

Dependency Parsing for Arabic Quran using Easy-First Parsing Algorithm

Alfiya El Hafsa¹, Mochammad Arif Bijaksana², Arief Fatchul Huda³

^{1,2}Department of Informatics, Universitas Telkom,

³Department of Mathematics, Universitas Islam Negeri Sunan Gunung Djati Bandung
email: elhafsa@student.telkomuniversity.ac.id¹, arifbijaksana@telkomuniversity.ac.id²,
afhuda@uinsgd.ac.id³

(Received: 4 Agustus 2020/ Accepted: 28 September 2020 / Published Online: 20 Desember 2020)

Abstract

Arabic is the main language of Al-Quran. Nowadays, many people are studying the Language of Al-Quran, called Quran Arabic. For the beginners, it is important for them to understand the syntactic relationship in a sentence found in the Qur'an. If they do not understand enough, the interpretation will be different and wrong. It will turn into dangerous because Al-Quran is a source of guidance for Muslims' life. Dependency parsing is very important for linguistic research, especially for rich languages such as the Arabic Language. This study aims to build dependency parsing, in order to make it easier to get to understand syntactic relationship information in sentences. This study uses a parsing method called deterministic parsing, which the method used is shift-reduce parsing with the Easy-First parsing algorithm. The evaluation used labelled attachment score calculation. The score generated from the evaluation was 69.7, beforehand, the comparison both the system results and the gold standard have been done. 62 sentences found the correct head and relation in each word. The number of words found to be wrong is not more than 3 words in one sentence. Evaluation scores produced are not exorbitant due to the complicated tag set used and lacking test sentences.

Keywords: Dependency Parsing, Easy-first Parsing, Quran Arabic

INTRODUCTION

Al-Qur'an is a book of guidance which is the curriculum of life for humans in their lives to achieve happiness in the afterlife (Salim, 2015). The main language of the Quran is Arabic Language. A word can be derived into many words in Arabic Language. If we take a look in Arabic dictionary, the derivative words are mostly unlisted or uninserted in the basic words, that's why Arabic Language being difficult for beginner, because they have to know the basic words first, then find the meaning of the word intended in the part (index) of basic form words. Nahwu is a branch of Arabic Language about to arranging sentences in accordance with Arabic rules, both related to the location of words in a sentence or word condition (harakat and final form) in a sentence. Nahwu focuses on how to arrange words into a perfect sentence, both in terms of words arrangement or final change of each word in a sentence that known as i'rab.

Arabian concerns about standardization by establishing grammatical, as the symbol of culture and literature. Sometimes the standards of intellectuality by using common idioms regarded as personal image of individuals. Then, the development of Arabic Language science has adopted special knowledge about grammar and its derivatives. However, for speakers who the mother tongue isn't Arabic will find difficulties, such as the practice of Arabic Language in boarding schools. Study of the desire to master Arabic was raised, there are at least three things regarding to this learning problem. First, the widespread practice of i'rab using, it leads students to find difficulties. Mastery of Arabic isn't only about the ability to translate word into sentence by using i'rab. If we take a more explore, i'rab doesn't help us to learn using Arabic, but it's about the Arabic itself (Yoyo & Mukhlis, 2019).

In the sentence "ضَرَبَ بَكْرٌ زَيْدًا" which means "Bakr has hit Zaid", there is a syntactic relationship both word 'Bakr' as the subject who 'hitting', and 'Zaid' as the object that is 'beaten'. If the syntactic relationship is wrong, then the meaning of the sentence will be different. The perpetrator will be the victim, while the victim will become the perpetrator. For that reason, a parsing dependency for Arabic Quran was developed, in order to show syntactic relationships between words in a sentence.

Parsing is a descriptive linguistics exercise that involves breaking down the text into elements of speech with evidence of the form, function, and syntax of the relationship of each part so that the text can be well understood (ThoughtCo., 2019). Easy-first parsing is a transition based parsing which is a type of syntactic parsing algorithm (Nguyen et al., 2018). The Easy-First parsing algorithm performs work based on the difficulty level of the task. Thus, the easiest jobs (simplest) are executed first before finally executing the more difficult ones. Of course the most difficult work will be put at the very last step. This algorithm works in an easy first order, unlike other deterministic parsers which are limited by parsing order (left-to-right) (Goldberg & Elhadad, 2010).

Dependency Parsing aims to identify sentences and determine their syntactic structure (Medium, 2019). Dependency parsing is a necessary process for analyzing sentence structure, especially for the Arabic language of the Quran which has a rich language. In NLP, dependency parsing is very useful for some work cases involving text data such as question answering and information extraction. Dependency parsing is important to understand the natural language, that the performance may have direct effect on NLP applications (Pei et al., 2015). For example on question "Who is the current Indonesian Minister of Education?", dependency parsing will very helpful to determine the semantic relationship between words in the information source sentence by identifying their head-dependent, so that the answer will be found.

In the field of linguistics, dependency parsing is also an aspect that plays an important role, especially for languages that are rich in morphological aspects. Many researchers have experimented with dependency parsing in various languages. In Indonesian language there is a paper studied by Kamayani and Purwarianti. The paper describes Indonesian dependency grammar based on Stanford Dependency Label. The parsing method used is deterministic parsing which is implemented in Prolog, by using Covington algorithm for free word order, it's separated the lexicon, grammar, and algorithm. The features of structures in words and dependency rules are represented using GULP (Graph Unification Logic Programming). The grammar is separated from parsing algorithm by the unification-based feature structures. The feature structures are notated and unification included in the grammar rule. The parsing won't be affected by the rule order (Kamayani & Purwarianti, 2011). This parser works for simple Indonesian sentences only, not yet able to handle sentences with complex clauses. Based on Stanford Dependencies, there are some rules can't be used in Indonesian. There is also a paper studied by (Marton et al., 2010). They delve the participation of various lexical features and inflectional dependency parsing to Arabic, using Columbia Arabic Treebank (CaTiB) based on the gold standard and prediction of POS conditions. The method used in this paper is also deterministic parsing. The paper said that the quality of parsing produced in gold is better than the quality of parsing in POS prediction conditions (Marton et al., 2010).

In this study, the method used is alike with (Kamayani & Purwarianti, 2011) and (Marton et al., 2010) work, which is using deterministic parsing method. It is said that the deterministic dependency parsing is one of the most reliable methods for parsing dependencies, providing that the deterministic parser is driven by qualified classifiers (Nivre, 2012). The gap in relevant studies (Marton et al., 2010) is that it was odd when a feature was used. Two correct dependency relations increase even though they do not use a feature, while one that uses them does not produce correct dependency relations. The parser used is

MaltParser v1.3, a transition-based parser with input buffers and stacks and also determining the next state prediction with the SVM classifier. In this study, the deterministic parsing used is shift-reduce parsing, where the Easy-First parsing algorithm by Goldberg and Elhadad is implemented. The reason for choosing to use Easy-first parsing due to the sequence of processing is pretended by making easier decisions before the more difficult ones. While, at the decision point, shift-reduce parsers usually rely on syntactic information on the left hand side, because it is limited by the processing order from left to right. For all of this reason, it is hoped that it can solve the feature's involvement problem. In addition, this parsing algorithm greatly exceeds the left-to-right deterministic algorithm (Goldberg & Elhadad, 2010). Then, the evaluation was carried out using the labeled attachment score measurement method.

The purpose of this study is to make Arabic learners easier to understand and speak Arabic fluently. Of course, this language fluency is also followed by the correct use of Arabic grammar. Therefore, it is hoped that the dependency parsing for Arabic will make people who want to learn Arabic will easily understand sentence structure.

METHOD

In this section, we will explain about the processes needed to obtain the relevant results in dependency parsing that will be evaluated with Labelled Attachment Score (LAS). Here are some of the methods used in this research. The first stage is preprocessing. Pre-processing is a process of form changing, from unstructured data into structured data. The purpose of preprocessing is to obtain data that is ready to be processed by the system by eliminating the inappropriate ones (Aqila & Bijaksana, 2020). In short, pre-processing is turning text into term indexes. In the area of Text Mining, pre-processing data is used to extract the interest and significant knowledge from unstructured text data (Kannan et al., 2015). In this stage, pre-processing is tokenization. Tokenisation is the task of cutting strings into identifiable linguistic units that are part of language data (Bird et al., 2009). Tokenization is a the process of breaking a continuous character (long text) into more meaningful or detailed parts (Taqla, 2019). A token is defined as a text section that is defined as a meaningful unit for the aim of text analysis (Mullen et al., 2018). For example, if there is a sentence *قُلْ هُوَ اللهُ أَحَدٌ*, then will be cut off per word for tokenization as follows 'قُلْ', 'هُوَ', 'الله', 'أَحَدٌ'.

Table 1. Preprocessing Data

Input Sentence	Tokenization Result
فَهُوَ فِي عَيْشَةٍ رَاضِيَةٍ	فَهُوَ', 'فِي', 'عَيْشَةٍ', 'رَاضِيَةٍ'
خَلَقْنَا الْإِنْسَانَ فِي كَبَدٍ	خَلَقْنَا', 'الْإِنْسَانَ', 'فِي', 'كَبَدٍ'
اللَّهُ يَسْتَهْزِئُ بِهِمْ	اللَّهُ', 'يَسْتَهْزِئُ', 'بِهِمْ'
وَأَرْسَلَ عَلَيْهِمْ طَيْرًا أَبَابِيلَ	وَأَرْسَلَ', 'عَلَيْهِمْ', 'طَيْرًا', 'أَبَابِيلَ'
قُلْ يَا أَيُّهَا الْكَافِرُونَ	قُلْ', 'يَا أَيُّهَا', 'الْكَافِرُونَ'
لَا أَعْبُدُ مَا تَعْبُدُونَ	وَلَا', 'أَنْتُمْ', 'أَعْبُدُونَ', 'مَا', 'أَعْبُدُ'
وَلَا أَنَا عَابِدٌ مَّا عَبَدْتُمْ	وَلَا', 'أَنَا', 'عَابِدٌ', 'مَا', 'عَبَدْتُمْ'
لَكُمْ دِينُكُمْ وَلِيَ دِينِ	لَكُمْ', 'دِينُكُمْ', 'وَلِيَ', 'دِينِ'

The next is parsing stage. Parsing is a way to break down a set of input data (for example, from a file or keyboard) and to determine the syntactic structure of each word (Wardana et al., 2019). Parsing is a process of creating sentence structure (Kamayani & Purwarianti, 2011). The parsing algorithm used is Easy-First parsing, where the parser is included in shift-reduced parsing method. The data used to carry out this step is an Arabic corpus of the Quran taken from <http://corpus.quran.com> (Dukes & Habash, 2010). The parsing process begins by creating an easy attachment decision to create multiple dependency structures, then it becomes more difficult dependency structures until well-formed dependency trees are created (Li et al., 2019). An input sentence is executed with this

parsing. The heads of two adjacent structures will be connected at each action taken. In this action, one of the structures is made as the parent of the other structure. While another structure (the child) is removed from the partial structure list. Additionally, each action is selected for a specific action pair by setting a score for the action pair based on the current parser conditions. The score determination is held by learning the data using linear models, that involving weight vector feature representation. In the course of training, the parser learns its ideas of easy and difficult, and learns to delay certain types of decisions until it has more knowledge (Goldberg & Elhadad, 2010). Every time the parsing selects an invalid action, it will be skipped and not performed. Instead, the weight parameter vector is updated by reducing the weight of features associated with invalid actions, while valid actions are increased in weight so that they have the highest current values. Then, sort the sentences based on the updated value. This process will continue to be repeated until successfully selecting a valid action. Valid action is a condition that the affirmed edge surely exists in the gold parse and also child proposed has found all of its own children.

The evaluation in this study was carried out by counting the number of words with dependencies and the correct labels in each sentence. Labeled Attachment Score (LAS) is a common evaluation metric used for dependency parsing. This testing method evaluates the output by considering how many words get the correct syntax head and also the correct label (Nivre & Fang, 2017). Of course, this calculation involves the condition of the gold standard sentence that has been tested.

RESULT AND DISCUSSION

Result

The results of tokenization on preprocessing stage can be seen in table 2. Sentences are broken up into several words, it's called tokens. Those results will be entered into the parsing stage.

Table 2. Result of Tokenization

Result of Tokenization
فَهُوَ, فِي, عَيْشَةٍ, رَاضِيَةٍ,
خَلَقْنَا, الْإِنْسَانَ, فِي, كَبَدٍ,
اللَّهُ, اِسْتَهْزِئْ, بِهِمْ,
وَأَرْسَلْ, عَلَيْهِمْ, طَيْرًا, أَبَابِيلَ,
قُلْ, يَا أَيُّهَا, الْكَافِرُونَ,
وَلَا, أَنْتُمْ, عَابِدُونَ, مَا, أَعْبُدُ,
وَلَا, أَنَا, عَابِدٌ, مَا, عَبَدْتُمْ,
لَكُمْ, دِينِكُمْ, وَلِي, دِينِ,
أُولَئِكَ, الَّذِينَ, اشْتَرَوْا, الضَّلَالَةَ, بِالْهُدَى,
فَمَا, رَبِحَتْ, تِجَارَتُهُمْ,
وَمَا, كَانُوا, مُهْتَدِينَ,
صَمٌّ, أَوْ, بُكْمٌ, أَوْ, أَعْمَى, فَهُمْ, لَا, يَرْجِعُونَ,
أُولَئِكَ, أَصْحَابُ, النَّارِ,

After going through the tokenization stage, then the test sentence going to the parsing stage, and the id from the token, token, head id, and dependency relation will appear in the result. The results of resulted parsing from several sentences can be seen in table 3 to table 6.

Table 3. Parsing Result of Input Sentence: "وَأَرْسَلَ عَلَيْهِمْ طَيْرًا أَبَابِيلَ" Quran verse (105:3)

Dependency Relation	Head_id	Form	Id
-	0	وَأَرْسَلَ (V+CONJ)	1
link	1	عَلَيْهِمْ (PP)	2
obj	2	طَيْرًا (N)	3
adj	3	أَبَابِيلَ (ADJ)	4

Table 4. Parsing Result of Input sentence: "فَهُوَ فِي عِيشَةٍ رَاضِيَةٍ" Quran verse (101:7)

Dependency Relation	Head_id	Form	Id
-	0	فَهُوَ (PRON)	1
link	1	فِي (P)	2
gen	2	عِيشَةٍ (N)	3
adj	3	رَاضِيَةٍ (ADJ)	4

Table 5. Parsing Result of Input Sentence: "اللَّهُ يَسْتَهْزِئُ بِهِمْ" Quran verse (2:15)

Dependency Relation	Head_id	Form	Id
subj	2	اللَّهُ (PN)	1
-	0	يَسْتَهْزِئُ (V)	2
link	2	بِهِمْ (PP)	3

Table 6. Parsing Result of Input Sentence: "خَلَقْنَا الْإِنْسَانَ فِي كَبَدٍ" surah Quran verse (90:4)

Dependency Relation	Head_id	Form	Id
-	0	خَلَقْنَا (PRON+V)	1
obj	1	الْإِنْسَانَ (N)	2
link	2	فِي (P)	3
gen	3	كَبَدٍ (N)	4

Tables above are shows the results of the dependency parsing of the sentences. The table 3 is about sentence "وَأَرْسَلَ عَلَيْهِمْ طَيْرًا أَبَابِيلَ", and the Table 4 is about sentence "فَهُوَ فِي عِيشَةٍ رَاضِيَةٍ". For the Table 3 and Table 4, the head id 0 is the root. Thus, the word "وَأَرْسَلَ" in Table 3 is the root, and the word "فَهُوَ" in Table 4 is the root. The head id 1 in Table 3 is "عَلَيْهِمْ" (ie the word "وَأَرْسَلَ") and in Table 4 is "فِي" (that is, the word "فَهُوَ") have a dependency relation link (link is a preposition phrase attachment). The head id number 2 in Table 3 is "طَيْرًا" (ie the word "عَلَيْهِمْ") has a dependency relation obj (obj is a verb object). The head id 2 in Table 4 is "عِيشَةٍ" (ie the word "فِي"), with dependency relation gen (gen is a preposition phrase). And the head id 3 in Table 3 is "أَبَابِيلَ" (ie the word "طَيْرًا") and in Table 4 is "رَاضِيَةٍ" (ie the word "عِيشَةٍ"), have a dependency relation adj (adj is an adjective).

Then, table 5 shows the results of the dependency parsing of the sentence "اللَّهُ يَسْتَهْزِئُ بِهِمْ". The word "اللَّهُ" in the sentence has a head id 2 (ie the word "يَسْتَهْزِئُ"), with dependency relation subj (subj is subject of a verb). Meanwhile, the word "يَسْتَهْزِئُ" has a head id 0, which means "يَسْتَهْزِئُ" is the root. The word "بِهِمْ" has a head id 2 (ie the word "يَسْتَهْزِئُ"), with a dependency relation link (preposition phrase attachment).

Table 6 above about the result of sentence "خَلَقْنَا الْإِنْسَانَ فِي كَبَدٍ". The word "خَلَقْنَا" in the sentence has a head id 0, which means "خَلَقْنَا" is the root. The word "الْإِنْسَانَ" has a head id 1 (ie the word "خَلَقْنَا"), with dependency relation obj (object of a verb). The word "فِي" has a head id

2 (ie the word "الإنسان"), with a dependency relation link (where link is a preposition phrase attachment). The word "كَبِدٌ" has a head id 3 (ie the word "في"), with a dependency relation gen (preposition phrase).

For the evaluation process, the program results are compared with the experts's result (gold standard). This comparison is held by considering head position and dependence relationship of each word in the sentence. The example results of the comparison in the input "فَهُوَ فِي عَيْشَةٍ رَّاضِيَةٍ" can be seen in Table 7. The last step that has been done is validation by measuring the system performance in parsing using Labeled Attachment Score (LAS). Provisions of comparative measurements carried out with a gold standard by experts. Based on 120 test sentences tested, the results obtained from the evaluation calculations with Labeled Attachment Score (LAS) is reached 69.7.

Table 7. The Example Result of The Comparison

Dependency Relation	Head_id	Form	Id	Result by
-	0	فَهُوَ (PRON)	1	System
<i>link</i>	1	في (P)	2	
<i>gen</i>	2	عَيْشَةٍ (N)	3	
<i>adj</i>	3	رَّاضِيَةٍ (ADJ)	4	
<i>subj</i>	2	فَهُوَ (PRON)	1	Experts (Gold Standard)
-	0	في (P)	2	
<i>gen</i>	2	عَيْشَةٍ (N)	3	
<i>adj</i>	3	رَّاضِيَةٍ (ADJ)	4	

Discussion

The results of the first stage are shown in table 2, it's called preprocessing stage, that the process carried out is tokenization. Sentences that have been tokenized will go into the next stage, that is the parsing stage by using the Easy-First parsing algorithm. The parsing stage is held by selecting the action pair that is calculating the action score. Parsing selects the action pair with the highest score. Then to select the next action pair, the score is recalculated and updated. This process will stop if the actions pair only find the root. The Table 3 to table 6 are the results of this process. The token in the sentence has a head which is another token, accompanied with the name of the dependency relationship. And tokens that have head id 0 are tokens whose dependency relationship is the root (the '-' sign in the dependency relation is referred to root). For example, in Table 5 the word اللّهُ (PN) has a head id 2, which is the word يَسْتَهْزِئُ (V) with the dependency relation subj (subject of a verb). Likewise with the word بِهِمْ (PP) has the same head as the dependent relation link (PP attachment).

In Table 7 above, there are a number of tokens in a sentence that have found the correct dependencies and relations; and also some other have not found the correct dependencies and relations. For example, in the sentence "فَهُوَ فِي عَيْشَةٍ رَّاضِيَةٍ" the system managed to find the correct dependencies, on the رَّاضِيَةٍ (ADJ) with the head في (P); and the token of عَيْشَةٍ (N) with the head عَيْشَةٍ (N). But in the same sentence, the system also failed to find the correct dependencies and relations in the tokens فَهُوَ (PRON) and في (P). Instead, the head of فَهُوَ (PRON) is في (P), and then في (P) is the root. Dependencies and relations found right here means that the head and relations found in the system are the same as those found by experts (gold standard). If the dependencies and relations found by the system are different from those found by experts, words/tokens do not count as words that find the correct

dependencies and relations in evaluation process. Thus, the tokens فُهُو (PRON) and فِي (P) won't be count as words that find the correct dependencies and relations.

Comparing results of the system's result and the gold standard by experts are obtained calculation data that can be calculated with Labeled Attachment Score (LAS). The calculation of Labeled Attachment Score (LAS) obtained is 69.7. LAS is the value obtained from the calculation of words correct number with the correct dependence on each tested sentence. The result obtained after all stages are carried out.

In a relevant study (Marton et al., 2010), the value of labeled attachment accuracy in POS prediction condition is 78.31 (gold POS conditions were better). The factors that influence this accuracy value are due to in this Marton's research there is morphological features using (the focus of their research), and there is a difference in the corpus using, which is POS tagset used has been reduced to only 6 tags (while tagset used in this study is more than 6 tags). The accuracy value in this relevant study was higher than the accuracy value in this study because the using of more complex feature contributions and corpus with more simple POS tagset. The text data used in this relevant research is better, because it did not only use text that comes from the Quran, but also used modern Arabic text which is usually used in formal education.

CONCLUSION

Based on the results of this study, the using of parsing deterministic method with the Easy-First parsing algorithm is obtained the total number of sentences that succeeded in issuing the parsing dependency with the correct head and relation in the number of 62 sentences. The highest amount of words in a sentence is 6 words, and the least number of words in a sentence is 3 words. The results in the other sentences were less than equal to 3 words in a sentence that is found incorrect parsing dependencies, and the head and its relation differ from the gold standard conditions. The result of Labeled Attachment Score (LAS) is 69.7 by comparing the results of the system and the gold standard. The result of accuracy is low because the test sentences used are a bit and the tagset used are no less than those used in relevant research. Therefore, a suggestion for further research is to improve the accuracy with more test sentences, and perhaps to use undiacritized sentences because it more flexible and have more features.

REFERENCES

- Aqila, N., & Bijaksana, M. A. (2020). Developing Set of Word Senses of Vocabulary in Al-Qur'an. *Edumatic: Jurnal Pendidikan Informatika*, 4(1), 83–90.
- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media.
- Dukes, K., & Habash, N. (2010). Morphological annotation of quranic Arabic. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, 2530–2536. Valletta, Malta.
- Goldberg, Y., & Elhadad, M. (2010). An efficient algorithm for easy-first non-directional dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 742–750. Stroudsburg, PA, USA.
- Kamayani, M., & Purwarianti, A. (2011). Dependency parsing for Indonesian. *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, 1–5. <https://doi.org/10.1109/ICEEI.2011.6021552>
- Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J., Nithya, M., Kannan, S., & Gurusamy, V. (2015). Preprocessing Techniques for Text Mining. *International Journal of Computer Science & Communication Networks*, 5(1), 7–16.

- Li, Z., Cai, J., & Zhao, H. (2019). Effective representation for easy-first dependency parsing. *15th Pacific Rim International Conference on Artificial Intelligence*, 351–363. Springer, Cham. https://doi.org/10.1007/978-3-030-29908-8_28
- Marton, Y., Habash, N., & Rambow, O. (2010). Improving Arabic dependency parsing with lexical and inflectional morphological features. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, 13–21. Stroudsburg, PA: Association for Computational Linguistics.
- Medium. (2019). *Dependency Parsing in NLP*. Retrieved from <https://medium.com/@5hirish/dependency-parsing-in-nlp-d7ade014186>
- Mullen, L. A., Benoit, K., Keyes, O., Selivanov, D., & Arnold, J. (2018). Fast, Consistent Tokenization of Natural Language Text. *The Journal of Open Source Software*, 3(23), 655. <https://doi.org/10.21105/joss.00655>
- Nguyen, B. D., Nguyen, K. Van, & Nguyen, N. L.-T. (2018). LSTM Easy-first Dependency Parsing with Pre-trained Word Embeddings and Character-level Word Embeddings in Vietnamese. *10th International Conference on Knowledge and Systems Engineering (KSE)*, 187–192. Vietnam: IEEE. <https://doi.org/10.1109/KSE.2018.8573397>
- Nivre, J. (2012). *Storage in Deterministic Dependency Parsing*. Retrieved from <https://pdfs.semanticscholar.org/bab0/0d5cac223027b22d5ba15dc19ec56773dee0.pdf>
- Nivre, J., & Fang, C.-T. (2017). Universal dependency evaluation. *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies*, 86–95. Sweden: Association for Computational Linguistics
- Pei, W., Ge, T., & Chang, B. (2015). An effective neural network model for graph-based dependency parsing. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 313–322. China: Association for Computational Linguistics. <https://doi.org/10.3115/v1/P15-1031>
- Salim, A. G. (2015). Peran Kitab Suci Al-Qur'an dalam Menjaga Eksistensi Bahasa Arab. *Okara*, 1, 1–16.
- Taqwa, S. R. (2019). Implementasi Ekstraksi Informasi Pada Surat Keputusan Menggunakan Metode Support Vector Machine. *Skripsi*. Program Sarjana. Universitas Komputer Indonesia.
- ThoughtCo. (2019). *What Is Parsing? Definition and Examples in English Grammar*. Retrieved from <https://www.thoughtco.com/parsing-grammar-term-1691583>
- Wardana, H. K., Swanita, I., & Yohanes, B. W. (2019). Sistem Pemeriksa Pola Kalimat Bahasa Indonesia berbasis Algoritme Left-Corner Parsing dengan Stemming. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, 8(3), 211. <https://doi.org/10.22146/jnteti.v8i3.515>
- Yoyo, Y., & Mukhlis, A. (2019). Historiography of the Arabic Grammar in Europe: The Legacy of Wright's Arabic Grammar. *Proceedings of the First International Conference on Progressive Civil Society*, 317(IConProCS), 212–215. <https://doi.org/10.2991/iconprocs-19.2019.45>