Think Positive! - Sentiment Style Transfer

Akanksha Gupta
akankshagupt@cs

Abhishek Somani
asomani@cs

Deep Chakraborty
dchakraborty@cs

1 Problem statement

We are attempting to solve the problem of sentiment transfer in text. This involves changing the sentiment of a given sentence from its source sentiment to a target sentiment label. The model uses unpaired training data, i.e., it doesn't contain pairs of sentences with source and target sentiments. The challenge is to learn a cross-sentiment translation model from such data whilst preserving the non-sentimental parts of the sentence. This is an interesting problem especially in the context of social media, where such a setup can be used to suggest improvements to typed comments/posts in terms of extreme sentiments / explicit language usage.

The goal of the current system is to learn a model that takes as input $(x_{v^{src}}, v^{tgt})$ where x is a sentence that exhibits v^{src} as its attribute, and outputs a sentence $y_{v^{tgt}}$ that has the same content as x, but exhibits target attribute v^{tgt}. Attributes can be $\{positive, negative\}$ in case of sentiment analysis. We plan to approach this problem using a neural style transfer model with encoder-decoder sequence-to-sequence architecture as proposed in Li et al. (2018). We propose to generate noisy paired data once the model is partially trained, and use it as a data augmentation strategy. This added weak supervision will serve to enforce consistency between the input sentence and the same sentence converted back to the source attribute after being converted to the target attribute, i.e., $x_{v^{src}} \equiv x'_{v^{src} \leftarrow v^{tgt} \leftarrow v^{src}}$. We expect that the model will learn additional information for sentiment translation and reduce the dependency on context obtained through deleting the attribute markers, as in the original approach. Finally, we evaluate the results using an automatic LSTM based sentiment classifier and manual evaluation for conversion accuracy and grammaticality of produced sentences.

2 Proposed vs. Accomplished

As proposed, we implemented the baseline model for our task based on the approach provided in Li et al. (2018). Also, We initially proposed to build a system comprising of two identical weight sharing models. The model would learn to take input $(x_{v^{src}}, v^{tgt})$, and output a sentence $(y_{v^{tgt}}$ that has the same content as x but exhibits target sentiment. Then it world take as input $(y_{v^{tgt}}, v^{src})$ and output a sentence x' that has the same content as y but exhibits original source sentiment. The loss function was supposed to maximize the similarity between x and x'. This is inspired from the cycle-consistency loss in Unpaired Image Style Transfer (Zhu et al., 2017) and would ensure that the sentence with the source sentiment attribute can be reconstructed from the converted sentence with target sentiment attribute. However, we realized that the generation of sentence $y_{v^{tgt}}$ based on the input $(x_{v^{src}}, v^{tgt})$ requires us to perform beam search which is difficult to back-propagate through. Hence, we modified the above idea by taking the partially trained baseline model, and using the sentences in the training corpus $x_{v^{src}}$ to generate sentences $y_{v^{tgt}}$ exhibiting target attribute $v^{tgt} \neq v^{src}$ as a means of generating noisy paired data. We then feed these sentences $y_{v^{tgt}}$ back into the model with $v^{tgt} = v^{src}$ of the original sentence x, and further train the model to get $x' \equiv x$ exhibiting v^{src}. We were able to successfully run experiments and ablation studies using this approach, and our approach showed promising results.

3 Related work

The problem of style transfer has been extensively tackled in the domain of images. However, in the domain of natural language, the research has been very active in the recent years. Li et al. (2018) propose a "Delete, Retrieve, Generate" architecture in which first, the phrases indicating the source sentiment label are identified and deleted, followed by retrieving similar sentences from the corpus containing target sentiment, and finally followed by a recurrent neural network to reconstruct sentences. The benefits of this approach are rapid training time compared to adversarial training. The disadvantage is that it uses retrieval mechanisms which cannot not generalized. Shen et al. (2017), explored style transfer for sentiment modification, decipherment of word substitution ciphers and recovery of word order. They used a Variational Autoencoder (VAE) as the base model and used an adversarial network to align different styles. Their evaluation only considered the classification accuracy which is a disadvantage in our case. Also, it uses adversarial network which would be difficult to scale. The Fu et al. (2017) paper proposed models is to learn separate content representations and style representations using adversarial networks. This concept is very similar to the methods applied for style transfer in images. Also, it proposes two novel evaluation metrics that measure two aspects of style transfer based on transfer strength and content preservation. Their mainly argue content preservation is another indispensable evaluation metric for style transfer. In Hu et al. (2017), the authors propose to use a generative model that learns interpretable latent representations and generates sentences with specified attributes. Their approach combines a VAE with attribute discriminators and imposes independence constraints on these attributes. This VAE with wake-sleep algorithm is useful for weak supervision by leveraging fake samples as extra training data. In Prabhumoye et al. (2018), the authors first learn a latent representation of the input sentence which is grounded in a language translation model in order to better preserve the meaning of the sentence while reducing stylistic properties. Then, adversarial generation techniques are used to make the output match the desired style. In Logeswaran et al. (2018), the authors use two different loss namely reconstruction loss to ensure that the model generates content compatible sentences by interpolating between auto-encoding and back-translation loss components and adversarial loss to enforce generated samples to be attribute compatible and realistic. In Zhang et al. (2018), similar to our idea, the authors use the style-preference information and word embedding similarity to produce pseudo parallel data using a statistical machine translation (SMT) framework. Later, a iterative back-translation approach is employed to jointly train two neural machine translation (NMT) based transfer systems. To control the noise generated during joint training, a style classifier is introduced to guarantee the accuracy of style transfer and penalize bad candidates in the generated pseudo data. The application of style transfer is really evident in the context of social media. In Santos et al. (2018), the authors build a model for converting offensive text by training an encoder-decoder framework that combines a collaborative classifier, attention and cycle consistency loss using non-parallel data.

We borrow different modules from different papers and build a end to end system specific to sentiment style transfer.

4 Your dataset

Our problem involves changing the sentiment of a given sentence from its source to a target sentiment label. Since we did not have parallel data of positive and negative sentences, we used the existing data-sets for sentiment analysis. Our approach is a two step process. First step involves generation of context sentence based on the source sentence and the second step requires generation of a sentence of target sentiment while maintaining the source context. The sentiment analysis datasets contains a corpus of positive and negative sentences. Let's say we have a sentence from the positive corpus. We first generate a context sentence based on our first step's algorithm. Next, during the next step, we generate a positive sentence based on the context sentence. We use the original positive sentiment sentence as our ground truth. To start with, we are using the Yelp dataset. The statistics are presented in Table 1.

Table 1: Yelp dataset statistics

#reviews			Vocab
split	negative	positive	
train	177218	266041	9592
validation	2000	2000	
test	500	500	

Table 2 describes a few examples from the Yelp dataset. In first two examples of each sentiment, the sentiment attribute is an adjective or a verb which can be considered easy to transform to the target sentiment. However, the third example describes a sentence with double negation making them difficult for transformation.

In terms of challenges, Firstly, We consider that the result of first step (delete) as a good representation of context and train our model to generate original sentence given the original sentence's sentiment. However, we don't have a method to validate the result of first step since we also don't have any sort of paired data of sentence and it's context. Secondly, our problem is the lack of parallel data. So, we will not be training for transformation of positive sentiment sentences to negative sentiment sentences and vice versa. However, our testing process would involve such transformations.

Table 2: Few examples from the dataset

Sentiment	Example
Positive	my parents love their burgers
Positive	great place for lunch or bar snacks and beer
Positive	the food was not super but good prices reasonable
Negative	the corn lacked butter
Negative	slow , over priced , i 'll go elsewhere next time
Negative	service was n't too bad - nice people .

In the data preprocessing stage, we append a BEGIN and END token to every sentence. Also, all numbers are assigned the same token and we don't use any sort of stop words.

5 Baselines

For our baseline, we implemented the **DeleteOnly** neural model described in Li et al. (2018). At a high level, the model first finds target attributes containing the source label v^{src} and deletes them. It then embeds the content $c(x, v^{src})$ into a vector using a Recurrrent Neural Network (RNN) encoder. It then concatenates the final hidden state with a learned embedding for v^{tgt}, and feeds this into an RNN decoder to generate y. The decoder attempts to produce sentences containing words indicative of the source content and target attribute, while remaining fluent. The detailed description of each of these sections including the training and inference procedure is described in the following sections.

Table 3: Model Parameters

Name	Dimension
Word embedding	9596 x 128
Style embedding	2 x 128
Encoder - gru - W	128 x 1536
Encoder - gru - U	512 x 1024
Encoder - gru - U_{rh}	512 x 512
Decoder - gru - W	1152 x 1536
Decoder - gru - U	512 x 1024
Decoder - gru - U_{rh}	512 x 512
Dense Layer - W_t	1164 x 2328
Dense Layer - W_o	128 x 1164

5.1 Delete Step

We implemented a simple method to delete attribute markers (n-grams) that have the most discriminative power. Formally, for any $v \in V$, we define the salience of an n-gram u with respect to v by its (smoothed) relative frequency in D_v

$$s(u, v) = \frac{count(u, D_v) + \lambda}{(\sum_{(v' \in \mathcal{V}, v \neq v')} count(u, D')) + \lambda} \quad (1)$$

where $count(u, D_v)$ denotes the number of times an n-gram u appears in D_v, and λ is the smoothing parameter. In our implementation, we used $\lambda = 1$. We declare u to be an attribute marker for v if $s(u, v)$ is larger than a specified threshold γ. In our implementation, for yelp dataset, we used $\gamma = 15$. The attributed markers can be viewed as discriminative features for a Naive Bayes classifier.

We define $a(x, v^{src})$ to be the set of all source attribute markers in x, and define $c(x, v^{src})$ as the sequence of words after deleting all markers in $a(x, v^{src})$ from x. For example, for "The chicken was delicious," we would delete "delicious" and consider "The chicken was. . . " to be the content.

5.2 Generate Step

The generate step accepts the sentence context from the Delete step and uses an encoder-decoder framework to generate a sentence containing target attribute markers. The model is illustrated in Figure 1. The model parameters and their sizes are presented in Table 3.

5.2.1 Encoder

The encoder is used to encode information from all the context vectors and target attribute style into a single vector for later use by the decoder.

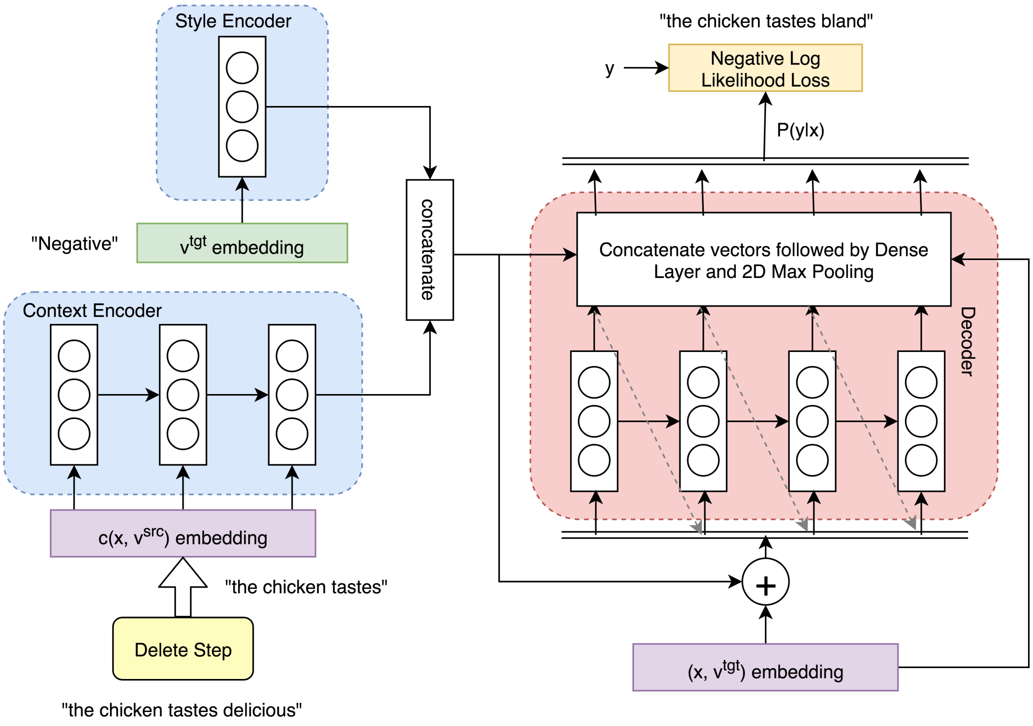

Style Encoder
"Negative"
v^{tgt} embedding
Context Encoder
$c(x, v^{src})$ embedding
"the chicken tastes"
Delete Step
"the chicken tastes delicious"
concatenate
"the chicken tastes bland"
y
Negative Log Likelihood Loss
P(y|x)
Concatenate vectors followed by Dense Layer and 2D Max Pooling
Decoder
+
(x, v^{tgt}) embedding

Figure 1: Model architecture with delete step followed by context and style encoder, followed by decoder for generating sentences with target attribute

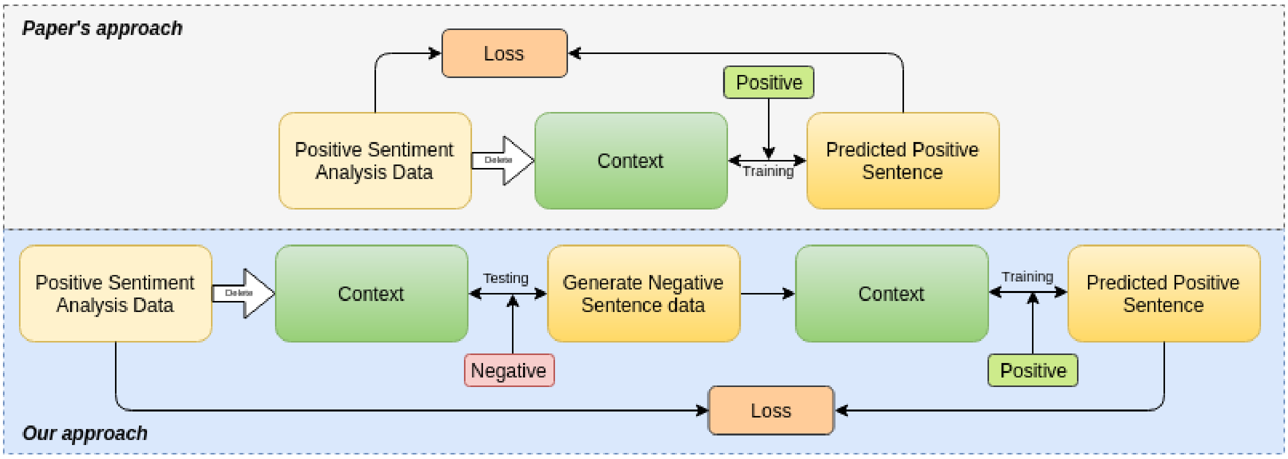
The encoder is implemented as a Gated Recurrent Unit (GRU) (Chung et al., 2015) with 512 hidden states. It takes as input the non-sentimental content $c(x, v^{src})$ of the sentences in the training corpus obtained from the previous step encoded by an embedding layer of size 128 that is initialized randomly and learned jointly during training. We then concatenate the 512-dimensional final hidden state of this recurrent network with a 512-dimensional embedding of the target attribute v^{tgt} to give a 1024-dimensional vector which is fed into each time-step of the decoder.

5.2.2 Decoder

The decoder is again implemented as a GRU with 512 hidden states. It accepts as input the corresponding sentences in the training corpus containing their original sentimental attributes prior to the delete step encoded using the same word embedding layer used in the encoder. However, the word embedding are also concatenated with the merged hidden state and style embedding from the encoder, to produce a $(128 + 1024) = 1152-$dimensional vector which is fed into each time-step of the decoder. Each 512-dimensional hidden step output of the decoder is then concatenated with the 1024-dimensional hidden state vector from the encoder as well as the 128-dimensional embedding of the target word going into each time-step, to yield a 1664-dimensional vector output at each time-step. The intuition behind doing this is to give encoder context at each word level to the decoder along with information about word embeddings. These concatenated vectors are then passed into a time-distributed dense layer of 128 units, and the 128-dimensional output vectors of this layer at each time-step are then used to compute a dot-product with the word embedding matrix of the entire vocabulary of size (128, vocab_size) to give similarity scores over each of the words in the vocabulary, which are then converted into probabilities using a softmax layer. This tells us the most likely word at each time-step.

5.3 Training and Loss

We use a negative log likelihood loss over the outputs of the decoder layer and train using the Adadelta optimizer (Zeiler, 2012) with a learning rate of 0.0001 and decay of 0.95. At training time, we do not have access to ground truth outputs that express the target attribute. Therefore the model is



Paper's approach
Loss
Positive
Positive Sentiment Analysis Data
Delete
Context
Training
Predicted Positive Sentence
Positive Sentiment Analysis Data
Delete
Context
Testing
Generate Negative Sentence data
Context
Training
Predicted Positive Sentence
Negative
Positive
Loss
Our approach

Figure 2: Our approach for generating noisy paired data and further training the model

trained to reconstruct the sentences in the training corpus given their content and original attribute value by maximizing:

$$L(\theta) = \sum_{(x,v^{src})\in\mathcal{D}} log(P(x|c((x,v^{src})),v^{src});\theta) \tag{2}$$

The training process follows teacher forcing which basically means that the target words of the decoder are the same as the input words to the decoder, offset by one time-step into the future.

5.4 Inference

Based on Beam Search proposed in (Graves, 2012) and (Boulanger-Lewandowski et al., 2013) for Neural Machine Translation, our implementation uses a simple left-to-right beam-search decoder to generate new translations that approximately maximize the trained conditional probability. It generates the target sentence word by word from left-to-right while keeping 10 active candidates (beam size) at each time step.

6 Our approach

Currently, the limitation of the baseline approach is that at training time, the model is trained to reconstruct the sentences in the training corpus given their content $c(x, v^{src})$ and original attribute value v^{src}, because of lack of paired data. We trained the baseline model for 50 epochs and then generated the psuedo parallel sentences (x, v^{src}) exhibiting target attribute v^{tgt} different from v^{src}. This noisy paired data is augmented with the training corpus for our approach. We then feed these sentences back into the model and further train the model for another 50 epochs. To demonstrate that the results in our method are not due to training for higher number of epochs, we trained the paper's approach for total 100 epochs and then performed evaluation on the results from this ablation study. The idea of generating paired data is inspired from the cycle-consistency loss in unpaired image style transfer (Zhu et al., 2017) that ensures that the sentence with the source sentiment attribute can be reconstructed from the converted sentence with target sentiment attribute. However, we cannot exactly implement the cycle-consistency loss as the similarity between reconstructed and original sentence, since the sentences at the output of the decoder are generated using beam-search which is difficult to back-propagate through. Therefore, we simply provide the source sentence as the target for the reconstructed sentence at the decoder. Our approach is described concisely in Figure 2.

7 Results and Error analysis

The performance of sentiment translation in text can be a hard task to analyze, especially since there can be many correct translations and standard tests cannot capture that. Our problem statement requires us to generate meaningful sentences with target attribute different from the source attribute without losing the context. So, we have two major requirements that need to be evaluated. First, that the generated sentence's attribute is different from the source sentence. Second, that the generated sentence is syntactically and semantically sound. We perform both automatic and human analysis to evaluate these criteria.

7.1 Automatic Evaluation

To assess the first requirement, we compiled generated results of Yelp's test corpus with attribute different from the source sentence. On this corpus, we performed sentiment analysis using two different approaches. In our first approach, we use a Naive Bayes sentiment classifier. We find that uni-gram Naive Bayes method does not capture long range dependencies. Also, it does not capture the context. For example, the source sentence - "the wine was very average and the food was even less", generated the result - "the wine was just right on the wine and the food was just less". Even though the result seemed slightly positive, it classified the attribute as negative. Hence, we believe, this approach is not suitable to evaluate our metric. Therefore, we omit the results from this method in our final table. In the second approach, we used a pre-trained model of a sentiment classifier trained on the Yelp training set which encodes a sentence into a vector using a bidirectional LSTM. It then performs average pooling of outputs and minimizes the logistic loss. Our metric is defined as the percentage of results whose attribute has been identified different from the source using the above sentiment analysis approaches. We performed evaluation on the test corpus which contains 500 sentences with positive attribute and 500 sentences with negative attribute, with the target conversion to both positive and negative sentiment. Some examples are presented in Tables 4 and 5.

To assess the second requirement, i.e., grammaticality of the converted sentences, we use the BLEU metric. It captures the similarity between two sentences. We first generate sentences with target attribute same as the attribute of the source sentence. Next, we compute BLEU score between our generated results and the test corpus. Some examples of our results are presented in the tables 6 and 7. A high BLEU score indicates that the output generated by the system is preserving the content by retaining the context words (constructed after removing attribute markers from the source sentence) as well as the attribute markers. Based on test corpus containing 500 sentences with positive attribute and 500 sentences with negative attribute, the resulting BLEU metric is 33. The score is high in accordance to the training process since we are construct the output while preserving the content. In the Table 6, Example 1 and 2 retain most of the words while example 3 does not. In our results, most of the generated sentences are similar to 1 and 2 therefore the BLEU score is high. Given that, the source attribute is same as the target attribute, the high BLEU score could also be a result of an inefficient delete step. Hence, we perform a similar analysis between our generated results and the online available human references for Yelp dataset. The scores are present in Table 8. According to both the analyses, our method performs better compared to the paper for all possible source to target sentiment transformation except the case where both the source and target attribute are positive.

7.2 Human Evaluation

The automatic evaluation methods are not sufficient to capture the incoherent nature of the inputs. Therefore, we did manual evaluation of 200 sentences, 50 each from the four types of conversions "Positive to Negative", "Positive to Positive", "Negative to Positive", "Negative to Positive". We performed two types of evaluations: sentiment labeling and checking for grammaticality and similarity to the source content, and scored it on a scale of 1 to 3 denoting "bad, neutral and good" respectively. We did our evaluation agnostic to the approach to avoid biasing ourselves towards our results. Last four rows of table 8 shows the human evaluation results. The relative scoring is consistent with the automatic methods i.e., there is correlation between automatic method and human evaluation. For sentiment transfer, our method is performing better for positive to negative and negative to negative, and for producing grammatically correct sentence and preserving the source content our method is performing better for all except positive to positive.

7.3 Qualitative Analysis and Discussion

Although according to automatic evaluation, for negative, our approach is performing worse than paper's approach in terms of transferring sentiment, manual analysis shows that the margin is not as huge as the results say. One reason could be that our system is trying to produce positive reviews of something which is always negatively discussed in the training data. Hence, the LSTM classifier is giving unreliable predictions for these outputs. For example, for the first row in Table 9, our method failed. It is probably because there were about 138 sentences related to 'pharmacy' in

Table 4: Some examples for negative to positive sentiment transformation

Input	Ours	Paper
it is n't terrible , but it is n't very good either	it is n't terrible , but it is also very simple and excellent	it is n't amazing but it is reasonably priced
this place is a terrible place to live !	this place is a total treasure !	this place is so worth checking !
the beer was nice and cold !	the beer was also very good !	the beer was nice and the food was cold and romantic !

Table 5: Some examples for positive to negative sentiment transformation

Input	Ours	Paper
i will definitely return often !	will not be returning again any time soon go elsewhere !	i hate this place coming here often and less than often !
the salads were fresh and crispy	the salads were sour and were not even crispy	the salads were ok , not crispy
the staff are very friendly and on the ball	and that is not the ball	it was really good on the ball

Table 6: Some examples for negative to negative sentiment transformation

Input	Ours	Paper
had to wait a month to get in .	had to wait another month to get in .	it was so loud i had to wait a month to get in .
i said it was disgusting to even serve this to diners .	i should never have said even to serve diners .	i said serve this place to go to diners .
the queen bed was horrible !	should have been a negative queen bed disgusting .	the queen did not really smell better !

Table 7: Some examples for positive to positive sentiment transformation

Input	Ours	Paper
the food all looked great	everyone enjoyed the food all looked great	the food was fast and the service is fast
the octopus sashimi is my favorite !	the octopus sashimi is my mouth !	the octopus sashimi is these great flavor
the food was good , steak bites and hummus plate a must .	the food was good , the steak and hummus plate was mediocre .	the food was good , steak bites and hummus plate are great .

the negative training data as compared to 60 in positive. One may then ask why the paper's approach is doing better. Based on manual analysis, we saw that in most of the sentences where our method fails, the paper's method is just forcefully trying to make the output positive without trying to preserve the source content. Also, for some other sentences where our method fails, we observed that the paper's approach changes the sentiment but does not produce grammatically sound sentences. We present some examples in Table 9. Further, this is also reflected in BLEU metric analysis and human evaluation. Both of them assign high score to our method as compared to the paper's method. A similar observation has been made in the case where both the source and target attribute are positive. Some examples are presented in the table 10 justifying the better performance of the paper's approach. We observed that our approach works well for positive to negative and negative to negative sentiment transformation based on all evaluation methods.

Table 8: Evaluation results using automatic and human evaluation

Approach	Pos to Neg	Pos to Pos	Neg to Pos	Neg to Neg
bidirectional LSTM (OURS)	**89**	73.2	33.6	**98.2**
bidirectional LSTM (PAPER)	65.4	**97.6**	**82.2**	81
BLEU b/w input output (OURS)	**0.262**	0.226	**0.258**	**0.289**
BLEU b/w input output (PAPER)	0.234	**0.255**	0.242	0.250
BLEU with human ref (OURS)	**0.0710**	0.0707	**0.0829**	**0.0860**
BLEU with human ref (PAPER)	0.0706	**0.0778**	0.0826	0.0826
Human Classification (OURS)	**80.65**	87.09	65.51	**96.78**
Human Classification (PAPER)	45.17	**93.54**	**93.10**	67.75
Human Grammatically/Similarity (OURS)	**2.35**	2.32	**2.2**	**2.22**
Human Grammatically/Similarity (OURS)(PAPER)	2.19	**2.58**	2	2.19

Table 9: Examples justifying paper's better results in the case of negative to positive sentiment transformation

Input:	i can't believe how inconsiderate this pharmacy is
Ours	this pharmacy is absolutely the worst !
Paper	this pharmacy knows great food and service is incredibly reasonable.
Input:	it isn't terrible , but it isn't very good either
Ours	it isn't terrible, but it is also very simple and excellent
Paper	it isn't amazing but it is reasonably priced
Input:	new owner , i heard - but i don't know the details
Ours	new owner , i heard - but i know what the details is
Paper	new owner , i heard great service - the details .
Input:	we sit down and we got some really slow and lazy service
Ours	we sit down and we got some service
Paper	we sit down for a quick immediately and we have some service

Table 10: Examples justifying paper's better results in the case of positive to positive sentiment transformation

Input:	it 's small yet they make you feel right at home
Ours	it 's small yet so comfortable and they make you feel right at home
Paper	it 's small yet they make you delicious

8 Contributions of group members

All members were involved in development of all modules, error analysis, human evaluation and report writing. However, primary responsibilities of each group member are listed below:

- Akanksha Gupta: did literature survey, implemented delete step and automated evaluation methods, did overall code design.
- Abhishek Somani: did data pre-processing and implemented the paper's model, ran experiments using this implementation, did manual error analysis.
- Deep Chakraborty: Conceptualized our approach, implemented and experimented with modifications for the proposed idea, did error analysis and overall paper organization.

9 Conclusion

In this paper, we tried to solve the problem of sentiment style transfer in text while preserving the content. We implemented a baseline model that deletes sentiment markers from input text and provides its context to an encoder, along with the desired target sentiment, followed by a decoder to generate output sentences with the desired target sentiment. While the baseline model uses completely unpaired data for training, we propose to improve it by generating noisy pairs from a partially trained model, and further training the model on these pairs as a data augmentation strategy to improve sentiment translation by implicitly enforcing cycle consistency. Our approach outperforms the paper's approach for sentiment conversion from positive to negative and negative to negative sentences. However, our results show that we fail to do better than the paper on negative to positive and positive to positive conversions, although human evaluation shows that converted sentences generated using our approach are more content preserving and grammatical. The results are a bit ironic, since our major aim was to convert negative sentences into positive sentences as depicted in our project title "Think Positive", which our approach failed to accomplish. For future work, we would like to implement the exact cycle consistency loss by back-propagating through a continuous approximation of beam search Goyal et al. (2017). We would also like to try different reconstruction loss functions Logeswaran et al. (2018) and better evaluation metrics Fu et al. (2017) proposed in recent works.

References

Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. (2013). Audio chord recognition with recurrent neural networks. In *ISMIR*, pages 335–340. Citeseer.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2015). Gated feedback recurrent neural networks. In *International Conference on Machine Learning*, pages 2067–2075.

Fu, Z., Tan, X., Peng, N., Zhao, D., and Yan, R. (2017). Style transfer in text: Exploration and evaluation. *CoRR*, abs/1711.06861.

Goyal, K., Neubig, G., Dyer, C., and Berg-Kirkpatrick, T. (2017). A continuous relaxation of beam search for end-to-end training of neural sequence models. *arXiv preprint arXiv:1708.00111*.

Graves, A. (2012). Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. (2017). Toward controlled generation of text. *arXiv preprint arXiv:1703.00955*.

Li, J., Jia, R., He, H., and Liang, P. (2018). Delete, retrieve, generate: A simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*.

Logeswaran, L., Lee, H., and Bengio, S. (2018). Content preserving text generation with attribute controls. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 5108–5118. Curran Associates, Inc.

Prabhumoye, S., Tsvetkov, Y., Salakhutdinov, R., and Black, A. W. (2018). Style transfer through back-translation. *arXiv preprint arXiv:1804.09000*.

Santos, C. N. d., Melnyk, I., and Padhi, I. (2018). Fighting offensive language on social media with unsupervised text style transfer. *arXiv preprint arXiv:1805.07685*.

Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. (2017). Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, pages 6830–6841.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhang, Z., Ren, S., Liu, S., Wang, J., Chen, P., Li, M., Zhou, M., and Chen, E. (2018). Style transfer as unsupervised machine translation. *CoRR*, abs/1808.07894.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*.