Sequence to Better Sequence: Style and Content Separation

Wenyue Hua

Abstract

This project reimplements the paper Sequence to Better Sequence: Continuous Revision of Combinatorial Structures (Mueller et al., 2017) which attempts improve discrete sequences in a specific way while retaining the original semantics as much as possible. However, results of the model shows that the original semantics is modified undesirably. The project thus tries to improve the model by separating "style" and "content" of sequence. The basic structure of the model and the improvement on the model are presented. Experiments based on three synthetic datasets are presented to show that the hypothetical separation of style and content does help retaining the semantics of sequences.

1. Background

Sequence to Better Sequence paper presents a model that can be efficiently used to revise a discrete sequence in order to improve its associated outcome after learning on observations of (sequence, outcome) pairs. Applications include modifying a gramatically problematic sentence to a grammatical one, improving an amino-acid sequence of synthetic protein for better clinical efficacy score and etc.

Here is the formal definition of the task: given a discrete sequence x_0, revise it to x^* in order to improve its associated outcome y:

$$x^* = argmax_{x \in \mathcal{C}_{x_0}} E[y \mid X = x]$$

where $\mathcal{C}_{x_0}$ is a set of feasible revisions to ensure that x^* remains natural and a minor revision of x_0.

1.1. Model Structure

The model uses a generative modeling framework which transforms the discrete optimization problem to a differentiable optimization problem by leveraging continuous-valued latent representations learned using neural networks. A variational autoencoder is used to learn a continuous latent representation of a sequence; a simple feed forward neural network based on the latent representation computes a score evaluating the outcome.

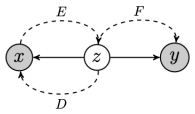

Figure 1. model structure

Given a sequence x, the encoder $\mathcal{E}$ produces a distribution over continuous latent representations z, which is a d-dimensional vector, conditioned on x. We assume z values form a Gaussian distribution. Given a latent representation z, the decoder $\mathcal{D}$ generates a sequence. It is expected that the generated x is a realistic/natural sequence given any z with reasonable prior probability. Given a latent representation z, the outcome predictor $\mathcal{F}$ predicts the outcome score y based on the learnt latent representation. Fig. 1 demonstrates the structure of the model.

The training loss consists of four parts: (1) reconstruction loss of the variational autoencoder to learn informative latent representations of sequences (2) mean square loss (MSE) of the outcome predictor to learn precise a score prediction (3) regularization loss of the variational autoencoder to ensure that the posterior distribution of latent representations of each sequence is close to Gaussian distribution (4) invariance loss to enforce outcome-predictions to be invariant to variation introduced by the encoding-decoding process.

- Reconstruction Loss:

$$\mathcal{L}_{reconstruction} = E_{z \sim q_{\mathcal{E}}(z|x)}[\log p_{\mathcal{D}}(x \mid z)]$$

- Mean Square Loss:

$$\mathcal{L}_{mse}(x, y) = [y - \mathcal{F}(mean(\mathcal{E}(x)))]^2$$

- Regularization Loss:

$$\mathcal{L}_{regularization} = KL(q_{\mathcal{E}}(z \mid x)||p(z))$$

- Invariance loss:

$$\mathcal{L}_{inv} = E_{z \sim p_Z}[\mathcal{F}(z) - \mathcal{F}(mean(\mathcal{E}(\mathcal{D}(z))))]^2$$

1.2. Sequence Optimization

Given a trained model and a discrete sequence x_0, the sequence is modified by applying optimization on its latent representation iteratively. After reaching a local optimum, decode the optimized latent representation to obtain the modified sequence. To ensure that all iterates remain in a feasible region $\mathcal{C}_{x_0}$, this optimization process in constrained. More details can be found in the original paper.

2. Style and Content Separation

Based on some results of the model, we find that although a sequence is modified to achieve a better outcome score, the original semantics is also changed unexpectedly. Below is an example where the model is trained to modify modern English text to a Shakespeare style. The sentence "where are you, henry??" is modified to "where art thou, keeper??", where the modified text has a different semantics from the original text.

# Steps	Decoded Sentence
x_0	**where are you, henry??**
100	where are you, henry??
1000	where are you, royal??
5000	where art thou now?
10000	which cannot come, you of thee?
x^*	where art thou, keeper??
x_0	**you are both the same size.**
100	you are both the same.
1000	you are both wretched.
5000	you are both the king.
10000	you are both these are very.
x^*	you are both wretched men.

Figure 2. Optimize sentences to Shakespeare style

This result motivates a simple modification on the original model: separation of a latent representation that can encode style and content in different subspaces, *i.e.* one part of the latent representation, z_1, encodes only information related to the outcome score, and the other part, z_2, encodes only the information unrelated to the outcome score. When applying optimization on the latent representation, we only optimize z_1. We hope the decoded sequence remains faithful to the semantics of the original sequence. The new model is referred to as "separation model" in the text below.

The change on the model is minor. The two loss functions below are slightly modified as below:

- Modified Mean Square Loss:

$$\mathcal{L}_{mse}(x, y) = [y - \mathcal{F}(mean(z_1))]^2$$

- Modified Regularization Loss:

$$\mathcal{L}_{regularization} = KL(q_{\mathcal{E}_1}(z_1 \mid x)||p(z_1)) + KL(q_{\mathcal{E}_2}(z_2 \mid x)||p(z_2))$$

The sequence optimization process is applied only on z_1 in inference time.

3. Implementation

The model is implemented in Pytorch. The variational autoencoder is a 1-layer LSTM, which is tested to have better results than RNN or GRU in this project. The hidden dimension is 256 and the latent dimension is 128. In the separation model, the style-related representation has dimension 16 and the content-related representation has dimension 112. We use AdamW optimizer with learning rate 1e-3 without learning rate scheduler for 30 - 100 epochs. When applying optimization in the inference time, the learning rate is 0.05 for 1,000 steps.

4. Experiment Result

I compare the original model and the separation model in three different synthetic datasets each having 100,000 training data and 1,000 test data. For each dataset, I construct a natural distribution p_X over sequences of lengths 10-20 whose elements stem from the vocabulary S = $\{A, B, C, D, E, F, G, H, I, J\}$ via different probabilistic grammars.

4.1. Counting A dataset

In the first dataset, for each sequence, the associated outcome y is simply the number of times A appears in the sequence, same as the synthetic dataset in the original paper. Table.1 is the statistics of the outcome scores of the dataset:

	mean	median	standard deviation
train	0.47	0.6	0.34
test	0.371	0.61	0.31

Table 1. statistics of outcome scores

Based on the probabilistics grammar, the statistics of negative log likelihood of all generated sequences is presented:

	mean	median	standard deviation
train	37.22	38.6	6.27
test	33.95	33.42	7.63

Table 2. statistics of negative log likelihood

To compare the two models, I first compare the improvements of the outcome scores on average of all data points in Fig.3. Although the separation model improves faster at the beginning, the two models converge to similar improvement after 1,000 steps.

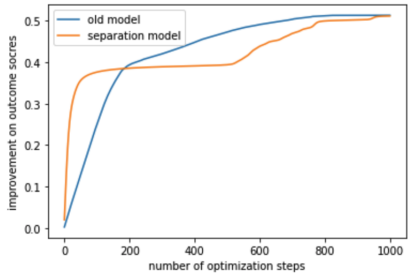

Figure 3. outcome improvement on Couting A dataset

To compare whether the modified sequence remains faithful to the original sequence semantically, I compare the proportion of "effective" modifications among all modifications on average for each sequence. "Effective" modifications are modifications directly affecting the outcome scores. In this dataset, if a modification changes a non-A symbol to an A, it is count as an effective modification. Although sequences are generated from a probabilistic grammar and ineffective modifications are inevitable for sequences to remain natural, if one model generates more ineffective modifications than the other model, I interpret it as this model changes the semantics more than the other. Fig.4 is the result.

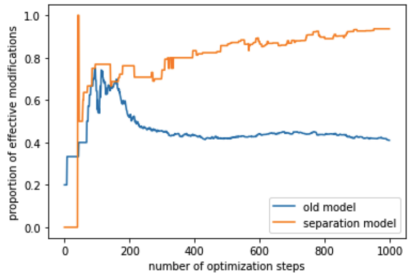

Figure 4. proportion of effective modifications

This graph shows that the separation model modifies the original sequence more effectively. The proportion converges to about 0.9 after 1,000 steps, while the old model changes more symbols "ineffectively" and the proportion converges to about 0.4.

However, better effective modification rate comes at the cost of higher negative log likelihood, i.e., the generated sequence is less natural under the probabilistic grammar. Fig.5 shows the result.

In this graph, sequences generated by the separation model become less natural while those generated by the old model is becoming more natural.

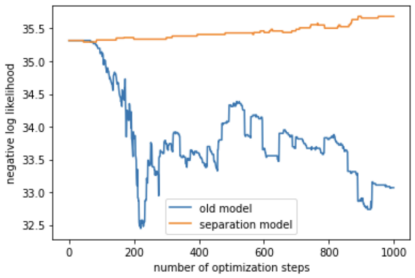

Figure 5. negative log likelihood

4.2. Counting A in first half dataset

This dataset is almost the same as the Counting A dataset except that the associated outcome y is the number of times A appears in the first half of the sequence. This is the statistics of the outcome scores of the dataset:

	mean	median	standard deviation
train	0.27	0.17	0.25
test	0.3	0.3	0.23

Table 3. statistics of outcome scores

The statistics of negative log likelihood of generated sequences are the same as that in the Counting A dataset.

Fig.6 is the graph for outcome improvements.

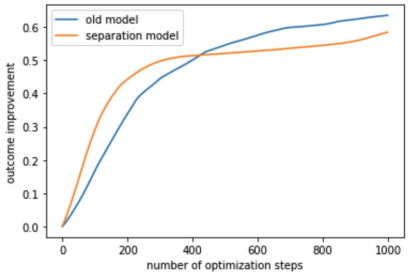

Figure 6. outcome improvement on Counting A in first half dataset

Fig.7 is the proportion of effective modifications of the two models and Fig.8 shows the negative log likelihood of generated sequences of the two models.

In this dataset, similar conclusions can be made about the two models: outcome improvement converge to similar number after 1,000 steps but the separation models make more effective modifications than the old model. The negative log likelihood of sequences generated are becoming smaller for both models, where the old model performs better.

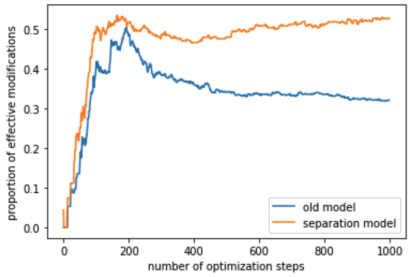

Figure 7. proportion of effective modifications

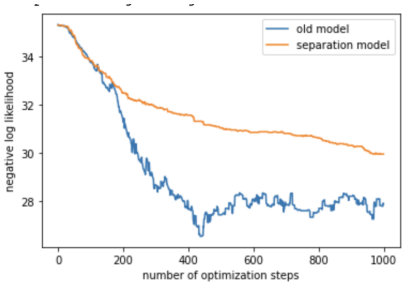

Figure 8. negative log likelihood

4.3. Comparison dataset

This dataset uses a different probabilistic grammar from the two datasets above. The associated outcome y is the proportion between number of occurrences of A and the number of occurrences of F. If A occurs more than F in a sequence, the score is 1. This is the statistics of the outcome scores of the dataset:

	mean	median	standard deviation
train	0.65	1	0.45
test	0.57	1	0.48

Table 4. statistics of outcome scores

The statistics of the negative log likelihood of all generated sequences are presented here:

	mean	median	standard deviation
train	37.81	39.76	6.19
test	33.44	33.41	7.61

Table 5. statistics of negative log likelihood

Fig.9 is the graph for outcome improvement. In this dataset, the separation model improves faster than the old model. It converges to about 0.45 while the old model only reaches about 0.25 after 1,000 steps.

Fig.10 is the graph showing proportion of effective modifications. The two models do no differ much on this dataset,

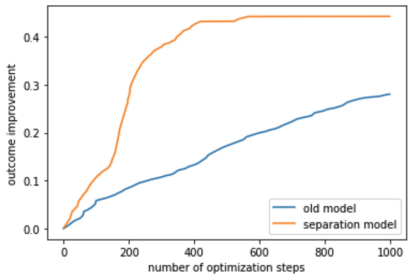

Figure 9. outcome improvement on Comparison dataset

with the separation model making slightly more effective modifications, converging to about 0.7 after 1,000 steps.

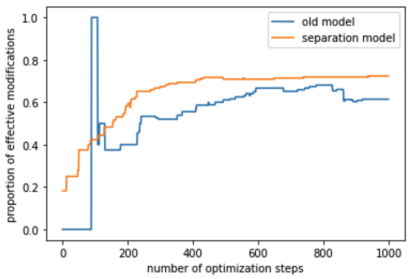

Figure 10. proportion of effective modifications

Fig.11 is the graph showing negative log likelihood, where we can see that the negative log likelihood remains basically unchanged after different number of optimization steps for both models.

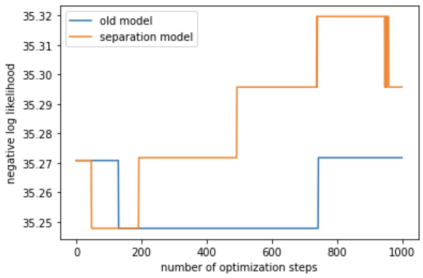

Figure 11. negative log likelihood

5. Conclusion

Based on experiments on three different datasets, we can conclude that the separation model and the old model make similar outcome improvement after 1,000 steps. The separation model achieves higher effective modification rate than the old model uniformly but sacrifices the naturalness of generated sequences to different extent. This implies that a balance between effective modification and naturalness of sequences might be achieved, for example by different sizes of outcome-related latent representations and sizes of outcome-unrelated latent representations.

References

Mueller, J., Gifford, D., and Jaakkola, T. Sequence to better sequence: continuous revision of combinatorial structures. In *International Conference on Machine Learning*, pp. 2536–2544. PMLR, 2017.