A New View of Multi-modal Language Analysis: Audio and Video Features as Text "Styles"

Zhongkai Sun
University of Wisconsin-Madison
zsun227@wisc.edu

Prathusha K Sarma*
Apple
prathyushaks.21@gmail.com

Yingyu Liang
University of Wisconsin-Madison
yliang@cs.wisc.edu

William A. Sethares
University of Wisconsin-Madison
sethares@wisc.edu

Abstract

Imposing the style of one image onto another is called *style transfer*. For example, the style of a Van Gogh painting might be imposed on a photograph to yield an interesting hybrid. This paper applies the adaptive normalization used for image style transfer to language semantics, i.e., the *style* is the way the words are said (tone of voice and facial expressions) and these are style-transferred onto the text. The goal is to learn richer representations for multi-modal utterances using style-transferred multi-modal features. The proposed Style-Transfer Transformer (STT) grafts a stepped styled adaptive layer-normalization onto a transformer network, the output from which is used in sentiment analysis and emotion recognition problems. In addition to achieving performance on par with the state-of-the art (but using less than a third of the model parameters), we examine the relative contributions of each mode when used in the downstream applications.

1 Introduction

Multi-modal language analysis expands textual analysis by utilizing co-occurring acoustic and visual information, and has recently become a popular topic in machine learning (Morency et al., 2011; Baltrušaitis et al., 2018). In both sentiment analysis (Wang et al., 2016; Zadeh et al., 2016) and emotion recognition (Busso et al., 2008; Mittal et al., 2019), the three modalities are combined to better represent the sentiment or emotional meaning of a passage. The idea of combing textual, acoustic, and visual features is obvious: individual modalities are not always able to convey as accurate an impression as multi-modal features, which typically provide more complete information. For instance, Fig. 1 shows a caricature where the sentiment may be easily understood from the audio and video, but not from textual analysis alone.

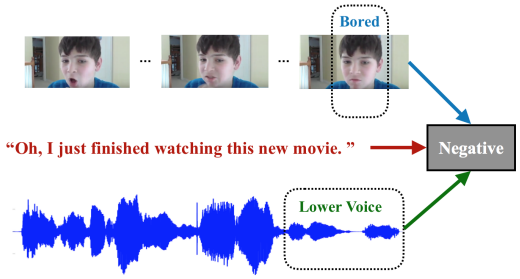

Figure 1: An example of multi-modal sentiment analysis. In this example, textual information is unrelated to the underlying sentiment while acoustic and visual features reflect the perceived sentiment.

Combining textual, acoustic, and visual features can be accomplished in a variety of ways ((Tsai et al., 2018; Zadeh et al., 2018a; Liang et al., 2018; Mai et al., 2019; Sun et al., 2019b; Mittal et al., 2019)). Among these, (Wang et al., 2018b; Sun et al., 2019a) consider text as the backbone and study methods to inject acoustic and visual information into the textual features that are typically extracted via pre-trained word/language models, e.g., Glove (Pennington et al., 2014), ELMO (Peters et al., 2018), and BERT (Devlin et al., 2018).

This paper also uses textual features as a backbone and studies a novel way of injecting non-text features into a primarily text-only model. A *style* vector is learned from the acoustic and visual features. This style vector is then transferred to a text input transformer encoder via **S**tepped **Ad**aptive **La**yer **N**ormalization (SAdLaN). While adaptive style transfer in image processing literature is well studied (Karras et al., 2018; Huang and Belongie,

* Work done while at UW-Madison

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 1956–1965
April 19 - 23, 2021.

2017; Park et al., 2019), the novelty in our work is to consider the corresponding audio and video to be the *style* of the text and transfer the non-verbal features' information to text-based models via the effective SAdLaN. Concretely, our model replaces the original layer normalization in the text transformer encoder with the proposed SAdLaN, which learns style scale and style bias from non-text features. The proposed model is named as Style-Transfer Transformer (STT) and it is tested on three benchmark datasets.

This paper makes three contributions. First, our model's performance is on par with the state-of-the-art but using only less than one third of the model parameters. Since our model does not require training a multimodal transformer from scratch to achieve the same results, the style transfer method benefits from both reduced model size and training time. Second, we introduce the Stepped Adaptive Layer Normalization (SAdLaN), which performs adaptive normalization as a function of the layer of the DNN encoder. Third, we study the contributions of each modality towards use in our downstream applications. While we know that multimodal embeddings would contribute more than each individual modality, such an examination highlights the relative strength of each mode, particularly on sentiment classification tasks.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 describes our proposed method Style Transfer Transformer (STT). Section 4 and 5 presents experimental results, and Section 6 concludes this work.

2 Related Work

Multi-modal Language Analysis: Previous work on multi-modal language analysis typically learns a novel network structure to capture interactions between text, audio, and video features via supervised learning. Liu et al. (2018) develop an efficient low-rank tensor fusion mechanism to learn the outer-product representation of multi-modal features, while (Zadeh et al., 2018a) learn a multi-modal memory gate that is applied to an LSTM to capture the flow of information in the different modalities. In their work (Liang et al., 2018) learn a specialized multi-modal fusion model by applying a novel multistage fusion in the recurrent network. Recently, the multi-modal transformer introduced by (Tsai et al., 2019) achieves the state-of-the-art performance by using bi-directional cross modal relationships between the different modalities. Note that the transformer architecture in STT is the same as in (Tsai et al., 2019). However, we depart from their modeling procedure by encoding audio and video jointly using a single bimodal transformer block, thereby eliminating additional cross modal attention blocks.

Adaptive Normalization: Adaptive normalization is widely used in image processing and computer vision. Huang and Belongie (2017) proposes adaptive instance normalization, which learns the affine parameters from the style vector to perform a style transfer in the image encoder's feature space. Karras et al. (2018) applies the AdIN in the generative adversarial network (GAN) to generate fake multi-styled human faces. Park et al. (2019) proposes a spatially-adaptive normalization for image semantic synthesis. Wang et al. (2018a) applies a spatial feature transformation for image super-resolution. This paper introduces the SAdLaN framework in which a bimodal style vector is transferred as a function of the DNN's depth.

Adapter Method in Pre-trained Language Models: Due to the enormous size of pre-trained language models (e.g. BERT, XLNet), the procedure of fine-tuning the models on downstream data may be inefficient. To address this "adapter" based methods have been developed. In their work (Houlsby et al., 2019) apply the task-specific layer at each encoder layer of BERT. Each task-specific layer contains two feed-forward projectors: one down-projects (maps the input vector to a low-dimensional space) and the other up-projects (maps the prior layer's output back to its original dimension). During training, only the parameters in these task-specific layers are updated. Stickland and Murray (2019) proposes another method: the task-specific layer is applied between the two layer-normalization in the encoder. Wang et al. (2020) applies parallel adapters which each learn different information from the knowledge base in order to enable the model to deal with the multi-task learning task.

Our proposed SAdLaN is motivated by the the adapter designed in (Houlsby et al., 2019). In their work, the authors find that lower layers of the transformer have less impact on the fine tuning objective of a given task, while higher layers are more vital (this is an intuitive observation, because,

lower layers are more likely to learn semantic meanings while the higher layers are more related to the specifics of the task). When using SAdLaN, lower layers of the STT are changed as little as possible to enable the model to learn basic semantic information from the input text, while the top layers allow a larger effect influenced by the style factors.

3 Style Transfer Transformer (STT)

We define the text, audio, and video features for a given utterance as $f_t \in \mathbb{R}^{l \times d_t}$, $f_a \in \mathbb{R}^{l \times d_a}$, $f_v \in \mathbb{R}^{l \times d_v}$, where l is the length of the modality sequence and d_t, d_a, d_v are embedding dimensions of each modality (each modalities' sequence length is forced to be the same by applying alignment). Figure 2 (*left*) shows an overview of our model STT. It consists of the following four steps:

- Step 1: learn a style vector from the audio and video features.
- Step 2: use the learned style vector during adaptive layer normalization for a text input transformer model.
- Step 3: take the style transferred text representation and pass it through a GRU to get the final multimodal embedding.
- Step 4: use this multimodal embedding for a downstream task.

Step 1 Learn a style vector: In keeping with the hypothesis that the text modality is the major contributor to the learned multimodal embedding, a style encoder is first applied to acoustic (f_a) and visual (f_v) features to learn the non-verbal style vector $f_s \in \mathbb{R}^{l_s}$. The STT first concatenates audio and video sequences at each time step, i.e., $f_{av} \in \mathbb{R}^{l \times (d_a + d_v)}$. The concatenated f_{av} is then input into the transformer model TRANSFORMER_{av}, which is a basic self-attention multi-head transformer encoder (with query = key = value; see (Vaswani et al., 2017)). The final state of TRANSFORMER_{av} is the style vector f_s.

Step 2 Adaptive style transfer onto text: The original layer norm (Ba et al., 2016) is defined as:

$$f_y = \frac{f_x - \mathbb{E}[f_x]}{\sqrt{\text{Var}[f_x] + \epsilon}} \times \gamma + \beta \tag{1}$$

where f_x is the input vector to the layer normalization, ϵ is a value added to the denominator for numerical stability, and γ, β are scalar and bias factors computed from the data for normalization. The style transfer techniques for image processing applications use factors s, b learned from a style vector to replace γ, β (Huang and Belongie, 2017; Karras et al., 2018).

Inspired by these, we propose our Stepped Adaptive Layer Normalization technique (SAdLaN). It also computes factors from our style vector f_s for the normalization, but with the key difference that it takes into account the depths of the layers. Since lower layers of the transformer have less impact on the fine tuning objective for a given task while higher layers are more vital to the task, the lower layers of the STT should be changed as little as possible to enable the model to learn basic semantic information from the input text, while the top layers should allow a larger effect influenced by the style factors.

Figure 2 (*right*) shows how the SAdLaN is applied to a text transformer encoder. Formally, we input the style vector f_s to a MLP layer to compute factors s^i, b^i for the normalization of the ith layer, and introduce a novel stepped ratio factor r_i where r_i gradually increases as a function of the depth of the layer. SAdLaN for the ith layer is then defined as:

$$\begin{aligned} f_y^i = &\left(\frac{f_x^i - \mathbb{E}[f_x^i]}{\sqrt{\text{Var}[f_x^i] + \epsilon}} \times \gamma + \beta\right) \\ &\times (1 + s^i \times r_i) + (b^i \times r_i). \end{aligned} \tag{2}$$

Here γ and β are computed as in the original layer norm (without using the style vector). The s^i, b^i are factors learned from the style vector: at each layer i, the learned style vector f_s is input to a specific MLP to learn the factors s^i, b^i. The r_i are defined as

$$r_i = (i - 1) \times \frac{\text{ratio}}{\#\text{layers}} \tag{3}$$

where ratio is a constant value used to limit the maximum r_i value. Thus, small r_i indicate the style factors have little impact on the layer norm and vice versa. Note that SAdLaN reduces to the original layer norm (1) when $\text{ratio} = 0$.

Step 3 Style transferred multimodal embedding: The output of the last adapted layer from Step 2 is considered to be the style transferred text sequence. This styled text sequence is then passed through a GRU to learn the final multimodal embedding.

Step 4 Downstream applications: Multimodal embeddings learned in Step 3 can be passed to a softmax layer or a MLP for any downstream task of choice.

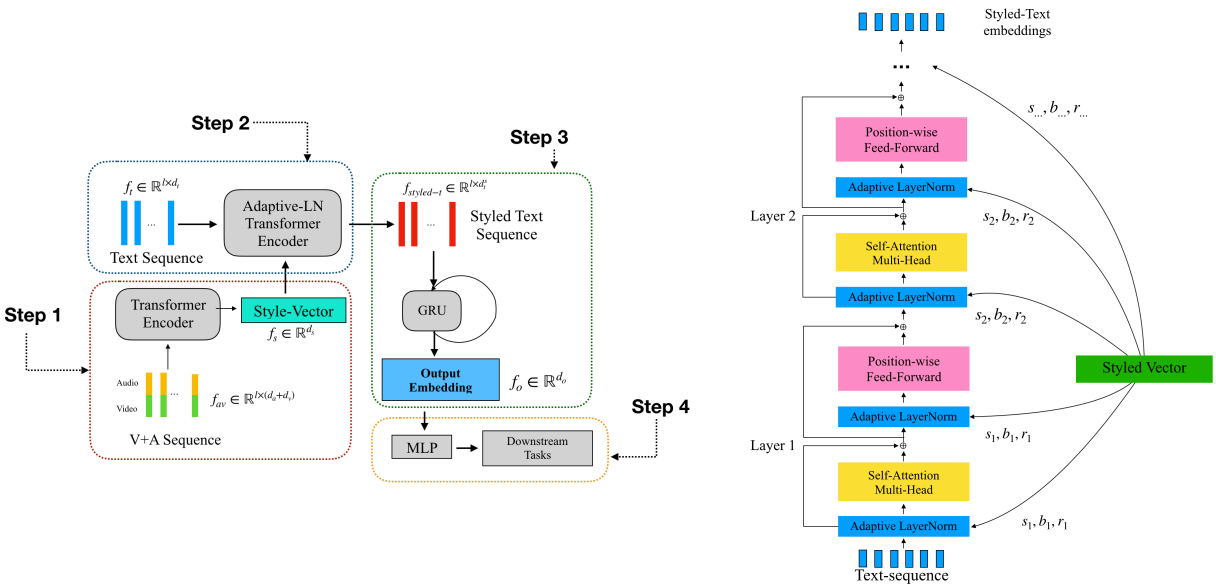

Figure 2: *left:* An overview of the STT model. First, a style vector is learned from audio and video features. Second, the style vector is used along with the text to learn a styled-text sequence by a transformer encoder with the adaptive layer normalization. Finally, use a GRU to get the final multimodal embedding and use a MLP for the downstream task. See the main text for the details. *right:* The STT Transformer encoder with the adaptive layer normalization. Scalar and bias factors are learned from the style vector. Smaller ratios are applied to the bottom layers so that accurate semantic features can be captured, while larger ratios are applied to top layers to encourage the learning of the styled features.

4 Experimental Setup

This section provides a brief overview of the experimental setup, datasets, and baseline methods used for comparison.

4.1 Data sets

In order to test the performance of the STT, three benchmark multi-modal datasets are selected: i) CMU-MOSI, ii) CMU-MOSEI (Zadeh et al., 2016, 2018b), and iii) IEMOCAP (Busso et al., 2008). The first two are standard datasets for sentiment analysis while IEMOCAP is a standard emotion recognition dataset.

- **CMU-MOSI:** This data set contains 2199 utterance-level video segments. Each video segment is labeled with sentiment scores that range from -3 (extremely negative sentiment) to $+3$ (extremely positive sentiment).
- **CMU-MOSEI:** This data set contains 22856 utterance-level video segments. Annotation of each segment is the same as in the CMU-MOSI dataset.
- **IEMOCAP:** The original dataset contains 10000 examples with 9 different emotion annotations. In this paper, we follow (Tsai et al., 2019) and choose four emotions (angry, sad, happy, and neutral) with balanced distributions for the evaluation.

4.2 Baselines

We compare the proposed STT with several baselines: i) Early and Late Fusion LSTM (Zadeh et al., 2016), ii) **RAVEN** (Wang et al., 2018b), iii) **MCTN** (Tsai et al., 2018), iv) **LMF** (Liu et al., 2018), and v) **RMFN** (Liang et al., 2018). The state of the art **Mult** (Tsai et al., 2019), which applies 6 cross-modal transformers to learn cross relationships between modalities, is also included in our evaluations.

To keep comparisons fair, our experiments use the same multimodal features as in (Liu et al., 2018; Zadeh et al., 2016, 2018a; Liang et al., 2018; Tsai et al., 2019). Here are some high level details with regards to each modalities' features:

- **Textual features:** Glove (Pennington et al., 2014) word embeddings of 300 dimensions are used as inputs to obtain textual features.
- **Acoustic features:** They are extracted by COVAREP (Degottex et al., 2014) and have 74 dimensions. These features contain information on frequency, volume, pitch, MFCC, etc.

	MOSI	MOSEI	IEMOCAP
batch size	48	128	48
learning rate	0.002	0.0001	0.003
transformer hidden dim	40	40	40
GRU hidden dim	60	40	60
# of encoder layers	8	8	4
# of heads	8	10	8
style ratio	0.3	0.2	0.25
dropout	0.2	0.5	0.3

Table 1: This table presents the best hyper-parameter settings of the STT model for all data sets reported in our experiments.

They are aligned at the word-level: for every word, its corresponding acoustic feature is the average of all audio frame's features between the start time and the end time of that specific word.

- **Visual features:** They are extracted by Facet (iMotions, 2017) and have 34 dimensions. Features include facial landmarks, action units, etc. They are also aligned at the word level.

In consistency with prior work on these three datasets, several performance metrics are used to evaluate the models. On CMU-MOSI and CMU-MOSEI, binary accuracy, weighted F1 score, mean absolute error, 7-class accuracy, and correlation with human labels are reported. For IEMOCAP, binary accuracy and weighted f1 score are used.

Hyperparameters: We perform grid search on the hyperparameter values. For STT, i) the learning rates of Adam are from 0.0001 to 0.001; ii) dropout ratios are from 0 to 0.5; iii) transformer encoder's hidden layers are of dimensions 40 and 60; iv) hidden states of GRU are of dimensions 40 and 60; v) the numbers of transformer encoder layers are from 4 to 8; vi) the numbers of heads in the multi-head attention layer are 8 and 10; vii) style ratios are from 0.1 to 0.5. The best hyper-parameter settings in our experiments are presented in table 1.

5 Experimental Results

Tables 2 and 3 report results from the baselines and STT on the three benchmark data sets. We can see that STT performs on par with the baselines and SOTA considered in our experiments. Since there exists a disparity between the results reported in (Tsai et al., 2019) and the results obtained from reproducing the source code released by the authors, we report numbers from reproduced Mult as Mult_{rep} and report numbers from the paper as $\text{Mult}_{\text{paper}}$. Note that STT beats the performance of Mult_{rep} while matching the performance of $\text{Mult}_{\text{paper}}$. To keep comparisons fair, we compare STT with the averages of scores attained by $\text{Mult}_{\text{paper}}$ and Mult_{rep} and note that on all three benchmark datasets on average we do better than Mult. While one may argue that reproducing the baseline is essential, we can offer as an explanation that the lack of clarity in reproducibility should not penalize our modeling efforts. To document our experimental evaluations and hypothesis in a credible manner, we report numbers from the paper as well as the reproduction. Furthermore, we note that STT is more efficient than Mult: it achieves better performance using less than a third number of parameters. See Table 4. This is because it uses 2 transformer blocks as opposed to 6 blocks in Mult.

To verify if the performance of STT increases with the number of parameters, we investigate increasing: 1) the dimension of the transformer encoder's hidden layer, 2) the number of encoder layers and number of heads, and 3) the number of layers in the GRU. However, we do not find improvements in performance by increasing the number of model parameters. We posit that this performance limitation is due to the relatively crude features extracted for audio and video analysis and suggest exploration in this direction for improved performance.

5.1 Ablation Studies

In this section we present results from ablation studies performed by i) varying ratio and studying the effect of different ratio values on the transformer's performance, ii) providing as input to the transformer unimodal features for downstream sentiment classification tasks, as well as iii) evaluating the effects of non verbal features.

5.1.1 Study 1: Varying Stepped Ratio

Table 5 presents results from the ablation study that shows the performance of different ratio values in equation (2). When $\text{ratio} = 0$, the model contains only a single transformer using text as the only input and standard layer normalization. This setup is meant to demonstrate the performance of STT when using text alone. "No stepped ratio" means the same style scalar and bias are used for all layers, i.e., $r_i = 1$ for all i in (2). Varying the ratio from 0.1 to 2 investigates the influence of its values.

Table 5 shows the effect of the stepped ratio on

Data View	CMU-MOSI					CMU-MOSEI				
	Acc-2	F-score	MAE	Acc-7	Corr	Acc-2	F-score	MAE	Acc-7	Corr
EF-LSTM	75.9	75.5	1.035	32.7	0.611	77.2	77.5	0.632	46.4	0.623
LF-LSTM	77.8	77.7	1.009	34.8	0.645	79.4	80.2	0.611	48.3	0.666
RAVEN	78.0	76.6	0.915	33.2	0.691	79.1	79.5	0.614	50.0	0.662
MCTN	79.3	79.1	0.909	35.6	0.676	79.8	80.6	0.609	49.6	0.670
RMFN	78.4	78.0	0.922	38.3	0.681	NA	NA	NA	NA	NA
Mult(paper)	**83.0**	**82.8**	0.871	**40.0**	0.698	**82.5**	82.3	**0.580**	**51.8**	**0.703**
Mult(rep)	81.7	81.8	0.874	38.1	0.708	82.0	81.9	0.585	50.8	0.690
STT	82.4	82.2	**0.847**	38.9	**0.733**	82.1	**82.6**	0.586	51.2	0.695

Table 2: Results on CMU-MOSI and CMU-MOSEI. Best numbers are in bold. Note that for MAE, lower is better, while for the other metrics, higher is better.

Data View	IEMOCAP							
Emotions	Happy		Angry		Sad		Neutral	
	Acc-2	F-score	Acc-2	F-score	Acc-2	F-score	Acc-2	F-score
EF-LSTM	85.5	84.8	85.7	83.1	81.2	80.3	66.3	65.3
LF-LSTM	85.4	85.7	83.6	82.9	79.9	80.1	67.1	67.2
RAVEN	87.3	85.8	85.1	84.6	83.8	82.9	69.5	69.1
MCTN	84.9	83.1	79.7	80.4	80.5	79.6	62.3	57.0
RMFN	87.5	85.8	85.1	84.6	83.8	82.9	69.5	69.1
Mult(paper)	**90.7**	**88.6**	**87.4**	87.0	86.7	86.0	**72.4**	**70.7**
Mult(rep)	88.7	86.9	87.0	**87.2**	86.6	86.3	70.6	69.4
STT	88.3	87.8	87.3	87.0	**87.5**	**87.4**	70.1	68.5

Table 3: Results on IEMOCAP. Best numbers are in bold.

Model	CMU-MOSI	CMU-MOSEI	IEMOCAP
Mult	1.54M	1.55M	1.53M
STT	**0.44M**	**0.44M**	**0.38M**

Table 4: The number of parameters in each model for the different tasks. Hyper-parameters with best performance are selected. "M" means one million.

	Acc-2	F-score	MAE	Acc-7	Corr
ratio = 0	81.1	81.0	0.871	37.4	0.718
ratio = 0.1	81.3	81.6	0.889	38.6	0.707
ratio = 0.2	82.1	**82.2**	0.853	39.2	0.713
ratio = 0.3	**82.4**	**82.2**	**0.847**	38.9	**0.733**
ratio = 0.5	81.9	81.9	0.864	37.0	0.722
ratio = 1.0	81.3	81.2	0.869	**39.5**	0.714
ratio = 2.0	81.4	81.4	0.871	38.0	0.718
no stepped ratio	79.9	79.8	0.912	37.4	0.697

Table 5: Ablation study and effect of different ratios on CMU-MOSI. Ratio values vary in different levels as defined in (2). "ratio = 0" means the original layer normalization is applied. "no stepped ratio" means that the scale and bias terms are applied equally to all layers.

the performance. Using $\text{ratio} = 0$, i.e., using only text features, worsens the performance. This is intuitive since additional modalities capture information not present in text. On the other hand, a large ratio like 1 or 2 degrades the performance of STT. This is also consistent with our intuition, since large ratios correspond to an almost complete dependency on the audio and video features (which are not as deeply studied as the representation of text). Thus an intermediate value for ratio should work best, and our experiments suggest a value of about 0.3. This supports our hypothesis that one should utilize the text embedding to capture the major semantics and utilize the audio and video embeddings to capture additional crucial stylistic information injected into text in a more gentle manner.

Finally, we see that using the stepped ratio is better than setting the same stepped ratio at each layer. This is consistent with our intuition that lower layers of the transformer model corresponds to basic semantics that should remain unaffected by external information.

5.1.2 Study 2: Unimodal Input Features

To study the effects of audio or video only input to the transformer model, we set up experiments in which the STT takes as input each of audio or video modes alone. For each modality, we vary the stepped ratio from 0.1 to -0.5 and report the best numbers. Table 6 presents results from STT using unimodal text or audio input when evaluated against the CMU-MOSI data set. From Table 6, while it is hard to determine if individually audio or video makes for a better input on the sentiment classification task, it is evident that the combined bimodal (audio+video) input does better than each unimodal input.

5.1.3 Study 3: Effect of Non-verbal Features

The results in Table 6 quantitatively show the effectiveness of non-verbal features, however, it's also vital to demonstrate that the non-verbal features can help to successfully classify specific examples' sentiments. Figures 3 and 4 present two case-studies that show how the STT model is able to capture non-verbal information effectively by comparing

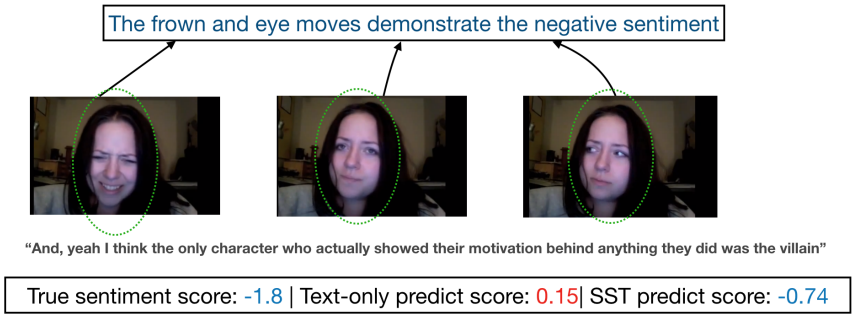

Figure 3: Case Study Example 1. The text in this utterance doesn't have a clear sentiment, however, the woman's frown and eye movements reveal her disappointment in the characters' motivations.

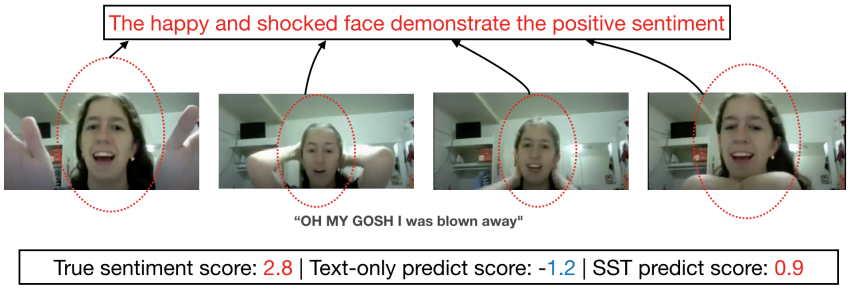

Figure 4: Case Study Example 2. The text "blown away" alone may be ambiguous. However, combined with the woman's happy facial expressions, the overall support is towards a strong positive sentiment.

	Acc-2	F-score	MAE	Acc-7	Corr
STT without audio/video	81.1	81.0	0.871	37.4	0.718
STT with audio	81.1	81.4	0.887	38.0	0.711
STT with video	81.7	81.5	0.888	37.9	0.720
STT with audio and video	**82.4**	**82.2**	**0.847**	**38.9**	**0.733**

Table 6: This table presents the effect of unimodal audio/video inputs to STT when evaluated on the CMU-MOSI data set.

the prediction of our STT model with that of a text-only model (stepped ratio $= 0$). We select two typical examples where our STT model is able to predict the correct sentiment score while the text only model is not. Figure 3 shows an example in which the text is neutral but the facial expression clearly demonstrates a negative sentiment. The STT model is able to predict the negative sentiment by using the facial features. Figure 4 shows another example with ambiguous text "blown away." Again, the text-only model does not predict correctly, but our STT is able to exploit the information in the visual features to predict the true sentiment.

5.2 Applications in Pre-trained Language Models

Our proposed method replaces the vanilla layer normalization in the transformer's layers with a stepped style adaptive layer normalization (SAdLaN). The same technique can also be applied to expand the capabilities of a pre-trained text-only transformer model. To validate this, we consider a pre-trained language model BERT and apply SAdLaN to every layer in BERT. This allows applying it on multimodal data sets, by fine-tuning the SAdLaN parameters or fine-tuning both the SAdLaN and BERT parameters. To clarify, differences in results reported in the previous sections and here lie in the transformer architecture. Since the BERT language model is pre-trained in prior work, the number of transformer layers differ in BERT and STT. The BERT language model in addition to using the transformer architecture also follows a *cloze* task in the language model. Unlike BERT, STT does not perform random masking on word tokens in the input.

It is also possible to combine styled layer normalization with the adapter method (Houlsby et al., 2019). The adapter method injects some trainable modules called adapters between the feed-forward layer and the layer-normalization in each transformer encoder layer of BERT. Each adapter contains two feed-forward projection layers, connected by a non-linear activation. The first feed-forward layer projects its input (of dimension d) to a smaller dimension, then the second layer maps the output

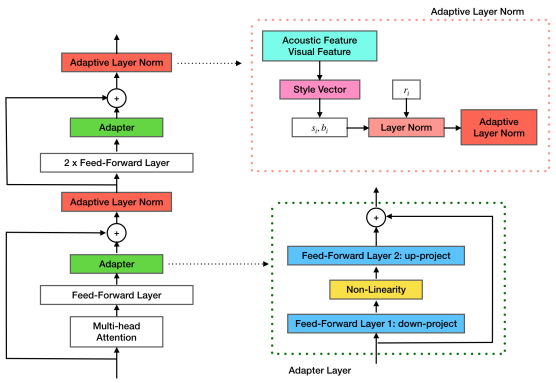

Figure 5: This figure illustrates injecting the adapter and the adaptive layer-normalization into BERT's transformer encoders. At each encoder layer of the BERT model, the adapter network is injected between the feed-forward layer and the adaptive layer normalization. During fine tuning, any part of the model can be frozen for ablation studies.

of the first layer back to dimension d. Houlsby et al. (2019) showed that injecting adapters and fine-tuning only the parameters of the adapters achieve competitive results against fine-tuning the whole BERT. We can combine SAdLaN with adapters to achieve similar results that again use fewer trainable parameters. Figure 5 illustrates the combination.

We now present experiments on these applications of SAdLaN (combined with BERT, or with BERT+adapters). The BERT model used in our experiments is the "bert-base-uncased" version in (Devlin et al., 2018). Evaluation uses the CMU-MOSI and CMU-MOSEI datasets. Raw text is used as the input textual feature instead of the pretrained Glove embeddings used in the other experiments, while acoustic and visual features remain the same.

Results: Tables 7 and 8 present the performance of different methods fine-tuning BERT on multimodal data.

- The BERT model performs roughly the same with or without SAdLaN (see the first two rows in both tables). This is likely due to the large model parameters in BERT (110M) as opposed to the much fewer parameters in SAdLaN ($\leq$ 1M).

- Updating only the parameters of SAdLaN and the final logistic regression layer demonstrates an obvious improvement compared to using the final regression layer only (see the third to fifth rows). This confirms that our method is able to inject acoustic and visual information into the transformer model.

- Combining SAdLaN with the adapter method performs better than using the adapter method alone (see the last two rows). In this case, the performance is similar to fine-tuning the whole BERT (in the first row), while updating far fewer parameters (about only 1.5%). Thus our method improves the adapter method and enables efficient training on multimodal tasks.

	Acc-2	F-score	MAE	Corr	Size
BERT	**86.6**	**86.0**	**0.683**	**0.80**	110M
BERT + SAdLaN	85.1	85.0	0.689	**0.80**	111M
SAdLaN	**83.0**	**83.3**	**0.794**	**0.76**	1.5M
(BERT output)	80.5	81.1	0.89	0.69	0.06M
(a + v + BERT output)	81.0	81.0	0.885	0.69	0.06M
Adapter	84.1	84.3	0.72	0.77	3.1M
Adapter + SAdLaN	**85.1**	**85.2**	**0.696**	**0.79**	3.5M

Table 7: Results of fine-tuning BERT, adapter, and our method SAdLaN on CMU-MOSI. Logistic regression is used as the final classifier. The training only updates the parameters in the first column (and those in the logistic regression). (Bert output) means a simple average of all the layers hidden vector, (a + v + Bert output) means a direct concatenation of acoustic, visual, and BERT hidden vectors. Best results in each block are in bold.

	Acc-2	F-score	MAE	Corr	Size
BERT	**85.9**	**85.9**	0.533	0.76	110M
BERT + SAdLaN	85.5	85.7	**0.527**	**0.77**	112M
SAdLaN	**84.8**	**84.9**	**0.565**	**0.73**	1.7M
(BERT output)	82.8	83.0	0.582	0.68	0.07M
(a + v + BERT output)	82.7	82.8	0.583	0.67	0.07M
Adapter	**85.2**	85.1	0.543	0.751	3.0M
Adapter + SAdLaN	**85.2**	**85.3**	**0.533**	**0.77**	3.6M

Table 8: Results from fine tuning BERT, adapter, and our method SAdLaN on CMU-MOSEI. Annotations are the same as in Table 7.

6 Conclusion and Future Work

Inspired by the success of style transfer algorithms in image processing, this paper proposed the novel Style Transfer Transformer (STT) in which layer-normalization in the transformer model is replaced with a style Stepped Adaptive Layer Normalization (SAdLaN). The model is used to learn comprehensive multimodal representations for sentiment analysis and emotion recognition. Experiments on benchmark data sets established the effectiveness of the proposed method. Furthermore, ablation studies provided supports for our hypothesis of injecting audio and video to highly efficient text

embeddings enhances the performance of the text embedding in multimodal tasks without the need for larger models or training data.

While our work is a first step towards learning multimodal embeddings via style transfer of non textual features onto text, as part of future work we will consider learning to inject non-verbal information into the text model in a recursive manner in order to achieve a higher model performance. The acoustic and visual features can be processed separately; besides of analyzing examples that benefit from the STT, it's also worthwhile to study examples that are negatively impacted by the method. While transformer models promise improved results, reproducibility in these models is a cause for concern since transformer models are particularly sensitive to initial conditions. As part of future work we will perform experiments to establish significance in observed results and report averaged hyper-parameters.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2018. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335.

Gilles Degottex, John Kane, Thomas Drugman, Tuomo Raitio, and Stefan Scherer. 2014. Covarep—a collaborative voice analysis repository for speech technologies. In *ICASSP*, pages 960–964. IEEE.

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*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. *arXiv preprint arXiv:1902.00751*.

Xun Huang and Serge J. Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. *CoRR*, abs/1703.06868.

iMotions. 2017. Facial expression analysis. *imotions.com*.

Tero Karras, Samuli Laine, and Timo Aila. 2018. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948.

Paul Pu Liang, Ziyin Liu, Amir Zadeh, and Louis-Philippe Morency. 2018. Multimodal language analysis with recurrent multistage fusion. *arXiv preprint arXiv:1808.03920*.

Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. 2018. Efficient low-rank multimodal fusion with modality-specific factors. *ACL*.

Sijie Mai, Haifeng Hu, and Songlong Xing. 2019. Modality to modality translation: An adversarial representation learning and graph fusion network for multimodal fusion.

Trisha Mittal, Uttaran Bhattacharya, Rohan Chandra, Aniket Bera, and Dinesh Manocha. 2019. M3er: Multiplicative multimodal emotion recognition using facial, textual, and speech cues. *arXiv preprint arXiv:1911.05659*.

Louis-Philippe Morency, Rada Mihalcea, and Payal Doshi. 2011. Towards Multimodal Sentiment Analysis: Harvesting Opinions from The Web. In *ICMI 2011*, Alicante, Spain.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Asa Cooper Stickland and Iain Murray. 2019. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. *arXiv preprint arXiv:1902.02671*.

Zhongkai Sun, Prathusha Sarma, William Sethares, and Yingyu Liang. 2019a. Learning relationships between text, audio, and video via deep canonical correlation for multimodal language analysis.

Zhongkai Sun, Prathusha K Sarma, William Sethares, and Erik P Bucy. 2019b. Multi-modal sentiment analysis using deep canonical correlation analysis. *arXiv preprint arXiv:1907.08696*.

Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Multimodal transformer for unaligned multimodal language sequences. *arXiv preprint arXiv:1906.00295*.

Yao-Hung Hubert Tsai, Paul Pu Liang, Amir Zadeh, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2018. Learning factorized multimodal representations. *arXiv preprint arXiv:1806.06176*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Liwei Wang, Yin Li, and Svetlana Lazebnik. 2016. Learning deep structure-preserving image-text embeddings. In *CVPR*, pages 5005–5013.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.

Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. 2018a. Recovering realistic texture in image super-resolution by deep spatial feature transform. *CoRR*, abs/1804.02815.

Yansen Wang, Ying Shen, Zhun Liu, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. 2018b. Words can shift: Dynamically adjusting word representations using nonverbal behaviors. *arXiv preprint arXiv:1811.09362*.

Amir Zadeh, Paul Pu Liang, Navonil Mazumder, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2018a. Memory fusion network for multi-view sequential learning. In *AAAI*.

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. 2016. Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv preprint arXiv:1606.06259*.

AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2018b. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In *ACL*, pages 2236–2246.