World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:18, No:7, 2024

Evaluating Generative Neural Attention Weights-Based Chatbot on Customer Support Twitter Dataset

Sinarwati Mohamad Suhaili, Naomie Salim, Mohamad Nazim Jambli

Abstract—Sequence-to-sequence (seq2seq) models augmented with attention mechanisms are increasingly important in automated customer service. These models, adept at recognizing complex relationships between input and output sequences, are essential for optimizing chatbot responses. Central to these mechanisms are neural attention weights that determine the model's focus during sequence generation. Despite their widespread use, there remains a gap in the comparative analysis of different attention weighting functions within seq2seq models, particularly in the context of chatbots utilizing the Customer Support Twitter (CST) dataset. This study addresses this gap by evaluating four distinct attention-scoring functions—dot, multiplicative/general, additive, and an extended multiplicative function with a *tanh* activation parameter — in neural generative seq2seq models. Using the CST dataset, these models were trained and evaluated over 10 epochs with the AdamW optimizer. Evaluation criteria included validation loss and BLEU scores implemented under both greedy and beam search strategies with a beam size of $k = 3$. Results indicate that the model with the *tanh*-augmented multiplicative function significantly outperforms its counterparts, achieving the lowest validation loss (1.136484) and the highest BLEU scores (0.438926 under greedy search, 0.443000 under beam search, k = 3). These findings emphasize the crucial influence of selecting an appropriate attention-scoring function to enhance the performance of seq2seq models for chatbots, particularly highlighting the model integrating *tanh* activation as a promising approach to improving chatbot quality in customer support contexts.

Keywords—Attention weight, chatbot, encoder-decoder, neural generative attention, score function, sequence-to-sequence.

I. Introduction

IN today's digital world, customer support plays a critical role in improving the overall user experience. To ensure efficient and effective customer support, companies are increasingly turning to chatbots. These are automated agents that have proven to be powerful alternative solutions. Chatbots are designed to handle a large number of customer inquiries across various industries, reducing human workload and shortening response time. As customer expectations for smooth and accurate interactions increase, there is a constant need to improve the performance and capabilities of chatbots. One of the common approaches to chatbot development is based on the seq2seq model, adopted from machine translation. Seq2seq models, a class of neural network architectures, increasingly form the core of these chatbot systems because of their ability to map input sequences (query) and output sequences (response) - a fundamental requirement for natural and appropriate response prediction.

Although attention mechanisms within seq2seq models are successful in several natural language processing (NLP) tasks, such as machine translation and summarization, including chatbot, little attention has been paid to the various attention weights — such as dot, multiplicative/general, additive, and modified attention weights - which could potentially impact the quality of response predictions, especially in text chatbot datasets. With this in mind, this study aims to compare and evaluate four distinct types of attention weights in the neural generative seq2seq model: dot, multiplicative/general, additive, and modified multiplicative (involving a *tanh* activation function). The performance evaluation is based on the sparse categorical loss during training and the BLEU score for different search strategies using greedy and beam search. The following is outlined main contributions of this study:

- Training and evaluating generative seq2seq models using the CST dataset with four different attention weights.
- Comparing the performance of these models using sparse categorical loss and BLEU scores.
- Identifying the attention weights that perform most effectively and consistently across different evaluation metrics.
- Gaining insight into the potential benefits and limitations of each of the attention weights.

The structure of this paper is outlined as follows: Section II reviews related work, while Section III presents a detailed description of the model. Section IV presents the methodology of our experiments. The results obtained from the experiments are presented and discussed in Section V. Finally, Section VI summarizes the research results and provides directions for future studies.

II. Related Work

Seq2seq models, characterized by an encoder-decoder (E2D) architecture, have been used extensively in various natural language processing (NLP) tasks such as chatbots, machine translation, question answering, text summarization, image captioning, and sentiment analysis. Since their introduction as a key technique for neural machine translation (NMT) [1], seq2seq models have evolved significantly. Innovations such as Long Short-Term Memory (LSTM) [2] and Gated Recurrent Units (GRU) [3] have been instrumental in overcoming the challenges associated with vanishing or

S.M. Suhaili is with the Centre for Pre-Universiti Studies, Universiti Malaysia Sarawak, Kota Samarahan, Sarawak, Malaysia 94300 (e-mail: mssinarwati@unimas.my).
N. Salim is with Universiti Teknologi Malaysia, Malaysia.
M.N. Jambli is with Universiti Malaysia Sarawak, Malaysia.

exploding gradients, thus enhancing the models' ability to process and generate more complex text sequences.

Despite the rapid advancements in NLP led by Large Language Models (LLMs) and transformer architectures, the application of these cutting-edge models in specific domains such as chatbot technology needs to be carefully considered. Although transformers represent a significant leap in attention-based mechanisms, their optimal applicability varies across different NLP tasks. Empirical evidence, such as in [4], suggests that in chatbot applications, seq2seq models perform better compared to transformer models. This finding has directed the focus of the current study to seq2seq models to improve the prediction accuracy of chatbot responses. These models, while effective, also face challenges, especially when handling complex and lengthy inputs.

Bidirectional approaches in seq2seq models, which involve modeling in reverse order, are often utilized to capture dependencies in utterances more accurately [5] [6]. However, they reach their limits with fixed-length vectors, leading to decoding problems for longer sentences [3] and potentially inaccurate responses due to compression of the input. To mitigate this issue, studies in [7] and [8] introduce attention mechanisms by augmenting another layer into the decoder and acting as an interface between the E2D structure. This layer enables the decoder to repeatedly read and search relevant parts of the source sentence, thereby enhancing the accuracy of predictions. Motivated by the effectiveness of attention mechanisms in machine translation, numerous studies have been conducted to investigate the potential of this technique in the context of chatbots. In [9], attention mechanisms were incorporated into the E2D architecture to improve the relevance between question and answer. The authors present a novel model, Hierarchical Recurrent Attention Network (HRAN), to improve context-based response generation in conversational agents. HRAN employs a hierarchical attention mechanism to capture the variability in the meaning of words and utterances within a unified framework. The effectiveness of the model is assessed by both automatic evaluation and human evaluation. The result shows that HRAN outperforms existing cutting-edge models for context-based response generation. Contrasting with HRAN, other studies have investigated the inclusion of external knowledge sources and conversational flow to improve the predictive capabilities of chatbots [10]. Another notable approach is an attention-based neural E2D architecture that leverages knowledge graph and corpus joint embedding for task-oriented systems [11].

In the context of attention efficiency, the concept of 'short attention' has been introduced to speed up the computations of the attention mechanism, which is particularly beneficial for lengthy input and output scenarios [12]. Techniques such as matrix transformation and convolutional operations have been shown to increase model efficiency and skillfully manage longer dialogue sequences. In addition, in [13] a comparative study was conducted which revealed that the attention-based bi-directional recurrent neural network (bi-RNN) model outperformed the baseline approaches in terms of the BLEU score. The study further demonstrated that the bi-directional long short-term memory (biLSTM) model performed better with Glove embeddings, while the bi-directional gated recurrent unit (biGRU) model performed better with FastText embeddings. The study also investigated the impact of complementary deep learning methods, such as batch size and hidden size of RNN, on different models of seq2seq architectures based on various word embeddings with RNN encoder types.

Despite these advancements, there is still a deficiency in the comparative literature dealing with the different weighting of attention in chatbot models. The present study attempts to address this gap by comparing and evaluating the effects of different attention weights on a chatbot dataset. The performance evaluation includes both sparse categorical loss during training and BLEU scores across multiple search strategies, such as greedy and beam search. This approach aims to develop a nuanced understanding of how variations in attention weighting affect chatbot efficiency and response quality.

III. Model Description

In this section, the model architecture implemented in this study briefly discusses the word representation model, seq2seq Learning Task Model, and neural attention.

A. Word Representation Model

To enable computers to understand text data, it must be transformed into a numerical form, a process known as embedding. This process involves mapping words or phrases to vectors of real numbers in a multidimensional space. An embedding layer, which is an essential part of this process, can be initialized with pre-trained models, FastText being a notable example. Developed by Facebook, Inc. and represents a significant advance in the field of prediction-based word embedding models. It aims to address a key limitation of conventional word embedding models that overlook the internal structure of words. In contrast to the conventional models, FastText, as detailed in [14] and [15], enhances the Skip-Gram model, proposed in [16]. It represents each word as a collection of characters n-grams, thereby acknowledging its morphology. In this model, a word's vector is computed as the sum of its n-grams, allowing the model to generate meaningful vectors even out-of-vocabulary (OOV) words. This feature is especially beneficial for inflected languages where specific word forms might not appear in the training set. Moreover, FastText's design enables efficient and expedited training compared to many alternative pre-trained word embedding models.

B. Seq2seq Model

Initially designed for machine translation, the seq2seq model facilitates the transformation of input sequences from one domain (e.g., Malay sentences) into corresponding output sequences in another domain (e.g., their English translations). Given its effectiveness, it has garnered the attention of researchers and has been adapted for tasks such as caption generation, text summarization, and chatbot

interactions. In the context of chatbots, a natural language query generates a corresponding natural language response. A central approach to the seq2seq implementation is the encoder-decoder framework. It essentially consists of an encoder, a context vector, and a decoder. The encoder captures and compresses the essence of the input into a fixed-dimensional hidden state. Subsequently, the decoder uses this state to produce the desired output. While the foundational architecture uses two RNNs, enhancements can be achieved using advanced RNN variants such as LSTM/GRU [1]. In the encoding phase, the input is converted into a vector representation leveraging an RNN-based encoder to encapsulate the essential context and details of the input sequence. Subsequently, a second RNN, functioning as a decoder, employs this vector to produce the desired output sequence. A fundamental representation of the seq2seq model during the training phase is depicted in Fig. 1.

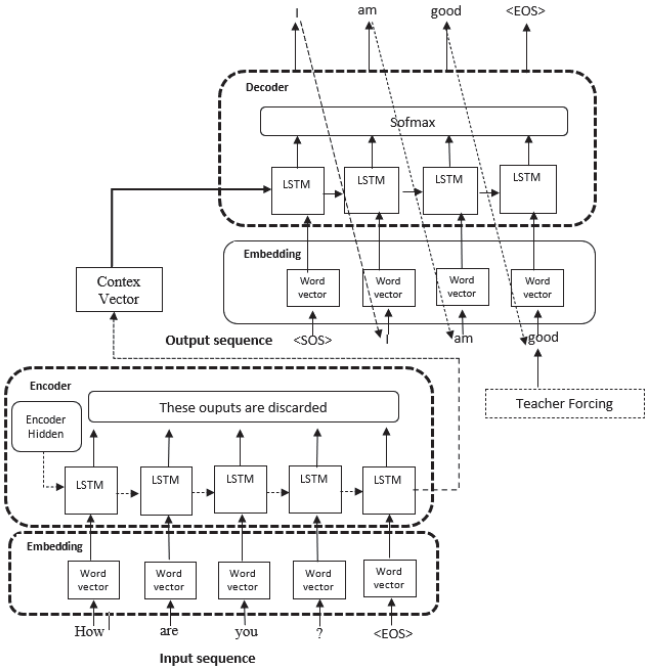

Fig. 1 Fundamental Architecture of Encoder-Decoder during Training, however, Teacher Forcing is ignored during Inference

From this figure, it can be seen that the process of mapping the words contained in each input statement or utterance, denoted by $x = \{x_1, x_2, x_3, ..., .x_n\}$, (where n is the length of the statement), into an embedded representation (φ^{x^n}). The result of this mapping is then passed to Recurrent Neural Network (RNN) models, which take the hidden state of the first encoder as input. The RNNs sequentially generate hidden representations and output vectors at each time step, taking a word and the previous state's hidden state as input and providing output and an updated hidden state until they reach the end of the input, which is indicated by a unique token. However, the encoder's outputs at each time step are not considered because they are consolidated in the context vector (C). This context vector contains information about all input elements, which facilitates accurate prediction of the response by the decoder. The calculation of the hidden states and the context vectors is shown in (1) and (2), respectively.

$$\bar{h}_m = f_1(\varphi^{x^n}, \bar{h}_{m-1}) \tag{1}$$

$$c = f_2(\{\bar{h}_0, \bar{h}_1, ...\bar{h}_M\}) \tag{2}$$

where $\bar{h}$ denotes the hidden state, c denotes the context vector formed from the hidden states of the encoder, and f_1, f_2 are nonlinear functions that may include LSTM as used in this study.

The context vector serves as the initial hidden state of the decoder and facilitates the transfer of information from the encoder. This study addresses the use of bidirectional encoders where both forward and reverse RNNs are implemented. The input sequence is processed in both directions and the resulting forward and backward hidden states are merged before being passed to the decoder.

During the decoding process, the first timestep, the <SOS> token is presented along with the context vector as input to the RNNs. The <SOS> token serves as a starting point for decoding and facilitates the generation of the first word of the chatbot response by analyzing the context vector. The initial output of the RNNs during decoding will likely be *"I"*. For the next timestep, *"I"* is used as input along with the hidden state from the previous timestep. This process is repeated and generates the output *"am"*. The generation of the output continues until the RNNs encounter a unique token that is identified as <EOS>. Teacher forcing is one of the techniques that can also be used to improve the performance of the model. Teacher forcing is implemented while training the network, using the actual output of the training data as input for the next time step, as implemented in this study. Given the context vector as c and all previously predicted outputs as $\{y_1, y_2, y_3, ...y_{t-1}\}$, the decoder was trained to predict the subsequent token y_t which refers to the maximum likelihood estimation of y_t. The prediction is given by y, the output vector, and c, the context vector, hence, the $p(y)$ is calculated as in (3):

$$p(y) = \prod_{t=1}^{T} p(y_t | y_1, y_2, y_3, ...y_{t-1}, x_t) \tag{3}$$

and produces a token with a conditional probability for each timestep t as in (4):

$$p(y_t | y_1, y_2, y_3, ...y_{t-1}, x_t) = \bar{g}(y_{t-1}, s_t, c_t) \tag{4}$$

where $\bar{g}(.)$ is a softmax function and s_t is the hidden state of the decoder at the timestep t which can be computed as in (5) as follows:

$$s_t = f(s_{t-1}, y_{t-1}, c_t) \tag{5}$$

C. Neural Attention Mechanism

The vanilla seq2seq model is based on the RNNs mentioned earlier, which use the temporal dynamics of the input data to produce sequential output data. However, the relevance of the output generated at a given timestep and the input sequence utilized to derive this result remains uncertain. Furthermore, only the final states (context vector) serve to

initiate the decoder, ignoring all intermediate states. While this approach works well for small or medium-length sequences, capturing extensive sequences within a single vector becomes increasingly difficult as the sequence length increases. RNNs analyze tokens sequentially while preserving a state vector that captures the data of the processed tokens. This sequential data encoding can lead to uncontrolled information spread. Yet, for extended sequences, the model may miss details from earlier tokens at the final state because the gradient vanishing issues, reducing its efficiency. While LSTMs mitigate the challenges of vanishing and exploding gradients, they do not remove them completely. In addition, RNN models may not be able to handle increasingly complex features and provide reliable results.

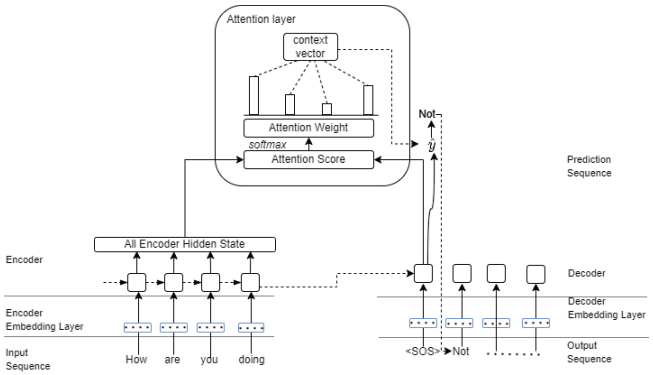

Fig. 2 Attention-based Encoder-Decoder

The introduction of attention processes in [7] and [8] has alleviated the problem of the vanilla seq2seq model. An attention process facilitates a model to immediately determine the state of an earlier part of the sentence and make conclusions from it. All previous states are accessible to the attention layer. It can provide more accurate information about distant relevant tokens by weighing them according to a learned measure of relevance to the current token, as depicted in Fig. 2. At each decoding phase, it determines which elements of the source are most relevant. Rather than compressing the entire source into a single vector, the encoder outputs token representations for all source data. Moreover, the fundamental idea of attention is to use all of the encoder's intermediate states, rather than discarding them, to create the context vectors that the decoder uses to generate the output sequence by applying attention weighting. The relevant part of the input is determined by calculating the attention weighting, which then determines the output.

During the attention mechanism, each word in the input sequence is assessed for its relevance to every output cell. For every y_t in the output y, it is influenced by the context vector c_t (source context for decoder step t) are used in an information filter for all encoder's hidden states $h = \{h_1, h_2, h_3, ..., h_{m_x}\}$ of the encoder, which can be calculated as in (7)-9):

$$c_t = \sum_{i=1}^{m_x} \alpha_{t_i} h_i \tag{6}$$

Where α_{t_i} is calculated by

$$\alpha_{t_i} = \frac{exp(e_{ti})}{\sum_{j=i}^{m_x} exp(e_{tj})} \tag{7}$$

where $e_{ti} = align(s_{t-1}, h_i)$ refers to the variants of the score function that considers

$$M1\ Dot\ e_{ti} = s_t^T \bar{h}_i$$

$$M2\ Multiplicative\ e_{ti} = s_t^T W_a \bar{h}_i$$

$$M3\ Additive\ e_{ti} = v_a^T tanh(W_a s_t + U_a \bar{h}_i)$$

$$M4\ *Multiplicative\ e_{ti} = tanh(s_t^T W_a \bar{h}_i)$$

Where α_{t_i} represents the attention weights determined by the model while, W_a,U_a and V_a, implies additional weight parameters for the model to learn. The *align* is an alignment model for evaluating the relationship between the input of position i and the output of position t.

Typically, M1, M2, and M3 are the existing scores in the seq2seq model, as highlighted in several studies [7], [8], [13]. M2, or the multiplicative scoring function, computes attention weights through a linear interaction between input and output features. While the linear nature of M2 is effective in many scenarios, it has its limitations, particularly in the context of complex natural language processing tasks. This linear approach may not fully capture the complicated, non-linear relationships present in language data that are often important in chatbot applications. In contrast to this study, the *tanh* activation is introduced on the existing multiplicative scoring function. The *tanh* function can squash the output lying in the range of -1 and 1. This bounded output can be advantageous for the model's performance, as it ensures a normalized and consistent scale for attention weights, which is crucial for stable and meaningful comparisons between different parts of the input sequences. It also solves some of the underlying limitations of M2 and is necessary for stable and interpretable attention distributions. This modification slightly attempts to increase the model's ability to understand and interpret complicated language patterns, which is crucial for producing a more effective chatbot.

IV. Methodological Approach

This section provides an overview of the current methodological approach to research the neural generative attention weights in the seq2seq model. Before this model is performed, several preprocessing steps are required to conduct the current experimental study. The first step begins with splitting the initial dataset into a training set and a test set. The whole dataset is split into 75% and 35% for the training and validation/test sets, respectively. The study uses Kaggle's 'Customer Support on Twitter (CST)' dataset, comprising 2,811,774 tweets and replies. This dataset contains 1,537,843 tweets (54.69%) from consumers and 1,273,931 (45.31%) from customer support agents. Of particular note, approximately 1.27 million tweets from consumers in this dataset contained responses from customer service representatives, providing a rich corpus of real-world

conversation data for analysis. The selected dataset, which is characterized by its realistic nature and the manageable size of the messages, is particularly well suited for the study of recurrent networks.

Preparing the dataset for the modeling process involves a series of pre-processing and feature extraction steps. These pre-processing steps include expanding abbreviations, removing emojis, emoticons, mentions, URLs, HTML tags, and special characters, correcting spelling errors normalizing the text to lowercase, and appending a special token for words that are not in the vocabulary. In addition, the data are restructured to integrate tokens and facilitate the processing of decoder input. During the exploratory data analysis (EDA), the max_length of the sequence was set to 39, which corresponds to the 95th percentile of the dataset distribution. The data are then organized to meet the requirements of the E2D model and the text is tokenized. The uniformity of the sequences is ensured by padding shorter sequences and truncating longer sequences to obtain a consistent format.

For feature extraction, a transfer learning approach is adopted, utilizing FastText pre-trained word embeddings as mentioned earlier to speed up training and increase model performance. This approach considers knowledge transfer between networks trained on different datasets. The result of this step is incorporated into the neural generative attention model depending on different scoring functions (attention weights): dot, multiplicative, additive, and extended multiplicative by adding act *tanh* into the function as discussed in the previous section, which is trained with a training set. The training of this model to predict the response matches the ground-truth answers. The training process can be represented as minimizing the loss function $L(\theta)$, where θ denotes the model parameters. The objective is to find the optimal θ that minimizes the difference between the predicted response and the ground truth, which can be defined as:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N}\mathscr{L}(y_i,\hat{y}_i)$$

where $L(\theta)$ is the average loss over the training set, N is the number of examples in the training set, y_i refers to the target label for i, $\hat{y}_i$ is the predicted response for i generated by the model, and $\mathscr{L}(y_i,\hat{y}_i)$ is the loss for i calculated using a loss function suitable for the problem at hand such as sparse-categorical cross-entropy loss for this case. The optimization process to minimize $L(\theta)$ is based on an AdamW optimizer [17]. This method iteratively updates the parameter θ based on the gradient of the loss function concerning θ [18]. These iterations continue until a stopping criterion is met. e.g., a predefined number of epochs. The final result is an optimized set of parameter θ that can be used to make predictions that are very close to ground truth. The lower the loss value the better during the training process. Finally, we prepare the validation/test dataset accordingly and make use of it to evaluate the models. The methodology followed in this work is depicted in Fig. 3.

This experiment was performed using TensorFlow [19] and Keras, a Python-based deep neural network package. A

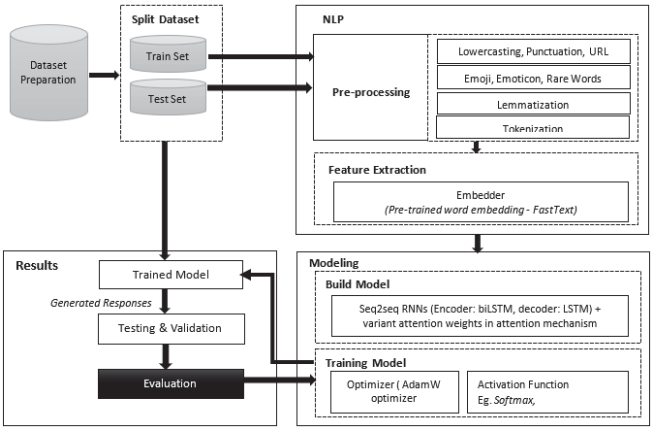

Fig. 3 Illustration of the Methodological Approach

TABLE I
HYPERPARAMETER SETTING

Parameter	Setting
Max Length Input	39
Embedding size	300
Batch Size	64
Hidden Unit	480
Learning rate	0.003
Clipvalue	0.5
Optimizer	AdamW
Word embedding	FastText
Encoder types	Bidirectional

Juypyter notebook hosted as open source under the name Google Colaboratory or Colab Pro+ was used for the models, with a high specification of memory, 89/50 GB RAM and GPU via subscription and tested with a batch size of 64. Moreover, the LSTM hidden size is tested with 480 units (the LSTM units that our memory space can hold). The four different types of score functions (dot, multiplicative, additive and *multiplicative (extended multiplicative)) were compared with a learning rate of 0.003 for the optimization [20]. The hyperparameter learning rate feeds into the optimization function. To mitigate the 'exploding gradient' issue, a gradient clipping value of 50.0 is incorporated. This approach ensures that the gradients do not expand exponentially, thus avoiding potential overflow or surpassing steep changes in the cost function. All weights and biases are initialized following the Xavier Uniform Glorot and Bengio distribution [21]. The study uses 300-dimensional pre-trained word embeddings for FastText. Initially, the experiment setting also uses an early stopping technique to prevent overfitting. However, there are limitations to training the models as our memory resources are not occupied by the early stopping technique for training. The hyperparameters and for training the models are listed in Table I.

V. RESULT AND DISCUSSION

In this section, the experimental results of the model for the aforementioned dataset. The experiment evaluated the performance of the different score functions in the neural attention mechanism with the pre-trained FastText embedding

TABLE II
COMPARISON OF DIFFERENT ATTENTION WEIGHTS (SCORE FUNCTIONS) IN TERMS OF LOSS AND BLEU SCORE

Training Phase			Inference Phase	
FastText				
Model	Loss	Val Loss	BLEU	
			Greedy Search	Beam Search, k=3
M1	1.072472	1.138756	0.436507	0.428405
M2	1.071038	1.137435	0.438132	0.430391
M3	1.073800	1.138623	0.438506	0.440251
M4	**1.070897**	**1.136484**	**0.438926**	**0.443000**

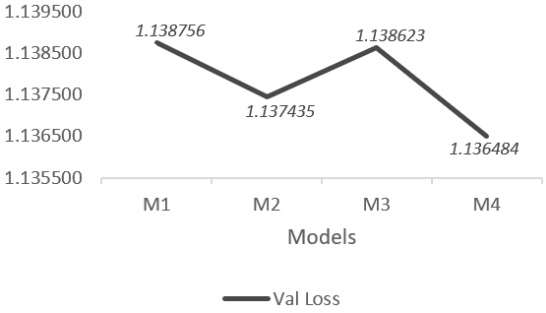

Fig. 4 Validation loss for different attention weights during the training phase

Fig. 5 BLEU score for different attention weights during the inference phase

as an input feature to a model. Table II and Figs. 4 and 5 show the performance results of the different scoring functions on the model using the sparse-categorical entropy loss during training and the BLEU scores metric in the inference phase. The result shows that the M4 model which incorporates a *tanh* activation function into a multiplicative attention scoring function, consistently outperforms the other models across all evaluated metrics. This model achieved the lowest validation loss of 0.136484 (see Fig. 4), indicating superior generalizability to unseen data, and also it recorded the highest BLEU scores under both greedy search (0.438926) and beam size with size k = 3 (0. 0.443000). These results suggest that the M4 model generates better response predictions, highlighting the potential benefits of introducing non-linearity into the scoring function in the attention process.

While the other models —M1-Dot, M2-Multiplicative, and M3-Additive- show comparable results, a clear incremental improvement in BLEU scores is observed as the analysis progresses from M1-Dot to the M4 model, as shown in Fig. 5. This progression suggests that increasing the complexity of the attention-scoring mechanism, e.g., by including non-linear activation functions, allows the model to learn more nuanced relationships between input and output sequences, thereby improving the quality of the generated responses. In addition, these results also show that the choice of scoring function is a key factor in the model's performance. Adding *tanh* activation to a multiplicative model (M4) has yielded the best results highlighting the role of the attention process in chatbot seq2seq tasks. The *tanh* function provides non-linearity in the model and allows the function to squash the output into the range between -1 and 1, allowing the model to focus more effectively on relevant information. When the attention process assigns weights to different parts of the input, the *tanh* function ensures that these weights are scaled appropriately. This potentially allows a clearer distinction between relevant and irrelevant parts of the input, facilitating more nuanced attention, which indirectly improves the performance of the chatbot model. Furthermore, it can be observed from the results that the improved model significantly outperformed the base methods, and this improvement is statistically significant at $p < 0.05$. This finding justifies the effective performance of our proposed model, underscoring the role of the attention process in chatbot seq2seq tasks.

VI. CONCLUSION

The study aimed to evaluate and compare different attention-scoring functions —Dot (M1), Multiplicative/General (M2), Additive (M3), and Extended Multiplicative with *tanh* activation (M4)— in neural generative seq2seq models, specifically for chatbot applications in a CST dataset. The results of the experiment show that the M4 model yields the most promising result for all evaluation metrics. This model provided the lowest validation loss (1.136484) and the highest BLEU scores (0.438926 for greedy search, 0.443000 for beam search with size, k = 3), demonstrating its effectiveness in making appropriate response predictions in a chatbot application. This result suggests that introducing a non-linear activation function into the attention mechanism enables the model to learn more complex, nuanced relationships between input and output sequences, thereby providing more appropriate response predictions in chatbot applications. There is a progressive improvement in performance with respect to the BLEU score as the analysis moves from a simple dot product (M1) to an extended multiplicative model. This highlights the significant potential of the choice of the scoring function to influence the improvement of the seq2seq model's capabilities in text chatbot datasets. Although this model was trained for 10 epochs, without an early stopping technique due to computational limitations, it represents a remarkable advance in understanding the impact of different attention weights of seq2seq models, especially in CST datasets. For future work, it is suggested to extend the training over more epochs for these models, investigating alternative activation functions within the attention weights, examining various

model optimization techniques, and testing these models across diverse datasets to contribute a more comprehensive understanding of their capabilities and limitation in different scenarios.

ACKNOWLEDGMENT

The authors would like to thank the Ministry of Higher Education Malaysia for partially funding this research under the Fundamental Research Grant Scheme (FRGS/1/2022/ICT06/UTM/01/1) with grant vote No. R.J130000.7851.5F568. Furthermore, appreciation is extended to Universiti Malaysia Sarawak (UNIMAS) and Universiti Teknologi Malaysia (UTM) for providing the necessary resources for this research work.

REFERENCES

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL* (A. Moschitti, B. Pang, and W. Daelemans, eds.), pp. 1724–1734, ACL, 2014.

[4] M. Hardalov, I. Koychev, and P. Nakov, "Towards automated customer support," in *Artificial Intelligence: Methodology, Systems, and Applications: 18th International Conference, AIMSA 2018, Varna, Bulgaria, September 12–14, 2018, Proceedings 18*, pp. 48–59, Springer, 2018.

[5] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, p. 3776–3783, AAAI Press, 2016.

[6] A. Xu, Z. Liu, Y. Guo, V. Sinha, and R. Akkiraju, "A new chatbot for customer service on social media," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, (New York, NY, USA), p. 3506–3510, Association for Computing Machinery, 2017.

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

[8] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (Stroudsburg, PA, USA), Association for Computational Linguistics, 2015.

[9] C. Xing, W. Wu, Y. Wu, J. Liu, Y. Huang, M. Zhou, and W.-Y. Ma, "Topic aware neural response generation.," in *AAAI* (S. P. Singh and S. Markovitch, eds.), pp. 3351–3357, AAAI Press, 2017.

[10] Z. Wang, Z. Wang, Y. Long, J. Wang, Z. Xu, and B. Wang, "Enhancing generative conversational service agents with dialog history and external knowledge," *Comput. Speech Lang.*, vol. 54, pp. 71–85, 2019.

[11] F. Kassawat, D. Chaudhuri, and J. Lehmann, "Incorporating joint embeddings into goal-oriented dialogues with multi-task learning," in *European Semantic Web Conference*, pp. 225–239, Springer, 2019.

[12] G.-P. Yang and H. Tang, "Supervised attention in sequence-to-sequence models for speech recognition," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7222–7226, 2022.

[13] S. M. Suhaili, N. Salim, and M. N. Jambli, "A comparative analysis of generative neural attention-based service chatbot," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 8, 2022.

[14] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," 2016. cite arxiv:1607.04606Comment: Accepted to TACL. The two first authors contributed equally.

[15] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext.zip: Compressing text classification models.," *CoRR*, vol. abs/1612.03651, 2016.

[16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[17] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[18] C. C. Aggarwal, *Neural networks and deep learning: A textbook*. Cham, Switzerland: Springer International Publishing, 2 ed., 2023.

[19] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning.," in *OSDI*, vol. 16, pp. 265–283, 2016.

[20] L. Mou and Z. Jin, *Tree-Based Convolutional Neural Networks: Principles and Applications*. Springer Publishing Company, Incorporated, 1st ed., 2018.

[21] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks.," in *AISTATS* (Y. W. Teh and D. M. Titterington, eds.), vol. 9 of *JMLR Proceedings*, pp. 249–256, JMLR.org, 2010.

Sinarwati Mohamad Suhaili received her Bachelor's Degree (Hons) in Information Technology (Computational Science) and Master's Degree in Advanced Information Technology (Networking) from Universiti Malaysia Sarawak (Unimas). Currently a lecturer at Unimas's Centre for Pre-University Studies, she brings over a decade of teaching experience in computing. She is also actively pursuing her Ph.D. in Computing, with a focus on Artificial Intelligence. Her research interest spans various domains including Artificial Intelligence, Machine Learning, Deep Learning, Generative AI, Natural Language Processing, and the application of ICT in assistive technology.

Prof. Dr. Naomie Salim received the B.Sc. degree in computer science from the Universiti Teknologi Malaysia, the M.Sc. degree in computer science from the University of Western Michigan, and the Ph.D. degree in information studies from the University of Sheffield. She is currently a Professor with the Faculty of Engineering, School of Computing, Universiti Teknologi Malaysia. She has authored over 100 journals and conference papers since the inception of her research career. Her main research interests include text mining, machine learning, information retrieval, cheminformatics, and natural language processing.

Assoc. Prof. Dr. Mohamad Nazim Jambli received his PhD (Computing Science) from Newcastle University (UK), his Master of ICT from Griffith University, Queensland, Australia, and Bachelor Degree (Hons) in Information Technology (Computer System Technology) from Universiti Malaysia Sarawak. His ICT related research interest is ranging from Mobile Ad-hoc Network (MANET), Mobile Wireless Sensor Network (MWSN), Vehicular Ad-hoc Sensor Network (VASNET), Internet of Thing (IoT), Blockchain Technology, Recommendation Systems, System Performance and Energy-efficient Routing Protocols. He is currently the principle investigator for several research and community project grants provided by the Ministry of Education (MoE), Sarawak Multimedia Authority (SMA) and Sarawak State Agencies in Malaysia.