The Impact of Local Attention in LSTM for Abstractive Text Summarization

1st Puruso Muhammad Hanunggul
School of Computing
Telkom University
Bandung, Indonesia
hanunggulmp@gmail.com

2nd Suyanto Suyanto
School of Computing
Telkom University
Bandung, Indonesia
suyanto@telkomuniversity.ac.id

***Abstract*—An attentional mechanism is very important to enhance a neural machine translation (NMT). There are two classes of attentions: global and local attentions. This paper focuses on comparing the impact of the local attention in Long Short-Term Memory (LSTM) model to generate an abstractive text summarization (ATS). Developing a model using a dataset of Amazon Fine Food Reviews and evaluating it using dataset of GloVe shows that the global attention-based model produces better ROUGE-1, where it generates more words contained in the actual summary. But, the local attention-based gives higher ROUGE-2, where it generates more pairs of words contained in the actual summary, since the mechanism of local attention considers the subset of input words instead of the whole input words.**

Keywords—abstractive, local attention, LSTM, text summarization

I. INTRODUCTION

Text processing is one of the important artificial intelligence technologies, especially a natural language processing (NLP). One of the systems popularly developed on text processing is text summarization. Many methods have been carried out to improve the results of the text summarization. One of them is a sequence model architecture with attention mechanism. It uses an LSTM with attention mechanism based on NMT to improve the system of machine translation quality [1].

One of the models that can summarize a text with the language is similar to written by human is an abstractive text summarization (ATS) since it tends to duplicate the paraphrasing process rather than only summarizing. The text summarized by this technique looks more similar human writing and produce more concise summary than an extractive summarization technique.

There are many ways to improve the quality and performance in building the model such as augmenting the data, use multi-document dataset, or improving the attention mechanism [2]. In this case, local attention is applied to building the ATS model, a dataset containing text and summary results of each text (Golden Summary) is needed. There are few things that have to be considered such as the division of data train, validation and testing, batch size and tuning the parameter for training.

Text summarization, also known as automatic text summarization, has already researched from the late 50's. It exists due the rapid growth of web or information over loading [3]. Most researches use a sequence to sequence model, one of deep neural network methods, since this method is powerful and achieves an excellent performance on difficult task [4], [5].

A text summarization is one of tasks in NLP that uses a sequence to sequence method [6]. One of the most popular text summarization technique is a graph based approach [7]. Recurrent Neural Network (RNN) is often used for the sequence-to-sequence task. This model has been developed in different variations. Some researches provides a different seq2seq model form view point of network structures, training strategies, and summary generation algorithm [8]. One of them is by implementing the attentional encoder-decoder RNN on an ATS, where the system generates an output as a summary of the given text that uses DUC-2004 dataset. The proposed model focuses on a specific problem in ATS to improve the performance [9]. Another method to solve sequence to sequence task is by using a Long Short-Term Memory (LSTM). The idea of LSTM is quite similar to an RNN, which allows some feedback loops in the structures [10], [11], [12].

The "attention" mechanism, which is an important concept to train a neural network [13], [14], [15]. In the context of NMT, it jointly translates and aligns the words. It computes and builds an alignment model based on some hidden states [16].

An experiment state to add residual connections across time steps to RNN model, which explicitly enhances the interaction between current state and hidden states. This also allows training errors to be directly back-propagated through residual connections and effectively alleviates the gradient vanishing problem. The existed attentional mechanism is reformulated over residual connections, which also is also called residual attention [17].

NMT experiment has been conducted which lately has often been developed. From the experiment, at least it divided into two simple and effective attentional mechanism classifications: global and local attention. The former considers all hidden states in the encoding-decoding when producing a context. Conversely, local attention only considers the hidden state subset of a text that will be processed when encoding [1].

Another recent work that has shown the encoder-decoder attention mechanisms (NMT) are different from the word alignment in statistical machine translation. that attention mechanisms pay more attention to context tokens when translating ambiguous words [18].

978-1-7281-4520-4/19/$31.00 ©2019 IEEE

II. Developed System

The proposed model to produce abstractive text summaries of the given text is illustrated by Fig. 1. The detail descriptions are given in some subsections below.

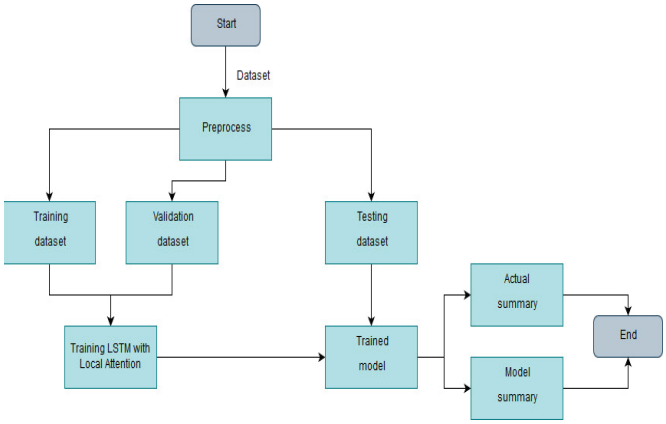

Fig. 1. Block diagram of the proposed ATS

A. Dataset

The dataset used to build the model is "Amazon Fine Food Reviews". It is a form of review in English text dataset for ATS, as illustrated by Fig. 2. Each text in the data has different sentence lengths. From all the data, the experiment takes several sample of data that processed in pre-processing the data.

	Summary	Text
0	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	"Delight" says it all	This is a confection that has been around a fe...
3	Cough Medicine	If you are looking for the secret ingredient i...
4	Great taffy	Great taffy at a great price. There was a wid...

Fig. 2. Example of Amazon Fine Food Reviews dataset

The next dataset is GloVe, which is used for word embedding. It is a Word2Vec dataset to convert a words into each vector shapes. It is a word vector with 100d, where each word is converted into a one-hundred-dimensional vector, as illustrated by Fig. 3.

```
study
[ 0.10404   0.76422  -0.44189   0.56495   0.026498 -0.25053   0.23908
  0.46664  -0.58399   0.13708  -0.35437  -0.084445  0.08687   0.38537
  0.13428   0.12989   0.30906  -0.14014  -0.25594   0.14428  -0.53372
 -0.25133  -0.43828   0.18881  -0.58895  -0.55457   0.14944  -0.82725
 -0.51495   0.013099 -0.96975   0.32704  -0.44485   0.3594   -0.55081
  0.17646  -0.16309   0.64685  -0.14992   0.14487  -1.2578    0.061788
 -0.9206   -0.20293  -0.47886  -0.03832   0.27542  -0.091164 -0.47323
 -0.3371    0.94271  -0.45062  -0.13272   0.50086  -0.23059  -2.2808
  0.014948 -0.63596   1.658     0.34713  -0.14633   0.82949   0.68876
 -0.53228   0.86526   0.43288  -0.34027   0.46548   0.70521   0.18666
  0.35918   0.55835   0.54192   0.27822  -1.0126   -0.21314  -0.036058
 -0.28431  -1.0313   -0.75662   0.29316   0.6694   -0.49838   0.13957
 -1.8072   -0.036577  0.72382  -0.65688  -0.63435   0.082457  0.34798
 -0.46783   0.1491   -0.060672  0.027716  0.50999  -0.533    -0.92447
  0.3229   -0.042366]
```

Fig. 3. Example of GloVe dataset

B. Pre-processing

Before conducting the experiment, the author has prepared several things that are used, including preparing a dataset. Because the experiment that author did is Text Summarization, the dataset will be a text along with each summary and a Word2Vec to convert the word to a vector.

To shorten training time due to the limitations of the author's device, the author determines the dataset by selecting data where the maximum number of characters is not more than 300 characters and not less than 25 characters for each text. The author analyzed the used dataset which the average number of tokens in the dataset is 38.15 tokens in the text and 2.2 tokens in the summary of the entire dataset. After the division, the data are processed through few step, consisting of clean text and tokenization, filtering, and the last is batch distribution.

In the clean text and tokenization step, all characters or symbols that are not needed on the data are omitted and all letters use lowercase. Then all data is converted into tokens in each text and summary. In the filtering step, the GloVe dataset is a reference for word embedding. From the word embedding process, there are several words that are not registered in GloVe. In this case, the model is not paying attention on the occurrence limitation of word, so that the entire word is not registered in the word embedding index (GloVe) expressed as <UNK>.

The last step is batch distribution. The dataset that has passed through the filter process will be arranged according to the batch size. From a batch, one text is selected that has the longest number of tokens for which the number will be a reference. <EOS> phrase is added at the end of every summary token. All text contained in the batch will be added to the <PAD> token until the text has the same number of tokens as the reference text. Similar things are also applied to summaries in each text. This applied to make each array in a batch has same fixed-length.

C. LSTM Model

The method used in writing this journal is LSTM. The key idea of LSTM is to combine old cell states with new ones. So that LSTM is often used to build text processing models as a Sequence-to-Sequence based method [19].

Encoder. There are two stages of encoder: forward and backward encoder [20]. The forward encoder reads the text that has been converted into vector from the front. In contrast, the backward encoder reads the sequence vector from behind. They are formulated as

$$h_t = LSTM\left(x_t, h_{t-1}\right) \tag{1}$$

where h_t is hidden state for encoder at *time step t* and x_t is the input. The LSTM has some formulas

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{2}$$

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{3}$$

$$\hat{c}_t = \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{4}$$

$$c_t = f_t \times c_{t-1} + i_t \times \hat{c}_t \tag{5}$$

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{6}$$

$$h_t = o_t \times \tanh\left(c_t\right) \tag{7}$$

where f_t, i_t, and o_t are forget gate, input gate, and output gate, respectively, $\hat{c}_t$ is vector candidate from new context, while c_t is a vector that has been upgrade or context vector [20]. Furthermore, $W \in \mathbb{R}^{h \times d}$, $U \in \mathbb{R}^{h \times h}$, and $b \in \mathbb{R}^{h}$ are trainable parameters for the matrix weights.

Decoder. The input results that have been converted into vector form or have been passed through the encoding process are read by the decoder process to produce English words. The expression on the decoder is the same as the encoder with different inputs, which is formulated as

$$s_t = LSTM\left(y_{t-1}, c_t, s_{t-1}\right) \tag{8}$$

where s_t is hidden state for decoder, y_{t-1} is the output from previous time step, and c_t is a context vector.

Local Attention. It considers a small subset of the source positions in encode-decode process when deriving the context vector [1]. The architecture of local attention is illustrated by Fig. 4.

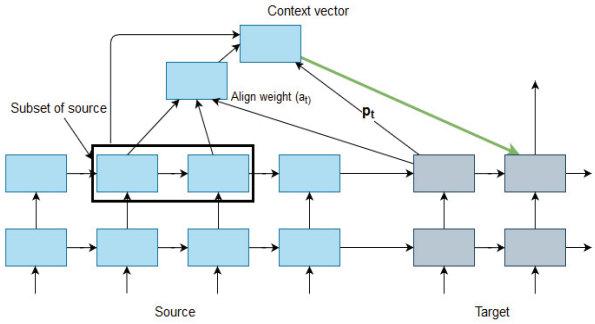

Fig. 4. Architecture of local attention

D. Model Parameters

There are several parameters that are specified for building the model. The model has 121.782 pairs of data (text and summary) based on the dataset that has been filtered. Those data are separated with 80% from filtered dataset (97.439 data) are used for training, 12% from filtered dataset (14.615 data) are used for validation, and 8% from filtered dataset (9.728 data) for testing. The number of batch size for all data is 32, with the hidden state of encoder and decoder was fixed at 300, learning rate is 0.001, and dropout is 0.5. For optimization used is Adam optimizer.

Those of all parameter are used for generating three abstractive summarization model that using LSTM, which the model consists of a model with global attention, and two models with local attention that the value is set each D = 5 and D =7.

III. Results and Discussion

Based on the model, we compare three models with different parameters. The first model is using global attention, second model is using local attention D = 5, and the last model is using local attention D = 7.

We generated the result by using data testing. The source text of data test are used as an input to the trained model. After the system give the generated summary based on data test, we compare those summary with actual summary using ROUGE.

ROUGE is a metric used to evaluate the summarization generated by the model. ROUGE measures it by looking at the similarities between summaries of model results with unigram and bigram summaries made by humans. ROUGE F1 score with each precision and recall have formula as follow.

A. Quantitative Analysis

Based on ROUGE Score, the author compares the all three model, which detailed in Table I. Those model are evaluated in ROUGE-1 and ROUGE-2 F1-score. ROUGE-1 score is measures the correctness of generated summary by seeing unigram (each word) patterns and ROUGE-2 score measures by referring to bigram (two word pairs) patterns on the text.

TABLE I. ROUGE SCORES

Model	ROUGE-1	ROUGE-2
LSTM with Global Attention	**0.05689**	0.00064
LSTM with Local Attention (D = 5)	0.04790	**0.00131**
LSTM with Local Attention (D = 7)	0.04840	0.00076

The first model, which used global attention has better performance than the other model on ROUGE-1 score 0.05689. But, also this model that has the lower performance on ROUGE-2 score 0.00064 than the other. Because the first model is using global attention, it means first model can produce a better single word for summarization but, local attention is the way better to produce two word pairs or more. The second model has highest score on ROUGE-2. It caused by the dataset itself has average number of token is 38, and most of text contain only five until ten tokens per sentence. It proves that the window length has an impact in generating the result text.

However, all models have higher ROUGE-1 score than ROUGE-2. It proves that the model can produce words for summarization correctly as individual, but not as word pairs. It caused by the summary is only contain a few words. Therefore, the system is difficult to generate correct word that equal or more than 2 pairs. This system can be enhanced using syllable-based model, instead of the word-based one, by exploiting a model of syllabification, such as described in [21], [22], [23].

B. Qualitative Analysis

Some examples of summarization results are listed in Table II, where the source text is taken from the “Amazon Fine Food Reviews” dataset. All models can generate word that even not exist in the source text. So the model not randomly generates result only based on the existing word in source text and can learn words from source text as it is an ATS technique result.

All summarizations produced by the model are hard to get a proper word as a generated result. It caused by most of dataset model contain unknown word and symbol that cannot be specified in ROUGE score. As detailed in Table 2, on the third example result, the actual summary is “Smooooooooth” not listed in the word embedding dataset.

The model can generate the same word as the actual summary but it cannot be converted into the text since it is out of vocabulary (OOV).

TABLE II. SAMPLES OF SUMMARIZATIONS PRODUCED BY EACH MODEL

Source text	My wife bought me this fruit cake this past Holiday Season, never have I had a better fruit cake, I am in my late 50's and I have had many thru the years. If you never had a good fruit cake try this one, but you have to order early in the season or you may miss out, only so many to go round. [this text is taken from the "Amazon Fine Food Reviews" dataset]
Source summ.	Best Cake Ever
Global summ.	Great !
D=5 summ.	Best ! ! ! ! ! ! ! ! ! ! ! ! ! !
D=7 summ.	Great !
Source text	This is the only tea in my house. I love its peachy taste and the ginger is very subtle. I have a cup every evening to put a soothing touch on my day. I got the pack of 6 so I would never run out! Haven't tried the other ones those are next but wanted to make sure I had enough of this one. [this text is taken from the "Amazon Fine Food Reviews" dataset]
Source summ.	Love it!
Global summ.	I
D=5 summ.	yummy!!!!!!!!!!!!!!!
D=7 summ.	Delicious!
Source text	Keurig is amazing and Green Mountain coffee is just as amazing. This is coffee that will open your eyes in the morning, as well as provide a welcome break during the day. It's not bitter, its smoooooooooooth and mellow. No aftertaste. Love it!!!!!!!!!! [this text is taken from the "Amazon Fine Food Reviews" dataset]
Source summ.	Smooooooooooth
Global summ.	<UNK>
D=5 summ.	Great!!!!!!!!!!!
D=7 summ.	<UNK>

IV. CONCLUSION

The global attention-based model produces better ROUGE-1, where it generates more words contained in the actual summary. But, the local attention-based gives higher ROUGE-2, where it generates more pairs of words contained in the actual summary, since the mechanism of local attention considers the subset of input words instead of the whole input words. Since the dataset is written using informal words, it contains a lot of symbols and unknown phrases those are not listed in the word embedding dataset. Therefore, the ROUGE score is not higher than the score from usual English text model. Resetting all parameters may give higher scores for both models. Some methods can be developed to improve the performance of both models, such us changing the dataset into any other containing article text instead of review text, rebuilding the model using more optimal parameters, or handling the OOV in data pre-processing.

REFERENCES

[1] M.-T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," arXiv, 2015.

[2] Z. Cao, W. Li, S. Li, and F. Wei, "Improving Multi-Document Summarization via Text Classification," 2016.

[3] K. Jezek and J. Steinberger, "Automatic summarizing: (The state of the art 2007 and new challenges)," Proc. Znalosti, no. February, pp. 1–12, 2008.

[4] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," pp. 1–9, 2014.

[5] R. E. Neapolitan and R. E. Neapolitan, "Neural Networks and Deep Learning," Artif. Intell., pp. 389–411, 2018.

[6] C. Khatri, G. Singh, and N. Parikh, "Abstractive and Extractive Text Summarization using Document Context Vector and Recurrent Neural Networks," 2018.

[7] J. Tan, X. Wan, and J. Xiao, "Abstractive Document Summarization with a Graph-Based Attentional Neural Model," in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017.

[8] T. Shi, Y. Keneshloo, N. Ramakrishnan, and C. K. Reddy, "Neural Abstractive Text Summarization with Sequence-to-Sequence Models," pp. 1–28, 2018.

[9] R. Nallapati, B. Zhou, C. N. dos santos, C. Gulcehre, and B. Xiang, "Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond," Feb. 2016.

[10] C.-F. Wang, "The Vanishing Gradient Problem - Towards Data Science," Towards Data Science, 2018. [Online]. Available: https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484. [Accessed: 17-Jul-2019].

[11] F. A. Gers and J. Schmidhuber, "Reccurent Nets that Time and Count," IEEE, vol. 3, 2000.

[12] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[13] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent Models of Visual Attention," pp. 1–9, 2014.

[14] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end Continuous Speech Recognition using Attention-based Recurrent NN: First Results," pp. 1–10, 2014.

[15] K. Xu et al., "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," 2015.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," pp. 1–15, 2014.

[17] C. Wang, "RRA: Recurrent Residual Attention for Sequence Learning," 2017.

[18] G. Tang, R. Sennrich, and J. Nivre, "An Analysis of Attention Mechanisms: The Case of Word Sense Disambiguation in Neural Machine Translation," vol. 1, pp. 26–35, 2019.

[19] B. Prijono, "Belajar Pembelajaran Mesin Indonesia," IndoML.com, 2018. [Online]. Available: https://indoml.com/2018/04/13/pengenalan-long-short-term-memory-lstm-dan-gated-recurrent-unit-gru-rnn-bagian-2/.

[20] J. Browniee, "Encoder-Decoder Long Short-Term Memory Networks," Machine Learning Mastery, 2017. [Online]. Available: https://machinelearningmastery.com/encoder-decoder-long-short-term-memory-networks/.

[21] E. A. Parande, S. Suyanto, "Indonesian graphemic syllabification using a nearest neighbour classifier and recovery procedure," International Journal of Speech Technology, Springer, vol. 22 no. 1, pp. 13-20 (2019). DOI: https://doi.org/10.1007/s10772-018-09569-3.

[22] S. Suyanto, "Flipping Onsets to Enhance Syllabification," International Journal of Speech Technology, Springer, Print ISSN: 1381-2416, Online ISSN: 1572-8110, vol. 22 no. 4, pp. 1031-1038 (2019). DOI: https://doi.org/10.1007/s10772-019-09649-y.

[23] S. Suyanto, S. Hartati, A. Harjoko, and D. V. Compernolle, "Indonesian Syllabification Using a Pseudo Nearest Neighbour Rule and Phonotactic Knowledge," Speech Communication, Elsevier, vol. 85, pp. 109-118 (2016). DOI: https://doi.org/10.1016/j.specom.2016.10.009.