# Composed Variational Natural Language Generation for Few-shot Intents

**Congying Xia**[1][*]**, Caiming Xiong**[2]**, Philip Yu**[1] **and Richard Socher**[2]
[1]University of Illinois at Chicago, Chicago, IL, USA
[2]Salesforce Research, Palo Alto, CA, US
`{cxia8, psyu}@uic.edu,{cxiong, rsocher}@salesforce.com`

## Abstract

In this paper, we focus on generating training examples for few-shot intents in the realistic imbalanced scenario. To build connections between existing many-shot intents and few-shot intents, we consider an intent as a combination of a domain and an action, and propose a composed variational natural language generator (CLANG), a transformer-based conditional variational autoencoder. CLANG utilizes two latent variables to represent the utterances corresponding to two different independent parts (domain and action) in the intent, and the latent variables are composed together to generate natural examples. Additionally, to improve the generator learning, we adopt the contrastive regularization loss that contrasts the in-class with the out-of-class utterance generation given the intent. To evaluate the quality of the generated utterances, experiments are conducted on the generalized few-shot intent detection task. Empirical results show that our proposed model achieves state-of-the-art performances on two real-world intent detection datasets.

## 1 Introduction

Intelligent assistants have gained great popularity in recent years since they provide a new way for people to interact with the Internet conversationally (Hoy, 2018). However, it is still challenging to answer people's diverse questions effectively. Among all the challenges, identifying user intentions from their spoken language is important and essential for all the downstream tasks.

Most existing works (Hu et al., 2009; Xu and Sarikaya, 2013; Chen et al., 2016; Xia et al., 2018) formulate intent detection as a classification task and achieve high performance on pre-defined intents with sufficient labeled examples. With this

---

[*]Work was done when Congying was a research intern at Salesforce Research.

ever-changing world, a realistic scenario is that we have imbalanced training data with existing many-shot intents and insufficient few-shot intents. Previous intent detection models (Yin, 2020; Yin et al., 2019) deteriorate drastically in discriminating the few-shot intents.

To alleviate this scarce annotation problem, several methods (Wei and Zou, 2019; Malandrakis et al., 2019; Yoo et al., 2019) have been proposed to augment the training data for low-resource spoken language understanding (SLU). Wei and Zou (2019) introduce simple data augmentation rules for language transformation like insert, delete and swap. Malandrakis et al. (2019) and Yoo et al. (2019) utilize variational autoencoders (Kingma and Welling, 2013) with simple LSTMs (Hochreiter and Schmidhuber, 1997) that have limited model capacity to do text generation. Furthermore, these models are not specifically designed to transfer knowledge from existing many-shot intents to few-shot intents.

In this paper, we focus on transferable natural language generation by learning how to compose utterances with many-shot intents and transferring to few-shot intents. When users interact with intelligent assistants, their goal is to query some information or execute a command in a certain domain (Watson Assistant, 2017). For instance, the intent of the input "what will be the highest temperature next week" is to ask about the weather. The utterance can be decomposed into two parts, "what will be" corresponding to an action "Query" and "the highest temperature" related to the domain "Weather". These actions or domains are very likely to be shared among different intents including the few-shot ones (Xu et al., 2019). For example, there are a lot of actions ("query", "set", "remove") that can be combined with the domain of "alarm". The action "query" also exists in multiple domains like "weather", "calendar" and "movie". Ideally, if we

can learn the expressions representing for a certain action or domain and how they compose an utterance for existing intents, then we can learn how to compose utterances for few-shot intents naturally. Therefore, we define an intent as a combination of a domain and an action. Formally, we denote the domain as $y_d$ and the action as $y_a$. Each intent can be expressed as $y = (y_d, y_a)$.

A composed variational natural language generator (CLANG) is proposed to learn how to compose an utterance for a given intent with an action and a domain. CLANG is a transformer-based (Vaswani et al., 2017) conditional variational autoencoder (CVAE) (Kingma et al., 2014). It contains a bi-latent variational encoder and a decoder. The bi-latent variational encoder utilizes two independent latent variables to model the distributions of action and domain separately. Special attention masks are designed to guide these two latent variables to focus on different parts of the utterance and disentangle the semantics for action and domain separately. Through decomposing utterances for existing many-shot intents, the model learns to generate utterances for few-shot intents as a composition of the learned expressions for domain and action.

Additionally, we adopt the contrastive regularization loss to improve our generator learning. During the training, an in-class utterance from one intent is contrasted with an out-of-class utterance from another intent. Specifically, the contrastive loss is to constrain the model to generate the positive example with a higher probability than the negative example with a certain margin. With the contrastive loss, the model is regularized to focus on the given domain and intent and the probability of generating negative examples is reduced.

To quantitatively evaluate the effectiveness of CLANG for augmenting training data in low-resource intent detection, experiments are conducted for the generalized few-shot intent detection task (GFSID) (Xia et al., 2020). GFSID aims to discriminate a joint label space consisting of both existing many-shot intents and few-shot intents.

Our contributions are summarized below. 1) We define an intent as a combination of a domain and an action to build connections between existing many-shot intents and few-shot intents. 2) A composed variational natural language generator (CLANG) is proposed to learn how to compose an utterance for a given intent with an action and a domain. Utterances are generated for few-shot in-

tents via a composed variational inferences process. 3) Experiment results show that CLANG achieves state-of-the-art performance on two real-world intent detection datasets for the GFSID task.

## 2 Composed Variational Natural Language Generator

In this section, we introduce the composed variational natural language generator (CLANG). As illustrated in Figure 1, CLANG consists of three parts: input representation, bi-latent variational encoder, and decoder.

### 2.1 Input Representation

For a given intent $y$ decomposed into a domain $y_d$ and an action $y_a$ and an utterance $x = (w_1, w_2, ..., w_n)$ with $n$ tokens, we designed the input format like BERT as ([CLS], $y_d$, $y_a$, [SEP], $w_1$, $w_2$, ..., $w_n$, [SEP]). As the example in Figure 1, the intent has the domain of "weather" and the action of "query". The utterance is "what will be the highest temperature next week". The input is represented as ([CLS], weather, query, [SEP], what, will, be, the, highest, temperature, next, week, [SEP]).

Texts are tokenized into subword units by Word-Piece (Wu et al., 2016). The input embeddings of a token sequence are represented as the sum of three embeddings: token embeddings, position embeddings (Vaswani et al., 2017), and segment embeddings (Devlin et al., 2018). The segment embeddings are learned to identify the intent and the utterance with different embeddings.

### 2.2 Bi-latent Variational Encoder

As illustrated in Figure 1, the bi-latent variational encoder is to encode the input into two latent variables that contain the disentangled semantics in the utterance corresponding to domain and action separately.

Multiple transformer layers (Vaswani et al., 2017) are utilized in the encoder. Through the self-attention mechanism, these transformer layers not only extract semantic meaningful representations for the tokens, but also model the relation between the intent and the utterance. The embeddings for the domain token and the action token output from the last transformer layer are denoted as $\mathbf{e}_d$ and $\mathbf{e}_a$. We encode $\mathbf{e}_d$ into variable $z_d$ to model the distribution for the domain and $\mathbf{e}_a$ are encoded into variable $z_a$ to model the distribution for the action.
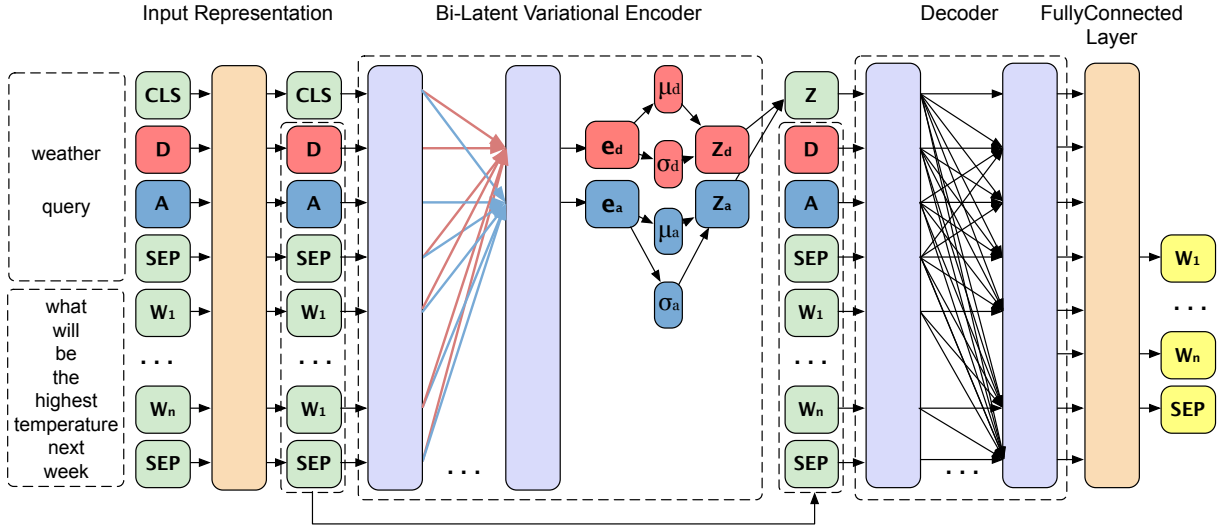
Ideally, we want to disentangle the information

Figure 1: The overall framework of CLANG.

for the domain and the action, making $\mathbf{e}_d$ attend to tokens related to domain and $\mathbf{e}_a$ focus on the expressions representing the action. To achieve that, we make a variation of the attention calculations in transformer layers to avoid direct interactions among the domain token and the action token in each layer.

Instead of applying the whole bidirectional attention to the input, an attention mask matrix $\mathbf{M} \in \mathbb{R}^{N \times N}$ is added to determine whether a pair of tokens can be attended to each other (Dong et al., 2019). $N$ is the length of the input. For the $l$-th Transformer layer, the output of a self-attention head $\mathbf{A}_l$ is computed via:

$$
\begin{aligned}
\mathbf{Q} &= \mathbf{T}^{l-1}\mathbf{W}_Q^l, \\
\mathbf{K} &= \mathbf{T}^{l-1}\mathbf{W}_K^l, \\
\mathbf{V} &= \mathbf{T}^{l-1}\mathbf{W}_V^l, \\
\mathbf{A}_l &= \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} + \mathbf{M}\right)\mathbf{V},
\end{aligned}
\tag{1}
$$

where the attention mask matrix calculated as:

$$
\mathbf{M}_{ij} = \begin{cases} 0, & \text{allow to attend;} \\ -\infty, & \text{prevent from attending.} \end{cases}
\tag{2}
$$

The output of the previous transformer layer $\mathbf{T}^{l-1} \in \mathbb{R}^{N \times d_h}$ is linearly projected to a triple of queries, keys and values parameterized by matrices $\mathbf{W}_Q^l, \mathbf{W}_K^l, \mathbf{W}_V^l \in \mathbb{R}^{d_h \times d_k}$. $d_h$ is the hidden dimension for the transformer layer, and $d_k$ is the hidden dimension for a self-attention head.

The proposed attention mask for the domain token and the action token is illustrated in Figure 2. The Domain $y_d$ and the action $y_a$ are prevented

from attending to each other. All the other tokens have are allowed to have full attentions. The elements in the mask matrix for the attentions between domain and action are $-\infty$, and 0 for all the others.

The disentangled embeddings $\mathbf{e}_d$ and $\mathbf{e}_a$ are encoded into two latent variables $z_d$ and $z_a$ to model the posterior distributions determined by the intent elements separately: $p(z_d|x, y_d), p(z_a|x, y_a)$. The latent variable $z_d$ is conditioned on the domain $y_d$, while $z_a$ is controlled by the action $y_a$. By modeling the true distributions, $p(z_d|x, y_d)$ and $p(z_a|x, y_a)$, using a known distribution that is easy to sample from (Kingma et al., 2014), we constrain the prior distributions, $p(z_d|y_d)$ and $p(z_a|y_a)$, as multivariate standard Gaussian distributions. A reparametrization trick (Kingma and Welling, 2013) is used to generate the latent vector $z_d$ and $z_a$ separately. Gaussian parameters ($\mu_d$, $\mu_a$, $\sigma_d^2$, $\sigma_a^2$) are projected from $\mathbf{e}_d$ and $\mathbf{e}_a$:

$$
\begin{aligned}
\mu_d &= \mathbf{e}_d\mathbf{W}_{\mu_d} + b_{\mu_d}, \\
\log(\sigma_d^2) &= \mathbf{e}_d\mathbf{W}_{\sigma_d} + b_{\sigma_d}, \\
\mu_a &= \mathbf{e}_a\mathbf{W}_{\mu_a} + b_{\mu_a}, \\
\log(\sigma_a^2) &= \mathbf{e}_a\mathbf{W}_{\sigma_a} + b_{\sigma_a},
\end{aligned}
\tag{3}
$$

where we have $\mathbf{W}_{\mu_d}, \mathbf{W}_{\mu_a}, \mathbf{W}_{\sigma_d}, \mathbf{W}_{\sigma_a} \in \mathbb{R}^{d_h \times d_h}$, $b_{\mu_d}, b_{\mu_a}, b_{\sigma_d}, b_{\sigma_a} \in \mathbb{R}^{d_h}$. Noisy variables $\varepsilon_d \sim \mathcal{N}(0, \mathbf{I}), \varepsilon_a \sim \mathcal{N}(0, \mathbf{I})$ are utilized to sample $z_d$ and $z_a$ from the learned distribution:

$$
\begin{aligned}
z_d &= \mu_d + \sigma_d \cdot \varepsilon_d, \\
z_a &= \mu_a + \sigma_a \cdot \varepsilon_a.
\end{aligned}
\tag{4}
$$

The KL-loss function is applied to regularize the prior distributions for these two latent variables to

be close to the Gaussian distributions:

$$\mathcal{L}_{KL} = \mathbb{D}_{\mathrm{KL}}[q(z_d|x, y_d), p(z_d|y_d)]$$
$$+ \mathbb{D}_{\mathrm{KL}}[q(z_a|x, y_a), p(z_a|y_a)]. \quad (5)$$

A fully connected layer with Gelu (Hendrycks and Gimpel, 2016) activation function is applied on $z_d$ and $z_a$ to compose these two latent variables together and outputs $z$. The composed latent information $z$ is utilized in the decoder to do generation.
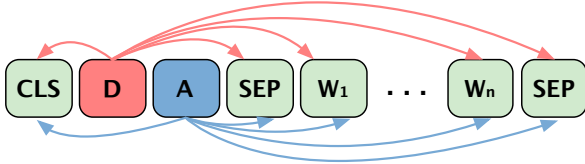


Figure 2: The attention map of domain and action in the encoder.

## 2.3 Decoder

The decoder utilizes the composed latent information together with the intent to reconstruct the input utterance $p(x|z_d, z_a, y_d, y_a)$. As shown in Figure 1, a residual connection is built from the input representation to the decoder to get the embeddings for all the tokens. To keep a fixed length and introduce the composed latent information $z$ into the decoder, we replace the first [CLS] token with $z$.

The decoder is built with multiple transformer layers to generate the utterance. Text generation is a sequential process that we use the left context to predict the next token. To simulate the left-to-right generation process, another attention mask is utilized for the decoder. In the attention mask for the decoder, tokens in the intent can only attend to intent tokens, while tokens in the utterance can attend to both the intent and all the left tokens in the utterance.

For the first token $z$ which holds composed latent information, it is only allowed to attend to itself due to the vanishing latent variable problem. The latent information can be overwhelmed by the information of other tokens when adapting VAE to natural language generators either for LSTM (Zhao et al., 2017) or transformers (Xia et al., 2020). To further increase the impact of the composed latent information $z$ and alleviate the vanishing latent variable problem, we concatenate the token representations of $z$ to all the other token embeddings output from the last transformer layer in the decoder. The hidden dimension increases to $2 \times d_h$ after the concatenation. To reduce the hidden dimension

to $d_h$ and get the embeddings to decode the vocabulary, two fully-connected (FC) layers followed by a layer normalization (Ba et al., 2016) are applied on top of the transformer layers. Gelu is used as the activation function in these two FC layers. The embeddings output from these two FC layers are decoded into tokens in the vocabulary. The embeddings at position $i = \{1, ..., n - 1\}$ are used to predict the next token at position $i + 1$ till the [SEP] token is generated.

To train the decoder to reconstruct the input, a reconstruction loss is formulated as:

$$\mathcal{L}_r = -\mathbb{E}_{q(z_d|x, y_d), q(z_a|x, y_a)}[\log p(x|z_d, z_a, y_d, y_a)].$$
$$(6)$$

## 2.4 Learning with contrastive loss

Although the model can generate utterances for a given intent, such as "are there any alarms set for seven am" for "Alarm Query", there are some negative utterances generated. For example, "am i free between six to seven pm" is generated with the intent of "Alarm Query". This would be because in the training, it lacks supervision to distinguish in-class from out-of-class examples especially for few-shot intents. To alleviate the problem, we adopt a contrastive loss in the objective function and reduce the probability to generate out-of-class samples.

Given an intent $y = (y_d, y_a)$, an in-class utterance $x^+$ from this intent and an out-of-class utterance $x^-$ from another intent. The contrastive loss constrains the model to generate the in-class example $x^+$ with a higher probability than $x^-$. In the same batch, we feed the in-class example $(y_d, y_a, x^+)$ and the out-of-class example $(y_d, y_a, x^-)$ into CLANG to model the likelihood: $P(x^+|y)$ and $P(x^-|y)$. The chain rule is used to calculate the likelihood of the whole utterance: $p(x|y) = p(w_1|y)p(w_2|y, w_1)...p(w_n|y, w_1, ..., T_{w_{n-1}})$. In the contrastive loss, the log-likelihood of the in-class example is constrained to be higher than the out-of-class example with a certain margin $\lambda$:

$$\mathcal{L}_c = \max\{0, \lambda - \log p(x^+|y) + \log p(x^-|y)\}. \quad (7)$$

To leverage challenging out-of-class utterances, we choose the most similar utterance with a different intent as the out-of-class utterance. Three indicators are considered to measure the similarity between the in-class utterance and all the utterances with a different intent: the number of shared uni-grams $s_1$, bi-grams $s_2$ between the utterances and

3382

the number of shared uni-grams between the name of intents $s_3$. The sum of these three numbers, $s = s_1 + s_2 + s_3$, is utilized to find the out-of-class utterance with the highest similarity. If there are multiple utterances having the same highest similarity $s$, we random choose one as the negative utterance.

The overall loss function is a summation of the KL-loss, the reconstruction loss and the contrastive loss:

$$\mathcal{L} = \mathcal{L}_{KL} + \mathcal{L}_r + \mathcal{L}_c. \tag{8}$$

### 2.5 Generalized Few-shot Intent Detection

Utterances for few-shot intents are generated by sampling two latent variables, $z_d$ and $z_a$, separately from multivariate standard Gaussian distributions. Beam search is applied to do the generation. To improve the diversity of the generated utterances, we sample the latent variables for $s$ times and save the top $k$ results for each time. The overall generation process follows that of Xia et al. (2020).

These generated utterances are added to the original traning dataset to alleviate the scare annotation problem. We finetune BERT with the augmented dataset to solve the generalized few-shot intent detection task. The whole pipeline is referred as BERT + CLANG in the experiments.

| Dataset | SNIPS-NLU | NLUED |
|---|---|---|
| Vocab Size | 10,896 | 6,761 |
| #Total Classes | 7 | 64 |
| #Few-shot Classes | 2 | 16 |
| #Few-shots / Class | 1 or 5 | 1 or 5 |
| #Training Examples | 7,858 | 7,430 |
| #Training Examples / Class | 1571.6 | 155 |
| #Test Examples | 2,799 | 1,076 |
| Average Sentence Length | 9.05 | 7.68 |

Table 1: Data Statistics for SNIPS-NLU and NLUED. #Few-shot examples are excluded in the #Training Exampels. For NLUED, the statistics is reported for KFold_1.

## 3 Experiments

To evaluate the effectiveness of the proposed approach for generating labeled examples for few-shot intents, experiments are conducted for the GF-SID task on two real-world datasets. The few-shot intents are augmented with utterances generated from CLANG.

### 3.1 Datasets

Following (Xia et al., 2020), two public intent detection datasets are used in the experiments: SNIPS-NLU (Coucke et al., 2018) and NLUED

(Xingkun Liu and Rieser, 2019). These two datasets contain utterances from users when interacting with intelligent assistants and are annotated with pre-defined intents. Dataset details are illustrated in Table 1.

**SNIPS-NLU**[1] contains seven intents in total. Two of them (RateBook and AddToPlaylist) as regraded as few-shot intents. The others are used as existing intents with sufficient annotation. We randomly choose 80% of the whole data as the training data and 20% as the test data.

**NLUED**[2] is a natural language understanding dataset with 64 intents for human-robot interaction in home domain, in which 16 intents as randomly selected as the few-shot ones. A sub-corpus of 11, 036 utterances with 10-folds cross-validation splits is utilized.

### 3.2 Baselines

We compare the proposed model with a few-shot learning model and several data augmentation methods. 1) Prototypical Network (Snell et al., 2017) (PN) is a distance-based few-shot learning model. It can be extended to the GFSID task naturally by providing the prototypes for all the intents. BERT is used as the encoder for PN to provide a fair comparison. We fine-tune BERT together with the PN model. This variation is referred to as BERT-PN+. 2) BERT. For this baseline, we over-sampled the few-shot intents by duplicating the few-shots to the maximum training examples for one class. 3) SVAE (Bowman et al., 2015) is a variational autoencoder built with LSTMs. 4) CGT (Hu et al., 2017) adds a discriminator based on SVAE to classify the sentence attributes. 5) EDA (Wei and Zou, 2019) uses simple data augmentations rules for language transformation. We apply three rules in the experiment, including insert, delete, and swap. 6) CG-BERT (Xia et al., 2020) is the first work that combines CVAE with BERT to do few-shot text generation. BERT is fine-tuned with the augmented training data for these generation baselines. The whole pipelines are referred to as BERT + SVAE, BERT + CGT, BERT + EDA and BERT + CG-BERT in Table 2. An ablation study is also provided to understand the importance of contrastive loss by removing it from CLANG.

---

|  | Many-shot | Few-shot | H-Mean | Many-shot | Few-shot | H-Mean |
|---|---|---|---|---|---|---|
|  | SNIPS-NLU 1-shot | | | SNIPS-NLU 5-shot | | |
| BERT-PN+ | $92.66 \pm 4.49$ | $60.52 \pm 7.58$ | $72.99 \pm 5.97$ | $95.96 \pm 1.13$ | $86.03 \pm 2.00$ | $90.71 \pm 1.19$ |
| BERT | $98.20 \pm 0.06$ | $44.42 \pm 4.35$ | $57.74 \pm 7.50$ | $98.34 \pm 0.10$ | $81.82 \pm 6.16$ | $89.22 \pm 3.74$ |
| BERT + SVAE | $98.24 \pm 0.09$ | $45.15 \pm 5.54$ | $61.67 \pm 5.11$ | $\mathbf{98.34 \pm 0.06}$ | $82.10 \pm 4.06$ | $89.49 \pm 2.47$ |
| BERT + CGT | $98.20 \pm 0.07$ | $45.80 \pm 5.68$ | $62.30 \pm 5.17$ | $98.32 \pm 0.14$ | $82.65 \pm 4.31$ | $89.78 \pm 2.83$ |
| BERT + EDA | $98.20 \pm 0.08$ | $47.52 \pm 5.96$ | $63.87 \pm 5.29$ | $98.09 \pm 0.18$ | $82.00 \pm 3.47$ | $89.30 \pm 2.12$ |
| BERT + CG-BERT | $98.13 \pm 0.15$ | $63.04 \pm 5.49$ | $76.65 \pm 4.24$ | $98.30 \pm 0.17$ | $86.89 \pm 4.05$ | $92.20 \pm 2.32$ |
| BERT + CLANG | $\mathbf{98.34 \pm 0.10}$ | $\mathbf{64.63 \pm 6.16}$ | $\mathbf{77.86 \pm 4.39}$ | $\mathbf{98.34 \pm 0.06}$ | $\mathbf{88.04 \pm 1.34}$ | $\mathbf{92.90 \pm 0.71}$ |
|  | NLUED 1-shot | | | NLUED 5-shot | | |
| BERT-PN+ | $81.24 \pm 2.76$ | $18.95 \pm 4.42$ | $30.67 \pm 5.53$ | $83.41 \pm 2.62$ | $60.28 \pm 4.19$ | $69.93 \pm 3.49$ |
| BERT | $94.00 \pm 0.93$ | $7.88 \pm 3.28$ | $14.39 \pm 5.66$ | $\mathbf{94.12 \pm 0.89}$ | $51.69 \pm 3.19$ | $66.67 \pm 2.51$ |
| BERT + SVAE | $93.80 \pm 0.70$ | $8.88 \pm 3.66$ | $16.01 \pm 6.06$ | $93.60 \pm 0.63$ | $54.03 \pm 3.91$ | $68.42 \pm 3.06$ |
| BERT + CGT | $94.00 \pm 0.66$ | $9.33 \pm 3.68$ | $16.78 \pm 6.16$ | $93.61 \pm 0.63$ | $54.70 \pm 4.06$ | $68.96 \pm 3.17$ |
| BERT + EDA | $93.78 \pm 0.66$ | $11.65 \pm 4.89$ | $20.41 \pm 7.56$ | $93.71 \pm 0.64$ | $57.22 \pm 4.35$ | $70.95 \pm 3.35$ |
| BERT + CG-BERT | $\mathbf{94.01 \pm 0.70}$ | $20.39 \pm 5.77$ | $33.12 \pm 7.92$ | $93.80 \pm 0.60$ | $61.06 \pm 4.29$ | $73.88 \pm 3.10$ |
| BERT + CLANG | $93.60 \pm 0.79$ | $\mathbf{22.03 \pm 6.10}$ | $\mathbf{35.29 \pm 8.05}$ | $93.29 \pm 0.86$ | $\mathbf{66.44 \pm 3.07}$ | $\mathbf{77.56 \pm 2.05}$ |

Table 2: Generalized few shot intent detection with 1-shot and 5-shot settings on SNIPS-NLU and NLUED. Seen is the accuracy on the seen intents ($acc_s$), Unseen/Novel is the accuracy on the novel intents ($acc_s$), H-Mean is the harmonic mean of seen and unseen accuracies.

### 3.3 Implementation Details

Both the encoder and the decoder use six transformer layers. Pre-trained weights from BERT-base are used to initialize the embeddings and the transformer layers. The weights from the first six layers in BERT-base are used to initialize the transformer layers in the encoder and the later six layers are used to initialize the decoder. Adam optimizer (Kingma and Ba, 2014) is applied for all the experiments. The margin for the contrastive loss is 0.5 for all the settings. All the hidden dimensions used in CLANG is 768. For CLANG, the learning rate is 1e-5 and the batch size is 16. Each epoch has 1000 steps. Fifty examples from the training data are sampled as the validation set. The reconstruction error on the validation set is used to search for the number of training epochs in the range of [50, 75, 100]. The reported performances of CLANG and the ablation of contrastive loss are both trained with 100 epochs.

The hyperparameters for the generation process including the top index $k$ and the sampling times $s$ are chosen by evaluating the quality of the generated utterances. The quality evaluation is described in section 3.5. We search $s$ in the list of [10, 20], and $k$ in the list of [20, 30]. We use $k = 30$ and $s = 20$ for BERT + CLANG in NLUED, while use $k = 30$ and $s = 10$ for all the other experiments. When fine-tuning BERT for the GFSID task, we fix the hyperparameters as follows: the batch size is 32, the learning rate is 2e-5, and the number of the training epochs is 3.

### 3.4 Experiment Results

The experiment results for the generalized few-shot intent detection task are shown in Table 2. Performance is reported for two datasets with both 1-shot and 5-shot settings. For SNIPS-NLU, the performance is calculated with the average and the standard deviation over 5 runs. The results on NLUED are reported over 10 folds.

Three metrics are used to evaluate the model performances, including the accuracy on existing many-shot intents ($acc_m$), the accuracy on few-shot intents ($acc_f$) together with their harmonic mean ($H$). As the harmonic mean of $acc_m$ and $acc_f$, $H$ is calculated as:

$$H = 2 \times (acc_m \times acc_f)/(acc_m + acc_f). \quad (9)$$

We choose the harmonic mean as our evaluation criteria instead of the arithmetic mean because the overall results are significantly affected by the many-shot class accuracy $acc_m$ over the few-shot classes $acc_f$ in arithmetic mean (Xian et al., 2017). Instead, the harmonic mean is high only when the accuracies on both many-shot and few-shot intents are high. Due to this discrepancy, we evaluate the harmonic mean which takes a weighted average of the many-shot and few-shot accuracy.

As illustrated in Table 2, the proposed pipeline BERT + CLANG achieves state-of-the-art performance on the accuracy for many-shot intents, few-shot intents, and their harmonic mean for the SNIPS-NLU dataset. As for the NLUED dataset, BERT + CLANG outperforms all the baselines

on the accuracy for few-shot intents and the harmonic mean, while achieves comparable results on many-shot intents compared with the best baseline. Since the many-shot intents have sufficient training data, the improvement mainly comes from few-shot intents with scarce annotation. For example, the accuracy for few-shot intents on NLUED with the 5-shot setting improves 5% from the best baseline (BERT + CG-BERT).

Compared to the few-shot learning method, CLANG achieves better performance consistently in all the settings. BERT-PN+ achieves decent performance on many-shot intents while lacks the ability to provide embeddings that can be generalized from existing intents to few-shot intents.

For data augmentation baselines, CLANG obtains the best performance on few-shot intents and the harmonic mean. These results demonstrate the high quality and diversity of the utterances generated form CLANG. CGT and SVAE barely improve the performance for few-shot intents. They only work well with sufficient training data. The utterances generated by these two models are almost the same as the few-shot examples. The performance improved by EDA is also limited since it only provides simple language transformation like insert and delete. Compared with CG-BERT that incorporates the pre-trained language model BERT, CLANG further improves the ability to generate utterances for few-shot intents with composed natural language generation.

From the ablation study illustrated in Table 3, removing the contrastive loss decreases the accuracy for few-shot intents and the harmonic mean. It shows that the contrastive loss regularizes the generation process and contributes to the downstream classification task.

| | Many-shot | Few-shot | H-Mean |
|---|---|---|---|
| | NLUED 1-shot | | |
| CLANG | $93.60 \pm 0.79$ | $22.03 \pm 6.10$ | $35.29 \pm 8.05$ |
| $-\mathcal{L}_v$ | $93.88 \pm 0.84$ | $21.76 \pm 6.44$ | $34.92 \pm 8.48$ |
| | NLUED 5-shot | | |
| CLANG | $93.29 \pm 0.86$ | $66.44 \pm 3.07$ | $77.56 \pm 2.05$ |
| $-\mathcal{L}_v$ | $92.94 \pm 0.72$ | $65.26 \pm 2.95$ | $76.64 \pm 2.06$ |

Table 3: Ablation study for removing the contrastive loss $\mathcal{L}_v$ from CLANG on NLUED.

## 3.5 Result Analysis

To further understand the proposed model, CLANG, result analysis and generation quality evaluation are provided in this section. We take the fold 7 of the NLUED dataset with the 5-shot setting as an example. It contains 16 novel intents with 5 examples per intent.

The intent in this paper is defined as a pair of a domain and an action. The domain or the action might be shared among the many-shot intents and the few-shot intents. The domain/action that exists in many-shot intents is named as a seen domain/action; otherwise, it is called a novel domain/action. To analyze how well our model performs on different few-shot intents, we split few-shot intents into four types: a novel domain with a seen action ($\text{Novel}_d$), a novel action with a seen domain ($\text{Novel}_a$), both domain and action are seen ($\text{Dual}_s$), both domain and action are novel ($\text{Dual}_u$). We compare our proposed model with CG-BERT on these different types. As illustrated in Table 4, CLANG consistently performs better than CG-BERT on all the types. The performance for intents with a seen action and a novel domain improves 20.90%. This observation indicates that our model is better at generalizing seen actions into novel domains.

| | Total | $\text{Novel}_d$ | $\text{Novel}_a$ | $\text{Dual}_s$ | $\text{Dual}_u$ |
|---|---|---|---|---|---|
| Number | 16 | 4 | 8 | 3 | 1 |
| CG-BERT | 58.76% | 47.76% | 60.43% | 67.34% | 63.16% |
| CLANG | 67.88% | 68.66% | 62.58% | 75.51% | 84.21% |
| +Improve | 9.12% | 20.90% | 2.15% | 8.17% | 21.05% |

Table 4: Accuracies on different types of few-shot intents.

As a few-shot natural language generation model, diversity is a very important indicator for quality evaluation. We compare the percentage of unique utterances generated by CLANG with CG-BERT. In CG-BERT, the top 20 results are generated for each intent by sampling the hidden variable for once. There are 257 unique sentences out of 320 utterances (80.3%). In CLANG, the top 30 results for each intent are generated by sampling the latent variables for once. We got 479 unique sentences out of 480 utterances (99.8%), which is much higher than CG-BERT.

Several generation examples are shown in Table 5. CLANG can generate good examples (indicated by G) that have new slots values (like time, place, or action) not existing in the few-shot examples (indicated by R). For example, G1 has a new time slot and G5 has a new action. Bad cases (indicated by B) like B1 and B5 fill in the sentence with improper slot values. CLANG can also learn sentences from other intents. For instance, G3 transfer the expression in R3 from "Recommendation Events" to "recommendation movies". However,

B4 fails to transfer R4 into the movie domain.

| Intent: **Alarm Query** |
| --- |
| R1: what time is my alarm set for tomorrow morning |
| G1: what time is my alarm set for this weekend |
| B1: how much my alarm set for tomorrow morning |
| R2: i need to set an alarm how many do i have set |
| G2: do i have an alarm set for tomorrow morning |
| B2: how many emails i have set |
| Intent: **Recommendation Movies** |
| R3 (events): is there anything to do tonight |
| G3 (movies): are there anything movie tonight |
| R4 (events): what bands are playing in town this weekend |
| B4 (movies): what bands are playing in town this weekend |
| Intent: **Takeaway Order** |
| R5: places with pizza delivery near me |
| G5: search for the delivery near me |
| B5: compose a delivery near me |
| G6: places with pizza delivery near my location |
| B6: places with pizza delivery near my pizza |

Table 5: Generation examples from CLANG. R are real examples in the few-shots, G are good generation examples and B are bad cases.

A case study is further provided for the Alarm Query intent with human evaluation. There are 121 unique utterances generated in total. As shown in Table 6, 80.99% are good examples and 19.01% are bad cases. Good cases mainly come from four types: Add/Delete/Replacement which provides simple data augmentation; New Time slot that has a new time slot value; New Question that queries alarm in new question words; Combination that combines two utterances together. Bad cases either come from a wrong intent (intents related to Query or Alarm) or use a wrong question word.

| Type | Count | Percent |
| --- | --- | --- |
| Add/Delete/Replacement | 33 | 27.27% |
| New Time slot | 30 | 24.79% |
| New Question | 28 | 23.14% |
| Combination | 7 | 5.79% |
| Total Good Cases | 98 | 80.99% |
| Wrong Intent (Query) | 10 | 8.26% |
| Wrong Intent (Alarm) | 7 | 5.79% |
| Wrong Question | 6 | 4.96% |
| Total Bad Cases | 23 | 19.01% |

Table 6: Generation case study for the Alarm Query intent.

## 4 Related Work

**Generative Data Augmentation for SLU** Generative data augmentation methods alleviates the problem of lacking data by creating artificial training data with generation models. Recent works (Wei and Zou, 2019; Malandrakis et al., 2019; Yoo et al., 2019) have explored this idea for SLU tasks like intent detection. Wei and Zou (2019) provide data augmentation ability for natural language with simple language transformation rules like insert, delete and swap. Malandrakis et al. (2019) and Yoo et al. (2019) utilize variational autoencoders (Kingma and Welling, 2013) to generate training data for SLU tasks. Malandrakis et al. (2019) investigates templated-based text generation model to augment the training data for intelligent artificial agents. Yoo et al. (2019) generate fully annotated utterances to alleviate the data scarcity issue in spoken language understanding tasks. These models utilize LSTM as encoders (Hochreiter and Schmidhuber, 1997) with limited model capacity. Xia et al. (2020) provide the first work that combines CVAE with BERT to generate utterances for generalized few-shot intent detection.

Recently, large-scale pre-trained language models are proposed for conditiaonal text generation tasks (Dathathri et al., 2019; Keskar et al., 2019), but they are only evaluated by human examination. They are not aiming at improving downstream classification tasks in low-resource conditions.

**Contrastive Learning in NLP** Contrastive learning that learns the differences between the positive data from the negative examples has been widely used in NLP (Gutmann and Hyvärinen, 2010; Mikolov et al., 2013; Cho et al., 2019). Gutmann and Hyvärinen (2010) leverage the Noise Contrastive Estimation (NCE) metric to discriminate the observed data from artificially generated noise samples. Cho et al. (2019) introduce contrastive learning for multi-document question generation by generating questions closely related to the positive set but far away from the negative set. Different from previous works, our contrastive loss learn a positive example against a negative example together with label information.

## 5 Conclusion

In this paper, we propose a novel model, Composed Variational Natural Language Generator (CLANG) for few-shot intents. An intent is defined as a combination of a domain and an action to build connections between existing intents and few-shot intents. CLANG has a bi-latent variational encoder that uses two latent variables to learn disentangled semantic features corresponding to different parts in the intent. These disentangled features are composed together to generate training examples for few-shot intents. Additionally, a contrastive loss is

adopted to regularize the generation process. Experimental results on two real-world intent detection datasets show that our proposed method achieves state-of-the-art performance for GFSID.

## Acknowledgments

## References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

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

Yun-Nung Chen, Dilek Hakkani-Tür, Gökhan Tür, Jianfeng Gao, and Li Deng. 2016. End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding. In *INTERSPEECH*, pages 3245–3249.

Woon Sang Cho, Yizhe Zhang, Sudha Rao, Asli Celikyilmaz, Chenyan Xiong, Jianfeng Gao, Mengdi Wang, and Bill Dolan. 2019. Contrastive Multi-document Question Generation. *arXiv e-prints*, page arXiv:1911.03047.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: a simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *arXiv preprint arXiv:1606.08415*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Matthew B Hoy. 2018. Alexa, siri, cortana, and more: An introduction to voice assistants. *Medical reference services quarterly*, 37(1):81–88.

Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. 2009. Understanding user's query intent with wikipedia. In *WWW*, pages 471–480.

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-Volume 70*, pages 1587–1596. JMLR. org.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3581–3589. Curran Associates, Inc.

Nikolaos Malandrakis, Minmin Shen, Anuj Goyal, Shuyang Gao, Abhishek Sethi, and Angeliki Metallinou. 2019. Controlled text generation for data augmentation in intelligent artificial agents. *arXiv preprint arXiv:1910.03487*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

IBM Watson Assistant. 2017. Defining intents. In *https://cloud.ibm.com/docs/assistant-icp?topic=assistant-icp-intents*.

Jason W Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Congying Xia, Chenwei Zhang, Hoang Nguyen, Jiawei Zhang, and Philip Yu. 2020. Cg-bert: Conditional text generation with bert for generalized few-shot intent detection. *arXiv preprint arXiv:2004.01881*.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S Yu. 2018. Zero-shot user intent detection via capsule neural networks. *arXiv preprint arXiv:1809.00385*.

Yongqin Xian, Bernt Schiele, and Zeynep Akata. 2017. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4582–4591.

Pawel Swietojanski Xingkun Liu, Arash Eshghi and Verena Rieser. 2019. Benchmarking natural language understanding services for building conversational agents. In *Proceedings of the Tenth International Workshop on Spoken Dialogue Systems Technology (IWSDS)*, pages xxx–xxx, Ortigia, Siracusa (SR), Italy. Springer.

Hu Xu, Bing Liu, Lei Shu, and P Yu. 2019. Open-world learning and application to product classification. In *The World Wide Web Conference*, pages 3413–3419. ACM.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *ASRU*, pages 78–83.

Wenpeng Yin. 2020. Meta-learning for few-shot natural language processing: A survey. *arXiv preprint arXiv:2007.09604*.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*.

Kang Min Yoo, Youhyun Shin, and Sang-goo Lee. 2019. Data augmentation for spoken language understanding via joint variational generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7402–7409.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*.