# Emotion-Guided Text Generation with VAE

**Kushal Tirumala, Christine Yu, Sarina Liu, Kevin Yu**
California Institute of Technology

## Abstract

VAE architectures have been recently very successful at realistic text generation. In the standard VAE architecture, the prior for the latent variable is taken from a normal $N(0, 1)$ distribution. In this paper, an emotion guided GMM is introduced as the prior for a structured latent variable $\mathbf{c}$. The effects of introducing different GMM priors are analyzed, as well as the control they give over the final text generation.

## 1 Introduction

### 1.1 Background

Text analysis is a large part of current machine learning and natural language processing (NLP) applications. These applications range from text summarization to sentiment analysis to large scale inference. As complex machine learning models become more tractable, increasingly complex learning tasks such as text generation and neural machine translation have gained a lot of attention. Specifically, probabilistic models have proven to be a promising attempt at capturing the complexity of tasks such as text generation. There have been many approaches to this particularly challenging task, including recurrent neural networks (RNNs) and generative adversarial networks (GANs). However, one particular model that has been relatively successful is the variational autoencoder (VAE).

The VAE consists of a generative network, which generates data samples by decoding latent factors, typically denoted $\mathbf{z}$, that are drawn from a prior distribution. The generative network is then combined with an inference network, which encodes training data samples to create the distribution in the latent space.

One reason why the VAE has been successful in text generation applications is that it theoretically has the ability to assign meaning to different variables in the latent space. However, in practice, this is very difficult, as there is a lot of semantic information encoded in the sentence, such as topic, emotion, sentiment, tone, etc. However, in the traditional VAE, the prior distribution is a standard Gaussian, which does not have the complexity to capture all this information. This project attempts to combine a more sophisticated prior in an attempt to capture emotion in the latent space and further use this to generate emotion-guided text.

### 1.2 Previous Work

Some prior work has been done in attempt to generate emotionally-guided text. Hu et al. (2017) modify the latent variable $\mathbf{z}$ to the latent vector $(\mathbf{z}, \mathbf{c})$, where $\mathbf{z}$ represents the typical unstructured latent variable drawn from a Gaussian distribution, but $\mathbf{c}$ represents a structured variable they try to assign meaning to. The sentence generator is then conditioned on the vector $(\mathbf{z}, \mathbf{c})$, and a discriminator is used on $\mathbf{c}$ to quantitatively measure how closely the generated sentences match the emotion it was meant to represent [1].

One issue is that because the input sentences are discrete and thus non-differentiable, gradient propagation no longer works exactly between the discriminator and the generator. Instead, an approximation based on a softmax with continuously decreasing temperature is used. This method modifies the discrete values to be continuous and is generally quite accurate.

Finally, the structured variable $\mathbf{c}$ can be drawn to contain both discrete and continuous variables that encode various sentiment attributes, such as sentiment, topic, and in our case, emotion [1]. The prior from which it is drawn can also vary depending on the goal of the generation of text. However, in a simple application, it is drawn from a binomial distribution that selects from $[0.5, -0.5]$ with equal probability to represent positive and negative sentiments. A diagram illustrating the full model is below:

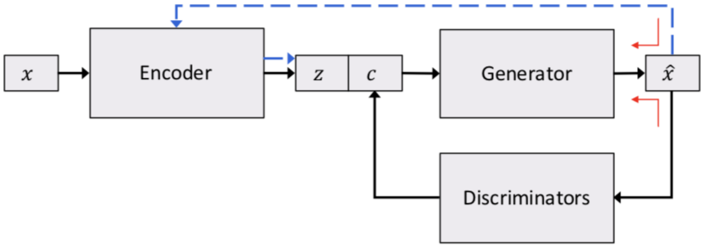


Figure 1: This figure demonstrates the model from Hu et al. (2017) where the discriminator is acted upon $\mathbf{c}$. The red arrows denote the areas where the softmax approximation is used [1].

Wang et al. (2019) propose a slightly different model for topic-guided text generation. It comprises of two parts: a neural topic model (NTM) and a neural sequence model (NSM) [5].

The NTM is intended to understand global semantic meaning across an entire document. It takes the bag-of-words representation of a document $\mathbf{d}$ where each element of $\mathbf{d}$ represents the count of a specific word in the document, and the length of $\mathbf{d}$ is the size of the entire dictionary [5]. A Gaussian random vector is passed through a softmax function to parameterize the multinomial distribution, following Miao et al. (2017) [2]. The marginal likelihood is then maximized during the training of the NTM.

The NSM builds upon a traditional VAE in which the latent variable $\mathbf{z}$ is sampled from a standard Gaussian distribution. Because of the lack of complexity in the standard Gaussian distribution, the latent variable $\mathbf{z}$ is sampled from a topic-guided Gaussian Mixture Model (GMM) with certain parameters being taken from the NTM [5]. By introducing this additional complexity, this model is more capable of guiding the text generation towards certain topics. The full model is depicted below:

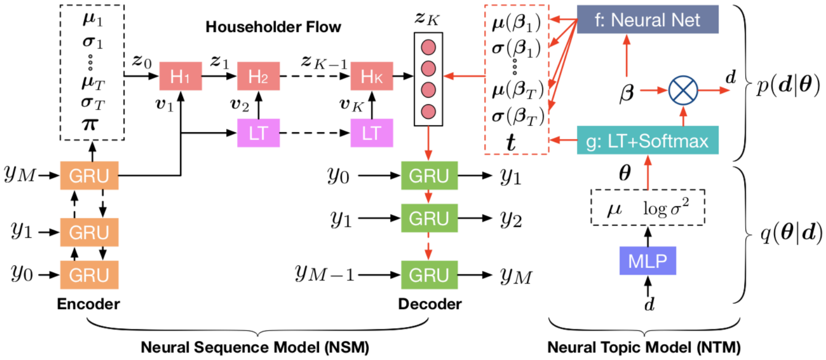


Figure 2: Above is the full model described in Wang et al. (2019). Here, the red arrows depict the generation of text, the primary interest. In the NTM, the marginal likelihood for the bag-of-words representation, followed by its outputs that are used in the GMM. This is then used to sample the latent variables used to guide the sample generation [5].

# 2 Approach

A necessary part of the standard VAE model includes a prior for the latent variable $\mathbf{z}$. This distribution is usually taken to be a normal $N(0, 1)$ distribution. By changing how the prior of this latent variable works, some control is introduced into the generative model. Most of the methods center around building a decent prior that is able to learn emotion distinction. The approach can be broken into two parts: prior specification and VAE model training.

Following Wang et al. (2019), a GMM is applied as the prior [5]. The dataset first used is the International Survey On Emotion Antecedents And Reactions (ISEAR), which consists of 7666 sentences followed by their representative emotion [3]. Then, these sentences need to be encoded with sentence vectors. The procedure for this step is varied in three ways:

1. Consider the last hidden state of a vanilla RNN: each sentence is run through the RNN and the last hidden state is taken as the encoding of the entire sentence
2. Considering word vectors: the Global Vectors (GloVE) embeddings are used for each word in the sentence (excluding stop words), and the average of the embeddings are taken to create the sentence vector.
3. Consider weighted word vectors: the GloVE embeddings for each word in the sentence (excluding stop words) are weighted with term frequency-inverse document-frequency (TF-IDF) statistic for the word.

Word vectors identify similarities across different words by representing them in a continuous vector space. The number corresponding to each word serves as the word's distributed weight across dimensions. Each dimension represents a meaning and the word's weight on that dimension measures how closely the word is associated with the meaning. The denotation is embedded across the dimensions of the vector. For a list of words from each sentence in the dataset, spaCy's parser is applied to remove punctuation. Then, a vector is extracted for each word, and all the word vectors for the sentence are stacked together. The vectors have length 96, and the corresponding columns of each of the word vectors are averaged to create a sentence vector of the same length for the whole dataset. To visualize, the two-principal components are found from the resulting sentence vectors.

TF-IDF uses weight to measure how important word is to emotion relative to a document corresponding to an emotion in a collection of documents. Term frequency counts the number of times specific word appears in document for emotion divided by total number of words in document. Inverse document frequency is the logarithm of the total number of documents or emotions used divided by the number of documents corresponding to emotions where the specific word appears, measuring the importance of each term. TF-IDF is the product of the term frequency and inverse document frequency [6]. Each sentence vector was computed by averaging word vectors scaled with their TF-IDF across each sentence.

Now with the dataset of sentence vectors, a GMM is trained with $k$ mixture components on the sentences, where $k = [2, 5, 10, 20, 30]$. The number of mixture components are varied until the GMM seemed to learn a distinction amongst the emotions.

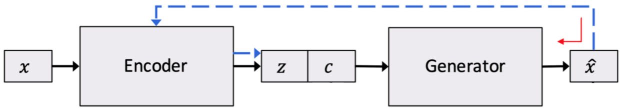


Figure 3: Diagram of model used for training. Note, the model does not include the same discriminator network as the original model.

This VAE model is very similar to that in Hu et al. (2017). The generator is an LSTM-RNN with hidden state dimension 64. Then, just as per Hu et al. (2017), there are two latent variables that the generative model is conditioned on: $\mathbf{x} \sim G(\mathbf{z}, \mathbf{c}) = P_G(\mathbf{x}|\mathbf{z}, \mathbf{c})$, where $\mathbf{z}$ represents the unstructured latent variable (as in the normal VAE architecture), and $\mathbf{c}$ is introduced as a structured latent variable [1]. Note that this paper mainly works with changing the prior of $\mathbf{c}$ as opposed to $\mathbf{z}$. This latent variable is introduced to capture structure about sentences, which in the emotion-guided sentence generation case becomes capturing emotion in sentences.

Next, the prior is changed to sample from the emotion-guided GMM in part 1. Sampling from the prior for latent variable **c** is done from the emotion-guided GMM. The corresponding mixture component for the sample is tracked, and that mixture component is one-hot encoded.

The VAE is trained as per Hu et al. (2017) with a few simplifications. These simplifications are added to directly analyze the effect of an emotion-guided prior on sentence-generation capability, especially to what extent control over the sentence generation can be introduced. The dataset used for training the VAE is the Stanford Sentiment Treebank (SST) dataset, which consists of around 215,154 sentences pulled from movie reviews. It contains 2837 sentences, all of which have length $\leq$ 15 words [4]. The length of the sentences is restricted so that the distribution of training samples are somewhat similar. Note that our model does not alter the loss from Hu et al. (2017).

One important note is that the proposed model does not have the same complexity as the original model. The original model proposed introducing an individual discriminator for each attribute code in **c** to measure how well the generated samples match the desired attributes, and drive the generator to produce improved results [1]. For the sake of directly measuring how different priors affect standard VAE text generation and minimizing complexity of the model, our proposed model focuses on a more standard VAE architecture.

To enable conditional generation of sentences, one-hot encoding of a specific mixture component is forced on the **c** prior. This is then used as a constant prior whenever a sentence is generated. The resulting generation is therefore conditioned on a mixture component.

## 3 Results and Discussion

Below is a table of the training results for different priors (these should not differ much since only the prior is changed, but this is included for completeness):

| | Loss | Recon | KL |
|---|---|---|---|
| Baseline | 2.0144 | 1.8455 | 1.1917 |
| GMM ($k = 7$) | 1.8810 | 1.6791 | 1.4245 |

Table 1: Training results for different priors after 19000 iterations

As per the loss function, the VAE is optimized to minimize the reconstruction error of observed real sentences, and at the same time regularize the encoder to be close to the prior $p(\mathbf{z})$ [1]. The actual implementation uses a KL term weight that linearly anneals from 0 to 1 during training [1]. We see that the loss for the emotion-guided GMM is lower than the baseline, specifically the reconstruction loss. This is reasonable to expect, since allowing the prior for **c** to be sampled from a learned emotion distribution allows for more realistic generation than taking the prior for **c** as $N(0, 1)$ (because in terms of encoding structure in the prior, $N(0, 1)$ intuitively would not do a good job).

The emotion-guided GMM prior is first analyzed. The original (sentence, emotion) dataset was very clustered into one domain. However, this occurrence can partly be justified since some emotions, such as anger and disgust or sadness and shame, are very similar in meaning.

But because of the overlap, the process of manually labeling a new dataset from scratch was considered. However, using this pre-prepared dataset is still the best option. From sentence to sentence, the one element that changes is emotion, which can be difficult to regulate when generating a new dataset that pulls from various sources.

Over all the different pipelines for training a GMM, the TF-IDF weighted word vector encoding of our dataset works best. A majority of the specific mixture components of a GMM are the same true label as given in the dataset. The best output is shown below in Figure 5.

Based on Figure 5, there are two main distribution types among these four emotions. Anger and disgust are able to correctly separate from shame and sadness. However, the difference between anger and disgust and the difference between shame and sadness are not very significant. Therefore, the GMM model is able to disentangle some of the data, but not all of it.

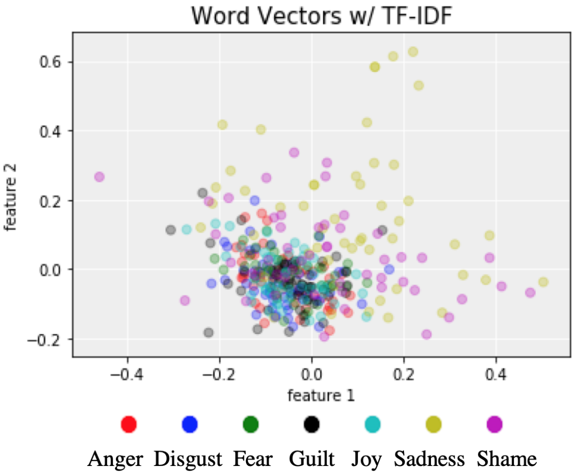


Figure 4: Each point represents one sentence in the ISEAR dataset and is colored based on the corresponding emotion.

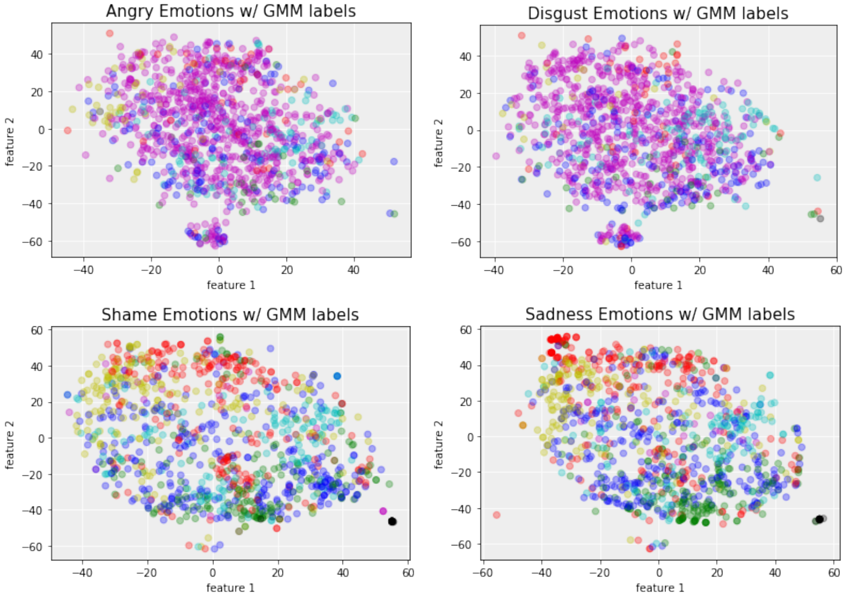


Figure 5: In these visualizations, the emotion is fixed, and the result is the distribution of the mixture components the model learned.

There is currently no quantitative way of evaluating how well the GMM learned mixture components, specifically if mixture components learned had any correlation with emotion. Instead, the performance of GMM training is evaluated qualitatively by inspecting how well the GMM learned distinction between points (i.e sentences). Given that the input dataset was all sentences distinguished by the one attribute emotion, it is then inferred that the distinction the GMM learns correlates with emotion.

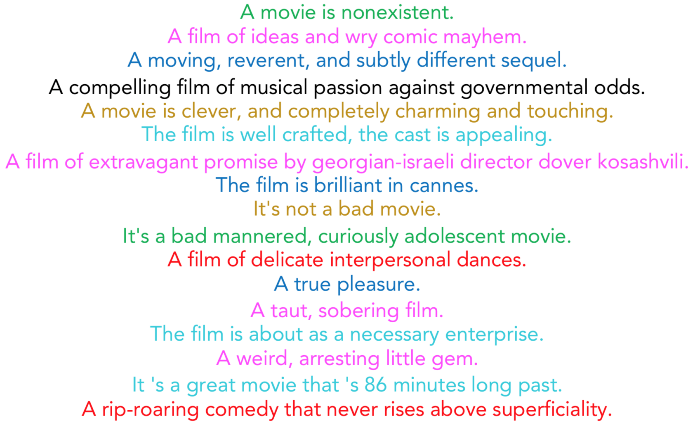
Then, in the baseline model, the prior for emotion (i.e our latent variable **c**) is set up to have a normal $N(0, 1)$ distribution, which allows the behavior of latent variable priors in a standard VAE model to be captured. Below, we see the output of a few sentences generated via the baseline model, along with the value the **c** had as a prior for each generated sentence:

```
prior c = 0.9099
Generated: a film that never rises above easy joie de vivre.

prior c = 1.6257
Generated: a film that loses sight of the year.
```

```
prior c = -0.5879
Generated: a film that never rises above easy, and utterly charming.

prior c = -0.3200
Generated: a film that never rises above easy joie de vivre.

prior c = -0.0510
Generated: the film is hampered by its courage, the film breaks your
heart.

prior c = -0.9421
Generated: the film is hampered by its courage of its convictions and
martha while huppert ...

prior c = -0.3567
Generated: the movie is n't quite unengaging.

prior c = 0.5557
Generated: the film sparkles with the wisdom and humor of the stand -
up comic.

prior c = -0.2686
Generated: a film that never rises above easy, but languorous.

prior c = 0.2950
Generated: the film is bogus and inspiring and the stories life.
```

In the baseline sentence generation, the prior for $c$ is generally close to the mean of $\mu = 0$ (which makes sense since the prior is a $N(0, 1)$ distribution). There are few notable aspects, one of which is that sentences tend to repeat when the prior is a normal distribution. To an extent, this aligns with the idea of the text generation being conditioned on $\mathbf{c}$ with $prior(c) \sim N(0, 1)$, since we then expect the conditional generation to generate very similar sentences. However, seeing as how the space of possible tokens (words) is relatively big (around $18,000$), it seems unlikely that the exact same sentences will be generated on different runs [4]. This may imply that the model gives a heavy weight to the initial values for $\mathbf{c}$ during the sentence generation. There is no clear trend explaining the difference between generated sentences with negative and positive values as the prior for $\mathbf{c}$. In general, the sentence generation seems to be a bit random when $prior(c) \sim N(0, 1)$, which implies the standard VAE architecture needs to incorporate a more representative distribution for $\mathbf{c}$ to have a chance at capturing even primitive emotion distinctions (such as positive and negative sentiment).

For the improved prior, namely the GMM with optimal mixture components, sentences are again conditionally generated. Because of the behavior of the GMM, even if the model can distinguish between emotions, it is difficult to see which emotions correspond to which mixture components. Regardless, some control is seen in the sentence generation:



A movie is nonexistent.
A film of ideas and wry comic mayhem.
A moving, reverent, and subtly different sequel.
A compelling film of musical passion against governmental odds.
A movie is clever, and completely charming and touching.
The film is well crafted, the cast is appealing.
A film of extravagant promise by georgian-israeli director dover kosashvili.
The film is brilliant in cannes.
It's not a bad movie.
It's a bad mannered, curiously adolescent movie.
A film of delicate interpersonal dances.
A true pleasure.
A taut, sobering film.
The film is about as a necessary enterprise.
A weird, arresting little gem.
It 's a great movie that 's 86 minutes long past.
A rip-roaring comedy that never rises above superficiality.

Figure 6: For every generated sentence, the first step is to sample from the GMM prior to get a value for latent variable $\mathbf{c}$. The sentences above are colored according to the sample's mixture component.

The goal was to generate sentences based on emotions, but the resulting sentences are split more between positive and negative sentiment. Generally, the green and magenta sentences have more negative, critical context, and the remaining have more positive, supportive context.

Note that the SST dataset is biased towards a relatively binary good and bad sentiment emotion distribution. Movie reviews generally carry either good or bad sentiment. Ambivalence is rarely seen in film review sentences, which is the suspected reason for why the final results separate by positive or negative but not by subtleties within different emotions.

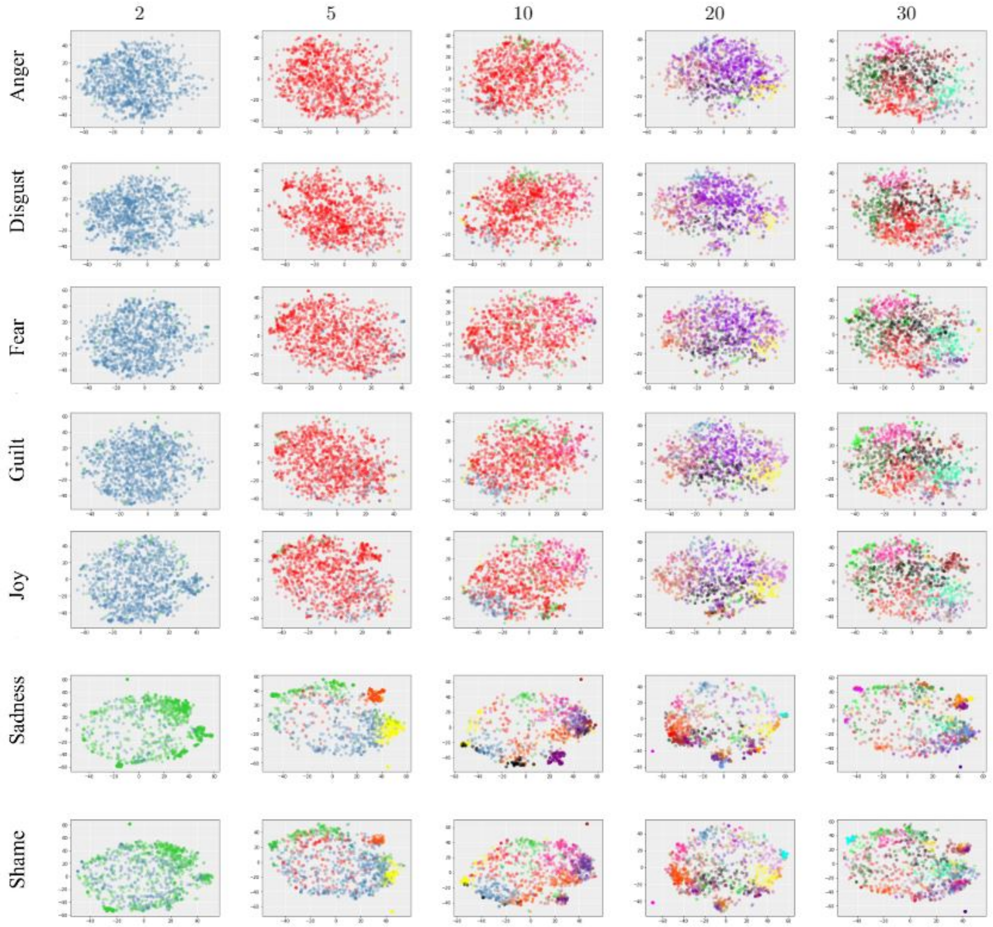


Figure 7: Distributions of mixture components for 7 emotions. The GMM was trained using 2, 5, 10, 20, and 30 components. Each color corresponds to a different mixture component.

Figure 7 shows the distributions of mixture components for seven emotions. The GMM was trained using different numbers of mixture components, and the plots above can be used to distinguish which numbers of mixture components give better results. Generally, for all numbers of mixture components, the distributions for joy, guilt, fear, disgust, and anger are similar, while the distributions for shame and sadness are similar. For 2, 5, and 10 mixture components, the distributions for joy, guilt, fear, disgust, and anger appear largely dominated by one mixture component and look nearly identical. Only at 20 and 30 mixture components do the plots exhibit some distinctions in distributions with different clusterings of mixture components for these five emotions. 30 mixture components depict the most obvious differences in distributions among all emotions.

## 4 Conclusion

To create a model that achieves emotion-guided sentence generation, the proposed VAE model extends the standard VAE model by adding a structured latent variable $c$ whose prior is taken as an emotion-guided GMM. Specifically, using a GMM with $k = 7$ mixture components trained on unlabeled sentences from the ISEAR dataset provides a decent GMM prior. Comparing the sentence generation with the emotion-guided GMM prior with a baseline $prior(c) \sim N(0, 1)$ shows that an emotion-guided GMM prior is able to at least introduce some level of sentiment control. As per the analysis, using a more initially separated dataset for training the emotion-guided GMM would allow more emotion-based control over sentence generation.

### 4.1 Future Research

Many aspects of the current pipeline can be improved. Training an emotion-guided GMM is challenging mainly because most word representations (i.e. word vectors) capture contextual higher level abstract information about words, as opposed to sentiment information. For instance, “happy” and “sad” are more similar than “happy” and “rainbow” because “happy” and “sad” are both emotions. Along those lines, datasets inputted into the GMM will be highly entangled, making it difficult for it to learn the distinction between points in an unsupervised fashion. We suggest a supervised/ semi-supervised technique that utilizes the emotion labels for the sentences. In addition, a dataset with many examples from few, very distinct emotion labels may produce better results. One of the main problems with the ISEAR dataset is that emotions tended to be grouped together due to their similarity, such as “anger” and “disgust."

Another area for improvement would be in the incorporation of the prior. Specifically, the latent variable **c** can be used to encode all emotion information. Assuming the prior (whether it is a GMM or some other probabilistic model) has learned emotion distinction well, there is still the problem of trying to capture a complex probabilistic model with one latent variable. If sentence generation can be conditioned on multiple latent variables, a more disentangled representation of our latent space can be learned. In this context, the latent space is an emotion space. This is one of the main problems in the area controllable text generation, but adding latent variables arbitrarily will essentially amount to overfitting, such that conditioning on all these latent variables may not actually allow more control over the generation. Therefore, it is important to balance these two to the right degree. Imagine conditioning the latent variable on another latent variable that captures more basic emotion information, such as good/bad sentiment. Then, based on the output of that latent variable, the prior can be conditioned so that it learns a distribution based on the general sentiment. This may allow more control over the sentence generation, as well as increase the model’s ability to learn a disentangled representation of the latent space.

## 5 Acknowledgments

We would like to thank Professor Yisong Yue and Eric Zhan for mentoring us throughout this project.

## 6 References

[1] Hu, Zhiting, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. "Toward Controlled Generation of Text." *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. 2017.

[2] Miao, Yishu, Edward Grefenstette, and Phil Blunsom. "Discovering discrete latent topics with neural variational inference." *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR.org, 2017.

[3] Scherer, K., and H. Wallbott. "The ISEAR questionnaire and codebook." Geneva Emotion Research Group (1997).

[4] Socher, Richard, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." *Proceedings of the 2013 conference on empirical methods in natural language processing*. 2013.

[5] Wang, Wenlin, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. "Topic-Guided Variational Autoencoders for Text Generation." 2019.

[6] Wu, Ho Chung, et al. "Interpreting tf-idf term weights as making relevance decisions." *ACM Transactions on Information Systems (TOIS)* 26.3 (2008): 13.

`https://github.com/chyu0818/cs159.git`