	93.00	84.18	81.29	72.31	86.54	87.60	88.30	75.90	78.30	87.60
B+PO+SP	91.71	85.60	81.94	76.03	87.02	87.90	88.70	74.90	79.80	87.90
B+PO+SW	93.19	84.92	81.29	73.14	86.93	87.80	88.70	76.40	78.50	87.80
B+PO+PTO	91.99	86.48	80.65	83.47	87.88	88.50	89.20	76.70	84.30	88.50
B+PTO+SP	91.80	85.67	84.52	79.75	87.46	88.20	88.90	75.70	82.30	88.20
B+PTO+SW	93.28	84.58	80.00	83.06	87.43	88.10	88.60	75.20	84.80	88.10
B+PO+SP+SW	92.91	84.18	81.94	71.07	86.43	87.40	88.30	76.00	77.50	87.40
B+PO+PTO+SP	91.71	86.34	83.87	78.51	87.55	88.30	89.30	75.80	81.70	88.30
B+PO+PTO+SW	93.09	84.79	80.00	77.69	86.99	87.80	88.60	75.20	81.60	87.80
B+PTO+SP+SW	93.00	83.57	81.94	78.10	86.72	87.60	88.00	74.10	82.20	87.60
ALL FEATURES	92.82	83.84	81.29	76.03	86.48	87.40	88.20	74.30	80.50	87.40

Table 25 shows that there is a correlation of the proposed feature for the improvement of performance for ARG0, ARG1, ARG2, and ARGM-TMP separately, and for overall arguments when tested on hand-labeled data.

1. The accuracy of ARG0 has a positive correlation with the SW feature. By adding the SW feature and its combination (SW, SP +SW, PO+SW, PTO+SW) to the system has proven to improve the accuracy of ARG0. While the F-1 score of ARG0 has a positive correlation with the SP, PO and PTO features. By adding these features and its combination (SP, PO+SP, PTO+SP, PO+PTO+SP) to the system have proven to improve the F-1 score of ARG0.
2. The accuracy and F-1 score of ARG1 has a positive correlation with the PO, PTO and SP features. By added these features and its combination (SP, PO+SP, PTO+SP, PO+PTO+SP) to the system has proven to improve the accuracy and F-1 score of ARG1.
3. The accuracy of ARG2 has a positive correlation with a combination of PTO and SP features. By added these combination features (PTO+SP, PO+PTO+SP) to the system has proven degraded the accuracy of ARG2. While the F-1 score of ARG2 has a positive correlation with the PO and SW features. By adding these features and its combination (PO, PO+SW) to the system has proven to improve the F-1 score of ARG2.
4. The accuracy of ARGM-TMP has a positive correlation with the PTO feature. By added this feature and its combination (PTO, PTO+SW, PO+PTO) to the system has proven to improve the accuracy of ARGM-TMP. While the F-1 score of ARGM-TMP has a positive correlation also with the PTO feature. By adding these features and its combination (PTO, PO+PTO+SP, PTO+SW) to the system have proven to improve the F-1 score of ARGM-TMP.
5. The accuracy and the F-1 score of all arguments have a positive correlation with PTO and the combination of PO and PTO features. By added PTO and PO+PTO combination to the system has proven to improve the accuracy of all arguments.

4.2.4. Differences of Measures

Generally, the proposed system can improve the system performance. By adding the combination of proposed features has proven increased the performance of ARG0, ARG1, ARG2, ARGM-TMP and all arguments. But however, counter-use of all features proposed simultaneously decreases performance (lower than baseline). The performance decline in the use of all these features is likely due to the use of a combination of SW and SP features. From a series of experiments showed that the combination of SP and SW on one package did not produce a good performance, even in accuracy and F-1 score. The possible cause is due to the considerable variation of the second word have a large number of possible values with the same POS.

Table 26. The Performance Of Proposed Features Combination

Auto Labeled				Hand Labeled			
FEATURES	ACC	FEATURES	F1	FEATURES	ACC	FEATURES	F1
ALL FEATURES	81.64	B+PTO+SP+SW	81.90	B+PO+SP+SW	86.43	B+PO+SP+SW	87.40
B+PTO+SP+SW	81.68	ALL FEATURES	81.90	ALL FEATURES	86.48	ALL FEATURES	87.40
B+SP+SW	81.72	B+SP+SW	82.00	B+SP+SW	86.54	B+SW	87.60
B+PO+SP+SW	81.76	B+PO+SP+SW	82.00	B+SW	86.69	B+SP+SW	87.60
Baseline (B)	81.92	Baseline (B)	82.10	B+PTO+SP+SW	86.72	B+PTO+SP+SW	87.60
B+SW	81.96	B+SW	82.10	B+PO+SW	86.93	B+PO+SW	87.80
B+PTO+SW	82.00	B+PTO+SW	82.10	B+PO+PTO+SW	86.99	B+PO+PTO+SW	87.80
B+PO+SW	82.07	B+PO+SW	82.30	B+PO+SP	87.02	B+PO+SP	87.90
B+PO+PTO+SW	82.15	B+PO+PTO+SW	82.30	B+PO	87.17	B+SP	88.00
B+PO	82.23	B+PO	82.50	B+SP	87.20	B+PO	88.00
B+PTO	82.54	B+PTO	82.70	Baseline (B)	87.40	Baseline (B)	88.10
B+PO+PTO	82.74	B+PO+PTO	82.80	B+PTO+SW	87.43	B+PTO+SW	88.10
B+SP	83.13	B+SP	83.40	B+PTO+SP	87.46	B+PTO+SP	88.20
B+PO+PTO+SP	83.13	B+PTO+SP	83.40	B+PO+PTO+SP	87.55	B+PO+PTO+SP	88.30
B+PTO+SP	83.16	B+PO+PTO+SP	83.40	B+PTO	87.79	B+PTO	88.40
B+PO+SP	**83.40**	B+PO+SP	**83.70**	B+PO+PTO	**87.88**	B+PO+PTO	**88.50**

4.2.5. Interpretation of Data

Between the Propbank and AlQuran data, there were some differences that produced some new arguments. To address this problem, some features were augmented to baseline system, so the training data was able to recognize these new arguments. In accordance to the hypothesis, the addition of four new features and its

combination were able to improve the performance of the system with some combination options.

The SW feature only contributed positively to ARG0 when tested in both types of data, this is due to the Quranic data there were quite a few verses that have similar word patterns in ARG0. The SW feature contributed negatively to ARG2, this was due to the considerable variation of the second word have a large number of possible values but have the same POS and are not present in Propbank, e.g. "to Allah", "to Moses", "to Satan", etc.

The SP feature was capable of contributing positively to ARG0 and ARG1, both in terms of accuracy and F-1. The PO feature also contributed positively to ARG0 and ARG1, both in terms of accuracy and F-1. The PTO feature contributed positively to ARG2 and ARGM-TMP, both in terms of accuracy and F-1.

The PO+SP combination greatly affected the performance gains of accuracy and F-1 in ARG1 when tested on auto labeled data. The combination PTO+SP had greatly affected the performance gains of accuracy in ARG2 when tested on hand-labeled data and the F-1 in ARGM-TMP when tested on auto labeled data. And the combination of PO+PTO had greatly affected the performance gains of accuracy in ARGM-TMP when tested on auto labeled data.

4.3. Summary of Findings

Based on the objectives of this research, to consider the best feature combination, this section analyze against the performance of accuracy and F-1 scores for overall arguments, taking into account the average accuracy and F-1 scores of ARG0, ARG1, ARG2, and ARGM-TMP.

Table 27. The Performance of Proposed Features When Tested on Auto Labeled Data

FEATURES	ACCURACY						F-MEASURE					
	ARG0	ARG1	ARG2	ARGM-TMP	AVG	ALL	ARG0	ARG1	ARG2	ARGM-TMP	AVG	ALL
Baseline (B)	94.06	78.79	85.19	77.66	83.92	81.92	91.50	83.70	52.30	71.60	74.78	82.10
B+SP	94.39	81.19	85.19	76.60	84.34	83.13	92.70	85.40	50.50	75.40	76.00	83.40
B+SW	94.73	78.35	81.48	77.66	83.06	81.96	91.10	83.60	50.60	74.90	75.05	82.10
B+PO	93.83	79.68	85.19	77.66	84.09	82.23	91.80	84.30	52.30	71.60	75.00	82.50
B+PTO	93.83	79.86	85.19	82.98	85.46	82.54	92.00	84.30	53.50	73.20	75.75	82.70
B+SP+SW	94.62	78.26	79.63	75.53	82.01	81.72	90.90	83.50	48.90	75.10	74.60	82.00
B+PO+SP	94.17	82.08	85.19	75.53	**84.24**	**83.40**	92.80	85.90	50.80	74.70	**76.05**	**83.70**
B+PO+SW	94.62	78.70	81.48	77.66	83.12	82.07	91.00	83.90	51.20	75.30	75.35	82.30
B+PO+PTO	93.83	80.21	85.19	84.04	**85.82**	**82.74**	92.20	84.50	52.90	73.50	**75.78**	**82.80**
B+PTO+SP	93.61	81.37	85.19	81.91	**85.52**	**83.16**	92.50	85.20	49.50	78.60	**76.45**	**83.40**
B+PTO+SW	94.51	78.26	79.63	82.98	83.84	82.00	91.00	83.40	49.40	77.20	75.25	82.10
B+PO+SP+SW	94.28	78.44	81.48	76.60	82.70	81.76	91.10	83.60	48.40	75.40	74.63	82.00
B+PO+PTO+SP	93.83	81.46	85.19	80.85	85.33	83.13	92.40	85.50	49.50	77.60	76.25	83.40
B+PO+PTO+SW	94.39	79.33	77.78	79.79	82.82	82.15	91.30	84.00	49.40	75.40	75.03	82.30
B+PTO+SP+SW	94.17	77.91	79.63	81.91	83.41	81.68	91.10	83.10	46.70	77.80	74.68	81.90
ALL FEATURES	94.17	78.17	79.63	80.85	83.21	81.64	91.00	83.30	46.20	77.90	74.60	81.90

When tested on auto labeled data, the highest accuracy and F-1 for all arguments was obtained by adding a combination of PO+SP features. While the highest average accuracy for the above four arguments was obtained by adding a combination of PO+PTO features, and for the highest average F-1 values was obtained by adding the PTO+SP features. In this section, the three combinations of features above were analyzed.

Table 27 shows that the highest accuracy and F-1 for overall arguments was obtained by adding a combination of PO+SP features, that was 83.40% and 83.70%. But in more detail on the performance of ARG0, ARG1, ARG2, and ARGM-TMP, the average accuracy for these four arguments was only 84.24%. For this features combination, the ARGM-TMP accuracy was not good only 75.53%, lower than the baseline features.

The augmentation of this feature combination to the system improve the accuracy and the F-1 for overall arguments but has not resulted in stable average performance for all arguments. One possible cause is as described in Table 17 that the increase in ARGM-TMP correlates with the addition of the PTO feature.

The second highest overall accuracy performance of the argument was obtained by adding the PTO+SP features of 83.16% with F-1 of 83.40%. In more detail on the four arguments above, this features combination achieved the second highest for the average value of accuracy that was equal to 85.52%. For the average of the F-1 score on these four arguments achieved the highest score, that was 76.50%. Therefore, the addition of PTO+SP features to the baseline system is considered very good to improve the overall performance both all arguments and against the four arguments above.

While the highest average accuracy for the above four arguments was obtained by adding a combination of PO+PTO features, but the accuracy for overall arguments not really good. Therefore, the addition of these features to the baseline system is not really good for overall arguments.

Table 28. The Performance of Proposed System When Tested on Hand Labeled Data

FEATURES	ACCURACY						F-MEASURE					
	ARG0	ARG1	ARG2	ARGM-TMP	AVG	ALL	ARG0	ARG1	ARG2	ARGM-TMP	AVG	ALL
Baseline (B)	92.17	86.41	78.71	77.27	83.64	87.40	91.90	89.10	74.80	80.60	84.10	88.10
B+SP	93.19	84.45	80.00	73.55	82.80	87.20	91.80	88.90	75.40	79.70	83.95	88.00
B+SW	91.99	85.73	81.94	76.03	83.92	86.69	91.20	88.40	75.40	78.40	83.35	87.60
B+PO	91.99	86.68	79.35	73.55	82.89	87.17	91.80	89.20	76.20	78.10	83.83	88.00
B+PTO	91.99	86.14	80.00	84.71	**85.71**	**87.79**	91.90	89.00	75.20	85.10	**85.30**	**88.40**
B+SP+SW	93.00	84.18	81.29	72.31	82.70	86.54	91.20	88.30	75.90	78.30	83.43	87.60
B+PO+SP	91.71	85.60	81.94	76.03	83.82	87.02	91.60	88.70	74.90	79.80	83.75	87.90
B+PO+SW	93.19	84.92	81.29	73.14	83.13	86.93	91.40	88.70	76.40	78.50	83.75	87.80
B+PO+PTO	91.99	86.48	80.65	83.47	**85.65**	**87.88**	91.90	89.20	76.70	84.30	**85.53**	**88.50**
B+PTO+SP	91.80	85.67	84.52	79.75	85.43	87.46	91.80	88.90	75.70	82.30	84.68	88.20
B+PTO+SW	93.28	84.58	80.00	83.06	85.23	87.43	91.40	88.60	75.20	84.80	85.00	88.10
B+PO+SP+SW	92.91	84.18	81.94	71.07	82.52	86.43	91.10	88.30	76.00	77.50	83.23	87.40
B+PO+PTO+SP	91.71	86.34	83.87	78.51	85.11	87.55	91.80	89.30	75.80	81.70	84.65	88.30
B+PO+PTO+SW	93.09	84.79	80.00	77.69	83.89	86.99	91.20	88.60	75.20	81.60	84.15	87.80
B+PTO+SP+SW	93.00	83.57	81.94	78.10	84.15	86.72	91.10	88.00	74.10	82.20	83.85	87.60
ALL FEATURES	92.82	83.84	81.29	76.03	83.50	86.48	90.80	88.20	74.30	80.50	83.45	87.40

When tested on hand-labeled data, the highest accuracy and F-1 for all arguments was obtained by adding a combination of PO+PTO features. While the highest average accuracy for the above four arguments was obtained by adding PTO features, and for the highest average F-1 values was obtained the same value between

addition the combination of PO+PTO features and addition of PTO features. In this section, the three combinations of features above were analyzed.

Table 28 shows that the addition of the proposed features improves the performance for ARG0, ARG1, ARG2 and ARGM-TMP and overall arguments. From the combination of proposed features, the highest accuracy performance of overall arguments was obtained by adding PO and PTO features of 87.88%. In more detail on the performance of the ARG0, ARG1, ARG2, and ARGM-TMP, the average accuracy for the four arguments above achieved the second highest value of 85.65%, which means that the accuracy distribution for the four arguments above with the addition of these two features is quite stable.

```
Attribute Evaluator (supervised, Class (nominal): 20 Arg):
        Gain Ratio feature evaluator

Ranked attributes:
 0.6845    4 Position
 0.4826    3 Phrase Type
 0.4764   18 PT Order
 0.4561    8 NH_PP
 0.4078   19 Posit Order
 0.3436    2 Path
 0.3098   12 1St POS
 0.3033   14 Last POS
 0.2723    9 1St Word
 0.2531    6 Head Word
 0.2428   15 Syntactic Frame
 0.2408   13 2nd POS
 0.2364   11 Last Word
 0.2336   16 Cons Order
 0.2245   17 Arg Order
 0.2171   10 2nd Word
 0.0827    7 Subcategorization
 0.0553    1 Predicate
 0.0234    5 Voice

Selected attributes: 4,3,18,8,19,2,12,14,9,6,15,13,11,16,17,10,7,1,5 : 19
```

Figure 26. The Proposed Features Evaluation

To evaluate the worth of the proposed features in the dataset, re-evaluate all features using the Gain Ratio Attribute Evaluator was conducted. Figure 26 shows the results of the evaluation. The PO and PTO features were in a row of top features. The SP feature was still classified as an influential feature, while the SW feature was not very influential.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1. Conclusions

Based on the experimental results, the following conclusions are obtained:

1. The semantic argument classification on Quran data can be constructed using the training data from the Propbank corpus. The decrease in performance when the training data of the Propbank tested on the Quranic data was caused by the existing of a new argument, the argument contained in the Quran but not in the Propbank. To address this problem, some features were added to the training data to recognize the new argument. The experiment was proven that all proposed features was able to improve the performance when they were added to the baseline system separately or with some combination option.
2. The SW feature only contributed positively to ARG0 in both types of data, this was caused there were a few verses that have similar word pattern in ARG0 in Quranic data such as "those who", "among the people", "who disbelieved", "those one". The SW feature contributed negatively to ARG2, this was due to the considerable variation of the word with the same POS but are not present in Propbank, e.g. "to Allah", "to Moses", "to Satan", etc
3. The SP feature was capable of contributing positively to ARG0 and ARG1, both in terms of accuracy and F-1. The PO feature also contributed positively to ARG0 and ARG1, both in terms of accuracy and F-1. The PTO feature contributed positively to ARG2 and ARGM-TMP, both in terms of accuracy and F-1.
4. The PO+SP combination greatly affected the performance gains of accuracy and F-1 in ARG1 when tested on auto labeled data. The combination PTO+SP had greatly affect the performance gains of accuracy in ARG2 when tested on hand-labeled data and the F-1 in ARGM-TMP when tested on auto labeled data. And

the combination of PO+PTO had greatly affect the performance gains of accuracy in ARGM-TMP when tested on auto labeled data.

5. In accordance with the problem statement, by using SVM Linear, the experiment was proven that the performance of semantic argument classification on Quran data using Propbank Corpus as training data could be improved by augmenting the proposed features to the baseline system with some combination option. When tested on auto labeled data, the best performance was obtained by augmenting PTO+SP features, improved the accuracy by 1.25% and improved the F-1 score by 1.30% of the baseline system. In more detail on ARG0, ARG1, ARG2, and ARGM-TMP, these features improved the accuracy by 1.60% and improved the F-1 score by 1.68% of the baseline system. When tested on hand-labeled data, the best performance was obtained by augmenting PO+PTO features, improved the accuracy by 0.47% and improved the F-1 score by 0.40% of the baseline system. In more detail on ARG0, ARG1, ARG2, and ARGM-TMP, these features improved the accuracy by 2.00% and improved the F-1 score by 1.43% of the baseline system.

5.2. Recommendations

Based on the findings and the conclusion of the study, there are some recommendations:

1. The manual processes on argument identification need to be replaced, this is to integrate them in one system. Suitability can be improved by increasing the accuracy of argument identification process.
2. The verses of AlQuran need to be experimented using other predicates.
3. The verses of AlQuran need to be experimented using other features combinations.
4. The verses of AlQuran need to be experimented using domain adaptation method.

BIBLIOGRAPHY

[1] R. Tzong and H. Tsai, "BIOSMILE: Adapting Semantic Role Labeling for Biomedical Verbs: An Exponential Model Coupled with Automatically Generated Template Features," *Proceedings of the BioNLP Workshop on Linking Natural Language Processing and Biology at HLT-NAACL, vol. VI,* pp. 57-64, 2006.

[2] Ng, D. Dahlmeier and H. Tou, "Domain Adaptation for Semantic Role Labeling in the Biomedical Domain," *Bioinformatics Advance Access,* 2010.

[3] S. Pradhan, W.Ward and J. Martin, "Towards Robust Semantic Role Labeling," *Computational Linguistics 34(2),* pp. 289-310, 2008.

[4] D. Gildea and D. Jurafsky, "Automatic Labeling of Semantic Roles," *Association for Computational Linguistics, vol. 28, no. 3,* pp. 245-288, 2002.

[5] S. Pradhan, K. Hacioglu, V. Krugler, W. Ward, J. H. Martin and D. Jurafsky, "Support Vector Learning for Semantic Argument Classification," *Machine Learning, vol. 60, no. 1-3,* pp. 11-39, 2005.

[6] F. Ronen and S. James, The Text Mining Handbook, New York, United States of America: Cambridge University Press, 2007.

[7] Oxford Dictionaries Language Matters, "Oxford University Press," 2015. [Online]. Available: http://www.oxforddictionaries.com/definition/english/. [Accessed March 2015].

[8] M. Palmer and P. Kingsbury, "From Treebank to PropBank," *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002), Las Palmas, Spain,* 2002.

[9] M. Palmer, D. Gildea and N. Xue, "Semantic Role Labelling," *Graeme Hirst, Ed.: Morgan & Claypool Publisher,* 2010.

[10] A. B. Sharaf and E. Atwell, "Knowledge Representation pf The Quran Through Frame Semnatic - A Corpus-Based Approach," *Proceedings of the Fifth Corpus Linguistics Conference,* 2009.

[11] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuk, "Natural Language Processing (Almost) from Scratch," *Journal of Machine Learning Research 12 (2011) 2461-2505,* 2011.

[12] S. Pradhan, W. Ward, K. Hacioglu, J. H. Martin and D. Jurafsky, "Shallow Semantic Parsing using Support Vector Machines," *HLT-NAACL,* pp. 233-240, 2004.

[13] N. Xue and M. Palmer, "Calibrating Features For Semantic Role Labeling," *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'04),* 2004.

[14] K. Toutanova, A. Haghighi and C. D.Manning, "A global jointmodel for semantic role labeling," *Comput. Linguist. 34, 2 (June 2008),* p. 161–191, 2008.

[15] H. Yang and C. Zong, "Learning Generalized Features for Semantic Role Labeling," *ACM Trans Asian Low-Resour. Lang. Inf. Process, Vol. 15, No. 4, Article 28,* 2016.

[16] M. Fleischman and N. Kw, "Maximum Entropy Models for FrameNet Classification," *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP-03),* pp. 49-56, 2003.

[17] M. Vasantha and D. V. Subbiah, "Evaluation of Attribute Selection Methods with Tree based Supervised Classification-A Case Study with Mammogram Image," *International Journal of Computer Applications (0975 - 8887) Volume 8 No. 12,* 2010.

[18] E. Prasetyo, Data Mining Mengolah Data menjadi Informasi Menggunakan Matlab, Yogyakarta: Penerbit Andi, 2014.

[19] P. N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, New York: Pearson Education, 2006.

[20] V. Punyakanok, D. Roth, W.-t. Yih and D. Zimak, "Semantic role labeling via integerlinear programming inference," *Proceedings of the 20th International Conference on Computational Linguistics (COLING'04),* 2004.

[21] M. Surdeanu, S. Harabagiu, J. Williams and P. Aarseth, "Using predicate-argument structures for information extraction," *Proceedings of ACL-2003,* 2003.

[22] N. E. Ozgenzil and N. McCracken, "Semantic Role Labeling Using libSVM," *Preceedings*

of CONLL-2005, 2005.

[23] A. Haghighi, K. Toutanova and C. D. Manning, "A Joint Model for Semantic Role Labeling," *Computational Natural Language Programming,* pp. 173-176, 2005.

[24] L. Yang, "Generalizable Features Help Semantic Role Labeling," *23rd Pacific Asia Conference on Language, Information and Computation,* pp. 859-866, 2009.

[25] C. F. Baker, C. J. Fillmore and J. B. Lowe., "The Berkeley Framenet Project," *Proceedings of the COLING-ACL, Montreal, Canada,* 1998.

[26] Z. Wajdi, H. Abdelati and D. Mona, "A Pilot Propbank Annotation for Quranic Arabic," *Workshop on Computational Linguistics for Literature,* pp. 78-83, 2012.

[27] K. Dukes and T. Buckwalter, "A Dependency Treebank of the Quran using Traditional Grammar," *Proceedings of the 7th International Conference on Informatics and Systems (INFOS),* 2010.

[28] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, H. Motoda, A. Ng, B. Liu, P. S. Yu, Z. Zhou, M. Steinbach, D. J. Hand and D. Steinberg, "Top 10 Algorithms in Data Mining," *Knowledge and Information Systems, vol. 14, no. 1,* pp. 1-37, 2008.

[29] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini and C. Watkins, "Text Classification using String Kernels," *The Journal of Machine Learning Research, vol. 2,* pp. 419-444, 2002.

[30] N. Xue, "Labeling chinese predicates with semantic roles," *Comput. Linguist. 34, 2,* p. 225–255, 2008.

[31] A. Moschitti, D. Pighin and R. Basili, "Tree kernels for semantic role labeling," *Comput. Linguist. 34, 2 (June 2008),* pp. 193-224, 2008.

[32] A. Bjorkelund, B. Bohnet, L. Hafdell and P. Nugues, "A High-PerformanceSyntactic and Semantic Dependency Parser," *Coling 2010 : Demonstration Volume,* pp. 33-36, 2010.