Volume 66, Issue 1, 2022

Journal of Scientific Research

of

The Banaras Hindu University

Apply Paraphrase Generation for Finding and Ranking Similar News Headlines in Punjabi Language

Arwinder Singh [1], Gurpreet Singh Josan [2]

[a] University College, Ghanaur, Punjabi University, Patiala, arwinder@pbi.ac.in.

[b] Department of Computer Science, Punjabi University, Patiala, josangurpreet@pbi.ac.in

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 it is missing to apply it for finding similar news headlines. However, no approach is available for finding similar news headlines by using paraphrase generation in Punjabi Language. Hence, this paper presents an approach for finding similar news headlines by applying a paraphrase generation system to plug in the gap. Similar news is being detected by comparing the sentence vectors of news headlines. The input headline is compared with all available headlines using a combination of cosine and Jaccard similarity to find similar headlines. The detected news headlines are then ranked by comparing them with the given headline. To represent news headlines as sentence embeddings, RNN based Seq2Seq with different hyper-parameter settings and attention mechanisms is used. The direct method can restrict the process, so the proposed approach has applied a paraphrase generation system to rephrase the given headline for finding more relevant news headlines. To generate a paraphrase of the given headline, the current state-of-the-art transformer with an augmented encoder is being used as transformers can learn long-term dependencies. The effect of Jaccard and cosine similarity has also been tested along with the combination of both metrics but the combination of Jaccard and cosine similarity performs better for finding similar headlines. The news headlines extracted from four newspapers have been used for evaluation in this article. Further, different evaluation metrics have been applied for in-depth comparisons, the BLEU score has been calculated between detected and input headlines. Another evaluation is done by comparing embeddings of the detected headlines and inputs along with human judgements. The generation of sentence vectors further enhanced the evaluation measure and showing that the proposed approach produced state-of-the-art results.

Keywords Deep Neural Networks, Similar News Headline, Seq2Seq, Transformer, Paraphrase Generation, Sentence Vectors.

I. INTRODUCTION

Paraphrase generation is an important task in Natural Language Processing (NLP) which generates similar text with different expressions in the same language. It can be applied in various applications of NLP 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). The 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.

The available paraphrase generation approaches used hand-crafted rules (McKeown, 2003), complex paraphrase patterns (Zhao et al., 2009), thesaurus-based (Bolshakov and

DOI: 10.37398/JSR.2022.660139

Gelbukh, 2004; Kauchak and Barzilay, 2006) and statistical machine translation (Quirk et al., 2004; Wubben et al., 2010). But the Sequence-to-Sequence model (Sutskever et al., 2014) and deep generative models (Bowman et al., 2016; Chung et al., 2015) of a neural network improved the paraphrase generation task with attention (Bahdanau et al., 2014) and copying mechanisms (Gu et al., 2016; Song et al., 2018). There is lot of work done on paraphrase generation with RNN based Seq2Seq (Gupta et al., 2017, Prakash et al., 2016) whereas the transformer (Vaswani et al. 2017) proposed self-attention which further enhanced this challenging task (Bao et al., 2019; Egonmwan and Chali, 2019; Guo et al., 2019; Li et al., 2019; Roy and Grangier, 2019; Wang et al., 2019). To improve the learning ability with in deep layers, Singh and Josan, (2021) proposed a paraphrase generation approach by augmenting the encoder of the transformer model and achieved very impressive results.

Though paraphrase generation is applicable in various applications but it is missing for finding similar news headlines. There are lot of news available online as well as offline and people try to find related news. The proposed deep learning approach finds news headlines by comparing the semantic similarity between headlines. To find more relevant and similar news, this paper further applies paraphrase generation to rephrase the given headlines. If we have a headline H, then find a similar headline H1 without paraphrase generation and H2 by paraphrasing the given headline as shown in the Fugure 1. After getting multiple headlines for single headline, the approach ranks the headlines on the basis of cosine similarity. The most similar headline to the given headline will be at rank first and then second. The ability to learn long-term dependencies, transformer with augmented encoder (Singh and Josan, 2021) is being applied in this paper for paraphrasing the given headlines to find similar news.

In the proposed article, the similar news is being detected by comparing the sentence vectors with combination of cosine and Jaccard similarity metric. So, the important task of the article is to represent headlines as vectors. The work in Singh and Josan (2021) compared two approaches for generating sentence vectors in paraphrase detection system, one is by averaging word vectors and second is RNN based Seq2Seq whereas seq2seq has produced impressive results for paraphrase detection. The RNN based Seq2Seq with attention mechanism (Luong et al., 2015) discussed in Singh and Josan (2021) is being used in this article to make headlines as vectors. Then input headline's vector is then compared with all the headlines to find similar headline. This direct methodology can detect limited headlines. So that the input headline is then paraphrased and then their sentence vectors compared with all the headlines to find more relevant headlines. The advantage of using headlines is that the headlines are short in length and it takes less time for comparison. The proposed approach has explored new application of paraphrase generation which can detect multiple as well as more accurate news.

The news headlines collected in Singh and Josan, (2020) is being used in this article for finding similar news. To know whether the detected headline is similar to the given one or not, two automatic evaluations have been done. First is the BLEU score calculated between detected headline and input headline. The second by comparing the sentence embeddings of the detected and input headlines. For in-depth evaluation of the detected headlines, human judgements have been made. The proposed approach is very simple and straightforward to find similar as well as accurate news headlines.

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

II. RELATED WORK

H. Torun and A. B. (2018) presented an approach for finding similar news by summarizing the Turkish news. They have used extraction-based summarization technique to summarize the news articles. The authors collected 12000 news articles from different sources and then found 1500 similar news pairs.

There are various paraphrase generation approaches, some are traditional and others are recent state-of-the-art approaches. Data-driven explored by Madnani and Dorr (2010), rule-based and Statistical Machine Translation (SMT) can be seen as traditional approaches whereas the state-of-the-art models those are based on Recurrent Neural Networks and current transformers models.

In these days, research moved to 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 accepts paraphrase generation task as a sequence-to-sequence task. The attention mechanism proposed by (Bahdanau et al., 2014; Luong et al., 2015) and copying mechanism (Vinyals et al., 2015) enhanced paraphrase generation (Brad et al., 2017; Patro et al., 2018; Prakash et al., 2016). The work in Brad et al. (2017) used Statistical Machine Translation for generating paraphrases with transfer learning. The authors used the RNN based Seq2Seq model with attention proposed by Luong et al. (2015) and Sutskever et al. (2014). The work in Cao et al. (2017) extended the Seq2Seq as Copying and Rewriting (CoRe) for improving paraphrase generation task. One encoder and two decoders for copying and rewriting patterns have been used. The ability to learn semantic and syntactic relationships of sentences further extended by Patro et al. (2018) for paraphrase generation by combining the existing seq2seq with a pair-wise discriminator.

Their approach used an encoder, decoder and LSTM discriminator.

The transformer model developed by Vaswani et al. (2017) produced state-of-the-art results which is multi-head attention architecture. This model is able to learn long-term dependencies. A lot of work is done with transformer for generating paraphrases (Egonmwan and Chali, 2019; Wang et al., 2019; Li et al., 2019; Roy and Grangier, 2019; Bao et al., 2019). One more approach discovered by Egonmwan and Chali (2019) in which the authors combined RNN based Seq2Seq and transformer for improving paraphrase generation and stated that encoder is more important to learn semantic and syntactic features. 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.

The transformer model proposed the use of multi-layered architecture where multiple layers of encoder-decoder can be used. But in the multi-layered network, there can be degradation problem. So, Singh and Josan (2021) presented an approach for paraphrase generation by augmenting the encoder of the transformer. The ability of the model is to learn long-range dependencies with in deep layers, this paper applied the approach for paraphrasing the headlines for finding similar news.

To find a similar news on the basis of headlines is an important task. When there are thousands of news available, we find a similar news for the given one. But the normal approach can restrict the process and only able to find limited headlines. So, this article applies the advantage of paraphrase generation system for finding similar news. The paraphrase generation is very relevant when we work with news articles as there is lot of similar information in news articles. The newspapers publish same news with different expressions and there are lot of similar items in newspapers. So, we all deal with paraphrasing in our daily life to express the same thing with different variations. There can be lexical, phrasal or sentential paraphrases (Madnani and Dorr, 2010). The proposed research work is being used the advantage of phrasal paraphrases.

The task for finding similar news is done by representing the headlines as vectors. There can be different techniques for representing headlines as vectors but the work in Singh and Josan (2021) developed a paraphrase detection approach where two sentence vector generation approaches have been explored. Their approach presents a comparison between the models and shown the effectiveness of RNN based Seq2Seq. So, in the proposed article, the headlines have been represented as sentence vectors by applying RNN based Seq2Seq approach.

III. METHODOLOGY

A. Problem Formulation for Finding Similar News by Paraphrasing

To find a similar news on the basis of headlines is an important task. When there are thousands of news available, we can find a similar news for the given news. But the normal approach can restrict the process and can detect limited headlines. So, this article applies the advantage of paraphrase generation system for finding similar news. If we given a headline H, then find a similar headline H1 without paraphrase generation and H2 with paraphrase generation system as shown in Figure 1. Now, rank the headlines H1 and H2 on the basis of cosine similarity between their embeddings and put the similar news to given headline at rank one and then the second headline.

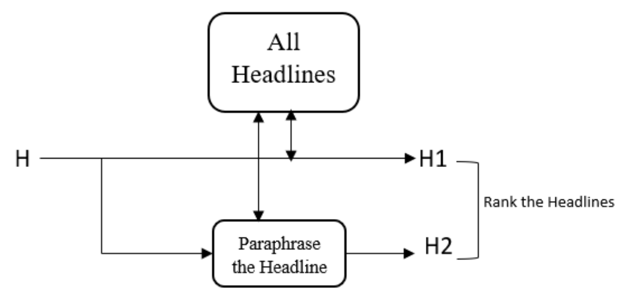

Figure 1: Problem Formulation for Finding Similar News by Paraphrasing

B. General Architecture to Find Similar News by Paraphrasing the Given Headline

There are various modules of the general architecture for finding similar headlines and are shown in Figure 2.

1) Input

This is the step of the proposed model where the inputs will be news headlines in Punjabi.

2) Pre-processing

Some pre-processing steps are done on input headline before finding the similar headlines such as to discard too small and too long headlines, to remove punctuation & special characters and last is to tokenize the input headline.

3) Generate Paraphrase

The next module is to generate paraphrase of the given headline so that more headlines can be detected. Here, state-of-the-art model “An Augmented Encoder to Generate and Evaluate Paraphrase” has been used for paraphrase generation.

4) Sentence Vector

The paraphrased headline is then passed to RNN based Seq2Seq model to generate sentence vector.

5) All Headlines as Sentence Vectors

This is the important phase of the architecture, as the similar headlines have been found by comparing the sentence vectors. RNN based Seq2Seq with attention mechanisms is being used for representing headlines as vectors.

6) Cosine & Jaccard Similarity

The combination of cosine and Jaccard similarity has been applied to detect similar headlines. The effect of cosine and Jaccard similarity has been discussed in result section.

7) Ranking the Headlines. In the last step of the architecture, we get multiple similar headlines. These headlines are then compared with original headline using cosine similarity. On the basis of the cosine similarity score, the headlines have been ranked. The headline has been ranked one with highest cosine score and ranked last with lowest cosine score.

C. Paraphrase Generation

1) Transformer with Augmented Encoder for Paraphrase Generation

The encoder in transformer model is a left part of the model as shown in Figure 3 to extract the features. To extract more relevant information, encoder suggests to use multiple layers of encoder. Here, each of the encoder has two sub-layers, multi head attention and a feed forward layer.

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. 1 & 2 where W_h^Q, W_h^K, W_h^V are parameter matrices and W^O is output projection. The attention score is

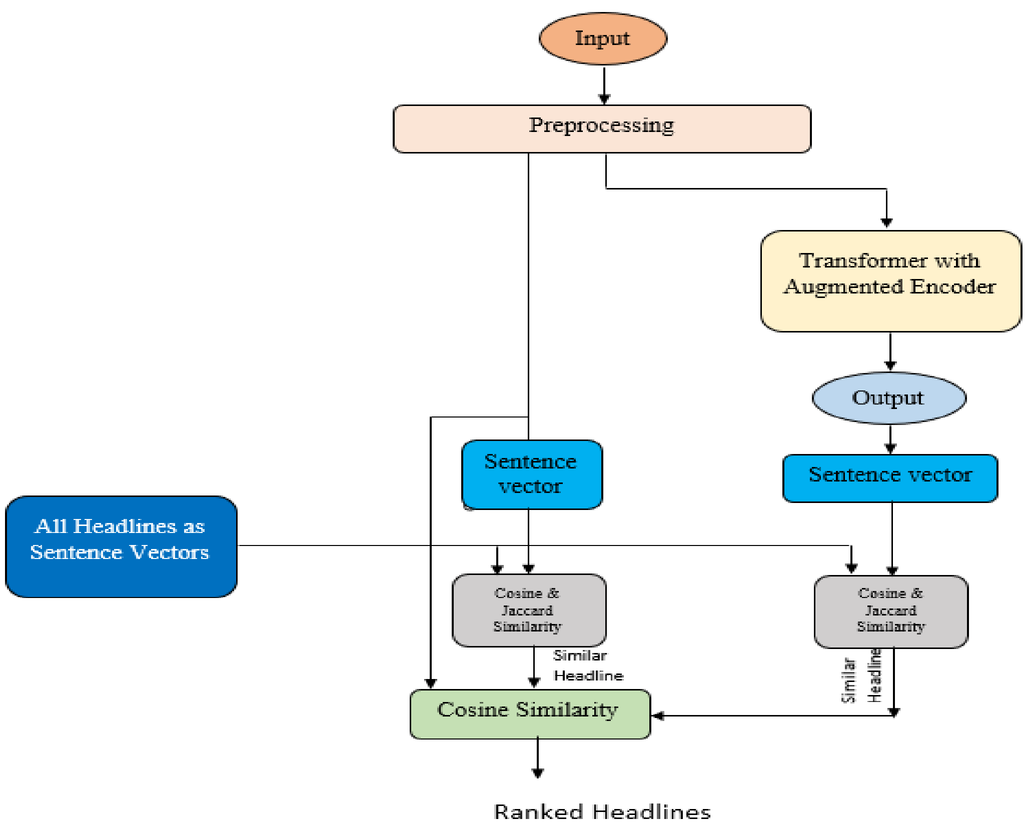

Figure 2: The Block Diagram to Find Similar Headline by Paraphrasing

scaled dot-product attention which is calculated as Eq. 3 for each representation.

$$\mathrm{Head_h} = \mathrm{Attention}(\mathrm{QW_i^Q}, \mathrm{KW_i^K}, \mathrm{VW_i^V}) \quad (1)$$

$$\mathrm{MultiHead(Q, K, V)} = \mathrm{Concat}(\mathrm{Head_1}, \ldots . \mathrm{Head_h})\mathrm{W^O} \quad (2)$$

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

The second sub-layer is a feed-forward networks which apply a fully connected (FC) layer, a ReLu activation and another fully connected layer.

$$\mathrm{FFN(x)} = \max(0, \mathrm{xW_1} + \mathrm{b_1})\,\mathrm{W_2} + \mathrm{b_2} \quad (4)$$

The transformer model uses residual connections around each of the sub-layer which adds the output of the layer with its input followed by layer normalization to improve the performance of the model. One more advantage of the transformer model is to use positional embeddings which are then added with the embeddings of the input sequences. 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. Then the fully-connected layer is the same as in the encoder. For details, follow the original paper (Vaswani et. al, 2017).

Augmented Encoder. The use of deep layers can improve paraphrase generation as discussed by Singh and Josan, (2021). The use of deep layers can extract more relevant features than single-layered network. But there may be a lack of long-term memory when using multiple layers and the use of deep layers can restrict the performance of the model. As discussed in Egonmwan and Chali, (2019), the encoder is more important to extract semantic and syntactic information from input sequences. So, the work in Abrishami et al., (2020) enhanced the encoder-decoder with hybrid input. But, the work in Singh et. al (2021) is presented transformer with augmented encoder to improve the working of the encoder. Their approach enhanced the encoder by passing initial inputs to the second linear layer as shown in Figure 3. The architecture adds the previous layer's output with inputs for making input to the next layer and this is shown as dotted lines in Figure 3. So, this ability of paraphrase generation system is applied for finding similar news.

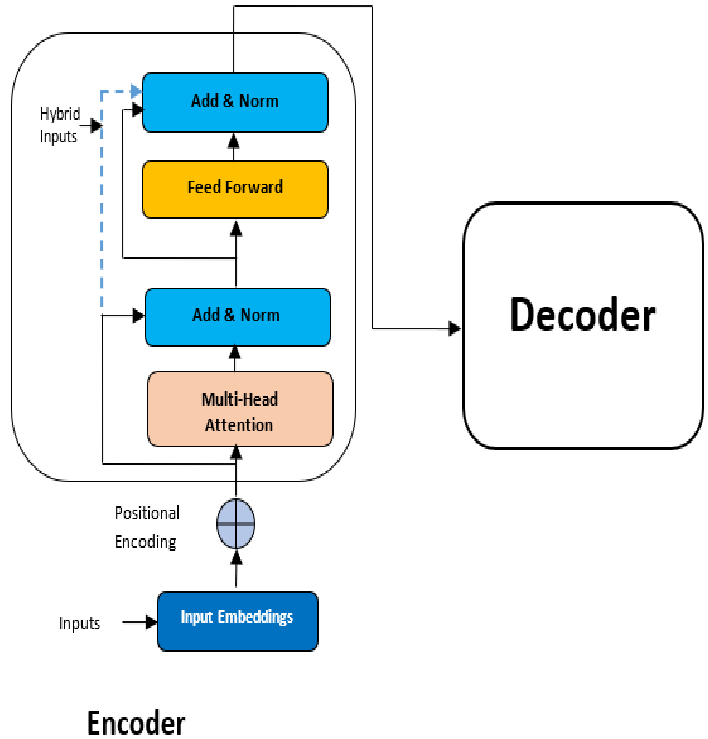

Figure 3: The Transformer with Augmented Encoder

D. Sentence Vector Generation

The next important phase of this article is to represent headlines as sentence vectors. The sentence vectors have been created with RNN based Seq2Seq approach where source and target sequences are same. The encoder is able to extract features from input sequences and put them as a thought vector. After training, this thought vector is read as sentence vectors. The process has been explored in this section.

1) Pre-processing. The Seq2Seq model performs better on clean and noise free training data. After getting input, system should perform some pre-processing on input text is as follows:

a) Clean Data

There may be some punctuation and special characters in input text, those should be removed from input data so that language model can perform better.

b) Append Target headlines with START & END

The decoder of the Seq2Seq model requires target data with identification marks at the beginning and end of the sentence. So, append START and END to the target headlines. One of the examples is shown in Figure 4.

c) Create Tokenizer

The deep learning models can't use text data directly and the text should be converted into numeric. For this purpose, we need a dictionary of unique words to map them into token ID's. So, the input text is tokenized using Keras tokenization layer.

d) Encode Text to Sequences

The next step is to convert text data to sequences.

e) PAD the Sequences

The length of all the sentences is not same and it can vary for all sentences. In the language models, the maximum length of all sequences should be equal to the length of longest sentence. To make all sentences equal, PAD the sequences with 0's up to MAX length.

2) RNN based Seq2Seq with Attention Mechanism

The approach has been applied to get sentence vectors in Punjabi language. The input sequences pass through different layers to process the input sequences as shown in Figure 4 where the encoder is made of following layers:

a) Input Layer

The responsibility of this layer is to take source sequences and pass these sequences to the embedding layer.

b) Embedding Layer

The keras embedding layer has been followed in this article which is responsible for converting text into fixed size vectors. The LSTM can't process the input sequences directly, so embedding layer will convert text into numeric i.e., word vectors.

c) LSTM Layer

It takes the fixed size vector generated by embedding layer as an input and generates internal states (ht, ct) and output at time step t. The internal states tell us what the LSTM has read at time step t whereas output at each time step is discarded. At the first-time step, LSTM is initialized randomly and at the last time step, LSTM stores the complete information of input sequences which is called a thought vector. At last time step, this encoded information is passed to the decoder. The important parameter in LSTM is the size of internal units. So, during the experiments, we have tried with 256 and 512 internal units and got best model with 512 units. The semantic and syntactic features extracted by the encoder are then passed to the LSTM layer of the decoder and now, decoder is able to produce one sequence at a time as shown in Figure 4.

The decoder is made of following layers:

a) Input Layer

This layer takes target sequences as input, the target sequences are appended with START and END to denote the start and end of the sentence respectively. For getting the sentence vectors, the target sequences are same as input sequences. This layer passes these sequences to the embedding layer.

b) Embedding Layer

This is again to convert input sequences into fixed size vectors as in the encoder.

c) LSTM Layer

This layer takes previously generated word and last hidden states (also known as teacher forcing) to generate next sequence. Initially, first word will be 'START' and hidden states are the output of encoder. Now, decoder will produce one sequence at a time. Here, greedy decoding has been followed that means a token with the highest probability will be generated. The generated sequences are then passed to final dense layer.

d) Dense Layer

The high dimensional vector (equal to LSTM units) generated by the LSTM of the decoder is not in readable form. So, we need to convert them into low dimensional one hot encoding which represents the target sequence or word. The dense layer is able to do this and produce a readable word or sequence. All the sequences are not predicted in one time step but this prediction takes many steps. Now, the question is that how many time steps will be? The number of time steps will be equal to MAX_LENGTH.

e) Attention Mechanism

Attention in deep learning allows the model to put more focus on important information to predict the output in a Sequence-to-Sequence models. The vanilla RNN can't pay more attention at relevant information. So, an attention mechanism can be used to pay attention on important information. This model has followed attention introduced by Luong et al. (2015) to improve base model. Luong et al. (2015) introduced the concept of global and local attention. The global attention uses all the hidden states of the encoder to generate the context vector whereas local attention considers a subset of the source positions per target word for the same. The illustrative example shown in Figure 4. The steps of the example are as follows:

a) Producing the Encoder Hidden States

The input sequences are passed through embedding layer to the encoder which is responsible to generate hidden states.

b) Decoder RNN

The previously generated hidden states and decoder outputs are passed through the decoder RNN to generate new hidden state for that time step.

c) Calculating the Alignment Score

Then alignment score is the dot product between new decoder hidden states and encoder hidden states.

d) Pass Alignment Scores through Softmax Layer

For each encoder hidden states, the alignment scores are combined and represented into a single vector and then applied softmax layer.

e) Generate Context Vector

The context vector is then calculated by multiplying hidden states and alignment scores.

f) Generate Final Output

The context vector and decoder hidden states are then concatenated and passed through fully connected layer to generate new sequence.

3) Find Similar Headlines

To find similar news is very straight forward task where the vectors with minimum distance are seen as similar headlines. The sentence vectors generated during previous section are compared with all the headlines. The top headline is selected with highest cosine score (eq 5). Then Jaccard similarity (eq. 6) has been calculated on that pair, if the score is more than 0.2 then it is read as similar headline.

$$\text{sim} = \frac{a.b}{|a|\,|b|} \qquad (5)$$

$$\text{Jaccard}\,(S1, S2) = \frac{S1 \cap S2}{S1 \cup S2} \qquad (6)$$

4) Ranking the Headlines

The cosine similarity is applied for ranking the headlines. The similarity score calculated here will denote that how much the input headline and detected headline are similar. The pair with higher similarity will be ranked first and the other will be ranked as second.

IV. DATASET

The news headlines have been collected in paper Singh and Josan (2020). When an event happens, various newspapers publish that event with a short headline and article. So, all the headlines collected in the paper Singh and Josan (2020) are

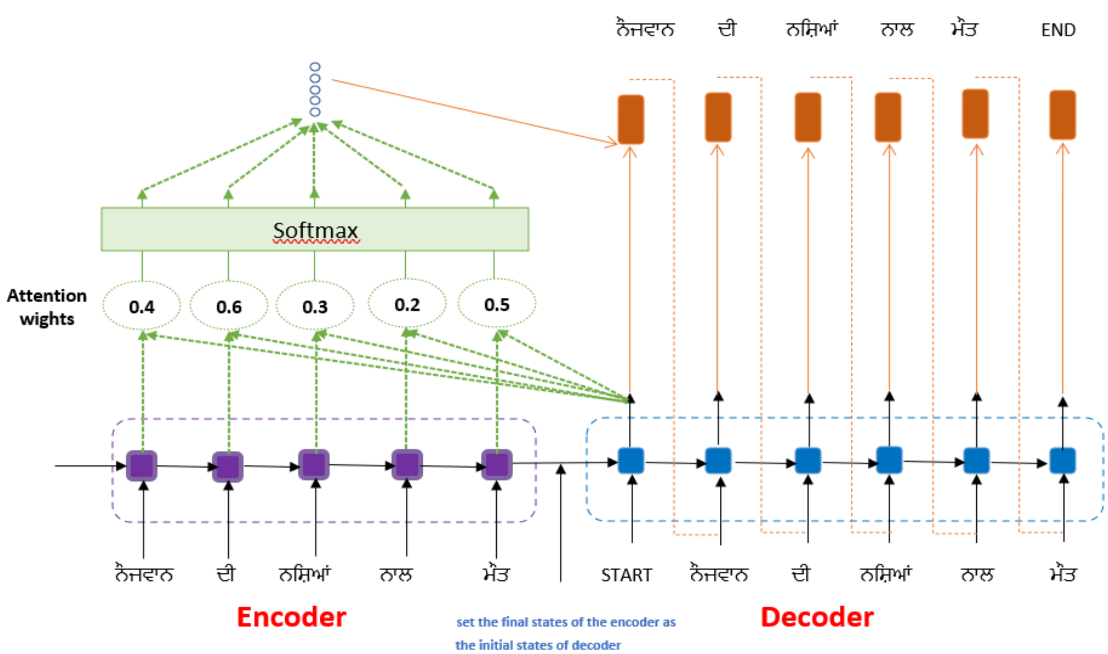

Figure 4: Seq2Seq with Attention Mechanism for Generating Sentence Vectors

being used in this paper for finding similar headlines by paraphrasing.

The dataset News Headlines includes 2,21,278 headlines with 42,498 vocabularies. The headlines with length less than 3 and greater than 20 have been discarded.

Table I: Statistics of Dataset

Dataset	Training	Test	Vocabulary
News Headlines	2,21,278	5000	42,498

V. TRAINING

The RNN based Seq2Seq model is trained with one LSTM layer whereas the dimension of hidden units has been tested with 256 and 512. Two optimizers have been used 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 model is stopped after 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. The details are given in Table II.

For the transformer with augmented encoder, the detail of hyper-parameters is shown in Table III. The dimension of the model and word embeddings are set to 512 whereas 2048 as hidden units. The dropout is 0.1 used. The transformer with augmented encoder is trained using 8 heads, 6 layers of encoder-decoder and batch size is 128. The model is trained on GPU NVIDIA Quadro P4000 for 30 epochs.

Table II: Hyper-parameters of RNN based Seq2Seq to Represent Headlines as Vectors

Hyper-parameter	Value
Batch size	64
Number of epochs	30
Hidden units of LSTM	256, 512
Optimizers	Adam
Loss function	categorical_crossentropy

Table III: Hyper-parameters of Transformer with Augmented Encoder for Paraphrase Generation of Headlines

Hyper-parameter	Value
Batch size	128
Number of epochs	30
Hidden units	2048
Embedding size	512
Dropout	0.1
Loss function	categorical_crossentropy
Optimizers	Adam

VI. EVALUATION

A. Metrics

A well-known evaluation metric BLEU (Papineni et al., 2002) has been used for evaluating the proposed approach for finding similar news. This metric is used to determine whether the detected news headlines are similar to the given headlines. This metric is designed to evaluate machine translation and checks the lexical overlap between given and detected headline.

So, the current approach to find similar headlines is then evaluated by comparing the sentence embeddings. The embeddings of the detected headlines and the given headlines are compared using cosine similarity and combination of cosine and Jaccard similarity. The reason for doing this is that the combination can detect lexical as well as semantic similarity between the news headlines. Again, the approach discussed in section 3.4 has been followed to create embeddings for headlines. This is shown as Seq2Seq Sentence Embeddings Similarity (Seq2SeqSES) in results table.

The qualitative metric is also performed for evaluating the proposed approach to find similar headlines. For this evaluation, three judges familiar with Punjabi language have been selected and assigned them 500 news headlines from the test set. The evaluators are asked to measure the paraphrases on three criteria: similar headline, partial similar headline and dissimilar headline. All the judges are asked to give a number 5 to similar headline, 3 to partial headline and 0 to dissimilar headline. The average results of this metric are shown in Table IV as human evaluations.

B. Automatic Evaluation for Finding Similar Headlines

The qualitative results on news headlines dataset are shown in Tables IV and V. The BLEU scores are calculated which are 31.42 for direct method and 32.28 in case of 512 hidden units with paraphrase generation. The sentence embeddings i.e. Seq2SeqSES has been evaluated using cosine similarity and combination of cosine & Jaccard similarity. Where the combination of cosine and Jaccard produced more accurate results i.e. 87.8% and 88.5% for direct and with paraphrase model respectively. One more comparison is done by using Jaccard similarity which is also not effective as this is just to check lexical similarity between pair of sentences. The results show us that the paraphrase generation task Singh and Josan, (2021) is able to generate accurate paraphrases for the given headline and then those paraphrased headlines further used to find more relevant news. The results shown in Table IV on News Headlines dataset with 256 hidden units where the combination produces better than Jaccard and cosine similarity respectively. The results with 512 hidden units are shown in Table V which is far better than the results with 256 hidden units of LSTM. Here, Jaccard or cosine similarity alone could not produce better than combination of Jaccard & cosine similarity.

Table IV: Performance of the proposed model on News Headlines with 256 Hidden Units

Model	Direct Method	Transformer with Augmented Encoder
BLEU_2	30.35	30.68
Jaccard Similarity	82.2%	83.8%
Cosine Similarity	76.35%	75.32
Seq2SeqSES with cosine and Jaccard Similarity	86.2%	86.8%

Table V: Performance of the proposed model on News Headlines with 512 Hidden Units

Model	Direct Method	Transformer with Augmented Encoder
BLEU_2	31.42	32.28
Jaccard Similarity	84.2%	83.8%
Cosine Similarity	75.33%	75.83%
Seq2SeqSES with cosine and Jaccard Similarity	88.8%	88.5%

C. Human evaluation on paraphrase generation

Table VI shows the scores calculated on Human evaluations for transformer with augmented encoder. As shown the results, the results after paraphrase generation are better than direct method. Here we can see that both of the methods can detect multiple accurate headlines.

Table VI: Human evaluation on News Headlines

Model	Similar Headlines	Partial Similar Headlines	Dissimilar Headlines
Direct Method	87%	6.5%	6.5%
Transformer with Augmented Encoder	88%	7%	5%

D. Examples of Detected Headlines by Paraphrasing

The detected similar headlines are shown in Table VII. The proposed system is detected very accurate headlines and the system can detect more similar news by paraphrasing the given headline. The more focus of the study in this paper on paraphrase generation system i.e. transformer with augmented encoder. To see the effect of paraphrase generation for finding similar news headlines, consider the following table where we can see the similar headlines detected after applying paraphrase generation system.

The proposed approach has shown that the paraphrase generation system works fine for generating accurate paraphrases for the given headlines which are further used for finding similar news headlines. The results shown in Table VII describes that the direct approach as well as by paraphrasing the headlines, we can detect most similar headlines. The advantage of using paraphrase generation is that it improves the results and we able to get more relevant news. For example, the first example in Table VII, the given news is describing about an event at "guruduara sahib" and the similar headlines detected are also related to those events.

The evaluation with different metrics i.e. Jaccard, cosine and combination of cosine and Jaccard similarity shows that the only Jaccard can detect lexically similar headlines but cosine similarity has focused on semantics of headlines. Whereas the combination of cosine and Jaccard is more affective to detect lexically as well as semantically similar headlines.

Table VII: Examples of Similar Headlines Detected by Applying Paraphrase Generation Approach

	Operation	**Input/Output**
1	Input Headline	ਗੁਰਦੁਆਰਾ ਸੰਗਤ ਸਾਹਿਬ ਇਬਰਾਹੀਮਪੁਰ ਮੰਡ ਵਿਖੇ ਸਮਾਗਮ
	Transliterated	Gurduara sangat sahib ibrahimpur mand vikhe samagam
	Similar Headline	ਗੁਰਦੁਆਰਾ ਸੰਗਤ ਸਾਹਿਬ ਇਬਰਾਹੀਮਪੁਰ ਮੰਡ ਚ ਹੋਏ ਸਮਾਗਮ
	Transliterated	Gurduara sangat sahib ibrahimpur mand ch hoe samagam
	Paraphrase of Input Headline	ਗੁਰਦੁਆਰਾ ਗੁਰਸਰ ਸਾਹਿਬ ਵਿਖੇ ਸਾਲਾਨਾ ਗੁਰਮਤਿ ਸਮਾਗਮ ਕਰਵਾਇਆ
	Transliterated	Gurduara gurusar sahib vikhe salana gurmat samagam karvaia
	Similar Headline After Paraphrase	ਗੁਰਦੁਆਰਾ ਗੋਦੜੀ ਸਾਹਿਬ ਚ ਧਾਰਮਿਕ ਸਮਾਗਮ ਕਰਵਾਇਆ
	Transliterated	Gurduara godri sahib ch dharmik samagam karvaia
2	Input Headline	ਹੁਣ 31 ਮਾਰਚ ਤੱਕ ਪ੍ਰਾਪਰਟੀ ਟੈਕਸ ਜਮ੍ਹਾਂ ਕਰਵਾਉਣ ਤੇ 10 ਫ਼ੀਸਦੀ ਛੋਟ ਕਮਿਸ਼ਨਰ
	Transliterated	Hun 31 march takk praparti taks jamhan karvaun te 10 feesdi chhot kamishnr
	Similar Headline	ਅੱਜ ਤੱਕ ਪ੍ਰਾਪਰਟੀ ਟੈਕਸ ਜਮ੍ਹਾਂ ਕਰਵਾਉਣ ਤੇ ਦਿੱਤੀ ਜਾਵੇਗੀ 10 ਫ਼ੀਸਦੀ ਛੋਟ ਕਮਿਸ਼ਨਰ
	Transliterated	Ajj takk praparti taks jamhan karvaun te ditti javegi 10 feesdi chhot kamishnar
	Paraphrase of Input Headline	31 ਮਾਰਚ ਤੱਕ ਪ੍ਰਾਪਰਟੀ ਟੈਕਸ ਜਮ੍ਹਾਂ ਕਰਵਾਉਣ ’ਤੇ 10 ਫ਼ੀਸਦੀ ਛੋਟ
	Transliterated	31 march takk praparti taks jamhan karvaun te 10 feesdi chhot
	Similar Headline After Paraphrase	31 ਮਾਰਚ ਤੱਕ ਪ੍ਰਾਪਰਟੀ ਟੈਕਸ ਜਮ੍ਹਾ ਕਰਵਾਉਣ ‘ਤੇ ਮਿਲੇਗੀ 10 ਫੀਸਦੀ ਛੋਟ
	Transliterated	31 march takk praprti taks jamhan karvaun te milegi 10 feesdi chhot
3	Input Headline	ਸੜਕ ਦੁਰਘਟਨਾ ਵਿਚ ਮਾਂ ਪੁੱਤ ਦੀ ਮੌਤ
	Transliterated	Sadak durghatna vichch maan putt dee maut
	Similar Headline	ਸੜਕ ਹਾਦਸੇ ਵਿੱਚ ਮੋਟਰਸਾਇਕਲ ਸਵਾਰ ਦੀ ਮੌਤ
	Transliterated	Sadak haadse vichch motorcycle sawaar dee maut
	Paraphrase of Input Headline	ਹਾਦਸੇ ਵਿਚ ਮਾਂ ਪੁੱਤ ਦੀ ਮੌਤ
	Transliterated	Haadse vichch maan putt dee muat
	Similar Headline After Paraphrase	ਹਾਦਸੇ ਚ ਜਖਮੀ ਔਰਤ ਦੀ ਮੌਤ
	Transliterated	Haadse ch jakhmi aurt dee maut

VII. CONCLUSION AND FUTURE WORK

The proposed article develops a new approach for finding similar news by applying paraphrase generation in Punjabi language. The approach is also able to rank the detected news headlines. The approach has used current state-of-the-art transformer with augmented encoder (Singh and Josan, 2021) for paraphrasing the headlines. The approach for finding similar news has been done by representing headlines as vectors. The vectors have been generated by using RNN based Seq2Seq. The framework proposed in this article is evaluated on news headlines collected in Singh and Josan (2020). The automatic as well as human judgements have been applied for in-depth evaluations. The BLEU scores have been calculated between detected and input headlines. Another evaluation is done by comparing the sentence embeddings of the detected headlines and input headlines. The proposed approach is effective to find similar headlines and also able to detect multiple headlines.

The current paraphrasing work deal with general applications of NLP such as question-answering, sentence simplification. But the work in this article is applied paraphrase generation in current problem i.e. for finding similar news headlines. This work can be extended in future for online news recommendation systems. The sentence vectors created using RNN based Seq2Seq performed well for detecting semantic and syntactic information but this can be compared by other models such as BERT for generating sentence vectors. The available news headlines dataset can be increased for determining the effect of the proposed approach.

REFERENCES

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

H. Torun and A. B. Inner, "Detecting similar news by summarizing Turkish news," 2018 26th Signal Processing and Communications Applications Conference (SIU), 2018, pp. 1-4,

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.

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.

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.

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.

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.

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.

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.

McKeown, K., R. (1983), “Paraphrasing Questions Using Given and new information” American Journal of Computational Linguistics 9, 1, pp. 1–10.

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.

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.

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.

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.

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.

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.

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.

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.

Singh, A. and Josan, G., S. (2021), "An Augmented Encoder to Generate and Evaluate Paraphrases in Punjabi Language", Turkish Journal of Computer and Mathematics Education, 12(13), pp. 134-151.

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.

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.

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.

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.

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.

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.

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.

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.

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.

***