Sentence Generation and Classification with Variational Autoencoder and BERT

Keshav Rungta
Dept. of Electrical & Computer Engineering
University of California, San Diego
San Diego, CA 92092
krungta@ucsd.edu

Geeling Chau
Dept. of Electrical & Computer Engineering
University of California, San Diego
San Diego, CA 92092
gchau@ucsd.edu

Anshuman Dewangan
Dept. of Computer Science
University of California, San Diego
San Diego, CA 92092
adewanga@ucsd.edu

Margot Wagner
Dept. of Bioengineering
University of California, San Diego
San Diego, CA 92092
mwagner@ucsd.edu

Jin-Long Huang
Dept. of Physics
University of California, San Diego
San Diego, CA 92092
jih002@ucsd.edu

Abstract

In this paper, we explore logical and style-specific sentence generation. Given a sentence (premise), the goal is to write a hypothesis that either agrees, contradicts, or says nothing about the premise. In order to achieve this goal we use the *Contradictory, My Dear Watson* dataset from Kaggle and the Stanford Natural Language Inference (SNLI) dataset. We leverage a conditional variational autoencoder for text generation that can take a premise and class label (entail, contradict, or neutral) to generate a hypothesis specific to that class type. We use the classification results from a BERT model in order to evaluate the style transfer to generating the sentences. While our BERT classifier achieved a loss of 0.291 and accuracy of 89.1% on the ground truth dataset, the classifier achieved a loss of 3.513 and accuracy of 38.7% on our best-performing conditional VAE generated examples, performing only slightly better than random chance. Our results suggest that more work needs to be done to algorithmically generate style-specific sentences following specific logical rules.

1 Introduction

With the release of GPT-3 by OpenAI, much excitement has surrounded the topic of text generation. With a short description, GPT-3 can generate paragraphs of text output which at times is indistinguishable from human-generated text. However, the generated text can suffer from being contradictory with itself or contradicting with historical facts. There is no current system to detect whether a language model is generating text that is sensible or measure how much it potentially contradicts other sources. Thus, it is important that we explore this topic of building a model to identify contradictory sentences as a starting point for incorporating logical soundness and a measure of contradiction to our language models.

In this work, we aim to explore style-specific text generation using variational autoencoders using the *Contradictory, My Dear Watson* dataset on Kaggle and the *Stanford Natural Language Inference* (SNLI) dataset. Both datasets consist of pairs of sentences, a premise and a hypothesis, with class labels categorizing the relationship between the two sentences: entailment, neutral, or contradiction. Our overarching goal is to generate a style-specific hypothesis given a premise and a class label. We then evaluate these generations using a BERT model which is used to classify the resulting generations using the ground truth premises.

For our approach, we break this problem into three parts:

1. Define a single variational auto-encoder that generates a hypothesis given a premise (with no regard to class label). *Evaluation metric*: BLEU scores compared to ”real” data.
2. Define three variational autoencoders, one for each class label, that generates a hypothesis given a premise of the corresponding class. *Evaluation metric*: BLEU scores compared to ”real” data; accuracy score when passed into a simple classifier using BERT.
3. Define a conditional variational auto-encoder that generates a hypothesis given a premise and specific class label. *Evaluation metric*: BLEU scores compared to ”real” data; accuracy score when passed into a simple classifier using BERT.

The results not only give insight into our ability to generate realistic, style-specific text, but can also be used for data augmentation into a classification task.

2 Related Works

We begin by gaining an understanding of variational autoencoders. Jaan Altosaar’s blog post gives basic intuition about the model [1]. *Topic-Guided Variational Auto-Encoder for Text Generation* by Wang et. al. gives an example of how variational autoencoders can be used for style-specific text generation [2].

In terms of implementation, William Falcon’s blog post provides an example of a PyTorch implementation of a basic variational auto-encoder [3]. Vadim Borosov gives an example of how to use a conditional variational auto-encoder [4].

To get deeper understanding about VAE, we read original VAE paper [5]. For usage of VAE in Natural Language Processing(NLP) context, we referred to [6]. From those papers we understood the technical detail about VAE, like the reparameterization trick and what loss function should we use.

Examples of how to use BERT to classify sentences are given on the Kaggle website [7]. We followed Huggingface’s documentation [8] to implement and fine tune BERT. In order to gain a better understanding of BERT and see how it is used in real-world problems, we read the original BERT paper [9].

3 Methods

3.1 Dataset

We use the Kaggle dataset entitled “Contradictory, My Dear Watson,” (MDW) which is currently part of a “Getting Started Code Competition” on Kaggle [7] and the Stanford Natural Language Inference (SNLI) dataset [10]. The challenge is to train a model that is able to determine relationships between sentences. Given two sentences, there are three ways they could possibly be related: one sentence could entail another, it could contradict another, or the two sentences could neither support nor contradict, i.e. be neutral (but still related). The dataset consists of pairs of sentences containing a premise and a hypothesis with one of three labels: entailment (0), neutral (1), or contradiction (2).

As an example, given the following premise:

> He came, he opened the door and I remember looking back and seeing the expression on his face, and I could tell that he was disappointed.

If the hypothesis follows as true from the information given in the premise, it is considered *entailment*. An example of this would be the following:

> Just by the look on his face when he came through the door I just knew that he was let down.

If the hypothesis may or may not be true but cannot be decided based on the given premise, it is *neutral*. A sentence like this would be:

> He was trying not to make us feel guilty but we knew we had caused him trouble.

Lastly, if the hypothesis goes against the initial premise, it is a *contradiction*. A contradictory example is:

> He was so excited and bursting with joy that he practically knocked the door off it's frame.

Although there could be many more neutral sentences that would not provide information to the premise, the dataset specifically use neutral hypothesis that are still somewhat semantically related to the premise. See a subset of the neutral hypothesis with their premise in **Table 1**. Thus, we can still use neutral as a class, different from simply generating random unrelated sentences.

premise	**neutral hypothesis**
The four Javis children? asked Severn.	Severn knows everything about the Jarvis children.
Be of good cheer,	Be of good cheer, for beer time is near.
evaluation questions.	There are evaluation questions on the topic.
The truth?	Will you tell the truth?

Table 1: A sub-sample of premises with their neutral hypothesis. This shows that neutral hypothesis are still related to the premise, although not supporting or contradicting.

In MDW, there are a total of **12120 unique pairs** and **8209 unique premises**, meaning that not every premise will have a duplicate entry that has a different hypothesis. The dataset includes premise-hypothesis pairs in fifteen languages: Arabic, Bulgarian, Chinese, German, Greek, English, Spanish, French, Hindi, Russian, Swahili, Thai, Turkish, Urdu, and Vietnamese which could be useful to extend this project, but we will focus on the English entries (**6870 unique pairs**), as it more than 56% of the dataset **Figure 1**. If the English dataset is too small, one option could also be to augment the data by using Google Translate on the other language entries [11].

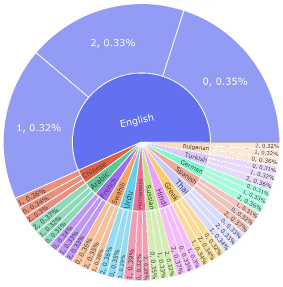


Figure 1: Language distribution in MDW dataset. Figure source: [11]

Since we only have 12k samples from MDW, we wanted to boost the amount of data by translating the non-English entries to English. We did this by calling the Google Translate API to convert the foreign text into English.

The SNLI dataset contains 570k English sentence pairs with the same labels of entailment, neutral, and contradictory as we originally saw in MDW. This dataset was curated by providing ”image captions” to Amazon Mechanical Turks to have humans generate sentences that agree, provide no info, or contradict the original caption. The dataset is then manually ranked to ensure quality of captions. Due to the large corpus of sentences, it is important to keep in mind possible biases that might be present in the data. By using Huggingface Datasets [12], we were easily able to obtain this data and output to a similar CSV format as the MDW was to work with our original dataloaders. In **Table 2**, we see an example set of premise and hypothesis from SNLI.

premise	This church choir sings to the masses as they sing joyous songs from the book at a church.
hypothesis (0: entailment)	The church is filled with song.
hypothesis (1: neutral)	The church has cracks in the ceiling.
hypothesis (2: contradictory)	A choir singing at a baseball game.

Table 2: Example premise and hypothesis from SNLI.

3.2 Hypothesis Generation Model - Variational Autoencoder

To generate a hypothesis given a premise, we built a variational autoencoder (VAE) (**Figure 2**). This model has the advantage of mapping to a smaller latent space rather than a strict point as with the non-variational autoencoder (NVAE), allowing for different yet still sensical sentences to be produced.

The VAE will have an LSTM encoder and decoder, along with two separate linear layers representing mean and variances for the latent space. The encoder will take the premise as input and output an embedding in a Gaussian latent space. The decoder will use this lower dimensional representation to generate a predicted hypothesis with teacher forcing for loss calculation and without teacher forcing for BLEU score calculation. The backpropagation loss will be the Cross Entropy Loss regularized by the Kulback-Leibler divergence term to prevent the latent space from shrinking to a point (i.e. standard deviation approaches 0).

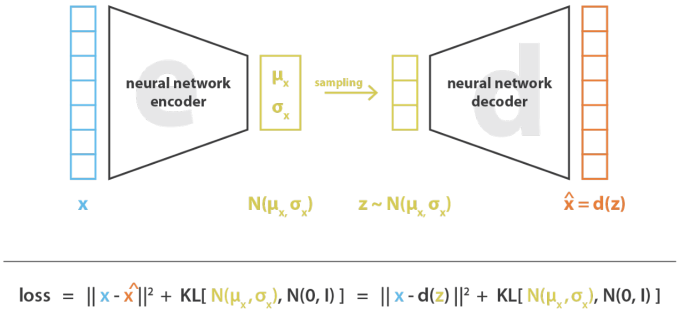

neural network encoder
e
μ_x
σ_x
sampling
neural network decoder
d
x
$N(\mu_x, \sigma_x)$
$z \sim N(\mu_x, \sigma_x)$
$\hat{x} = d(z)$
loss = $|| x - \hat{x} ||^2$ + KL[$N(\mu_x, \sigma_x)$, N(0, I)] = $|| x - d(z) ||^2$ + KL[$N(\mu_x, \sigma_x)$, N(0, I)]

Figure 2: VAE architecture and loss with KL regularisation. Figure source: [13]

This class-agnostic VAE will act as a baseline for generating hypothesis not from any particular class and for evaluating our model's efficacy for generating sensical sentences similar to the dataset's. To explore the model's ability to characterize the three types of hypothesis, we approach this in two ways: (1) build three VAE models trained and tested on the subset of the data specific to each class and (2) build a conditional VAE that takes the class as an input to output a predicted hypothesis for that class.

With these three approaches: class-agnostic baseline, 3 separate VAE models, and a conditional VAE, we can generate hypothesis for any given class and premise. To evaluate the efficacy of each model, we use loss, BLEU scores and classification accuracy with our in house BERT Classification Model as discussed in the next section.

In order to achieve the best performance possible, we explored variations of our baseline (class-agnostic) VAE model. First, we varied the learning rate, but left other hyperparametrs (hidden size, embedding size) the same due to time and compute constraints. Next, we experimented with various inputs into our LSTM decoder. For the baseline, we used only the *a*ctual hypothesis as the input into our decoder. In Version 2, we *c*oncatenated the actual hypothesis with the embedded output of our encoder as the input into our decoder. Lastly, we explored the effects of different dataset sizes on our performance.

3.3 Premise-Hypothesis Classification Model

To evaluate our model's ability to classify entailment, neutral, and contradictory hypothesis, we will use a fine-tuned BERT classification model. **Figure 3** illustrates how we can use a pre-trained BERT model to be fine tuned for this classification problem. A BERT model will be ideal for this case because it can take in concatenated pairs of sentences to classify their relationship. With our conditional variational auto-encoder, we can generate a hypothesis given a class (0, 1, or 2) and the premise, which we can then see if our fine-tuned BERT model will also classify the premise+hypothesis as the same class.

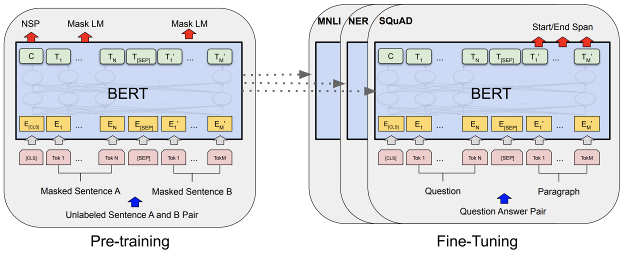


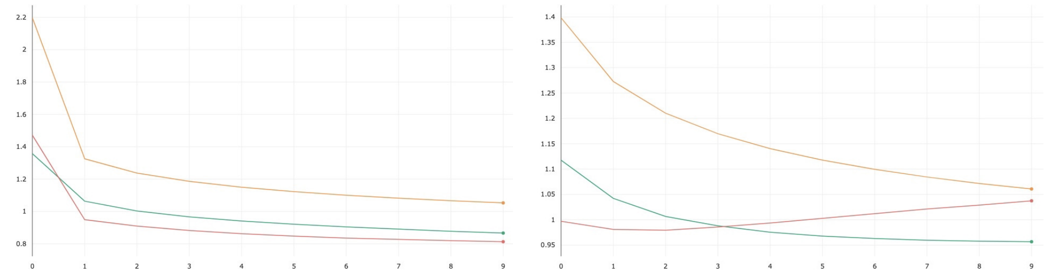
Figure 3: BERT Architecture that will be fine-tuned with our premise-hypothesis pairs (rather than the question-paragraph) pairs shown here. C is the 'Next Sentence Prediction' which will represent a classification for our hypothesis. Figure source: [9]

With this model trained on classifying our train data, we will evaluate its accuracy on the dataset and compare those metrics with accuracy on the generated hypothesis from our generation models.

4 Results

4.1 Baseline VAE Model

4.1.1 Hyperparameter Tuning



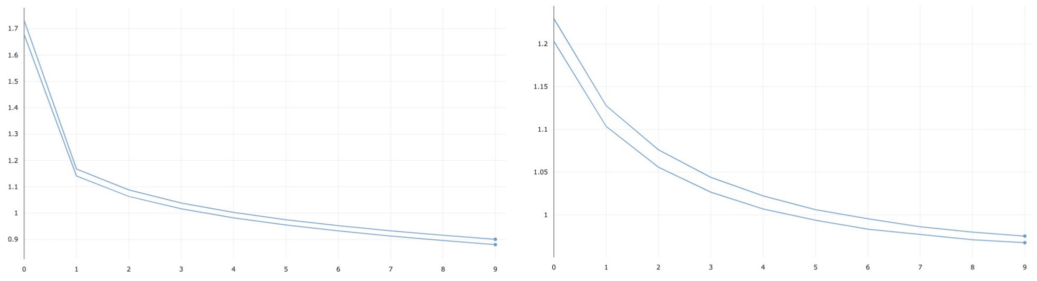
(a) Training Loss

(b) Validation Loss

Figure 4: Loss for baseline VAE model with 512 hidden size and 300 embedding size while varying learning rate. Legend: Yellow = 5e-5, Green = 5e-4, Red = 5e-3.

To begin, we trained our baseline (class-agnostic) model with 512 hidden size and 300 embedding size for 10 epochs. **Figure 4** depicts the results for varying learning rates. All loss plots were well-behaved with training loss asymptomatically decreasing and validation loss higher than training loss. A learning rate of 5e-3 shows overfitting with validation loss curve going up after 2 epochs. A learning rate of 5e-5 shows underfitting with the loss curve decreasing too slowly. The best learning rate was in between at 5e-4 as evident by the lowest validation loss after 10 epochs. Thus, we continued all our VAE generation models with learning rate of 5e-4.

4.1.2 Different LSTM Input/Output Architectures



(a) Training Loss

(b) Validation Loss

Figure 5: Loss for baseline VAE model with 5e-4 learning rate, 512 hidden size, and 300 embedding size while varying architecture. Legend: Bottom Curve = Version 1, using hypothesis only as input into decoder; Top Curve = Version 2, concatenating embedded output of encoder with the hypothesis as input into decoder.

Next, we experimented with various inputs into our LSTM decoder. As a reminder, in Version 1, we used only the actual hypothesis as the input into our decoder. In Version 2, we concatenated the actual hypothesis with the embedded output of our encoder as the input into our decoder. **Figure 4** depicts that the performance of both versions are comparable, with the validation loss for Version 1 slighly lower than that of Version 2. Consequently, we used the simpler Version 1 for the rest of our models.

We also tried the following architecture changes: 1) replace the decoder LSTM hidden and cell state with the output of the encoder at every time step; 2) replace the decoder LSTM hidden state (not cell state) with the output of the encoder at every time step; 3) concatenate the output of the encoder with the decoder LSTM's hidden state at every time step. While results are not shown in this report, all three versions performed more poorly than Version 1 suggested above and were discarded.

4.1.3 Data Augmentation

Experiment	Test Loss	BLEU-1	BLEU-4
MDW English Only	2.754	**34.73**	**5.90**
MDW w/ Translations	2.47	21.44	2.91
SNLI	**0.957**	26.0	3.74

Table 3: Performance of baseline VAE on different sized datasets using 5e-4 learning rate, 512 hidden size, and 300 embedding size. BLEU scores reported for deterministic caption generation.

Lastly, we can see the results of our data augmentation attempts in **Table 3**. These results show that amplifying the data helps a with test loss but potentially more slowly for BLEU scores, but this might be due to the smaller and more English-friendly test set that the MDW English Only subsample had which allowed for higher BLEU scores. In general, the SNLI dataset which includes 550k train samples compared to 12k samples in MDW, achieved the best test loss. Since the translated texts did not resemble native English as closely as we would have hoped and due to the minor improvements in metrics, we chose to augment our data primarily by using the SNLI dataset.

4.1.4 Summary of VAE Results

Using the architecture decisions made above, we ran different variations of the baseline VAE model using examples from specific classes only and a variation that attempts to learn the style of each

VAE Model	Test Loss	BLEU-1	BLEU-4
Baseline (class agnostic)	0.957	**26.0**	**3.74**
Class 0 examples only	**0.846**	25.8	3.74
Class 1 examples only	1.164	25.9	3.74
Class 2 examples only	0.967	24.6	3.48
Class Conditional	0.954	26.0	3.73

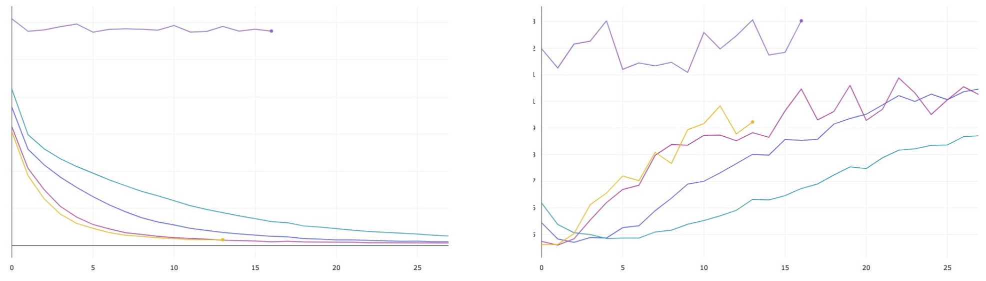
Table 4: Performance of different VAE models using 5e-4 learning rate, 512 hidden size, and 300 embedding size using the SNLI dataset. BLEU scores reported for deterministic caption generation.

class. In **Table 4**, we observe that the performance results are comparable across all model variations. This is expected because the dataset size is quite large so having a smaller amount of data to train on would not have a large effect on the model's ability to train. The slight fluctuation in loss can be accounted for by the potentially more similar or predictable hypothesis in certain label categories than others. For example, entailment might have hypothesis that are more similar to the premise which could make it easier for the model to guess the hypothesis prediction.

We also note that the BLEU scores are generally low (compared to that of the image captioning task from programming assignment 4, for example). There isn't really a good method to interpret these results, as the goal of our VAE model is to generate new hypotheses based on the premise. Consequently, while we want our captions to somewhat resemble the ground truth hypotheses, we can accept if the BLEU scores are low due to the creative differences the VAE model is trying to generate. In **Appendix: Generated Caption Examples**, we capture some logical and illogical generated sentences for the different variations of VAE models. As a positive, our models were able to learn the grammatical structure of a sentence, with a noun executing a verb. However, to a human, most of the generated hypotheses do not relate to the premise given, regardless of the class label passed. In particular, the generated sentences are not able to match the noun used in the original premise. The generated sentences follow similar patterns; the nouns are predominantly "a man" and the verbs are related to playing an instrument, wearing a particular article of clothing, or sitting or standing. It intuitively makes sense why the model has learned these particular actions, as they are heavily over-represented in the dataset and chances are have some relationship to the premise.

4.2 BERT Classification Model

4.2.1 Ground Truth Sentences



(a) Train Loss

(b) Validation Loss

Figure 6: Loss for BERT classification model with default parameters while varying learning rate. Legend (from top to bottom of train loss): Magenta = 5e-3, Blue = 5e-6, Purple = 1e-5, Red = 3e-5, Yellow = 5e-5.

Figure 6 depicts the results for varying learning rates when training our BERT classification model on 10% of the SNLI dataset. All loss plots were well-behaved with training loss asymptomatically decreasing and validation loss higher than training loss. Although the model appears to overfit the data in under 5 epochs, a learning rate of 5e-5 resulted in the lowest validation loss and was used for all experiments going forward.

Class Label	Test Loss	Acc
All classes	0.291	0.891
Class 0 examples only	0.240	0.913
Class 1 examples only	0.320	0.844
Class 2 examples only	0.139	0.919

Table 5: BERT classification performance on SNLI ground truth dataset.

Table 5 summarizes BERT classification performance on ground truth labels of the SNLI dataset. The BERT model achieves an astounding 89.1% classification accuracy on all class examples with a loss of 0.291. When presented class-specific examples, the model performed extremely well on class 0 (entailment) and class 2 (contradiction) examples with over 91% accuracy. The model relatively strugged to classify class 1 examples (neutral) with an accuracy of 84.4% and loss of 0.32. This accuracy is good enough to rank us at 22 in the MDW Kaggle Competition.

4.2.2 VAE-Generated Sentences

Temperature	VAE Test Loss	VAE BLEU-1	VAE BLEU-4	BERT Loss	BERT Acc
0	0.939	25.6	3.70	3.95	0.357
0.25	0.939	23.7	3.50	3.513	0.387
0.5	0.939	21.8	3.40	3.313	0.346
0.75	0.939	18.5	3.06	3.567	0.346
1	0.939	15.5	2.80	3.77	0.316

Table 6: BERT classification performance on sentences generated by conditional VAE using different temperatures and SNLI dataset.

With BERT model performance on the ground truth in mind, we subsequently attempt to classify the generated sentences from our VAE model. **Table 6** details the performance of VAE-generated sentences using the conditional VAE model while varying temperature. We firstly note that the VAE test loss is the same for all temperatures; this aligns with our intuition as the generated predictions are not used in the loss calculation. Second, VAE BLEU scores generally decrease as temperature increases. As mentioned above, it can be hard to interpret these results as a positive or negative in terms of model performance, as we do want generated sentences that are sufficiently different from the original hypothesis. Third, our BERT accuracy scores are extremely low, doing just about as well as random chance. This suggests that our conditional VAE model was not successful in learning the "style" of each class. However, we do see that a moderate temperature of 0.25 results in the highest BERT accuracy of 38.7%, which might indicate some style components were learned.

Class Label	VAE Test Loss	VAE BLEU-1	VAE BLEU-4	BERT Loss	BERT Acc
0	0.864	25.2	3.78	3.93	0.151
1	1.11	24.4	3.57	3.68	0.218
2	0.870	25.1	3.87	0.988	0.781

Table 7: BERT classification performance on sentences generated by class-specific VAE using temperature = 0.25 and SNLI dataset.

For our last analysis, we evaluated BERT classification performance on our class-specific VAE models to see how our VAE performed in generating examples of a specific class. In these experiments, the ideal result is that the BERT classification model predicts the same class label for the entire dataset. **Table 7** suggests that our VAE model does extremely poorly in generated class 0 (entailment) and class 1 (neutral) sentences, resulting in a BERT accuracy of 15.1% and 21.8%, respectively. As a reminder, the BERT model achieved 91.3% and 81.4% accuracy on these classes on ground truth examples, respectively. Interestingly, the BERT accuracy on our VAE generated examples for class 2 (contradiction) is 78.1%, not too much worse than the ground truth accuracy of 91.9%. We believe this is because in the ground truth dataset, a lot of the class 2 premise/hypothesis pairs are quite irrelevant to each other whereas class 0 and class 2 pairs need to have a strong correlation between the pairs. Since our model mostly generates hypotheses that have little resemblance to the premise, class 2 is the natural class for the BERT classifier to assign to our generated examples, resulting in a high accuracy.

5 Conclusion

With our best-performing conditional VAE model generating sentences that result in chance-level BERT classification accuracy, we conclude that more work needs to be done in order to enable style-transfer for logical arguments. Despite our inability to transfer style or maintain logic, the main success of this paper is to generate realistic-sounding sentences. For future work, we aim to explore alternative architectures, including using a transformer or BERT model for our VAE encoder/decoder that might be better able to pay ”attention” to the nouns of the sentence instead of an LSTM.

6 Individual Contributions

We all contributed in equal amounts to each of the different stages of the project. Every line of code was essentially reviewed and helped in debugging by every person (with logic) and the reports were also brainstormed as a group.

Geeling Chau

- Implemented dataloaders for MDW and SNLI datasets.
- Implemented nonvar autoencoder and train, val, test pipelines for sentence generation.
- Ran experiments for VAE model training.
- Wrote Methods for VAE and BERT, Results for VAE Sentence Generation.

Keshav Rungta

- Implemented the structure of the project
- Debugged the non-variational autonoencoder
- Wrote code to translate MDW dataset using Google Translate API
- Wrote code to generate sentences from different models to feed into BERT for classification

Anshuman Dewangan

- Debugged VAE; implemented loss function; implemented conditional VAE
- Attempted many architecture changes to get best performance
- Implemented experimental set-up; Ran experiments to generate test results
- Wrote Results, Conclusion, Appendix, part of Introduction, Progress Report
- Attended office hours to get guidance on project structure and implementation

Margot Wagner

- Implemented BERT model architecture
- Wrote part of introduction and methods
- Created most of powerpoint
- Implemented BERT accuracy scoring

Jin-Long Huang

- Added reparameterization trick to make autoencoder variational
- Debugged SNLI datasets
- Implemented train, val and test pipelines for BERT model
- Fine-tune BERT model

References

[1] Jaan Altosaar. *Tutorial - What is a Variational Autoencoder?*, August 2016. https://doi.org/10.5281/zenodo.4462916.

[2] Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. Topic-guided variational auto-encoder for text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 166–177, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[3] William Falcon. Variational autoencoder demystified with pytorch implementation., Dec 2020. https://towardsdatascience.com/variational-autoencoder-demystified-with-pytorch-implementation-3a06bee395ed.

[4] Vadim Borosov. Conditional variational autoencoder (cvae), Nov 2019. http://vadimborisov.com/conditional-variational-autoencoder-cvae.html.

[5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[6] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[7] Kaggle. Contradictory, my dear watson, Aug 2020. https://www.kaggle.com/c/contradictory-my-dear-watson/overview.

[8] The Hugging Face Team. Training and fine-tuning, 2020. https://huggingface.co/transformers/training.html#fine-tuning-in-native-pytorch.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[10] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[11] Narendra. Nlp augmenter, 5 fold bert &; translator, Aug 2020. https://www.kaggle.com/narendrageek/nlp-augmenter-5-fold-bert-translator.

[12] Thomas Wolf, Quentin Lhoest, Patrick von Platen, Yacine Jernite, Mariama Drame, Julien Plu, Julien Chaumond, Clement Delangue, Clara Ma, Abhishek Thakur, Suraj Patil, Joe Davison, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angie McMillan-Major, Simon Brandeis, Sylvain Gugger, François Lagunas, Lysandre Debut, Morgan Funtowicz, Anthony Moi, Sasha Rush, Philipp Schmidd, Pierric Cistac, Victor Muštar, Jeff Boudier, and Anna Tordjmann. Datasets. *GitHub. Note: https://github.com/huggingface/datasets*, 1, 2020.

[13] Joseph Rocca. Understanding variational autoencoders (vaes), Jul 2020. https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73.

[14] The Hugging Face Team. *BERT Tokenizer Documentation*, 2020. https://huggingface.co/transformers/model_doc/bert.html?highlight=berttokenizer#berttokenizer.

Appendix

Generated Caption Examples

premise	a man in a black hat opens his mouth.
actual hypothesis (class 1)	The governor prepared to deliver the speech that would deliver the votes.
good neutral	a man is looking at a camera.
premise	a young infant cries while having his or her pajamas button.
actual hypothesis (class 2)	A young baby smiles.
bad contradiction	a man is standing outside.

Table 8: One “good” and one “bad” generated hypotheses from baseline (class-agnostic)

premise	a man and a child are laughing at each other.
actual hypothesis (class 0)	Two people are laughing.
good entailment	a man is holding a child.
premise	a woman holds a newspaper that says ”real change”
actual hypothesis (class 0)	a woman holding a newspaper that says ”real change”
bad entailment	a man is wearing a shirt.

Table 9: One “good” and one “bad” generated hypotheses from class 0-specific VAE.

premise	a man talking into a microphone with a woman standing next to him.
actual hypothesis (class 1)	The woman is sitting in the chair next to the podium.
good neutral	the man is a professional musician.
premise	a woman in black reviews a message as she walks to work.
actual hypothesis (class 1)	The woman in black is being fired via text message.
bad neutral	a man is trying to fix a broken component.

Table 10: One “good” and one “bad” generated hypotheses from class 1-specific VAE.

premise	a man wearing a white shirt and a blue jeans reading a newspaper while standing
actual hypothesis (class 2)	A man is sitting down reading a newspaper.
good contradiction	the man is sitting on the couch.
premise	the small dog is running across the lawn.
actual hypothesis (class 2)	A cat is running up a tree.
bad contradiction	the man is wearing a red shirt.

Table 11: One “good” and one “bad” generated hypotheses from class 2-specific VAE.

premise	a man on a bicycle rides past a park, with a group of people in the background.
actual hypothesis (class 2)	a guy rides his bike in the middle of a park.
good contradiction	a man is sitting on a bench.
premise	a small dog runs to catch a ball.
actual hypothesis (class 0)	A little dog chases a ball.
bad entailment	a woman is holding a child.

Table 12: One “good” and one “bad” generated hypotheses from conditional VAE.