Unsupervised and Semi-supervised Style Transformation by Using Sentence-BERT as a Service

Ali Saheb Pasand
Final Project Report for CS886: Deep Learning and Natural Language Processing
Course Instructor: Professor Ming Li
Department of Computer Science, University of Waterloo, Waterloo, ON, Canada
Emails: asahebpa@uwaterloo.ca

***Abstract*—In this project, we have proposed a novel method for style transformation with a hybrid approach which combines unsupervised and semi-supervised manners in style transformation and requires low computational cost during training. We use a pre-trained sentence encoder to obtain reach embedding space for sentences. Then, the representation of sentence with an initial style is transformed to a representation with the desired style. Finally, a pre-trained BERT model with Language Model Head is used to generate a sentence given initial sentence and the transformed representation in the embedding space.**

Output examples show the effectiveness of the proposed method in transforming style without loosing non-stylistic properties (e.g. meaning and fluency). Two BLEU scores, self and transformed BLEU scores, are chosen as auto-evaluation metrics to compare different methods. Also, the result of human evaluation is reported. Our method has achieved acceptable style transformation performance with minimal computational cost during training stage, in contrast with the state-of-the-art methods which require significant computational power during training phase.

I. Introduction

Style transformation is one of the most challenging tasks in the field of Natural Language Processing. The goal of this task is to change the stylistic characteristics of a text (e.g. sentiment) in a way that non-stylistic properties (e.g. meaning) remain unaffected. Achieving this goal is not straightforward because of the following challenges:

1) Stylistic and non-stylistic features of a text are highly coupled in a way that separating them is not possible in many cases.
2) It is not possible to solve this task by adopting known supervised learning approaches since constructing paired sentences with the same meaning but different style is difficult. Because of lack of paired data, all the methods for solving this task have to adopt unsupervised or semi-supervised approaches, which inherently introduce more noise to the process compared to supervised approaches.
3) For style transformation task, there is no flawless metric for comparing different methods or to be used during training. Even human evaluation sometimes can not show if a method has preserved all non-stylistic properties and achieved all goals in transforming stylistic ones since the border between these two sets of features is vague.

The common practice in style transformation is to use auto-encoder structure in order to achieve latent space in which style and meaning are disentangled. During this procedure, methods usually use RNN or LSTM neural network structures which cannot lead to a reach and powerful latent space representation capable of preserving meaning and all the information about a sentence. In order to achieve reach representations for sentences, we have used pre-trained sentence transformer. The other point in which our model is different to other models is it demands minimal computational power during training stage. Other methods are based on training extremely deep networks in order to achieve proper embedding. We have bypassed this computational burden by using pre-trained sentence transformer which can provide us with proper embedding for sentences. Also, as the decoder we have used pre-trained transformer with language model head to generate the sentence by rephrasing words in the initial sentence; this decoder also requires minimal training since there is proper pre-trained transformer structures which are able to guess words to be used instead of masked words by using bi-directional context of the masked word. The only part which requires training is the component which transforms an embedded representation for a sentence in a style to a target representation which has the desired target style. We have shown that this component does not need to be deep and complex in order to lead to proper results. Our contribution are summarized as follows:

1) We introduce a novel training procedure to address style transformation in unsupervised and semi-supervised manners.
2) In contrast with common practice in style transformation, our model does not require significant computational power to be trained. Two out of three blocks in our model are pre-trained and we have connected them by a component with minimum complexity.
3) The results show that our model preserves fluency and meaning during style transformation as good as complex methods which require enormous computational power

during training.

In the following sections, first we will look at some related work in this area. Then, components used in our model will be explained in detail. In the end, the results on Yelp review dataset will be discussed and some future work will be suggested.

II. Related Work

There are two main approaches for solving style transformation in text. One of the major paths of research has aimed to infer a latent representation for the input, and manipulate that representation in order to alter the style. The other path is based on methods which do not alter the representations and try to address the problem with indirect manipulation of the latent space.

Among methods adopted the first approach, Hu et al. (2017) [1] proposed a generative model which can separate different styles in latent space by using a discriminator alongside with a variational auto-encoder structure (to obtain a generative model). Shen et al. (2017) [2] proposed a combination of cross-aligned auto-encoder with adversarial training to learn latent space with shared meaning distribution and a separated style distribution. Other methods such as John et al., 2018 [3]; Fu et al., 2018 [4]; Zhang et al., 2018 [5] and Zhang et al., 2018 [6]; has also adopted encoder-decoder architecture.

There are some approaches without altering latent representation. As an example there are some methods adopting Reinforcement Learning (because the sentence space is discontinued). For example, Xu et al. (2018) [7] proposed a method which uses cycles of Reinforcement Learning. Their target is transforming a sentiment to another. Li et al. (2018) [8] proposed a multi-stage method. Their model extracts words representing meaning by deleting phrases or words which show style, then retrieves new phrases which represent target style, and finally combines these phrases into the output. Prabhumoye et al, 2018 [9] have used back-translation to achieve style transformation. The term back-translation in their work is not the same as the meaning in Unsupervised Machine Translation; they first translate a sentence from a language into second language with this condition that the words used in the second language has no style property, and there are neutral. Then, they translate it back to the source language in a way that the translated sentence have the desired style. Lample et al., 2019 [10] considered style transformation as a special case of Unsupervised Machine Translation (UMT) problem (Lample et al., 2018 [11]). Yang et al., 2019 [12] have used the idea behind UMT in order to generate classical Chinese poems from Vernacular Chinese, and they have adopted Reinforcement Learning to solve under- and over-translation issues. Dai et al., 2019 [13], which is the state-of-the-art method in this task, have used transformer structure in order to get a context aware representation and transforming sentences directly.

The approach that we have adopted is the mixture of both mentioned paths. Two approaches that we have adopted are as the following:

1) Semi-supervised (based on altering the representation): We use the manipulated representation as the ground truth in order to achieve pseudo-parallel dataset.
2) Unsupervised path (based on manipulating latent space indirectly): Has stem for the idea that suggests we can consider style transformation as a special case of UMT in which first and second language are the same language with different styles.

Detailed explanation of these approaches is provided in section III-B2.

III. Methodology

To explain our methodology we need to explain all the components used in our model. Our proposed model has three main components:

1) Sentence Transformer: A pre-trained sentence encoder which gives us a representation for a sentence.
2) Embedding Transformation: A component which accepts an initial representation of a sentence and transforms that to a representation with the desired style.
3) Sentence Decoder: A pre-trained transformer which accepts the initial sentence alongside with the transformed representation of that sentence and generates the sentence with the desired style.

In the following three subsections, these components will be explained in detail.

A. *Sentence Transformer*

There have been different methods to obtain proper representations for sentences. The definition of a proper representation is different for different tasks. For example, in sentiment classification task, proper representation is the one which leads to the best separation between target classes in embedding space. In our task, proper representation is the one which maps sentences with the same meaning and the same style to points which are geometrically close to each other in the embedding space. An example of proper representation can be seen in Figure 1. In this embedding space, sentences with meaning close to each other will be mapped to points which are geometrically close to each other in the embedding space. By using a sentence encoder which is aware of the meaning and the pairwise relationship of sentences, we can achieve style transformation by simple vector arithmetic operations. For example in the Figure 1, a naive way to transform a positive review about food to a negative one, is to subtract the average of the source cluster (all positive reviews about food) and add the average of the target cluster (all negative reviews about food). This procedure can be seen in Figure 2. If the embedding space is proper, in the sense discussed before, the result will be a sentence which is about food but with negative sentiment.

We have used Sentence Transformer (also known as Sentence-BERT) proposed by Reimers and Gurevych [14] as the encoder for sentences. To train this transformer, they have started with a base transformer model, for example BERT (Devlin et al.,2018 [15]) or RoBERTa (Liu et al., 2019 [16]) .

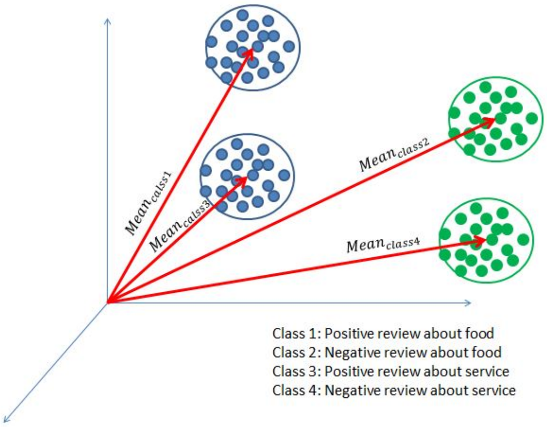
$Mean_{calss1}$
$Mean_{calss3}$
$Mean_{class2}$
$Mean_{class4}$
Class 1: Positive review about food
Class 2: Negative review about food
Class 3: Positive review about service
Class 4: Negative review about service

Fig. 1. Structure of embedding space proper for style transformation

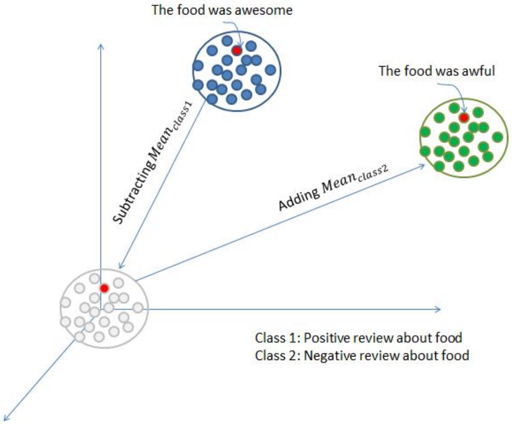
The food was awesome
The food was awful
Subtracting $Mean_{class1}$
Adding $Mean_{class2}$
Class 1: Positive review about food
Class 2: Negative review about food

Fig. 2. Style transforming with vector arithmetic (aligning the center of clusters)

Then they have retrained these models with a dataset consists of pairs of sentences with three different relation between them: contradiction, entailment, and neutral. They feed pairs of sentences through transformers with shared weights (siamese network structure). Their proposed encoder seems to be proper for our task as it embeds similar sentences to points which are geometrically close to each other in embedding space. As it is shown in Table I, sentences with the similar meaning has been mapped to vicinity of each other. In Table I, two query sentences and four closest neighbors of those sentences are shown. As it can be seen, the sentences close to each other have similar words, context, and sentiment.

B. Embedding Transformation

As discussed in the previous section, the embedding space resulted by sentence transformer has some geometrical characteristics which make it proper for style transformation. Now our goal is to transform a sentence representation in a way that if we decode the new representation, it leads to a sentence with the same meaning but different style. To achieve this goal, we have considered two methods:

TABLE I
FOUR NEAREST NEIGHBOURS FOR TWO EXAMPLES IN YELP REVIEW DATASET

Query: enjoy it with the blue corn tortilla chips or a side of fresh fruit.
Four nearest neighbors: 1. the corn fritters are crunchy and good, served with jalapenos and cucumber. 2. delicious fajitas, fresh corn tortillas and corn chips. 3. they serve warm tortilla chips and some pretty good salsa. 4. I especially loved the slight crisp to the edges of the corn tortillas.
Query: fabulous food (the duck was amazing) and great ambience.
Four nearest neighbors: 1. the food was delicious, especially the duck. 2. the duck special is also great... well prepared and very flavorful. 3. I had the duck special that was wonderful and full of great flavor. 4. and the duck soup is amazing.

1) Vector Arithmetic
2) Neural Network Style Transformer

In the following two subsections, these two methods will be discussed.

1) Vector Arithmetic: As discussed before and shown in Figure 2, a way to transform style is to consider different styles as clusters in embedding space. Since there is arithmetic relationship between sentences inside a cluster and sentences with the same meaning have been mapped to points close to each other, we can consider the vector calculated by averaging over all sentences with the same style as a proper vector representation of a style. Now in order to change the style, we can subtract the vector representing the style of the sentence, and then add the vector representing the target style to that.

Suppose that for each style there is a set containing sentence embedding of sentences with that style. This set can be written as following:

$$Style_j = \{e_1^{s_j}, ..., e_{k_j}^{s_j}\} \tag{1}$$

where $e_i^{s_j}$ is the embedding of the sentence i with style j and $e_i^{s_j} \in \mathbb{R}^d$. In our case $d = 768$. Also, k_j is the number of sentences with style s_j.

Now by taking average over the elements of these sets we will get vectors which represent different styles. We call the average of all sentence embedding with style s_j as $RepVec_{s_j}$ and it can be calculated as the following:

$$RepVec_{s_j} = Mean\{Style_j\} \tag{2}$$

Now that we have these average vectors, we can transform style of a sentence from style s_j to s_t as the following:

$$\hat{e}^{s_t} = e^{s_j} - RepVec_{s_j} + RepVec_{s_t} \tag{3}$$

To obtain the sentence with the new style, a simple method is to search for the closest point in the embedding space to $\hat{e}^{s_t}$

TABLE II
TRANSFORMING A SENTENCE FROM THE MODERN TRANSLATION OF HAMLET TO THE ORIGINAL VERSION OF THAT
(MEAN ADJUSTING)

Modern Translation: Assume, as it were, that you have only heard about him, Such as, ”I know his father and his friends, And in part him”, are you listening to me, Reynaldo?
Original Version: Take you, as ’twere, some distant knowledge of him, As thus, ’I know his father and his friends, And in part him.’ Do you mark this, Reynaldo?
Modern Translation: I really hope your virtues Will bring him to his usual way again, To the honor of both of you.
Original Version: So shall I hope your virtues Will bring him to his wonted way again, To both your honours.

TABLE III
TRANSFORMING A SENTENCE FROM THE ORIGINAL VERSION OF HAMLET TO THE MODERN TRANSLATION OF THAT
(MEAN ADJUSTING)

Original Version: What is’t, Ophelia, he hath said to you?
Modern Translation: What is it, Ophelia, that he has said to you?
Original Version: Thus twice before, and jump at this dead hour, With martial stalk hath he gone by our watch.
Modern Translation: It’s come twice before, and just appearing out of nothing, he’s gone past us at this dead hour with a warlike stalk.

among all elements in set $Style_t$. Two examples of this method can be seen in Tables II and III. These examples have been obtained by performing the introduced technique for solving two tasks [1]:

1) Transforming a sentence from Modern Translation of Hamlet to the Original Version of that (Table II).
2) Transforming a sentence from Original Version of Hamlet to the Modern Translation of that (Table III).

In the discussed method, we have assumed that the shape of clusters generated by sentences with different styles are the same; in general, it is not true. If we consider two different styles as two clusters in embedding space, these clusters may have different variance towards different directions of the embedding space. In that case, in addition to adjusting the center of clusters as shown in Figure 2, we need to perform scaling towards different dimensions of the embedding spaces. The geometric interpretation of this procedure can be seen in Figure 3. The mathematical expression for this procedure is almost the same as the previous method with an extra matrix vector multiplication. We need to define a scale matrix called as $Scale_{(s_j,s_t)}$. This matrix scales the dimensions of the cluster of style s_j in a way to fit the cluster of style s_t. The matrix is calculated as the following:

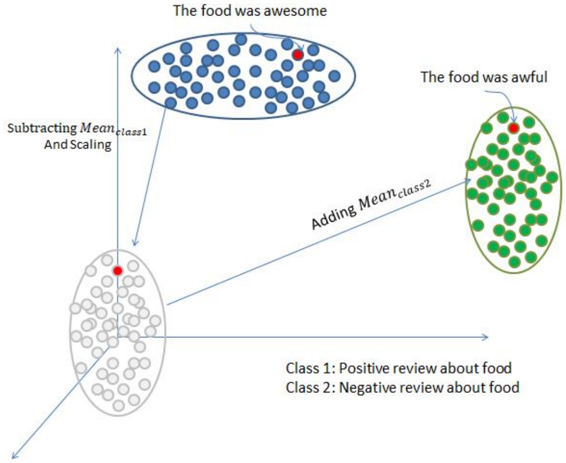
The food was awesome
The food was awful
Subtracting $Mean_{class1}$
And Scaling
Adding $Mean_{class2}$
Class 1: Positive review about food
Class 2: Negative review about food

Fig. 3. Style transforming with vector arithmetic (aligning the center of clusters and scaling)

$$Scale_{(s_j,s_t)} = \begin{bmatrix} \frac{\sigma_1^{s_t}}{\sigma_1^{s_j}} & 0 & \cdots & 0 \\ 0 & \frac{\sigma_2^{s_t}}{\sigma_2^{s_j}} & \cdots & 0 \\ 0 & \cdots & \ddots & \vdots \\ 0 & \cdots & 0 & \frac{\sigma_d^{s_t}}{\sigma_d^{s_j}} \end{bmatrix} \tag{4}$$

where ${\sigma_i}^{s_k}$ is the standard deviation of the cluster of sentences with style s_k towards the i-th dimension of the embedding space. By knowing this matrix, transforming style can be preformed as the following:

$$\hat{e}^{s_t} = (e^{s_j} - RepVec_{s_j}) \times Scale_{(s_j,s_t)} + RepVec_{s_t} \tag{5}$$

It must be noted that e^{s_j} is a vector with dimension $1 \times d$ and $Scale_{(s_j,s_t)}$ is a $d \times d$ matrix. The discussed method has not preformed well for style transformation. A possible reason is that in this case we do not impose any rotations during cluster reshaping. A better method is based on eigenvalue decomposition. The geometrical interpretation can be seen in Figure 4.

In this method our goal is to make the covariance matrices of the source and the target styles the same. Suppose the eigenvalue decomposition of the covariance matrix calculated based on all sentences with style s_j is as the following:

$$Cov_{s_j} = V_{s_j} \Sigma_{s_j} V_{s_j}^T \tag{6}$$

[1] Data has been obtained from the following link: https://github.com/tokestermw/tensorflow-shakespeare

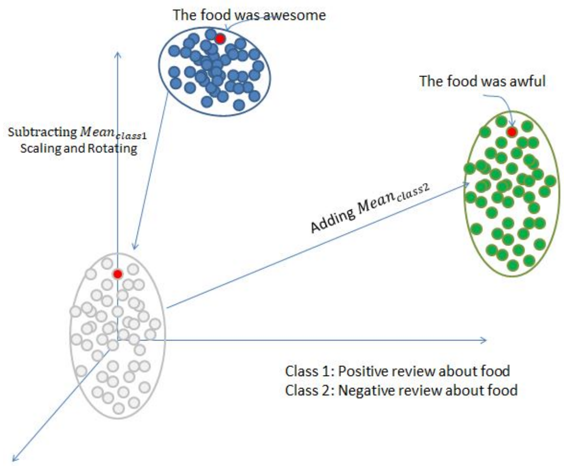
The food was awesome
The food was awful
Subtracting $Mean_{class1}$
Scaling and Rotating
Adding $Mean_{class2}$
Class 1: Positive review about food
Class 2: Negative review about food

Fig. 4. Style transforming with vector arithmetic (aligning the center of clusters rotating and scaling)

It can be shown that in order to reshape the cluster of style s_j to become the same as the cluster of style s_t, we should multiply all vectors in the cluster of style s_j by the following matrix ($Reshape_{(s_j,s_t)}$):

$$Reshape_{(s_j,s_t)} = V_{s_j} \Sigma_{(s_j,s_t)} V_{s_t}^T \tag{7}$$

where $\Sigma_{(s_j,s_t)}$ can be calculated as the following:

$$\Sigma_{(s_j,s_t)} = \Sigma_{s_j}^{-\frac{1}{2}} \Sigma_{s_t}^{\frac{1}{2}} \tag{8}$$

After calculating the matrix for the reshaping purpose, we can achieve style transformation as the following:

$$\hat{e}^{s_t} = (e^{s_j} - RepVec_{s_j}) \times Reshape_{(s_j,s_t)} + RepVec_{s_t} \tag{9}$$

The results obtained from this method can be seen in Tables IV and V.

2) Neural Network Style Transformer: The methods based on Vector Arithmetic has an important downside; all of these methods are based on this assumption that the shape of clusters for different styles are the same (all are hyper-ellipsoids), and the only difference is the scale towards different directions of the embedding space and the location of the centre of clusters. In general, this assumption is not valid; all sentences with a style lay on a manifold which may be completely different compared to the manifold of other styles in sense of shape and direction. Therefore, we need a method which can capture the manifold of each style and reshape that to become the same as the manifold of the other style.

Neural Networks are capable of manifold capturing and transformation. The only missing point in our task, which makes using Neural Networks in conventional ways impossible, is lack of paired sentences. Because of that, supervised methods cannot be used in order to achieve manifold learning and reshaping task.

TABLE IV
TRANSFORMING A SENTENCE FROM THE MODERN TRANSLATION OF HAMLET TO THE ORIGINAL VERSION OF THAT
(EIGENVALUE AND MEAN ADJUSTING)

Modern Translation: I don't care about my sincerity, but I could accuse myself of such things that it were better my mother had not had me.
Original Version: I am myself indifferent honest, but yet I could accuse me of such things that it were better my mother had not borne me.
Modern Translation: Why do we wrap the gentleman in our more rare breath?
Original Version: Why do we wrap the gentleman in our more rawer breath?

TABLE V
TRANSFORMING A SENTENCE FROM THE ORIGINAL VERSION OF HAMLET TO THE MODERN TRANSLATION OF THAT
(EIGENVALUE AND MEAN ADJUSTING)

Original Version: What to ourselves in passion we propose, The passion ending, doth the purpose lose.
Modern Translation: What we promise ourselves in a fit of passion,When the passion ends, so does the promise.
Original Version: I'll have thee speak out the rest of this soon.
Modern Translation: I'll have you speak out the rest of this soon.

The Neural Network structure used for this purpose consists of two auto-encoders which will be trained in unsupervised and semi-supervised manners. The steps taken are as the following:

1) Dataset Separation: The dataset will be divided into separate datasets each corresponding to one style. For example, Yelp dataset is divided into positive and negative reviews. It must be noted that these two datasets are not aligned. In case of aligned dataset, style transformation can be achieved with less error by known supervised learning techniques.

2) Training Auto-encoders Separately: Separate auto-encoder structures have been trained for each style. This step can be seen in Figure 5. After this step, we

have auto-encoders which have captured the manifolds of different styles. Since we do not have any parallel data, we need to exploit these two auto-encoders to perform style transformation in Unsupervised and Semi-supervised manners.

3) Unsupervised Path for Transforming Style: The idea has stem from two papers on ”Unsupervised Machine Translation” by Lample et al. 2018 [11] and 2017 [17]. The style transformation task is, in many ways, similar to Unsupervised Machine Translation as there is no parallel data in both tasks. The only difference is that the manifold of different styles is much closer to each other than two different languages. We will address this problem in Semi-supervised part. The idea behind unsupervised part of style transformation is that sentences in different styles are coming from the same manifold in some latent space which represents only meaning and no style or form. This idea stems from the philosophical observation that all material with the same meaning and different forms (picture, video, sentence in different languages or styles) are coming from the same space called ”Intelligible realm”.

   The method used to find the latent space in which two styles have a shared manifold can be seen in Figure 6. Suppose that we want to transform a positive review to a negative one (we omitted explanation of transforming negative review to positive review since both are similar with different paths). Embedding of a positive review is passed through encoder trained during step 2. Then, it will be passed through decoder trained during step 2 for negative reviews. Since there is no parallel data, we cannot compare this embedding with any gold transformed embedding to calculate the loss. Because of that, the representation calculated in this step will be passed through encoder trained for negative reviews. Then, it will be passed through decoder trained for positive reviews. Since we have an embedding decoded by Positive review encoder, we can compare that with the initial embedding by computing Mean-squared Error between them. This loss is called $LossPositivetoNegative_{Butterfly}$. The term ”Butterfly” has been used in the name because of the shape of the path. The similar procedure (with different ordering of blocks) can be performed for transforming negative reviews as well and the loss will be called $LossNegativetoPositive_{Butterfly}$.

4) Semi-supervised: The reason behind taking this step is that the results obtained by Vector Arithmetic techniques mentioned in section III-B1 were promising. It means that, although we do not have access to parallel data, we can obtain pseudo-parallel data by using Vector Arithmetic.

   This method is illustrated in Figure 7. The embedding of a positive review will be passed through the encoder trained on positive reviews. Then, it will be passed through the decoder trained on negative reviews. In this scenario since we have access to pseudo-parallel data by using Vector Arithmetic techniques, we can compute Mean-squared Error between the output of the decoder and the target embedding calculated by Vector Arithmetic. This loss is named $LossPositivetoNegative_{Semi}$. The similar path (with using other two blocks) can be passed to transform negative review to positive. The loss will be named $LossNegativetoPositive_{Semi}$.

   Between three mentioned Vector Arithmetic techniques, only the first one (adjusting the center of clusters) worked properly. A possible reason is that in neural network we work with batches, and probably, reshaping each batch with rotation and scale calculated over all sentences in dataset are not proper for reshaping batches.

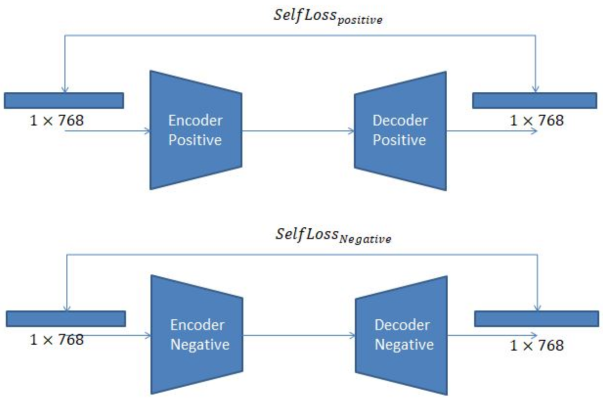

Fig. 5. Training Auto-encoders Separately

Paths 3 and 4 will be trained together using the combination of losses as follows:

$$Loss_{total} = \lambda_1 Loss_{Butterfly} + \lambda_2 Loss_{Semi} \tag{10}$$

where $Loss_{Butterfly}$ and $Loss_{Semi}$ are summation of two terms calculated on Unsupervised and Semi-supervised paths.

C. Sentence Decoder

Now that we have access to the embedding of the transformed sentence, we need a decoder to generate an actual sentence from that embedding. For this task, we have tried the powerful decoder called GPT-2 proposed by Radford et al. [18]. The problem in using this decoder is that we need to feed word embedding into the model and it does not accept sentence embedding. We tried to obtain word embedding by subtracting the embedding of a sentence with and without the specific word. We could not get a promising result by passing this word embedding to the model. The output sentences were not fluent and meaningful in many cases.

To generate the final sentence in a controlled way, in a way that fluency and meaning are preserved, pre-trained

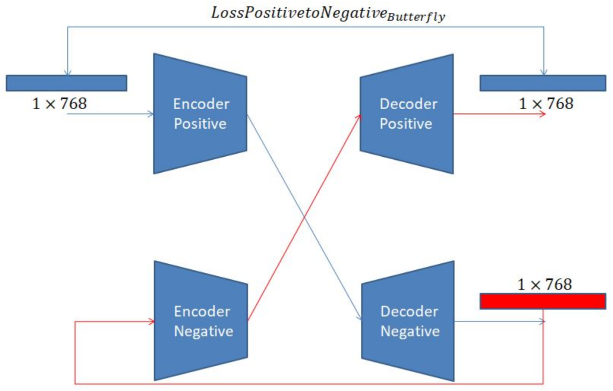

Fig. 6. Unsupervised Path of Style Transformation

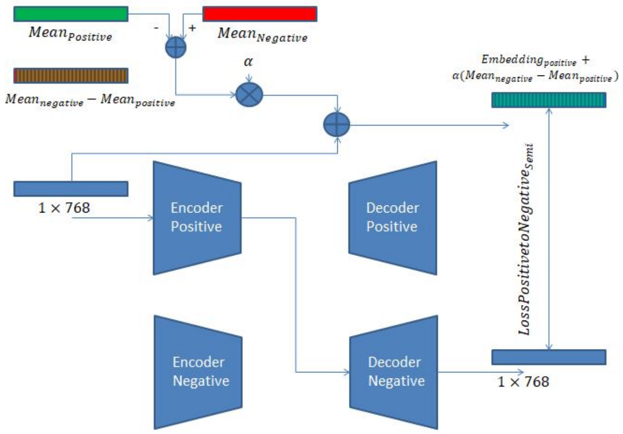

Fig. 7. Semi-supervised Path of Style Transformation

BERT model with Language Model Head is used. Wang and Cho in their article [19] have proposed three methods to generate meaningful sentences with BERT. They start with a seed sentence, a sentence with some initial words or only [CLS] token, and they try to predict the words which can come after that initial seed. We have adopted the idea behind their method and used that in other way; since most of the style transformation tasks are actually paraphrasing, we have used the initial sentence as the seed to the model. In each iteration, we mask a word in the sentence and feed the masked sentence into BERT. BERT's Language Model Head predicts the word which is suitable for the masked position. After few iterations, we call these iterations "Iteration Inside", we get a sentence. Since we are trying to achieve style transformation, we want this sentence to have the target style. To check if we have achieved the target sentence or not, we pass the sentence through Sentence Transformer discussed before and compare the embedding of that sentence with the embedding computed by Embedding Transformation component (Section Embedding Transformation III-B). The iteration outside is the number of times that we perform random masking and sentence generation. The other input parameters to this decoder are as the following:

TABLE VI
DETAIL OF DATASET

	Positive Reviews	Negative Reviews
Train	266041	177219
Validation	2000	2000
Test	500	500

TABLE VII
AN EXAMPLE OF ITERATIONS HAPPENING DURING INSIDE ITERATIONS

Original Sentence:
ever since joes has changed hands it's just gotten worse and worse.

Transformation of the sentence during inside iterations:
Iteration 1:
ever since joes has changed, (*) it's just gotten worse and worse.
Iteration 4:
ever since joes has changed, it has just got worse and (*) worse.
Iteration 7:
but since joes has changed, it has just got worse and worse (*).
Iteration 11:
but since has changed, it has only got worse and better (*).
Iteration 14:
ever since joes has changed,(*) it has only got better and better.

1) Temperature: Is smoothing parameter for the distribution based on which model chooses words. Higher means more uniform distribution; lower means less smooth distribution (more peaks). If temperature is 1.0 it means no change imposes on the distribution.
2) Number of Candid Words: Controls how many candidates the model should consider for each masked position. If this value is greater than one, $k > 1$, decoder will choose a random word among k words with the highest probabilities. That random choosing will be based on the probability of each of the candid words (instead of always choosing the one with the highest probability). It means that, among those candidates, the one with higher probability has the higher chance to be chosen.
3) Range of output length: Specifies the maximum length of the output sentence.
4) Number of Inside Iterations: The number of times that random words will be masked and the decoder tries to guess that masked positions. An example of inside iterations can be seen in Table VII. In each iteration, the word or character before sign (*) was masked in the beginning of the iteration and the model has guessed the word or character before the sign. Number of Outside Iterations: The number of times the original sentence will be fed to the model. In other words, the number of sentences decoder generates.

In the end, the sentence whose representation has the minimum distance to the transformed embedding will be chosen as the final result. The whole procedure can be seen in Figure 8.

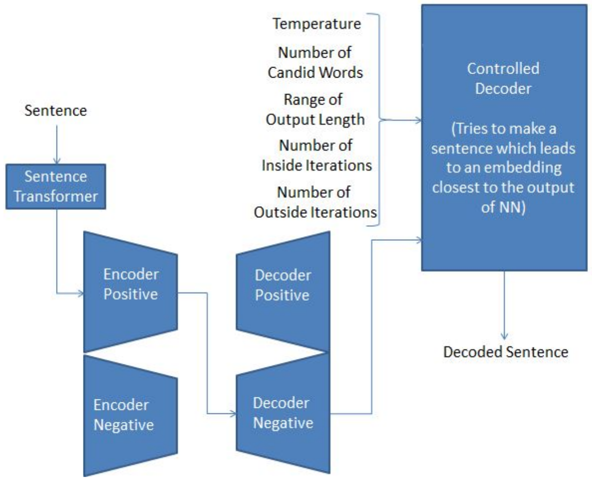

Fig. 8. The Whole Procedure of Transforming Style

In this example there is a positive review which we want to transform it to a negative one.

IV. Results

We have performed tests on Yelp review dataset. The detail of dataset can be seen in Table VI

As the component for calculating the embedding of sentences, We have used pre-trained Sentence Transformer called 'bert-base-nli-mean-tokens'. It is a Bert model with mean pooling and trained on dataset nli (Reimers and Gurevych [14] for more details). The Encoder and Decoder for the part that we try to transform embedding are feed-forward neural networks with two layers and Relu activation function. The parameters of Sentence Decoder are as the following:

1) Temperature = 1
2) Number of Candid Words = 13
3) Range of Output Length: $[Input_{Length}, Input_{Length} + 2]$
4) Number of Inside Iterations = $Input_{Length}$
5) Number of Outside Iterations = 100

One of the challenges in unsupervised style transformation is that there is no proper metric to evaluate a method except human evaluation, which is costly and not possible on large datasets. Among all metrics, we have chosen the following three metrics for evaluation:

1) Self BLEU Score: BLEU score between original sentence and the transformed sentence.
2) Transformed BLEU score: BLEU score between the transformed review by our algorithm and the gold standard transformed sentences (Sentences transformed from positive to negative or vice versa by human) provided by the githup repository [2] related to article Dai et al., 2019 [13].
3) Human Evaluation: Human have compared transformed sentences with the original sentences to see what percentage of them have preserved non-stylistic attributes and achieved acceptable transformation in stylistic ones.

Among the mentioned metrics, BLEU scores are not flawless metrics. Self BLEU score is not proper since the transformed version, if the transformation has been done perfectly, may not have a lot of overlapping n-grams with the original sentence. The transformed BLEU score is not proper since there is not only one correct transformed sentence to compare the results with. We have calculated these two scores to compare our model to other models. The BLEU scores can be seen in Table VIII. The values from other methods have been gathered from article Dai et al., 2019 [13]. The human evaluation results can be seen in Table IX. Unfortunately there is no proper human evaluation values for other techniques to compare the results with. Some examples of our method alongside with the output of other known methods can be seen in Tables X and X; the results of other methods have been gathered from paper dai et al., 2019 [13]. More examples from our model can be seen in folder uploaded on dropbox [3]. The results have been post-processed to replace [UNK] tokens with their counterparts in the actual sentence.

V. Conclusions and Future Work

Although our method has not beaten the state of the art method in automatic evaluation metrics (BLEU), the human evaluation shows the effectiveness of our method in transforming style with preserving meaning and fluency. We believe BLEU scores are not a good metric for comparison in unsupervised tasks; comparing the transformed sentence with a golden sentence generated by a human being will not lead to a fair comparison as they might be different correct transformed sentences with different used sets of words. Human evaluation is the only proper comparison in this task in which our method performed well. A better method of human evaluation would be giving participants sentences generated with different methods and want them to sort the sentences from the best to the worst. This method of evaluation was not available to us because of limited time for this project. It can be considered as one of the tasks for future work. In the examples shown in Tables X and XI, our method has performed as good as state-of-the-art method in those examples. A valid conclusion can be drawn only after performing a proper Human Evaluation.

The main advantage of our method over other methods is that it demands minimal computational cost during training; two components out of three are pre-trained and required minimum adjustments. The second component also consists of four non-complex and shallow networks and even with those simple structure, it could achieve an acceptable performance.

Some paths for future work come to our mind:

1) Using more sophisticated language models such as GPT-2 instead of BERT with language model head. BERT has not been designed to be used as a decoder and the error introduced by BERT is one of the main reasons for

[2] https://github.com/fastnlp/style-transformer

[3] https://www.dropbox.com/sh/qj0bvbc6u9yj7qi/AACswIYtbFEn3aTQ8ypcCA18a?dl=0

TABLE VIII
BLEU Scores

Method	Transformed BLEU	Self BLUE
RetrieveOnly (Li et al., 2018) [8]	0.4	0.7
TemplateBased (Li et al., 2018) [8]	13.7	44.1
DeleteOnly (Li et al., 2018) [8]	9.7	28.6
DeleteAndRetrieve (Li et al., 2018) [8]	10.4	29.1
ControlledGen (Hu et al., 2017) [1]	14.3	45.7
CrossAlignment (Shen et al., 2017) [2]	4.3	13.2
MultiDecoder (Fu et al., 2018) [4]	9.2	37.9
CycleRL(Xu et al., 2018) [7]	2.8	7.2
StyleTransformerConditional (Dai et al., 2019) [13]	17.1	45.3
StyleTransformerMultiClass (Dai et al., 2019) [13]	**20.3**	**54.9**
Ours	8.7	28.6

TABLE IX
Percentage of Successful Transformations
(Successful in Preserving meaning, Fluency and achieving target style)

	Positive Reviews To Negative	Negative Reviews to Positive
Train	62	50
Test	47	54

TABLE X
Comparing output of different methods
(Negative to Positive))

Input	the food 's ok , the service is among the worst i have encountered .
DAR	the food 's ok , the service is among great and service among .
CtrlGen	the food 's ok , the service is among the randy i have encountered .
StyleTrans	the food 's delicious , the service is among the best i have encountered .
Ours	the food is excellent , and service is certainly the best i have encountered .
Human	the food is good , and the service is one of the best i 've ever encountered .
Input	always rude in their tone and always have shitty customer service !
DAR	i always enjoy going in always their kristen and always have shitty customer service !
CtrlGen	always good in their tone and always have shitty customer service !
StyleTrans	always nice in their tone and always have provides customer service !
Ours	always nice in their tone , always having excellent customer service !
Human	such nice customer service , they listen to anyones concerns and assist them with it .

bad performance in some cases. With BERT, we only can achieve style transformation through paraphrasing. This structure will not work for more complex style transformation tasks, for example changing a poem to text and vice versa. For those task, we need to use a structure designed to be a proper decoder (e.g. GPT-2).

2) The auto-encoders used as the second component are not deep enough to capture the complicated manifold of each style. Using a deeper structure may improve the results drastically, but it requires to be fed by bigger dataset to avoid over-fitting.
3) Human evaluation scheme can be improved as suggested. Amazon Mechanical Turk can be used to sort all the sentences transformed by different methods based on human's preference.

References

[1] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, "Toward controlled generation of text," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1587–1596.

[2] T. Shen, T. Lei, R. Barzilay, and T. Jaakkola, "Style transfer from non-parallel text by cross-alignment," in *Advances in neural information processing systems*, 2017, pp. 6830–6841.

[3] V. John, L. Mou, H. Bahuleyan, and O. Vechtomova, "Disentangled representation learning for text style transfer," *arXiv preprint arXiv:1808.04339*, pp. 1301–1310, 2018.

[4] Z. Fu, X. Tan, N. Peng, D. Zhao, and R. Yan, "Style transfer in text: Exploration and evaluation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

TABLE XI
COMPARING OUTPUT OF DIFFERENT METHODS
(POSITIVE TO NEGATIVE))

Input	everything is fresh and so delicious !
DAR	small impression was ok , but lacking i have piss stuffing night .
CtrlGen	everything is disgrace and so bland !
StyleTrans	everything is overcooked and so cold !
Ours	everything is not fresh and bad .
Human	everything was so stale .
Input	these two women are professionals .
DAR	these two scam women are professionals .
CtrlGen	shame two women are unimpressive .
StyleTrans	these two women are amateur .
Ours	these two women not professionals ;
Human	these two women are not professionals .

[5] Y. Zhang, N. Ding, and R. Soricut, “Shaped: Shared-private encoder-decoder for text style adaptation,” *arXiv preprint arXiv:1804.04093*, 2018.

[6] Z. Zhang, S. Ren, S. Liu, J. Wang, P. Chen, M. Li, M. Zhou, and E. Chen, “Style transfer as unsupervised machine translation,” *arXiv preprint arXiv:1808.07894*, 2018.

[7] J. Xu, X. Sun, Q. Zeng, X. Ren, X. Zhang, H. Wang, and W. Li, “Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach,” *arXiv preprint arXiv:1805.05181*, 2018.

[8] J. Li, R. Jia, H. He, and P. Liang, “Delete, retrieve, generate: A simple approach to sentiment and style transfer,” *arXiv preprint arXiv:1804.06437*, 2018.

[9] S. Prabhumoye, Y. Tsvetkov, R. Salakhutdinov, and A. W. Black, “Style transfer through back-translation,” *arXiv preprint arXiv:1804.09000*, 2018.

[10] G. Lample, S. Subramanian, E. Smith, L. Denoyer, M. Ranzato, and Y.-L. Boureau, “Multiple-attribute text rewriting,” 2018.

[11] G. Lample, M. Ott, A. Conneau, L. Denoyer, and M. Ranzato, “Phrase-based & neural unsupervised machine translation,” *arXiv preprint arXiv:1804.07755*, 2018.

[12] Z. Yang, P. Cai, Y. Feng, F. Li, W. Feng, E. S.-Y. Chiu, and H. Yu, “Generating classical chinese poems from vernacular chinese,” *arXiv preprint arXiv:1909.00279*, 2019.

[13] N. Dai, J. Liang, X. Qiu, and X. Huang, “Style transformer: Unpaired text style transfer without disentangled latent representation,” *arXiv preprint arXiv:1905.05621*, 2019.

[14] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” *arXiv preprint arXiv:1908.10084*, 2019.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.

[17] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato, “Unsupervised machine translation using monolingual corpora only,” *arXiv preprint arXiv:1711.00043*, 2017.

[18] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[19] A. Wang and K. Cho, “Bert has a mouth, and it must speak: Bert as a markov random field language model,” *arXiv preprint arXiv:1902.04094*, 2019.