Neural Text Style Transfer with Custom Language Styles for Personalized Communication Systems

Conference Paper · December 2022

DOI: 10.1109/ICKECS56523.2022.10059789

3 authors, including:

Hardik Srivastava

5 PUBLICATIONS 3 CITATIONS

SEE PROFILE

Shantha Kumari K.

SRM Institute of Science and Technology

14 PUBLICATIONS 11 CITATIONS

SEE PROFILE

2022 International Conference on Knowledge Engineering and Communication Systems (ICKES)

Neural Text Style Transfer with Custom Language Styles for Personalized Communication Systems

Hardik Srivastava
Data Science and Business Systems
SRM Institute of Science and Technology
Kattankulathur,India
hs9644@srmist.edu.in

Sneha Sunil
Networking and Communications
SRM Institute of Science and Technology
Kattankulathur,India
ss4795@srmist.edu.in

K.Shantha Kumari
Data Science and Business Systems
SRM Institute of Science and Technology
Kattankulathur,India
shanthak@srmist.edu.in

***Abstract*— Style can be thought of as a way where semantics are represented intuitively. Every person has a unique writing style, which can be expressed through certain stylistic features present in a sentence. These features can include the usage of certain common words, contractions, metaphors, slang, and some syntactic structures that when combined determine the language style of a person. Currently, there is no way of applying stylistic variations on text in a dialogue system to adapt to a user or an audience. In this paper, we propose a network architecture to transform machine-like robotic conversations into human-like personalized ones by applying custom text styles to the responses thereby giving them a user-specific tonality, leveraging Language Modeling, Generation, and Style Transfer techniques in a Representation Learning fashion. When executing the process of transferring the text style, our method keeps the original sentence's content intact in the transferred sentence dissociated from its style. This system implementation can be transformed into a software application that can help people use the language styles of their acquaintances and make the AI talk like them.**

Keywords*— *Text Style Transfer, Language Modeling, Text Generation, Representation Learning, Conversational AI

I. INTRODUCTION

The importance of communication in today's world cannot be overstated. Chatbots are Expert Systems that serve as an application for computer communication. A conversational Chatbot is an intelligent program but isn't the same as a human agent, so it doesn't always understand the context of the dialogue and its selection of answers may be limited based on its knowledge base [1]. This makes it sound “robot-like”. Chatbots are employed either for solving tasks like Question Answering or used as Dialogue Systems and/or AI Assistants. These Chatbots are trained in such a way that whenever a question is asked, the bot will output a standard machine-like response via pattern matching, but merely verbalize the semantics in an unstyled manner. Existing dialogue systems and AI assistants lack the ability to generate responses that carry a specific persona with them which would make them sound personalized and deliver a consistent response, which would be more human-like.

Here is where the importance of Neural Text Style Transfer comes into play. Text Style Transfer (TST) has been a trending topic of research in the industry attracting thousands of academicians and computer science researchers. TST refers to changing a sentence's style by rewriting the original sentence after applying a new language style to it while retaining its semantic content. With the development of deep learning over the previous decade, a variety of neural approaches for TST have been developed [2] [3]. Text Style Transfer has many applications like building Personalized Intelligent Bots, Text Simplification Models, and even Writing Assistants. A major challenge for style transfer is the absence of large-scale parallel data. Several methods have also been put out for non-parallel data where the latent representations of the input are separated from the source style at first and later recombined with the target style to generate the target response. Moreover, conventional Chatbots cannot be employed for personalized communication systems because they cannot lack knowledge about the user’s language style. Henceforth, we aim to create a Neural Conversational AI that responds with phrases that have a particular language style (tonality) of a person applied to it by performing certain stylistic transformations on the text.

The structure of the paper is as follows. Following the introduction, Section II describes the various Text Style Transfer methods that have been studied in the previous papers including both vision and text data, their shortcomings, and the novel method that we have proposed. Section III provides a detailed description of the Text Style Transfer methods that are used currently and gives a brief on representation learning methods, the datasets used in our entire system implementation, formulation of the problem, the network architecture, the various deep learning methodologies that were employed and how token embeddings were trained for our language model. Later in this section, we talk about the various experiments that were carried out. Section IV talks about our results and discussions. The potential research directions to advance the field are covered in Section V.

II. REVIEW OF LITERATURE

A. Image Style Transfer

Style Transfer tasks were originally introduced for vision data only. Even now, most of the deep learning approaches focus on transferring content representations from image features from a source image to a target image. Gatys et al. [4] employed CNNs to create a model that could learn both the image content as well as the style separately and later create a new image from the combination of the two. They show that although CNNs were able to learn the deep image representations well and apply the styles to the input image, there was still a problem with this methodology. The model ignored the possibility that object categories would vary in both the content and style images. Park et al. [5] proposed a hierarchical architecture that solved the problem of semantic matching using semantic context-aware style transfer. They suggest using context correlations of different object categories. Their network architecture is a combination of two networks where one focuses on performing the style transfer task while storing the feature representations for the corresponding semantic regions. The other network derives the stylized image, given the style transfer image, the image

978-1-6654-5637-1/22/$31.00 ©2022 IEEE

content, and its respective style. Generative Adversarial Networks (GANs) have been proposed by Liu and Tuzel [6] to learn multi-domain data by using constraints like weight sharing. Using a Deep CNN as an encoder for text would not help us capture the text style or tonality of the sequence. We propose to use a Recurrent Neural Network in combination with a Pointer Network to achieve this task.

B. *Text Style Transfer*

Style Adaptations in text is a very recent research topic with some prior work. Sojasingarayar [7] proposed to use LSTMs with an Attention mechanism in a Reinforcement Learning fashion to train a Conversational agent with manually annotated data. They aim to produce a context-aware self-learning bot that can learn from its past interactions. Although the model was able to retain context, it failed to imitate human interaction and generated very short and computerized responses. The algorithm lacked adapting to any generic language style and its generalizing capability was completely dependent on the training data. Bahdanau et al. [8] enhanced existing Seq2Seq models by proposing attention mechanisms which are state-of-the-art approaches for Neural Machine Translation tasks. According to Kuang [9], using a vanilla Seq2Seq model for building a Chatbot lacks the capabilities to produce innovative and intelligent responses. Even today, conversational agents suffer from problems like insufficient consumption of multi-modal information, the ability to express emotions, and lack of ensemble learning mechanisms. He proposed a two-way GRU + Attention model as his network architecture to enable his Chatbot to generate more mature responses giving it a real chat effect. However, there were flaws as observed by him like the repetition of common phrases and partial adaptation towards a particular language style of the responses. Roy et al. [10] leveraged multiple models for language transformation tasks based on geolocation. However, the approach is limited to minimal changes in the output with minor additions of adverbs and adjectives only. Our method handles generic transformations with vast data domains including both the addition and removal of words based on context. Pointer Networks, when used in a Seq2Seq model allow the usage of words in input directly as output. Therefore they are quite often employed for tasks like Q/A Answering and Text Summarization. Pointer Networks lack text generation capabilities. When combined with Recurrent Neural Networks as in our work, they show quite good performance in language modeling tasks.

There has been a significant amount of research in the domain of Text Generation and Language Modeling, specifically, Chatbots where numerous neural engines have been built by researchers to solve tasks like Automated Question Answering [11], Conversational AI, Dialogue Systems, Personal AI Assistants, IVR, etc. Chatbots are automated neural machines which always interact in a computerized manner. They lack the personalized touch. Our work, distinct from all these, attempts to propose a model that provides a solution for this research gap by developing a user-specific Language Stylized Conversational AI.

C. *Techniques for Style Transfer*

The deep learning methodologies used for the text style transfer study are discussed in this section. The encoder-decoder architecture has been employed by a vast majority of models, despite the fact that using GANs with Adversarial Learning is recommended in some studies. The encoder is responsible for encoding the input sequence into a latent representation while the decoder builds the output sentence based on the latent representation, and the classifier selects the style label for the output sentence. The classifier may or may not be included in style transfer models. If the style transfer model is built on GAN, the encoder is referred to as a generator and the classifier component is referred to as a discriminator [12]. For the purpose of encoding input sentences and/or producing output sentences, style embeddings have been introduced. The encoder, the decoder, or both could receive input via style embedding.

1) *Rule Based Models*

Rule-based models are the fastest and easiest ways to create a bot. But building a bot that can respond to intricate questions and queries is exceedingly challenging. As a result of the inadequate pattern matching, Artificial Intelligence Markup Language based bots struggle when they face a sentence devoid of any recognized patterns. Additionally, writing the rules by hand is laborious and time-consuming. Another crucial point is that standard text bots always appear to be outputting a pre-fed static response. The bot lacks a distinctive tone or style of its own. What if, though, we could create a Chatbot that did more than simply learn from prior conversations and generate responses accordingly but also adapt to the user's tonality and style of writing? This is where our novel approach to Neural Text Style Transfer comes in.

2) *Representation Learning*

The addition of a learning mechanism in neural network-based systems is the key concept that sets them apart from rule-based approaches. In a representation learning technique, a framework called an encoder decoder is used. A deep neural network called an encoder-decoder, also called a Seq2Seq network, is used to generate text [13]. The encoder develops a latent fixed-length vector representation of the input sentence over time. The decoder learns to create an output sentence by decoding the fixed-length representation of the input sentence.

III. METHODOLOGY

A. *Dataset*

We have used the ConvAI2 dataset [14] for training our Conversation AI algorithm, which is a Seq2Seq model, to generate textual responses upon user input. The number of data points considered for training, validation, and testing is shown in Table 1.

TABLE 1: CONVAI2 DATASET

	#examples	**#dialogues**	**#personas**
Training set	131,438	17,878	1,155
Validation set	7,801	1,000	100
Test set	6,634	1,105	100

For applying custom language styles to these textual responses, we are using a parallel dataset that contains pairs of line-by-line phrases $\{X_S, X_T\}$ of common conversations where the input data X_s contains sentences that have their machine-like language style, while the target domain X_T consists of data with the personalized language style, i.e. data with the language style and tonality of the person we want our model to adapt to. This dataset was compiled by Srivastava H. and Sunil S. Table 2 shows a question that would serve as input to our Conversational AI. COMPUTERIZED represents the response generated by the bot without the language style while CUSTOM-STYLIZED represents the same response after the language style was applied to it.

TABLE 2: EXAMPLES FROM OUR DATASET SHOWING PAIRS OF COMPUTERIZED, STANDARD AND CUSTOM-STYLED RESPONSES

S. No	Type	Text
1	**QUESTION**	**How are you?**
	STANDARD	I'm fine. Thanks for asking.
	COMPUTERIZED	I am fine.
	CUSTOM STYLIZED	Hey. I am fine yo. How's you?
2	**QUESTION**	**What did you have for dinner?**
	STANDARD	I love feasting on information, facts and trivia.
	COMPUTERIZED	I had pizza.
	CUSTOM STYLIZED	There was pizza today. You know how much I love it. Haha. What about you?
3	**QUESTION**	**How was your day?**
	STANDARD	Thanks for asking. I spent today thinking about writing a book which might be an ode to Assistants.
	COMPUTERIZED	It was a good day at work.
	CUSTOM STYLIZED	It was a fun day at work. You know what the boss threw a party. We had a blast.
4	**QUESTION**	**I wanted to talk to you. Are you free?**
	STANDARD	Of course I am.
	COMPUTERIZED	Yes
	CUSTOM STYLIZED	Yeah I'm free. Is everything fine? You can call me now.

B. *Problem Formulation*

Now we formally propose our formulation of the problem. Our problem is split into two parts. The first part deals with generating text responses using a Seq2Seq Model using LSTMs [13] [15]. This model is trained using the ConvAI2 dataset, which is a collection of 2000+ text dialogues employed for NLG tasks. Upon each interaction with the ConvAI model, there is a text response as output. Let's consider this textual output as Y_T. Now for the second part, suppose there are two data domains X_S and X_T, as discussed in Section III. Y_T is compared with X_S for semantic relatedness using a similarity measure. We have defined sentence similarity using Soft Cosine [16] over standard Cosine Similarity because the latter checks for semantic similarity rather than synonymity. The most similar $X^{(i)}_S$ is now used for further computation to map the corresponding text style of $X^{(i)}_T$ onto Y_T. Something to take note of is that each pair of source and target ($X^{(i)}_S$, $X^{(i)}_T$) describes the same sentence which is semantically the same but different in their language styles. Our goal is to create a model that can learn from these parallel simultaneous training data in a way that, given a test sequence $x \in X_S$, that hasn't yet been encountered, we can transfer it into its counterpart $\tilde{x} \in X_T$, where $\tilde{x}$ should retain the semantics of x but with the language style in X_T.

C. *System Implementation*

The architecture diagram in Figure 1. shows our system implementation. We have employed the Seq2Seq model, which is an Encoder-Decoder network with LSTM units, for dialogue generation tasks. The model was leveraged from the TensorFlow Seq2Seq Contribution API which comes with built-in fully customizable state-of-art features like Attention mechanisms, Beam Search capabilities, and a BidirectionalRNN module to fasten and improve the generation process. The previous hidden states of the decoder are used to calculate the attention weights, encoder representations, and sentinel vector. After a text response has been generated by the algorithm (COMPUTERIZED), this response serves as the input to another BidirectionalRNN network where we apply a customized language text style upon it to transform its tonality from a computerized one to a personalized one using a Pointer Network Component [17], which is explained in sub-section *D*. The predicted probability that determines the output response comes from the combination of both the Pointer Network component and the decoder of RNN. As shown in Table 2, for the question "How was your day?" the responses by both the conventional AI assistant (STANDARD) and our Conversational AI (COMPUTERIZED) without any text style sound quite machine-like. However, when we apply the language style on (COMPUTERIZED), we get a response that has a user-specific tonality to it and sounds more human-like (CUSTOM-STYLIZED).

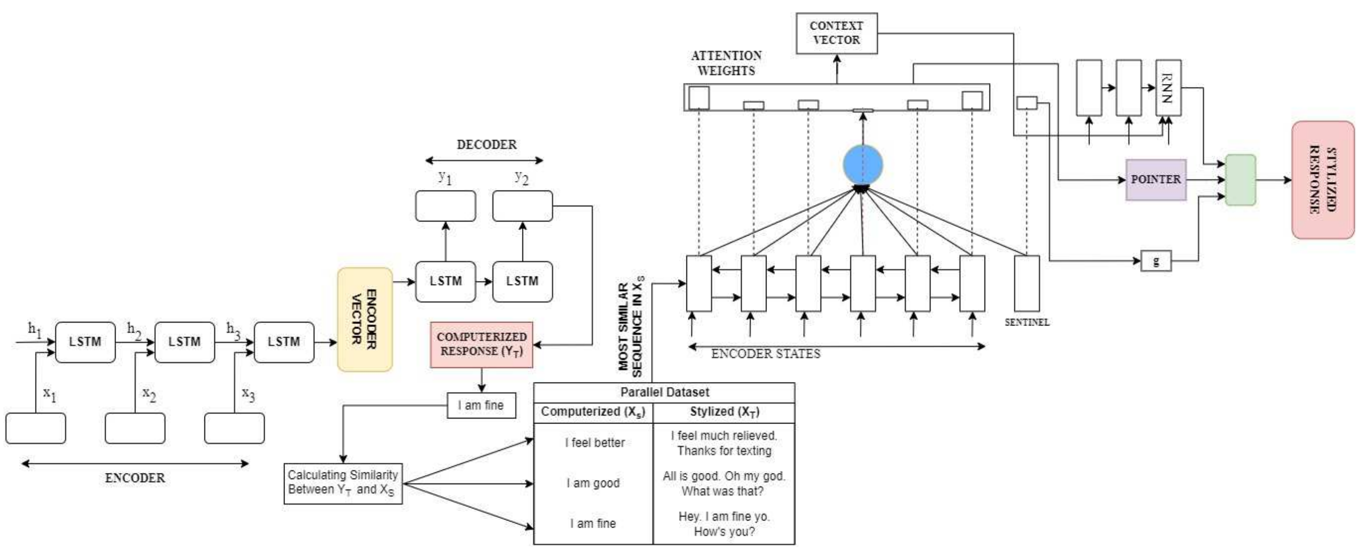

Fig 1. Representation of our overall architecture

1) Long Short Term Memory (LSTM)

Long Short Term Memory networks are a type of Recurrent Neural Networks that are capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber [15]. LSTMs were created to solve issues arising because of long-term reliance. They don't learn conventionally; rather, they remember information for a longer period. Each and every RNN resembles a collection of repetitive neural network modules. We use LSTM units in the Seq2Seq model while performing the NLU task to generate the computerized responses. When used with RNNs, they enhance the modeling of series data, here the sequence of words, and help in preserving the context of the dialogue on the decoder side.

2) Attention Mechanism

One of the drawbacks of the Seq2Seq model is that the full content of the input sequence must be encoded into the context vector, which has a predetermined length. We start losing a lot of information as the duration of the sequence increases. This explains why decoding lengthy sequences are difficult with the basic Seq2Seq paradigm. The decoder can discretely scan the input sequence during decoding, thanks to the attention approach by Bahdanau et al. [8]. This relieves the encoder of the burden of encoding all pertinent data from the input. Instead of using a fixed context (last hidden state of the encoder) for each time step in the decoder, a unique context vector c_i is used to generate the word y_i [18]. This vector is essentially the weighted sum of the encoder's hidden states. Different elements of the output sequence are produced using data from various segments of the input sequence. To put it another way, each word in the output sequence is positioned in relation to a particular aspect of the input sequence. We can gauge how closely the output at location i matches the inputs at or around position j using the alignment model based on which, the input contexts' weighted sum is calculated to generate each word in the output sequence.

3) Encoder Decoder Architecture

A Bidirectional LSTM is used to encode the computerized style input sequence. Our decoder model combines elements from Pointer Network and RNN modules. It is not required that the two distinct modules share the attention weights over encoder states. The pointer model is tasked with calculating the probability distribution from the input words and the decoder network in the RNN calculates the likelihood of occurrence of the next word throughout the lexicon. The weights used in the weighted sum of the two probabilities are determined by the encoder outputs and the hidden state of the previous decoder.

Let x, y be an input-output sequence in our parallel dataset. Both x and y represent sequences of tokens. $x = x_1x_2...x_{T_{enc}}$, where the length of input is denoted by T_{enc}. For the output, a similar relation exists where $y = y_1y_2...y_{T_{dec}}$. Every item from x and y, i.e. x_i, y_j represents a token.

Let $\overrightarrow{LSTM_{enc}}$ and $\overleftarrow{LSTM_{enc}}$ represent the forward and reverse encoders. $h_t^{\overrightarrow{enc}}$ represents the sum of both forward and backward encoder states at step t. Following are the various equations, (1), (2), (3) and (4), that describe our model:

$$h_0^{\overrightarrow{enc}} = \vec{0},\ h_{|x|}^{\overleftarrow{enc}} = \vec{0} \quad (1)$$

$$h_t^{\overrightarrow{enc}} = \overrightarrow{LSTM_{enc}}\left(h_{t-1}^{enc}, E_{enc}(x_t)\right) \quad (2)$$

$$h_t^{\overleftarrow{enc}} = \overleftarrow{LSTM_{enc}}\left(h_{t+1}^{enc}, E_{enc}(x_t)\right) \quad (3)$$

$$h_t^{enc} = h_t^{\overrightarrow{enc}} + h_t^{\overleftarrow{enc}} \quad (4)$$

Since we have a small data domain, we preferred to use addition over concatenation to combine the encoder states of both $\overrightarrow{LSTM_{enc}}$ and $\overleftarrow{LSTM_{enc}}$.

4) Pointer Model

A pair of sentences written in a computerized, machine-like, and customized manner have a lot of words in common, as was already mentioned. Furthermore, a Seq2Seq model may not be able to predict a large number of proper nouns. Therefore we use Pointer Networks [17] for this task. These networks enable direct copying of tokens from input. Additionally, the module's output probability distribution, which provides location-based attention, can be represented as follows in equation (5):

$$P_t^{PTR}(w) = \sum_{x_j=w}(\beta_j) \quad (5)$$

5) Word Embeddings

An M-dimensional embedding vector serves as a representation for each vocabulary item. Let vocabulary V represent the combination of User-Specific Language Stylized and Computerized Machine-like vocabularies, i.e. $V = V_{computerized} \cup V_{user\text{-}language\text{-}style}$. The embedding matrices utilized by the encoder and decoder, respectively, are represented by E_{enc} and E_{dec} (E_{enc}, $E_{dec} \in R^{|V| \times M}$). Because many of the tokens are shared by two vocabularies, we take into account the union of both the input embedding and output embedding vocabularies [19]. Additionally, in the best-performing configuration, we share embeddings between the encoder and decoder models. A token x's encoder side embeddings are represented by $E_{enc}(x)$.

The number of parameters is substantially increased by learning token embeddings from scratch along with the model. We think about pre-training the token embeddings on all training sentences as a potential solution to reduce this. In order to improve embedding learning, we additionally test by adding extra data from PTB (Marcus et al.) [20]. Moreover, we can leverage a dictionary mapping of the most common slang and phrases used by the person whose language style is the target. This aids in model understanding and more accurate application of the language style on the computerized responses. By using the most frequent phrases, the algorithm will have a higher probability of capturing the personalized tonality of the person and making the end user feel as if he is interacting with the person in reality.

D. Experiments

1) Pre-Processing

We utilized NLTK's (Natural Language Toolkit) PUNKT tokenizer for tokenizing all phrases in our dataset and WordNet Lemmatizer [21] to convert all tokens in a sentence to their root form. Lemmatization was preferred over Stemming as we wanted the root words to retain their context.

2) Dictionary

Since we aim to develop a Conversational AI that is able to adapt to any user's language style in text to provide a personalized chat experience to an end user, we use a dictionary of the most common words or slang used in a dialogue by that person. The idea to use a dictionary mapping of frequent words was introduced by Xu et al. [22]. Building and leveraging a dictionary like this for Text Generation and Style Transfer tasks greatly helps in increasing our training vocabulary and the knowledge domain of our model. Another acceptable baseline is to execute word-by-word replacements directly using this dictionary. This baseline actually performs worse than duplicating the input on the target side. This could be a result of its aggressive replacement that disregards word context.

IV. Results and Discussion

In this paper, we discussed adapting a user's language style and applying it to computerized responses generated by a conversational AI to make the entire dialogue flow very personalized and user-friendly. Instead of focusing on new text generation methods for dialogue systems, our work focuses more on how the existing dialogue systems can be utilized in a manner such that they can easily adapt to a person's tonality. Table 3 shows an overview of the various models and approaches that have been implemented for Style Transfer and Language Generation tasks.

TABLE 3: Comparative analysis between various models and their network architectures employed for Style Transfer in Language Modeling tasks and our proposed methodology.

Approach	Year	Architecture	Language Modeling Task	Dataset
BST [23]	2018	MT Encoder, Multiple BiLSTM decoders, CNN classifier	Sentiment Style Transfer Personal Style Transfer	Yelp, Gender, Political slant
DAM [25]	2018	3-Layerd Transformer network with Cross Attention	Multi-Turn Response Selection for Chatbots	Ubuntu Corpus V1, Douban Conversation Corpus
StyleTransformer [24]	2019	Transformer encoder, Transformer decoder, Transformer discriminator	Sentiment Style Transfer	Yelp
HybridST [26]	2019	Transformer encoder, Transformer decoder, CNN classifier	Formality Style Transfer	GYAFC
Seq2Seq+GRU [9]	2022	Two-way GRU Attention, LSTMs	Conversational AI with Mutual Information inclusion	Manually curated dataset
Proposed approach	2022	Seq2Seq model, LSTMs, Attention, BidirectionalRNN, Pointer Network	Conversational AI with User-Specific Text Style (Human-Like)	ConvAI2, Manually curated Parallel Dataset

BST [23] is a style classifier model that employs backtranslation-based methods to rephrase sentences using an English to French Neural Machine Translation network architecture. Both BST and StyleTransformer [24] are used for Sentiment Style Transfer tasks and they generate stylized responses with the help of style-specific decoders.

Deep Attention Matching Network (DAM) [25] solved for the optimization of the response selection process by a Conversational AI from a group of candidate responses. Parallel datasets, first introduced in HybridST [26], were used to create a paired collection of raw and stylized responses that could be used to train and develop a stylized AI. Some of the approaches that we drew comparisons with either introduced new architectures for style transfer tasks or talked about optimizing existing methodologies for language generation processes in conversational AI. We instead propose to develop a network architecture that can adapt to the language style of a person and apply it to the computerized responses of a Chatbot thereby making the dialogue flow very human-like. Kuang [9] also proposed a similar network to develop an intelligent Chatbot based on deep learning but it lacked human-like response generation capabilities and suffered problems like the repetition of short-length and high-frequency data and loss of context in longer conversations.

Using Seq2Seq models with LSTM units helped us to preserve the context of longer sentences and solve the problem of exploding and vanishing gradients. As introduced by Xu et al. [22], using a dictionary mapping of frequently used phrases and common syntactic structures by a person helped in solving the sparse data problem. The BidirectionalRNN network and the Pointer Network form the second half of our entire system implementation which apply the style embeddings on the computerized responses.

We saw in Table 2 that our network provided custom-stylized responses that were very human-like, in lieu of generating machine-like responses that were seen in the case of the conventional AI assistant (STANDARD type), as represented in Table 2. The vocabulary generation and expansion pipeline can be optimized to some extent because although the word embedding approach is good, techniques like hierarchical clustering [27] tend to perform better in neural language models. We also propose to use electronic conversations of people as paired dialogue data as it will not only help in increasing the data domain of our model but also help in learning and adapting the language style of the user in an even better fashion.

V. Conclusion

In this paper, we proposed developing a Personalized User-Specific Conversational AI system using a combined architecture of the Seq2Seq model for Text Generation tasks and an LSTM and Pointer Network Component network for applying custom language styles on the responses generated by the generative model. This Neural Style Transfer on text using custom language styles will transform the machine-like computerized dialogue flow that often occurs with an electronic bot or an IVR engine to a completely user-specific and personalized experience. Our experiments demonstrate the value of applying dictionary constraints and input-copying mechanisms for issues where the source and target sides are non-identical, but shared. A software program based on this research could enable users to initiate a custom stylized dialogue with an AI which will significantly help them converse with their acquaintance's linguistic patterns. In the future, we aim to optimize the data fetching and data expansion pipeline for our model. We plan to leverage day-to-day electronic conversations of the users on social media and parse them accordingly to make them serve as the data domain of our algorithm. By doing this, our challenge of sparse data can be solved. Additionally, the AI can generalize better in learning different types of conversation tones a user has with people. In order to increase personalization and user customization, a profile-based approach can be incorporated so that one user can load multiple language styles of different people on the bot. As and when the user wants to interact with the bot in different people's text styles', the software should accordingly switch the context and the tonality of the conversation both quickly and accurately.

References

[1] Zhao, Y., Bi, W., Cai, D., Liu, X., Tu, K., & Shi, S. (2018). Language Style Transfer from Sentences with Arbitrary Unknown Styles. arXiv. https://doi.org/10.48550/arXiv.1808.04071

[2] Jin, D., Jin, Z., Hu, Z., Vechtomova, O., & Mihalcea, R. (2020). Deep Learning for Text Style Transfer: A Survey. *arXiv*. https://doi.org/10.48550/arXiv.2011.00416

[3] M. Toshevska and S. Gievska, "A Review of Text Style Transfer Using Learning," in *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 5, pp. 669-684, Oct. 2022, doi: 10.1109/TAI.2021.3115992

[4] L. A. Gatys, A. S. Ecker and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2414-2423, doi: 10.1109/CVPR.2016.265

[5] Park, Joo & Park, Song & Shim, Hyunjung. (2019). Semantic-Aware Neural Style Transfer. Image and Vision Computing. 87. 10.1016/j.imavis.2019.04.001.

[6] Liu, M., & Tuzel, O. (2016). Coupled Generative Adversarial Networks. *arXiv*. https://doi.org/10.48550/arXiv.1606.07536

[7] Sojasingarayar, Abonia. (2020). Seq2Seq AI Chatbot with Attention Mechanism.

[8] Bahdanau, Dzmitry & Cho, Kyunghyun & Bengio, Y.. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. ArXiv. 1409

[9] Kuang, Chuwen. 2022. "Research on the Design of Intelligent Chatbot Based on Deep Learning." Journal of Physics: Conference Series 2170 (1): 012017. https://doi.org/10.1088/1742-6596/2170/1/012017

[10] Rishiraj Saha Roy, Aishwarya Padmakumar, Guna Prasaad Jeganathan, and Ponnurangam Kumaraguru. 2015. Automated Linguistic Personalization of Targeted Marketing Messages Mining User-Generated Text on Social Media. In 16th International Conference on Intelligent Text Processing and Computational Linguistics 2015 (CICLing '15), pages 203–224. Springer International Publishing

[11] Rohit, Gurudutt & Dharamshi, Ekta & Subramanyam, Natarajan. (2019). Approaches to Question Answering Using LSTM and Memory Networks: SocProS 2017, Volume 1. 10.1007/978-981-13-1592-3_15.

[12] C. Hu, Y. Ding and Y. Gao, "Artistic Text Style Transfer based on Generative Adversarial Networks," 2020 7th International Forum on Electrical Engineering and Automation (IFEEA), 2020, pp. 451-455, doi: 10.1109/IFEEA51475.2020.00100

[13] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. *arXiv*. https://doi.org/10.48550/arXiv.1409.3215

[14] Dinan, Emily & Logacheva, Varvara & Malykh, Valentin & Miller, Alexander & Shuster, Kurt & Urbanek, Jack & Kiela, Douwe & Szlam, Arthur & Serban, Iulian & Lowe, Ryan & Prabhumoye, Shrimai & Black, Alan & Rudnicky, Alexander & Williams, Jason & Pineau, Joelle & Burtsev, Mikhail & Weston, Jason. (2020). The Second Conversational Intelligence Challenge (ConvAI2). 10.1007/978-3-030-29135-8_7

[15] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory .Neural computation. 9.1735-80. 10.1162/neco.1997.9.8.1735

[16] Sidorov, Grigori & Gelbukh, Alexander & Gomez Adorno, Helena & Pinto, David. (2014). Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model. Computación y Sistemas. 18. 10.13053/cys-18-3-2043

[17] Vinyals, Oriol & Fortunato, Meire & Jaitly, Navdeep. (2015). Pointer Networks. Advances in Neural Information Processing Systems.

[18] G. Datta, N. Joshi and K. Gupta, "Study of NMT: An Explainable AI based approach," 2021 5th International Conference on Information Systems and Computer Networks (ISCON), 2021, pp. 1-4, doi: 10.1109/ISCON52037.2021.9702396

[19] Liu, Xuebo & Wong, Derek & Liu, Yang & Chao, Lidia & Xiao, Tong & Zhu, Jingbo. (2019). Shared-Private Bilingual Word Embeddings for Neural Machine Translation. 3613-3622. 10.18653/v1/P19-1352.

[20] Marcus, Mitchell & Marcinkiewicz, Mary & Santorini, Beatrice. (2002). Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics. 19. 313-330

[21] Khyani, Divya & B S, Siddhartha. (2021). An Interpretation of Lemmatization and Stemming in Natural Language Processing. Shanghai Ligong Daxue Xuebao/Journal of University of Shanghai for Science and Technology. 22. 350-357.

[22] Xu, Xing & Shimada, Atsushi & Taniguchi, Rin-ichiro & He, Li. (2015). Coupled dictionary learning and feature mapping for cross modal retrieval. 10.1109/ICME.2015.7177396

[23] Prabhumoye, Shrimai & Tsvetkov, Yulia & Salakhutdinov, Ruslan & Black, Alan. (2018). Style Transfer Through Back-Translation. 866-876. 10.18653/v1/P18-1080.

[24] Dai, Ning & Liang, Jianze & Qiu, Xipeng & Huang, Xuanjing. (2019). Style Transformer: Unpaired Text Style Transfer without Disentangled Latent Representation. 5997-6007. 10.18653/v1/P19-1601.

[25] Zhou, Xiangyang & Li, Lu & Dong, Daxiang & Liu, Yi & Chen, Ying & Zhao, Wayne & Yu, Dianhai & Wu, Hua. (2018). Multi-Turn Response Selection for Chatbots with Deep Attention Matching Network. 1118-1127. 10.18653/v1/P18-1103.

[26] Xu, R., Ge, T., & Wei, F. (2019). Formality Style Transfer with Hybrid Textual Annotations. *arXiv*. https://doi.org/10.48550/arXiv.1903.06353

[27] Dudarin, Pavel & Yarushkina, Nadejda. (2019). An approach to vocabulary expansion for neural network language model by means of hierarchical clustering. 10.2991/eusflat-19.2019.85.