 doi:10.20944/preprints202104.0630.v1

*Article*

# Unified Model for Paraphrase Generation and Paraphrase Identification

**Divesh Kubal [1,†,‡] and Hemant Palivela [1,‡]**

[1] Research Scholar, Department of Computer Engineering, JJTU; divesh.datascience@gmail.com
[2] Head of AI and Machine Learning, eClerx Services 2; hemant.datascience@gmail.com

**Abstract:** Paraphrase Generation is one of the most important and challenging tasks in the field of Natural Language Generation. The paraphrasing techniques help to identify or to extract/generate phrases/sentences conveying the similar meaning. The paraphrasing task can be bifurcated into two sub-tasks namely, Paraphrase Identification (PI) and Paraphrase Generation (PG). Most of the existing proposed state-of-the-art systems have the potential to solve only one problem at a time. This paper proposes a light-weight unified model that can simultaneously classify whether given pair of sentences are paraphrases of each other and the model can also generate multiple paraphrases given an input sentence. Paraphrase Generation module aims to generate fluent and semantically similar paraphrases and the Paraphrase Identification system aims to classify whether sentences pair are paraphrases of each other or not. The proposed approach uses an amalgamation of data sampling or data variety with a granular fine-tuned Text-To-Text Transfer Transformer (T5) model. This paper proposes a unified approach which aims to solve the problems of Paraphrase Identification and generation by using carefully selected data-points and a fine-tuned T5 model. The highlight of this study is that the same light-weight model trained by keeping the objective of Paraphrase Generation can also be used for solving the Paraphrase Identification task. Hence, the proposed system is light-weight in terms of the model's size along with the data used to train the model which facilitates the quick learning of the model without having to compromise with the results. The proposed system is then evaluated against the popular evaluation metrics like BLEU (BiLingual Evaluation Understudy):, ROUGE (Recall-Oriented Understudy for Gisting Evaluation), METEOR, WER (Word Error Rate), and GLEU (Google-BLEU) for Paraphrase Generation and classification metrics like accuracy, precision, recall and F1-score for Paraphrase Identification system. The proposed model achieves state-of-the-art results on both the tasks of Paraphrase Identification and paraphrase Generation.

**Keywords:** Paraphrase Identification, Paraphrase Generation, Natural Language Generation, Language Model, Encoder Decoder, Transformer

## 1. Introduction

Natural Language Generation (NLG) can be viewed as a task of developing systems that can automatically write summaries, explanations, or narratives in either English or other languages. These NLG systems aim to generate or produce unambiguous and clear natural language just as the way people communicate with each other. Currently, there is a multitude of real-world applications where NLG can be used. These applications can range from chatbots or question answering systems [1–3] for an interactive textual communication or to generate weather reports or to caption images as well as generating human-like summaries [4–6] from research or academic papers, news articles and stories and also in machine translation systems [7].

1. Mr. XYZ wrote a book on Artificial Intelligence
2. A book on Artificial Intelligence was written by Mr. XYZ
3. Mr. XYZ is an author of book related to Artificial Intelligence

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2021 doi:10.20944/preprints202104.0630.v1



These sentences convey almost the same meaning and hence they are the paraphrases of each other even though sentence 1 and 2 depicts that the book is completed but same cannot be said about sentence 3.

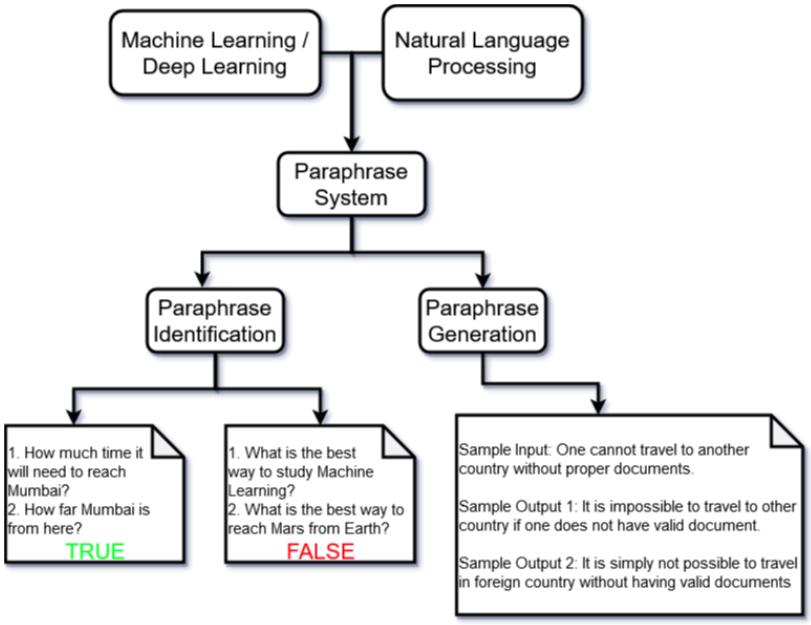


**Figure 1.** Paraphrasing System

Figure 1 depicts the general idea about the paraphrasing system. This paper presents a unified approach to solve both the sub-tasks of Paraphrase Generation and Identification using the same model. There has been a wide array of systems developed to solve the task of Paraphrase Identification and Paraphrase Generation. The Paraphrase Identification task is viewed as a supervised machine learning problem that can be solved by using traditional semantic similarly based techniques and with state-of-the-art deep learning algorithms like Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), etc. The Paraphrase Generation problem can be solved by using simple lexical features and word ordering or restructuring methods or by using templates extracted from WikiAnswers repositories. The most recent advancements in solving the task of Paraphrase Generation tasks involves using Generative Adversarial Networks (GANs), sequence-to-sequence based-models, encoder-decoder based-models, and transformer-based models.

This paper presents a unified system that combines the data selection variety parameter along with a custom fine-tuned T5 model especially for the Paraphrase Generation task which also solves the problem of Paraphrase Identification. An important feature of the T5 model is that it can be trained on multiple tasks simultaneously. The proposed system is trained on both the tasks of Paraphrase Identification and Paraphrase Generation tasks simultaneously and hence a lot of computational time and resources are saved as compared to training multiple models one at a time. This paper is structured by giving a detailed literature survey on Paraphrase Identification and Generation systems. Then, for both the tasks, a mathematical problem formulation is done and the proposed unified system architecture is discussed. This paper gives a thorough results analysis and concludes with a future scope and directions to improve.

## 2. Mathematical Problem Formulation

Paraphrasing can be sub-divided into two tasks namely Paraphrase Identification and Paraphrase Generation. The Paraphrase Identification task can be viewed as a discriminative type of task which tells whether a sentence pair points to the same meaning. In this task, the system might output a probability between 0 and 1 wherein the

value tending to 1 depicts the sentence pair as a paraphrase of each other otherwise not. In some cases, the identification system outputs a semantic score which when normalized can help to discriminate between sentences pair. The Paraphrase Generation task aims to automatically generate one or multiple candidate paraphrases given the reference or input sentence. The aim is to generate semantically same and fluent paraphrases.

### *2.1. Paraphrase Identification (PI) Task*

The PI task is viewed as a supervised machine learning task and is modeled as follows: Given a sentence pair $(S_1, S_2)$, the aim is to find the target (1 or 0 which depicts the given sentence pair is paraphrase of each other or not respectively) where the sentence $S_1 = \{w_1, w_2, w_3, ..., w_n\}$ and $S_2 = \{w_1, w_2, w_3, ..., w_m\}$. It depicts that both the sentences length may vary. The output can be a probability between 0 and 1 or some normalized semantic scoring mechanism.

### *2.2. Paraphrase Generation (PG) Task*

In the PI task, the aim is to generate a candidate sentence given an input sentence. Given an input sentence or a reference sentence $S_1$ where $S_1 = \{w_1, w_2, w_3, ..., w_n\}$, the aim is to generate one or more candidate sentences $S_2 = \{w_1, w_2, w_3, ..., w_m\}$, $S_3 = \{w_1, w_2, w_3, ..., w_o\}$, ... $S_4 = \{w_1, w_2, w_3, ..., w_p\}$. In this task too, the sentence length of the generated candidate sentences and the input or reference sentence may vary.

## 3. Related Work

The paraphrase generation dates back to the year 1983 [8]. [9], [10] and [11] attempted to solve the Paraphrase Generation task by using machine translation whereas [12] and [13] proposed lexical based methods which generates paraphrases by words substitution. In recent times, due to advancements in the field of deep learning, ][14] and [15] proposed neural network based methods.

The Paraphrase Generation techniques can be broadly classified in two major categories as follows:

1. Controlled Paraphrase Generation Methods
   The idea behind this approach is to generate paraphrases controlled by some templates or syntactic trees [16] and [17]. An approach to generate paraphrases was proposed by [18] which uses a mix of syntactic tree and tree encoder using Long Short Term Memory (LSTM) neural network. The main limitation is that it fails when the input dataset is noisy and grammatically not correct. A retriever-editor based approach was proposed by [19] to generate paraphrases. In this approach, a most similar source-target pair is selected by using embedding distance concerning the source. Then the role of the editor is to modify the input sentence by using a transformer. The retriever first selects the most similar source-target pair based on the embedding distance with the source. Then the editor modifies the input accordingly based on the Transformer [20]. The limitation of this model is that it needs to train from scratch even though this model can restructure the sentence and introduce new words or can perform word substitution.
2. Pre-trained Language Models fine-tuning
   Large language models like GPT-2 [21] can be used to generate sentences in paraphrasing tasks. By using GPT-2, [22] and [23] proposed a paraphrase generation approach which exploits the capability of GPT-2 of understanding the language as the GPT-2 is generatively trained on the huge open-domain corpus. This approach aims to fine-tune the weights of the GPT-2 pretrained model. The major limitation is the source-copying in the output is observed. This limitation is taken care of in the proposed system (unified model) in this paper.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2021 doi:10.20944/preprints202104.0630.v1



## 4. Unified System Architecture

This paper proposes a unified system architecture capable of performing both the paraphrasing tasks of identification and generation. The following are the major system components:

1. Data Collection/Aquisition
2. Data Sampling Selection and Preprocessing
3. Text-to-text Transformer Hyperparameter Tuning
4. Text-to-text Transformer Training
5. Evaluation
   (a) Paraphrase Identification
   (b) Paraphrase Generation

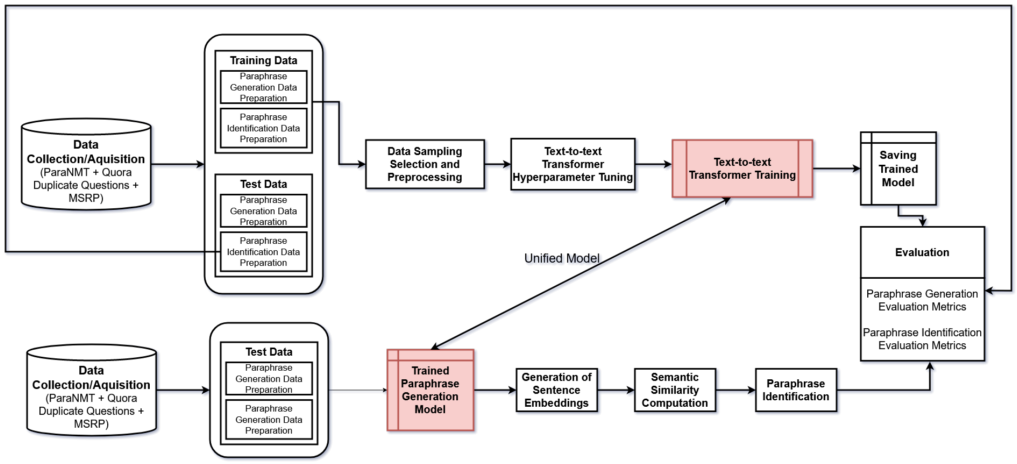


**Figure 2.** Paraphrase Identification Process Flow

### *4.1. Data Collection/Aquisition*

The data is collected from different sources like PARANMT-50M [24], Quora duplicate questions pair [25] and Microsoft Paraphrase Research Database (MSRP) [26]. The ParaNMT database consists of more than 50 million sentential paraphrase pairs in the English language. To generate the huge ParaNMT corpus, a back-translation system was used. A Czech to English Neural Machine Translation (NLT) system was used to extract sentences written in Czech to English. The Quora duplicate questions pair dataset consists of a total of 404290 sentence pairs. This data was split into 70-30 percent training and testing set. For training, the data consists of 283003 sentence pairs and in test data, there are 121287 sentence pairs. The MSRP dataset is filtered from a large sentence pairs database consisting of about 9516684 sentences. These sentences were extracted from news clusters spanning 2 years from World Wide Web (WWW). The final MSRP database consists of around 5800 sentence pairs. The training set consists of 4076 sentence pairs and 1725 sentence pairs in the test set. These three types of data are used to train the paraphrase model.

### *4.2. Data Sampling Selection and Preprocessing*

The objective is to promote data diversity by filtering and sampling the original data. It is observed that the paraphrase generation model outputs proper paraphrases without repetition by maximizing increasing the lexical, semantic, and syntactic diversity in the data used in training. This enables the paraphrasing model to generate more diverse paraphrases with the same meaning but having rich and varied vocabulary. The following transformations are applied to the training data to increase data diversity:

 doi:10.20944/preprints202104.0630.v1



1. Remove the sentence pairs having more than 60% unigram, bi-gram or tri-gram overlap. This discourages the final trained model to copy the input sentence and maximizes the probability of generating diverse paraphrase.
2. Removing the sentence pairs having very less semantic similarly by using Sentence-BERT [27]. This forces the final trained model to generate semantically similar sentences.
3. In Quora and MSRP dataset, selecting only sentence pairs which are labelled as 1. (Here 1 denotes that the sentence pairs are paraphrases of each other)

By performing this step, the three main important parameters of diversity, semantically similar, and fluency are preserved. The diversity is preserved by restricting the copying. The semantically similar parameter is preserved by not allowing the model to train on those sentences which are not semantically similar. The final parameter of fluency is preserved by making sure the generated paraphrase is grammatically correct. This is automatically taken care of because, in all the training data sources, the sentence pairs are grammatically fluent. After applying all these filters, the data size shrunk to approximately 2 lakh sentence-pairs.

### *4.3. Paraphrase Generation Model Building*

The system is trained by keeping the objective of Paraphrase Generation. To train the system on the sentence pairs data, the Text-To-Text Transfer Transformer [28] algorithm is used. In this study, a T5-base pretrained model is used which is then fine-tuned on the task of paraphrase generation. While fine-tuning for paraphrasing task, the hyper-parameters are also fine-tuned by using heuristics. The trained model is then used to generate paraphrases by specifying beam search and nucleus sampling, etc. The same trained model is used for Paraphrase Identification by extract sentence vector representations from the model.

### *4.4. Why Text-To-Text Transfer Transformer?*

Transfer learning has proved to be a very powerful technique in the field of Natural Language Processing (NLP). In transfer learning, an algorithm is first trained on a data-rich task (general/open or closed domain data) and then the trained model is finetuned on another downstream task. The Text-To-Text Transfer Transformer (T5) algorithm aims to convert every language problem into a text-to-text format. T5 is trained on a mix of labeled (Colossal Clean Crawled Corpus) and unlabelled data. The T5 model gives state-of-the-art results on more than 20 NLP tasks. Hardly any technique performs consistently as T5 while preserving flexibility to train on any downstream task. In this paper, the T5-base model is used for fine-tuning and hyperparameter fine-tuning. This version of the T5 model contains approximately 220M parameters having 12-layers, 3072 feed-forwards hidden states, 768-hidden layers, and 12-heads.

### *4.5. Model Architecture Setup*

The T5 model used in this study is trained only for one task that is for text or paraphrase generation. The self-attention technique technique utilized in transformer takes input a sequence of input and generates a sequence of output which is of the same length that of input. In this case, the each element in the output sequence is computed by calculating a weighted average of the input sequence provided. Here, $y^i$ depicts the $i^{th}$ element in the output sequence and similarly $x^j$ depicts the $j^{th}$ element in the input sequence. Futher, each element $y^i$ can be computed as $\sum_j w_{i,j} x_j$. $w_{i,j}$ is the weight which is to be optimized and this is the function of $x^i$ and $x^j$. In this architecture, the encoder takes input the sequence of items (text tokens) and the decoder generates output sequence (text tokens).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2021 doi:10.20944/preprints202104.0630.v1



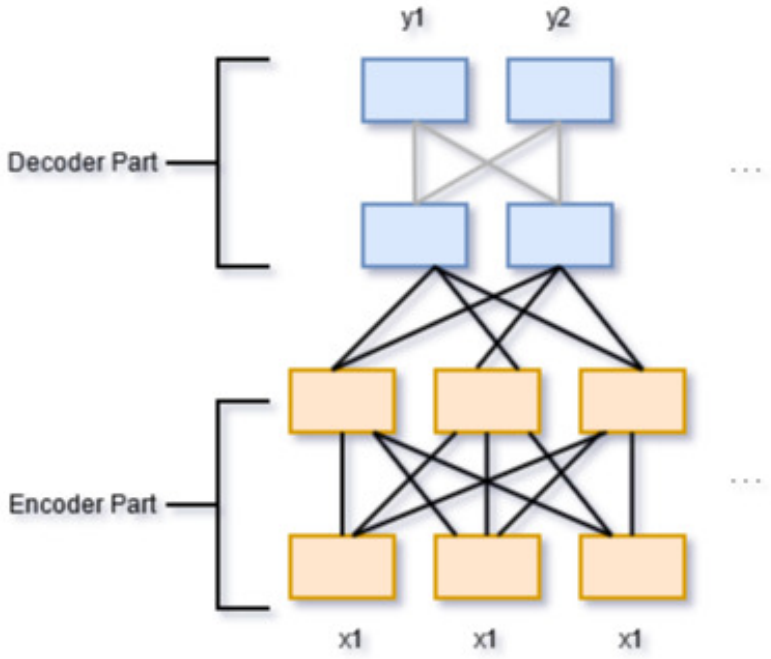


**Figure 3.** Encoder Decoder Model Schematic

The encoder used in this paper and depicted in the figure 3 utilizes a fully-visible attention mask. This type of fully-visible attention masking allows the self attention system to look into any of the entry of input when it simultaneously produces output in sequence by sequence fashion. As this model is used to generate paraphrases, a special prefix is added to the model which tells the model to generate text. Hence, this type of masking is appropriate when there is particular prefix supplied to the model. This prefix can also be said as providing a context to the model. In inference phase, this prefix is again used.

 doi:10.20944/preprints202104.0630.v1



**Table 1.** Final hyperparameters list used to finetune T5 Model for Paraphrase Generation and Paraphrase Identification Task

| Hyperparameter | Value |
|---|---|
| adam_epsilon | 1e-08 |
| best_model_dir | outputs/best_model |
| cache_dir | cache_dir/ |
| cosine_schedule_num_cycles | 0.5 |
| do_lower_case | False |
| dynamic_quantize | False |
| early_stopping_consider_epochs | False |
| early_stopping_metric | eval_loss |
| early_stopping_metric_minimize | True |
| early_stopping_patience | 2 |
| adafactor_eps | (1e-30, 0.001) |
| adafactor_clip_threshold | 1.0 |
| adafactor_decay_rate | -0.8 |
| adafactor_scale_parameter | False |
| adafactor_relative_step | False |
| adafactor_warmup_init | False |
| eval_batch_size | 8 |
| evaluate_during_training | False |
| evaluate_during_training_silent | True |
| evaluate_during_training_steps | 2000 |
| evaluate_during_training_verbose | False |
| evaluate_each_epoch | True |
| fp16 | False |
| gradient_accumulation_steps | 1 |
| learning_rate | 0.004 |
| max_seq_length | 300 |
| multiprocessing_chunksize | 500 |
| n_gpu | 1 |
| no_cache | False |
| no_save | False |
| num_train_epochs | 200 |
| optimizer | Adafactor |
| output_dir | outputs/ |
| overwrite_output_dir | True |
| process_count | 14 |
| polynomial_decay_schedule_lr_end | 1e-07 |
| polynomial_decay_schedule_power | 1.0 |
| reprocess_input_data | True |
| save_steps | 50000 |
| scheduler | constant_schedule_with_warmup |
| skip_special_tokens | True |
| train_batch_size | 64 |
| train_custom_parameters_only | False |
| use_cached_eval_features | False |
| use_early_stopping | False |
| use_multiprocessing | False |
| warmup_ratio | 0.06 |
| model_class | T5Model |
| do_sample | False |
| early_stopping | True |

 doi:10.20944/preprints202104.0630.v1



### *4.6. Hyperparameter Tuning of T5 Model*

It is not possible to fine-tune the T5 model by using the grid-search technique due to a high number of parameters and the model size. Hence, heuristic-based hyperparameter tuning was performed. The important parameters which were finetuned are learning rate, maximum sequence length, training batch size, and number of training epochs. The table 1 depicts the hyperparameters used to train the T5 paraphrase generation model.

### *4.7. Paraphrase Identification*

The main highlight of this paper is using the same model for Paraphrase Generation and Paraphrase Identification task. The same hyperparameter fine-tuned T5 model is utilized to solve the problem of Paraphrase Identification. The aim is to extract sentence vectors from sentence pairs and then computing semantic similarity between sentence pairs. Figure 2 depicts the Paraphrase Identification workflow. The figure. The semantic similarity between sentence pairs is computed as follows:

1. Initially, load the trained T5 model on paraphrase Generation task.
2. Tokenize the sentence pair and extract token ids and then add padded tokens to it.
3. Extraction of attention mask and segment tokens.
4. With the help of token ids, attention mask, and segment token, compute the sentence vectors. In this step, we need to compute the pooled sentence representations which represent the embeddings for each word in the sentence.

### *4.8. Training Time and System Configuration*

The entire model was trained for 200 epochs on a system with 120GB of RAM. The GPU used was A-100-PCIE having RAM of 40GB. The algorithm took 74 hours to train on Paraphrase Generation Task. The proposed system is lightweight and efficient and it can also be deployed into an actual production environment. By optimizing the parameters in beam search and sampling parameters, the performance can be further improved during the generation phase. Currently, the system can utilize multiple GPU's for multiple paraphrase generation.

## 5. Evaluation Metrics

In this paper, separate evaluation metrics are used for Paraphrase Generation and Paraphrase Identification task. The following are the metrics used for the Paraphrase Identification task:

1. Accuracy:
   Before going into accuracy, precision, and F1-score, the following terminologies are important concerning the Paraphrase Identification task.
   - True Positives (TP): Model predicts sentence-pair is a paraphrase of each other and in reality, it is so.
   - True Negatives (TN): Model predicts sentence-pair is not a paraphrase of each other and in reality too, they are not paraphrases of each other.
   - False Positives (FP): Model predicts sentence-pair is a paraphrase of each other, but they are not paraphrases of each other (Type I error)
   - False Negatives (FN): Model predicts sentence-pair is not a paraphrase of each other, but they are paraphrases of each other. (Type II error)

   Accuracy is the fraction of predictions our model got right. Here, the accuracy can be summarized as (TP+TN)/total.
2. Precision: Precision is the value denoting by how often the model is correct if the model predicts that particular sentence-pairs are paraphrases of each other. Precision can be denoted by using: TP/predicted(Yes)
3. Recall: Recall tells when sentence-pairs are actually paraphrases of each other, how often does the model predict Yes. Recall can also be called sensitivity or True Positive rate. It is denoted as TP/actual(Yes).

4. F1-score:
   F1-score is the harmonic mean of precision and recall (True Positive Rate).

The Paraphrase Generation task is evaluated by using the following metrics:

1. ROUGE (Recall-Oriented Understudy for Gisting Evaluation):
   ROUGE is based only on recall and it is one of the most common metrics used for summarization tasks. But it can also be used to evaluate paraphrases. It has various types like ROUGE-1, ROUGE-2, ROUGE-N, and ROUGE-(L/W/S) depending on the feature. ROUGE-N is based on the number of grams. If unigram is set, then ROUGE-1 computes the recall by analyzing the matching unigrams. ROUGE-L/W/S denotes the ROUGE on Longest Common Subsequence (LCS), Weighted LCS, and skip-bigram co-occurrence statistics respectively. In this paper, ROUGE-1, ROUGE-2, and ROUGE-L are used.
2. BLEU (BiLingual Evaluation Understudy):
   BLEU counts the matching N-grams in the generated/candidate translation to N-grams in the gold or reference text. Here the unigram is token-wise and the bi-gram is word pair. To penalize a generated translation or paraphrase which generates a lot of reasonable words, the n-gram counting is modified. In this paper, BLUE-1/2/3/4 is used for evaluation.
3. GLEU (Google-BLEU):
   GLEU is a variant of BLEU score and it is aimed towards more evaluation close to human judgments. GLUE overcomes the BLUE's drawback of per sentence reward objective. GLUE works by computing n-gram precisions over the gold/truth/reference paraphrases but here the more weight is assigned to N-grams which have been changed from the source.
4. WER (Word Error Rate)
   WER is one of the most common metrics used in Automatic Speech Recognition (ASR) but can also be used to evaluate Paraphrase Generation. The WER can be summarized by, Word Error Rate = (Substitutions + Insertions + Deletions) / Number of Words Spoken. Where substitutions denote a replacement of a word, insertion denotes a word is added and deletions identify the words which are deleted.
5. METEOR (Metric for Evaluation of Translation with Explicit ORdering)
   METEOR works by modifying the precision and recall computations. It replaces them with a weighted F-Score. This F-score is based on mapping 1-grams or unigrams along with a penalty function whenever an incorrect word order is encountered.

## 6. Results and Comparative Analysis

The Paraphrase Generation and Paraphrase Identification are evaluated on the same model but by using different evaluation metrics as the former is a generation task and the latter is viewed as a classification task. The table 2 depicts the evaluation results for the Paraphrase Generation task. The evaluation is carried on three test datasets of ParaNMT, MSRP, and Quora on evaluation metrics of ROUGE-1, ROUGE-2, ROUGE-L, METEOR, BLEU, BLEU-1, BLEU-2, BLEU-3, BLEU-4, GLEU, and WER. The table 3 represents the evaluation results for the Paraphrase Identification task. The evaluation for the Paraphrase Identification task was carried on MSRP and Quora datasets by using evaluation metrics of accuracy, precision, recall, and F1-score. To evaluate the PI task, a threshold of 0.726 was set. This threshold denotes that the sentence-pair will be assigned as 1(are paraphrases of each other) if the semantic score(computed from the unified model) is greater than the threshold (0.726) else the prediction will be assigned as 0. After trial and error, this threshold provided a good balance in precision, recall, and f1-score for both the classes for both test datasets. It can be seen that the unified trained model worked pretty well in both the tasks of Paraphrase Generation and Identification.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2021 doi:10.20944/preprints202104.0630.v1



**Table 2.** Evaluation results for Paraphrase Generation Task

| Evaluation Metrics | Test Datasets | | |
|---|---|---|---|
| | **ParaNMT** | **MSRP** | **Quora** |
| ROUGE-1 Precision | 0.523674138 | 0.635033051 | 0.639637286 |
| ROUGE-1 Recall | 0.544363435 | 0.506638129 | 0.629653077 |
| ROUGE-1 F1-Score | 0.525774226 | 0.552428414 | 0.628682324 |
| ROUGE-2 Precision | 0.357972449 | 0.482774133 | 0.544849372 |
| ROUGE-2 Recall | 0.370601416 | 0.38672292 | 0.541250036 |
| ROUGE-2 F1-Score | 0.359194076 | 0.420419585 | 0.54093177 |
| ROUGE-3 Precision | 0.507414232 | 0.615410955 | 0.634670867 |
| ROUGE-3 Recall | 0.517249065 | 0.495184851 | 0.625172433 |
| ROUGE-3 F1-Score | 0.506064072 | 0.538796863 | 0.624795039 |
| METEOR | 0.480186424 | 0.497118907 | 0.614391206 |
| BLEU | 0.29330398 | 0.32358232 | 0.503377545 |
| BLEU-1 | 0.54548532 | 0.483069097 | 0.642862228 |
| BLEU-2 | 3.70E-01 | 0.371630754 | 0.543135925 |
| BLEU-3 | 3.01E-01 | 0.316771713 | 0.508984305 |
| BLEU-4 | 0.266415057 | 0.281732748 | 0.493364226 |
| GLEU | 0.58188349 | 0.574202108 | 0.674050573 |
| WER | 0.726122861 | 0.615041374 | 0.683562504 |

**Table 3.** Evaluation Results for Paraphrase Identification Task

| **Dataset (threshold = 0.726)** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|---|
| MSRP | 82.0513 | 73.6842 | 87.5 | 79.9999 |
| Quora | 87.17948 | 78.9473 | 93.75 | 85.71428 |

**Table 4.** Comparative Analysis for Paraphrase Generation for Quora Dataset

| **Work by** | **ROUGE-1** | **ROUGE-2** | **BLEU** | **METEOR** |
|---|---|---|---|---|
| Seq2Seq [29] | 58.77 | 31.47 | 36.55 | 26.28 |
| Residual LSTM [29] | 59.21 | 32.43 | 37.38 | 28.17 |
| VAE-SVG-eq [14] | - | - | - | 25.5 |
| Pointer-generator [29] | 61.93 | 36.07 | 40.55 | 30.21 |
| RL-ROUGE [29] | 63.35 | 37.33 | 41.83 | 30.21 |
| RbM-SL [29] | 64.39 | 38.11 | 43.54 | 32.84 |
| RbM-IRLTRANSEQ+beam (size=6) [30] | - | - | 40.36 | 38.49 |
| RbM-IRL [29] | 64.02 | 37.72 | 43.09 | 31.97 |
| TRANSEQ+beam (size=6) [30] | - | - | 40.36 | 38.49 |
| **Unified Approach (ours)** | **62.8682** | **54.0932** | **50.3378** | **61.4391** |

**Table 5.** Comparative Analysis for Paraphrase Generation for MSRP Dataset

| **Method** | **BLEU** |
|---|---|
| Transfer Learning [31] | 12.91 ParaSCI [32] |
| 27.18 **Unified Approach (proposed)** | **32.35** |

 doi:10.20944/preprints202104.0630.v1



**Table 6.** Comparative Analysis for Paraphrase Identification for MSRP Dataset

| Work by | Accuracy (in percentage) |
|---|---|
| **Accuracy (in percentage)** | |
| Mihalcea et al. [33] | 65.4 |
| Kozareva and Montoyo [34] | 76.6 |
| Hassan [35] | 68.8 |
| Hu et al. ARC-I [36] | 69.6 |
| Hu et al. ARC-II [36] | 69.9 |
| Rus et al. [37] | 70.6 |
| Islam and Inkpen [38] | 72.6 |
| Yin et al. [39] | 72.5 |
| Fernando and Stevenson [40] | 74.1 |
| Wan et al. [41] | 75.6 |
| Pang et al. [42] | 75.94 |
| Socher et al. [43] | 76.8 |
| Madnani et al. [44] | 77.4 |
| Zhang et al. [45] | 77.5 |
| Word2vector + Hybrid Deep Learning [46] | 77.66 |
| GloVe + Hybrid Deep Learning [46] | 78.49 |
| Context2vec + Hybrid Deep Learning [46] | 79.88 |
| DeepPairwiseWord [47] | 83.4 |
| **Unified Approach (Proposed)** | **82.0513** |

**Table 7.** Comparative Analysis for Paraphrase Identification for Quora Dataset

| Work by | Accuracy (in percentage) |
|---|---|
| TF-IDF CatBoost [48] | 75.39 |
| pt-DECATT [49] | 88.4 |
| BERT [50] | 83.5 |
| XLNet [50] | 86 |
| RoBERTa [50] | 88.6 |
| ALBERT [50] | 86.7 |
| **Unified Approach (Proposed)** | **87.17948** |

Further, a comparative analysis was performed for both the tasks with existing available systems for different datasets. The table 4 and table 5 represents the comparative analysis performed for the Paraphrase Generation task for Quora and MSRP datasets respectively. It can be observed that the proposed system achieved higher scores as compared to existing systems. The proposed system achieved a ROUGE-1 score of 0.628, ROUGE2-score of 0.54, BLEU score of 0.5037, and METEOR score of 0.6143. The RbM-IRL [29] from table 4 achieved the best ROUGE-1 score of 64.02 but the proposed system surpassed it and the remaining systems in other metrics by a huge margin.

The Paraphrase Identification system is compared with the existing systems from the year 2006. The tables 6 and 7 depicts the comparisons based on accuracy for the task of the Paraphrase Identification system. In table 5, it can be seen that the proposed system achieves an accuracy of 82.0513% which surpasses the other existing systems except for the Deep Pairwise Word [47] which achieves 83.4% accuracy. The unified model is trained by keeping the objective of generation but still, the proposed approach computes a respectable accuracy level.

## 7. Conclusion and Future Scope

The paraphrasing tasks of Paraphrase Generation and Identification are the most important in the field of Natural Language Processing. Until now, both tasks can be

solved by using different individual approaches. The Paraphrase Generation can be solved by using conditional generation based neural networks and the task of Paraphrase Identification can be solved by viewing it as a sentence-pair binary classification task. This paper proposed a model that can perform both tasks by training a single model with the objective of generation. The final trained model achieved state-of-the-art results on both tasks while surpassing the major existing systems as well in their corresponding evaluation metrics. The generated paraphrases are not only grammatically fluent but there is negligible copying from the input sentence. The data sampling selection phase proved to be effective to avoid the output getting copied from the input. The proposed system further became strong by proper hyper-parameter tuning and the strengths of the T5 model for the text-to-text task of paraphrase generation. Further, due to the efficient and accurate sentential embeddings generation from the trained model and proper threshold value, the Paraphrase Identification task achieved respectable results. The proposed approach is designed in such a way that the end-to-end system can be used in a real-time production environment as the system takes advantage of multi-GPU training and parallel evaluations. The final model is evaluated on both the tasks for different test databases and then compared the individual tasks on independent test datasets with the existing system to give an idea about the performance of the proposed system. In this paper, the T5-base model is finetuned but it can be extended to T5-large and other variants too.

## References


1. Voorhees, E.M. The TREC question answering track. *Natural Language Engineering* **2001**, *7*, 361.
2. Harabagiu, S.M.; Maiorano, S.J.; Pasca, M.A. Open-domain textual question answering techniques. *Natural Language Engineering* **2003**, *9*, 231.
3. Mollá, D.; Vicedo, J.L. Question answering in restricted domains: An overview. *Computational Linguistics* **2007**, *33*, 41–61.
4. Cohn, T.; Lapata, M. Sentence compression beyond word deletion. Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), 2008, pp. 137–144.
5. Cohn, T.A.; Lapata, M. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research* **2009**, *34*, 637–674.
6. Galanis, D.; Androutsopoulos, I. An extractive supervised two-stage method for sentence compression. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010, pp. 885–893.
7. Koehn, P. *Statistical machine translation*; Cambridge University Press, 2009.
8. McKeown, K. Paraphrasing questions using given and new information. *American Journal of Computational Linguistics* **1983**, *9*, 1–10.
9. Quirk, C.; Brockett, C.; Dolan, W.B. Monolingual machine translation for paraphrase generation. Proceedings of the 2004 conference on empirical methods in natural language processing, 2004, pp. 142–149.
10. Zhao, S.; Niu, C.; Zhou, M.; Liu, T.; Li, S. Combining multiple resources to improve SMT-based paraphrasing model. Proceedings of ACL-08: HLT, 2008, pp. 1021–1029.
11. Wubben, S.; Van Den Bosch, A.; Krahmer, E. Paraphrase generation as monolingual translation: Data and evaluation. Proceedings of the 6th International Natural Language Generation Conference, 2010.
12. Bolshakov, I.A.; Gelbukh, A. Synonymous paraphrasing using wordnet and internet. International Conference on Application of Natural Language to Information Systems. Springer, 2004, pp. 312–323.
13. Kauchak, D.; Barzilay, R. Paraphrasing for automatic evaluation. Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, 2006, pp. 455–462.
14. Gupta, A.; Agarwal, A.; Singh, P.; Rai, P. A deep generative framework for paraphrase generation. Proceedings of the AAAI Conference on Artificial Intelligence, 2018, Vol. 32.
15. Fu, Y.; Feng, Y.; Cunningham, J.P. Paraphrase generation with latent bag of words. *arXiv preprint arXiv:2001.01941* **2020**.
16. Iyyer, M.; Wieting, J.; Gimpel, K.; Zettlemoyer, L. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059* **2018**.

17. Chen, M.; Tang, Q.; Wiseman, S.; Gimpel, K. Controllable paraphrase generation with a syntactic exemplar. *arXiv preprint arXiv:1906.00565* **2019**.
18. Kumar, A.; Ahuja, K.; Vadapalli, R.; Talukdar, P. Syntax-Guided Controlled Generation of Paraphrases. *Transactions of the Association for Computational Linguistics* **2020**, *8*, 330–345.
19. Kazemnejad, A.; Salehi, M.; Baghshah, M.S. Paraphrase generation by learning how to edit from samples. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 6010–6021.
20. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv preprint arXiv:1706.03762* **2017**.
21. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI blog* **2019**, *1*, 9.
22. Witteveen, S.; Andrews, M. Paraphrasing with large language models. *arXiv preprint arXiv:1911.09661* **2019**.
23. Hegde, C.; Patil, S. Unsupervised paraphrase generation using pre-trained language models. *arXiv preprint arXiv:2006.05477* **2020**.
24. Wieting, J.; Gimpel, K. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *arXiv preprint arXiv:1711.05732* **2017**.
25. Ansari, N.; Sharma, R. Identifying semantically duplicate questions using data science approach: a quora case study. *arXiv preprint arXiv:2004.11694* **2020**.
26. Dolan, W.B.; Brockett, C. Automatically constructing a corpus of sentential paraphrases. Proceedings of the Third International Workshop on Paraphrasing (IWP2005), 2005.
27. Reimers, N.; Gurevych, I. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813* **2020**.
28. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* **2019**.
29. Li, Z.; Jiang, X.; Shang, L.; Li, H. Paraphrase generation with deep reinforcement learning. *arXiv preprint arXiv:1711.00279* **2017**.
30. Egonmwan, E.; Chali, Y. Transformer and seq2seq model for Paraphrase Generation. Proceedings of the 3rd Workshop on Neural Generation and Translation, 2019, pp. 249–255.
31. Brad, F.; Rebedea, T. Neural paraphrase generation using transfer learning. Proceedings of the 10th International Conference on Natural Language Generation, 2017, pp. 257–261.
32. Dong, Q.; Wan, X.; Cao, Y. ParaSCI: A Large Scientific Paraphrase Dataset for Longer Paraphrase Generation. *arXiv preprint arXiv:2101.08382* **2021**.
33. Mihalcea, R.; Corley, C.; Strapparava, C.; others. Corpus-based and knowledge-based measures of text semantic similarity. Aaai, 2006, Vol. 6, pp. 775–780.
34. Kozareva, Z.; Montoyo, A. Paraphrase identification on the basis of supervised machine learning techniques. International conference on natural language processing (in Finland). Springer, 2006, pp. 524–533.
35. Hassan, S. *Measuring semantic relatedness using salient encyclopedic concepts*; University of North Texas, 2011.
36. Hu, B.; Lu, Z.; Li, H.; Chen, Q. Convolutional neural network architectures for matching natural language sentences. *arXiv preprint arXiv:1503.03244* **2015**.
37. Rus, V.; McCarthy, P.M.; Lintean, M.C.; McNamara, D.S.; Graesser, A.C. Paraphrase Identification with Lexico-Syntactic Graph Subsumption. FLAIRS conference, 2008, pp. 201–206.
38. Islam, A.; Inkpen, D. Semantic similarity of short texts. *Recent Advances in Natural Language Processing V* **2009**, *309*, 227–236.
39. Yin, W.; Schütze, H. Convolutional neural network for paraphrase identification. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2015, pp. 901–911.
40. Fernando, S.; Stevenson, M. A semantic similarity approach to paraphrase detection. Proceedings of the 11th annual research colloquium of the UK special interest group for computational linguistics, 2008, pp. 45–52.
41. Wan, S.; Dras, M.; Dale, R.; Paris, C. Using dependency-based features to take the'para-farce'out of paraphrase. Proceedings of the Australasian language technology workshop 2006, 2006, pp. 131–138.
42. Pang, L.; Lan, Y.; Guo, J.; Xu, J.; Wan, S.; Cheng, X. Text matching as image recognition. Proceedings of the AAAI Conference on Artificial Intelligence, 2016, Vol. 30.

43. Socher, R.; Huang, E.H.; Pennington, J.; Ng, A.Y.; Manning, C.D. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. NIPS, 2011, Vol. 24, pp. 801–809.
44. Madnani, N.; Tetreault, J.; Chodorow, M. Re-examining machine translation metrics for paraphrase identification. Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2012, pp. 182–190.
45. Zhang, X.; Rong, W.; Liu, J.; Tian, C.; Xiong, Z. Convolution neural network based syntactic and semantic aware paraphrase identification. 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, 2017, pp. 2158–2163.
46. Kubal, D.R.; Nimkar, A.V. A hybrid deep learning architecture for paraphrase identification. 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2018, pp. 1–6.
47. Lan, W.; Qiu, S.; He, H.; Xu, W. A continuously growing dataset of sentential paraphrases. *arXiv preprint arXiv:1708.00391* **2017**.
48. Chandra, A.; Stefanus, R. Experiments on Paraphrase Identification Using Quora Question Pairs Dataset. *arXiv preprint arXiv:2006.02648* **2020**.
49. Tomar, G.S.; Duque, T.; Täckström, O.; Uszkoreit, J.; Das, D. Neural paraphrase identification of questions with noisy pretraining. *arXiv preprint arXiv:1704.04565* **2017**.
50. Corbeil, J.P.; Ghadivel, H.A. BET: A Backtranslation Approach for Easy Data Augmentation in Transformer-based Paraphrase Identification Context. *arXiv preprint arXiv:2009.12452* **2020**.