Incorporating Pseudo-Parallel Data for Quantifiable Sequence Editing

Yi Liao*[1], Lidong Bing[2], Piji Li[1], Shuming Shi[2], Wai Lam[1], Tong Zhang[2]
[1]The Chinese University of Hong Kong
[2]Tencent AI Lab
{yliao, pjli, wlam}@se.cuhk.edu.hk
{lyndonbing, shumingshi, bradymzhang}@tencent.com

arXiv:1804.07007v1 [cs.CL] 19 Apr 2018

Abstract

In the task of quantifiable sequence editing (QuaSE), a model needs to edit an input sentence to generate an output that satisfies a given outcome, which is a numerical value measuring a certain property of the output. For example, for review sentences, the outcome could be review ratings; for advertisement, the outcome could be click-through rate. We propose a framework which performs QuaSE by incorporating pseudo-parallel data. Our framework can capture the content similarity and the outcome differences by exploiting pseudo-parallel sentence pairs, which enables a better disentanglement of the latent factors that are relevant to the outcome and thus provides a solid basis to generate output satisfying the desired outcome. The dual reconstruction structure further enhances the capability of generating expected output by exploiting the coupling of latent factors of pseudo-parallel sentences. We prepare a dataset of Yelp review sentences with the ratings as outcome. Experimental results show that our framework can outperform state-of-the-art methods under both sentiment polarity accuracy and target value errors.

1 Introduction

Typical neural text generation is observed suffering from the problems of repetitions in word n-grams, producing monotonous language, and generating short common sentences (Li et al., 2017). To solve these problems, some researchers branch out into the way of post-editing (could be under some guidance, say sentiment polarity) a given message to generate text of better quality. For example, skeleton-based text generation first outlines a skeleton in the form of keywords or phrases, and then starts from the skeleton to generate text (Wang et al., 2017; Xiao et al., 2016). Another line of works conduct editing on an existing sentence and expect that the output sentence will serve particular purpose better (Guu et al., 2018). Similarly in conversation, some systems perform post-editing the retrieved passages to generate a new sentence as the response (Song et al., 2016). The third type is to perform editing on the input under the guidance of specific style. For example, Shen et al. (2017) take a sentence with negative sentiment as input, and edit it to transfer its sentiment polarity into positive.

In this paper, we generalize the third type of post-editing into a more general scenario, named Quantifiable Sequence Editing (**QuaSE**). Specifically, in the training stage, each input sentence is associated with a numeric outcome. For example, the outcome of review sentences is the rating, ranging from 1 to 5; the outcome of each advertisement is its click-through rate. In the test stage, given an input sentence and a specified outcome, a model's task is to edit the input to generate a new sentence that will satisfy the specified outcome with high probability. Meanwhile, the output sentence should keep the content of the input. For example, given the input sentence "The food is terrible", the desired output sentence could be "The food is OK" under the expected outcome "3.1" (a neutral sentiment), and "The food is delicious" under the expected outcome "4.0". If no expected outcome is given, the model could generate "The food is extremely delicious", by defaulting the best outcome, and "The food is extremely terrible", by defaulting the worst outcome.

Our problem setting is more general than previous works in two major aspects: (1) The outcome here is numerical, and it can be regarded as a generalization of the categorical outcome in (Shen et al., 2017; Hu et al., 2017). With such nu-

*The work was done when Yi Liao was an intern at Tencent AI Lab.

merical outcome, it is not possible to construct two corpora as counter part of each other as done in (Shen et al., 2017). (2) The editing operation is under a quantifiable guidance, i.e. the specified outcome or the defaulted extrema. For example, we can specify a particular rating value, such as 3.1 or 4.0, as the expected outcome. Although (Mueller et al., 2017) also takes outcome-associated sentences as training data, their model does not consider an expected outcome in the generation of revised sentences.

Considering the goal of editing an input sentence is to generate an output satisfying a specified outcome and keeping the content topic unchanged, QuaSE is challenging in a few aspects. Firstly, the model should be able to perceive the association between an outcome and the relevant wordings. For the previous example "The food is terrible", the model needs to figure out that the low rating is indicated by the word "terrible", instead of "food". Secondly, when performing editing, the model should retain the content, and only edits the outcome-related wordings. Moreover, the model needs to take a specified outcome into account and generates an output that has the specified outcome value with high probability. Continuing the running example, given the expected outcome 3.1, "The food is OK" is an appropriate output, but "The food is extremely delicious" and "The service is OK" are not. Thirdly, we do not have readily available data, such as data point [input sentence: "The food is terrible", expected outcome: 4.0, output sentence: "The food is delicious"] to show what the revised output should look like, that meet our need to train models.

In this paper, we propose a framework that performs QuaSE with dual reconstruction by incorporating pseudo-parallel data. The fundamental module of our framework is a Variational Autoencoder (VAE) (Kingma and Welling, 2013) to encode each individual input sentence into a latent content factor and a latent outcome factor, capturing the content and the outcome related wordings respectively. We propose to leverage pseudo-parallel sentence pairs (i.e, the sentences in a pair have the same or very similar content, but different outcome values) to enhance our model's capability of disentangling the two factors, which allows attributing the wording difference of two sentences in a pair to the outcome factor, and the wording similarity to the content factor. For sentence reconstruction, our framework employs a Recurrent Neural Network (RNN) based decoder (Sutskever et al., 2014) that takes as input the combination of a content factor and an outcome factor. Moreover, we introduce a dual reconstruction structure to further enhance the capability of generating expected output by exploiting the coupling of latent factors of pseudo-parallel sentences. Specifically, it attempts to reconstruct one sentence in a pair from the combination of its outcome factor and the other sentence's content factor, based on the intuition that the wording difference in a pair is outcome-related. In the test stage, taking a sentence and a specified outcome value as input, our model generates a revised sentence which likely satisfies the specified outcome target, and meanwhile the content is preserved as much as possible.

To evaluate the efficacy of our framework, we prepare a dataset of Yelp review sentences with the ratings as outcome, in the interval of [1, 5]. In the comparison with state-of-the-art methods, experimental results show that our framework can generate more accurate revisions to satisfy the target outcome and transfer the sentiment polarity. Ablation studies illustrate the effectiveness of the proposed components. We will release the prepared dataset and the code of our model to facilitate other researchers to do further research along this line.

2 Model Description

2.1 Problem Setting and Model Overview

In the task of Quantifiable Sequence Editing (QuaSE), the aim is to edit an input sentence X_0 under the guidance of an expected outcome value R^* to generate a new sentence X^* that will satisfy R^* with high probability. For training a model, we are given a set of sentence-outcome tuples (X,R), where X denotes a sentence and R denotes the numerical outcome associated with X.

Before disclosing the proposed model, we first introduce a concept of pseudo-parallel sentences used in our model. Let (x, x') denote a pair of pseudo-parallel sentences, which is automatically generated from the above tuples. x and x' describe the same or similar content, but their outcomes are different. Note that we use lowercase letters to denote variables related to sentences in pairs for better clarity of the subsequent description.

Our proposed model for training is depicted

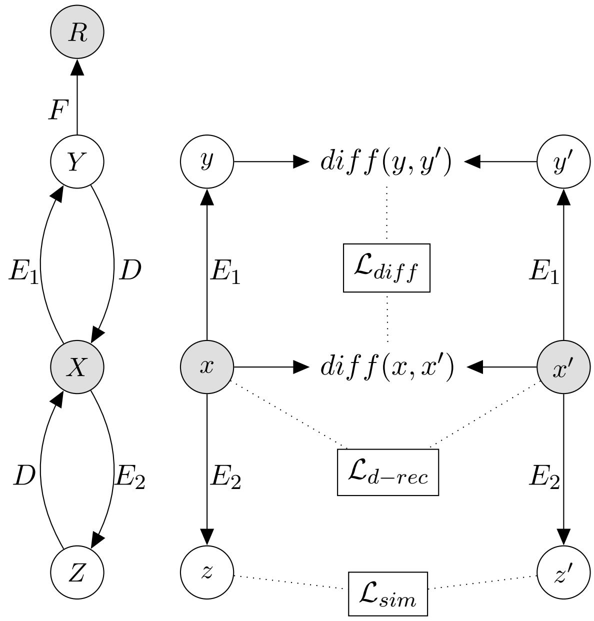

Figure 1: Model Overview. It is capable of incorporating individual sentences with the LHS and parallel sentence pairs with the RHS for joint training.

in Figure 1, with the notations given in Table 1. The left hand side models individual sentences. Specifically, it employs two encoders, i.e. E_1 and E_2, to encode a single sentence X into two latent factors Y and Z which capture the outcome and content properties respectively. An RNN-based decoder D takes the concatenation of Y and Z to reconstruct the input X. Moreover, a transformation function F predicts R with Y. The right hand side is designed to capture the differences between the two sentences in parallel sentence pairs. Specifically, the wording difference, i.e. $diff(x, x')$, is attributed to the outcome factor, i.e. $diff(y, y')$; the similar contents of two sentences should result in similar content factors, i.e. minimizing the loss $\mathcal{L}_{min}$; moreover, a dual reconstruction loss $\mathcal{L}_{d-rec}$ is minimized to enhance the capability of generating expected output. Overall, the model minimizes the losses from modeling single sentences and sentence pairs. After the model is properly trained, a separated component is applied for editing an input sentence to output a revision that satisfies a specified outcome target.

2.2 Modeling Single Sentence and its Rating

In probabilistic theory, we need to maximize the log-likelihood of observing the training sentence-outcome tuples (X,R), denoted as follows:

$$\begin{aligned} \log \int p(X, R) = \log \int p(X|Y, Z)p(Y, Z)dY dZ \\ + \log \int p(R|Y)p(Y)dY \end{aligned} \tag{1}$$

However, the integration in the first term on the right hand side is intractable. Inspired by the idea of VAE (Kingma and Welling, 2013), we alternatively maximize the Evidence Lower Bound (ELBO) (Blei et al., 2016) incorporating variational distributions, i.e. $q(Y|X)$ and $q(Z|X)$. Thus, this term is approximated as follows:

$$\begin{aligned} &\log \int p(X|Y, Z)p(Y, Z)dY dZ \geq -[\mathcal{L}_{rec} + \mathcal{L}_{kl}] \\ &\mathcal{L}_{rec} = -\mathbb{E}_{Y,Z\sim q(Y|X),q(Z|X)}[\log p(X|Y, Z)] \\ &\mathcal{L}_{kl} = KL[q(Y|X)|p(Y)] + KL[q(Z|X)|p(Z)] \end{aligned} \tag{2}$$

where, the term $\mathcal{L}_{rec}$ denotes the error of reconstructing X. As advocated by (Kingma and Welling, 2013) and (Bowman et al., 2016), the variational distributions $q(Y|X)$ and $q(Z|X)$ are modelled as Gaussian distributions, i.e. $q(Y|X) = \mathcal{G}(\mu_{Y|X}, \sigma_{Y|X})$, and $q(Z|X) = \mathcal{G}(\mu_{Z|X}, \sigma_{Z|X})$. The expectation $\mathbb{E}(\cdot)$ can be efficiently approximated using one Monte-Carlo sample, for example, $Y \sim q(Y|X)$ and $Z \sim q(Z|X)$. In practise, we can alternatively employ $Y = \mu_{Y|X}$ and $Z = \mu_{Z|X}$ instead of sampling since they are the means of the Gaussian distributions. We employ two encoders networks E_1 and E_2 to generate $\mu_{Y|X}$ and $\mu_{Z|X}$ respectively from the sequence X, i.e. $\mu_{Y|X} = E_1(X)$ and $\mu_{Z|X} = E_2(X)$. $p(X|Y, Z)$ is the probability of observing the sequence X given Y and Z, which is modelled by a decoder network D. Thus, the reconstruction loss can be rewritten as:

$$\mathcal{L}_{rec} = H(X, D(E_1(X), E_2(X))) \tag{3}$$

where $H(\cdot)$ is the cross entropy loss for the decoder output.

The term $\mathcal{L}_{kl}$ in Equation 2 denotes the KL-divergence (Kingma and Welling, 2013) between the variational posterior distribution and the prior distribution. Following previous works (Mueller et al., 2017), the priors $p(Y)$ and $p(Z)$ are defined as a zero-mean Gaussian distribution, i.e. $p(Y) = p(Z) = \mathcal{G}(\mathbf{0}, \mathbf{I})$. The loss $\mathcal{L}_{kl}$ serves as a regularization term enforcing that the variational posterior distribution resembles the prior distribution, which also avoids overfitting.

X	A sentence represented by a sequence of words
R	The outcome associated with the sentence X
Y	The outcome factor disentangled from the sentence X
Z	The content factor disentangled from the sentence X
(x, x')	A pair of pseudo-parallel sentences with similar contents but different outcomes
$diff(x, x')$	The explicit wording difference between x and x', defined in Equation 5
$diff(y, y')$	The difference between the outcome factors y and y', defined in Equation 5
E_1	The encoder that generates the outcome factor from the sentence X
E_2	The encoder that generates the content factor from the sentence X
D	The decoder that generates a sentence from an outcome factor Y and a content factor Z
F	The fully connected network that projects the outcome factor Y to the outcome R
U	The fully connected network that aligns $diff(x, x')$ with $diff(y, y')$
$\mathcal{L}_{rec}$	The reconstruction loss between a sentences and its reconstructed sentence
$\mathcal{L}_{kl}$	The KL-divergence between the variation posterior distribution and the prior
$\mathcal{L}_{mse}$	The prediction loss of F
$\mathcal{L}_{diff}$	The regression loss of U
$\mathcal{L}_{sim}$	The minimization loss between z and z'
$\mathcal{L}_{d-rec}$	The dual reconstruction loss between x and x'

Table 1: Notations in the model.

x	I will never come back to the restaurant.
x'	I will definitely come back to the restaurant, recommend!

Table 2: A pair of pseudo-parallel sentences.

The second term in Equation 1 models the log-likelihood of the outcomes. We adopt Taylor approximation for the calculation (as also used by (Mueller et al., 2017)), where this term is approximated by an affine transformation from the outcome factor Y to the outcome R, denoted as $F(Y)$. Then, we employ a loss function which measures the square error between R and $F(Y)$, denoted as:

$$\mathcal{L}_{mse} = (R - F(Y))^2 \tag{4}$$

Although Mueller et al. (2017) also model individual sentences and their outcomes, in their model, each sentence is only encoded into one latent factor to capture both outcome and content properties. In contrast, we disentangle a single sentence into two latent factors to model the outcome and the content separately to provide more flexibility. Moreover, such design facilitates the incorporation of the pseudo-parallel sentences, which will be described later.

2.3 Exploiting Pseudo-parallel Sentences.

As mentioned above, pseudo-parallel sentences are similar in terms of the content but different in terms of the outcome. For example, the sentences in Table 2 form a pair of pseudo-parallel review sentences because they both talk about "the restaurant", but with different sentiment ratings. Let y and y' denote their outcome factors, z and z' denote their content factors. We design three components to leverage pseudo-parallel sentence pairs to enhance our model's capabilities of disentangling the two types of factors and generating the desired output sentences.

2.3.1 Modeling Outcome Difference

We exploit the wording difference $diff(x, x')$ between x and x'. Note that the preparation (discussed later) determines that a pair of pseudo-parallel sentences are very likely to differ in the outcome factors, denoted as $diff(y, y')$. Thus, by aligning the surface wording difference of sentences in a pair and the difference in their outcome factors, we intend to improve the performance of the encoder E_1 for generating the outcome factor.

Specifically, $diff(x, x')$ and $diff(y, y')$ are defined as follows:

$$\begin{aligned} diff(x, x') &= inc(x, x') \oplus dec(x, x') \\ diff(y, y') &= y - y' = E_1(x) - E_1(x') \end{aligned} \tag{5}$$

where $inc(x, x')$ and $dec(x, x')$ are embeddings capturing the wording difference between x and x'. $inc(x, x')$ denotes the "increment" from x to x', i.e. the terms that appear in x' but not in x. $dec(x, x')$ denotes the "decrement". If there are multiple terms in the difference, we sample one term for inc and dec respectively. In the example shown in Table 2, $inc(x, x')$ is the embedding of "never", and $dec(x, x')$ could be the embedding of "definitely" or "recommend". The effect of outliers during sampling anneals since the training data contain sufficient pairs of sentences. The symbol $\oplus$ denotes concatenation. $diff(y, y')$ is defined as the subtraction between the outcome factors of the sentences.

We employ a regression network U to align $diff(x, x')$ and $diff(y, y')$, and the loss $\mathcal{L}_{diff}$ is computed as:

$$\mathcal{L}_{diff} = ||diff(y, y') - U[diff(x, x')]||^2 \quad (6)$$

The regression network is designed as a multi-layer perceptron network taking $diff(x, x')$ as the input.

2.3.2 Modeling Content Similarity

Besides the outcome difference, another property of two pseudo-parallel sentences is that they share similar content. To capture it, we design a loss function minimizing the square error between the content factors.

$$\mathcal{L}_{sim} = ||z - z'||^2 = ||E_2(x) - E_2(x')||^2 \quad (7)$$

Minimizing $\mathcal{L}_{sim}$ helps enforce that the encoder E_2 generates the content factor more accurately.

2.3.3 Dual Reconstruction

The decoder D is not only used in Section 2.2 to reconstruct a single training sentence, but also employed for generating output sentences in the test stage (described in Section 3). To improve the robustness of D, we propose a dual reconstruction component based on the pseudo-parallel sentences. Different from reconstructing an original sentence in Section 2.2, in the dual reconstruction given a sentence x, we reconstruct its dual sentence x'.

Specifically, we first encode x and x' into their outcome factors y/y' and content factors z/z'. Since x shares similar content with x', its content factor z, when combined with the outcome factor y' of x', should nearly reconstruct x'. The loss function can be written as:

$$\begin{aligned} \mathcal{L}^{d-rec}_{x';x} &= H(x', D(E_1(x'), E_2(x))) \\ &= H(x', D(y', z)) \end{aligned} \quad (8)$$

The same dual reconstruction process applies to the counterpart of x', i.e. x. Thus, the total dual reconstruction loss function is as follows:

$$\mathcal{L}_{d-rec} = \mathcal{L}^{d-rec}_{x';x} + \mathcal{L}^{d-rec}_{x;x'} \quad (9)$$

Note that the encoders E_1/E_2 and the decoder D here refer to exactly the same neural networks (i.e., the parameters are shared) as used in Section 2.2.

2.4 Specific Design of the Neural Networks

As mentioned in Table 1, five neural networks, namely E_1, E_2, D, F, and U, are involved in our proposed model. Their specific implementations are as follow.

E_1/E_2 : RNNs of GRU cells with a fully connected neural network appended to the last state to add some noise, which is a reparameterization alternative for sampling. Their outputs are the outcome and content factors, respectively.

D: An RNN of GRU cells. The RNN takes the concatenation of an outcome factor and a content factor as the initial state for decoding.

F: A fully connected network. It takes an outcome factor as input and outputs an outcome value.

U: A fully connected network. It takes $diff(x, x')$ as input to predict $diff(y, y')$.

2.5 Joint Training

Considering all the aforementioned components, we define a joint loss function as:

$$\begin{aligned} \mathcal{L}_{joint} =& \lambda_{rec}\mathcal{L}_{rec} + \lambda_{kl}\mathcal{L}_{kl} + \lambda_{mse}\mathcal{L}_{mse} + \\ & \lambda_{diff}\mathcal{L}_{diff} + \lambda_{sim}\mathcal{L}_{sim} + \lambda_{d-rec}\mathcal{L}_{d-rec} \end{aligned} \quad (10)$$

Each component in $\mathcal{L}_{joint}$ is associated with a weight. Following the sigmoid annealing schedule (Bowman et al., 2016; Mueller et al., 2017), we design the following steps to tune the weights.

1. Tune the weights λ_{rec} and λ_{mse} on the validation dataset under the metric MAE (refer to Section 4.2), while fixing the other weights to zeros. We set $\lambda_{rec} + \lambda_{mse} = 1$.

2. Fixed the weights tuned in the first step. For each remaining loss, gradually increase the weight from 0 to 1 during training, until $\mathcal{L}_{rec}$ or $\mathcal{L}_{mse}$ becomes worse.

The above strategy prioritizes the reconstruction loss λ_{rec} and the outcome prediction loss λ_{mse}, since they are core components for generating the revised sentences.

3 Generating Revisions

In the test stage, the trained model edits an input sentence X_0 and outputs a revised sentence X^* that is likely to satisfy the specified outcome target R^*, and meanwhile preserve the content as much as possible.

We first encode X_0 with E_1 and E_2 to get Y_0 and Z_0 respectively. The next step is to modify

Y_0 to get a new outcome factor Y^* that is likely to generate the target outcome R^*. We intend to find a suitable Y^* near Y_0, where the distance captures the degree of revisions. The process to determine a suitable Y^* is as follows. We first assume Y follows the Gaussian distribution $Y \sim \mathcal{G}(Y_0 = E_1(X_0), \sigma)$, the mean of which is Y_0. Then a threshold τ is set for the Gaussian distribution. The set $\mathcal{C} = \{Y : \mathcal{G}(Y|E_1(X_0), \sigma) > \tau\}$ is chosen as the feasible range for Y^*, which will expand if τ is set smaller, and thus allowing more possible revisions. Then Y^* is determined as follows:

$$Y^* = \arg\min_{Y \in \mathcal{C}} (F(Y) - R^*)^2 \tag{11}$$

Note that in (Mueller et al., 2017), Y^* is determined as $\arg\max_{Y \in \mathcal{C}} F(Y)$, which does not take the outcome target into account. Then the revised sentence X^* is generated using X_0 and Y^* via the decoder D:

$$X^* = D(Y^*, Z_0) \tag{12}$$

Thus, the content of X_0 is preserved with Z_0, and the expected outcome is achieved with Y^*.

4 Experiments

4.1 Dataset Preparation

Our dataset contains sentences extracted from the original Yelp review, where each review is associated with a rating in $\{1, 2, 3, 4, 5\}$. [1] Specifically, we employ the sentences with sentiment polarity (i.e. positive or negative) used in (Shen et al., 2017) as the primary portion of our data. After some cleaning, we obtain about 530K sentences. To add neutral sentences, we randomly select 70K sentences from those original reviews with neutral sentiment (i.e. rating 3). In total, our dataset contains 599K sentences, and we randomly hold 50K as test data, 10K as validation data, and the remaining is used for training.

For training, we need each input sentence having a rating value, and for test, we need to measure the rating of a generated sentence for checking if the generated sentence satisfies the specified outcome. Therefore, an automatic method is needed for measuring the rating values of the training sentences and the generated sentences. We employ the sentiment analyzer in Stanford CoreNLP (Manning et al., 2014) to do so. Specifically, we first invoke CoreNLP to output a probability of each rating in $\{1, 2, 3, 4, 5\}$ for a sentence, then we take the sum of the probability-multiplied ratings as the sentence rating. Hereafter, we use rating and outcome interchangeably. [2] Some statistics of the data is given in Table 3. The vocabulary size of the dataset is 9,625.

Rating interval	$[1, 2)$	$[2, 3)$	$[3, 4)$	$[4, 5]$
Sentence#	34273	231740	165159	167803

Table 3: Numbers of sentences in each rating interval.

For preparing the pseudo-parallel sentence pairs, we first follow the ideas in (Guu et al., 2018) to generate some initial pairs. Specifically, we first calculate Jaccard Index (JI) for each pair of sentences, and keep those with JI value no less then 0.5 as initial pairs. Note that such initial pairs could contain many false positives (roughly 10% as manually evaluated on the Yelp corpus in (Guu et al., 2018)) because the calculation of JI does not distinguish content words and outcome words. To solve this problem, we add another constraint: for an initial pair to be regarded as a pseudo-parallel sentence pair, the difference between the rating scores of its two sentences should be no less than 2. Here, the idea is that given the two sentences are similar enough in word level (i.e. JI $\geq$ 0.5), if their rating scores are dissimilar enough, it looks plausible to conjecture that their wording difference is outcome-related and causes the rating difference. In fact, such wording difference is exactly what we want to capture with pseudo-parallel sentence pairs. In total, we obtain about 604K sentence pairs from the single training sentences. For conducting the joint training with both single sentences and pseudo-parallel pairs, we make each data point composed of a single sentence and a sentence pair. To do so, we couple each sentence pair with one single sentence, thus we can use all pairs for training. Note that because we have more sentence pairs, some single sentences are used twice randomly.

[1] https://www.yelp.com/dataset/challenge

[2] One may think that would it be possible to use the original rating given by Yelp users as outcome for training? We did not use it for two reasons: (1) We want the ratings of training sentences and generated sentences are measured with a consistent method; (2) In fact, we find that the predicted rating with CoreNLP has a Pearson correlation of 0.85 with the rating given by users. Note that the original Yelp data only has ratings for entire reviews. We derived the sentence ratings by users like this: a sentence takes as its rating the average of the ratings of those reviews where it appears in. Human evaluation in (Shen et al., 2017) shows that a similarly deriving method for polarity is basically reasonable as well.

4.2 Evaluation Metrics

Considering that our model's task is to revise a sentence such that its outcome (predicted by Stanford CoreNLP) becomes close to a specified target, the evaluation metric should thus measure whether the outcome of the revised sentence is close enough to the target. We define the metric as the mean absolute error (MAE) between the specified target outcome and the outcomes of revised sentences.

$$MAE = \frac{1}{|S|} \sum_{X_i \in S} |R_i - R^*| \tag{13}$$

where S is the set of revised sentences, R^* is the target outcome, and R_i is the outcome of a revised sentence X_i.

4.3 Comparisons and Parameter Settings

Our proposed model is compared with two state-of-the-art models, which have their implementations released.

Sequence to Better Sequence (S2BS) (Mueller et al., 2017) [3]: For training, S2BS also requires each sentence is associated with an outcome. For testing, S2BS revises a sentence such that the revision is associated with a higher outcome. We employ our revision method to replace theirs for conducting comparison, because: (1) S2BS does not take a specified outcome while revising; (2) While revising, S2BS only generates "better sequence", i.e. having higher outcomes. Note that the training of S2BS is not specifically designed for only generating better revisions, thus, it is still a fair comparison. We tune the parameters for S2BS by following the suggestions in their source code.

Text Style Transfer (TST) (Shen et al., 2017)[4]: In TST, the sentiment of each sentence is labelled as negative or positive. The model is able to revise a negative sentence into positive, or vice versa. Their task can be treated as a special case of our QuaSE task: we set the outcome target to 1 for the input sentences that are associated with outcomes larger than the neutral outcome 3, thus, our task equals to revising a positive sentence into negative. We follow the suggested parameter settings reported in (Shen et al., 2017), since the corpus and the task are almost the same for this comparative experiment.

After tuning on the validation set, the weight parameters are as follows: $\lambda_{rec} = 0.75$, $\lambda_{kl} = 0.6$, $\lambda_{mse} = 0.25$, $\lambda_{diff} = 0.2$, $\lambda_{sim} = 0.2$, $\lambda_{d-rec} = 0.1$, and the dimensions of the outcome and content factors are 50 and 50 respectively. The parameter τ for revision takes $\exp(-100000)$ for both our model and S2BS, as defaulted in (Mueller et al., 2017).

4.4 Quantitative Study

We compare our model with S2BS by specifying five target ratings, namely 1, 2, 3, 4, and 5. The result is shown in Table 4. Each model is trained for three times and the average testing result is reported as the performance. Note that "Original" refers to the MAE between the targets and the ratings of input sentences. We can observe that the MAE values of both our model and S2BS are smaller than Original. It demonstrates that both models are able to revise the sentences towards the outcome targets. Furthermore, by comparing our model with S2BS, we observe that our model achieves smaller MAE than S2BS. It shows that our model does better in revising the sentences to satisfy the given targets. One major reason is that we design three components to leverage pseudo-parallel sentences. By modeling the wordings difference, the model captures the keywords that mainly causes the difference in the outcome. By enforcing the content factors of pseudo-parallel sentences to be similar, the model is capable to generate the content factor more precisely. Moreover, the dual reconstructions can guide the revision procedure with the hints from the parallel sentences. The MAE for T=5 is smaller than that for T=1. This is partially due to the fact that the outcomes of the test sentences are closer to 5, refer to Table 3. We also report the average Edit Distance between the input sentences and the revised sentences to measure the degree of revisions. For T=1 and T=5, our model conducts more revisions than S2BS, which brings in better MAE. For T={2, 3, 4}, our model generates more accurate sentences (i.e. better MAE) with less revisions.

We also compare our model with TST for sentiment polarity transfer. For fair comparison, we employ the same evaluation metric as used in (Shen et al., 2017): the sentiment accuracy of the transferred sentences. We define the revised sentences with ratings larger/smaller than 3 as positive/negative. The results are given in Table 5,

[3] https://github.com/shentianxiao/language-style-transfer
[4] https://bitbucket.org/jwmueller/sequence-to-better-sequence/

Model	MAE					Edit Distance				
	T=1	T=2	T=3	T=4	T=5	T=1	T=2	T=3	T=4	T=5
Original	2.2182	1.2379	0.8259	0.9279	1.7818	N/A	N/A	N/A	N/A	N/A
S2BS	1.6839	0.9444	0.7567	0.7572	1.3024	6.6439	5.342	4.9390	5.005	6.2290
Our Model	1.4162	0.6298	0.7408	0.5377	0.9408	7.9191	4.7	3.4505	4.13	8.0094

Table 4: MAE and Edit Distance for our proposed model and S2BS. T refers to the target outcome.

Model	Negative to Positive	Positive to Negative
TST	0.7280	0.7097
Our Model	0.8836	0.7862

Table 5: Accuracy Comparison with TST.

where two accuracy values are reported: negative to positive, and positive to negative. The results show that our model achieves better accuracy than TST in both transfer directions. For TST, the sentences with ratings close to neutral (i.e. 3.0) may cause some difficulty. For example, two sentences with ratings 3.2 and 2.8 are usually similar in sentiment related wordings. However, TST treats them as positive and negative respectively. In contrast, our method models the associations between each sentence and its outcome, and thus captures the sentiment wordings better. Our model is far better for transferring negative sentences into positive, moreover, both models achieve better performance for this transfer direction. We can probably attribute the reason to the imbalanced training data: 55% positive sentences v.s. 45% negative sentences.

We also calculate the log probability of the generated sentences, using a pre-trained RNNLM on Yelp data (Mikolov et al., 2010) [5], to measure the language model quality of the sentences. The results are shown in Table 6. Our model performs not as good as TST and S2BS. Generally speaking, it is because of the trade-off between the MAE/Accuracy performance and the sentence quality. Specifically, our model adds three components to model the pseudo-parallel sentence pairs for capturing the hints of outcome-related wordings, and thus to better satisfy the outcome targets in generating outputs, which may somehow sacrifice the language model quality. TST and S2BS basically employ a conventional VAE to model single sentences, and thus can better maintain the language quality.

[5] http://www.fit.vutbr.cz/~imikolov/rnnlm/

Model	Original	S2BS	TST	Our Model
Log prob.	-17.5	-19.6	-18.9	-22.2

Table 6: Language model comparison.

4.5 Qualitative Study

We show some examples produced by our model in Table 7. For each input sentence, we specify three outcome targets: 1, 3, and 5. The sentence ratings, produced by CoreNLP, are given in the last column.

For the first example "our first time and we had a great meal, wonderful service .", since it is already very positive, our model keeps its original form for T=5. For T=1, our model can change the positive words "great" and "wonderful" into negative ones very precisely. For the second example "horrible food", the revised sentences are generally correct for the given targets. One may notice that the predicted ratings by CoreNLP are not very accurate for some sentences. E.g. "horrendous" is obviously a negative word, and "amazing delicious food ! recommend !" should have a better rating than "their food amazing !". Our model also produces some revisions that are not very reasonable, e.g. "we have service" of the first example with T=3, "for no" of the third example with T=1.

4.6 Ablation Study

Recall that our model is a combination of a revised VAE, which disentangles a sentence into two factors to enable the subsequent design, and a coupling component modeling pseudo-parallel sentence pairs. For the three losses of the coupling component, we show their effects under the MAE metric in Table 8. "None" refers to that all three losses are removed, and it is basically worse than S2BS, which implies that only using the revised VAE does not work well. As more losses are used, the performance is gradually improved. Moreover, the dual reconstruction is more effective than the other two.

In the weight tuning, the first step only tunes the weights of $\mathcal{L}_{rec}$ and $\mathcal{L}_{mse}$. We observe that solely minimizing $\mathcal{L}_{rec}$ and $\mathcal{L}_{mse}$ also decreases

	Sentence	Rating
INPUT 1	our first time and we had a great meal , wonderful service .	4.5213
T=1	our first time and we had a terrible meal , stale service .	1.8157
T=3	our first time and we had a great meal , we have service .	4.0742
T=5	our first time and we had a great meal , wonderful service .	4.5213
INPUT 2	horrible food !	2.1045
T=1	horrendous	3.0378
T=3	their food amazing !	4.7902
T=5	amazing delicious food ! recommend !	4.7290
INPUT 3	decent food and wine selection , but nothing i will rush back for .	2.3131
T=1	decent food and wine selection , but nothing i will rush for no .	2.3943
T=3	decent food and wine selection , but i will never look back for .	2.9849
T=5	decent food and wine selection , but excellent service ... and will return !	4.7289
INPUT 4	this tire center is amazing .	4.3912
T=1	this tire center is horrible .	1.9071
T=3	this tire center is really good .	4.3331
T=5	this tire center is amazing .	4.3912
INPUT 5	food is very addiction tasty !	4.2859
T=1	food is just horrible here ?	2.0116
T=3	food is just addiction here !	3.1658
T=5	food is very yummy addiction !	3.2279

Table 7: Case study.

$\mathcal{L}_{sim}$, because in this process the encoder E_2 becomes more capable for disentangling the content factor, and thus z and z' could become similar as they come from similar inputs, i.e. pseudo-parallel sentences.

Another observation is that solely minimizing $\mathcal{L}_{rec}$ and $\mathcal{L}_{mse}$ increases $\mathcal{L}_{d-rec}$ after some training epochs. To analyze the reason, we assume there is a single sentence x in the training set. Then the loss of reconstructing x from y and z is included in $\mathcal{L}_{rec}$. Assume x also acts as a sentence in a pseudo-parallel pair, and thus the loss of reconstructing x from y and z' is included in $\mathcal{L}_{d-rec}$. The only difference between them lies in the content factors z and z'. Given that z and z' are not enforced to resemble each other when $\mathcal{L}_{sim}$ is excluded from this tuning step, $\mathcal{L}_{rec}$ and $\mathcal{L}_{d-rec}$ cannot be minimized simultaneously. Moreover, when we minimize $\mathcal{L}_{sim}$ in the second step with weights of $\mathcal{L}_{rec}$ and $\mathcal{L}_{mse}$ fixed, the loss $\mathcal{L}_{d-rec}$ also decreases, which complies with the above analysis.

Model	T=1	T=3	T=5
S2BS	1.6839	0.7567	1.3024
None	1.6639	0.7684	1.5434
$\mathcal{L}_{sim}$	1.6090	0.8258	1.5233
$\mathcal{L}_{diff}$	1.6793	0.8017	1.3140
$\mathcal{L}_{d-rec}$	1.5191	0.7784	1.1218
$\mathcal{L}_{sim}, \mathcal{L}_{diff}$	1.4991	0.8218	1.3705
$\mathcal{L}_{sim}, \mathcal{L}_{d-rec}$	1.4101	0.8027	1.1246
$\mathcal{L}_{diff}, \mathcal{L}_{d-rec}$	1.3879	0.7786	1.1413
ALL	1.4162	0.7408	0.9408

Table 8: Ablation study.

5 Related Work

5.1 Variational Autoencoder

Variational Autoencoder (VAE) (Kingma and Welling, 2013) has been shown as an effective unsupervised model to recover the latent representation of data. Compared with deterministic autoencoder for sequential data (Sutskever et al., 2014; Cho et al., 2014), the VAE is able to capture more coherent latent space, which is attributed to the prior of the latent variable. Due to the power of the VAE, many models have adopted VAE to solve various task. For example, Variational RNN (VRNN) (Chung et al., 2015) incorporates a high-level latent random variable into standard RNN to model highly structured sequential data such as natural speech. Variational Neural Machine Translation (VNMT) (Zhang et al., 2016) introduces a continuous latent variable to explicitly model underlying semantics of source sentences and to guide the generation of target translations. Miao et al. (2016) introduce a generic variational inference framework for generative and conditional models of text.

5.2 Text Revising and Text Style Transfer

Our investigated task is more general than text style transfer (Shen et al., 2017; Fu et al., 2018; Hu et al., 2017; Jhamtani et al., 2017; Melnyk et al., 2017; Zhang et al., 2018), which aims at transferring text from the original style into a target style. For example, transferring negative reviews into positive reviews; transferring modern text into Shakespeare style text. The task of text style transfer is similar with image style transfer (Gatys et al., 2016; Zhu et al., 2017; Liu and Tuzel, 2016; Kim et al., 2017; Liu et al., 2017; Yi et al., 2017). It is normally assumed in a text style transfer task that one corpus for each style is available. The texts of the two corpora do not form parallel sentence pairs, which is different from the parallel text that is widely adopted in sequence to sequence models (Sutskever et al., 2014). The only association between the non-parallel texts is that they share similar content. In contrast, our proposed task QuaSE assumes each sentence is associated with an outcome pertaining to continues values. Style transfer task can be treated as a special case of QuaSE, where their outcomes are binary. Mueller et al. (2017) propose a model to revise a sentence with lower outcome to a sentence with higher outcome. However, in the task of QuaSE, a sentence is revised to any specified outcome, which can be either higher or lower than the original outcome. Moreover, we employ pseudo-parallel sentence pairs in our corpus. By aligning the latent factors associated with the sentence pairs, our model captures the representations of content and outcome of the sentences more accurately. Similarly, Guu et al. (2018) leverage pseudo-parallel sentence pairs for generating sentences from prototypes. However, we treat the pseudo-parallel sentences differently. They assume that one sentence is the prototype of its paired sentence, while we investigate the detailed associations between pseudo-parallel sentences.

6 Conclusion

In this paper, we proposed a task of Quantifiable Sequence Editing (QuaSE), where a model needs to edit an input sentence towards the direction of a numerical outcome target. To tackle this task, we proposed a framework which can simultaneously exploit the single sentences and pseudo-parallel sentence pairs for training. For evaluation, we prepared a dataset with Yelp sentences and their ratings predicted by Stanford CoreNLP. Experimental results show that our framework can dominate the state-of-the-art methods under the measures of sentiment polarity accuracy and target value errors. Case studies show that our framework can indeed generate some interesting sentences, and ablations show that the proposed components work well as expected.

References

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. 2016. Variational inference: A review for statisticians. *arXiv*, abs/1601.00670.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. 2015. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems*, pages 2980–2988.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proceedings of 2016 Conference on Computer Vision and Pattern Recognition*, pages 2414–2423.

K. Guu, T. B. Hashimoto, Y. Oren, and P. Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics (TACL)*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1587–1596.

Harsh Jhamtani, Varun Gangal, Eduard H. Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence-to-sequence models. *arXiv*, abs/1707.01161.

Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. 2017. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1857–1865.

Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv*, abs/1312.6114.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169.

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. 2017. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708.

Ming-Yu Liu and Oncel Tuzel. 2016. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 469–477.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Igor Melnyk, Cícero Nogueira dos Santos, Kahini Wadhawan, Inkit Padhi, and Abhishek Kumar. 2017. Improved neural text attribute transfer with non-parallel data. *arXiv*, abs/1711.09395.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *Proceedings of the 33nd International Conference on Machine Learning*, pages 1727–1736.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of 11th Annual Conference of the International Speech Communication Association*, pages 1045–1048.

Jonas Mueller, David Gifford, and Tommi Jaakkola. 2017. Sequence to better sequence: Continuous revision of combinatorial structures. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2536–2544.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, pages 6833–6844.

Yiping Song, Rui Yan, Xiang Li, Dongyan Zhao, and Ming Zhang. 2016. Two are better than one: An ensemble of retrieval- and generation-based dialog systems. *arXiv*, abs/1610.07149.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Yufei Wang, Zhe Lin, Xiaohui Shen, Scott Cohen, and Garrison W. Cottrell. 2017. Skeleton key: Image captioning by skeleton-attribute decomposition. In *Proceedings of 2017 Conference on Computer Vision and Pattern Recognition*, pages 7378–7387.

Tong Xiao, Jingbo Zhu, Chunliang Zhang, and Tongran Liu. 2016. Syntactic skeleton-based translation. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 2856–2862.

Zili Yi, Hao (Richard) Zhang, Ping Tan, and Minglun Gong. 2017. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of 2017 International Conference on Computer Vision*, pages 2868–2876.

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. 2016. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 521–530.

Ye Zhang, Nan Ding, and Radu Soricut. 2018. Shaped: Shared-private encoder-decoder for text style adaptation. In *The 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of 2017 International Conference on Computer Vision*, pages 2242–2251.