On feature augmentation for semantic argument classification of the Quran English translation using support vector machine

Dina Khaira Batubara¹, Moch Arif Bijaksana², Adiwijaya³
School Of Computing, Telkom University, Indonesia¹,²,³
khairadina@gmail.com¹, arifbijaksana@telkomuniversity.ac.id², adiwijaya@telkomuniversity.ac.id³

Abstract. Research on the semantic argument classification requires semantically labeled data in large numbers, called corpus. Because building a corpus is costly and time-consuming, recently many studies have used existing corpus as the training data to conduct semantic argument classification research on new domain. But 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 it is not found in the training data. This research carries on semantic argument classification on a new domain that is Quran English Translation by utilizing Propbank corpus as the training data. To recognize the new argument in the training data, this research proposes four new features for extending the argument features in the training data. By using SVM Linear, the experiment has proven that augmenting the proposed features to the baseline system with some combinations option improve the performance of semantic argument classification on Quran data using Propbank Corpus as training data.

1. Introduction
1.1. Semantic Argument Classification
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 data. Semantic argument classification is also referred to as semantic role labeling (SRL) process. Originally building a good performance SRL system for one domain is conducted by manually semantic labeling process on a large data, 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 [3] [4].

However, conducting the semantic argument classification on different domains is not the same as on the same domain. 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 the existing SRL system has been tested and worked well on a sentence from the same domain, but it showed a sharp decline in performance when tested on a different domain [5]. The main
problem is when there is a new argument that found in the testing data but it is not found in the training data. To recognize the new argument in the 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 study performs 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 affected by its original languages, Arabic. In 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, we choose the verses with predicate “say” as the samples. Basic of choosing this keyword because “say” is one of the many emerging predicates in Quran 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.

By using SVM Linear Classifier, the ultimate goal is to improve the performance of semantic argument classification on Quran domain by taking the advantage of existing corpora and significantly reduce system development costs for Quran domain. This research focuses on the improvement of ARG0, ARG1, ARG2, and ARG-M-TMP by augmenting a set of feature to deal with a new argument on Quran domain.

1.2. Semantic Annotation and Propbank Corpus

The corpora or corpus is a collection of data in the form of text documents used in the case study text mining. Today, it has two kinds of the corpus that can be used for development of semantic annotation argument data, that FrameNet and PropBank (Proposition Bank). This research used Propbank corpus as training data. It is used because the annotation process between one label with another has an independent labeling. It is able to further facilitate the classification process.

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), etc. In addition to the core argument (ARG1-ARG4), there is some additional arguments marked as ARGMs. Examples of ARGMs are ARGM-LOC (indicate location), ARGM-DIR (indicates direction), ARGM-PRP (implying the destination), and so on [2]. Table 1 shows the argument label related to predicate say.

| Tag   | Description                      |
|-------|----------------------------------|
| ARG0  | Agent, The people who say        |
| ARG1  | Thing, Utterance                 |
| ARG2  | Patient, To whom said            |
| ARG3-5 | Explicit Argument               |
| ARGM  | Modifier (Additional Argument)   |

2. Related Work

The annotated corpuses FrameNet and Propbank of the initial research has greatly driven the development of Semantic Role Labeling. [6] was first introduced SRL system automatically based on statistical classifier trained on hand-annotated corpora FrameNet. In their initial work, gold or auto parse syntax tree was used as input and then extracting various lexical and syntactic features to identify the semantic role for a given predicate.

After [6], 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. For features and machine learning models, [6] as initial research used basic features
depicted on Table 2 and using the maximum entropy classifier. [7] resulted in two systems using decision tree classifier. The first system uses the same features with [6]. Then they showed performance improvement of another system which used some additional features. The state-of-the-art reported by Pradhan et al [8] where a wide range of novel features, including features extracted from named entities, verb clusters, verb sense, temporal cue word, dynamic context, etc were tested with an SVM classifier. And thereafter, most studies in SRL use SVM and maximum entropy classifier [9] [5].

In features engineering, the terms of finding the proper syntactic and semantic knowledge, the SRL researchers investigated features based on two formalisms, namely constituency grammar and dependency grammar [10] [11]. The initial basic features introduced by [6], and a number of additional features were later introduced by [12] [8] [13] [14]. It has been a long history that SRL systems have tried to use the dependence among semantic arguments.

In the Quran domain, there were some efforts to build the Quran corpus. [15] 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) [16]. [17] 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 compared them with matching frames and frame evoking verbs in the English FrameNet. These two studies focused 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.

So far, authors has not found a similar study using the Qur'an domain as test data. Hence, for analytical material, this research compare the performance of the baseline system that using the existing semantic features with the system after augmented with the new proposed features. 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.

3. Baseline & Proposed Scheme

3.1. Baseline System

Previous research shows that different features are required for different subtasks [12]. As baseline features, there are two kinds of features, the basic features, and additional features. Basic features are a set of features commonly used for semantic argument classification research, while additional features are a set of features added to the system to increase the performance. Basic features and additional features are adopted from previous research.

| Table 2. Baseline Features |
|-----------------------------|
| Basic Features [6]          | Additional Features                          |
| Predicate, Phrase Type, Path, Position, Voice, Headword, Sub-Categorization | Noun Head of PP, First/Last Word In Constituent, First/Last POS In Constituent [8] |
| Syntactic Frame [12]        | Constituent Order [18]                       |
| Argument Order [19]         |                                           |
3.2. Baseline System Performance

Table 3 shows the performance of classification using baseline features when tested on auto labeled and hand-labeled data. The performance is indicated by the value of accuracy, precision, recall, and F1. The table shows that the performance of the system increased significantly when additional features were added to the baseline system.

| 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 |

Table 4 shows the accuracy of the baseline system for the ARG0, ARG1, ARG2, and ARGM-TMP which become the focus of this research. As well as the performance of all arguments, the accuracy of these four arguments also increases when additional features are added to the system. From the results of this initial experiment, it can be concluded that the performance of the system can be improved by adding some features corresponding to the type of data.

| 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             |

3.3. Dataset

Propbank frameset data version 1.7 in XML format is used as training data. This data originally consists of 27.629 arguments. Due to resource limitations in running the experiment, resample technique using weka to filter data is conducted and only 90% data that consist of 24,865 arguments with 7261 predicates are taken.

For testing data, the Quran English translation by Ministry of Religious Affairs downloaded through the website Tanzil – Quran Navigator, translator by Shahih International is used. The Qur'an translation is unlabeled data. Because the supervised learning method is applied, the unlabeled Quran's translation data must first be labeled. There are two scenarios for the labeling process. The first scenario uses the existing automatic semantic role labeler, in this research using practnlpools 1.0. The result of this scenario is called Quran Auto Labeled Data. The second scenario uses hand-labeled data namely manually labeling the data by the author refer to Propbank annotation rule [2]. The result is called Quran Manual Labeled or Hand Labeled Data. This process generates labeled Qur'an translation data stored in XML format, according to the frameset of Propbank data.

3.4. Proposed Scheme

The steps undertaken generally consisted of feature extraction process and classification of a semantic argument. The system was able to perform the feature extraction process of semantic argument

1. http://propbank.github.io/
2. www.tanzil.net
3. https://pypi.python.org/pypi/practNLPTools/1.0
classification of an English sentence. This process played a role in producing quality features in the classification process. The general process of the system is shown in Figure 1.

![Figure 1. Experiment Scenario](image)

3.4.1. **Data Preparation.** 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].

3.4.2. **Argument Identification.** The focus of this research is to improve the argument classification process. Therefore, this research is limited on the process of argument identification which is assumed to have been done or using the data that has been conducted through the argument identification step.

3.4.3. **Preprocessing.** In preprocessing, the input were an XML file of PropBank data and Al-Quran translation data that has been structured and labeled. The necessary argument information was taken from this labeled data. This was the stage of extracting information from sentence where sentences, predicates, and arguments were taken.

3.4.4. **Feature Extraction.** Feature extraction is the process of extracting features from the data. The input was data generated form the preprocessing, where in one record data consisted of sentence, predicate, and argument. These three components were extracted and the features are shown in Table 2. The results of this process were the features extracted data, consisting of features for each record or data.

3.4.5. **Feature Evaluation.** Features evaluation is a process of selecting or evaluating the most relevant attribute on the entire data with predictive modeling problems are being worked on.
From features extracted data, then the features evaluation process was performed. This process was performed by Weka, using Gain Ratio Attribute Evaluator and search method Ranker. The process is by finding out the attributes or features from Quran domain that have a high value of correlation with the class. 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 that will be augmented to the training and testing data sets. The result obtained was the features rank based on the value of the gain ratio. Based on the results obtained in Figure 2, the proposed features were constructed by developing of Position, Phrase Type, Path, 1st POS and Last POS features. Noun Head of PP feature was not included because this feature did not work thoroughly, it was only for preposition type phrases.

3.4.6. Proposed Features. The proposed features was constructed from the most important features resulting from features evaluation process from Quran domain. The following are the proposed features developed from baseline features based on features evaluation process:

![Figure 2. Features Evaluation Process](image)

| Feature               | Description                                                                 |
|-----------------------|-----------------------------------------------------------------------------|
| Position Order (PO)   | PO was built from position and path features. PO represented the position of an argument from the predicate, but in this feature, the distance was also mentioned, for example, one position on the left, formulated into "left_1". |
| Phrase Type Order (PTO)| PTO was built from phrase type and position features. PTO represented the phrase type and position of an argument in one package, for example, "NP_left". |
| Second POS in Constituent (SP) | SP as well as the first/last POS in constituent, but they took the POS of the second word of the constituent. |
| Second Word in Constituent (SW) | SW as well as the first/last word in constituent, but they took the second word of the constituent, as a pair of SP features. |

3.4.7. Argument Classification. This research will use SVM as a classifier [8][9]. This research use libSVM [20] 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.
4. Result

Table 6 shows that the four new features improved the performance of the system with some combination option. Table 7 shows the enhancement in accuracy and F1 score for each ARG0, ARG1, ARG2, ARGM-TMP and overall arguments.

| 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        |

| Argument | Auto Labeled Data | Hand Labeled Data |
|----------|-------------------|-------------------|
|          | Acc               | F1                | Acc               | F1                |
| 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      | 1.48              | B+PO+SP          | 1.60              | B+PO+SP          | 0.47              | B+PO+PTO       | 0.40            | B+PO+PTO     |

4.1. Argument Performance

The SW feature only contributed positively to ARG0 when tested on both types of data, this was due to the Quranic data for there were quite a few verses that had similar patterns in ARG0. The SW feature contributed negatively to ARG2, this was due to the considerable variation of the word that had a large number of possible values with the same Part-Of-Speech but did 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 F1. The PO feature also contributed positively to ARG0 and ARG1, both in terms of accuracy and F1. The PTO feature contributed positively to ARG2 and ARGM-TMP, both in terms of accuracy and F1. The PO+SP combination greatly affected the performance gains of accuracy and F1 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 F1 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.

4.2. Feature Performance

The following are the features performance of the proposed system results for all arguments. Figure 3 show the features performance when tested on auto labeled data. For all arguments, the highest accuracy and F1 are obtained by adding PO+SP features, that is 83.40%, increase of 1.48% from the baseline features and 83.70%, increase by 1.60% from the baseline features. Figure 4 shows the features performance when tested on hand-labeled data. For all arguments, the highest accuracy and F1 are obtained by adding PO+PTO features. Accuracy is 87.88%, increase 0.47% from the baseline system and F1 is 88.50%, increase 0.40% from the baseline features.
5. Discussion

The objectives of this research is to consider the best feature combination which produce the best performance, this section analyzes against the performance of accuracy and F1 scores for overall arguments, taking into account the average accuracy and F1 scores of ARG0, ARG1, ARG2, and ARGM-TMP.

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. This section analyzes the three combinations of features above.

Table 8 shows that the highest accuracy and F-1 for overall arguments was obtained by adding a combination of PO+SP features, namely 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 addition of this feature combination to the system improved accuracy and F1 for overall arguments but did not result in stable average performance for all arguments. One possible cause was as described in Table 6 that the increase in ARGM-TMP correlated to the addition of the PTO feature.
Table 8. Performance of Proposed Features When Tested on Auto Labeled Data

| Features  | Performance | All Arguments | 4 Arguments | Enhancement           | Rank |
|-----------|-------------|---------------|-------------|-----------------------|------|
| PO+SP     | Accuracy    | 83.40         | 84.24       | Best Acc (All ARG)    |      |
|           |             |               |             | Low Acc (4 ARG)       |      |
|           | $F_1$       | 83.70         | 76.05       | Best $F_1$ (All ARG)  |      |
| PTO+SP    | Accuracy    | 83.16         | 85.52       | 2nd Best Acc (All ARG)| 1    |
|           |             |               |             | 2nd Best Acc (4 ARG)  |      |
|           | $F_1$       | 83.40         | 76.45       | Best $F_1$ (4 ARG)    |      |
| PO+PTO    | Accuracy    | 82.75         | 85.82       | Good Acc (All ARG)    | 2    |
|           |             |               |             | Best Acc (4 ARG)      |      |
|           | $F_1$       | 82.80         | 75.78       |                       |      |

The second highest overall accuracy performance of the argument was obtained by adding the PTO and SP features of 83.16% with $F_1$ of 83.40%. In more detail on the four arguments above, these features combination achieved the second highest for the average value of accuracy that equaled to 85.52%. For the average of the $F_1$ score on these four arguments achieved the highest score, which was 76.50%. Therefore, the addition of PTO+SP features to the baseline system was 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 was not really good. Therefore, the addition of these features to the baseline system was not really good for overall arguments.

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. This section analyzes these two combinations of features.

Table 9. Performance of Proposed Features When Tested on Hand Labeled Data

| Features  | Performance | All Arguments | 4 Arguments | Enhancement           | Rank |
|-----------|-------------|---------------|-------------|-----------------------|------|
| PO+PTO    | Accuracy    | 87.88         | 85.65       | Best Acc (All ARG)    | 1    |
|           |             |               |             | 2nd Best Acc (4 ARG)  |      |
|           | $F_1$       | 88.50         | 85.53       | Best $F_1$ (All ARG)  |      |
|           |             |               |             | Best $F_1$ (4 ARG)    |      |
| PTO+SP    | Accuracy    | 87.79         | 85.71       | 2nd Best Acc (All ARG)| 2    |
|           |             |               |             | Best Acc (4 ARG)      |      |
|           | $F_1$       | 88.40         | 76.45       | 2nd Best $F_1$ (All ARG)|      |
|           |             |               |             | 2nd Best $F_1$ (4 ARG)|      |

Table 9 shows that the addition of the proposed features improved the performance for ARG0, ARG1, ARG2 and ARG-M-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 ARG-M-TMP, the average accuracy for the four arguments above achieved the second highest value of 85.65%, this shows that the accuracy distribution for the four arguments above with the addition of these two features was quite stable. The average $F_1$ for the four arguments above also reached the highest value of 85.50%. Then the addition of these two features on the baseline system improved the accuracy and $F_1$ score as for all arguments and for the above four arguments very well and stable.
The second highest overall accuracy performance of the argument was obtained by adding the PTO feature of 87.79%. In more detail on the performance of the ARG0, ARG1, ARG2, and ARGM-TMP, the average accuracy for the four arguments above achieve the highest value of 85.71%, meaning that the accuracy distribution for the four arguments above with the addition of these two features is fairly stable. The average $F_1$ for the four arguments above reached the second highest score of 85.30%. Therefore the addition of this feature on the system improved the accuracy and $F_1$ for all and the four arguments above well and stable.

Generally, the proposed features improved the system performance. Adding some combination options of proposed features had increased the performance of ARG0, ARG1, ARG2, ARGM-TMP and all arguments. However, the counter-use of all proposed features simultaneously decreased the performance (lower than baseline). The performance declined in the use of all these features was 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.

6. Conclusions & Future Work

The semantic argument classification on Quran data can be constructed by using the training data from the Propbank corpus. The decreasing in the performance when the training data of the propbank were tested on the Quranic data was because of the new argument, the argument contained in the Quran but was not in the Propbank. To address this problem, this research proposes four new features for extending the argument features in the training data. The four new features was constructed from the most important features resulting from features evaluation process from Quran domain. From the experiment in accordance with the hypothesis, it concluded that augmentation of four new features separately or with some combination options to the baseline system improved the performance of the system. Based on the conclusion, there are some recommendations for feature work; replacing the manual process on argument identification to become integrated in one system, experimenting on verses of Quran using other predicates, using other features combinations and using domain adaptation method.

References

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

[2] 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.

[3] R. Tzong and H. T. e. al, "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.

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

[5] S. Pradhan, W. Ward and J. Martin, "Towards Robust Semantic Role Labeling," Computational Linguistics 34(2), pp. 289-310, 2008.
[6] D. Gildea and D. Jurafsky, "Automatic Labeling of Semantic Roles," *Association for Computational Linguistics*, vol. 28, no. 3, pp. 245-288, 2002.

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

[8] 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.

[9] N. E. Ozgenzil and N. McCracken, "Semantic Role Labeling Using libSVM," *Proceedings of CONLL-2005*, 2005.

[10] M. A. a. A. M. Mubarok, "Aspect-based sentiment analysis to review products using Naïve Bayes," in *AIP Conference Proceedings*, 2017.

[11] A. A. Pratiwi, "On The Feature Selection and Classification Based on Information Gain for Document Sentiment Analysis," *Applied Computational Intelligence and Soft Computing*, 2017.

[12] 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.

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

[14] 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.

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

[16] 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.

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

[18] 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.

[19] 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.
[20] C. Chang, L. A and J. Chin, "LibSVM : A Library For Support Vector Machines," ACM Transaction on Intelligent Systems and Technology; vol, 2; no. 3, pp. 27:1 - 27:27, 2011.

[21] 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.