Turkish Journal of Computer and Mathematics Education Vol.12 No.13 (2021), 134-151

Research Article

An Augmented Encoder to Generate and Evaluate Paraphrases in Punjabi Language

Arwinder Singh [a], Gurpreet Singh Josan [b]

[a] University College, Ghanaur, Punjabi University, Patiala,
[b] Department of Computer Science, Punjabi University, Patiala
Email: [a]arwinder@pbi.ac.in., [b]josangurpreet@pbi.ac.in

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 4 June 2021

Abstract: Paraphrase generation is an important task in Natural Language Processing (NLP) and is successfully applied in various applications such as question-answering, information retrieval & extraction, text summarization and augmentation of machine translation training data. A lot of research has been carried out on paraphrase generation but in the language of English only. However, no approach is available for paraphrase generation in Punjabi Language. Hence, this paper aims to plug in the gap by developing a paraphrase generation and evaluation model for the language of Punjabi. The proposed approach is divided into two phases: paraphrase generation and evaluation. To generate paraphrases, the current state-of-the-art transformer with improved encoder is being used as transformers can learn long-term dependencies. For evaluation, the sentence embeddings are used to check whether the generated paraphrase is similar to the given sentence or not. The sentence embeddings have been created using two approaches: Seq2Seq with attention and transformers. The proposed model is compared with the currently available state-of-the-art models on Quora Question pair dataset. However, for Punjabi, the proposed approach is evaluated on three datasets: news headlines, the sentential dataset from news articles and the third dataset is the translation of Quora Question pair into Punjabi. The automatic evaluation metrics BLEU, METEOR and ROUGE are used for depth evaluation along with human judgments. The proposed approach is straightforward and successfully applies for augmenting machine translation training data and sentence compression. The proposed approach establishes a new baseline for paraphrase generation in Indian regional languages in the future.

Keywords: Deep neural networks, Seq2Seq, Transformer, Paraphrase generation, Sentence vectors

1. Introduction

Paraphrasing is a way to express the same meaning with different expressions. When an event occurs, different newspapers write about such events with different styles and people also discuss these events on social media where millions of comments or tweets take place on a single event. This kind of diversity of text is semantically similar. So, we all deal with paraphrasing in our daily life to express the same thing with different variations. Paraphrasing can be categorized as the detection of paraphrases, their extraction and the generation of paraphrases (Madnani and Dorr, 2010). The research in the proposed article will only focus on paraphrase generation.

The generated paraphrases should be grammatically correct, semantically similar to the original text and must be different from the original one at a certain level. So, paraphrases can be expressed as lexical, phrasal and sentential (Bhagat and Hovy, 2013; Madnani and Dorr, 2010). The replacement of synonyms to represent the same meanings is seen as lexical paraphrases such as (declared→announced). The phrases describe the semantically same meaning are considered as phrasal paraphrases (was a student of University and studied at University). The generation of a semantically similar sentences with different expressions is known as a sentential paraphrase. Sentential paraphrases can be generated by replacing words or phrases or by changing the structure of sentences (Gu et al., 2016; Li et al., 2018; Li et al., 2019; Wang et al., 2019; Kazemnejad et al., 2020). For example: "Cristiano Ronaldo declared as the best soccer player of 2016" can be rephrased as "Cristiano Ronaldo announced as the best soccer player of 2016" and "I clean the house every Sunday" may be written as "The house is cleaned by me every Sunday." Paraphrase generation is very important in various Natural Language Processing applications i.e., query expansion (Hasan et al., 2016; Jones et al., 2006; Soni and Roberts, 2019), question-answering (Duclaye et al., 2003), summarization (Zhou et al., 2006), Paraphrases can also be helpful in dialogue assistance (Shah et al., 2018), augment MT training data (Fader et al., 2014), to extend semantic parsers (Berant et al., 2014), and for question generation (Song et al., 2018). Hasan et al. (2016) and Soni et al. (2019) applied paraphrase generation to understand difficult clinical terms.

Paraphrase generation is a very challenging task due to the complexity of natural languages but a lot of work is done on paraphrase generation. The traditional paraphrase generation approaches used hand-crafted rules (McKeown, 2003), complex paraphrase patterns (Zhao et al., 2009), thesaurus-based (Bolshakov and Gelbukh, 2004; Kauchak and Barzilay, 2006) and statistical machine translation (Quirk et al., 2004; Wubben et al., 2010). These traditional approaches have faced issues of data sparseness and feature extraction was time-consuming. So, a Sequence-to-Sequence model (Sutskever et al., 2014) and deep generative models (Bowman et al., 2016; Chung et al., 2015) of a neural network changed the scenario of text generation and moved research trends to predictive models. These approaches successfully applied to generate semantically similar and syntactically controlled

paraphrases. A Seq2Seq learning enhanced with attention (Bahdanau et al., 2014) and copying mechanisms (Gu et al., 2016; Song et al., 2018) and also combined with Variational Autoencoders (VAE) for paraphrase generation (Gupta et al., 2017). A pair-wise discriminators approach proposed by (Patro et al., 2019) for paraphrase generation. A little work is done to generate controllable paraphrases by Chen et al. (2019); Iyyer et al. (2018). These deep learning approaches produced remarkable results over traditional models but still fail to learn long-term dependencies and long sentences. Vaswani et al. (2017) proposed self-attention which is known as the transformer model. It's originally developed for machine translation and produced amazing results. The multi-head self-attention nature of this model helps to learn long-term dependencies which are further used for various natural language generation tasks and especially in paraphrase generation. There are various paraphrase generation models proposed with transformer architecture (Bao et al., 2019; Egonmwan and Chali, 2019; Guo et al., 2019; Li et al., 2019; Roy and Grangier, 2019; Wang et al., 2019).

Recurrent Neural Network processes one sequence at a time for language generation related tasks which restrict the diversity of the model and also slow down the performance of the model. It's also unable to learn long-term dependencies between sentences. The current state-of-the-art transformer model proposed by Vaswani et al. (2017) is a multi-head self-attention model. It can learn long-term dependencies and can also process multiple sequences at once along with keeping the track of positions of words. Though, various paraphrase generation models developed with such transformer's models, still there is a lack of long-term memory in deep layers in these approaches. Some of the researchers combined seq2seq with transformers (Egonmwan and Chali, 2019; Li et al., 2019) whereas seq2seq are unable to process long sentences. So, instead of using RNN architectures, the proposed article presents a new paraphrase generation and evaluation approach for Punjabi language using current state-of-the-art transformers. There are complex relationships between phrases and sentences in Punjabi language. Such findings can be improved by using multi-layered architecture. The deep layered architecture may suffer through degradation problem but this can be overcome by using residual connections between layers as described in (Prakash et al., 2016). The work done by (Abrishami et al., 2020) enhanced transformer (Vaswani et al., 2017) with hybrid input to use residual connections between multiple encoder-decoder layers. But the encoder is more important to extract long-term dependencies and also to capture semantic as well as syntactic relations between sentences (Egonmwan and Chali, 2019). The proposed article presents a new paraphrase generation approach for Punjabi by enhancing the transformer's encoder and the decoder is same as described in (Vaswani et al., 2017). The encoder is improved by combining the previous layer's output with initial inputs as input for the next layer. There is no such work done paraphrase generation in Punjabi language. The proposed approach further applies to sentence compression and augmenting machine translation training data.

The evaluation of paraphrase generation means whether the generated sentence is a paraphrase of the given sentence or not. The paraphrase generation models developed so far evaluated with BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) and METEOR. These evaluation metrics originally developed for automatic machine translation and can't only be trusted for the depth analysis of results of paraphrase generation. So, the proposed approach for paraphrase generation further evaluated using sentence embeddings. The generated paraphrases with the proposed approach evaluated with two types of sentence embeddings, one created in Singh and Josan (2021) and the second with transformers. Along with this, human evaluations, BLEU and ROUGE scores.

The proposed approach is evaluated on three datasets for getting the depth performance of the model. DATASET-I (Singh and Josan, 2021) is collected from news headlines, DATASET-II (Singh and Josan, 2021) is gathered from news articles and DATASET-III is the translation of Quora Question pairs in Punjabi. Our approach generates state-of-the-art results in paraphrase generation and performs better than previous models.

The next section presents related work followed by the proposed methodology. Section 4 provides an introduction to the datasets followed by experimental details. The depth evaluation of the proposed approach is explained in section 6. Then an analysis is presented in the section 7 followed by the conclusion and future work.

2.Related Work

There are various paraphrase generation approaches which can be categorized as traditional and recent state-of-the-art approaches. Traditional approaches may include Data-driven explored by Madnani and Dorr (2010), rule-based and Statistical Machine Translation (SMT). The state-of-the-art models are working with Recurrent Neural Networks and current transformers models. This section will explore these paraphrase generation approaches.

Data-driven approaches explored by Madnani and Dorr (2010) worked with distributional similarity for generating paraphrases but the accuracy of these approaches was limited due to alignment techniques and also faced issues of data sparseness. Paraphrase generation further enhanced with Vector Space Model (VSM) which is used to find "word meanings in context" (Schütze 1993). This hypothesis focuses to compute relatedness as

semantic similarity instead of dictionary or WordNet. Erk and Padó (2008); Malakasiotis and Androutsopoulos (2011); Shutova et al. (2012) worked to generate paraphrases with VSM. A little work is done with Statistical Machine Translation (SMT) for paraphrase generation by Zhao S. et al. (2009); Wubben S. et al. (2010); Xu W. et al. (2013) and such models also require a large amount of parallel data.

In these days, research moved to predictive models and various researchers worked on paraphrase generation using these predictive models (Prakash et al., 2016; Gupta et al., 2017; Li et al., 2018). A very famous Recurrent Neural Network (RNN) based model is seq2seq (Sutskever et al., 2014) which considered paraphrase generation as a sequence-to-sequence task. It's an encoder-decoder architecture where encoder and decoder are Long-Short-Term-Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997). These neural networks, especially the generative models (Bowman et al., 2016) which enhanced the text generation results. The conditional, attention (Bahdanau et al., 2014; Luong et al., 2015) and copying mechanism (Vinyals et al., 2015) further improved paraphrase generation which is used by various authors i.e. (Brad et al., 2017; Cao et al., 2017; Ma et al., 2018; Patro et al., 2018; Prakash et al., 2016). The proposed research by Brad et al. (2017) used SMT for paraphrase generation with transfer learning. The authors used the Seq2Seq model with attention proposed by Luong et al. (2015) and Sutskever et al. (2014). Cao et al. (2017) further extended Seq2Seq as Copying and Rewriting (CoRe) for improving paraphrase generation. They used one encoder and two decoders for copying and rewriting patterns. The work is done in the paper (Ma et al., 2018) focused to learn the semantic relationship between sentences and give it the name Word Embedding Attention Network (WEAN). They tried to generate new words as retrieval fashion instead of generative modeling. The ability to learn semantic and syntactic relationships of sentences further extended by Patro et al. (2018). They proposed a new model for paraphrase generation by combining existing seq2seq with a pair-wise discriminator. The proposed approach by the authors used an encoder, decoder and LSTM discriminator.

The basic Seq2Seq architecture can be extended by using multiple LSTM layers which are called stacked LSTM. By stacking multiple layers, the network may often suffer through degradation problems and such problems can be solved by using residual connections (He et al., 2016) between multiple layers. One such methodology proposed by Prakash et al. (2016) in which they used four stacked layers of LSTM with residual connections. The authors used residual connections used at layer 2 and their model produced impressive results. The residual connections have also been used in self-attention proposed by the Google team (Vaswani et al., 2017) where a model can be made using multiple layers.

The architecture proposed in (Vaswani et al., 2017), produced state-of-the-art results by stacking multiple encoder-decoder layers and multiple heads. This model is also able to learn long-term dependencies between sentences. Many researchers (Egonmwan and Chali, 2019; Wang et al., 2019; Li et al., 2019; Roy and Grangier, 2019; Bao et al., 2019) used this ability of transformer model for paraphrase generation and further extended transformer model for better results. (Wang et al., 2019) enhanced transformer by using multi encoders with PropBank to learn semantic as well as syntactic information. Another approach developed by Roy et al. (2019) is the combination of the transformer model and VAE as an unsupervised paraphrase model. The vanilla transformer acts as autoregressive where the decoder generates sequences one by one. This can be improved by modifying the decoder so that a decoder can produce sequences simultaneously. Bao et al. (2019) proposed a non-autoregressive paraphrase generation modeling.

Some of the researchers combined Seq2Seq and transformer for improving paraphrasing tasks. Li et al. (2019) developed a Decomposable Neural Paraphrase Generation (DNPG) in which they designed four modules i.e., separator, encoder, decoder and aggregator. The separator is used for finding multi granularities, encoder-decoder is the vanilla transformer for encoding and decoding and aggregator for combining the outputs. In their model, they used LSTM for separator and aggregator but transformer as encoder-decoder. One more approach discovered by Egonmwan and Chali (2019) in which the authors used one transformer's encoder and GRU encoder-decoder. They proposed a transformer encoder to extract features of input sentences and passed these features to the GRU encoder to generate fixed-size context vector. GRU decoder further decodes the output by reading this context vector.

RNN can process one sequence at a time which restricts the diversity of the model and slow down the performance of the model. On the other hand, transformers can process multiple sequences at a time so that they can produce state-of-the-art results. So, this research gap can be improved by using the transformer model. Another finding in this article is that instead of using residual connections with LSTM layers (Prakash et al., 2016), we should use residual connections within the currently proposed transformer by Vaswani et al. (2017). So, Abrishami et al., (2020) enhanced transformer with hybrid inputs for machine translation. The authors passed the original inputs to linear layer and they called it 'Hybrid Inputs'. The used these hybrid inputs in encoder as well as

in decoder and they simply added the inputs with outputs of previous layers. By doing this, their model achieved 36.7 BLEU score.

3.Methodology

3.1 Problem Formulation for Paraphrase Generation

Some of the traditional approaches described paraphrase generation as distributional similarity (Madnani and Dorr, 2010), little work solved paraphrase generation as a syntax-based problem (Erk and Padó, 2008; Malakasiotis and Androutsopoulos, 2011) and some of the researchers considered it as SMT problem (Zhao et al. 2009; Wubben et al., 2010; Xu et al., 2013). Though traditional approaches generated semantically and syntactically correct paraphrases but these approaches faced some issues. The feature extraction was time-consuming (Madnani and Dorr, 2010) and another was data sparseness. But the text generation capability of RNN changed the scenario to predictive models. One such method is Seq2Seq (Sutskever et al., 2014; Li et al., 2018) formulated paraphrase generation as sequence-to-sequence problem. It helped to work on this difficult task with low dimensions, taking small time for training, can recognize synonyms and preserve the semantic and syntactic relations.

The graphical representation of this is shown in Figure 1. Given an input sentence, S = [i1 ... iK] with the length K and we need to generate a target sentence T = [j1...jL] with length L. The generated sentence T should be semantically and syntactically similar to the input sentence S where lengths K & L may or may not be the same as shown in the below example. There can be various granularity levels of paraphrases such as lexical, phrase or sentence. For a machine it is difficult to find semantic and syntactic relations between words, phrases, or sentences. It becomes more challenging to find semantic relations in the noisy text (Xu et al., 2013) but the current state-of-the-art models produced effective results due to their multi-head self-attentive nature. So, the proposed work in this paper outperforms this difficult task for Punjabi language on sentences as well as on phrases.

Input sentence: ਮਾਮਲੇ ਦੀ ਸ਼ਿਕਾਇਤ ਪੁਲੀਸ ਨੂੰ ਦਿੱਤੀ ਗਈ ਹੈ |

(mamle di shikayat police noon ditti gayi hai)

(The matter has been reported to the police)

Target sentence: ਘਟਨਾ ਦੀ ਸੂਚਨਾ ਪੁਲਿਸ ਨੂੰ ਦੇ ਦਿੱਤੀ ਗਈ ਹੈ |

(ghatna di soochna police noon de ditti gayi hai)

(The incident has been reported to the police)

3.2 RNN based Seq2Seq

3.2.1 Introduction. A Seq2Seq architecture introduced by Sutskever et al. (2014) for the first time for neural machine translation and now it's widely used in language generation tasks. This architecture includes encoder-decoder which learns a model to map source sequences to target sequences. The sequences may be characters or words which are then represented as fixed-size vectors. An encoder uses LSTM or GRU cells to represent sequences into low-dimensional vectors. This encoded information summarized as latent features, hidden states, or simply a vector representation. The decoder takes this low dimensional vector as an input and regenerates the sequence into a high dimensional vector. As shown in figure 1, all the input tokens (words) feed to the encoder as an input and these can be seen as all the states of the encoder. The encoder produces internal states c_t and hidden states h_t at each time step. All the internal states have been discarded and only hidden states are passed to the decoder. This hidden state is also called a context vector which is the representation of the whole sentence. The encoded thought vector of the encoder becomes the input of the decoder. Now, the decoder is able to generate outputs as sequences after taking encoded sentences as input. To generate a sequence, decoder uses last hidden state and previously generated word (teacher forcing). Each of the target sentences in training data is marked with START and END for representing the start and end of the sentence respectively. This is just for information to the decoder on where to start and end the sentence. At the time of generation, the decoder reads START and generates first token of the target sentence i.e. 'ਘਟਨਾ' (ghatna) (incident). At next time step, the decoder takes the previously generated word 'ਘਟਨਾ' (ghatna) and the previous hidden state as input and generate the next word 'ਦੀ' (dee) (of). This process continues till the decoder reaches END. There is training objective is to maximize the log probability of the target token given the input token. To maximize the probability, a beam size hypothesis is used.

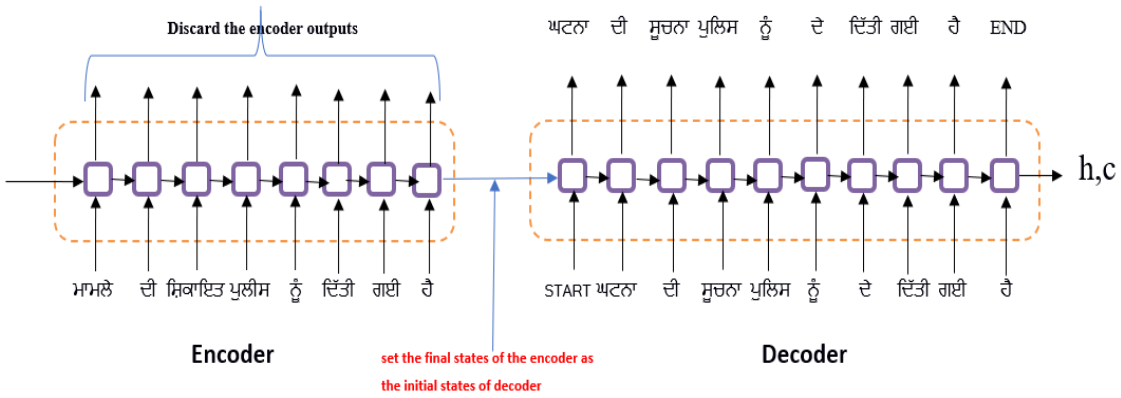

Figure 1: Architecture of Seq2Seq model

Here, both the encoder and decoder are LSTM networks to compute hidden state h_t using a particular approach by adding an internal state c_t at time step t. Input state c_t at time step t, the hidden state h_{t-1}, and the internal memory state c_{t-1} at time step t-1 is used by an LSTM for the generation of the hidden state h_t and internal state c_t at time step t. LSTM uses three learned gates to manage internal states: input (Eq. 1), forget (Eq. 2) and output (Eq. 3) gates.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (2)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (3)$$

There can be various approaches that can be used within encoder-decoder to parse input sequences and to generate the output sequences such as single LSTM, biLSTM, stacked LSTM and GRU. The previous research used these algorithms with attention (Bahdanau et al., 2014; Luong et al., 2015) to improve the results. We can conceptualize that the attention process keeps information that is currently important.

We present a small introduction to the seq2seq model and we first train our paraphrase generation model using this architecture. This architecture doesn't produce good results due to the complex semantic and syntactic relationships of sentences in Punjabi language. This architecture is also unable to learn long-term dependencies.

3.2.2 Problems with the Seq2Seq approach**.** The seq2seq architecture has some limitations when it processes long sentences. There can be long-range dependencies between sentences. A seq2seq architecture is unable to process these types of dependencies as it forgets to retain information of previous words when new words occur in the model. The next drawback is that an encoder-decoder methodology processes the sequences sequentially i.e. one sequence at a time which slows down the performance of the model and restricts the diversity of the model. The position of words in a sentence matters when we process a sentence but the seq2seq model does not learn the position of words. To overcome these drawbacks of original seq2seq (Sutskever et al., 2014), a current state-of-the-art Self-Attention model is known as "Transformer" model (Vaswani et al. 2017) can be used for better performance. It is first developed for machine translation and produced impressive results (Vaswani et al. 2017). It can process multiple words at a time so that it takes less time for training. It's also able to keep track of the positions of words. This architecture then successfully applied in various NLP applications, one of them is paraphrase generation (Egonmwan and Chali, 2019; Wang et al., 2019; Li et al., 2019; Roy and Grangier, 2019; Bao et al., 2019). The proposed research further applies current encoder-decoder with self-attention (Vaswani et al. 2017) to generate paraphrases in Punjabi language.

3.3 Proposed Approach

This section presents the architecture of the proposed paraphrase generation model. We first present an overview of the model and then explain the model in depth.

3.3.1 Overview. As shown in Figure 2, our model is composed of two modules: an augmented encoder and a decoder. This model minimizes the negative log-likelihood of target sequences to map an input sentence X to output sentence Y. The encoder-decoder is made up of n identical layers, where the encoder has two sub-layers i.e., Multi-Head Self-Attention and Feed Forward Network. The decoder has one additional masked multi-head attention layer.

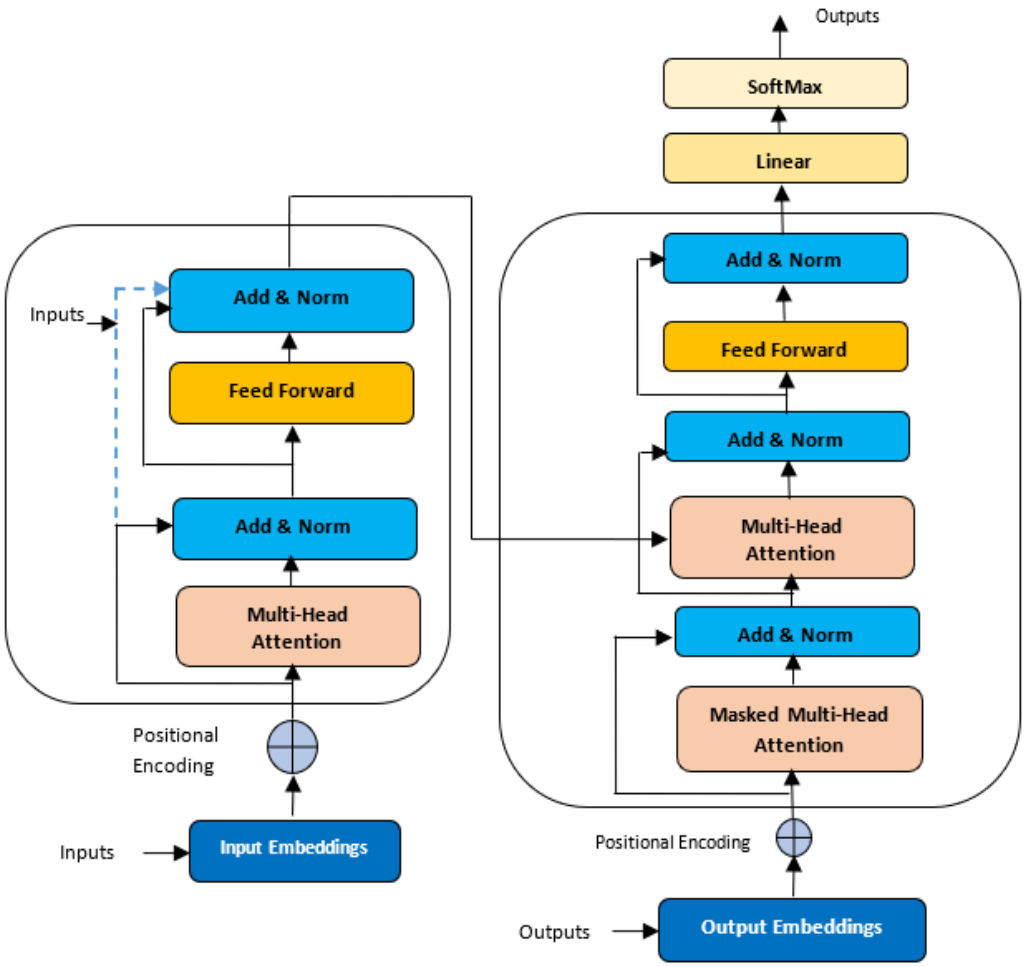

Figure 2: The Block Diagram of Transformer with Augmented Encoder

3.3.2 Encoder. The encoder is an important part like other Seq2Seq models to encode input sequences. The input of the encoder is fixed size vectors generated by the embedding layer. Then encoder converts input sequences X into contextual information E. This contextual information becomes the input of the decoder to produce output sequences Y. The proposed architecture uses n encoders and each encoder has two sub-layers: multi-head attention and feed-forward network. So, the input sentence is passed through n encoders for better learning of the input which generates one output for each token. These layers have residual connections around the layers.

3.3.3 Multi-Head Attention. The key advantage of the high performance of the transformer model is due to multi-head attention. The multi-head attention means the attention is calculated by n heads. Why there are multiple heads, not one? So, one head can extract limited features but multiple heads can extract multiple features. As described in the paper (Vaswani et al., 2017), there is a huge gap when using a single head and 8 heads.

The multi-head attention performs h times linear projection on queries, keys and values to produce multiple attentions. The queries, keys and values are input vectors. These attentions are then concatenated for making multi-head attention and linearly projected as Eq. 4 & 5 where W_h^Q, W_h^K, W_h^V are parameter matrices and W^O is output projection. The attention score is scaled dot-product attention which is calculated as Eq. 6 for each representation.

$$\text{Head}_h = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{4}$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{Head}_1, \ldots . \text{Head}_h)W^O \tag{5}$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

3.3.4 Position-wise Feed-Forward Networks. The second sub-layer is a feed-forward networks which apply a fully connected (FC) layer, a ReLu activation and another fully connected layer.

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)\, W_2 + b_2 \quad (7)$$

3.3.5 Residual Connections. To improve the language models, the transformer model suggests multiple encoder-decoder layers. There may be a degradation problem when using deep layers. To solve this issue, there are residual connections between layers. The transformer applies residual connections around each sub-layer which adds the output of the layer with its input followed by layer normalization. The residual connections are performed as Eq. 8 where x is the input.

$$\mathrm{Fl} = \mathrm{LayerNorm}\,(x + \mathrm{Sublayer}(x)) \quad (8)$$

3.3.6 Positional Encoding. The encoder-decoder has an embedding layer to represent text on fixed-size vector space similar to RNN based Seq2Seq. There is one additional layer is positional encoding in encoder-decoder to preserve the positions of words. For performing sequence-to-sequence operations on text, the order of the words matters. Instead of taking the current position of words, Vaswani et al. (2017) uses positional embedding to represent positions as fixed vectors as described in Eq. 9 and Eq. 10 where pos is the position of the word, 2i is the 2ith dimension of the position embedding vector and d_{model} is the dimension of the sequence. Embeddings of the input sequence are then added with positional embeddings for the input to the encoder.

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \quad (9)$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \quad (10)$$

3.3.7 Augmented Encoder. The current research for language generation is focusing on the use of deep layers so that model can learn better than single-layered network. There may be a lack of long-term memory when using multiple layers. The deep layers restrict the performance of the model. The work in Abrishami et al, (2020) suggested a hybrid input to the encoder-decoder to overcome this problem for machine translation. But for generating paraphrases, the more focus is to extract semantic and syntactic relations between sentences. So, the encoder is more important to learn semantic as well as syntactic relationships between sentences (Egonmwan and Chali, 2019). The work in this article has followed the approach proposed in (Abrishami et al., 2020) to augment encoder of original transformer (Vaswani et al., 2017). Instead of using hybrid input in encoder-decoder, the proposed approach enhances the encoder by passing initial inputs to the second linear layer as shown in Figure 2. The architecture adds the previous layer's output with inputs for making input to the next layer for encoder and this is shown as dotted lines in Figure 2. By doing this, the encoder can learn better semantic and syntactic information.

3.3.8 Decoder. The decoder is composed of three modules i.e. masked multi-head attention, encoder-decoder attention and a fully-connected layer. The masked multi-head attention is used to restrict the decoder to see only the generated tokens and hide the future tokens. This is done by masking the future tokens in a sentence. It means to predict the fourth word, only to attend the third, second and first words. The output of this layer is served as a Query for the next layer. The output of the encoder is used as Key and Value for encoder-decoder attention. Then the fully-connected layer is the same as in the encoder.

3.3.9 Softmax. Our decoder is able to produce fixed-size vector which includes float values and the fully-connected layer projects this vector into a large vector (equal to vocabulary size) which is called logits. Each cell of this vector stores scores of a unique word. So, softmax is a linear layer that will convert these scores into probability. Then the cell with the highest probability is selected and the word associated with this cell is the final output word.

4.Dataset

We evaluate our paraphrasing model for Punjabi language on three datasets. Two paraphrasing datasets are proposed in Singh and Josan (2020), the third dataset is created by translating the Quora question pair dataset from English to Punjabi. The details of each dataset are given below.

4.1 Phrasal and Sentential Paraphrases from News Articles

The newspapers are a big source of paraphrasing datasets (Dolan et al., 2004; Pronoza et al., 2016). When an event happens, various newspapers publish that event with a short headline and article. The two headlines describing the same event can be seen as phrasal paraphrases and sentential paraphrases can be collected from these similar articles as similar articles share the first two to three lines in common. The work in Singh and Josan (2020) presented a deep learning based approach for collecting paraphrases from four Punjabi newspapers. These

two datasets prepared from news headlines and articles for the period of six months. The collected paraphrases are from different domains and the readers can refer to the original paper for more details.

The first dataset is DATASET-I which is phrasal paraphrases from news headlines. It includes 1,10,639 paraphrasing pairs with 42,498 vocabulary. The headlines with length less than 3 and greater than 20 have been discarded.

DATASET-II is a sentential paraphrasing dataset that is collected from news articles. This large dataset has 1,85,069 pairs of sentences with vocabulary 74,040. This dataset contains sentences with lengths greater than 3 and less than 50.

4.2 Quora Question Pairs

Quora released a paraphrasing dataset of questions in January 2017 and it contains 400K pairs. The paraphrasing pair is associated with binary values 0 or 1 whereas 0 means the sentences are not paraphrases and 1 indicates the pair is a paraphrase. Out of 400K paraphrasing pairs, there are 1,42,000 pairs with a binary value is 1. These pairs are in English and then these are translated into Punjabi language using Google translator to create DATASET-III.

Dataset	Training	Test	Vocabulary
DATASET-I	1,10,639	5,000	42,498
DATASET-II	1,80,069	5,000	74,040
DATASET-III	1,42,000	5,000	42,569

Table 1: Statistics of Datasets

5.Experiments

5.1 Models

We have developed four models for depth comparisons as shown in Table 2. Our first model is Sequence-to-Sequence which is the base model. This simplest model is unable to learn complex semantic and syntactic relations between sentences of Punjabi. So that our second model is Sequence-to-Sequence with attention (Luong et al., 2015). The attention improves accuracy but again it's unable to learn long-term dependencies. Our next two models are the state-of-the-art models: transformer base (Vaswani et al., 2017) is to learn long-term dependencies and the last one is the transformer with augmented encoder which is the proposed model in this article.

Models	References
Sequence to Sequence	Sutskever et al., 2014
Sequence to Sequence with Attention	Luong et al., 2015
Transformer base	Vaswani et al., 2017
Transformer with Augmented Input	Our proposed approach

Table 2: Models

5.2 Training

The embedding layer is used to represent words as vectors. Both of the seq2seq models are trained with one LSTM layer and the dimension of hidden units of LSTM is set to 512. We have experimented with two optimizers i.e., Adam and rmsprop. The categorical_crossentropy is used as a loss function. The models have been run for 30 epochs and we also used early stopping to save the best model, so, our first model is stopped after 18 epochs and the second model has taken 22 epochs. We have used a batch size of 64 due to the large hidden size of LSTM units. These seq2seq models are trained on GPU NVIDIA Quadro P4000.

For the transformer base and transformer with augmented encoder models, the dimension of the model and word embeddings are set to 512 whereas hidden units are fixed to 2048. The dropout is set to 0.1 for both models. The transformer with augmented encoder is tested using 4 and 8 heads but the performance of 8 heads is better than 4 as earlier we have discussed more heads can extract multiple features. Again, we have tried using 4 & 6

encoder-decoder layers but the results of 6 layers are more accurate. The warmup steps are set to 4000 for the learning rate. The batch size is set to 128 for both of the transformer models and the models are trained on GPU NVIDIA Quadro P4000. All the models are run for 30 epochs. The training accuracies are shown in Figure 3, 4 & 5 for DATASET-I, DATASET-II, and DATASET-III respectively. The training accuracy for both datasets achieves up to 80%.

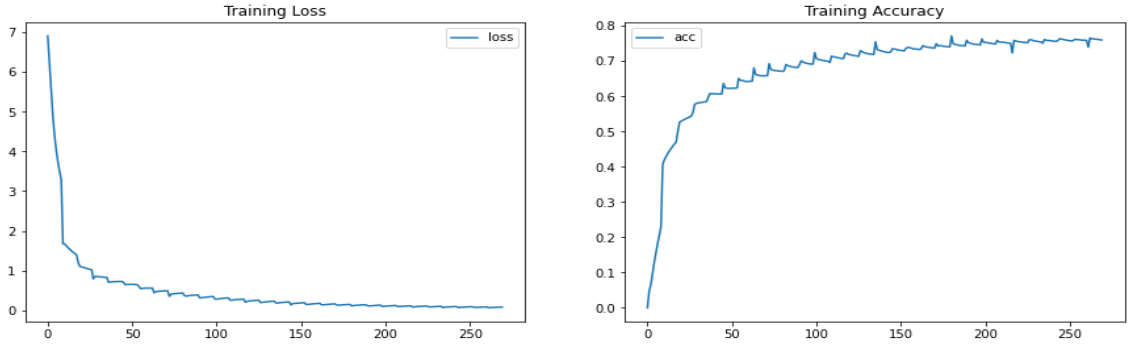

Figure 3: Training Accuracy of Transformer with Augmented Encoder on DATASET-I

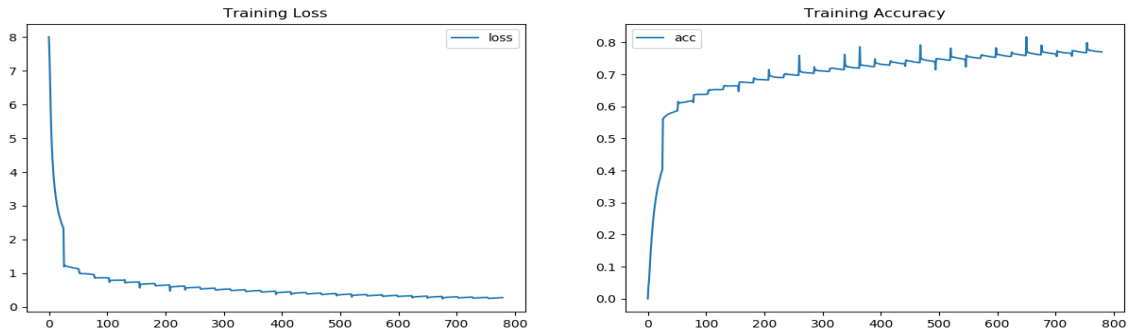

Figure 4: Training Accuracy of Transformer with Augmented Encoder on DATASET-II

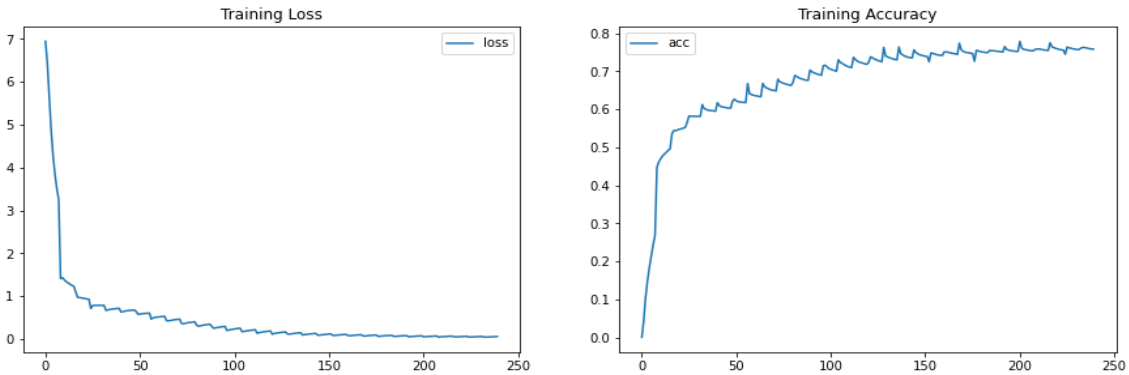

Figure 5: Training Accuracy of Transformer with Augmented Encoder on DATASET-III

6.Evaluation

6.1 Metrics

The proposed approach for paraphrase generation is then evaluated with quantitative and qualitative metrics. For quantitative evaluation, we have used a well-known evaluation metric BLEU (Papineni et al., 2002) which measures the lexical similarity between generated sentences and their references. This metric is originally designed to evaluate machine translation. So, various previous approaches have successfully applied this to evaluate paraphrase generation. Though there are various automatic metrics for paraphrase generation evaluation, BLEU can perform better on small as well as long sentences. Along with this, for comparison of our model with other models, we use METEOR and ROUGE metrics also.

The lexical similarity can't recognize the semantic similarity between sentences as it only checks n-gram word overlap. These types of metrics can't be trusted for depth evaluation, so we should move towards new evaluation metrics. Sharma et al. (2017) discussed different approaches which measure cosine similarity between sentence embeddings.

We have evaluated our model by finding the cosine similarity between sentence vectors of generated paraphrases and its reference. We have trained two models for creating sentence vectors, the approach proposed in (Singh and Josan, 2021) used to create sentence vectors and the name given to it as Seq2SeqSS (Sequence to Sequence Sentence Similarity) in result table. The second method to generate sentence vectors has followed the architecture proposed in Figure 2 in this work. The word vectors generated by the encoder in Figure 2 are then averaged to get sentence vector and this is referred as Transformers Sentence Vector Similarity (TSVS) in the results table. This is another contribution of the proposed study as there are no trained vector models available for Punjabi.

The automatic evaluations can fail at a certain level. So, the qualitative metric is also applied for depth evaluation. We select three judges familiar with Punjabi language and randomly assigned them 300 generated paraphrases from the test set. The evaluators are asked to measure the paraphrases on three criteria: exact paraphrase, partial paraphrase and no paraphrase. All the judges are asked to give a number 5 to exact paraphrase, 3 to partial paraphrase and 0 to no paraphrase. The average results of this metric are shown in Table 5 as a human evaluation.

6.2 Automatic Evaluation on Paraphrase Generation

The qualitative results on various datasets are shown in Tables 3 & 4. We compare our model with various other models on 100K Quora Question Pair dataset as shown in Table 3. Our model outperforms other models i.e. Residual LSTM, VAE-SVG and DNPG on BLEU and ROUGE. Our more focus on Transformer base and Transformer with augmented encoder. The transformer base performs better than the transformer with augmented encoder on ROUGE scores. But the transformer with augmented encoder produces a good semantic similarity score between generated paraphrases and their ground truth i.e. 88.27. The best results are in bold.

Model	**BLEU_2**	**METEOR**	**R-L**	**EACS**
Residual LSTM (Prakash et al., 2016)	17.57	-	32.40	-
VAE-SVG-eq (Gupta et al., 2017)	20.04	-	33.30	-
DNPG (Li et al., 2019)	25.03	-	37.75	-
Sequence to Sequence (Ours)	26.02	24.8	31.23	68.28
Sequence to Sequence with Attention (Ours)	28.34	25.29	34.14	72.22
Transformer base (Ours)	31.22	27.45	**46.32**	87.03
Transformer with Augmented Encoder (Ours)	**32.15**	**28.28**	45.43	**88.27**

Table 3: Performance of the proposed model against other models on Quora Question Pair Dataset

The results on DATASET-I, DATASET-II and DATASET-III are shown in Table 4. For these datasets, we have used four different metrics for the depth evaluation of our model. The transformer with augmented encoder performs better than all other models on DATASET-I since this dataset contains short sentences. The Seq2SeqSS and TSVS also good on this dataset. DATASET-II contains long sentences, so transformers understand long sentences very well and produce good accuracy on all five metrics. Most of the sentences in DATASET-III are also short as its translated version of the original Quora Question Pair dataset. Our model outperforms on Seq2Seq and transformer base model on all the datasets. The evaluation is also represented in charts for each dataset as in Figure 6, 7, 8. As we see, there are very little difference between transformer base and transformer with augmented encoder.

6.3 Human evaluation on paraphrase generation

Table 5 shows the scores calculated on Human evaluations for transformer base and transformer with augmented encoder. For all datasets, our model generates paraphrases that are very close to their ground truth. The

transformer with augmented encoder performs better on three datasets but the transformer base performs better on DATASET-III for generating good paraphrases.

DATASET-I				
Model	**BLEU_2**	**R-L**	**Seq2SeqSS**	**TSVS**
Sequence to Sequence (Ours)	25.28	31.08	69.35	68.8
Sequence to Sequence with Attention (Ours)	26.3	33.15	71.6	72.3
Transformer base (Ours)	31.08	42.22	73.22	76.2
Transformer with Augmented Encoder (Ours)	**31.18**	**42.52**	**74.55**	**77.83**
DATASET-II				
Sequence to Sequence (Ours)	26.23	29.32	70.32	67.3
Sequence to Sequence with Attention (Ours)	28.4	31.62	72.83	71.6
Transformer base (Ours)	31.25	35.77	75.68	**77.35**
Transformer with Augmented Encoder (Ours)	**32.04**	**36.78**	**77.56**	76.91
DATASET-III				
Sequence to Sequence (Ours)	26.08	31.11	69.12	67.23
Sequence to Sequence with Attention (Ours)	27.52	34.22	71.63	71.45
Transformer base (Ours)	31.22	**47.52**	74.32	75.34
Transformer with Augmented Encoder (Ours)	**33.54**	47.27	**75.85**	**76.23**

Table 4: Performance of the proposed model on DATASET-I, DATASET-II, DATASET-III

Dataset	**Model**	**Exact Paraphrase**	**Partial Paraphrase**	**No Paraphrase**
DATASET-I	Transformer base	71.24%	12.5%	16.26%
	Transformer with Augmented Encoder	**72.76%**	11%	16.24%
DATASET-II	Transformer base	75.15%	11.35%	13.5%
	Transformer with Augmented Encoder	**76.25%**	12.25%	11.5%
DATASET-III	Transformer base	**78.5%**	8%	13.5%
	Transformer with Augmented Encoder	77%	9.5%	13.5%

Table 5: Human evaluation on all datasets

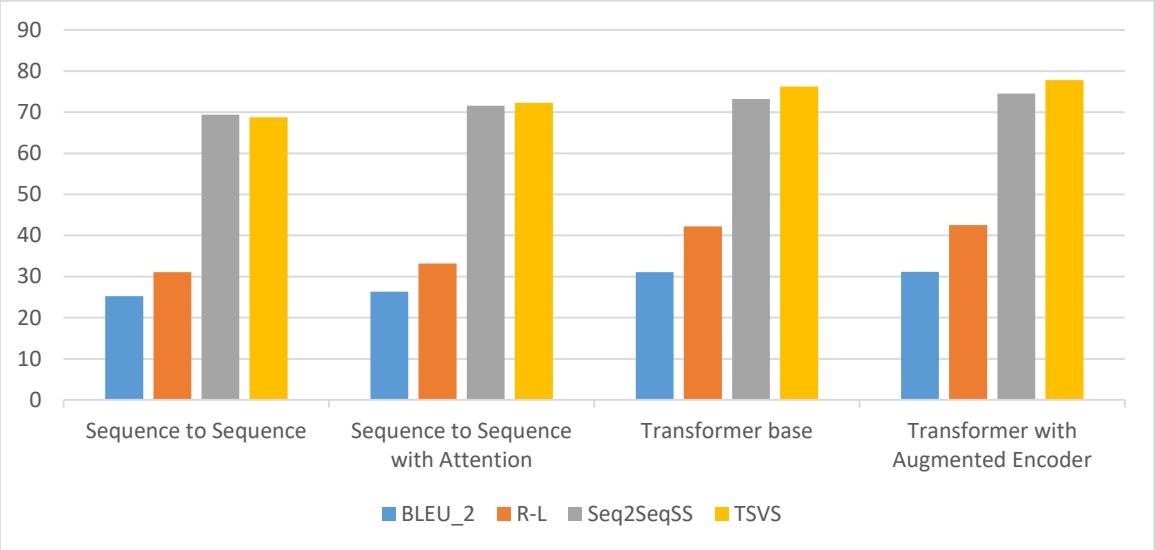

Figure 6: Evaluation of DATASET-I

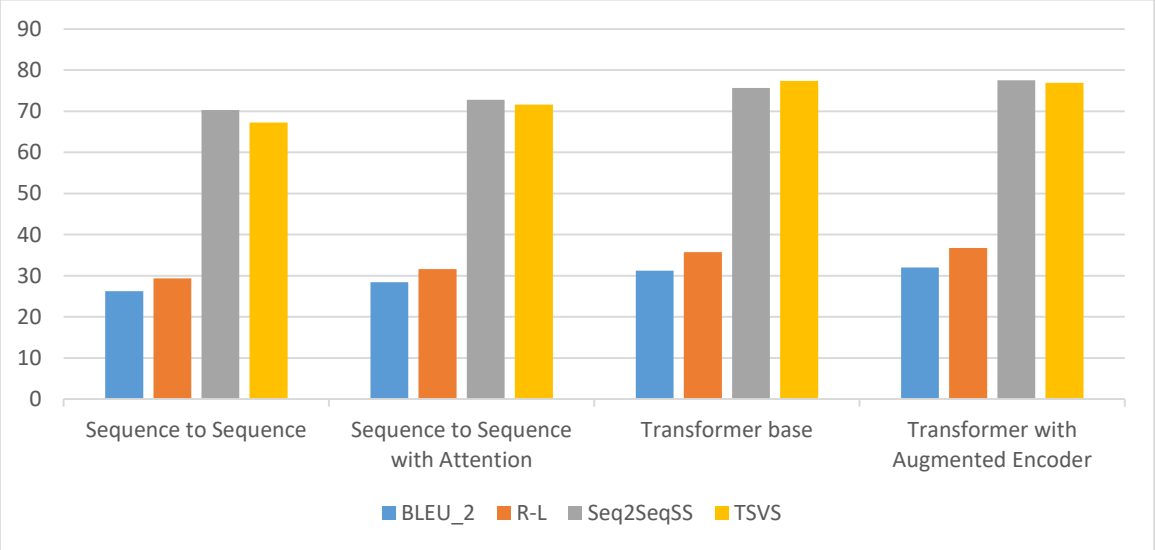

Figure 7: Evaluation of DATASET-II

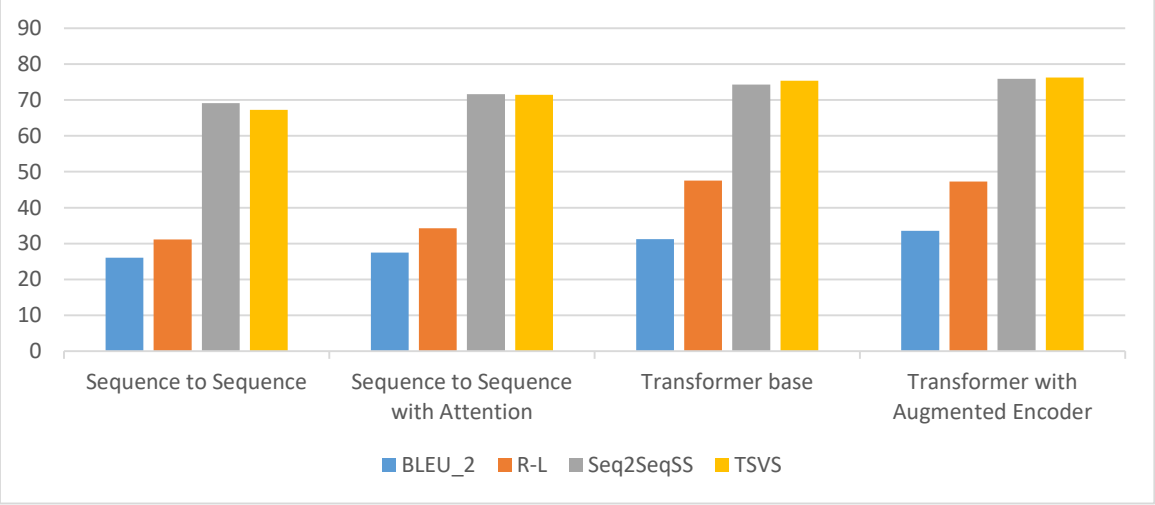

Figure 8: Evaluation of DATASET-III

6.4 Paraphrase Generation Examples

Our model has generated semantically similar and grammatically correct paraphrases. The generated paraphrases are very close to their references. Some of the generated paraphrases are shown in Table 6. Our model generates paraphrases with different variations such as in some cases only synonyms are change and sometimes phrases are paraphrased. In the second example of DATASET-III in Table 6, 'ਚੰਗੇ' (change) (good) is replaced with 'ਵਧੀਆ' (vadhia) (best) but in the first example of DATASET-I, 'ਸੜਕ ਦੁਰਘਟਨਾ' (sadak durghatna) (road accident) replaced with 'ਹਾਦਸੇ' (haadse) (accidents).

7.Analysis

The new approach is originally developed for paraphrase generation and evaluation results show the effectiveness of the model. So, we successfully apply this architecture for two other NLP applications i.e. sentence compression and augmenting machine translation training data.

7.1 Paraphrase Generation

We demonstrate that the transformer architecture can recognize syntactic and semantic representations of short as well as long sentences. The use of multiple layers can improve the paraphrase generation task instead of limited layers of encoders-decoders. To recognize the similarity between generated sentence and its reference, BLEU and ROUGE can just find the word overlaps which fail to detect semantic relations between sentences. So, we use two more approaches for detecting sentence similarity.

Dataset		Source	Generated
DATASET-I	1	ਸੜਕ ਦੁਰਘਟਨਾ ਵਿਚ ਮਾਂ ਪੁੱਤ ਦੀ ਮੌਤ (Sadak durghatna vich maan putt dee maut) (Mother and son killed in road accident)	ਹਾਦਸੇ ਵਿਚ ਮਾਂ ਪੁੱਤ ਦੀ ਮੌਤ (Hadse wich maan putt dee maut) (Mother and son killed in accident)
	2	ਦਿੱਲੀ ਦੇ ਹਿੰਸਾ ਪ੍ਰਭਾਵਿਤ ਇਲਾਕੇ ਵਿਚ ਸਿੱਖਾਂ ਨੇ ਵੰਡਿਆ ਲੰਗਰ (Dilli de hinsa parbhavit ilake vichch sikhan ne vandia langar) (Sikhs distribute langar in violence hit Delhi)	ਹਿੰਸਾ ਪ੍ਰਭਾਵਿਤ ਇਲਾਕੇ ਵਿਚ ਪੀੜਤਾਂ ਨੂੰ ਰਾਸ਼ਨ ਵੰਡਿਆ (Hinsa parbhavit ilake vichch peedtan noon rashn vandia) (Distributed rations to the victims in the violence affected area)
DATASET-II	1	ਪੁਲੀਸ ਨੇ ਕੇਸ ਦਰਜ ਕਰ ਕੇ ਜਾਂਚ ਸ਼ੁਰੂ ਕਰ ਦਿੱਤੀ ਹੈ | (Police ne kes daraj kar ke jaanch shuru kar ditti hai) (Police have registered a case and started investigation)	ਪੁਲੀਸ ਨੇ ਮੁਲਜ਼ਮਾਂ ਖ਼ਿਲਾਫ਼ ਕੇਸ ਦਰਜ ਕਰ ਕੇ ਜਾਂਚ ਸ਼ੁਰੂ ਕਰ ਦਿੱਤੀ ਹੈ| (Police ne mulzaman khilaff kes daraj kar ke jaanch shuru kar ditti hai) (Police have registered a case against the accused and launched an investigation)
	2	ਇਸ ਵਾਇਰਸ ਕਰਕੇ ਹੁਣ ਤਕ ਵਿਸ਼ਵ ਭਰ ਵਿੱਚ ਚਾਰ ਹਜ਼ਾਰ ਤੋਂ ਵੱਧ ਲੋਕ ਮਾਰੇ ਜਾ ਚੁੱਕੇ ਹਨ | (Is wairas karke hun takk vishav bhar vichch char hazaar ton vadhdh lok mare ja chukke han) (The virus has killed more than 4000 people worldwide so far)	ਦੁਨੀਆ ਭਰ ਵਿਚ ਇਸ ਜਾਨਲੇਵਾ ਵਾਇਰਸ ਕਾਰਨ ਹੁਣ ਤਕ 1000 ਲੋਕ ਮਾਰੇ ਜਾ ਚੁੱਕੇ ਹਨ | (Dunian bhar vichch is jaanleva wairas karn hun takk 1000 lok mare ja chukke han) (The deadly virus has killed more than 1000 people worldwide so far)
DATASET-III	1	ਕੀ ਰੱਬ ਸੱਚਮੁੱਚ ਸੰਪੂਰਣ ਹੈ? (Ki rabb sachchmuchch sampoorn hai?) (Is God really perfect?)	ਕੀ ਰੱਬ ਸੰਪੂਰਨ ਹੈ? Ki rabb sampoorn hai? (Is God perfect)
	2	ਮੈਂ ਵਧੀਆ ਸਪੀਕਰ ਕਿਵੇਂ ਬਣ ਸਕਦਾ ਹਾਂ? (main vadhia speekr kiven ban sakda haan?) (How can I become a better speaker?)	ਮੈਂ ਇੱਕ ਚੰਗਾ ਜਨਤਕ ਭਾਸ਼ਣਕਾਰ ਕਿਵੇਂ ਬਣ ਸਕਦਾ ਹਾਂ? (main ikk changa jantak bhashnkaar kiven ban sakda haan?) (How can I become a good public speaker?)

Table 6: Paraphrases generated using Transformer with Augmented Encoder

7.2 Sentence Compression

Sentence compression is a technique to reduce the long sentence without losing the meaning of the sentence. So, we apply our paraphrase generation model to sentence compression as well. Some examples of compressed

sentences are shown in Table 7. The input sentences are compressed at certain level so, the output sentences contain important information.

1.	Input Sentence	ਤੁਹਾਡੇ ਕੋਲ ਰਹਿਣ ਲਈ 24 ਘੰਟਿਆਂ ਦਾ ਸਮਾਂ ਹੈ, ਤੁਸੀਂ ਧਰਤੀ ਉੱਤੇ ਆਪਣਾ ਆਖਰੀ ਦਿਨ ਕਿਵੇਂ ਖਰਚ ਕਰੋਗੇ?
	Transliterated	Tuhade kol rahin layi 24 ghantian da saman hai, tusin dharti utte apna aakhri din kiven kharch kroge?
	English	You have 24 hours to live, how will you spend your last day on earth?
	Output Sentence	ਤੁਸੀਂ ਆਪਣੀ ਜ਼ਿੰਦਗੀ ਦੇ ਆਖ਼ਰੀ 24 ਘੰਟੇ ਕਿਵੇਂ ਬਿਤਾਓਗੇ?
	Transliterated	Tusin apni zindagi de aakhri 24 ghante kiven bitaoge?
	English	How will you spend the last 24 hours of your life?
2.	Input Sentence	ਤੇਜ਼ ਰਫ਼ਤਾਰ ਕਾਰ ਚਾਲਕ ਵਲੋਂ ਮਾਰੀ ਟੱਕਰ ਨਾਲ ਸਾਈਕਲ ਸਵਾਰ ਦੀ ਮੌਤ
	Transliterated	Tez raftaar kaar chaalk vallon mari takkar naal saikal savaar dee maut
	English	Cyclist killed in collision with speeding car
	Output Sentence	ਕਾਰ ਨਾਲ ਟਕਰਾਉਣ ‘ਤੇ ਸਾਈਕਲ ਸਵਾਰ ਦੀ ਮੌਤ
	Transliterated	Car naal takraun te saikal savaar dee maut
	English	Cyclist killed in car crash

Table 7: Sentence Compression

7.3 Augmenting Translation Training Data

The neural machine translation requires a large dataset for better results but there is a small parallel dataset in Indian regional languages. So, the training data can be augmented by applying the paraphrase approach as every time we get a new sentence as a paraphrase of the given sentence. The proposed methodology for paraphrase generation can be used to augment source training data. For the development of the machine translation systems from Punjabi to any other language, we can increase up to 70% of source data.

7.4 Advantages and Limitations of the Proposed System

Advantages: The work in proposed thesis is another contribution in the area of paraphrase generation. So, various paraphrase generation approaches have been explored in the study and best methods were selected for generating paraphrases. The fundamental aim of the study is to make a system for generating paraphrases in Punjabi

Disadvantages: The proposed system performs low for paraphrasing long sentences of length up to 30-35 words. Another drawback is that the generated paraphrases are partial paraphrase that means generated sentence is not semantically similar to given sentence. The limited paraphrasing dataset is another challenge of the proposed system. The system has been trained on small datasets which can be improved in future.

8.Conclusion and Future Work

The proposed article develops a new paraphrase generation and evaluation approach for Punjabi language using the current state-of-the-art transformer model. The original transformer’s encoder has been augmented with hybrid inputs to generate paraphrases. The evaluation is a paraphrase detection method to measure sentence similarity between generated sentence with given sentence. For evaluation, two types of sentence embeddings have been created for depth evaluation. This new approach then successfully applies in sentence compression and in augmenting machine translation training data. The framework proposed in this article is evaluated on three datasets. The automatic, as well as human evaluation, demonstrate that the proposed approach outperforms the baseline and current state-of-the-art models.

The current paraphrasing work deal with general applications of NLP such as question-answering, sentence simplification. But we find very little work with other real-life applications as (Hasan et al., 2016) and (Soni and Roberts, 2019) worked to rephrase difficult clinical terms. The area of paraphrasing can be expanded by applying it in modern NLP applications in future such as sentence compression, dialogue generation, conversational systems and natural language generation. The literature also shows us paraphrase generation is missing in Indian regional languages. The availability of paraphrasing datasets for these languages is another challenge. So, it will be interesting if we see future work of paraphrase generation in Indian languages.

References

1. Abrishami, M., M. J. Rashti and M. Naderan (2020), "Machine Translation Using Improved Attention-based Transformer with Hybrid Input," 2020 6th International Conference on Web Research (ICWR), 2020, pp. 52-57.
2. Bahdanau, D., Cho, K., and Bengio, Y. (2014), "Neural Machine Translation by Jointly Learning to Align and Translate", CoRR abs/1409.0473, pp. 1-15.
3. Bao, Y., Zhou, H., Feng, J., Wang, M., Huang, S., Chen, J., and Lei, L. (2019), "Non-autoregressive Transformer by Position Learning", ArXiv abs/1911.10677, pp. 1–12.
4. Berant, J., and Liang, P. (2014), "Semantic Parsing via Paraphrasing", In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Baltimore, Maryland, pp. 1415–1425.
5. Bhagat, R. and Hovy, E. (2013), "What Is a Paraphrase"? Computational Linguistics 39, 3, pp. 463–472.
6. Bolshakov, I. and Gelbukh, A. (2004), "Synonymous Paraphrasing Using WordNet and Internet", In proceedings of the 9th International Conference on Application of Natural Language to Information Systems NLDB-2004, Natural Language Processing and Information Systems, Springer-Verlag Berlin Heidelberg, pp. 189–200.
7. Bowman, S., R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R. and Bengio, S. (2016), "Generating Sentences from a Continuous Space", In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics, Berlin, Germany, pp. 10–21.
8. Brad, F. and Rebedea, T. (2017), "Neural paraphrase generation using transfer learning", In Proceedings of the 10th International Conference on Natural Language Generation, Association for Computational Linguistics, Santiago de Compostela, Spain., pp. 257–261.
9. Cao, Z., Luo, C., Li, W. and Li, S. (2017), "Joint copying and restricted generation for paraphrase", In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI Press, San Francisco, California, USA, pp. 3152–3158.
10. Chen, M., Tang, Q., Wiseman, S., and Gimpel, K. (2019), "Controllable Paraphrase Generation with a Syntactic Exemplar", In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, pp. 5972–5984.
11. Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. and Bengio, Y. (2015), "A Recurrent Latent Variable Model for Sequential Data", In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2. MIT Press, Cambridge, MA, USA, pp. 2980–2988.
12. Dolan, B., Quirk, C. and Brockett, C. (2004), "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources", In Proceedings of the 20th International Conference on Computational Linguistics, Geneva, Switzerland, pp. 350–356.
13. Duclaye, F., Yvon, F., Collin, O. and Marzin A. (2003), "Learning Paraphrases to Improve a Question-Answering System", In Proceedings of the 10th Conference of EACL, Workshop on Natural Language Processing for Question-Answering, EACL, Budapest, Hungary, pp. 35–41.
14. Egonmwan, E., and Chali, Y. (2019), "Transformer and seq2seq model for Paraphrase Generation", In Proceedings of the 3rd Workshop on Neural Generation and Translation, Association for Computational Linguistics, Hong Kong, pp. 249–255.
15. Erk, K. and Padó, S. (2008), "A structured vector space model for word meaning in context", In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, ACL, Honolulu, Hawaii, USA, pp. 897–906.
16. Fader, A., Zettlemoyer, L. and Etzioni, O. (2014), "Open Question Answering Over Curated and Extracted Knowledge Bases", In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, USA, pp. 1156–1165.
17. Gu, J., Lu, Z., Li, H. and Li, V., O., K. (2016), "Incorporating Copying Mechanism in Sequence-to-Sequence Learning", In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Berlin, Germany, pp. 1631–1640.
18. Guo, Y., Liao, Y., Jiang, X., Zhang, Q., Zhang, Y., and Liu, Q. (2019), "Zero-Shot Paraphrase Generation with Multilingual Language Models", ArXiv, pp. 1–9.

19. Gupta, A., Agarwal, A., Singh, P. and Rai P. (2017), "A Deep Generative Framework for Paraphrase Generation", In Thirty-Second AAAI Conference on Artificial Intelligence, CORR, Louisiana, USA, pp. 5149–5156.
20. Hasan, S., A., Liu, B., Liu, J., Qadir, A., Lee, K., Datla, V., Prakash, A. and Farri, O. (2016), "Neural Clinical Paraphrase Generation with Attention", In Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP), The COLING 2016 Organizing Committee, Osaka, Japan, pp. 42–53.
21. He, K., Zhang, X., Ren, S. and Sun, J. (2016), "Deep residual learning for image recognition", In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.
22. Hochreiter, S. and Schmidhuber, J. (1997), "Long short-term memory". Neural computation 9, pp. 1735–80.
23. Iyyer, M, Wieting, J., Gimpel, K. and Zettlemoyer, L. (2018), "Adversarial Example Generation with Syntactically Controlled Paraphrase Networks", In Proceedings of the 2018 Conference of the. North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, New Orleans, Louisiana, pp. 1875–1885.
24. Jones, R., Rey, B., Madani, O. and Greiner, W. (2006), "Generating Query Substitutions", In Proceedings of the 15th International Conference on World Wide Web, Association for Computing Machinery, New York, NY, USA, pp. 387–396.
25. Kauchak, D. and Barzilay, R. (2006), "Paraphrasing for Automatic Evaluation", In Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Association for Computational Linguistics, New York, USA, pp. 455–462.
26. Kazemnejad, A., Salehi, M. and Soleymani, M. (2020), "Paraphrase Generation by Learning How to Edit from Samples", In proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6010-6021.
27. Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R., S., Urtasun, R., Torralba, A. and Fidler, S. (2015), "Skip thought vectors", In: NIPS, pp. 3294–3302.
28. Li, Z., Jiang, X., Shang, L. and Li, H. (2018), "Paraphrase Generation with Deep Reinforcement Learning", In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, pp. 3865–3878.
29. Li, Z., Jiang, X., Shang, L. and Liu, Q. (2019), "Decomposable neural paraphrase generation", In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, pp. 3403–3414.
30. Lin, C., Y. (2004), "ROUGE: A Package for Automatic Evaluation of Summaries", In Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, pp. 74–81.
31. Luong, T., Pham, H. and Manning, C.D. (2015), "Effective approaches to attention-based neural machine translation", In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, pp. 1412–1421.
32. Ma, S., Sun, X., Li, W., Li, S., Li, W. and Ren, X. (2018), "Query and output: Generating words by querying distributed word representations for paraphrase generation", In NAACL-HLT, Association for Computational Linguistics, New Orleans, Louisiana, pp. 196–206.
33. Madnani, N. and Dorr, B., J. (2010), "Generating Phrasal and Sentential Paraphrases: A Survey of Data-Driven Methods", Computational Linguistics 36, 3, pp. 341–387.
34. Malakasiotis, P. and Androutsopoulos, I. (2011), "A generate and rank approach to sentence paraphrasing", In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Edinburgh, Scotland, UK. pp. 96–106.
35. McKeown, K., R. (1983), "Paraphrasing Questions Using Given and new information" American Journal of Computational Linguistics 9, 1, pp. 1–10.
36. Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013), "Efficient estimation of word representations in vector space", CoRR abs/1301.3781.
37. Papineni, K., Salim, R., Todd, W. and Wei J. Z. (2002), "BLEU: A Method for Automatic Evaluation of Machine Translation", In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, USA, pp. 311–318.
38. Patro, B., N., Kurmi, V., K., Kumar, S. and Namboodiri, V., P. (2018), "Learning Semantic Sentence Embeddings using Pair-wise Discriminator", In Proceedings of the 27th International Conference on Computational Linguistics, COLLING, pp. 2715–2729.

39. Prakash, A., Hasan, S., A., Lee, K., Datla, V., Qadir, A., Liu, J. and Farri, O. (2016), "Neural paraphrase generation with stacked residual LSTM networks", In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, The COLING 2016 Organizing Committee, Osaka, Japan, pp. 2923–2934.
40. Pronoza, E., Yagunova, E. and Pronoza, A. (2016), "Construction of a Russian Paraphrase Corpus: Unsupervised Paraphrase Extraction", In Proceedings of Information Retrieval: 9th Russian Summer School, RuSSIR 2015, Springer International Publishing, Saint Petersburg, Russia, August 24-28, 2015, pp. 146—157.
41. Quirk, C., Brockett, C. and Dolan, W. (2004), "Monolingual Machine Translation for Paraphrase Generation", In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Barcelona, Spain, pp. 142–149.
42. Roy, A. and Grangier, D. (2019), "Unsupervised Paraphrasing without Translation". In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, pp. 6033–6039.
43. Schütze, H. (1993), "Word space", In Advances in Neural Information Processing Systems 5, Morgan Kaufmann, San Francisco, CA, USA. pp. 895–902.
44. Shah, P., Dilek, Z., H., Tür, G., Rastogi, A., Bapna, A., Kennard, N., N. and Heck, L. (2018), "Building a Conversational Agent Overnight with Dialogue Self-Play", CoRR abs/1801.04871, pp. 1–11.
45. Sharma, S., Asri, L., E., Schulz, H. and Zumer, J. (2017), "Relevance of Unsupervised Metrics in Task-Oriented Dialogue for Evaluating Natural Language Generation", ArXiv, abs/1706.09799, pp. 1-10.
46. Shutova, E., Cruys, T. and Korhonen, A. (2012), "Unsupervised metaphor paraphrasing using a vector space model", In Proceedings of COLING 2012: Posters, Mumbai, India, pp. 1121–1130.
47. Singh, A. and Josan, G., S. (2020), "Construction of Paraphrasing Dataset for Punjabi: A Deep Learning Approach", International Journal of Advanced Science and Technology, 29(6), pp. 9433-9442.
48. Singh, A. and Josan, G., S. (2021), "A Deep Network Model for Paraphrase Detection in Punjabi", In Singh P.K., Singh Y., Kolekar M.H., Kar A.K., Chhabra J.K., Sen A (eds) Recent Innovations in Computing, ICRIC 2020. Lecture Notes in Electrical Engineering, vol 701, Springer, Singapore, pp. 173-185.
49. Song, L., Wang, Z., Hamza, W., Zhang, Y. and Gildea, D. (2018), "Leveraging Context Information for Natural Question Generation", In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, New Orleans, Louisiana, pp. 569–574.
50. Soni, S. and Roberts, K. (2019), "A Paraphrase Generation System for EHR Question Answering", In Proceedings of the 18th BioNLP Workshop and Shared Task, Association for Computational Linguistics, Florence, Italy, pp. 20–29.
51. Sutskever, I., Vinyals, O. and Le, Quoc, V. (2014), "Sequence to sequence learning with neural networks", Advances in neural information processing systems 2, pp. 3104–3112.
52. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., N., Kaiser, L. and Polosukhin, I. (2017), "Attention is All you Need", In NIPS. NIPS, Long Beach, CA, USA, pp. 5998–6008.
53. Vinyals, O., Fortunato, M. and Jaitly, N. (2015), "Pointer networks". In Proceedings of the 28th International Conference on Neural Information Processing Systems – Volume 2. MIT Press, Cambridge, MA, USA, pp. 2692–2700.
54. Wang, S., Gupta, R., G., Chang, N. and Baldridge, J. (2019), "A Task in a Suit and a Tie: Paraphrase Generation with Semantic Augmentation", In The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI, AAAI Press, Honolulu, Hawaii, USA, pp. 7176–7183.
55. Wubben, S., Bosch, A., Van, D. and Krahmer, E. (2010), "Paraphrase Generation as Monolingual Translation: Data and Evaluation", In Proceedings of the 6th International Natural Language Generation Conference, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 203–207.
56. Xu, W., Ritter, A. and Grishman, R. (2013), "Gathering and generating paraphrases from twitter with application to normalization", In Proceedings of the Sixth Workshop on Building and Using Comparable Corpora. Association for Computational Linguistics, Sofia, Bulgaria, pp. 121–128
57. Zhou, L., Lin, C., Y., Munteanu, D., S. and Hovy, E., (2006), "ParaEval: Using Paraphrases to Evaluate Summaries Automatically", In Proceedings of the Human Language Technology Conference of the

NAACL, Main Conference, Association for Computational Linguistics, New York City, USA, pp. 447–454.

58. Zhao, S., Lan, X., Liu, T. and Li, S. (2009), "Application-driven Statistical Paraphrase Generation", In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Association for Computational Linguistics, Suntec, Singapore, pp. 834–842.