Automatic Single Document Summarization for Indonesian News Article Using Abstract Meaning Representation [ONLINE]

Amany Akhyar and Masayu Leylia Khodra

September 27, 2024

Automatic Single Document Summarization For Indonesian News Article Using Abstract Meaning Representation

Amany Akhyar
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
amany@staff.stei.itb.ac.id

Masayu Leylia Khodra
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
masayu@informatika.org

***Abstract*—With the increasing number of online news sources, effective summarization becomes essential to provide readers with concise and informative content. This study focuses on developing an automatic summarization system for single Indonesian news articles using Abstract Meaning Representation (AMR). Leveraging a machine learning-based AMR parser, the system constructs sentence representations, selects subgraphs to build summary graphs, and generates summary texts. The baseline uses retrained Word2Vec and selects the top three most similar sentences via cosine similarity for ROUGE evaluation against IndoSum's abstractive summary. Despite not surpassing baseline performance, the proposed system achieves an average ROUGE-1 of 0.62833, ROUGE-2 of 0.54449, and ROUGE-L of 0.58889. The findings indicate that while the proposed system effectively summarizes, it tends to prioritize initial sentences during subgraph selection, which is crucial for constructing accurate summary graphs. This tendency highlights areas for further improvement. Future research can build upon these findings by employing advanced graph construction algorithms for summary graphs and alternative text generation techniques. This study contributes to ongoing efforts to enhance text summarization systems and provides valuable lessons for future research in this field.**

Keywords—Abstract Meaning Representation (AMR), Extractive Summarization, Semantic Graphs, Sentence Scoring, IndoSum

I. INTRODUCTION

The rapid proliferation of online news sources has led to an overwhelming amount of diverse information available to readers. According to [1] , 59% of internet users only read headlines rather than full articles. This behavior underscores the necessity for effective summarization techniques that can provide comprehensive overviews of news articles without requiring extensive reading time. Summaries offer more informative content than headlines while maintaining brevity [2].

Summarization, which transforms source text into shorter versions, can be categorized into extractive and abstractive methods [3]. Extractive summarization selects key text units directly from the source [4]–[6], whereas abstractive summarization rephrases the key information in a novel way [5]. The IndoSum dataset, a benchmark for Indonesian text summarization, includes both extractive and abstractive summaries [5]. The NeuralSum method [7] was the best summarization method applied in the paper that proposed the IndoSum dataset. The NeuralSum method is a single document extractive summarization method with 300 dimensions word embedding. This method obtains an F1 ROUGE-1 of 0.6796.

Research [8] can exceed the results of the NeuralSum method with extractive summarization using sentences clustering method, which uses Word2Vec as the representation and pays attention to sentence semantics (sentence similarity), thus reached F1 ROUGE-1 of 0.72. Another way to represent sentence semantics is by using Abstract Meaning Representation (AMR). Research [6] applied AMR to do extractive summarization of single documents on the CNN/Dailymail English dataset. The F1 ROUGE-1 obtained by applying this technique is 0.831. It can surpass other studies that use the representation of word embeddings and recurrent neural network (RNN) [4], [9], [10].

Research on Indonesian text summarization, particularly using the IndoSum dataset, has seen limited exploration in abstractive methods [11]–[13]. On the other hand, extractive methods have been more extensively studied [11], [13]–[16]. One approach uses sequences of words (from sequential pattern mining) as text representation, which preserves the meaning of the text, thus enhancing readability and improving the performance of unsupervised deep learning in producing summaries [14]. Another approach incorporates feature weighting, has demonstrated improved computational efficiency compared to neural network-based methods [15]. Additionally, graph-based representations, where sentences are treated as nodes connected by semantic similarity, pairs with genetic algorithms are shown good result and efficiently select key information as summary [16]. Our work contributes to the extractive summarization field by employing Abstract Meaning Representation (AMR), a graph-based approach that captures sentence semantics while preserving key information for the summary.

This research aims to build upon these foundations by utilizing a machine learning-based AMR parser, developed by [17], to construct sentence representations for summarization. The primary contribution of this study lies in combining and modifying existing research to develop a comprehensive pipeline for summarizing Indonesian news articles. Additionally, this study adds post-processing techniques to enhance the readability of the generated summaries.

By addressing the limitations identified in previous works, such as the issues with phrase-based nodes in AMR parsing [18], and by applying sentence scoring methods to solve unordered word in generated summaries, this research provides a significant step forward in the field of Indonesian text summarization. Despite the system not surpassing the baseline performance, the insights gained offer valuable lessons for future research, making this study an important contribution to the ongoing efforts to improve automatic summarization systems.

II. RELATED WORKS

Recent studies have adopted Abstract Meaning Representation (AMR) for document summarization [18]–[27]. For instance, [21] emphasizes AMR's advantages over traditional tree structures by maintaining a single sentence structure, facilitating argument sharing with multiple predicates, and restoring implicit sentence elements for complete semantics. Additionally, [19] introduces a novel method for providing traceable summaries by aligning them with the original AMR graphs, thus offering a new approach to assessing summary faithfulness. Another study, using AMR for biomedical summarization [23], long dialogue summarization [20], and timeline summarization [26].

AMR has also gained traction in low-resource languages. For example, [27] introduces the Persian AMR (PAMR) corpus, which can be used for various natural language processing tasks, including text summarization. Meanwhile, [25] focuses on the Indonesian language, developing an AMR-to-text generation model that improves summarization performance.

The summarization steps by [18] begins with sentence clustering using agglomerative methods and cosine similarity. The longest sentence from each cluster is selected and converted into AMR graphs through a rule-based approach. These graphs are then merged into a source graph, where concept merging is based on identical words rather than phrases (the rule-based parser may generate phrases for nodes, which it shouldn't [28]). This study differs from [18] in several key aspects. First, this study focuses on single-document summarization rather than multi-document, making the clustering process unnecessary. Second, it employs a machine learning-based AMR parser [17], instead of a rule-based one. The machine learning parser generates nodes using words only, improving the concept merging process. Additionally, the study aims to enhance the parser's performance with additional training data.

III. PROPOSED SYSTEM

The details of the proposed system are divided into three main processes, such as forming sentence representation, summarization, and summary text generation. The document used for this study is the IndoSum dataset. The detail of the proposed system, including the steps carried out in each process, are shown in Figure 1.

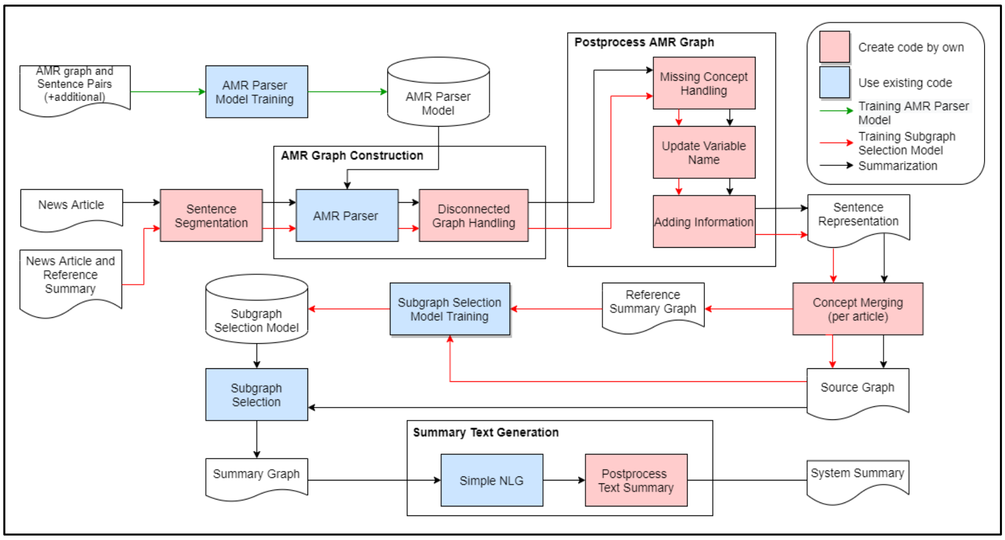

Fig. 1. Proposed System

The sentence representation (source graph) is constructed using the Indonesian AMR parser [17], known as the amr_parser library. Summarization is performed by selecting nodes and edges from the source graph (subgraph selection) utilizing the semantic_summ library [29]. The text generation for the summaries, done with Simple Natural Language Generation (Simple NLG) included in the semantic_summ library, initially produces unordered words. To address this, we introduce a post-processing step that employs a sentence scoring method to produce coherent and readable summaries. The quality of the generated summaries (system summary) is evaluated using ROUGE-1, ROUGE-2, and ROUGE-L metrics.

A. Sentence Representation with AMR

The first step is to retrain the AMR parser model with additional data. The first additional data is sentence variations which up to 5 sentence variations are made for one AMR graph. This is done so that the same AMR can be generated from a variety of different sentences. The second additional data is sentences containing conjunctions ("and" and "or"). This is done so that the AMR can be generated like conjunction phenomena according to the AMR guidelines [28].

The second step is to build a sentence representation. First, the news article is segmented and detokenized into sentences using the NLTK library and saved to the .txt file. This file will be used as the input for the AMR parser. If there is a case where a graph is disconnected, it will be handled by connecting the disconnected node to the root with the "mod" relation. The following step is handling the empty concept, which is a bug from the AMR parser. The system will delete empty concepts contained in the graph. The next step is to update the variable name by using the penman 1.2.0 library. This is done so that the variable name, which originally "vv+digit" turns into "concept initials+digit" per the AMR guidelines [28]. The next step to be taken is to generate additional information. This is done so that the .txt file has tags as follows: id, snt-type, snt, tok, alignments, node, root, edge, and AMR graph. The tag must be present for the summarization process with the semantic_summ library to be carried out. The result of this process is a .txt file, also known as sentence representation. The Figure 2 is an example of the sentence representation for "Iago Aspas yang masuk di babak kedua turut mendonasi gol (*Iago Aspas who entered in the second half also donated a goal*)".

```
# ::id 123_iago_aspas.6 ::snt-type body
# ::snt Iago Aspas yang masuk di babak kedua turut mendonasi gol.
# ::tok Iago Aspas yang masuk di babak kedua turut mendonasi gol .
# ::alignments 8-9|0 0-1|0.0 1-2|0.0.0 3-4|0.0.1 5-6|0.0.1.0 6-7|0.0.1.0.0 9-10|0.1
# ::node 0     donasi        8-9
                                        …
# ::root 0      donasi
# ::edge donasi           ARG0        iago          0               0.0
                                        …
 (d / donasi
   :ARG0 (i / iago
                                        …
```

Fig. 2. Sentence Representation

B. Summarization

The process of summarization begins by combining the AMR graph from each sentence representation into one graph (also called a combined graph). This step is done by connecting the root of each AMR graph to the main ROOT node with the relation "sent". Combining these graphs is done per news article. Next, the system will merge the same concept based on the same word and synonyms. This process is also called concept merging. In previous research [18], the concept merging is carried out only on the same word. As for getting the synonyms of a word, the Tesaurus library is used. The Figure 3 and 4 is an example of the concept merging step for "aspas" and "gol" nodes. To form a summary graph, the subgraph selection will be done. The system will seek the maximum value of the subgraph score from selected nodes and edges. This maximum value seek process is done using Integer Linear Programming (ILP). ILP generates a binary that indicates whether a node or edge is selected as a subgraph. The ILP process is carried out with several rules to make sure the summary graph is connected. This process is done using the semantic_summ library.

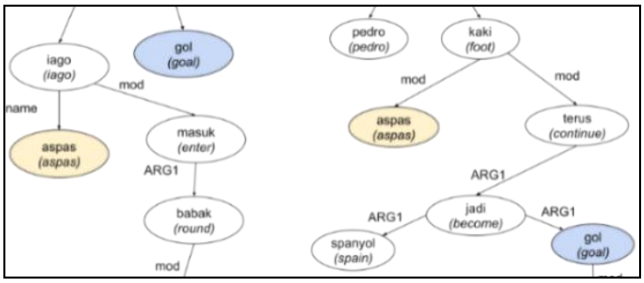

Fig. 3. Before concept merging (two nodes for "aspas" and "gol" concepts)

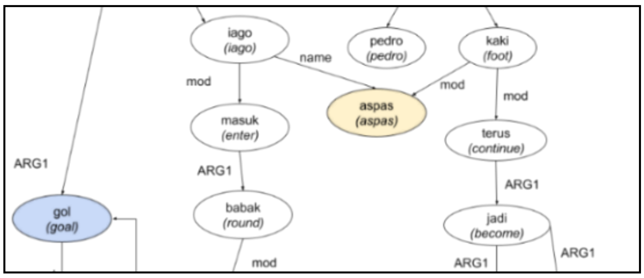

Fig. 4. After concept merging (only one node for "aspas" and "gol" concepts)

In determining the weight for node and edge, supervised learning is carried out with ILP and ramp loss. The system carried out supervised learning for 5 epochs to produce node and edge weights. For this stage, pairs of source graphs and summary graphs are needed. The resulting weight is then multiplied with the node and edge features. The result of this multiplication is the subgraph score. ILP is applied using the Gurobi library. The node and edge features used and the constraint used in the ILP are based on [18]. Figure 5 is an example of a selected subgraph. This graph is also known as a summary graph.

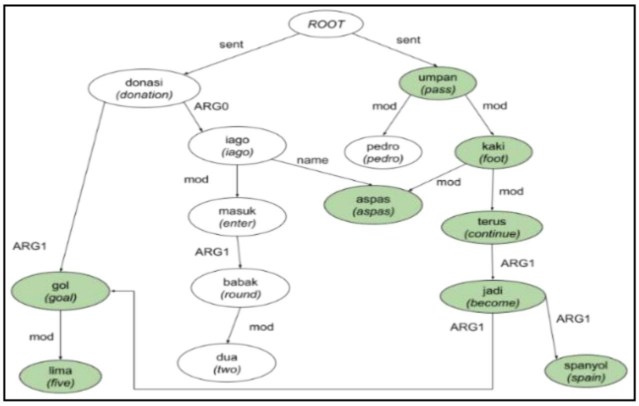

Fig. 5. Selected subgraph (or summary graph)

C. Text Generation

The summary text generation from the summary graph was done using Simple NLG based on [18]. The result is in the form of an unordered word set. In order to make the result easier to understand, postprocessing was carried out to extract the news article sentences into a summary text. Sentence extraction is done by giving a score to each sentence (sentence scoring).

Sentence scores are generated by counting the words in the sentence of the news article that matches the word set from the Simple NLG (counting) or calculating the similarity of the sentence in the news article with the word set from the Simple NLG (cosine similarity). The three sentences with the highest score will be taken and considered as the system summary. The number of sentences taken is three because the average number of sentences in IndoSum's abstractive summary is three.

IV. Results and Discussion

A. AMR Parser

In the AMR parser experiment, retraining was carried out on the AMR parser model with additional training data. The hyperparameter used is the best hyperparameter from [17], namely the XGBoost model with hyperparameter learning rate 0.1 and max depth 8. This model is then used to predict the AMR label.

In the training of the AMR parser model, 5-fold cross-validation was carried out using training data. Model V1 uses 2,124 training data including sentence variations. Model V2 is using 2,684 training data including sentence variations and conjunctions. To test each AMR parser model, predictions of the AMR label were made on the test data. The following is a comparison of the SMATCH results obtained.

From the Table I, it can be seen that Model V2 obtains the highest SMATCH. Meanwhile, a quite high increase in

SMATCH value is seen in the data of c-gedung-roboh and f-bunuh-diri. Thus, it can be concluded that Model V2 is getting better for labeling AMR news sentences. Based on this experiment, Model V2 is used as the AMR parser model for summarization.

TABLE I. THE TEST RESULT OF RETRAINED AMR PARSER MODEL

Data	Original Model [17]	Model V1	Model V2
Simple Sentence	0.82	0.83	0.83
b-salah-darat	0.68	0.68	0.68
c-gedung-roboh	0.58	0.69	0.69
d-indo-fuji	0.68	0.69	0.69
f-bunuh-diri	0.59	0.68	0.68
g-gempa-dieng	0.67	0.67	0.68

B. Summarization Experiment

In the summarization experiment, the system will summarize the validation data set of IndoSum. Table II contains the experiments with different summary text postprocessing configurations. The summary results (also called system summary) are then compared with IndoSum's abstractive summary (also called reference summary).

The first experiment was carried out by comparing the lemmatization effect on the words in the news article sentence compared with the word set. Lemmatization is a process to get the basic form of the word. Sastrawi library is used for lemmatization. The comparison of postprocess models that use lemmatization (version 1) and those that do not use lemmatization (version 2) for news article words is carried out to anticipate if there is a basic form of the word in the Simple NLG word set so that the words in news article sentences derived from these basic form of the word can still get the score. From the experiment results (Table III), it can be seen that postprocess version 2 is better than version 1 in producing system summary. Thus, postprocess version 2 will be used for further experiments.

The next experiment is adding word variation to the word set produced by Simple NLG. The word set from the Simple NLG will be added with a basic form of the word (postprocess version 3) or synonyms of the word (postprocess version 4) so that the news article sentence that has the word that corresponds to these additional words can still get an additional sentence score. The addition of the basic form of the word is using the Sastrawi library, and the synonyms are using the Tesaurus library. From the experimental results, it can be seen that the postprocess version 2 which are scored based on the Simple NLG results without any additions (neither basic form nor synonyms) is the best among version 2, 3, and 4. Thus postprocess version 2 will be used as a comparison for the next experiment.

TABLE II. THE EXPERIMENT CONFIGURATION OF SUMMARIZATION SYSTEM

Postprocess Model Name	News Article's Word Lemmatization	Additional Words for Word Set		Normalization	Sentence Score		Word Representation		Word Set as Vocabulary
		Lemmatization	Synonym		Counting	Cosine Similarity	TF	Word2Vec	
1	-	-	-	✓	✓	-	-	-	-
2	✓	-	-	✓	✓	-	-	-	-
3	✓	✓	-	✓	✓	-	-	-	-
4	✓	✓	✓	✓	✓	-	-	-	-
5	✓	-	-	-	✓	-	-	-	-
6	✓	-	-	✓	-	✓	-	✓	-
7	✓	-	-	✓	-	✓	✓	-	✓
8	✓	-	-	✓	-	✓	-	✓	✓

TABLE III. THE EXPERIMENT RESULT OF SUMMARIZATION SYSTEM

Metrics	Summary Text Postprocessing Model							
	1	2	3	4	5	6	7	8
ROUGE-1	0.629	0.630	0.628	0.617	0.625	0.375	0.633	0.619
ROUGE-2	0.550	0.551	0.550	0.537	0.540	0.255	0.550	0.533
ROUGE-L	0.593	0.594	0.592	0.579	0.585	0.312	0.594	0.577

The next experiment was not to apply normalization to sentence scores (version 5). This is to anticipate that the system does not tend to take short sentences as a system summary. The experimental results show that the postprocess that applies normalization (version 2) is better than without normalization (version 5). Thus, the best version of postprocess based on the same word count (counting) is version 2 (normalization).

The next postprocess experiment is to use cosine similarity instead of counting to score each sentence. There are two kinds of word representations used in this postprocess, namely Word2Vec and Term Frequency (TF). Postprocess version 6 scores sentences based on cosine similarity and Word2Vec representation. Each word vector from the news sentence and word process would be compared by cosine similarity and made into a matrix. The average of this similarity matrix would then be taken as a sentence score. Postprocess version 7 scores sentences based on cosine similarity and term frequency (TF) representation. The word set from the Simple NLG will be used as vocabulary, and the news sentence vector dimensions will be the same as the length of this vocabulary. Each time the vocabulary word appears in the news sentence, then the frequency value in the vector's corresponding index will be increased. The vector then would be compared by cosine similarity and taken as sentence score. In the postprocess experiment based on cosine similarity, version 7 is better than version 6. Then an experiment is carried out using the version 7 flow (with vocabulary), but with the Word2Vec representation (also called version 8).

From the experimental result, it can be seen that the postprocess version 7 is the best among all models. Thus, at the testing stage, the best postprocess (version 7) will be carried out. Table III shows the overall result of the experiment stage using the validation data set of IndoSum.

C. Summarization Testing

The next stage would be the summarization testing stage, in which the system will summarize the testing data set of IndoSum. The summary results (system summary) are then compared with IndoSum's abstractive summary (or reference summary). As a comparison, another summarization system was also developed as the baseline for this work.

The flow of the baseline is as follow. The representation used by the baseline is Word2Vec (300 dimensions) which has been retrained with the training data. Each sentence of the news article will be compared with each sentence of the reference summary to form a similarity matrix. Similarity is determined by cosine similarity. Top 3 most similar news article sentence will be labeled TRUE. The ROUGE of the TRUE labeled sentence will be calculated based on IndoSum's abstractive summary. The number of sentences labeled is three, because the average number of sentences in the reference summary is three. Table IV is the comparison between the proposed system, baseline, and related study [5].

TABLE IV. THE TESTING RESULTS

Metrics	Proposed System	Baseline	NeuralSum [5]
ROUGE-1	0.62833	0.79763	0.6796
ROUGE-2	0.54449	0.75088	0.6165
ROUGE-L	0.58889	0.78185	0.6724

Based on the summarization testing results, the AMR summary system has not exceeded the baseline performance nor the NeuralSum. The documents with the largest ROUGE-1 difference (0.93333) between proposed system and baseline are as shown in Figure 6. In Figure 6, it is seen that the AMR summarization system tends to choose the node whose original word is in the initial sentence. Figure 7 shows the word set (with English translation in italics) generated by Simple NLG in this case.

System Summary by Proposed System (ROUGE-1 = 0.0476)
Lisbon , 23 November 2016, laga Sporting CP melawan Real Madrid memasuki menit ke - 58. Wasit William Collum asal Skotlandia menghentikan pertandingan usai Bryan Ruiz melanggar Mateo Kovacic. Pandangannya tertuju pada Gareth Bale yang terduduk di tengah lapangan. .
Baseline Summary by Baseline (ROUGE-1 = 1.0)
Dibandingkan dengan dua kesebelasan rival di Spanyol, Barcelona dan Atletico Madrid, skuat yang dimiliki oleh Madrid memang jauh lebih dalam. Barcelona, rival terberat mereka saat ini di La Liga, memiliki ketergantungan yang begitu besar terhadap trio Lionel Messi, Luis Suarez, dan Neymar. Dengan kondisi skuat yang begitu dalam, Zidane tak kesulitan melakukan rotasi.

Fig. 6. The documents with the largest ROUGE-1 difference

pertandingan lapangan bryan pandangannya cp bale rival tengah tak menit ruiz menghentikan kovacic waktu lisbon menepi pasrah 23 wasit laga collum wales awal melanggar tertuju sporting gareth real terduduk madrid tampak 58 mateo menghadap memasuki kemudian begitu november lain 2016 william pemain skotlandia

(competition field bryan his gaze cp bale rival center not minute ruiz stop kovacic time lisbon step aside surrender 23 referee match collum wales early fouled fixed on sporting gareth real sit down madrid looks 58 mateo facing enter later so november other 2016 william player skotlandia)

Fig. 7. The unordered word set generated from the summary graph by using Simple NLG

Figure 8 shows the original news article (with English translation in italics and numbering as sentence order) from the same case. From the text in Figure 8, the baseline summary is formed from the underlined sentence. Those sentences are the exact same as the reference summary sentences, hence 1 as the ROUGE-1. The sentences in bold are the sentences selected by the proposed system. In the Simple NLG results, there are many words that appear in sentences that tend to be in an initial position (sentences 1 to 4). This is because the word in the initial sentence has a greater probability of being selected as a subgraph because of the greater weight of the subgraph selection model.

1) **Lisbon , 23 November 2016, laga Sporting CP melawan Real Madrid memasuki menit ke - 58.** *(Lisbon, 23 November 2016, Sporting CP match against Real Madrid had entered the 58th minute.)*
2) **Wasit William Collum asal Skotlandia menghentikan pertandingan usai Bryan Ruiz melanggar Mateo Kovacic.** *(Scottish referee William Collum stopped the match after Bryan Ruiz fouled Mateo Kovacic.)*

...

4) **Pandangannya tertuju pada Gareth Bale yang terduduk di tengah lapangan.** *(His gaze was fixed on Gareth Bale, who was sitting in the middle of the field.)*

...

16) Dibandingkan dengan dua kesebelasan rival di Spanyol, Barcelona dan Atletico Madrid, skuat yang dimiliki oleh Madrid memang jauh lebih dalam. (*Compared to the two rival teams in Spain, Barcelona and Atletico Madrid, the squad owned by Madrid is indeed much deeper.*)
17) Barcelona, rival terberat mereka saat ini di La Liga, memiliki ketergantungan yang begitu besar terhadap trio Lionel Messi, Luis Suarez, dan Neymar. (*Barcelona, their current toughest rivals in La Liga, have a huge dependence on the trio of Lionel Messi, Luis Suarez and Neymar.*)

...

21) Dengan kondisi skuat yang begitu dalam, Zidane tak kesulitan melakukan rotasi. (*With such a deep squad condition, Zidane had no trouble rotating.*)

Fig. 8. The original news article that summarized

As these nodes (which original word from the initial sentences) are selected as a summary graph, the original word would be generated by Simple NLG into the word set. In the summary text postprocess, the initial sentences would have the highest sentence score because they were much more similar to the word set than the following sentences (many terms match, thus increasing the frequency and similarity). Thus, the system summary would be formed by the initial sentences.

V. CONCLUSION

From the research that has been done, it can be concluded that the retrained AMR parser model with additional data (in the form of sentence variations and conjunctions) improves the model's performance in labeling AMR node pairs, especially on news data. The automatic summarization system for single documents in Indonesian using Abstract Meaning Representation can be built with 3 main processes: the sentence representation construction, the subgraph selection, and the summary text generation.

The machine learning-based AMR parser represents a word inside the nodes (not phrases) makes the process of concepts merging in the summary system work well. However, the performance of proposed system still cannot exceed the baseline performance. The performance of the automatic summarization system with AMR resulted in an average ROUGE-1 of 0.62833, ROUGE-2 of 0.54449, and ROUGE-L of 0.58889. The contributing factor that results in lower performance is the proposed system tends to select nodes whose original words come from the initial sentence of the news article as the summary graph, hence also extracting the initial sentence in summary text generation.

Suggestions that can be given for further research are to use the algorithm from [6] to construct a summary graph. Employing a different text generator, such as the one used in

[30], or leveraging Large Language Models (LLMs) for generating text from the graph summary can also be explored. Last but not least, another method for evaluating summary text such as ROUGE-WE [31] are recommended.

ACKNOWLEDGMENT

Amany Akhyar thank LPDP (Indonesia Endowment Fund for Education) for the financial support during her study.

REFERENCES

[1] M. Gabielkov, A. Ramachandran, A. Chaintreau, and A. Legout, "Social Clicks: What and Who Gets Read on Twitter," 2016, doi: 10.1145/2896377.2901462.

[2] J. A. León, "The effects of headlines and summaries on news comprehension and recall," *Read. Writ.*, vol. 9, no. 2, pp. 85–106, 1997, doi: 10.1023/A:1007928221187.

[3] E. Hovy and C.-Y. Lin, "Automated text summarization and the SUMMARIST system," *Proc. a Work. held Balt. Maryl. Oct. 13-15, 1998*, p. 197, 1998, doi: 10.3115/1119089.1119121.

[4] S. Dohare, H. Karnick, and V. Gupta, "Text Summarization using Abstract Meaning Representation," Jun. 2017.

[5] K. Kurniawan and S. Louvan, "IndoSum: A New Benchmark Dataset for Indonesian Text Summarization," *Proc. 2018 Int. Conf. Asian Lang. Process. IALP 2018*, pp. 215–220, Oct. 2018.

[6] R. Mishra and T. Gayen, "Automatic Lossless-Summarization of News Articles with Abstract Meaning Representation," in *Procedia Computer Science*, 2018, vol. 135, pp. 178–185, doi: 10.1016/j.procs.2018.08.164.

[7] J. Cheng and M. Lapata, "Neural Summarization by Extracting Sentences and Words," *Proc. 54th Annu. Meet. Assoc. Comput. Linguist.*, pp. 484–494, 2016, doi: 10.18653/v1/P16-1046.

[8] Z. Cai, N. Lin, C. Ma, and S. Jiang, "Indonesian automatic text summarization based on a new clustering method in sentence level," in *ACM International Conference Proceeding Series*, 2019, pp. 30–35, doi: 10.1145/3341620.3341626.

[9] R. Nallapati, B. Zhou, C. N. dos santos, C. Gulcehre, and B. Xiang, "Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond," *CoNLL 2016 - 20th SIGNLL Conf. Comput. Nat. Lang. Learn. Proc.*, pp. 280–290, Feb. 2016.

[10] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *ACL 2017 - 55th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap.*, vol. 1, pp. 1073–1083, 2017, doi: 10.18653/v1/P17-1099.

[11] Z. P. Ayudhia and S. Suyanto, "Deeper Investigation on Extractive-Abstractive Summarization for Indonesian Text," *2024 ASU Int. Conf. Emerg. Technol. Sustain. Intell. Syst. ICETSIS 2024*, no. 2022, pp. 1704–1708, 2024, doi: 10.1109/ICETSIS61505.2024.10459565.

[12] M. A. Mutasodirin and R. E. Prasojo, "Investigating Text Shortening Strategy in BERT: Truncation vs Summarization," *2021 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2021*, pp. 1–5, 2021, doi: 10.1109/ICACSIS53237.2021.9631364.

[13] R. Wijayanti, M. L. Khodra, and D. H. Widyantoro, "Single Document Summarization Using BertSum and Pointer Generator Network," *Int. J. Electr. Eng. Informatics*, vol. 13, no. 4, pp. 916–930, 2021, doi: 10.15676/ijeei.2021.13.4.10.

[14] D. S. Maylawati, Y. J. Kumar, F. Kasmin, and M. A. Ramdhani, "Deep sequential pattern mining for readability enhancement of Indonesian summarization," *Int. J. Electr. Comput. Eng.*, vol. 14, no. 1, pp. 782–795, 2024, doi: 10.11591/ijece.v14i1.pp782-795.

[15] A. Wilson and A. Zahra, "Improving Text Features in Text Summarization Using Harris Hawks Optimization," *ICIC Express Lett.*, vol. 16, no. 9, pp. 923–932, 2022, doi: 10.24507/icicel.16.09.923.

[16] N. Khotimah and A. S. Girsang, "Indonesian News Articles Summarization Using Genetic Algorithm," *Eng. Lett.*, vol. 30, no. 1, pp. 152–160, 2022.

[17] A. R. Ilmy and M. L. Khodra, "Parsing Indonesian Sentence into Abstract Meaning Representation using Machine Learning Approach," *2020 7th Int. Conf. Adv. Informatics Concepts, Theory Appl. ICAICTA 2020*, Mar. 2021.

[18] V. Severina and M. L. Khodra, "Multidocument Abstractive Summarization using Abstract Meaning Representation for Indonesian Language," in *Proceedings - 2019 International Conference on Advanced Informatics: Concepts, Theory, and Applications, ICAICTA 2019*, 2019, doi: 10.1109/ICAICTA.2019.8904449.

[19] P. Landes and B. Di Eugenio, "CALAMR: Component ALignment for Abstract Meaning Representation," *2024 Jt. Int. Conf. Comput. Linguist. Lang. Resour. Eval. Lr. 2024 - Main Conf. Proc.*, pp. 2622–2637, 2024.

[20] Y. Hua, Z. Deng, and K. McKeown, "Improving Long Dialogue Summarization with Semantic Graph Representation," *Proc. Annu. Meet. Assoc. Comput. Linguist.*, pp. 13851–13883, 2023, doi: 10.18653/v1/2023.findings-acl.871.

[21] T. Ming, H. Chen, and Y. Yang, "A Method of Semantic Redundant Information Filtering for Abstract Meaning Representation Graph," *Proc. - 2nd Int. Conf. Data Sci. Bus. Anal. ICDSBA 2018*, pp. 367–372, 2018, doi: 10.1109/ICDSBA.2018.00076.

[22] P. Kouris, G. Alexandridis, and A. Stafylopatis, "Text summarization based on semantic graphs: an abstract meaning representation graph-to-text deep learning approach," *J. Big Data*, vol. 11, no. 1, 2024, doi: 10.1186/s40537-024-00950-5.

[23] G. Frisoni, P. Italiani, S. Salvatori, and G. Moro, "Cogito Ergo Summ: Abstractive Summarization of Biomedical Papers via Semantic Parsing Graphs and Consistency Rewards," *Proc. 37th AAAI Conf. Artif. Intell. AAAI 2023*, vol. 37, pp. 12781–12789, 2023, doi: 10.1609/aaai.v37i11.26503.

[24] H. Chen, T. Ming, S. Liu, and C. Gao, "Semantic Summarization of Reconstructed Abstract Meaning Representation Graph Structure Based on Integer Linear Pragramming," *Dianzi Yu Xinxi Xuebao/Journal Electron. Inf. Technol.*, vol. 41, no. 7, pp. 1674–1681, 2019, doi: 10.11999/JEITdzyxxxb-41-7-1674.

[25] T. H. Daryanto and M. L. Khodra, "Indonesian AMR-to-Text Generation by Language Model Fine-tuning," *2022 9th Int. Conf. Adv. Informatics Concepts, Theory Appl. ICAICTA 2022*, pp. 1–6, 2022, doi: 10.1109/ICAICTA56449.2022.9932960.

[26] B. Mansouri, R. Campos, and A. Jatowt, "Towards Timeline Generation with Abstract Meaning Representation," *ACM Web Conf. 2023 - Companion World Wide Web Conf. WWW 2023*, pp. 1204–1207, 2023, doi: 10.1145/3543873.3587670.

[27] N. Tohidi, C. Dadkhah, R. N. Ganji, E. G. Sadr, and H. Elmi, "PAMR: Persian Abstract Meaning Representation Corpus," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 23, no. 3, 2024, doi: 10.1145/3638288.

[28] L. Banarescu *et al.*, "Abstract Meaning Representation for Sembanking," *Proc. 7th Linguist. Annot. Work. Interoperability with Discourse*, pp. 178–186, 2013.

[29] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N. A. Smith, "Toward Abstractive Summarization Using Semantic Representations," *Assoc. Comput. Linguist.*, pp. 1077–1086, 2015, doi: 10.3115/V1/N15-1114.

[30] Hardy and A. Vlachos, "Guided Neural Language Generation for Abstractive Summarization using Abstract Meaning Representation," *Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018*, pp. 768–773, Aug. 2018.

[31] J.-P. Ng and V. Abrecht, "Better Summarization Evaluation with Word Embeddings for ROUGE," Association for Computational Linguistics, 2015.