MixPoet: Diverse Poetry Generation via Learning Controllable Mixed Latent Space

Xiaoyuan Yi, 1 Ruoyu Li, 3 Cheng Yang, 2 Wenhao Li, 1 Maosong Sun 1∗

1Department of Computer Science and Technology, Tsinghua University
Institute for Artificial Intelligence, Tsinghua University
State Key Lab on Intelligent Technology and Systems, Tsinghua University
2Beijing University of Posts and Telecommunications
36ESTATES PTE LTD, Singapore
{yi-xy16, liwh16}@mails.tsinghua.edu.cn, sms@tsinghua.edu.cn
yangcheng@bupt.edu.cn, ruoyuli1995@gmail.com

Abstract

As an essential step towards computer creativity, automatic poetry generation has gained increasing attention these years. Though recent neural models make prominent progress in some criteria of poetry quality, generated poems still suffer from the problem of poor diversity. Related literature researches show that different factors, such as life experience, historical background, etc., would influence composition styles of poets, which considerably contributes to the high diversity of human-authored poetry. Inspired by this, we propose MixPoet, a novel model that absorbs multiple factors to create various styles and promote diversity. Based on a semi-supervised variational autoencoder, our model disentangles the latent space into some subspaces, with each conditioned on one influence factor by adversarial training. In this way, the model learns a controllable latent variable to capture and mix generalized factor-related properties. Different factor mixtures lead to diverse styles and hence further differentiate generated poems from each other. Experiment results on Chinese poetry demonstrate that MixPoet improves both diversity and quality against three state-of-the-art models.

1 Introduction

Poetry is one of the most valuable cultural heritages for human beings. Characterized by its elegant expressions, colorful contents and diverse styles, this literary genre appeals to people across different ages and nationalities. Automatic poetry generation has attracted growing attention in the past several years because of its considerable research value in exploring computer creativity and building humanizing AI, which could also benefit the construction of intelligent assistants for entertainment and education.

Recent models mainly make efforts and achieve significant progress in improving some primary criteria of poetry quality, such as context coherence (Yan 2016) and topic relevance (Ghazvininejad et al. 2016; Li et al. 2018). However, beyond these criteria, generated poems still suffer from the problem of poor diversity.

Intuitively, the fundamental requirements of diversity in poetry generation could be two-fold: (1) poems generated with different topic words should be distinguishable from each other (inter-topic diversity) and (2) with the same topic word, the model should be able to generate distinct poems (intra-topic diversity). Nevertheless, most existing models fail to meet such requirements since they tend to remember some common patterns in the corpus and produce repetitive and generic contents, even with different topic words as input (Zhang et al. 2017; Yi et al. 2018a).

To address this problem, we must figure out what contributes to diversity. Related literature theories demonstrate that different factors would influence writing manners of human poets, such as their life experience (Dilthey 1985), historical background (Owen 1990), school of literary, etc. These factors lead to differences in thoughts, feelings, and expressions in poetry composition, which underlie the diverse styles of poets and make human-authored poems highly distinguishable, as observed in (Zhang et al. 2017). Figure 1 gives an example: under the same topic (the war), the poem created by a poet who lived in a powerful and prosperous dynasty tends to express strong confidence and aspiration; by contrast, the other created by a poet living in troubled times shows the sorrow and worry of being invaded.

Inspired by this, we propose a novel model, MixPoet,
which absorbs different influence factors to improve the diversity of generated poems. To exploit the underlying properties of factors, we resort to semi-supervised Variational AutoEncoder (VAE) (Kingma et al. 2014). We don’t assume the independence of the latent variable and influence factors because poetry style is tightly coupled with semantics (Emb-bler 1967). Instead, our model disentangles the latent space into some subspaces and makes each conditioned on one factor (together with the keyword) by adversarial training, to capture and mix generalized factor-related semantics. In the training phase, our model can predict factors of unlabelled poems and thus be trained in a semi-supervised manner. In the testing phase, by specifying different values for each factor, we can create various mixtures of factor properties that bring distinctive new styles for generated poems.

With the same given keyword, by manually varying the mixture, one can create distinct poems that simultaneously express properties of multiple factors, achieving intra-topic diversity. With different keywords as input, our model can automatically infer an appropriate factor mixture for each keyword and thus generate more distinguishable poems, improving inter-topic diversity.

In summary, our contributions are as follows:

- To the best of our knowledge, we are the first effort at generating poems that mix the properties of different factors for the sake of better diversity.
- We innovatively propose a semi-supervised MixPoet model to disentangle the latent space into different factor-conditioned subspaces by adversarial training.
- We experiment on Chinese poetry. Automatic and human evaluation results show that our model can controllably mix different factors and improve both the diversity and quality of generated poems, against three current state-of-the-art models.

2 Related Work

As an important chapter of automatic natural language generation, poetry generation has interested researchers for decades. After the early attempt of template-based models (Gervás 2001), systems based on statistical machine learning methods, such as genetic algorithms (Manurung 2003) and statistical machine translation approaches (He, Zhou, and Jiang 2012), make the first breakthrough and generate barely satisfactory poems.

The past several years have witnessed the rapid progress of neural networks, which also show notable advantages in poetry generation. Existing works mainly target at improving some primary criteria of poetry quality. At first, the Recurrent Neural Network (RNN) is used to generate fluent poems (Zhang and Lapata 2014; Hopkins and Kiela 2017). After that, pursuing better context coherence, the Polish model (Yan 2016) embellishes a generated poem several times. To enhance topic relevance, the Hafez system (Ghazvininejad et al. 2016) extracts more related keywords to bring more abundant topic information; the working memory model (Yi et al. 2018a) leverages an internal memory to store and access multiple topic words.

Despite the significant improvement on these criteria, models mentioned above fail to meet a higher requirement, the diversity. To handle this problem, the MRL model (Yi et al. 2018a) uses reinforcement learning to encourage high-TF-IDF words, which improves inter-topic diversity. The USPG model (Yang et al. 2018a) generates stylistic poetry by maximizing the mutual information between styles and poems, which promotes intra-topic diversity. Since USPG is trained in an unsupervised manner, the learned styles are indistinguishable and uninterpretable.

VAE has recently proven to be effective for generating various types of text (Zhao, Zhao, and Eskenazi 2017; Zhang et al. 2016). Related to our work, Yang et al. (2018b) use VAE to learn a context-conditioned latent variable for poetry generation. Hu et al. (2017) suppose the independence of latent space and attributes to generate single sentences but without constraints on semantics. Li et al. (2018) use adversarial training to match generated poems and given titles to strengthen topic relevance.

Our motivation and method considerably differ from these works. For better diversity, we apply adversarial training to the latent space (instead of explicit poems) and disentangle it into factor-conditioned (neither factor-independent nor context-conditioned) subspaces to involve various styles and generate diverse poems under the control of both required topic and factors. Besides, our model is semi-supervised and can be trained well with a fraction of labelled data.

3 Model

Before detailing the proposed MixPoet, we first formalize our task. Define $x$ as a poem with $n$ lines $x_i, x_2, \ldots, x_n$, each line with $l_i$ words as $x_i = x_{i,1}, x_{i,2}, \ldots, x_{i,l_i}$, and $w$ as a keyword representing the main topic. Suppose there are $m$ factors, $y_1, \ldots, y_m$. Since influence factors are quite complicated concepts, to simplify the problem, we discretize each factor $y_i$ into $k_i$ classes. By specifying different classes (values) for each factor, we can create $\prod_{i=1}^{m} k_i$ factor mixtures, with each leading to a new distinctive style. As poems with manually annotated factor labels are rare, we also utilize unlabelled data and define $p_h(x,w,y_1,y_2,\ldots,y_m)$ and $p_u(x,w)$ as the empirical distributions over labelled and unlabelled datasets respectively. Our goal is to generate poems which are relevant to $w$ on topic and concurrently accord with the mixed factors on style.

3.1 Basic Generator

We first present a basic generator, one of our baselines, which is also a part of MixPoet. We adopt an effective structure similar to that in (Yan 2016; Yi et al. 2018a).

Define $s_{i,j}$ as the corresponding GRU (Cho et al. 2014) decoder hidden state. Then the probability distribution of each $x_{i,j}$ to be generated is computed as:

$$s_{i,j} = GRU(s_{i,j-1}, e(x_{i,j-1}; g_{t-1}), 1) \tag{1}$$
$$s_{i,0} = f(e(w), a_i) \tag{2}$$
$$p(x_{i,j} | x_{i,<j}, x_{<i}, w) = softmax(f(s_{i,j})) \tag{3}$$

where $[;]$ means concatenation; $e(·)$ represents the embedding; $x_{<i}$ is the abbreviation of $x_1, \ldots, x_{i-1}$ (similar to
We first give the formalization based on a single factor, which is adopted in our previous work (Chen et al. 2019). We introduce the semi-supervised framework of our model, where we approximate the true prior distribution $p(x, y|z)$ and have

$$p(x, y|z) = p(y|w)p(z|x, w)p(x, z, y).$$

Such decomposition indicates how a poem is generated: if the user doesn’t provide any label, the model predicts an appropriate factor class by the keyword, then draws a sample of $z$ according to the required topic ($w$) and factor ($y$), and finally generates a poem ($x$). During this process we could manipulate both the topic and style of generated poems by separately specifying the keyword and factor class.

Then for labelled data, we can derive the lower bound:

$$\mathbb{E}_{q_w(z|x,w,y)}[\log p(x|z,w,y)] - KL[q_w(z|x,w,y)||p_\theta(z|w,y)] + \log p_\omega(y|w) = -\mathcal{L}(x, y, w) \leq \log p(x, y|w),$$

where we approximate the true prior distribution $p(z|w,y)$ and posterior distribution $q(z|x,w,y)$ with a prior network $p_\theta(z|w,y)$ and a recognition network $q_\omega(z|x,w,y)$ respectively. $\theta$ and $\phi$ are corresponding parameter sets.

By optimizing Eq. (6), we reconstruct the poem $x$, and minimize the KL divergence of the posterior and prior distributions. Besides, we also incorporate a classifier $p_\omega(y|w)$ to predict appropriate factor classes when the user doesn’t provide any label. $\omega$ represents the parameters of classifiers.

Since the labelled data is too limited to train the model well, as (Kingma et al. 2014), to utilize unlabelled poems, we treat the unobserved $y$ as another latent variable. In a similar vein, we can derive and maximize:

$$\mathbb{E}_{q_w(y|x,w)}[\mathcal{L}(x, y, w)] + H(q_\omega(y|x, w)) = -\mathcal{U}(x, w) \leq \log p(x|w),$$

where another classifier $q_\omega(y|x, w)$ is trained to infer classes for unlabelled poems with Gumbel-softmax (Jang, Gu, and Poole 2017) during the training process.

Ultimately, the total semi-supervised loss is:

$$\mathcal{L} = \mathbb{E}_{p(x,w,y)}[\mathcal{L}(x, y, w)] - \alpha \star \log q_\omega(y|x, w)] + \beta \star \mathbb{E}_{p_\omega(x,w)}[\mathcal{U}(x, w)],$$

where we also add the classification loss to the first term to train the classifier $q_\omega(y|x, w)$ utilizing both supervised and unsupervised signals. $\alpha$ and $\beta$ are hyper-parameters.

Figure 2 diagrams our model. In detail, we take the whole poem $x$ as a long sequence and feed it into a bidirectional GRU. Then we concatenate the last forward and backward hidden states to form $h$, the vector representation of $x$. The classifiers are implemented with Multi-Layer Perceptron (MLP): $p_\omega(y|w) = \text{softmax}(\text{MLP}(e(w), h))$ and $q_\omega(y|x, w) = \text{softmax}(\text{MLP}(e(w), h))$. We refer to $p_\omega(x|y,z,w,y)$ as the decoder (parameterized by $\psi$), which is just the basic generator introduced in Sec. 3.1 except that we set the initial decoder state as $s_{i,0} = f(e(w), o_i, z, c(y))$ to involve the latent variable and the factor.

### 3.3 Latent Space Mixture

The formulas above only focus on a single factor. To incorporate $m$ factors, we can assume that the latent space can be disentangled into $m$ subspaces $z = [z_1; \cdots; z_m]$. Without loss of generality, we give the formulation when $m=2$. By further assuming the independence of influence factors and the conditional independence of these subspaces, we have $p(z|w,y) = p(z_1|w,y_1)p(z_2|w,y_2)$. Accordingly, we need to replace the classifiers in Sec. 3.2 with $p_\omega(y_1|w)$, $p_\omega(y_2|w)$, $q_\omega(y_1|x,w)$ and $q_\omega(y_2|x,w)$ to predict $y_1$ and $y_2$ respectively. This disentanglement indicates that we can independently draw $z_1$ and $z_2$ from corresponding factor-conditioned subspaces to form the whole latent variable. That is, we get a latent space which mixes the properties of different factors. We design two methods to learn such mixed latent space.

**Mixture for Isotropic Gaussian Space** We call the first method MixPoet-IG since we assume the latent variable follows the isotropic Gaussian distribution as previous related works (Kingma et al. 2014) (Yang et al. 2018) usually do.

Then we can rewrite the KL divergence in Eq. (6) as $KL[q_\omega(z_1|x,w,y)||p_\theta(z_1|w,y)] + KL[q_\omega(z_2|x,w,y)||p_\theta(z_2|w,y)]$. Since $z_1$ and $z_2$
Algorithm 1: Training Process of MixPoet-AUS

1: for number of iterations do
2: Sample labelled batch \( \{x, w, y_1, y_2\} \);
3: Sample unlabelled batch \( \{x, w\} \) and sample corresponding predicted labels \( y_1 \sim q_\phi(\cdot|x, w), y_2 \sim q_\psi(\cdot|x, w) \);
4: Sample the posterior latent variable \( z \) and the prior \( z_1, z_2 \) with Eq. (9);
5: Train the four classifiers \( (\omega) \), recognition network \( (\phi) \) and decoder \( (\psi) \) in Eq. (8);
6: Adversarially train the recognition network \( (\phi) \) and prior networks \( (\theta) \) with Eq. (12);
8: end for

also follow the isotropic Gaussian distribution, we implement the recognition networks and the prior networks with MLP, for example, \( p_{\phi}(z_1|w, y_1) \sim \mathcal{N}(\mu_1, \sigma_1^2 I) \) where \( |\mu_1; \log \sigma_1^2| = MLP(e(w), e(y_1)) \). Then we can analytically minimize these two KL terms, draw samples of latent variables with the reparametrization trick (Kingma and Welling 2014) and train the whole model with Eq. (8).

Adversarial Mixture for Universal Space The second method is called MixPoet-AUS. Despite the tractability of computation, the isotropic Gaussian distribution may fail to learn more complex representations as discussed in (Dilokthanakul et al. 2017). We want to only keep the independence of \( z_1 \) and \( z_2 \) but with the internal dimensions of each subspace entangled. By this means, the model can learn more generalized latent representations with enough capacity to hold the broad concepts of influence factors and meanwhile control them independently.

Therefore, we don’t specify any concrete form of the latent space. Instead, we use a universal approximator (Makhzani et al. 2015) and make the model learn arbitrary complex forms by itself. In detail, for a conditional distribution \( q(z|c) \) with a condition \( c \), we assume:

\[
q(z|c, \eta) = \delta(z - MLP(c, \eta)), \tag{9}
\]

where \( \eta \) is random noise and \( \delta \) is the impulse function. By replacing \( c \) with a certain condition (e.g., \( w, y_1 \)) and sampling \( \eta \sim \mathcal{N}(0, 1) \) we can get samples of required latent variables (e.g., \( z_1 \)).

Then we use the density ratio loss (Rosca et al. 2017) to approximate the KL term as follows:

\[
KL[q_\phi(z|x, w, y_1, y_2)]p_\phi(z_1|w, y_1)p_\phi(z_2|w, y_2) = \mathbb{E}_{q_\phi(z|x, w, y_1, y_2)} \left[ \log \frac{q_\phi(z|x, w, y_1, y_2)}{p_\phi(z_1|w, y_1)p_\phi(z_2|w, y_2)} \right] \tag{10}
\]

\[
\approx \mathbb{E}_{q_\phi(z|x, w, y_1, y_2)} \left[ \log \frac{C_\psi(z, y_1, y_2)}{1 - C_\psi(z_1, z_2; y_1, y_2)} \right],
\]

where \( C_\psi \) is a latent discriminator (parameterized by \( \psi \)) which discriminates between latent values sampled from the posterior distribution and the ones independently sampled from the two factor-conditioned prior distributions.

| # of       | MC     | CL   | Others | UNK     | Total |
|------------|--------|------|--------|---------|-------|
| PT         | 799    | 608  | 675    | 9,052   | 11,134|
| TT         | 1,481  | 977  | 1,122  | 8,993   | 12,573|
| UNK        | 8,547  | 9,543| 7,654  | -       | 25,744|
| Total      | 10,827 | 11,128| 9,451  | 18,045  | 49,451|

Table 1: Statistics of CQCF. MC: military career, CL: countryside life, PT: prosperous times, TT: troubled times. ‘Others’ means poems that don’t belong to MC or CL. ‘UNK’ means unknown. We detail the collection methodology of CQCF in the supplementary file.

As in (Mohamed and Lakshminarayanan 2016) (Zhao et al. 2018), we use adversarial training to minimize this ratio loss which alternately optimizes the discriminator by:

\[
\max_{\psi} \mathbb{E}_{p_\psi(z_1|w, y_1)p_\psi(z_2|w, y_2)} \left[ \log \left( 1 - C_\psi(z_1; z_2, y_1, y_2) \right) \right] + \mathbb{E}_{q_\phi(z|x, w, y_1, y_2)} \left[ \log C_\psi(z, y_1, y_2) \right], \tag{11}
\]

and trains the recognition and prior networks by:

\[
\max_{\phi, \theta} \mathbb{E}_{q_\phi(z_1|w, y_1)q_\phi(z_2|w, y_2)} \left[ \log C_\psi(z_1; z_2, y_1, y_2) \right] - \mathbb{E}_{q_\psi(z|x, w, y_1, y_2)} \left[ \log C_\psi(z, y_1, y_2) \right]. \tag{12}
\]

In this adversarial training, we consider the prior network as a ‘generator’ and the latent values sampled from the recognition network as ‘real data’ in the standard Generative Adversarial Networks (Goodfellow et al. 2014). The complete training process is shown in Algorithm 1.

When the discriminator is successfully cheated \( (C_\psi(\cdot) \approx 0.5) \), the KL divergence can be minimized close to zeros. In this way, the model learns a sophisticated latent space and disentangles it into different factor-conditioned subspaces. In Sec. 4.1, we will show that compared to Mixpoet-IG, MixPoet-AUS learns more distinguishable latent representations and achieves better diversity.

3.4 Training

For MixPoet-IG, to alleviate the vanishing latent variable problem in VAE training, besides the annealing trick (Vinyals et al. 2016), we also add a BOW loss (Zhao, Zhao, and Eskenazi 2017) to Eq. (8) to force \( z \) to capture more global information. For MixPoet-AUS, since the discriminator is a crucial part for adversarial training, we adopt a powerful projection discriminator recently proposed in (Miyato and Koyama 2018) and apply the spectral normalization (Miyato et al. 2018) to the discriminator to stabilize the training process.

4 Experiments

4.1 Data

We mainly experiment on two typical factors: living experience and historical background. We discretize the first one into three classes: military career, countryside life and others; and the second one into two classes: prosperous times and troubled times. By mixing these factors, we can create six new styles. Then we build a labelled corpus called Chinese Quatrain Corpus with Factors (CQCF), which contains...


| Models   | inter-JS ↓ | intra-JS ↓ | LMS ↑ |
|----------|------------|------------|-------|
| fBasic   | -          | 9.15%      | 0.34  |
| Basic    | 2.58%      | -          | 0.31  |
| CVAE     | 2.34%      | 38.2%      | 0.33  |
| USPG     | 1.89%      | 5.01%      | 0.37  |
| MRL      | 1.28%      | -          | 0.33  |
| MixPoet-IG | 1.55%  | 8.55%      | 0.37  |
| MixPoet-AUS | 1.39% | 3.73%      | 0.39  |
| GT       | 0.12%      | -          | 0.68  |

Table 2: Automatic evaluation results of diversity. inter-JS: inter-topic Jaccard similarity. intra-JS: intra-topic Jaccard similarity. For calculating inter-JS, USPG and MixPoet predict appropriate styles/mixtures in terms of keywords.

49,451 poems, and each poem is labelled on at least one of the two factors. Statistics of CQCF are reported in Table 1. Besides, we also collect a Chinese Quatrain Corpus (CQC) as unlabelled data which comprises 117,392 poems. For CQC, we randomly select 4,500 poems for validation and testing, respectively, and the rest for training. For CQCF, we use 5% for validation, 5% for testing.

We use TextRank (Mihalcea and Tarau 2004) to extract keywords from poems to build <keyword, poem, labels> triplets, as in (Yi et al. 2018a).

4.2 Setups

We set the sizes of hidden state, context vector, latent variable, word embedding and factor embedding to 512, 512, 256, 256 and 64 respectively. The activation function is leaky ReLU for the discriminator and prior networks and is tanh for others. \( d = 3 \) in Eq.(4); \( \alpha = \beta = 1 \) in Eq. 1. Adam (Kingma and Ba 2015) with mini-batches (batch size=128) is used for optimization. To avoid overfitting, we also adopt dropout and \( l_2 \) norm regularization. For MixPoet-AUS, we update the discriminator five times per update of other parts. We first pre-train our model using both CQC and CQCF, and then fine-tune it with only CQCF. In testing, we adopt beam search (beam size=20) and apply explicit constraints to the search process to ensure that generated poems can meet the requirements of rhyme and rhythm. For fairness, all baselines share the same configuration.

4.3 Baselines for Comparisons

We compare the following baseline:\[1\] GT: ground truth, i.e. human-created poems. Basic: the generator introduced in Sec. 3.1. CVAE (Yang et al. 2018b): a conditional VAE with a hybrid decoder for poetry generation. USPG (Yang et al. 2018a): an unsupervised stylistic poetry generator which supports ten styles and can automatically infer an appropriate style by the input. MRL (Yi et al. 2018a): a reinforcement learning model which achieves the so-far best inter-topic diversity. fBasic (Wei, Zhou, and Cai 2018): a supervised stylistic poetry generator. We also pre-train fBasic with both CQC and CQCF, and then fine-tune it with CQCF. fBasic takes a straightforward structure, but it represents the typical supervised paradigm of style control.

4.4 Diversity Evaluation

As in (Yi et al. 2018a), we use Jaccard Similarity (JS) to evaluate diversity automatically. For inter-topic diversity, we generate 4,500 poems with different keywords but not any manually specified style/mixture, and then calculate JS of them. For intra-topic diversity, we calculate JS of poems generated with the same keyword but different specified styles. Besides, to prevent these models cheating by producing ill-formed content, we test the Language Model Score (LMS) (Yi et al. 2018a) of generated poems. Higher LMS indicates moderate fluency closer to human-authored poetry.

Table 2 shows that on inter-topic diversity, our model outperforms most baselines and gets very close to MRL. Though with distinct keywords as input, most models tend to generate repetitive phrases (see Figure 5) which inevitably worsen diversity. MixPoet and USPG incorporate diverse styles to further differentiate generated poems. However, the unsupervised design of USPG results in indistinguishable and uninterpretable learned styles which have no explicit semantic meaning and are too similar. Consequently, even with fewer styles (3*2 vs. 10), MixPoet still surpasses USPG.

MRL obtains the best inter-topic diversity by penalizing high-frequency words but fails to achieve intra-topic diversity. If without extra post-processing, MRL and Basic can only generate the same poem by a given keyword (equivalent to intra-JS=100%). CVAE could produce somewhat different poems by utilizing different samples of \( z \), but these poems heavily overlap with each other (intra-JS=38.2%). We can also see MixPoet-AUS gets better diversity than MixPoet-IG, as the former can learn more discriminable latent mixtures, we will analyse more in Sec. 4.7.

4.5 Factor Control Evaluation

Compared to USPG and MRL, our model attributes diversity to the differences of various styles and interprets each style different poems by utilizing different samples of \( z \). We can also see MixPoet-AUS gets better diversity than MixPoet-IG, as the former can learn more discriminable latent mixtures, we will analyse more in Sec. 4.7.

\[1\]Since our model supports a single keyword, for fairness, we remove the keyword extension module of CVAE and fBasic.
as a mixture of factor properties. Therefore, we also test if the generated poems are consistent with given factor classes.

For automatic evaluation, we generate 4,000 poems with each mixture and different keywords. Then we use a strong semi-supervised classifier (Miyato, Dai, and Goodfellow 2017), which achieves 0.87 and 0.74 F1 values for the two factors respectively, to measure the accuracy. For human evaluation, we generate 20 poems with each mixture (20*6 in total) and invite experts to identify the classes.

As shown in Figure 3, fBasic, the typical supervised method, performs the worst due to the quite limited and sparse labelled data. Benefiting from the semi-supervised structure, our model gets noticeable improvement. More than 80% and 60% of the generated poems meet specified classes of the two factors, respectively. Such results manifest that, to some extent, a poem generated by MixPoet can simultaneously express the properties of multiple factors.

### 4.6 Poetry Quality Evaluation

Since automatic metrics (e.g., perplexity and BLEU) deviate from the human evaluation manner (Yi et al. 2018a), we directly adopt human evaluation to assess quality. Following (Yan 2016; Zhang et al. 2017; Yi et al. 2018a), we consider: **Fluency** (is the generated poem well-formed?), **Coherence** (is the poem as a whole thematically and logically structured?), **Meaningfulness** (does the poem convey certain messages?), **Aesthetics** (does the poem have some poetic and artistic beauties?), **Topic Relevance** (is the poem consistent with the given topic word?) **Overall Quality** (the general impression on the poem). Each of the six criteria is scored on a 5-point scale ranging from 1 to 5.

We use MixPoet-AUS, which achieves better results in the above assessments, for human quality evaluation and subsequent analyses and refer to it as MixPoet. Then for each model, we generate 40 poems with different randomly-selected keywords. For GT, we choose poems containing corresponding keywords. Therefore, we get 240 (40*6) poems in total. Then we invite ten experts to evaluate in a blind review manner. Each poem is randomly assigned to two experts, and we average the two scores to mitigate personal biases. We refer to (Zhang et al. 2017; Yi et al. 2018a) for more details of the evaluation protocol.

As shown in Table 3 (Set 1), MixPoet gets notable improvement compared to other models. USPG and MixPoet infer appropriate labels by themselves in terms of different keywords; Set 2: the ones generated with the six mixtures (we present the average scores of all mixtures). Diacritic ** (p < 0.01) indicates MixPoet significantly outperforms baseline models; + (p < 0.05) and ++ (p < 0.01) indicate GT significantly outperforms all models. The Quadratic Weighted Kappa of human annotations is 0.67, which indicates acceptable inter-annotator agreement.

| Sets | Models | Fluency | Coherence | Meaning | Aesthetics | Relevance | Overall Quality |
|------|--------|---------|-----------|---------|-----------|-----------|----------------|
|      | Basic  | 3.00    | 2.54      | 2.30    | 2.71      | 2.54      | 2.35          |
|      | USPG   | 3.09    | 2.65      | 2.61    | 2.98      | 2.73      | 2.63          |
|      | CVAE   | 3.34    | 2.78      | 2.64    | 3.13      | 2.70      | 2.81          |
|      | MRL    | 3.91    | 3.66      | 3.36    | 3.73      | 3.19      | 3.55          |
|      | MixPoet| 4.18**  | 4.10**    | 3.75**  | 4.10**    | 3.39      | 3.98**        |
|      | GT     | 4.25    | 4.36**    | 4.19**  | 4.20      | 3.99**    | 4.25**        |

Table 3: Human evaluation results of quality. Set 1: poems generated without manually specified mixtures. USPG and MixPoet infer appropriate labels by themselves in terms of different keywords; Set 2: the ones generated with the six mixtures (we present the average scores of all mixtures). Diacritic ** (p < 0.01) indicates MixPoet significantly outperforms baseline models; + (p < 0.05) and ++ (p < 0.01) indicate GT significantly outperforms all models. The Quadratic Weighted Kappa of human annotations is 0.67, which indicates acceptable inter-annotator agreement.

4.7 Further Analyses

In Figure 4 (a), we visualize points sampled from the prior distributions conditioned on the six mixtures. We can find MixPoet-AUS learns more discriminable latent representations, but Mixpoet-IG fails to distinguish different mixtures. In Figure 4 (b), we vectorize poems by a neural language model and visualize them. We can see poems generated by our model, which mixes two factors (MC&PT), covers and bridges the two regions of human-authored poems, which indicates that our model successfully achieves the mixture not only on the latent space but also on generated poems.

From Figure 4 (c), we can observe that for Basic, CVAE and USPG, a few most frequent words account for a large proportion of generated contents, which leads to quite poor diversity. For instance, the most frequent five words account over 10% of all contents generated by Basic. By contrast,
Figure 4: (a) Visualization of samples of $z$ conditioned on the keyword ‘spring wind’ and different mixtures. (b) Visualization of human-created and MixPoet-generated poems. (c) Accumulative frequency of the most frequent k words in generated poems. (d) Keywords with the highest prediction probability for each mixture.

Figure 5: (a) Two poems generated by Basic using different keywords. Repetitive phrases are marked in red. (b) Using the same keyword, two poems generated by MixPoet with different mixtures. Phrases meeting different factor classes are marked in corresponding colors.

5 Conclusion and Future Work
In this work, inspired by related literature theories, we propose MixPoet to address the problem of poor diversity in poetry generation. Based on a semi-supervised VAE, our model disentangles the latent space into different subspaces with each conditioned on one factor which influences human poetry composition. In this way, the generated poems can simultaneously express mixed properties of multiple factors to some degree. By varying the mixture for the same keyword or inferring appropriate factor classes with different keywords, our model differentiates generated poems and hence promotes intra-/inter-topic diversity and quality against three state-of-the-art models.

In the future, we will endeavor to incorporate more factors, such as love experience, school of literary and gender, with finer-granularity discretization. We will also consider modeling the dependence of influence factors, since some factors may be correlated with each other, e.g., gender and living experience, and then apply our model to other kinds of text like story and essay.

Acknowledgments
This research is supported by the Major Program of the National Social Science Fund of China (Project No.18ZDA238).

\(^2\)MixPoet will be incorporated into Jiuge, the THUNLP online poetry generation system (https://jiuge.thunlp.cn).
[Zhang et al. 2017] Zhang, J.; Feng, Y.; Wang, D.; Wang, Y.; Abel, A.; Zhang, S.; and Zhang, A. 2017. Flexible and creative chinese poetry generation using neural memory. In ACL, 1364–1373. Association for Computational Linguistics.

[Zhao et al. 2018] Zhao, J.; Kim, Y.; Zhang, K.; Rush, A. M.; and LeCun, Y. 2018. Adversarially regularized autoencoders. In ICML.

[Zhao, Zhao, and Eskenazi 2017] Zhao, T.; Zhao, R.; and Eskenazi, M. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In ACL, 654–664. Association for Computational Linguistics.