Disentangling Representations from Pre-Trained Language Models

A

Thesis

Presented to
the faculty of the School of Engineering and Applied Science
University of Virginia

in partial fulfillment
of the requirements for the degree

Master of Science

by

Aniruddha Dave

May 2021

APPROVAL SHEET

This

Thesis

is submitted in partial fulfillment of the requirements
for the degree of

Master of Science

Author: Aniruddha Dave

This Thesis has been read and approved by the examing committee:

Advisor: Yangfeng Ji

Advisor:

Committee Member: Yanjun Qi

Committee Member: Hongning Wang

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, School of Engineering and Applied Science

May 2021

Abstract

Pre-trained language models dominate modern natural language processing. They rely on self-supervision to learn general-purpose representations. Given the redundant information encoded in these representations, it is unclear what information encoded leads to superior performance on various tasks and whether it can be even better if it is encoded in an interpretable way. In this work, with stylistic datasets, we explore whether style and content can be disentangled from sentence representations learned by pre-trained language models. We devise a novel approach leveraging multi-task and adversarial objectives to learn disentangled representations. The latent space is divided into different parts and fine-tuned so that they encode different information. Our approach is demonstrated using parallel datasets with different styles from various domains. We show that style and content spaces can be disentangled from the sentence representations through this simple yet effective approach.

Acknowledgements

First and foremost, I would like to thank my advisor, Professor Yangfeng Ji. Without his assistance and dedicated involvement throughout my graduate program, this work would not have been possible. He has always been open to questions and taught me how to perform research. I highly appreciate his unconditional support.

I would also like to thank the committee members of my thesis, Professor Yanjun Qi and Professor Hongning Wang, for devoting their time to provide me with their valuable advice. Their feedback has helped me further refine my work.

Finally, I would like to thank my family for their continuous encouragement and support throughout my life. This accomplishment would not have been possible without them. Thank you.

Table of Contents

Chapter 1

Introduction

1.1 Background

Large pre-trained models (Devlin et al., 2018; Yang et al., 2019; Radford et al., 2018) have come to dominate modern Natural Language Processing (NLP) in a variety of downstream tasks by learning general-purpose representations. They have constantly pushed the state-of-the-art in NLP tasks such as Question Answering, Textual Entailment, Machine Translation, Natural Language Inference (NLI) (Rajpurkar et al., 2016; Wang et al., 2018; Edunov et al., 2018; Liu et al., 2019). Each new model introduces a deeper and wider architecture while learning multiple tasks without any supervision (Radford et al., 2019). This highly performing modeling capability is a consequence of multiple layers of non-linear transformations of input sequences. Such transformations make the intermediate features **latent**, that is, they do not have any explicit meaning and are not interpretable. It is not obvious what information gets encoded in the latent representations learned by these models, which leads to competitive performance in a multitude of downstream tasks. This, in turn, hinders the model's robustness and interpretability. Moreover, it is not clear whether the performance can be improved further by encoding the representations in an interpretable way. Therefore, it is essential to learn how the information is encoded in these representations and find out methods to improve this process. In this work, we try to address the issue of obscurity in the sentence representations from pre-trained language models. We propose to push these models on learning disentangled representations.

1.2 Disentangled Representations

Disentangled representations map different aspects of data into distinct, independent, and complementary low-dimensional latent vector spaces. They have become an increasingly important research topic for making deep learning models more interpretable (Cheng et al., 2020). A sequence of operations, such as selecting, combining, and switching, can be performed on the learned disentangled representations to utilize them in downstream applications. They can be used in tasks like domain adaption (Liu et al., 2018), style-transfer (Lee et al., 2018), conditional generation (Denton and Birodkar, 2017; Burgess et al., 2018), and few-shot learning (Verma et al., 2018). Disentangled representations have been widely used in various domains, such as, images (Lee et al., 2018; Tran et al., 2017), videos (Hsieh et al., 2018; Li and Mandt, 2018) and speech (Chou et al., 2018; Zhou et al., 2019). However, disentangled representations have received limited attention in natural language processing (John et al., 2018).

Compared to images that have apparent independent factors of variation such as size, position, color, and orientation which have a physical grounding, natural language text lacks such attributes that can be formalized in terms of actions of symmetry subgroups (Higgins et al., 2018). To overcome this challenge and disentangle various text attributes, we can consider two factors: style and content (John et al., 2018). The content embedding is designed to encode the semantic information of a sentence. The style embedding can then be used to represent any desired attribute of text such as sentiment, personality, formality, etc. Disentangling style and content in a text are essential as they can benefit many downstream tasks. Disentangled representations can help generate text in a controlled manner while manipulating its various attributes. They can be used in dialog/conversational systems to induce various personalities that interact with humans. They can also be highly useful for aiding humans in writing various types of text. In this work, we explore whether style and content as two text attributes can be disentangled from sentence representations in pre-trained language models.

1.3 Thesis Overview

Section 2 gives an overview of the research carried out related to probing neural networks and disentangling latent representations of text learned by neural networks. In Section 3 we introduce our approach towards probing and disentangling sentence representations. We first probe sentence representations from pre-trained language models for style information. We then propose an approach to disentangle sentence representations by fine-tuning the pre-trained models. We design multi-task and adversarial loss functions to segregate the information learned by the distinct vector spaces. Section 4 discusses the experimental setup and we discuss the results in Section 5.

We evaluate our approach using four style-transfer datasets. We demonstrate the efficacy of our method by comparing the classification scores from the style and content spaces. We evaluate the content space representations by ranking the sentences based on the content embedding and comparing the ranks of sentences with complementary styles. This gives an idea about how close the sentences are in their meaning. We both qualitatively and quantitatively show that latent representations of text sentences can be disentangled using this simple yet effective approach.

Chapter 2

Related Works

2.1 Disentangled Representation Learning for Text

Disentangling the latent space in neural networks has been widely explored in the context of images/videos in computer vision (Chen et al., 2016; Higgins et al., 2018, 2016). These approaches learn disentangled representations in a purely unsupervised manner. They do no use any style labels. Images have clear independent factors of variation such as size, position, color, rotation etc. which have a physical grounding and can be formalized in terms of action of symmetry subgroups (Higgins et al., 2018). Unfortunately, natural language text does not have attributes that have physical grounding and can be formalized mathematically. Hence, we have not observed disentangled attributes in text without using supervision.

In NLP, the definition of ”style” itself is vague. Researchers have often treated sentiment as a style attribute. Rao and Tetreault (2018) treat ”formality” of writing as the style. They create a parallel corpus for style transfer. Jhamtani et al. (2017) create a parallel dataset of 17 Shakespeare plays and their corresponding modern English conversions. (Pryzant et al., 2020) treat ”bias” in a sentence as style and design a model to automatically remove the bias words neutralizing the sentence. They create a dataset that contain sentences from wikipedia articles that were tagged as exhibiting some kind of bias sentences and the corresponding edited neutral sentence.

We use these three definitions of style for our work, namely, old-modern writing style, formality, bias.

Shen et al. (2017) align the recurrent hidden decoder states of original and style-transferred sentences using a pair of adversarial discriminators. Fu et al. (2018) propose two methods for controlling style. In the first approach they train style-specific embeddings while in the second approach they train style-specific decoders. Zhao et al. (2018) use the multi-decoder approach along with a Wasserstein-distance penalty to align content representations of sentences with different styles. Xu et al. (2018) and Logeswaran et al. (2018) use the cyclic consistency of back translation to ensure content preservation. These methods employ reinforcement learning and are usually difficult to train.

Some work on disentangling representations in text has focused on generating factored representations for controlled generation tasks. Larsson et al. (2017) use a CNN to generate sentence representations and traverse the latent space for manipulating the sentiment. Some of the recent work has focused on learning distinct representations for syntax and semantics (Chen et al., 2019; Ravfogel et al., 2020; Zhang et al., 2021).

John et al. (2018) use adversarial and multi-task objectives to learn separate style and content vectors. They use a variational autoencoder model to reconstruct the sentence. They use a style classifier to evaluate the encoded style information while they use bag-of-words features to evaluate the content information.

Inspired by this idea, we create adversarial and multi-task objectives for sentence representations from pre-trained language models. We use a style classifier as an adversary so that the style information is captured only in the style space. We use some similarity measurements to impose certain constraints on the latent space.

Chapter 3

Disentangling Representations

Figure 3.1 shows the overview of our approach. We use an encoding transformer as our base model. We assume that the style and content information may either be encoded in separate dimensions or they could be a linear combination of these individual dimensions. Hence, we attach a fully-connected linear layer to the transformer model to take that into consideration. Then we design auxiliary losses for style and content embeddings for disentanglement.

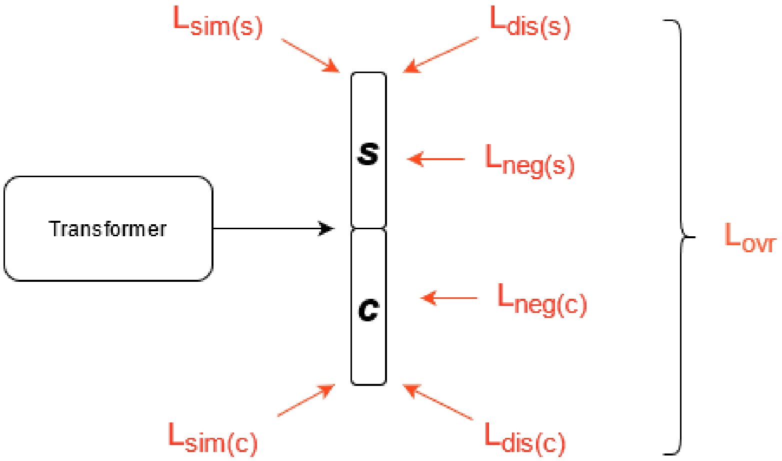

Figure 3.1: Overview of our Approach

We divide the sentence representation from a transformer model into two parts, first, the style space $\mathcal{V}_s$ and second, as the content space $\mathcal{V}_c$. For any example x, its latent representation $\vec{z}$ can be decomposed as:

$$\vec{z} = \vec{z}_s \oplus \vec{z}_c \tag{3.1}$$

with $\vec{z}_s \in \mathcal{V}_s$ and $\vec{z}_c \in \mathcal{V}_c$.

We choose to use the same style classifier on both the style and content embeddings. The style classifier will be used to make the style embedding encode all the style information while it will be used as an adversary on the content embedding space so that no style information gets encoded in it. If we use a different style classifier for both embedding spaces then the style classifier for content embedding may just learn to be a bad classifier rather than de-learning the style information. Thus, in order to share weights we maintain the size of the both style and content spaces to be equal. We choose the first half of the latent representations to be the style space and the second half to be the content space.

The transformer model is fine-tuned with a joint loss function for disentangling information between $\mathcal{V}_s$ and $\mathcal{V}_c$. The overall loss function consists of two discriminators and two similarity measurements as described below.

3.1 Style Discrimination

We want the style space to encode all the stylistic information from a text, while the content space should not encode any style-related information. We train a single discriminator D to predict style from the style and content embeddings. The style discriminator D is a binary classifier defined on the style $\mathcal{V}_s$ and content space $\mathcal{V}_c$. For any sample x and its style label y, the discriminator of style representation $\vec{z}$ is defined as below,

$$p(y_s) = \sigma(\vec{w}\vec{z}_s + b) \tag{3.2}$$

which represents the style prediction probability given $\vec{z}_s$.

The corresponding loss function on which the discriminator is trained is,

$$J_{dis(s)} = \mathrm{CE}(y_s, y) \tag{3.3}$$

$\vec{w}$ and b are parameters of the style discriminator and CE(.) represents the cross-entropy loss. Similarly, we can define the style prediction probability for the content embedding $\vec{z_c}$ as,

$$p(y_c) = \sigma(\vec{w}\vec{z_c} + b) \tag{3.4}$$

where $\vec{w}$ and b are same parameters of the discriminator and the corresponding loss function to train the discriminator is,

$$J_{dis(c)} = \mathrm{CE}(y_c, y) \tag{3.5}$$

In the training process the discriminator is trained using Equations 3.3 and 3.5.

Intuitively, we would like the prediction of Equation 3.2 as accurate as possible while the prediction of Equation 3.4 to be as inaccurate as possible. Therefore, the loss function for fine-tuning the transformer model using the discriminator is defined as,

$$L_{dis} = \mathrm{CE}(y_s, y) - \mathrm{CE}(y_c, y) \tag{3.6}$$

where CE(.) represents the cross-entropy loss and y is the ground truth label for the sentence.

3.2 Similarity Measurements

In addition to the discriminators, we would also like to impose two (dis)similarity constraints into the content space and the style space. For an aligned pair (x, x') with x and x' have similar content but different styles, the (dis)similarity constraints imposed in $\mathcal{V}_s$ and $\mathcal{V}_c$ will make sure $\vec{z_c}$ and $\vec{z'_c}$ are close to each other in $\mathcal{V}_c$, while $\vec{z_s}$ and $\vec{z'_s}$ are separated apart in $\mathcal{V}_s$. To be specific, we measure the similarity between two vectors with ℓ_2 norm and define the corresponding loss function as,

$$L_{sim} = \|\vec{z}_c - \vec{z'_c}\|_2^2 - \|\vec{z}_s - \vec{z'_s}\|_2^2 \tag{3.7}$$

This loss function tries to minimize the distance between the content embeddings of two sentences with same meaning while maximizing the distance between the style embeddings of the pair with different styles.

3.3 Negative Sampling

If we use only Equation 3.7 to fine-tune the transformer model then that may result in the model to learn the trivial vector $(\vec{0})$ for the content representation. This would satisfy our similarity requirements but make the model loose all the content information. In order to avoid this situation we introduce negative sampling (Mikolov et al., 2013) in our approach. We select a few samples from the training data and add introduce constraints to avoid problems where the model may completely disregard the encoded information. The negative sampling constraints are added on both the content and style embeddings separately which are as described below.

Content Embeddings For each sentence x, we select k sentences randomly from training data (except x' which is same in content but only differs in style). These k sentences serve as negative samples since they differ completely in meaning with the given sentence.

Let's say we have m training examples with $\vec{z^i_c}$ denoting the content embedding learned by the model for the $i'th$ sentence. Then we define the negative sampling loss as,

$$L_{neg(c)} = -\frac{1}{k}\sum_{j=1}^{j=k} \|\vec{z^i_c} - \vec{z^j_c}\|_2^2 \tag{3.8}$$

This loss function tries to maximize the distance of content embedding between two sentences that have different content. This constraint helps us in learning distinct content vectors for different sentences while the similarity constraint helps in learning the same vector for sentences with the same content. Thus, we can avoid the problem of learning trivial vectors.

Style Embeddings We use a similar negative sampling approach to learn better representations for the style embedding. We minimize the distance between two style embeddings that belong to the same style while maximizing the distance between two style embeddings that belong to different styles.

Let's say we have m training examples with $\vec{z_s^i}$ denoting the style embedding learned by the model for the $i'th$ sentence. For each sentence x, we select k sentences randomly from training data. Then we define the negative sampling loss as,

$$L_{neg(s)} = \frac{1}{k} \sum_{j=1}^{j=k} NS(\vec{z_s^i}, \vec{z_s^j}) \tag{3.9}$$

where NS is defined as,

$$NS(\vec{z_s^i}, \vec{z_s^j}) = \begin{cases} \|\vec{z_s^i} - \vec{z_c^j}\|_2^2, & \text{if } i \text{ and } j \text{ belong to the same style,} \\ -\|\vec{z_s^i} - \vec{z_c^j}\|_2^2, & \text{if } i \text{ and } j \text{ belong to different styles,} \end{cases} \tag{3.10}$$

This loss function ensures that we learn the same style embeddings for sentences with same style irrespective of the content. It also tries to learn different style embeddings for sentences with different style.

3.4 Fine-Tuning Algorithm

The overall loss function for fine-tuning a transformer model is as shown below:

$$L_1 = L_{dis}$$

$$L_2 = L_{sim} + L_{neg(c)} + L_{neg(s)}$$

$$L_{ovr} = L_1 + L_2 \tag{3.11}$$

Here,

L_{dis} makes the model encode the style information only in the style embeddings,

L_{sim} makes the content embedding similar for sentences with same semantics and contrasting style while making their style embeddings dissimilar,

$L_{neg(c)}$ makes the model learn unique content vectors for sentences with different meanings,

and,

$L_{neg(s)}$ makes the model focus on the style embedding irrespective of the content, allowing the model to learn the same style embedding for a particular style label.

Algorithm 1: Fine-tuning Algorithm

```
for ( each epoch ) {
    if epoch < l then
        /* Pre-train the discriminator */
        minimize $J_{dis(s)}$
        minimize $J_{dis(c)}$
    else
        if epoch is odd then
            /* Train the discriminator */
            minimize $J_{dis(s)}$
            minimize $J_{dis(c)}$
        else
            /* Fine-tune the model */
            for ( each mini-batch ) {
                if if iteration number is odd then
                    minimize $L_{dis}$
                else
                    minimize $L_{sim} + L_{neg(c)} + L_{neg(s)}$
                end
            }
        end
    end
}
```

Algorithm 1 describes our algorithm for fine-tuning the model to learn disentangled representations.

Since the pre-trained model already learns some information about the style and content we pre-train the style classifier for l epochs[1] so that it can intelligibly discriminate the learned vectors. Then we alternate between training the classifier and training the transformer model to minimize our objective functions. Since the value of L_{dis} is on a different scale it might result into uneven optimization over the different loss functions. Hence, we alternate the training between L_1 and L_2 in each mini-batch. This also helps us eliminate the need of hyper-parameters to weight the different loss functions in the overall loss function.

The details about how we setup the experiments and training are explained in the next section.

[1] we use l = 3

Chapter 4

Experimental Setup

4.1 Transformer Models and Sentence Representation

In this work we consider two transformer models, namely, **BERT** (Devlin et al., 2018) and **XLNet** (Yang et al., 2019) transformer models. BERT is a type of autoencoding model (AE) while XLNet is a type of autoregressive model.

We use the last hidden state of these models as the sentence representation. The last hidden state for a single sentence is a two-dimensional vector of the number of hidden states and the sequence length. We take the average of the hidden state over the sequence length dimension to get a single vector of length equal to the hidden state size of the model and treat this as the sentence representation.

4.2 Datasets

We use aligned stylistic datasets that comprise of pairs of sentences with complementary styles and similar content. This helps us evaluate both style classification on the style space and content preservation on the content space.

The four datasets we use for the experiments are as follows:

- **Shakespeare**: This dataset is comprised of modern English translations of 17 Shakespeare plays (Jhamtani et al., 2017)

Dataset	Train	Dev	Test
Shakespeare	18395	1218	1462
Wiki-Neutrality Corpus	53803	700	1000
GYAFC-EM	52595	2877	1416
GYAFC-FR	51967	2788	1332

Table 4.1: Dataset Overview

- **WNC** (Wiki-Neutrality Corpus): This dataset contains sentences from wikipedia articcles that were tagged as exhibiting some kind of bias sentences and the corresponding edited neutral sentence (Pryzant et al., 2020),

- **GYAFC-EM** and **GYAFC-FR**: This is a part of the Grammarly's Yahoo Answers Formality Corpus (Rao and Tetreault, 2018). They contain informal sentences from different domains on Yahoo Answers and their corresponding formal sentences written by human annotators. The two domains that we use are the Entertainment & Music (EM) and Family & Relationships (FR).

We use the standard training, development and testing splits of each of the datasets which are described in Table 4.1.

4.3 Disentangling Representations

Algorithm 1 describes our algorithm to learn disentangled representations using transformer models.

The style classifier is first pre-trained on the training data until the it's classification accuracy is better than a random model [1]. The style classifier is a single-layer neural network which is trained on the Binary Cross Entropy loss function as described in Equations 3.3 and 3.5. Next, the classifier is then alternately trained with the transformer model.

We fine-tune the transformer model using the loss function described in Equation 3.11. We alternate between L_1 and L_2 for each mini-batch during the training process.

[1] Experimentally we find that the training for first 3 epochs is sufficient

We use Adam (Kingma and Ba, 2014) for optimizing the loss function. We terminate the model training after 50 epochs. Since, we are fine-tuning the model we use low values of the learning rate. We use different values of learning rate for both L_1 and L_2 losses that are defined in Equation 3.11. The learning rate for L_1 loss is 10^{-8} while for the L_2 loss is 10^{-5}.

4.4 Evaluation Metrics

To evaluate the degree by which the style information is present in the embeddings we use style classification accuracy for the two embeddings.

To visualize how the latent space changes while fine-tuning we use TSNE (Van der Maaten and Hinton, 2008) to project both the style and content spaces into 2-dimensional plots as shown in Figure 5.1.

We also perform a quantitative evaluation of the content space by rank estimation. We randomly sample 100 sentences from the test set. For each sentence, we estimate the rank of the complementary style sentence, R', among all samples. Next, we estimate the mean reciprocal rank (MRR) of all samples as shown below,

$$MRR = \frac{1}{n}\sum_{i}^{n}\frac{1}{R'_i} \tag{4.1}$$

where, n=100. This metric measures how closely sentences with similar meaning but different styles are ranked. We also another metric for evaluating content that estimates how many sentences have the corresponding R' (rank of the complementary style sentence) within a 10% rank.

Chapter 5

Results and Discussion

5.1 Style Classification

Firstly, we evaluate the extent to which the style information is encoded in the style and content embeddings. Table 5.1 shows the style classification results. We first estimate the style classification accuracy from the pre-trained model using the entire sentence representation. Next, we disentangle the latent space using the approach described in Section 3. Then, we estimate the style classification accuracy on the individual style and content spaces. As we can see from the results in Table 5.1, the style classification accuracy of the style space in the disentangled model is better than the pre-trained model. This indicates that all the style embedding has learned more style information. On the other hand, the style classification accuracy for the content

	Shakespeare	GYAFC-EM	GYAFC-FR	WNC
	Pre-Trained Model			
Entire Representation	79.83	81.25	80.40	60.14
	Disentangled Model			
Style Space	84.28	85.67	86.01	65.3
Content Space	48.63	48.21	47.11	51.4

Table 5.1: Performance of style and content latent spaces on the pre-trained model and the disentangled model

embedding falls to approximately 50%. Thus, we can say that the content embedding loses all the style information.

5.2 Clustering Quality

The two latent spaces are evaluated based on their clustering quality. Ideally the style space should exhibit two distinct clusters that belong to separate styles while the content space shouldn't exhibit any clusters.

We project the style and content embeddings on a 2-dimensional plot using the t-distributed stochastic neighbor embedding method (t-SNE) (Van der Maaten and Hinton, 2008). A single batch of the test data is randomly selected and it's stlye and content embeddings are generated from the transformer model. Next, these embeddings are plotted using t-SNE on a 2-dimensional plot. The embeddings are generated and plotted after every 10 epochs of training. This gives us a qualitative measure of how the embeddings for different styles are separated in the style and content spaces.

As we can see in the Figure 5.1, the style embeddings show some separation of the sentences belonging to different style as the training progresses. However, the content embedding for sentences of different styles are a mixed group of cluster. This validates our hypothesis, that the sentences of different styles need to be far away in the style space, however pairs of sentences with different styles but same meaning should to be together in content space.

Apart from this qualitative measure of clustering, we also estime the V-measure score (Rosenberg and Hirschberg, 2007) for both style and content spaces while fine-tuning. A higher score means better and more separate clusters while a lower score indicates no clustering. As we can see in Table 5.2, the V-measure scores for the style space increase with training. The V-measure score for the content space is not shown in the table as it was always 0 during fine-tuning indicating no clustering.

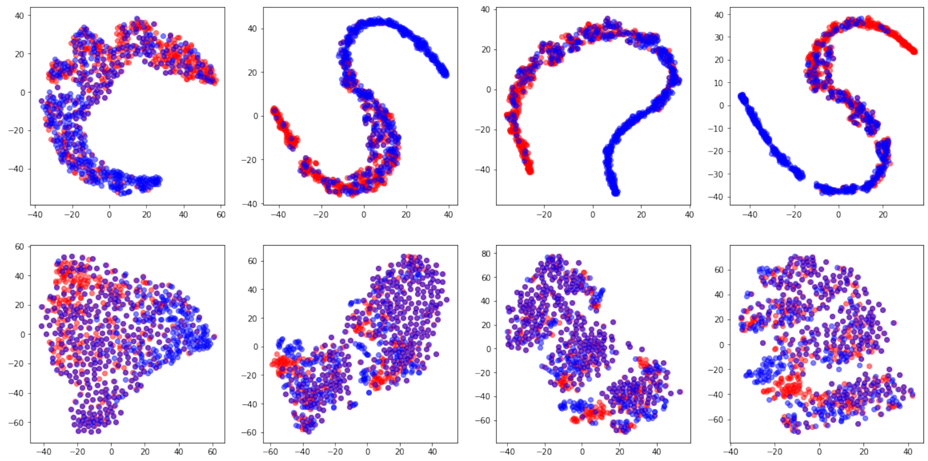

Figure 5.1: TSNE visualization of Style (above) and Content (below) latent spaces at 10, 20, 30, 40 epochs of fine-tuning
Red: Shakespearean English, Blue: Modern English
Sentences with different styles get separated in style space while they remain close in content space

Epoch	Shakespeare	GYAFC-EM	GYAFC-FR	WNC
	Style Space			
0	0.03	5.4×10^{-4}	0.04	0
10	0.05	0.05	0.08	4.4×10^{-5}
20	0.24	0.19	0.16	4.5×10^{-5}
30	0.38	0.24	0.23	5×10^{-5}
40	0.39	0.27	0.25	2×10^{-4}
50	0.41	0.27	0.27	2.7×10^{-4}

Table 5.2: V-measure score for style during fine-tuning (Higher score better clustering)

5.3 Content Evaluation

The content of a sentence is usually hard to evaluate without human supervision. We use two strategies to evaluate the content embeddings. Rank estimation is used to quantitatively evaluate the content space. 100 sentences are randomly sampled from the test set. For each sentence, the rank of complementary style sentence, R', is estimated. Next, the mean reciprocal rank(MRR) of the batch of all samples is estimated using Equation 4.1. This metric measures how closely sentences with similar meaning but different styles are ranked. Table 5.3 shows the MRR of the content space versus the style space. As we can see from the results, content space

	Shakespeare	GYAFC-EM	GYAFC-FR	WNC
Content Space	0.34	0.42	0.41	0.5
Style Space	0.28	0.25	0.23	0.49
Content Space	76.4	91.4	89.2	100
Style Space	64.8	61.2	58.2	99.6

Table 5.3: Mean Reciprocal Rank (Above); Percentage sentences in test set for which R' is within 10% (Below);

performs better at ranking sentences closer in meaning than the style space. This indicates that the content space has learn some content-based information which is not present in the style space.

The results for WNC dataset indicate almost the same content information in both the style and content spaces. We believe that this may be due to the way the dataset has been created. The WNC dataset was created using biased sentences from Wikipedia articles and their corresponding neutral edits. These edits are very small in nature and the resulting neutral sentence is not significantly different than the biased one. This causes the model to learn the same representation for both the biased and neutral sentences.

Chapter 6

Conclusion and Future Work

In this thesis, we explain the significance of disentagled representations, it's challenges and design a novel approach to solve this problem. We use multi-task and adversarial objectives along with negative sampling to learn disentangled representations from pre-trained models. The results indicate that with some fine-tuning it is possible to separate out the style and content information into different vectors. Both qualitative and quantitative metrics are used to evaluate the efficacy of the proposed approach.

This research is still in early phases. Despite demonstrating how disentangled learning can be achieved using pre-trained transformer models, there are many limitations in our proposed framework. Firstly, we assume the style and content latent spaces to be of the same size. Since style is a less complex attribute that content it should use fewer dimensions. We assume the same size as we want the classifier to share the weights of both the latent spaces so that we don't learn a bad classifier during the adversarial training. Secondly, we assume that the text has only two attributes that are independent. Natural language text contains multiple attributes which may or may not be independent. Lastly, we assume that all sentences in the training data differ in meaning and try to maximize the distance between them in the content space. An approach to rank sentences based on the semantics and use that as supervision while fine-tuning might be helpful to eliminate this problem.

The emphasis of this thesis was to propose and demonstrate that representations from pre-trained language models can disentangled, in turn, making these models more interpretable. However, a lot more experimental research needs to be carried out in order to make the pre-trained language models interpretable. We hope that this research might be a strong starting point towards understanding how pre-trained langauge models learn.

References

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. (2018). Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*.

Chen, M., Tang, Q., Wiseman, S., and Gimpel, K. (2019). A multi-task approach for disentangling syntax and semantics in sentence representations. *arXiv preprint arXiv:1904.01173*.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*.

Cheng, P., Min, M. R., Shen, D., Malon, C., Zhang, Y., Li, Y., and Carin, L. (2020). Improving disentangled text representation learning with information-theoretic guidance. *arXiv preprint arXiv:2006.00693*.

Chou, J.-c., Yeh, C.-c., Lee, H.-y., and Lee, L.-s. (2018). Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations. *arXiv preprint arXiv:1804.02812*.

Denton, E. and Birodkar, V. (2017). Unsupervised learning of disentangled representations from video. *arXiv preprint arXiv:1705.10915*.

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

Edunov, S., Ott, M., Auli, M., and Grangier, D. (2018). Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Fu, Z., Tan, X., Peng, N., Zhao, D., and Yan, R. (2018). Style transfer in text: Exploration and evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., and Lerchner, A. (2018). Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2016). beta-vae: Learning basic visual concepts with a constrained variational framework.

Hsieh, J.-T., Liu, B., Huang, D.-A., Fei-Fei, L., and Niebles, J. C. (2018). Learning to decompose and disentangle representations for video prediction. *arXiv preprint arXiv:1806.04166*.

Jhamtani, H., Gangal, V., Hovy, E., and Nyberg, E. (2017). Shakespearizing modern language using copy-enriched sequence-to-sequence models. *arXiv preprint arXiv:1707.01161*.

John, V., Mou, L., Bahuleyan, H., and Vechtomova, O. (2018). Disentangled representation learning for non-parallel text style transfer. *arXiv preprint arXiv:1808.04339*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Larsson, M., Nilsson, A., and Kågebäck, M. (2017). Disentangled representations for manipulation of sentiment in text. *arXiv preprint arXiv:1712.10066*.

Lee, H.-Y., Tseng, H.-Y., Huang, J.-B., Singh, M., and Yang, M.-H. (2018). Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 35–51.

Li, Y. and Mandt, S. (2018). Disentangled sequential autoencoder. *arXiv preprint arXiv:1803.02991*.

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

Liu, Y.-C., Yeh, Y.-Y., Fu, T.-C., Wang, S.-D., Chiu, W.-C., and Wang, Y.-C. F. (2018). Detach and adapt: Learning cross-domain disentangled deep representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8867–8876.

Logeswaran, L., Lee, H., and Bengio, S. (2018). Content preserving text generation with attribute controls. *arXiv preprint arXiv:1811.01135*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*.

Pryzant, R., Martinez, R. D., Dass, N., Kurohashi, S., Jurafsky, D., and Yang, D. (2020). Automatically neutralizing subjective bias in text. In *Proceedings of the aaai conference on artificial intelligence*, volume 34, pages 480–489.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Rao, S. and Tetreault, J. (2018). Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. *arXiv preprint arXiv:1803.06535*.

Ravfogel, S., Elazar, Y., Goldberger, J., and Goldberg, Y. (2020). Unsupervised distillation of syntactic information from contextualized word representations. *arXiv preprint arXiv:2010.05265*.

Rosenberg, A. and Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420.

Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. (2017). Style transfer from non-parallel text by cross-alignment. *arXiv preprint arXiv:1705.09655*.

Tran, L., Yin, X., and Liu, X. (2017). Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1415–1424.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Verma, V. K., Arora, G., Mishra, A., and Rai, P. (2018). Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4281–4289.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

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

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zhang, X., van de Meent, J.-W., and Wallace, B. C. (2021). Disentangling representations of text by masking transformers.

Zhao, J., Kim, Y., Zhang, K., Rush, A., and LeCun, Y. (2018). Adversarially regularized autoencoders. In *International conference on machine learning*, pages 5902–5911. PMLR.

Zhou, H., Liu, Y., Liu, Z., Luo, P., and Wang, X. (2019). Talking face generation by adversarially disentangled audio-visual representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9299–9306.