

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



# Medical Expertise Style Transfer using Denoising Autoencoder

## Authors

Mohammad Sabik Irbaz - 160041004

Abir Azad - 160041024

Anika Tasnim Preoty - 160041044

Tani Barkat Shalanyuy - 160041083

## Supervisor

Md. Kamrul Hasan, Ph.D.

Professor, Department of CSE,  
System and Software Lab (SSL),  
Islamic University of Technology (IUT)

*A thesis submitted in partial fulfilment of the requirements for the degree of  
B. Sc. Engineering in Computer Science and Engineering (CSE)*

**Academic Year: 2019-2020**

Department of Computer Science and Engineering (CSE)  
Islamic University of Technology (IUT),  
A Subsidiary Organ of the Organization of Islamic Cooperation (OIC)  
Gazipur-1704, Dhaka, Bangladesh

---

## Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Mohammad Sabik Irbaz, Abir Azad, Anika Tasnim Preoty and Tani Barkat Shalanyuy under the supervision of Md. Kamrul Hasan, Professor of the Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

*Authors:*

---

Mohammad Sabik Irbaz  
Student ID - 160041004

---

Abir Azad  
Student ID - 160041024

---

Anika Tasnim Preoty  
Student ID - 160041044

---

Tani Barkat Shalanyuy  
Student ID - 160041083

Approved By

Supervisor:

---

Md. Kamrul Hasan, Ph.D.  
Professor, Department of CSE,  
System and Software Lab (SSL),  
Islamic University of Technology (IUT)

---

## Abstract

Due to the huge cognitive bias and the curse of knowledge, there is a notable communication gap between experts and laymen. This communication gap creates a huge problem in the medical domain. The patients do not understand what the doctors (domain expert) are saying and the doctors also face some ambiguity issues since they are not used to the laymen style. Bridging the gap between laymen and experts is a challenging task as it requires the models to have expert intelligence in order to modify text with a deep understanding of domain knowledge and structures. To bridge the gap between doctors and patients, we proposed a new approach of text style transfer for non-parallel data. Our proposed approach is based on masking expert terms and denoising autoencoder. We trained and tested our approach on MSD dataset and achieved a stable score in BLEU, perplexity, and style accuracy metrics.

## Keywords

Text Style Transfer, Transformer, Deep Learning, Denoising Autoencoder, BERT

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Overview . . . . .	6
1.2	The Problem . . . . .	6
1.3	Motivation . . . . .	7
1.4	Innovative Aspects . . . . .	8
1.5	Research Challenges . . . . .	8
1.5.1	Limited dataset . . . . .	8
1.5.2	Availability of hardware like GPU . . . . .	9
1.5.3	Unstable Evaluation Metrics . . . . .	9
1.6	Thesis Outline . . . . .	9
<b>2</b>	<b>The Problem</b>	<b>10</b>
<b>3</b>	<b>Related Works</b>	<b>11</b>
3.1	Delete and Retrieve . . . . .	11
3.2	Dual Reinforcement Learning . . . . .	13
3.2.1	Reward for changing style . . . . .	14
3.2.2	Reward for preserving content . . . . .	14
3.2.3	Overall reward . . . . .	14
3.3	Controlled Generation . . . . .	14
3.4	Shortcomings of Existing models . . . . .	15
<b>4</b>	<b>Background Study</b>	<b>16</b>
4.1	Word Embeddings . . . . .	16
4.2	Seq2Seq . . . . .	17
4.3	Attention Mechanism . . . . .	18
4.4	Transformer . . . . .	19
4.5	BERT . . . . .	20
4.6	RoBERTa . . . . .	21
4.7	Autoencoder . . . . .	22
4.8	Under / over-complete hidden layer . . . . .	23
4.9	Denoising Autoencoder . . . . .	24
<b>5</b>	<b>Proposed Method</b>	<b>25</b>
5.1	Expertise Classifier . . . . .	26
5.2	Masking Expert Terms . . . . .	26
5.3	Denoising Autoencoder . . . . .	27
5.4	Layman Text Generator . . . . .	27
<b>6</b>	<b>Experimental Results</b>	<b>28</b>
6.1	Dataset . . . . .	28
6.2	Experimental Setup . . . . .	28
6.3	Training . . . . .	29
6.4	Evaluation Metrics . . . . .	29
6.4.1	Style Accuracy . . . . .	29
6.4.2	Perplexity . . . . .	29
6.4.3	BLEU . . . . .	30
6.5	Result Analysis . . . . .	30

**List of Figures**

1	Examples of Medical Expertise Text Style Trasnsfer . . . . .	7
2	An Overview for Expertise Style Transfer . . . . .	10
3	Delete & Retrieve . . . . .	12
4	Dual Reinforcement Learning . . . . .	13
5	Controlled Generation . . . . .	15
6	Encoder Decoder Model . . . . .	16
7	Seq2Seq Without Attention Mechanism . . . . .	17
8	Seq2Seq with Attention Mechanism . . . . .	18
9	Multihead Attention . . . . .	19
10	Transformer Architecture . . . . .	20
11	BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings. . . . .	20
12	The two steps in BERT architecture in (Devlin et al., 2018) . . . . .	21
13	Autoencoder . . . . .	22
14	Under / over-complete hidden layer . . . . .	23
15	Denoising Autoencoder . . . . .	24
16	Proposed Model Architecture . . . . .	25
17	Masking Expert Terms . . . . .	26

**List of Tables**

1	Example of MSD Dataset . . . . .	28
2	Required time analysis . . . . .	29
3	Geneated Layman Text Examples by our approach . . . . .	30
4	Result Analysis . . . . .	30

---

# 1 Introduction

## 1.1 Overview

Several approaches and strategies have been taken to instill a processing and understanding power of human language to Computers. As years have progressed these investments have successfully inculcated a large percentage of this ability into computers. As such bringing a lot of improvements in our daily lives and interactions with computers.

A few years ago people could not communicate with machines or ask them to perform a particular task. Today with chatbots such as ALEXA, SIRI, Google ASSISTANT, CORTANA people are able to converse with devices in real-time. For individuals to be able to establish a conversation and a machine responds, the machine should have acquired machine learning capabilities, natural language generation and processing skills.

Natural Language Generation and Natural Language Processing is a subset of methods of Artificial Intelligence which manipulates human language, tries to interpret and understand it as well as slendering the communication opening between Humans and Computers. Natural language Processing can be used in several domains such as health, medical , and business domains. A wide range of technologies and tasks use NLP. Examples include; Question Answering, text summarization, Machine Translation, Sentiment Analysis, Auto Correct, text classification.

Text style transfer is a contemporary and pertinent text-to-text generation task under NLG. Text style transfer mainly studies how text can adapt to different situations, audience, purpose by making some adjustments in text such as grammar, emotion,tone, fluency and complexity. The task of expertise style transfer is that of converting expert text to layman language.

This statement can better be expantiated with an example, A fresher(layman) who just got admitted into university might find it difficult to understand the lectures from the professor(Expert—) as he or she uses jargons. In this domain expertise style transfer tries to breakdown the expert's level that is the professor to a layman that is the student such that the lecture is well understood. Another example could be in a medical milieu whereby a doctor(Expert) gives a prescription to a patient(Layman) and he or she does not understand.

When the patient is unable to construe what the doctor is saying this is known as the Curse of knowledge. The curse of knowledge arises in a situation whereby the expression of intentions and knowledge is unfulfilled. We exploit the task of Expertise style transfer in the medical domain.

## 1.2 The Problem

The curse of knowledge could have grave effects. A doctor examining a patient might not know what exact words he should use to make his patients understand the medical science behind his condition. The patient, on the other hand, might crave to know what's wrong with his body in a way that he can understand and be able to take actions which could remedy his ill health situation.

The scourge of information (Camerer et al., 1989) is an unavoidable intellectual predisposition displayed across all areas, prompting errors between a specialist's recommendation and a layman's comprehension of it (Tan and Goonawardene, 2017). Zeroing in on the clinical field, during meetings, patients (laymen) think that its hard to comprehend the specialist's (master) language in this manner making a gigantic correspondence hole. A solution for such a circumstance will be for specialists to precisely uncover a patient's careful affliction in straightforward terms. Misinterpretations in such a situation prompts disappointment in conclusion, delayance in treatment and in the most pessimistic scenario passing. Consequently changing the aptitude level of writings is along these lines basic for successful correspondence between the two players.

Many cause **dyspnea**, pleuritic chest pain, or both.

The most common symptoms, regardless of the type of fluid in the pleural space or its cause, are **shortness of breath** and chest pain.

The **lesion** slowly enlarges, often **ulcerates**, and spread to other skin areas. Lesions heal slowly, with scarring.

The **sores** slowly enlarge and spread to nearby tissue, **causing further damage**. Sores heal slowly and may result in permanent scarring.

**About 1/1000** hypertensive patients has a pheochromocytoma.

The incidence of Pheochromocytomas may be **quite small**.

In patients with papilledema, vision is usually not affected initially, but seconds-long graying out of vision, flickering, or blurred or double vision may occur.

At first, papilledema may be present without affecting vision.

Fleeting vision changes (blurred vision, double vision, flickering, or complete loss of vision) typically lasting seconds are characteristic of papilledema.

Figure 1: Examples of Medical Expertise Text Style Trasnsfer

### 1.3 Motivation

Expertise style transfer aims at improving the readability of a text by reducing the complication level, such as explaining the complex terminology using words from a layman's vocabulary. More so it also aims to improve the expertise level based on context, meaning and the informational content, so that laymen's expressions can be detailed, explicit, faultless and professional.

---

When a doctor speaks the language is obscure and the patient becomes puzzled as to the meaning of what is being said to him. Increasing readability of a text by reducing the expertise level, such as breaking down complex medical terms and sentences is consequently a primordial task to bridge the discrepancies between a doctor and a patient. Building a model to solve this task requires parallel data that is a doctor's prescription alongside a patient's explanation for proper understanding. Solving this task not only simplifies the professional language, but also improves the accuracy and expertise level of laymen descriptions using simple words. This task also requires the models to have expert intelligence in order to modify text with a deep understanding of domain knowledge and structures.

Thus tackling the problem of discrepancies between an expert's advice and a layman's understanding of it became the paramount motivation towards working on an expertise style transfer system in natural language processing.

On one hand, expertise style transfer aims at improving the readability of a text by reducing the expertise level, such as explaining the complex terminology. On the other hand, it also aims to improve the expertise level based on context, so that laymen's expressions can be more accurate and professional.

## **1.4 Innovative Aspects**

Various works have been performed on Text style transfer such as Xu et al. that tried to apply a phrase-based machine translated with a parallel corpus. Gatys et al. explored the use of convolutional neural networks(CNN) to extract content and style features from images separately but this could not apply to text since disentangling content from style features was not possible with CNN.

Fu et al. proposed two TST models, which adopted an adversarial learning approach to implicitly disentangle the content and style in text. The first method used multiple decoders for each type of style to generate text of different styles from a common content embedding. Meanwhile in the second approach, style embeddings are learned and augmented to a content embedding, and one decoder is used to generate output in different styles.

We propose a method which uses non-parallel data without disentanglement, using a probabilistic classifier. We created an Expert classifier Ex that can classify a series of tokens as expert or laymen and gives a value in percentile, directing the expertise level of the sequence. We train our model to predict the masked tokens and get the original input laymen sequence y.

## **1.5 Research Challenges**

### **1.5.1 Limited dataset**

Text related tasks required a lot of data. Given the fact that this work is in the medical field, thus medical data is a primordial necessity. Unfortunately to train our model the only available dataset is the MSD which stands for Merck, Sharp Dohme who are the founders of Merck Co



---

one of the largest pharmaceutical companies across the globe. MSD Dataset (Cao et al., 2020) made up of data collected from Merck Manuals (for doctors and consumers). The training data was annotated by 3 domain experts (doctors) and parallel sentences of the test data was provided by another doctor. There are 245023 non-parallel sentences in expert and layman styles and 1450 parallel sentences in expert and layman styles. For a model to be at its utmost performance it requires training with a huge amount of data. Parallel data are scarce in many real-world text style transfer applications.

### **1.5.2 Availability of hardware like GPU**

GPUs are optimized for training models. Because they can process multiple computations simultaneously. High Memory bandwidth and large number of cores are very important for proper training using GPU. Unfortunately, GPUs are expensive and not available for most of the researchers.

### **1.5.3 Unstable Evaluation Metrics**

Accuracy, fluency and content similarity are used to measure how well an expert text is converted to a layman text. Even at this age of deep learning revolution, we are far from inventing a specific way to evaluate natural language generation tasks. Even though, we use BLEU, PPL and some other metrics to evaluate the tasks, these metrics cannot evaluate using the underlying meaning of the reference and generated sentence.

## **1.6 Thesis Outline**

In the first chapter, we highlighted our study in a concise and brief manner, by precisely stating the problem at hand. The problem at hand is stated and explained in Chapter 2. In chapter 3, we discussed briefly on the existing state of the art models. In the next chapter, we go through each of the building blocks of our proposed approach like Transformers, Denoising Autoencoder and many more. Moving on, in Chapter 5 we have stated our proposed method, proposed algorithm, and a flow chart which provides a detailed intuition of our proposed method. In the next chapter we show results and related analysis of how our proposed method solves the medical expertise style transfer problem. In the last chapter, we share a condensed version of our work and the references and credits used.

## 2 The Problem

We have an input Expert sequence  $X$  containing  $n$  number of word vectors where  $X(n)$  denotes the  $n$ th vector representation of the whole sentence. Here the range of  $n$  is from 0 to  $\text{len}(X)-1$ . More specifically, Let  $x = x_1, x_2, x_3, \dots, x_n$  be an Expert sentence. For now we have to convert this expert sentences into a simpler sentence  $Y$  having  $m$  number of vectors where  $X(m)$  represents the word vector of the  $n$ th position. Here  $m$  ranges from 0 to  $\text{len}(Y)-1$ . Therefore our desired laymen sentence would be  $y = y_1, y_2, y_3, \dots, y_n$ .

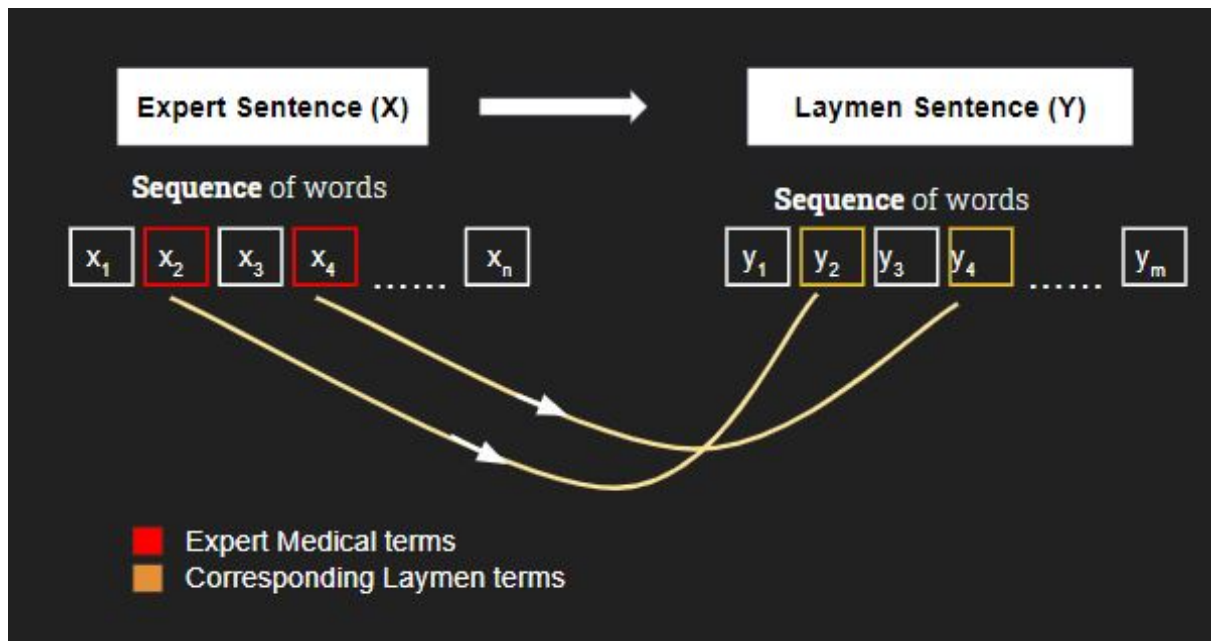


Figure 2: An Overview for Expertise Style Transfer

Here the conversion from  $X$  to  $Y$  has to be maintained considering two interrelated factors. First one is the decrement of the expertise level of the sentence. While constructing the new sequence of words, their combined expertise level certainly has to decrease significantly as well as maintain the authentic meaning of the word at the same time. Converting to the latter cannot afford to lose its content information. So the conversion has to be done in a controlled manner precisely handling two different factors in mind, style accuracy and content similarity while also producing syntactically and semantically correct sentences.

---

### 3 Related Works

The surge of research interest is immense in this text generative domain so far. Currently the three main state of the art methodology in this domain are I) Delete And retrieve II) Dual RL III) Controlled Gen. Promising to give the state of the art results in each technology, the main target of these text generative models is to transfer style, both in a supervised and unsupervised manner.

The main accomplishment of natural language generation (NLG) systems does not only depend on their ability to rebuild the sense of the source sentence but also upon careful consideration of transferring other attributes such as style and sentiment . Hence the interest in text attribute transfer in a more controlled and sophisticated manner has largely increased among NLP enthusiasts who now aim to edit a sentence in a way that can change the attribute keeping the context the same.

Attempts have been made on both Supervised and Unsupervised learning in this domain. Most models of text generation are task specific- they do not apply on generic text-style transfer generation when the generation is controllable. On the other hand, recent use of neural networks, Variational AutoEncoders and GAN tried to attempt a generic approach, but failed due to the uncontrollable nature of the generated text.

Text style transfer mainly aims to paraphrase the source text in the designated style conserving its original content .The application scenarios of these types of models are also vast, some includes: transferring a positive review to a negative one , revising an informal text into a formal one etc. One major challenge of these state of the art models are lack of parallel data. It is not only hard but very rare to get aligned sentences with the same content but different style.

We will now be looking into the existing three state of the art models stated above.

#### 3.1 Delete and Retrieve

In this model ([Li et al., 2018](#)), the authors considered changing the attributes of a sentence. Altering a particular attribute of the source text while keeping the attribute independent part of the sentence intact was the key idea of text style transfer in this model.

The method deletes phrases relating to the source sentences original attribute content in order to extract content words . It then retrieves new words relating with the target attribute. A neural model is then used to flawlessly combine these two into a final to fluently merge these into an output that has altered style but exact content as the source.

The ultimate goal here is to transform a sentence structure having one attribute ( for example positive sentiment into another one where the attribute is changed into one with a different attribute (for example : negative sentiment), keeping the rest of the content of the sentence intact.

Generally, pair of sentences aligned with the same content but different style are rarely obtainable; a particular model and its associated system must be able to learn by itself how to disentangle attributes and content with unaligned sentences only.

The noble observation by the authors in this paper was the fact that attributes of a sentence are usually marked by attribute markers, i.e: words or phrases in the sentence that indicate a particular attribute. If we are able to alter that marker, leaving the rest of the sentence intact, attribute transfer can be easily accomplished.

The baseline of this observation was to identify attribute markers from unaligned corpora of negative and positive sentences. This is done by finding words/groups of words that occur with higher frequency within sentences of one attribute than the other (e.g., “best” and “very happy” are positive markers). Second, in the source sentence, the positive markers are deleted and the remainder of the words kept as content. Lastly, a sentence with similar content is retrieved from the corpus with negative sentences.

The structure of similar models and their underlying concept is shown briefly using the figure 3.

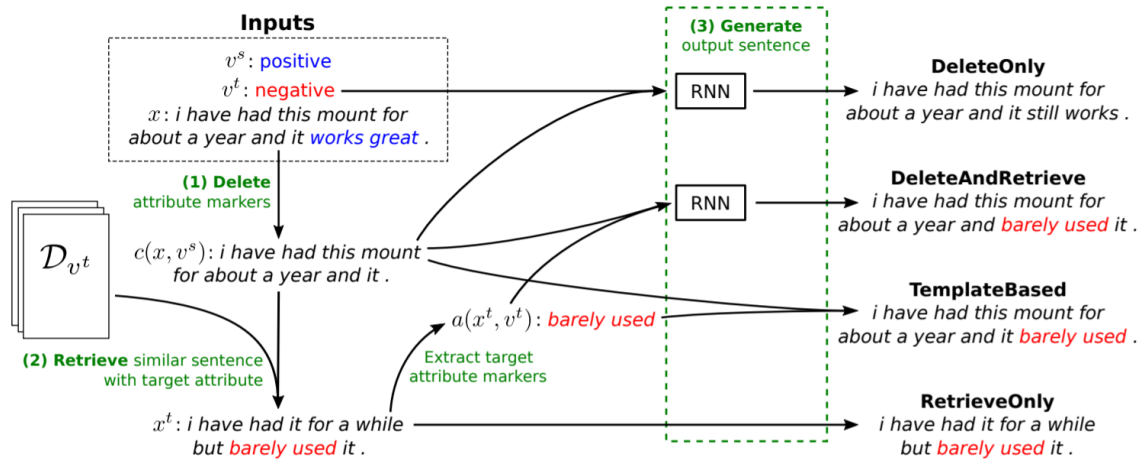


Figure 3: Delete & Retrieve

1. **Delete**: All the methods use the exact idea to separate the words in the source sentence  $x$  into a set of attribute markers  $a(x, v_{src})$  and a sequence of content words  $c(x, v_{src})$ .
2. **Retrieve**: 3 out of 4 models search the given corpus to find and retrieve a sentence  $x_{tgt}$  that with the target attribute  $v_{tgt}$  and with a content similar to the target sentence  $x$ .
3. **Generate**: Given the content  $c(x, v_{src})$ , target attribute  $v_{tgt}$ , and (optionally) the retrieved sentence  $x_{tgt}$ , each system generates  $y$ , either in a rule-based fashion or with a neural sequence-to-sequence model.

### 3.2 Dual Reinforcement Learning

In most models the text style transfer system consists of two steps. First, separating the style and the content and second adding the content with the desired style. But it is often seen that the style and the content are linked to each other in such a subtle way that separating them is impossible.

Fuli Luo et al argued how this two step process can be further modified into just one to solve the problem described above due to the the two-step process, they came up with the idea of one-step mapping model between the source corpora and the target corpora of different styles. (Luo et al., 2019) Since parallel data is very rare in this field, they considered learning of the source-to-target and target-to-source mapping models as a dual task hence proposing a dual reinforcement learning algorithm (e.g: DualRL) to train them.

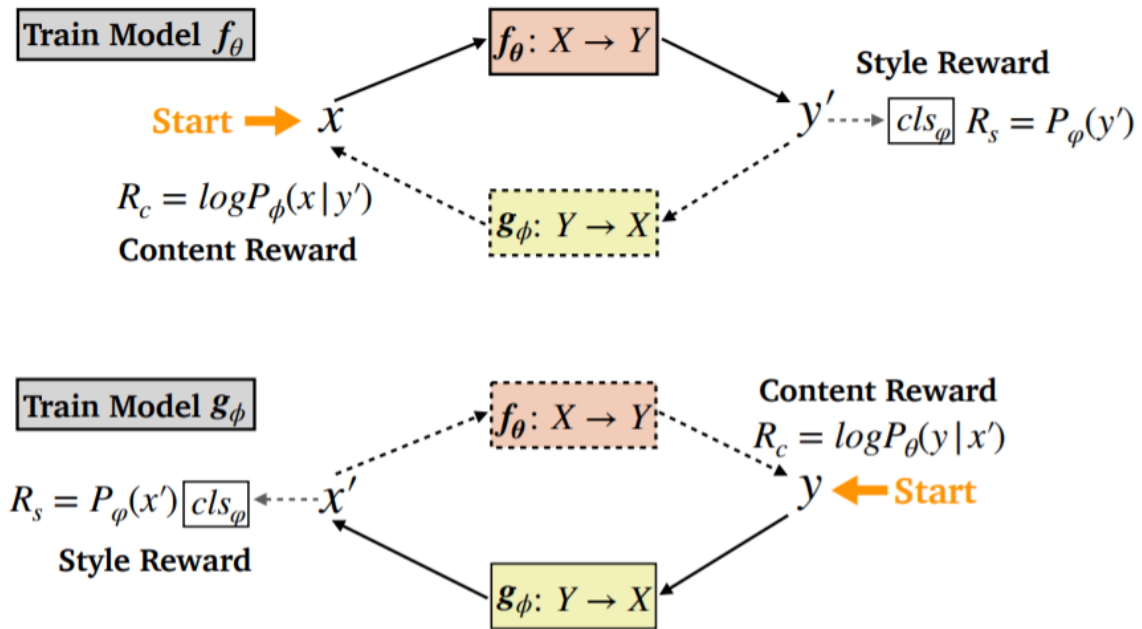


Figure 4: Dual Reinforcement Learning

Figure 4 shows the flow of work in Dual RL. In simpler terms, the forward model  $f_\theta: X \rightarrow Y$  transfers the sequence  $x$  with style  $s_x$  into a sequence  $y_0$  with style  $s_y$ , while the backward model  $g_\omega: Y \rightarrow X$  transfers the sequence  $y$  with style  $s_y$  into a sequence  $x_0$  with style  $s_x$ .

Fortunately, we have seen that text style transfer always occurs in pairs, hence it is easier to loop the dual processes back and forth, so that both directions can provide us valuable feedback in order to direct the training of the two style transfer models without using parallel data.

---

Now to define the rewards of the RL model. As we know they have "normative" content, stipulating what you want the model to accomplish, the rewards of this model is designed in such a way that the text style transfer is achieved.

### 3.2.1 Reward for changing style

A pre-trained binary style classifier [Kim, 2014] is used to evaluate how well the transferred sentence  $y'$  matches the target style. Formally, the style classifier reward is formulated as

$$R_s = P(s_y|y'; \phi)$$

where  $\phi$  is the parameter of the classifier and is fixed during the training process.

### 3.2.2 Reward for preserving content

It can be estimated how much the content is preserved in  $y'$  by means of the probability that the model  $\mathbf{g}$  reconstructs  $x$  when taking  $y'$  as input. Formally, the corresponding reconstruction reward is formulated as

$$R_c = P(x|y'; \phi)$$

where  $\phi$  is the parameter of model  $\mathbf{g}$

### 3.2.3 Overall reward

the final reward is the harmonic mean of the above two rewards

$$R = (1 + \beta^2) \frac{R_c \cdot R_s}{(\beta^2 \cdot R_c) + R_s}$$

In this way, the authors did not need to explicitly separate content and style, which is an extremely hard task even when the user has access to parallel data. Experiments were done on transferring sentiment and formality and results clearly showed how Dual RL outperforms any of the existing models. Empirically demonstrating the effectiveness of learning two one-step mapping models and the proposed DualRL training algorithm.

## 3.3 Controlled Generation

Zhiting Hu et al aims to generate text sentences whose attributes are generated at a controlled manner. This control can be achieved by learning disentangled latent representations with designated semantics. (Hu et al., 2017) They proposed a new neural generative model that includes variational auto-encoders (VAEs) and holistic attribute discriminators for effective imposition of semantic structures.

VAE and GAN have made a lot of advancement in the image domain but generating sentences remains a challenge here. Generating realistic sentences is difficult as the generative models have to capture complex semantic structures hidden in the folds of the sentences.

The proposed new text generative model allows a highly disentangled representation with desired semantic structure, and results in generating a sentence that has dynamic inclusion of attributes. The VAEs will have effective imposition of structures on the latent code. Differentiable Softmax was used as the optimization of end to end structures which acts as the annealing of discrete cases and causes it to converge fast. An additional discriminator, ie. The probabilistic encoder will help capture the underlying modelled aspects and act as a guide to the generator to avoid entanglement during attribute code manipulation.

In the model, they had an unstructured latent code in which the dimensions are entangled and  $z$  is structured code which targets a notable and independent semantic feature of sentences, controlling a sentence attribute. They proposed that the generator will condition on the combined vector  $(z, c)$ , and generate samples that satisfy the attributes as directed in the structured code  $c$ .

A recurrent network is created such that for each individual discriminator design is measured to see how well the generated samples match the desired attributes, and make the generator to produce improved results.

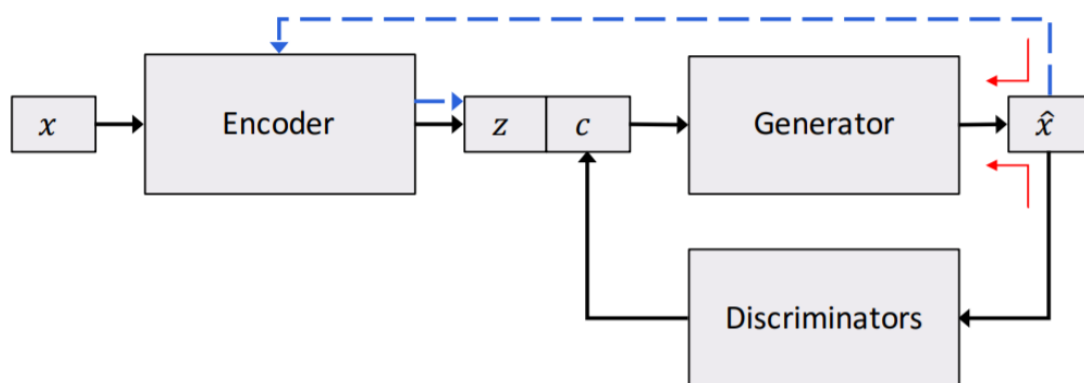


Figure 5: Controlled Generation

### 3.4 Shortcomings of Existing models

Controlled-Gen performs very well in terms of content similarity but poorly in terms of style accuracy. Delete Retrieve performs very well in terms of style accuracy but poorly in terms of content similarity (Cao et al., 2020). Dual RL takes too much time to train and test and does not perform robustly in terms of style accuracy and content similarity in a different dataset.

## 4 Background Study

### 4.1 Word Embeddings

In short Word Embeddings are logical numerical representations of text. One of the early problem of natural language processing is that machine has to learn the human language. Now this is important because as human we can perceive languages and realizes its meaning. We know the difference between "king" or "queen" and "man" or "woman". But for computers its not so much evident. The idea is that we create a high dimensional vector space where we can represent each word using a unique vector representation, where the similar words will have similar representation or more specifically they will be closer in the vector space.

Implementation of word embedding brought about a revolution in natural language processing as it could now effectively detect the similarity or difference between words within sentences. As they are represented in vector form, linear operations could easily modify the word representations. One perfect example would be like, "King" - "Man" + "Woman" = "Queen".

There are some other important factors too. Words might have multiple meaning. Some words can have different meaning in different positions. In that case, the modern word embeddings have different representation for the same words having different meaning. One example can be, "I always park (pk1) my car in the park (pk2)". In this sentence each 'park' has different meaning and so they should have different embeddings. In the sentence "Drive through the parking lot", the word "parking" then has to most relevant to the park (pk1) of the first sentence. On the other hand, in the sentence "Mr. Raj always wanted to visit the Ramna Park", the word "Park" is to be more closely related to the park (pk2) of the first sentence. An efficient word embedding will encode the words into vectors keeping these similarities and dissimilarities. They hold the syntactic and semantic properties of the close words like in (Mikolov et al., 2013).

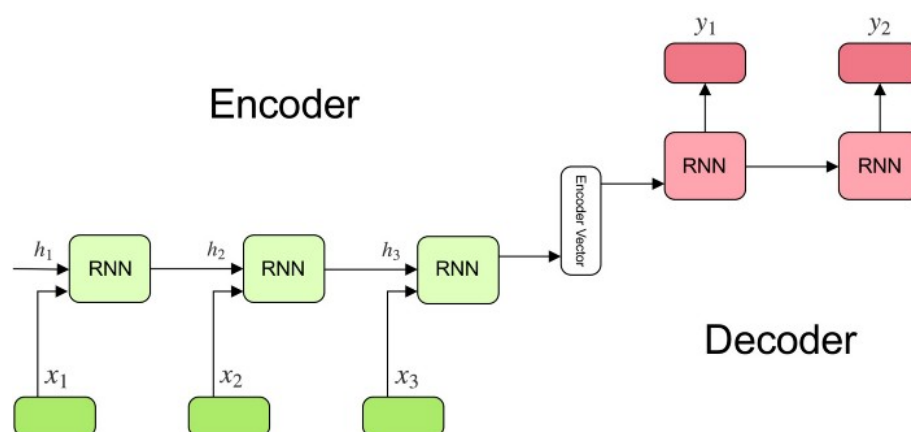


Figure 6: Encoder Decoder Model



## 4.2 Seq2Seq

Seq2Seq or Sequence to Sequence modeling was first proposed by Google (Sutskever et al., 2014), a general end-to-end approach for sequence learning. This was primarily introduced for neural machine translation, translating from one language to another language. Later it started become popular in text summarization, Sentiment conversion and all types of sequential models. Sentences are basically a sequence of words that can be converted into a sequence of word vectors or embedding. From the perspective of architecture, Seq2Seq models have two parts- the Encoder and the Decoder. (Cho et al., 2014)

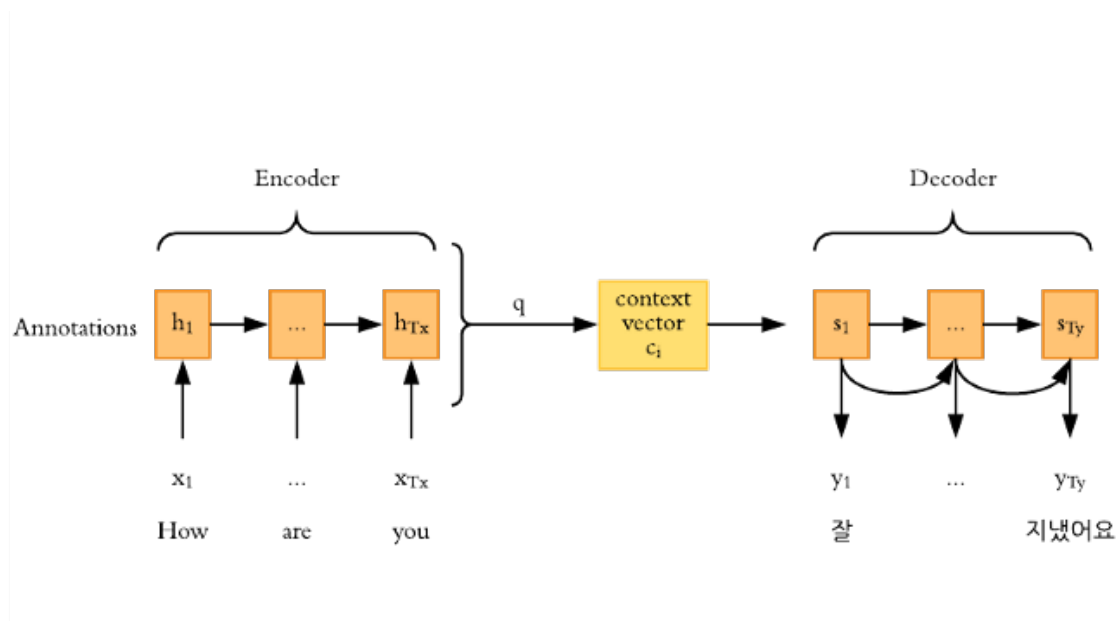


Figure 7: Seq2Seq Without Attention Mechanism

Figure 6 shows a RNN based Seq2Seq model with corresponding Encoder and Decoder section. The input sequence goes through the Encoders sequentially. The encoders are basically a series of RNNs or LSTMs. Each input vector goes through an encoder RNN and forwards a hidden representation to the next layer along with the next input vector. Thus each output of the RNN contain all the information of the previous input sequences. Thus final a context vector is created which is called Encoder representation.

Now the model need to decode the context vector to the preferred output sequence through the Decoder which is also a series of RNNs or LSTMs. The decoder RNNs decode the context vector sequentially by producing one output vector at a time. Figure 6 shows an example of Encoder-Decoder based Seq2Seq model.

### 4.3 Attention Mechanism

In the Encoder Decoder model the sequence data is computed one by one. Due to its long correlated inter-dependencies in the context vector, for larger sentences it is more likely that the initial information might be lost. That's why attention mechanism was introduced. In the figure the encoder decoder model with attention is shown. Here the words 'How', 'are', and 'you' are converted into word embedding in creating the encoder vector.

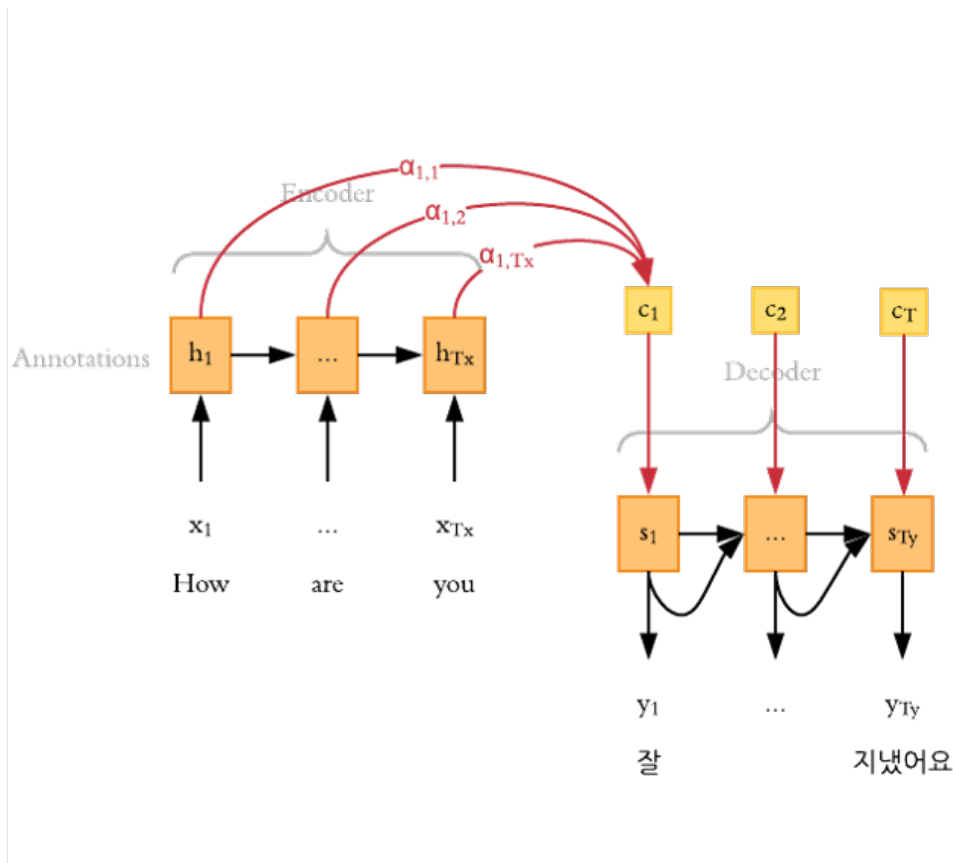


Figure 8: Seq2Seq with Attention Mechanism

While decoding, apart from the context vector, the decoder is also fed with a direct connection from the input sequence representing the relevance of the corresponding words, dictating where the model should focus more seriously to predict the generated sentence precisely. Figure 7 shows sequence to sequence model without attention and Figure 8 shows sequence to sequence model with attention.

## 4.4 Transformer

In the Transformer model, (Vaswani et al., 2017) proposed a new network architecture for sequence to sequence modeling solely based on the attention mechanism. Sequential models like RNN and LSTM process the sentences word by word, thus creating a huge problem in the process of parallelization.

They introduced multi-head attention in their architecture. It's basically the implementation of self attention, where the input sequence learns the similarities among itself denoting which words more related to each word. Thus it can have a deeper understanding about the language and its construction. The multi-headed attention is learned using three vector values, Q (query), K (key) and V (value). These parameters essentially contain the relationship among the word vectors.

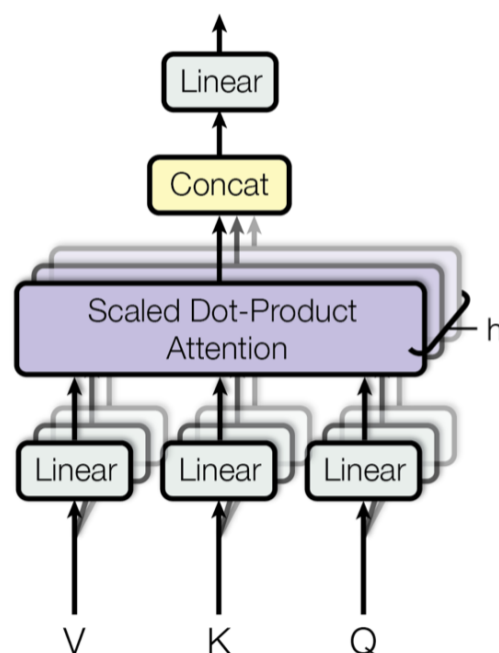


Figure 9: Multihead Attention

One of the reasons sequential models like RNN were so successful for language processing, was that it actually goes through the sentence one by one, thus learning the positional relationship among the words. For solving this issue in Transformer, Positional Encoding was introduced. It is maintained using corresponding a sin and cos function for each odd and even positions. Given enough GPU computational power Transformers can theoretically save infinitely long correspondence relationship among the word vectors, while in case of RNN/LSTM longer sequences tend to lose its initial information. Along with that as the data is not processed one by one it is highly efficient in case of parallelization. Figure 10 gives a complete overview.

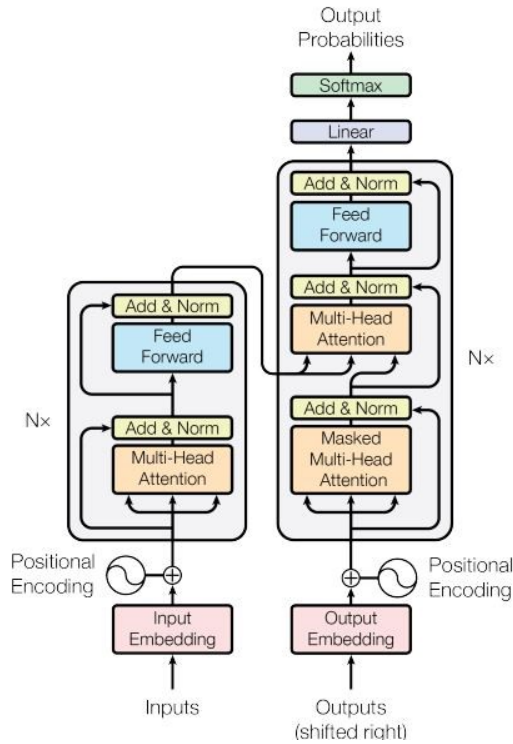


Figure 10: Transformer Architecture

#### 4.5 BERT

In the paper by Devlin et. al.(2019) (Devlin et al., 2018) a new language representation model BERT (Bidirectional Encoder Representations from Transformers) was introduced which improves fine-tuning based approaches. It is designed to pre-train deep bidirectional representations.

To understand the framework the concept of MLM and NSP is needed to be clear. A masked language model (MLM) randomly masks words in the sentence and tries to predict them using their context. Next sentence prediction (NSP) is replacing the next sentence with another random sentence from the corpus in order to train a model that is capable of understanding sentence relationships. These sentence relationships cannot be captured by language modelings and they are required for tasks such as question-answering.

Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	##ing	[SEP]
Token Embeddings	$E_{[CLS]}$	$E_{my}$	$E_{dog}$	$E_{is}$	$E_{cute}$	$E_{[SEP]}$	$E_{he}$	$E_{likes}$	$E_{play}$	$E_{##ing}$	$E_{[SEP]}$
	+	+	+	+	+	+	+	+	+	+	+
Segment Embeddings	$E_A$	$E_A$	$E_A$	$E_A$	$E_A$	$E_A$	$E_B$	$E_B$	$E_B$	$E_B$	$E_B$
	+	+	+	+	+	+	+	+	+	+	+
Position Embeddings	$E_0$	$E_1$	$E_2$	$E_3$	$E_4$	$E_5$	$E_6$	$E_7$	$E_8$	$E_9$	$E_{10}$

Figure 11: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

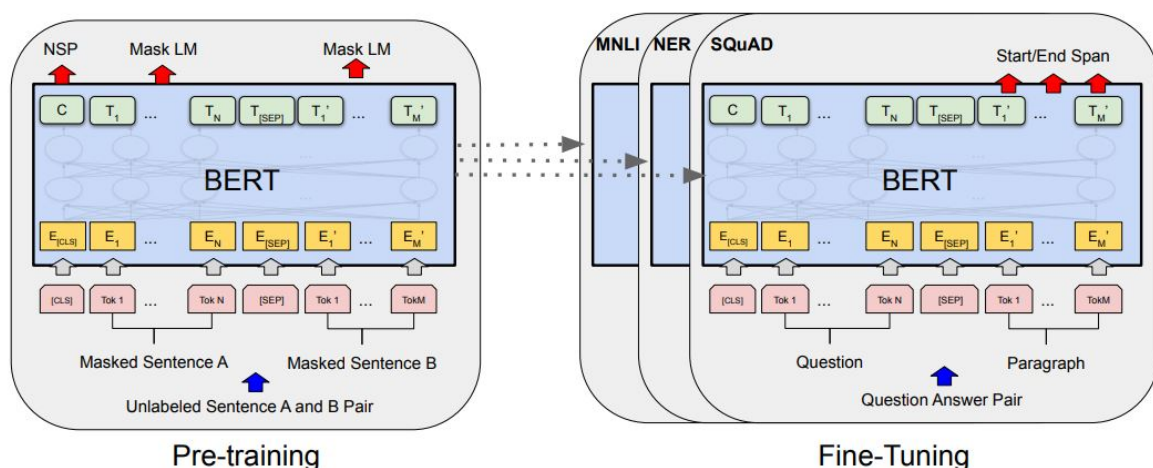


Figure 12: The two steps in BERT architecture in (Devlin et al., 2018)

The BERT framework is composed of two steps: Pre-training BERT: the model is trained using unlabeled data on two tasks which are MLM and NSP. Fine-tuning BERT: first, the model is initialized with the pre-trained parameters and next, the parameters are fine-tuned for the desired downstream task.

#### 4.6 RoBERTa

The RoBERTa model (Liu et al., 2019) was proposed by Facebook in 2019. They presented a replication study of the BERT model which includes a careful evaluation of the effects of hyperparameter tuning and training set size. In their experimentation, they found that BERT was significantly undertrained. RoBERTa builds on BERT's language masking strategy, wherein the system learns to predict intentionally hidden sections of text within otherwise unannotated language examples. RoBERTa, which was implemented in PyTorch, modifies key hyperparameters in BERT, including removing BERT's next-sentence pretraining objective, and training with much larger mini-batches and learning rates. This allows RoBERTa to improve on the masked language modeling objective compared with BERT and leads to better downstream task performance. Their process includes:

- training the model longer, with bigger batches, over more data;
- removing the next sentence prediction objective;
- training on longer sequences; and
- dynamically changing the masking pattern applied to the training data.

They also collect a large new dataset (CC-NEWS) of comparable size to other privately used datasets, to better control for training set size effects. The RoBERTa model substantially improves the performance over BERT. They showed the importance of their design decisions in the BERT model.

## 4.7 Autoencoder

Autoencoders are artificial neural networks, prepared in an unsupervised way, that mean to initially learn encoded portrayals of our information and afterward produce the info information (as intently as could be expected) from the learned encoded portrayals. Accordingly, the yield of an autoencoder is its expectation for the information.

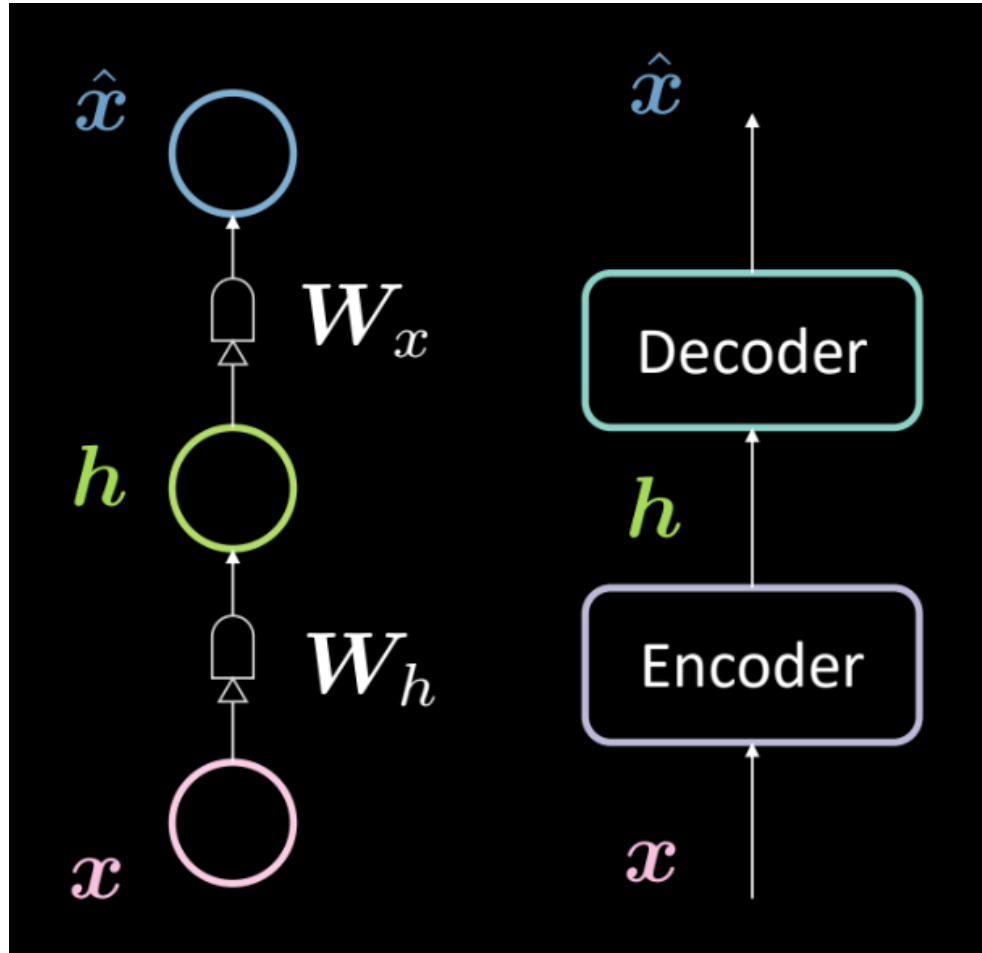


Figure 13: Autoencoder

Figure 13 shows the architecture of a basic autoencoder. As before, we start from the bottom with the input  $\mathbf{x}$  which is subjected to an encoder (affine transformation defined by  $W_h$ , followed by squashing). This results in the intermediate hidden layer  $\mathbf{h}$ . This is subjected to the decoder (another affine transformation defined by  $W_x$  followed by another squashing). This produces the output  $\hat{x}$ , which is our model's prediction/reconstruction of the input. As per our convention, we say that this is a 3 layer neural network. We can represent the above network mathematically by using the following equations:

$$h = f(W_h * x + b_h)$$

$$\hat{x} = g(W_x * h + b_x)$$

The essential utilizations of an autoencoder is for inconsistency identification or picture denoising. We realize that an autoencoder's undertaking is to have the option to remake information that

lives on the complex for example given an information complex, we would need our autoencoder to have the option to recreate just the information that exists in that complex. Accordingly we compel the model to recreate things that have been seen during preparing, thus any variety present in new data sources will be taken out on the grounds that the model would be obtuse toward those sorts of bothers.

Another use of an autoencoder is as a picture blower. On the off chance that we have a transitional dimensionality  $d$  lower than the information dimensionality  $nn$ , at that point the encoder can be utilized as a blower and the secret portrayals (coded portrayals) would address all (or the vast majority) of the data in the particular information however take less space.

#### 4.8 Under / over-complete hidden layer

At the point when the dimensionality of the secret layer  $d$  is not exactly the dimensionality of the info  $nn$  then we say it is under finished secret layer. What's more, correspondingly, when  $d > n$ , we consider it an over-complete secret layer. Figure 14 shows an under-complete secret layer on the left and an over-complete secret layer on the right.

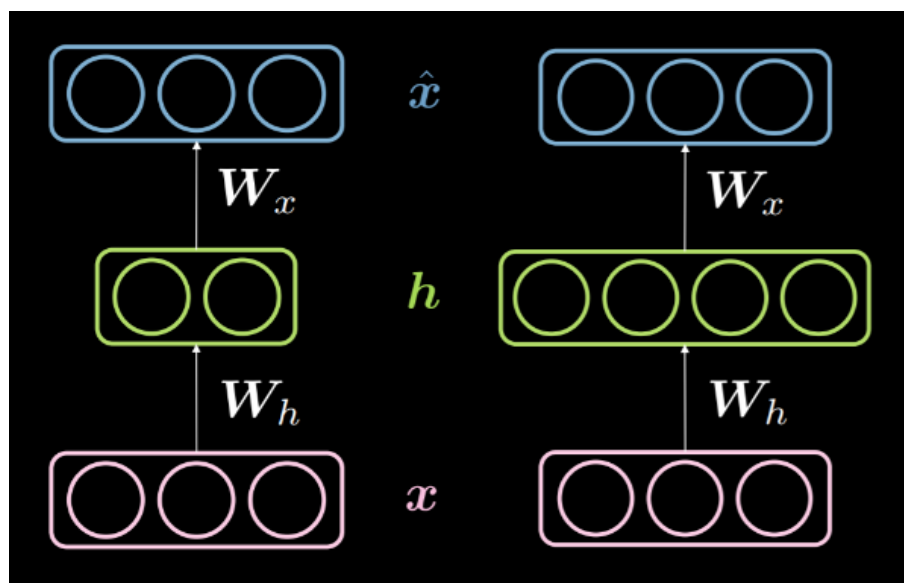


Figure 14: Under / over-complete hidden layer

As talked about over, an under-complete secret layer can be utilized for pressure as we are encoding the data from contribution to less measurements. Then again, in an over-complete layer, we utilize an encoding with higher dimensionality than the information. This makes streamlining simpler.

Since we are attempting to reproduce the info, the model is inclined to replicating all the information highlights into the secret layer and passing it as the yield hence basically carrying on as a character work. This should be maintained a strategic distance from as this would suggest that

our model neglects to learn anything. Subsequently, we need to apply some extra limitations by applying a data bottleneck. We do this by obliging the potential arrangements that the secret layer can take to just those designs seen during preparing. This considers a specific remaking (restricted to a subset of the information space) and makes the model unfeeling toward everything not in the complex.

It is to be noticed that an under-complete layer can't act as a personality work basically in light of the fact that the secret layer needs more measurements to duplicate the information. Subsequently an under-complete secret layer is more averse to overfit when contrasted with an over-complete secret layer however it could in any case overfit. For instance, given an amazing encoder and a decoder, the model could basically relate one number to every information point and become familiar with the planning. There are a few techniques to maintain a strategic distance from overfitting like regularization strategies, design techniques, and so forth.

#### 4.9 Denoising Autoencoder

In this model, we accept we are infusing a similar uproarious circulation we will see as a general rule, with the goal that we can figure out how to powerfully recuperate from it. By contrasting the information and yield, we can tell that the focuses that generally on the complex information didn't move, and the focuses that distant from the complex moved a great deal. Figure 15 shows the manifold of the denoising autoencoder and the intuition of how it works.

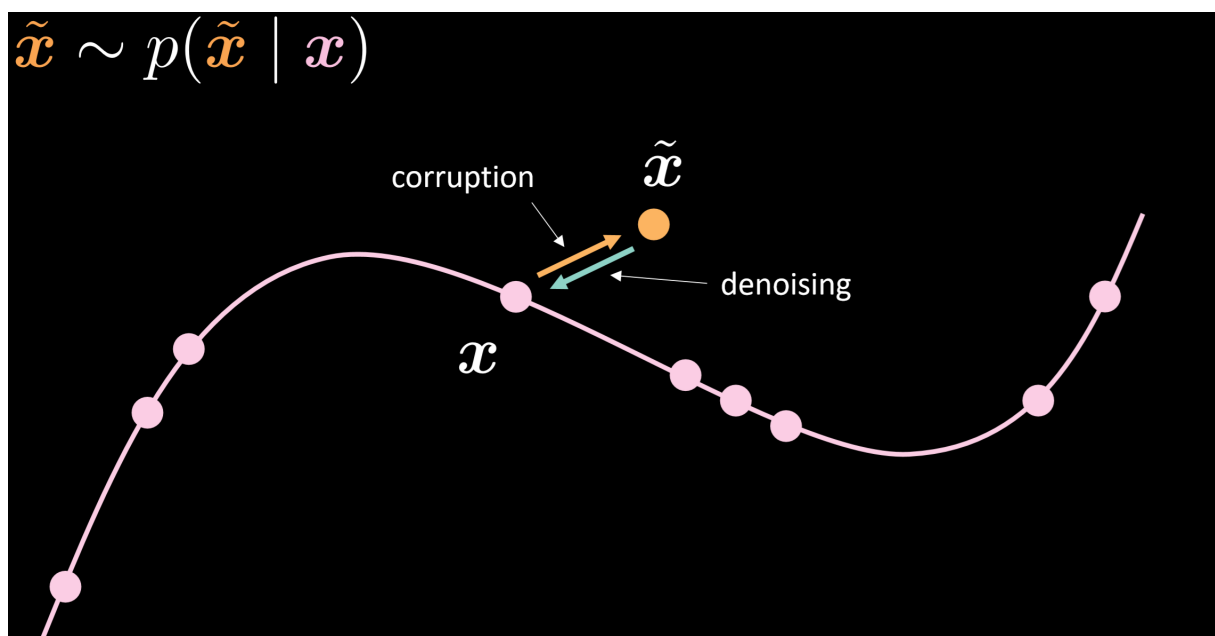


Figure 15: Denoising Autoencoder



## 5 Proposed Method

None of the existing baseline approaches are stable across all the metrics. We proposed a novel approach that performs better and stable across all the metric scores. We divide our full approach into four stages as mentioned in Figure 16.

In the very first stage, we train an expertise classifier using the MSD Dataset (Cao et al., 2020) while using the text as input and styles as labels.

Next, we mask the expert terms using the expertise classifier that we trained on the first stage. After that, we take the layman text corpus from the MSD dataset, mask 15% of each text and use a denoising autoencoder to generate the original text. By doing this, we are training a model which can predict layman terms for masked regions according to the context provided by the unmasked terms.

Finally, we combine everything by taking MSD test data. For each test data, we take an expert text, mask the expert terms using the expertise classifier which e trained on the first step and later generate a layman text from the masked sentence using the trained denoising autoencoder from the third stage.

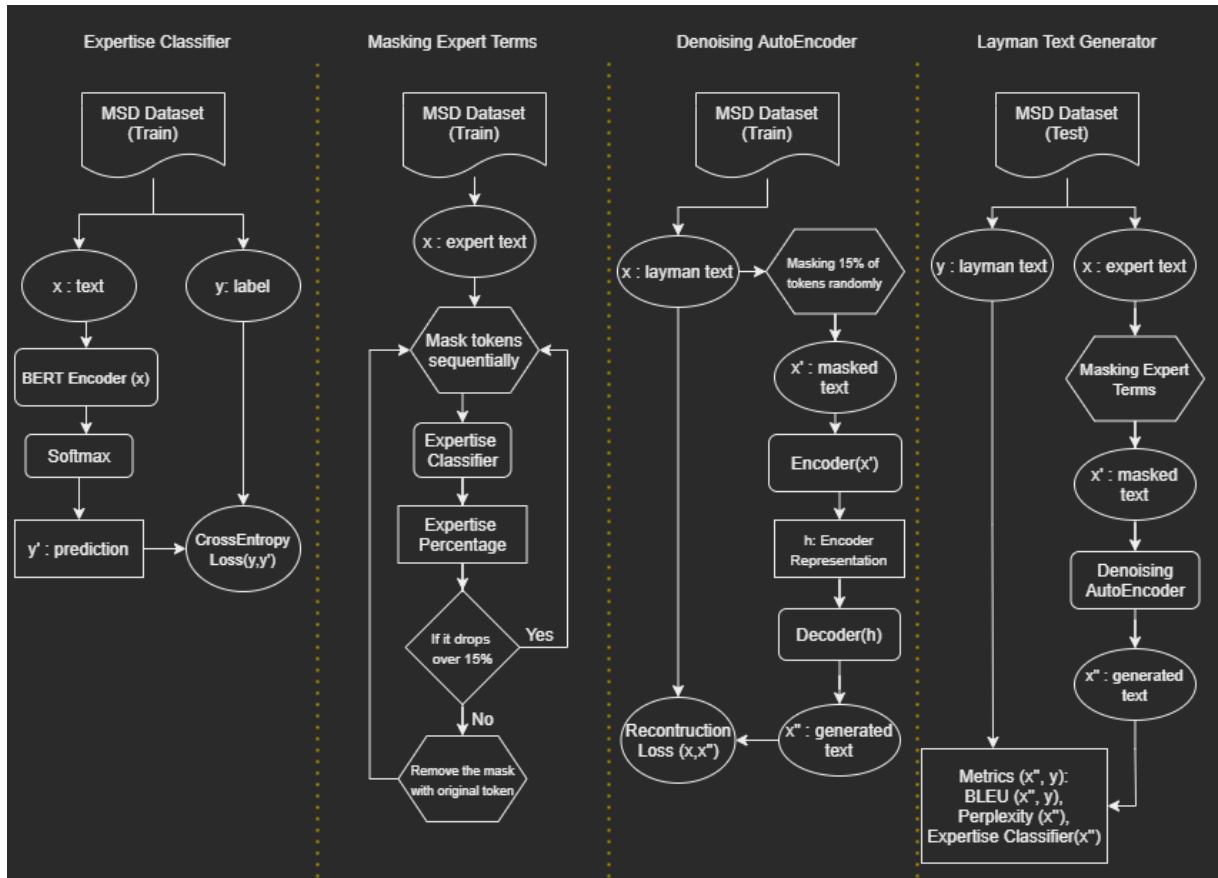


Figure 16: Proposed Model Architecture

---

## 5.1 Expertise Classifier

MSD Dataset has both expert and laymen text. We take the texts as input and style as labels. '0' stands for the expert text label and '1' stands for the layman text label. From Figure 16, we can get a brief idea about the expertise classifier.

We fine-tune the BERT Encoder (Devlin et al., 2018) model while attaching a softmax layer on top. The softmax layer predicts the probability of the sentence being an expert or layman. We use Cross-Entropy Loss as the loss function. For our case, we specifically use Binary Cross-Entropy (BCE) Loss.

$$BCELoss = -(y_i * \log(p_i) + (1 - y_i) * \log(1 - p_i))$$

## 5.2 Masking Expert Terms

```
def masking_expert_terms(text):
    initial_expertise := expertise_classifier. predict(text)
    # getting initial expertise score

    masked_text := text
    # creating a copy of original text

    for i in range( **len(text)** ):
        masked_text[i] := '[MASK]' # Adding [MASK] token

        current_expertise := expertise_classifier. predict(masked_text)
        # getting initial expertise score

        if ( initial_expertise - current_expertise > 0.15 * len(text) ):
            initial_expertise := current_expertise
            continue
        # if initial_expertise drops over 15% then keep the mask
        # else remove the mask with original term
        masked_text[i] = text[i]

    return masked_text
```

Figure 17: Masking Expert Terms

---

Figure 17 is the step-by-step pseudocode for this stage. There was no training part. We used the expertise classifier of the previous stage for our inferencing. If we describe this stage into words, we get the following:

- Get an expert text.
- Mask tokens one by one.
- Get the expertise percentage using expertise classifier.
- Does the percentage drop more than 15% ?
  - => If yes, keep the mask and go to step 2.
  - => If no, replace the mask with original term and go to step 2.
- Stop when all the terms are checked.

### 5.3 Denoising Autoencoder

In our previous stage, we masked the expert term. Now we need to predict the possible layman terms in the masked regions. For this, we get the layman corpus from the MSD dataset for this stage. For each sample, we mask 15% of the terms and feed it to denoising autoencoder. (Vincent et al., 2008)

In the denoising autoencoder model architecture, we have an encoder and a decoder. The encoder generates the encoded representation of the masked sentences and the decoder generates meaningful sentences while filling up the masked regions with layman terms from context. We use Reconstruction Loss as a loss function since our main task here is to generate the original layman text before masking.

$$\text{Reconstruction Loss} = \frac{1}{2} * ||x - x''||^2$$

### 5.4 Layman Text Generator

In this stage, we combine the previous stages and test our generated output with the required metrics. Here, we take the expert terms from the MSD test set. For each expert text, we mask the expert terms using the same steps we followed in Figure 17. Then, use the denoising autoencoder we trained on the laymen corpus to generate layman style text. The generated text fills up the masked terms with layman terms. The generated text is compared with the reference layman text for evaluation. We use BLEU, perplexity and style accuracy as evaluation metrics.

## 6 Experimental Results

### 6.1 Dataset

For our experimentation, we are using MSD Dataset (Cao et al., 2020). MSD stands for Merck, Sharp Dohme, founders of Merck Co. , one of the largest pharmaceutical companies in the world. MSD Dataset (Cao et al., 2020) is collected from Merck Manuals (for doctors and consumers). The train data was annotated by 3 doctors (domain expert) and parallel sentences of the test data was provided by another doctor. There are 245023 non-parallel sentences in expert and layman styles and 1450 parallel sentences in expert and layman styles.

Table 1: Example of MSD Dataset

Text	Style	Concepts
Myocardial fibrosis , left ventricular hypertrophy , and cardiomyopathy can develop .	Expert	[{"range": [0, 2], "term": "myocardial fibrosis", "cui": ["C0151654"]}, {"range": [3, 6], "term": "left ventricular hypertrophy", "cui": ["C0232306"]}, {"range": [8, 9], "term": "cardiomyopathy", "cui": ["C0878544"]}]
Chronic use can also damage the heart , causing scarring and thickening of the heart muscle and eventually leading to heart failure .	Laymen	[{"range": [0, 1], "term": "chronic", "cui": ["C1555457"]}, {"range": [6, 7], "term": "heart", "cui": ["C0018787"]}, {"range": [9, 10], "term": "scarring", "cui": ["C0008767"]}, {"range": [14, 15], "term": "heart", "cui": ["C0018787"]}, {"range": [15, 16], "term": "muscle", "cui": ["C4083049"]}, {"range": [20, 22], "term": "heart failure", "cui": ["C0018802"]}]
For confirmation , selected noninvasive and invasive cardiac tests are usually done.	Expert	[{"range": [1, 2], "term": "confirmation", "cui": ["C1611825"]}, {"range": [6, 7], "term": "invasive", "cui": ["C1334278"]}, {"range": [7, 9], "term": "cardiac test", "cui": ["C4529960"]}, {"range": [10, 11], "term": "usually", "cui": ["C3888388"]}]

Given the adjusted gatherings of sentences in expert and buyer MSD, they build up a comment stage to encourage master explanations. They enlist three specialists to choose sentences from every rendition of gathering to explain sets of sentences that have a similar importance yet are written in various styles. The recruited specialists are officially medicinally prepared, and are able to comprehend the semantics of the clinical writings. To stay away from emotional decisions in the comments, they are not permitted to change the substance. Especially, the specialists are Chinese who additionally know English as a subsequent language. Hence, we furnish the English substance went with a Chinese interpretation as help, which assists with expanding the explanation speed while guaranteeing quality.

### 6.2 Experimental Setup

We did all our experiments using Google Colab which is a hosted Jupyter notebook service. We used it because Google Colab provides free GPU for 12 hours a day. In our experiments we used Numpy, Pandas, etc. for data processing and PyTorch for training and testing. PyTorch (Paszke et al., 2017) is an open source machine learning framework. We chose 10% randomly as validation set from the MSD dataset. Table 2 shows the time needed for computations. We trained the expertise classifier for 10 epochs and Denoising Autoencoder for 10 epochs. The other

two stages (Masking Expert Terms and Layman Text Generator) did not need GPU computations, they needed CPU computations.

Table 2: Required time analysis

Stage	Hardware	Required Time
Expertise Classifier	GPU	5 hours (10 epochs)
Masking Expert Terms	CPU	10 minutes
Denoising Autoencoder	GPU	11 hours(10 epochs)
Layman Text Generator	CPU	30 minutes
	<b>Total Time</b>	<b>17 hours (approx.)</b>

### 6.3 Training

We used the pretrained BERT (Devlin et al., 2018) model encoder adding a softmax layer on top to train the expertise classifier. Initially, we freeze the pre-trained weights and train the frozen model. Later, we unfreeze the whole model and train the whole model. This enables in reaching a higher accuracy in few epochs.

Later, We used a pretrained denoising autoencoder (Vincent et al., 2008) model to train our denoising autoencoder. Initially, we freeze the pre-trained weights and train the frozen model. Later, we unfreeze the whole model and train the whole model. This enables in reaching a higher accuracy in few epochs.

### 6.4 Evaluation Metrics

#### 6.4.1 Style Accuracy

We use the trained expertise classifier leveraging MSD dataset which we used to verify if the generated sentence is layman or not. We calculate style accuracy by,

$$\text{Accuracy} = \frac{\text{identified-layman-text}}{\text{total-layman-text}}$$

#### 6.4.2 Perplexity

Fluency is usually measured by the perplexity of the transferred sentence. We fine-tune the state-of-the-art pretrained language model, GPT-2 (Radford et al., 2019), on the training set of each dataset for each style.

Table 3: Geneated Layman Text Examples by our approach

Expert Text	Masked Text	Generated Layman	Reference Layman
Urea breath testing and stool antigen testing .	[MASK] breath testing and stool [MASK] testing .	Breath testing and stool testing .	Testing of breath and stool
Often , there is no visible oro pharyngeal inflammation .	Often , there is no visible [MASK] [MASK] inflammation .	Often , there is no visible throat inflammation .	Often , there is no visible inflammation in the throat.
Night vision may eventually be lost .	[MASK] [MASK] may eventually be lost .	The patient may eventually be lost .	Seeing at night or in the dark or dim light becomes more difficult .
Nausea , vomiting , constipation , severe prostration , restlessness , and irritability are common .	[MASK] , vomiting , constipation , severe [MASK] , restlessness , and irritability are common .	Vomiting , constipation , severe fatigue , restlessness, and, irritability are common .	Nausea , vomiting , constipation , extreme fatigue , irritability , and restlessness are common .

### 6.4.3 BLEU

Content Similarity measures how much content is preserved during style transfer. We calculate 4-gram BLEU (Papineni et al., 2002) between model outputs and inputs. Programmed measurements for content closeness are seemingly questionable, since the first data sources as a rule accomplish the most noteworthy scores. We accordingly lead human assessment. To assess over the whole test set, just layman annotators are included, yet we guarantee that the layman style sentences are went with as references to help understanding. Every annotator is approached to rate the model yield given both info and gold references. The rating goes from 1 to 5, where higher qualities show that more semantic substance is protected.

## 6.5 Result Analysis

Table 4: Result Analysis

Approaches	Style Accuracy	BLEU	Perplexity
Delete And Retrieve	74.67	2.95	3.92
ControlledGen	11.70	<b>13.13</b>	5.97
<b>(Ours)</b>	<b>78.76</b>	8.17	<b>3.12</b>

Table 4 shows a comparative analysis of the existing approaches and our proposed approach. Here we can see that our approach works better than the previous baseline models in style

---

accuracy and perplexity. And, our approach works better than Delete and Retrieve (DR) but worse than ControlledGen in terms of Content Similarity.

Our approach works better in Style Accuracy than previous baseline models because we initially remove the expert terms which are responsible for making the sentence expert. Later, we are removing them with layman terms which gives us an upperhand here. Our perplexity score works better because we generate the layman sentence from denoising autoencoder.

## 7 Conclusion

We proposed a new approach which performs stable across all evaluation metrics. More hyperparameter tuning and training time needed to get to the state of the art result. In future, we plan to extend the work to transfer laymen style text to expert style text also. We also plan to perform further human evaluation after training and testing due to the deficiency of proper evaluation metrics.

## References

- Colin Camerer, George Loewenstein, and Martin Weber. 1989. The curse of knowledge in economic settings: An experimental analysis. *Journal of political Economy*, 97(5):1232–1254.
- Yixin Cao, Ruihao Shui, Liangming Pan, Min-Yen Kan, Zhiyuan Liu, and Tat-Seng Chua. 2020. Expertise style transfer: A new task towards better communication between experts and laymen. *arXiv preprint arXiv:2005.00701*.
- Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. *arXiv preprint arXiv:1703.00955*.
- Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. A dual reinforcement learning framework for unsupervised text style transfer. *arXiv preprint arXiv:1905.10060*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

- 
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Sharon Swee-Lin Tan and Nadee Goonawardene. 2017. Internet health information seeking and the patient-physician relationship: a systematic review. *Journal of medical Internet research*, 19(1):e9.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008.
- Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103.