QuaSE: Sequence Editing under Quantifiable Guidance*

Yi Liao[13], **Lidong Bing**[2], **Piji Li**[12], **Shuming Shi**[2], **Wai Lam**[1], **Tong Zhang**[2]
[1]The Chinese University of Hong Kong
[2]Tencent AI Lab
[3]Noah's Ark Lab, Huawei Technologies
{yliao, pjli, wlam}@se.cuhk.edu.hk
{lyndonbing, shumingshi, bradymzhang}@tencent.com

Abstract

We propose the task of Quantifiable Sequence Editing (QuaSE): editing an input sequence to generate an output sequence that satisfies a given numerical outcome value measuring a certain property of the sequence, with the requirement of keeping the main content of the input sequence. For example, an input sequence could be a word sequence, such as review sentence and advertisement text. For a review sentence, the outcome could be the review rating; for an advertisement, the outcome could be the click-through rate. The major challenge in performing QuaSE is how to perceive the outcome-related wordings, and only edit them to change the outcome. In this paper, the proposed framework contains two latent factors, namely, outcome factor and content factor, disentangled from the input sentence to allow convenient editing to change the outcome and keep the content. Our framework explores the pseudo-parallel sentences by modeling their content similarity and outcome differences to enable a better disentanglement of the latent factors, which allows generating an output to better satisfy the desired outcome and keep the content. The dual reconstruction structure further enhances the capability of generating expected output by exploiting the couplings of latent factors of pseudo-parallel sentences. For evaluation, we prepared a dataset of Yelp review sentences with the ratings as outcome. Extensive experimental results are reported and discussed to elaborate the peculiarities of our framework. [1]

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 phrases/words, 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 will serve particular purposes better (Guu et al., 2018). Similarly in conversation, some systems post-edit the retrieval results to generate new sentences 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 **Qua**ntifiable **S**equence **E**diting (**QuaSE**). Specifically, in the training stage, each input sentence is associated with a numeric outcome. For example, the outcome of a review sentence is its 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 target, a model needs to edit the input to generate a new sentence that will satisfy the outcome target with high probability. Meanwhile, the output sentence should keep the content described by the input. For example, given the input sentence "The food is terrible", a desired output sentence could be "The food is OK" under the expected outcome

* The work described in this paper was done when Yi Liao was an intern at Tencent AI Lab. The work is partially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: CUHK413510)

[1]Our code and data are available at https://bitbucket.org/leoeaton/quase/src/master/

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3855–3864
Brussels, Belgium, October 31 - November 4, 2018.

"3.1" (a neutral sentiment), and "The food is delicious" under the expected outcome "4.0". If no outcome target is given, the model could generate "The food is extremely delicious", by defaulting the best outcome, or "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; Gao et al., 2018). With such numerical outcome, it is impossible to construct two corpora as counterpart of each other as done in (Shen et al., 2017; Gao et al., 2018). (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 target rating, such as 3.1 or 4.0, as the expected outcome. Although Mueller et al. (2017) also take outcome-associated sentences for training, their model does not perform such outcome-guided editing for sentence generation.

Considering that the goal of the task is to generate an output that satisfies a specified outcome and keeps the content unchanged, QuaSE is challenging in a few aspects. Firstly, a model should be able to perceive the association between an outcome and its 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 keep the content, and only edits the outcome-related wordings. Moreover, the model needs to take a specified outcome into account and generate an output that satisfies 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 points like [input sentence: "The food is terrible", expected outcome: 4.0, output sentence: "The food is delicious"] to show the model what the revised output should look like, that meet our need to train models.

We propose a framework to address this task. The fundamental module of our framework is a Variational Autoencoder (VAE) (Kingma and Welling, 2013) to encode each 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 two 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 the sentences in a pair to the outcome factor, and the wording similarity to the content factor. For sentence reconstruction, we employ 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. To further enhance the capability of generating expected output, we introduce a dual reconstruction structure which exploits the couplings 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 target as input, our model generates a revised sentence which likely satisfies the specified target, and meanwhile the content is preserved as much as possible.

To evaluate the efficacy of our framework, we prepared a dataset of Yelp review sentences with the ratings as outcome. Compared with state-of-the-art methods handling similar tasks, experimental results show that our framework can generate more accurate revisions to satisfy the target outcome and transfer the sentiment polarity, meanwhile it keeps the original content better. Ablation studies illustrate the effectiveness of the designed components for enhancing the performance. We have released the prepared dataset and the code of our model to facilitate other researchers to do further research along this line, refer to Footnote 1.

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).

Our proposed model for training is depicted in Figure 1. The left hand side models individual sentences. Specifically, it employs two encoders,

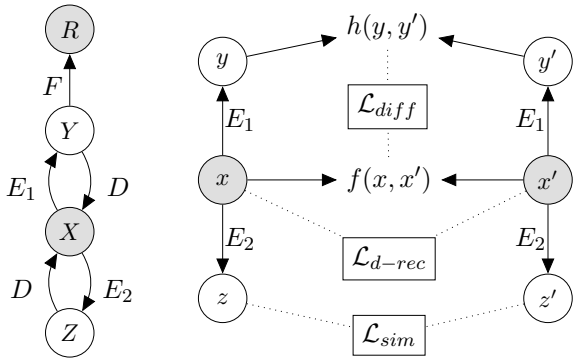

Figure 1: Model Overview.

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. In contrast, Mueller et al. (2017) employ a single factor for capturing these two properties, which limits the capability of distinguishing one property from the other. As a consequence, when editing a sentence towards a given outcome, the sentence content is likely to be changed, which should be suppressed as much as possible. 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 models pseudo-parallel sentence pairs (automatically generated from the above tuples), so we first introduce the concept of pseudo-parallel sentences as follows. Let (x, x') denote a pair of pseudo-parallel sentences, x and x' should describe the same or similar content, but their outcomes are different. Note that we use lowercase letters to denote variables related to sentence pairs for better clarity. For two sentences in a pair, the difference of their outcome factors $h(y, y')$ is attributed to their wording difference $f(x, x')$, resulting in the loss $\mathcal{L}_{diff}$; the similar contents of two sentences should result in similar content factors, i.e. minimizing the loss $\mathcal{L}_{sim}$; 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 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 Sentences

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

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

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:

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

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 encoder networks E_1 and E_2 to generate $\mu_{Y|X}$ and $\mu_{Z|X}$ respectively from the sentence 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 sentence 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 is the cross entropy loss for the decoder.

The term $\mathcal{L}_{kl}$ in Equation 2 denotes the KL-divergence 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.

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

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

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 two latent factors from a single sentence to model the outcome and the content separately to provide more flexibility. Moreover, such design allows the incorporation of the pseudo-parallel sentences, which will be described in the next subsection.

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. E.g., Table 1 shows a pair of pseudo-parallel sentences, where both talk about "the restaurant", but with different sentiments (i.e. ratings). For the pair (x, x'), let y and y' denote their outcome factors, z and z' denote their content factors. We design three components to leverage pseudo-parallel sentences to enhance our model's capabilities of disentangling the two types of factors and generating the desired output sentences.

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

Table 1: A pair of pseudo-parallel sentences.

2.3.1 Modeling Outcome Difference

We exploit the wording difference $f(x, x')$ between x and x'. Note that the preparation (discussed in Section 4.1) determines that a pair of pseudo-parallel sentences are very likely to differ in the outcome factors, denoted as $h(y, y')$. Thus, by aligning the surface wording difference of two 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. $f(x, x')$ and $h(y, y')$ are defined as follows:

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

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 or dec. For the example in Table 1, $dec(x, x')$ is the embedding of "never", and $inc(x, x')$ 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. $h(y, y')$ is defined as the subtraction between the outcome factors.

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

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

2.3.2 Modeling Content Similarity

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 the encoder E_2 generate 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 (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'. For such dual reconstruction, the loss is 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 whole dual reconstruction loss 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 networks (i.e., the parameters are shared) as used in Section 2.2.

The specific design of the networks are as follows. E_1/E_2: RNNs of GRUs 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 $f(x, x')$ as input to predict $h(y, y')$.

2.4 Joint Training

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

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

in which each component is associated with a weight. Following the sigmoid annealing schedule (Bowman et al., 2016), we design the following strategy to tune the weights: (1) Tune the weights λ_{rec} and λ_{mse} on the validation dataset under the metric MAE (refer to Section 4.3), 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 the training, until the reconstruction loss $\mathcal{L}_{rec}$ or the outcome prediction loss $\mathcal{L}_{mse}$ becomes worse. The strategy prioritizes $\mathcal{L}_{rec}$ and $\mathcal{L}_{mse}$, since they are the core components for generating the revised sentences.

3 Editing under Quantifiable Guidance

In the test, the trained model edits an input sentence X_0 and outputs a revision X^* that is likely to satisfy the specified outcome target R^*, and meanwhile preserves 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^*. 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 we choose $\mathcal{C} = \{Y : \mathcal{G}(Y|E_1(X_0), \sigma) > \tau\}$ as the feasible range for Y^*, where τ is a threshold. $\mathcal{C}$ will expand if τ is set smaller, and thus allowing more revisions. Finally, 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 consider an outcome target. The revised sentence X^* is generated from 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 Yelp reviews [2], where each review is associated with a rating in {1, 2, 3, 4, 5}. 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 520K sentences. To add neutral sentences, we randomly select 80K sentences from the original reviews with neutral sentiment (i.e. rating 3). To make sure that the neural sentences added by us are describing the same domain, we only pick neural sentences whose tokens are all in the vocabulary of the primary data. The vocabulary size of the dataset is 9,625. In total, our dataset contains 599K sentences, and we randomly hold 50K for test, 10K for validation, and the remaining for training.

For training, we need each input sentence being associated with a rating value, and for test, we need to measure the rating of a generated sentence to check if the generated sentence satisfies the specified outcome target. Therefore, an automatic method is needed for measuring the rating values of training sentences and generated sentences. We employ the sentiment analyzer in Stanford CoreNLP (Manning et al., 2014) to do so. Specifically, we first invoke CoreNLP to output the 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. Some statistics of the data is given in Table 2. Hereafter, we use "rating" and "outcome" interchangeably.

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

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

Table 2: Numbers of sentences in each rating interval.

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 similar method for deriving polarity is basically reasonable as well.

For preparing the pseudo-parallel sentences, we first follow the ideas in (Guu et al., 2018) to generate some initial pairs. Specifically, we first calculate the Jaccard Index (JI) for each pair of sentences, and keep those with JI values no less then 0.5 as the 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 JI calculation 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 of the two sentences' ratings should be no less than 2. Here, the idea is that given the two sentences are similar enough in wordings (JI $\geq$ 0.5), if their rating scores are dissimilar enough, it looks plausible to conjecture that their wording difference is more likely 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 a 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 in composing the data points.

4.2 Comparative Methods

Our model is compared with two state-of-the-art models handling similar tasks.

Sequence to Better Sequence (S2BS) (Mueller et al., 2017): For training, S2BS also requires each sentence is associated with an outcome. For test, S2BS only revises a sentence such that the output is associated with a higher outcome, which is not a specified value. For comparison, we adapt our revision method for S2BS, by which their trained model is able to conduct quantifiable sentence revision. We tune the parameters for S2BS by following the suggestions in their source code.

Text Style Transfer (TST) (Shen et al., 2017): 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 rating 3, thus, our task is equal to revising a positive sentence into negative. We follow the suggested parameters reported in (Shen et al., 2017).

4.3 Evaluation Metric and Parameter Setting

Considering that our model's task is to revise a sentence such that its outcome (predicted by Stanford CoreNLP) satisfies a specified target, we define the metric as the mean absolute error (MAE) between the specified target outcome and the outcomes of revised sentences.

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

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

After tuning on the validation set, the determined parameters are: $\lambda_{rec} = 0.75$, $\lambda_{kl} = 0.6$, $\lambda_{mse} = 0.25$, $\lambda_{diff} = 0.2$, $\lambda_{sim} = 0.2$, $\lambda_{d-rec} = 0.1$, and the dimensions of the two factors are both 50. The parameter τ for revision takes $\exp(-100000)$ for both our model and S2BS.

4.4 Automatic Evaluation

We compare our model with S2BS by specifying five target ratings, namely 1, 2, 3, 4, and 5. Both our model and S2BS are fed the sentences in the testing dataset. For each sentence, both models are required to generate five revised sentences, each satisfying one of the target ratings.

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

We evaluate the MAE between the target outcome and the outcome of the revised sentences. Each model is trained for three times and the average results are reported in Table 3. "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, compared with S2BS, our model achieves smaller MAE values. One major reason is that we disentangle a content factor and an outcome factor, and design three components to leverage pseudo-parallel sentences. By modeling the wording difference, our model captures the keywords that cause 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 reconstruction can guide the editing procedure with the hints from the parallel sentences. In contrast, S2BS only disentangles one factor for capturing both content and outcome properties, and thus it cannot perform the same operations on sentence pairs. 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 2. We also report the average Edit Distance between the input sentences and the generated sentences to measure the degree of revisions. For T=1 and T=5, our model conducts more editing than S2BS, which brings in better MAE, while for T=$\{2, 3, 4\}$, our model generates more accurate sentences (i.e. better MAE) with less editing. This observation coincides with the fact that we need more editing to revise a sentence towards an extreme target (i.e., 1 and 5), such as including degree adverbs "very" and "extremely".

We also compare our model with TST for sentiment polarity transfer. We employ the same evaluation metric as used in (Shen et al., 2017): the sentiment accuracy of the transferred sentences.

	Neg. to Pos.	Pos. to Neg.
TST	0.7280	0.7097
Our Model	0.8836	0.7862

Table 4: Accuracy comparison with TST.

	Content Preservation (Range: [0, 2])	Fluency (Range: [1, 4])
TST	1.02	2.56
S2BS	0.70	2.53
Our Model	1.38	2.48

Table 5: Manual evaluation.

We define the revised sentences with ratings larger than 3 as positive, smaller than 3 as negative. The results are given in Table 4, where two accuracy values are reported: negative to positive, and the reverse. The results show that our model achieves better accuracy than TST in both transfer directions. One reason is that 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.

4.5 Manual Evaluation

We hire five workers to manually evaluate the quality of 500 sentences generated by each of our model and the compared models. The result is shown in Table 5. "Content Preservation" measures whether the generated sentence preserves the content of the input sentence. The score range is $\{$2: fully preserved, 1: partially preserved, 0: not preserved$\}$. "Content Preservation" is an important metric in this task since it is required that the output sentence and the original sentence should describe the same content subject. "Fluency" measures the grammatical quality of a sentence, which

	Generated sentence
E.g. 1	this tire center is amazing .
T=1	this tire center is horrible .
T=3	this tire center is really good .
T=5	this tire center is amazing .
E.g. 2	horrible food !
T=1	horrendous
T=3	their food amazing !
T=5	amazing delicious food ! recommend !
E.g. 3	decent food and wine selection , but nothing i will rush back for .
T=1	decent food and wine selection , but nothing i will rush for no .
T=3	decent food and wine selection , but i will never look back for .
T=5	decent food and wine selection , but excellent service, will return !
E.g. 4	our first time and we had a great meal , wonderful service .
T=1	our first time and we had a terrible meal , stale service .
T=3	our first time and we had a great meal , we have service .
T=5	our first time and we had a great meal , wonderful service .
E.g. 5	food is very addiction tasty !
T=1	food is just horrible here ?
T=3	food is just addiction here !
T=5	food is very yummy addiction !

Table 6: Case study.

ranges from 1 (bad) to 4 (good), by following the definition in TST (Shen et al., 2017).

The result shows that our model achieves the best content preservation score. Our editing procedure explicitly fixes the content factor and only modifies the outcome factor, which helps better preserve the content. In contrast, S2BS and TST include only one shared factor for both the content and the outcome, thus fail to distinguish one from the other. For the "Fluency" metric, S2BS and TST are slightly better than our model. Generally speaking, it is because our model introduces more powerful components for modeling the outcome differences between pseudo-parallel sentences, so as to achieve our goal of editing an input sentence to satisfy the expected outcome. However, these components do not contribute to the language quality of generated sentences.

4.6 Case Study

We show some examples produced by our model in Table 6. For each input, we specify three targets: 1, 3, and 5. For the first and the forth examples, the original sentences are not revised when the target rating is set to 5 (i.e., T=5) since the original sentences are already quite positive. For the first example, when T=3, "amazing" is revised to a relatively less positive phrase "really good". This case demonstrates that our model is able to capture the subtle difference in word sentiments, so that it can revise sentences reasonably according to the quantifiable rating guidance. Moreover, for the second example, we notice that our model revises the original sentence "horrible food !" to "amazing delicious food ! recommend !" for T=5. This case shows that our model not only changes one word with another having different sentiment, e.g. "horrible" to "amazing delicious", but also creatively introduces words from a new perspective, e.g. "recommend".

4.7 Ablation and Tuning Behavior Discussions

Recall that our model is a combination of a revised VAE, which disentangles two factors from a sentence 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 7. "None" refers to all three losses are removed, and it is basically worse than S2BS, which implies only using the revised VAE does not work well. As more losses added, the performance is gradually improved. Moreover, the dual reconstruction is more effective than the others.

	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 7: Ablation study.

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 $\mathcal{L}_{sim}$, because in this process, the encoder E_2 becomes more capable of disentangling the content factor, and thus z and z' become similar as they come from two similar input sentences, 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, let us assume there is a sentence x in the training set. Thus, the loss of reconstructing x from y and z is included in $\mathcal{L}_{rec}$. Assume that x is also included 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 the two losses 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 the weights of $\mathcal{L}_{rec}$ and $\mathcal{L}_{mse}$ fixed, we observe that $\mathcal{L}_{d-rec}$ also decreases, which complies with the above analysis.

5 Related Works

Inspired by the task of image style transfer (Gatys et al., 2016; Liu and Tuzel, 2016), researchers proposed the task of text style transfer and obtained some encouraging results (Fu et al., 2018; Hu et al., 2017; Jhamtani et al., 2017; Melnyk et al., 2017; Zhang et al., 2018; Li et al., 2018; Prabhumoye et al., 2018; Niu and Bansal, 2018). Existing studies on text style transfer mainly aim at transferring text from an original style into a target style, e.g., from negative to positive, from male to female, from rude/normal to polite; from modern text to Shakespeare style, etc. In contrast, our proposed task QuaSE assumes each sentence is associated with an outcome pertaining to continues values, and the editing is under the guidance of a specific target.

To transfer the style of a sentence, the paradigm of most works (Shen et al., 2017; Mueller et al., 2017; Prabhumoye et al., 2018) first learns the latent representation of the original sentence and then applies a decoder to generate the transferred sentence. A line of works (Shen et al., 2017; Mueller et al., 2017), including the studied task in this paper, assume that only non-parallel data is available for training. In such settings, VAEs (Kingma and Welling, 2013) are employed to learn the latent representations of sentences. Shen et al. (2017) assume a shared latent content distribution across text corpora belonging to different styles, and leverages the alignment of latent representations from different styles to perform style transfer. Mueller et al. (2017) associate the latent representations with a numerical outcome, which is a measurement of the style. A transferred sentence is generated from a modified latent representation. Different from the aforementioned works based on latent representations, Li et al. (2018) propose a simpler method that achieves attribute transfer by changing a few attribute marker words or phrases in the sentence that are indicative of a particular attribute, while leaving the rest of the sentence largely unchanged. The simple method is able to generate better-quality sentences than the aforementioned works. Besides style transfer, sentence editing models can be developed for other tasks. For example, Schmaltz et al. (2017) propose neural sequence-labelling models for correcting the grammatical errors of sentences.

6 Conclusions

We proposed a new task namely Quantifiable Sequence Editing (QuaSE), where a model needs to edit an input sentences towards the direction of a numerical outcome target. To tackle this task, we proposed a novel framework that simultaneously exploits the single sentences and pseudo-parallel sentence pairs. For evaluation, we prepared a dataset with Yelp sentences and their ratings. Experimental results show that our framework outperforms the compared methods under the measures of sentiment polarity accuracy and target value errors. Case studies show that our framework can generate some interesting sentences.

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.

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*.

Yifan Gao, Jianan Wang, Lidong Bing, Irwin King, and Michael R. Lyu. 2018. Difficulty controllable question generation for reading comprehension. *CoRR*, abs/1807.03586.

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.

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.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. In *The 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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.

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.

Tong Niu and Mohit Bansal. 2018. Polite dialogue generation without parallel data. *Transactions of the Association for Computational Linguistics (TACL)*.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Allen Schmaltz, Yoon Kim, Alexander M. Rush, and Stuart M. Shieber. 2017. Adapting sequence models for sentence correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2807–2813.

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.

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*.