ICIC Express Letters
Part B: Applications
Volume **14**, Number **1**, January **2023**

ICIC International ©2023 ISSN 2185-2766
pp. **29–38**

FUZZY NEURAL MACHINE TRANSLATION: A PRELIMINARY STUDY

Nhat-Hoang Phan Nguyen[1,2], Minh Nguyen Le[1,2]
Long Hong Buu Nguyen[1,2,*] and Dien Dinh[1,2]

[1]Faculty of Information Technology
University of Science, Ho Chi Minh City
227 Nguyen Van Cu Street, Ward 4, District 5, Ho Chi Minh City, Vietnam
{ npnhoang18; nlminh18 }@apcs.fitus.edu.vn; ddien@fit.hcmus.edu.vn
*Corresponding author: nhblong@fit.hcmus.edu.vn

[2]Vietnam National University, Ho Chi Minh City
Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

Received July 2022; accepted October 2022

Abstract. *Existing neural machine translation (NMT) models have achieved notable results in recent years. However, due to the complexities of extracting and understanding linguistic insights, we hypothesize that using a Fuzzy mechanism along with NMT will make the process more efficacious in expressing different aspects of word meanings. In this paper, we perform a preliminary study to improve the NMT models by integrating a Fuzzy layer into the Transformer architecture, one of the state-of-the-art encoder-decoder architectures. The experimental results on both the IWSLT'15 En-Vi and IWSLT'14 De-En datasets deliver encouraging results proving that applying fuzzy logic to the NMT models is a promising research direction.*

Keywords: Neural machine translation, Transformer, Fuzzy neural network, Fuzzy logic, Sequence-to-sequence

1. **Introduction.** Neural machine translation (NMT) [1, 2] is a relatively new technique for machine translation (MT) and has achieved auspicious results in recent years thanks to its sequence-to-sequence architecture [3, 4] and attention mechanism [5]. The most familiar framework in NMT models is the encoder-decoder framework, where the encoder is responsible for encoding the sentence in the source language into a context vector; then, a decoder relies on it to envision the context and translate it into a sentence in the target language. This framework has recently become the motivation of many studies on MT tasks.

For the translation with low-resource datasets, the techniques such as data augmentation [6], back translation [7], and exploiting multilingual word similarity [8] are applied to improving existing model performances. There is also the approach of integrating additional structural semantic information in abstract meaning representation graphs into the existing NMT model [9, 10], whose improvements rely on the understanding of the data by making use of semantic information.

Since our language has been referred to as the "shell" of our thinking [11], we communicate in natural language by making vague but superficial relations to prior mental representations [12, 13, 14]. As natural language is full of imprecision and vagueness, mapping terms and phrases between the source and target languages in translation tasks is usually not a one-to-one correspondence. As a result, rather than thinking of our natural language as a set of terms, Ross [15] suggested that we should consider it a collection of interpretations, with elements representing our mental representations and cognitive

DOI: 10.24507/icicelb.14.01.29

models. Therefore, we hypothesize that fuzzy set theory can be applied more deeply to natural language, especially in MT tasks. The NMT model then transforms the source language into interpretations instead of word surfaces before translating into the target language. We suppose this will improve the understanding and language modeling of NMT models.

To the best of our knowledge, there have been no published studies on fuzzy logic to machine translation prior to this study. Therefore, in this paper, we do a preliminary study in this approach by integrating the theory in fuzzy logic into the Transformer architecture, one of the most famous encoder-decoder architectures currently, to evaluate the feasibility of fuzzy logic in MT tasks.

The structure of this paper is represented as follows. In Section 2 we review the fundamental of our proposed models. The proposed fuzzy integrated models will be discussed in Section 3. Section 4 contains the main results of our experiments. Finally, we present our conclusions from this approach and the future works in Section 5.

2. Background.

In this section, we provide the background about Fuzzy Block and the Transformer base architecture.

2.1. Fuzzy Block.

The Fuzzy Block used in this paper, which is illustrated in Figure 1, is inspired by Deng et al. [18]. It consists of two layers besides the input layer: Membership Function Layer and Fuzzy Rule Layer. Assume the input dimension is n and the number of linguistic labels for each input is m. The Membership Function Layer assigns m fuzzy degrees, which correspond to m linguistic labels, to each input. Here Gaussian function is utilized as membership functions, as in existing works [16, 17, 18, 19]. For each input x_i, $1 \leq i \leq n$ and linguistic term j, $1 \leq j \leq m$, fuzzy degree $f_{i,j}$ of

$$f_{i,j} = e^{\frac{-\left(x_i - \mu_{i,j}\right)^2}{\sigma_{i,j}^2}} \tag{1}$$

is assigned according to two learnable parameters $\mu_{i,j}$ and $\sigma_{i,j}$; therefore, this layer introduces $O(mn)$ parameters to the model.

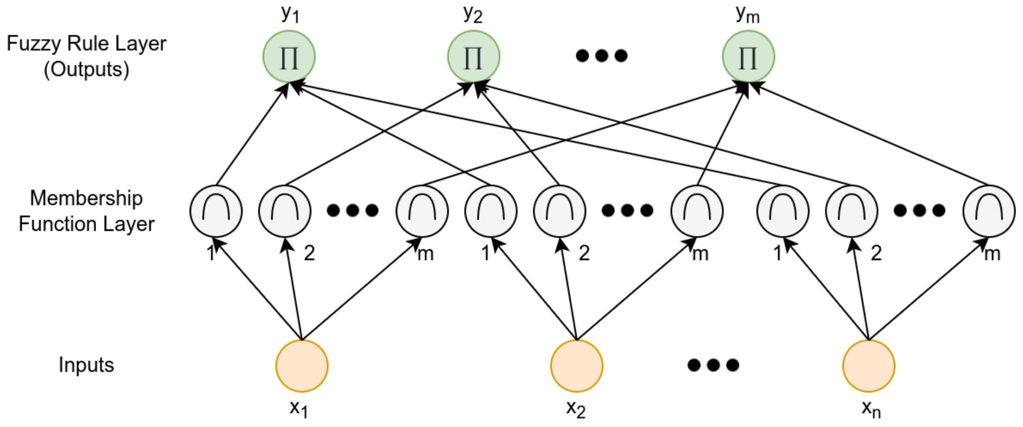

FIGURE 1. Fuzzy Block structure

In the Fuzzy Rule Layer, we realize m IF-THEN fuzzy rules of the form

Rule j: **IF** x_1 is $A_{1,j}$ AND x_2 is $A_{2,j}$ ... AND x_n is $A_{n,j}$ **THEN** y_j is B_j

where $1 \leq j \leq m$; x_i is the i-th input, $1 \leq i \leq n$; $A_{i,j}$ is the fuzzy set of the j-th label of the i-th input; y_j is the j-th output, and B_j is the consequent fuzzy set of the rule. In other words, the j-th fuzzy rule connects the j-th linguistic label of all inputs by taking conjunction of the clauses of the form "x_i is $A_{i,j}$.", which is the value of membership function assigned to x_i by the j-th linguistic label, which is $f_{i,j}$ defined in the Membership Function Layer.

Each node in m nodes of the Fuzzy Rule Layer is responsible for the fuzzy degree of a specific linguist label over all inputs by performing fuzzy "AND" operation, particularly, the node j $(1 \leq j \leq m)$ computes the product on fuzzy degrees of that linguist label over all inputs:

$$y_j = \prod_{i=1}^{n} f_{i,j} \tag{2}$$

The output of this layer is the fuzzy representation of dimension m of the fuzzy rule antecedents. This fuzzy representation could be fused with neural representation of a neural network as in the existing work [18], or be seen as features extracted from the inputs, which is the approach taken by this paper. Compared to a rule-based fuzzy system, the Fuzzy Block does not have a defuzzification process, because output fuzzy degrees are taken as extracted "features".

2.2. **Transformer architecture.** The Transformer architecture, which is proposed by Vaswani et al. [5] and is illustrated in Figure 2, is used as the baseline for our proposed models. The encoder consists of N identical layers, where the last layer output becomes the context vector for cross-attention on other N decoder layers. Each layer in the encoder has two sub-layers: multi-head self-attention, and a position-wise fully-connected feed-forward network; in these sub-layers, residual connections, followed by layer normalization, are employed. The decoder also has N identical layers, and each layer has three sub-layers:

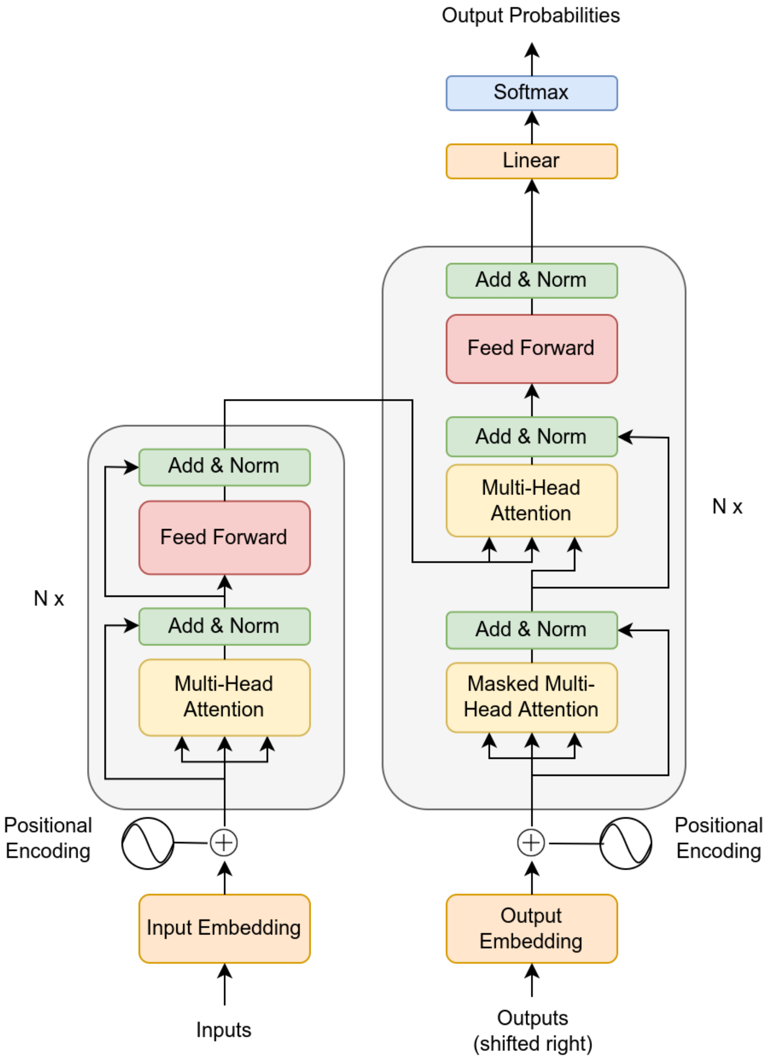

FIGURE 2. Transformer architecture [5]

masked multi-head self-attention; multi-head cross-attention, which receives keys and queries from output context vectors of the encoder; and the position-wise fully-connected feed-forward network; in each of these sub-layers, residual connections [20] and layer normalization [21] of the result are also employed. Each token flows in N encoder and N decoder layers with dimension d_{model}.

The multi-head attention sub-layer uses the scaled dot product attention; the attention matrix of the matrices of queries Q, keys K, and values V of dimension d_k, d_k, (Q and K have the same dimension d_k) and d_v, respectively, are computed as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

The product of Q and K^T is scaled by the factor $\frac{1}{\sqrt{d_k}}$ to address potential vanishing gradient problem. In the case of self-attention, Q, K, and V are the same embedded data, and in the case of cross-attention, Q would be the embedded output data, and K, V would be the context vector. For the multi-head attention, instead of single time with key dimension $d_k = d_{model}$, the attention function is performed h times, each with a separate linear projection of Q, K, V into dimensions of d_q/h, d_k/h and d_v/h, respectively, then h results are concatenated and linear projected the last time to dimension d_{model}:

$$\text{Multihead}(V, K, Q) = \text{Concat}_{i=1}^{h}(\text{Attention}(QA_i, KB_i, VC_i))W \tag{4}$$

where A_i, B_i, C_i are parameters in $\mathbb{R}^{d_k \times \frac{d_k}{h}}$, $\mathbb{R}^{d_k \times \frac{d_k}{h}}$ and $\mathbb{R}^{d_k \times \frac{d_v}{h}}$, respectively, for $1 \le i \le h$, and W is in $\mathbb{R}^{d_v \times d_{model}}$. In the decoder, the masked version of multi-head self-attention is performed to avoid attending to future tokens, by masking out all positions of illegal connection in the input of the softmax in Equation (3).

Due to no recurrence and no convolution, positional encodings are added to the input and output embeddings to inject relative and absolute positional information. Sine and cosine functions of different frequencies are utilized as positional encodings PE:

$$\begin{aligned} PE_{p,2i} &= \sin\left(\frac{p}{10000^{2i/d_{model}}}\right) \\ PE_{p,2i+1} &= \cos\left(\frac{p}{10000^{2i/d_{model}}}\right) \end{aligned} \tag{5}$$

since for any offset k, PE_{pos+k} is representable as linear function of PE_{pos}, and this allows the model to easily attend by relative positions.

3. **Proposed Models.**

Regarding the ability of fuzzy logic, we hypothesize that the performance of existing neural machine translation models could be enhanced with the Fuzzy Block since fuzzy logic rules are capable of capturing word surfaces into linguistic terms and connect these terms. In this study, we propose ten hybrid models based on the Transformer architecture and group them into three main groups. The grouping we use depends on the position of the Fuzzy Block: inside the encoder layers for the first four models, outside of the encoder layers for the following four models, and parallel to the encoder layers for the last two models.

Figure 3 describes various places to use Fuzzy Block in the encoder side of the baseline architecture for the first eight proposed models. The number inside the circle is the identifier for each proposed model. If a numbered circle is inside the encoder layer, painted light-gray, N Fuzzy Blocks are inserted or substituted for N multi-head selfattention sublayer in all N encoder layers. Otherwise, only one Fuzzy Block is inserted or substituted for the input embedding layer.

Figure 4 illustrates our model 9, where a Fuzzy Block is associated with the embedded and encoded inputs in parallel with N encoder layers. The result context vector of the encoder is the concatenation of the output of the last encoder layer and the output of the Fuzzy Block; this modifies the dimension of the context vector, which makes the model

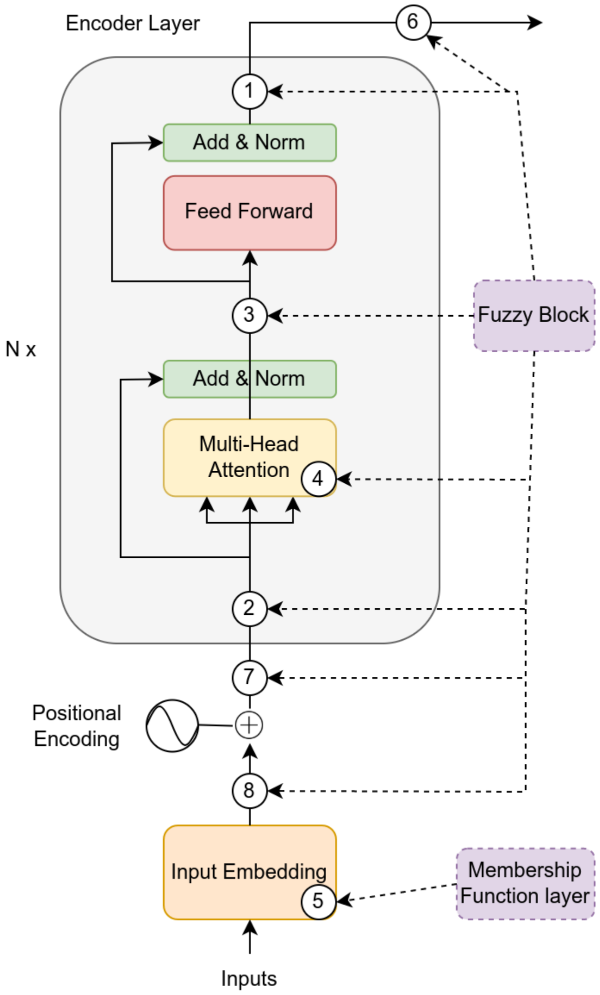

Figure 3. Proposed models with Fuzzy Block in various placements

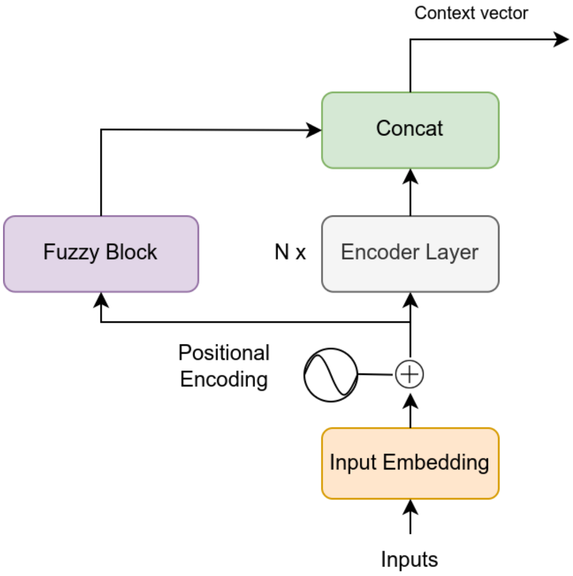

Figure 4. Model 9: Fuzzy Block parallel with N encoder layers

dimension of the encoder different from the dimension of the decoder. Depicted by Figure 5, we made a further effort to propose our model 10, where we add another Fuzzy Block to the decoder side, parallel with N decoder layers. Its output is concatenated with the

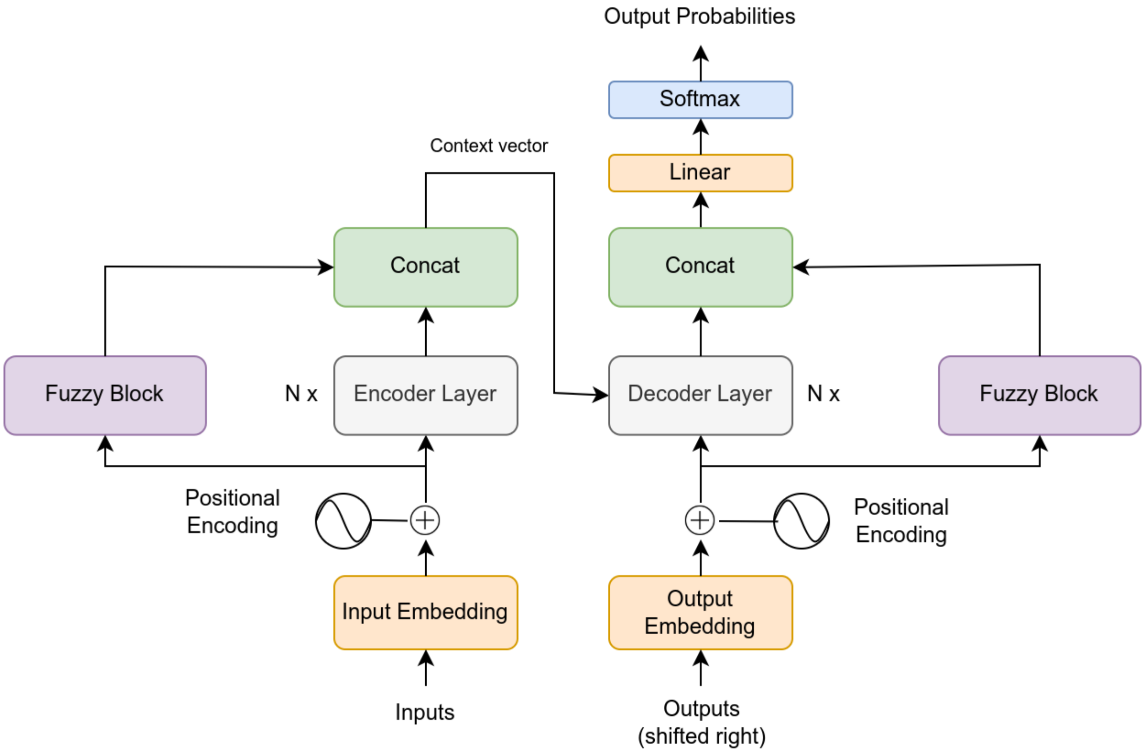

FIGURE 5. Model 10: Two Fuzzy Blocks parallel with N encoder layers and N decoder layers

output of the last decoder layer before feeding into the final linear layer. Keeping input information in models 9 and 10 is unnecessary as they are connected parallel to encoder or decoder layers; therefore, residual connections are not involved in these models.

To implement residual connections in the first eight models, the output dimension m of a Fuzzy Block should be the same as its input dimension n, i.e., $m = n$. For simplicity, the assumption $m = n = d_{model}$ is produced for all proposed models, even in those models that do not utilize it.

4. Experiment and Results.

In this section, we present how we conduct experiments to test our hypothesis, and the results of these experiments.

4.1. Datasets.

In our experiments, we used two datasets: the IWSLT'15 English-Vietnamese dataset containing about 133K pairs of sentences, proposed by Cettolo et al. [22]; the other dataset used in our experiment is the IWSLT'14 German-English dataset, containing approximately 172K parallel sentences, which was proposed by Cettolo et al. [23]. We preprocessed data before the training phase and split the corpus into training, development, and test set. The statistic of the two datasets we used is shown in Table 1.

TABLE 1. IWSLT'15 English-Vietnamese and IWSLT'14 German-English dataset summary

Dataset	**#tokens**		**#sent**	**#docs**
	En	**Vi**		
train	2.44M	2.87M	133K	1,192
dev(tst2012)	27,988	34,298	1,553	14
test(tst2013)	26,729	33,683	1,268	18
test(tst2015)	20,850	26,235	1,080	12

Dataset	**#tokens**		**#sent**	**#docs**
	De	**En**		
train	3.24M	3.46M	172K	1,361
dev(dev2012)	20.8K	21.6K	1,165	7
test(tst2013)	22.4K	23.3K	1,363	9
test(tst2014)	27.6K	28.1K	1,414	10

4.2. **Model configuration.** In our study, we used the original Transformer architecture [5] as a baseline to evaluate the efficacy of the proposed hybrid architectures. For simplicity, we used the same values of optimizer and regularization in the original work [5] for the baseline and our proposed models. Those values are

- *Optimizer*: Adam optimizer [24] with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. We also used $warmup\_steps = 4000$.
- *Regularization*: Apply dropout [25] to the output of each sub-layer before it is added to the input of the next sub-layer, with the dropout rate $p = 0.1$.

We trained our models using Google Colab with one GPU P100-PCIE-16GB. We trained for 60 epochs for each model, which took roughly 6 hours.

Because of hardware limitations, we implemented the Transformer model architecture with $d_{model} = 128$ and $d_{ff} = 512$, which means reducing the number of dimensions by a quarter compared to the original architecture proposed by Vaswani et al. [5]. This will significantly reduce the number of parameters in the model, thereby affecting the learning speed and performance of the model. However, that will not affect the outcomes of this study much, as we only focused on the performance differences of prototypes with the same configuration.

For the evaluation metric, we used BLEU (Bilingual Evaluation Understudy) score [26] to compare the similarity between hypothesis text and reference text.

4.3. **Results.** After experimenting with the Fuzzy Block integration models proposed with two datasets, IWSLT'15 English-Vietnamese and IWSLT'14 German-English, the results shown in Table 2 generally demonstrate that most models have a more elevated BLEU than the baseline's score. Specifically, for the English-Vietnamese dataset, the scores are mainly higher in the range of 0.2-0.7; meanwhile, for the German-English dataset, many of our models' scores are 0.3-2.2 more increased than the baseline model's. However, some models have poor performance, which partly reveals the essence of the Transformer architecture components. According to the results shown in Table 2:

- Putting Fuzzy Block in various positions inside the encoder does not convey a conspicuous impact when the BLEU score is almost not much different from the baseline. Remarkably, substituting self-attention with Fuzzy Block (model 4) reduces Transformer's translation capability when the results show that the BLEU declines by about 1.0 score compared to the original Transformer.
- The translation quality differs greatly depending on the placement for putting Fuzzy Block outside the encoder. For example, for IWSLT'15 English-Vietnamese, the best result is 23.12 when putting Fuzzy Block at the output of the encoder (model 6), since

Table 2. Performances of different models

Fuzzy Block placement	**Model**	**BLEU**	
		En-Vi	**De-En**
No Fuzzy Block	baseline	22.43	27.68
Fuzzy Block inside encoder	model_1	22.71	27.95
	model_2	22.47	28.11
	model_3	22.80	26.80
	model_4	21.13	26.27
Fuzzy Block outside encoder	model_5	0.88	0.89
	model_6	**23.12**	27.59
	model_7	22.46	**29.92**
	model_8	12.21	12.63
Fuzzy Block parallel with encoder/decoder	model_9	22.69	28.97
	model_10	22.71	28.77

with the dataset IWSLT'14 German-English, model 7 achieved an outstanding high score of 29.92 when placing Fuzzy Block right after input embedding and positional encoder. In contrast, putting Fuzzy Block in place of word embedding (model 5) and right after the word embedding (model 8) eventually decreases the models' recognition and translation capability.

- Finally, placing the Fuzzy Block outside the Transformer, in parallel with the encoder and decoder layers (model 9, model 10) as a feature extractor layer, both provide better results than the baseline, with an improvement of approximately 0.27 with the dataset English-Vietnamese and 1.2 with the German-English dataset. In addition, the results also indicate that putting one or two Fuzzy Blocks does not form much difference when the scores of these two models are almost similar.

Figure 6 pictures the validation score accomplished over 60 epochs in training, and the results show that the proposed models all have moderately stable performance for different datasets. For example, in both datasets, English-Vietnamese and German-English, models 9 and 10 achieve higher and earlier validation scores when compared to the original Transformer since many other models have almost no difference in validation scores.

A comparison of the convergence of models during training with 60 epochs, with two datasets, English-Vietnamese and German-English, is shown in Figure 7. The illustration demonstrates that except for the poor performance of models 5 and 8, most other models

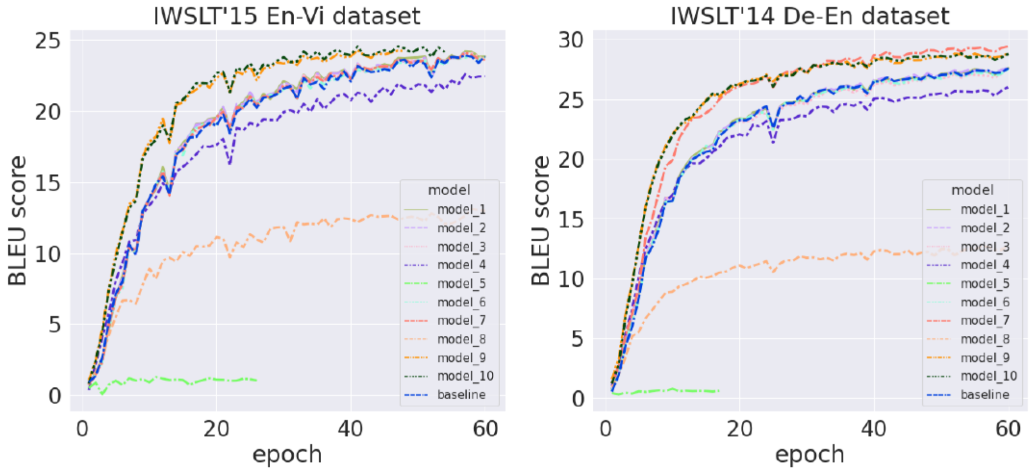

FIGURE 6. (color online) Validation over epoch of all models

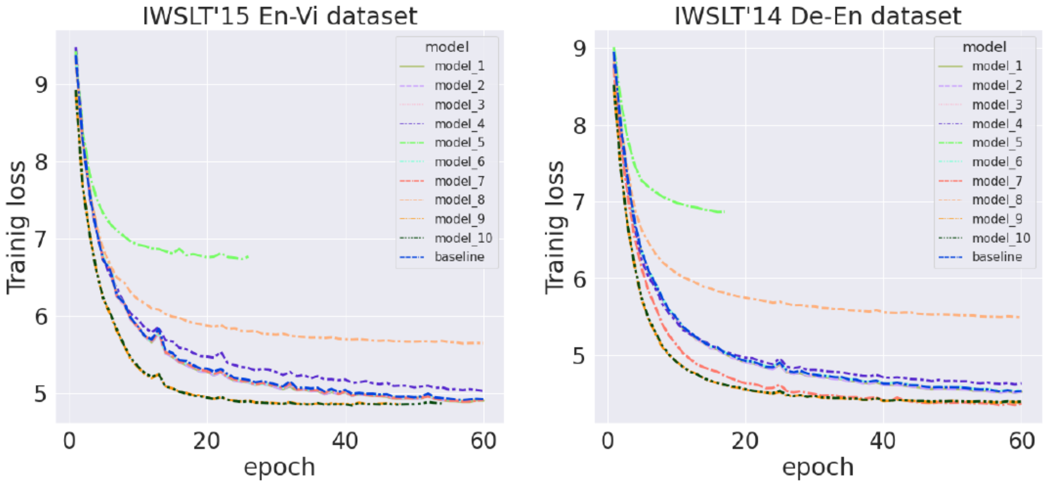

FIGURE 7. (color online) Convergence analysis of training

have a convergence rate that is not broadly distinct from the baseline model. Specifically, model 9 and model 10 have much more rapid convergence than baseline for both datasets. For the German-English dataset, model 7 also achieves the convergence rate nearly analogous to the two models where Fuzzy Block is used as the feature extractor.

5. **Conclusions.** This study experimented with the feasibility of applying fuzzy logic to neural machine translation tasks by integrating Fuzzy Blocks into Transformer architecture. As a result, most of our proposed hybrid models have more favorable translation results when compared to the original Transformer architecture. Consequently, using fuzzy logic for machine translation tasks is a probable research direction as it offers a novel approach to enhancing current models' translation quality. However, our study stops at a preliminary contention to assess the potential of this method. It is a premise for further investigations on fuzzy logic language models and language representation in the future.

Acknowledgment. This research is supported by research funding from Advanced Program in Computer Science, University of Science, Vietnam National University - Ho Chi Minh City.

REFERENCES

[1] J. Devlin et al., Fast and robust neural network joint models for statistical machine translation, *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland, 2014.

[2] D. Bahdanau, K. Cho and Y. Bengio, Neural machine translation by jointly learning to align and translate, *arXiv Preprint*, arXiv: 1409.0473, 2014.

[3] A. Graves, Generating sequences with recurrent neural networks, *arXiv Preprint*, arXiv: 1308.0850, 2013.

[4] I. Sutskever, O. Vinyals and Q. V. Le, Sequence to sequence learning with neural networks, *Advances in Neural Information Processing Systems*, vol.27, 2014.

[5] A. Vaswani et al., Attention is all you need, *Advances in Neural Information Processing Systems*, vol.30, 2017.

[6] C. Ngo and H. T. Trinh, *Styled Augmented Translation (SAT)*, https://github.com/vietai/SAT, 2021.

[7] L. D. Cuong and T. N. T. Thu, Vietnamese-English translation with transformer and back translation in VLSP 2020 machine translation shared task, *Proc. of the 7th International Workshop on Vietnamese Language and Speech Processing*, 2020.

[8] T.-V. Ngo et al., Improving multilingual neural machine translation for low-resource languages: French, English-Vietnamese, *arXiv Preprint*, arXiv: 2012.08743, 2020.

[9] L. H. B. Nguyen, V. Pham and D. Dinh, Integrating AMR to neural machine translation using graph attention networks, *2020 7th NAFOSTED Conference on Information and Computer Science (NICS)*, Ho Chi Minh City, Vietnam, 2020.

[10] L. H. B. Nguyen, V. Pham and D. Dinh, Improving neural machine translation with AMR semantic graphs, *Mathematical Problems in Engineering*, vol.2021, https://doi.org/10.1155/2021/9939389, 2021.

[11] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning – I, *Information Sciences*, vol.8, no.3, pp.199-249, 1975.

[12] F. Ungerer and H.-J. Schmid, *An Introduction to Cognitive Linguistics*, Routledge, 2013.

[13] H. A. Simon, *On the Forms of Mental Representation*, University of Minnesota Press, Minneapolis, 1978.

[14] N. Shea, *Representation in Cognitive Science*, Oxford University Press, 2018.

[15] T. J. Ross, *Fuzzy Logic with Engineering Applications*, John Wiley & Sons, 2005.

[16] F.-J. Lin, C.-H. Lin and P.-H. Shen, Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive, *IEEE Transactions on Fuzzy Systems*, vol.9, no.5, pp.751-759, 2001.

[17] C.-T. Lin et al., Support-vector-based fuzzy neural network for pattern classification, *IEEE Transactions on Fuzzy Systems*, vol.14, no.1, pp.31-41, 2006.

[18] Y. Deng et al., A hierarchical fused fuzzy deep neural network for data classification, *IEEE Transactions on Fuzzy Systems*, vol.25, no.4, pp.1006-1012, 2017.

[19] C.-F. Juang and C.-T. Lin, An online self-constructing neural fuzzy inference network and its applications, *IEEE Transactions on Fuzzy Systems*, vol.6, no.1, pp.12-32, 1998.

[20] K. He et al., Deep residual learning for image recognition, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016.

[21] J. L. Ba, J. R. Kiros and G. E. Hinton, Layer normalization, *arXiv Preprint*, arXiv: 1607.06450, 2016.

[22] M. Cettolo, J. Niehues, S. Stuker, L. Bentivogli, R. Cattoni and M. Federico, The IWSLT 2015 evaluation campaign, *Proc. of the 12th International Workshop on Spoken Language Translation*, http://workshop2015.iwslt.org, 2015.

[23] M. Cettolo et al., Report on the 11th IWSLT evaluation campaign, *Proc. of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*, pp.2-17, 2014.

[24] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *arXiv Preprint*, arXiv: 1412.6980, 2014.

[25] N. Srivastava et al., Dropout: A simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research*, vol.15, no.1, pp.1929-1958, 2014.

[26] K. Papineni et al., BLEU: A method for automatic evaluation of machine translation, *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, pp.311-318, 2002.