CG-BERT: Conditional Text Generation with BERT for Generalized Few-shot Intent Detection

Congying Xia[1], **Chenwei Zhang**[2], **Hoang Nguyen**[1], **Jiawei Zhang**[3], **Philip Yu**[1]

[1]University of Illinois at Chicago, IL, USA; [2]Amazon, WA, USA;
[3]Florida State University, FL, USA
{cxia8,hnguy7,psyu}@uic.edu;
cwzhang@amazon.com; jzhang@ifmlab.org

Abstract

In this paper, we formulate a more realistic and difficult problem setup for the intent detection task in natural language understanding, namely Generalized Few-Shot Intent Detection (GFSID). GFSID aims to discriminate a joint label space consisting of both existing intents which have enough labeled data and novel intents which only have a few examples for each class. To approach this problem, we propose a novel model, Conditional Text Generation with BERT (CG-BERT). CG-BERT effectively leverages a large pre-trained language model to generate text conditioned on the intent label. By modeling the utterance distribution with variational inference, CG-BERT can generate diverse utterances for the novel intents even with only a few utterances available. Experimental results show that CG-BERT achieves state-of-the-art performance on the GFSID task with 1-shot and 5-shot settings on two real-world datasets.

1 Introduction

Intent Detection (ID) is a key task in spoken dialogue systems, which aims at understanding the intents behind users' inputs (Tur and De Mori, 2011). In real-world applications, such as Google Assistant, detecting the intents in users' utterances is crucial for downstream tasks, such as dialogue management and dialog state tracking.

In this ever-changing digital world, intelligent assistants need to have the ability to adapt to customers' new requests promptly. State-of-the-art intent detection models (Haihong et al., 2019; Zhang et al., 2018; Goo et al., 2018) capitalize on large amounts of labeled data to train supervised deep learning classification models. These models can be trained solely based on the existing intents, but they generalize poorly to novel intents. It is difficult and labor-intensive to collect large-scale, high-quality annotations on novel intents and re-train the whole model.

Inspired by human's ability to adapt existing knowledge to new concepts quickly with only a few examples, few-shot learning (FSL) (Li et al., 2006) has drawn a lot of attention recently. FSL approaches (Vinyals et al., 2016) are expected to discriminate the novel classes from each other with only a few examples, namely, few-shots. However, this formulation offers no incentive to maintain a globally consistent label space with the existing classes. From a practical point of view, we would like the model to incorporate these novel classes which only have a few labeled examples into the label space of existing classes that have enough labeled data. This ability is crucial for the intent detection task in practice, as it relaxes the assumption that we only need to detect the novel intents.

In this work, we propose a more difficult yet realistic problem setup for intent detection in the low-resource scenarios, which is called Generalized Few-Shot Intent Detection (GFSID). As shown in the left part of Figure 1, GFSID aims to correctly classify utterances that belong to both existing and novel intents. This terminology is adopted from the Generalized Few-shot Learning (Shi et al., 2019) which is an extension setup from FSL. The performance of FSL models deteriorates drastically in discriminating the joint label space, despite their good performance on novel label space (Shi et al., 2019). Compared to FSL, GFSID is a much more challenging task since the model has a bias on the existing classes over the novel classes and prefers to predict the test samples as the existing classes.

We propose to ease the scarce annotation problem in the GFSID task by generating more utterances for the novel intents and convert the generalized few-shot problem into a supervised classification problem. As illustrated in Figure 1, we model the utterance distributions associated with
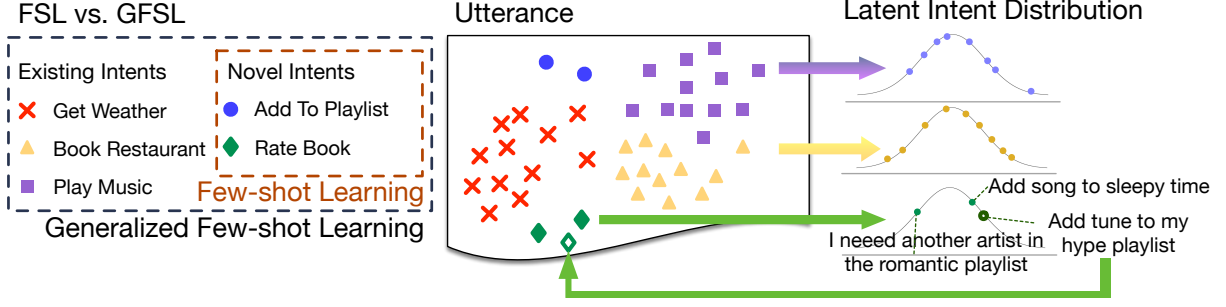
Figure 1: Compared to FSL which only needs to discriminate among the novel intents, GFSID needs to discriminate all the intents including the existing ones and the novel ones. We model the distribution for the diverse utterances associated with one intent and generate new utterances by sampling from this learned distribution. We augmented the training dataset with the generated examples for the novel intents, and convert the task into a supervised classification task.

the intents and generate new utterances for novel intents through sampling from this learned distribution. However, it's difficult to learn a good distribution with only a few examples. To transfer the knowledge in a large amount of unlabeled data, we utilize the powerful pre-trained language model BERT (Devlin et al., 2018) to learn such a distribution. Since the distribution learned through BERT is unregulated for different intents, we propose to adopt the idea from Conditional Variational Auto-Encoder (CVAE) (Kingma et al., 2014) to add a latent space mapping for BERT and regularize the BERT feature space to form a unit Gaussian distribution conditioned on a certain intent.

In this paper, we propose a conditional text generation model, Conditional Text Generation with BERT (CG-BERT), which incorporates CVAE in BERT to model a distribution over diverse utterances having the same intent. Through the proposed model, we are able to generate more utterances for the novel intents and augment the training dataset. We utilize the state-of-the-art text classification model BERT to do intent detection with the augmented dataset in a supervised way.

To summarize, our main contributions are:

- We propose a more realistic and challenging problem setup in low-resource conditions, Generalized Few-Shot Intent Detection (GFSID), which aims to discriminate a joint label space consisting of both existing intents which have enough labeled data and novel intents which only have a few examples for each class.

- A novel conditional text generation model, CG-BERT, is proposed to solve the GFSID task. CG-BERT uses a latent variable to model the probability distribution of diversely expressed

utterances that belongs to a certain intent. It is able to generate pseudo labeled examples for the novel intents and alleviate the scarce annotation problem for GFSID.

- The experiments conducted for the proposed GFSID task show the effectiveness of our proposed model on two real-word intent detection datasets.

2 Problem Formulation

In this section, we give the definition for the generalized few-shot intent detection (GFSID) task. Given an existing intent set $\mathcal{Y}_{ex}$, a large amount of labeled examples $\mathcal{D}_{ex} = \{(x_i, y_i), i = 1, 2, ..., |\mathcal{D}_{ex}|\}$ with $y_i \in \mathcal{Y}_{ex}$ are available. The novel intent set can be denoted as $\mathcal{Y}_{novel}$, where each intent has K examples $\mathcal{D}_{novel} = \bigcup_n \{(x_{n,k}, y_n)\}_{k=1}^K$. These two intent sets are disjointed, i.e., $\mathcal{Y}_{ex} \cap \mathcal{Y}_{novel} = \varnothing$; the overall intent space can be denoted as $\mathcal{Y}_{joint} = \mathcal{Y}_{ex} \cup \mathcal{Y}_{novel}$.

Generalized few-shot intent detection intends to classify a given utterance not only as one of the existing intents but also as the novel intents. Formally, given a new query utterance x, the GFSID task aims at inferring the most likely intent of x, i.e.,

$$\hat{y} = \arg\max_{y \in \mathcal{Y}_{joint}} p(y|x, \mathcal{D}_{ex}, \mathcal{D}_{novel}). \quad (1)$$

Compared to the traditional few-shot classification task which only needs to separate the few-shot classes $\mathcal{Y}_{novel}$, GFSID is much more technically challenging and requires the discrimination of a much larger label space $\mathcal{Y}_{joint}$ instead.

3 The Proposed Model

Conditional Text Generation with BERT (CG-BERT) is proposed to model the distribution of diverse utterances with a given intent. It is able

to generate more utterances for the novel intent through sampling from the learned distribution.

The overall framework of CG-BERT is illustrated in Figure 2. Particularly, CG-BERT adopts the CVAE framework and incorporates BERT into both the encoder and the decoder. Formally, the encoder encodes the utterance x and its intent $y \in \mathcal{Y}_{joint}$ together into a latent variable z and models the posterior distribution $p(z|x, y)$, where y is the condition in the CVAE model. The decoder decodes z and the intent y together to reconstruct the input utterance x. To generate new utterances for an novel intent $y \in \mathcal{Y}_{novel}$, we sample the latent variable z from a prior distribution $p(z|y)$ and utilize the decoder to decode z and $y \in \mathcal{Y}_{novel}$ into new utterances.

3.1 Input Representation

The input representation follows that of BERT (Devlin et al., 2018). In order to encode both the intent and the utterance, the input in our model is a pair of intent and utterance text sentences packed together.

Given an intent $y = (L_1, L_2, ..., L_{T_1})$ with T_1 words and an utterance $x = (X_1, X_2, ..., X_{T_2})$ with T_2 words, we add special start-of-sequence ([CLS]) token at the beginning of the input and a special end-of-sequence ([SEP]) token at the end of each sentence. The first intent sentence, ([CLS], L_1, L_2, ..., L_{T_1}, [SEP]), is referred to as S_1 and the second utterance sentence, (X_1, X_2, ..., X_{T_2}, [SEP]) is named as S_2. As shown in Figure 2, S_1 and S_2 are concatenated together as the whole input. [SEP] not only marks the sentence boundary, but also is used for the model to learn when to terminate the decoding process. [CLS] is used as the representation for the whole input and we encode the embeddings for [CLS] to the latent variable z.

Texts are tokenized into subword units by Word-Piece (Wu et al., 2016). As illustrated in Figure 2, three types of embedddings will be obtained for each token: token embeddings, position embeddings (Vaswani et al., 2017), and segment embeddings (Devlin et al., 2018) which identifies the intent and the utterance. The input representation of a given token is constructed by summing these three embeddings and represented as $\mathbf{H}^0 = [\mathbf{h}_1^0, ..., \mathbf{h}_T^0]$ with a total length of T tokens.

3.2 The Encoder

As shown in Figure 2, the encoder in our model encodes the input (both the intent and the utterance) into a latent variable that models the distribution of

diverse utterances for a given intent. To obtain deep bidirectional context information, we utilize multiple bidirectional Transformer encoders (Vaswani et al., 2017), namely Transformer blocks in this paper, to encode the input. These Transformer blocks not only extract the semantic information for the input, but also models the attention between the intent tokens and the utterance tokens.

The BERT-base model is used as the backbone of CG-BERT to incorporate the information pretrained from a large amount of unlabeled data. As BERT-base has 12 transformer blocks, we utilize the first 6 transformer blocks in the encoder and the last 6 blocks are involved in the decoder.

The input representation $\mathbf{H}^0 = [\mathbf{h}_1^0, ..., \mathbf{h}_T^0]$ is encoded into contextual layer representations through the transformer blocks. For each layer, the layer representation $\mathbf{H}^l = [\mathbf{h}_1^l, ..., \mathbf{h}_T^l]$ is computed with an l-layer Transformer block $\mathbf{H}^l = \text{Transformer}_l(\mathbf{H}^{l-1})$, $l \in \{1, 2, \cdots, 6\}$. In each Transformer block, multiple self-attention heads are used to aggregate the output vectors of the previous layer. For the l-th Transformer layer, the output of a self-attention head $\mathbf{A}_l$ is computed via:

$$\begin{aligned} \mathbf{Q} &= \mathbf{H}^{l-1}\mathbf{W}_Q^l, \\ \mathbf{K} &= \mathbf{H}^{l-1}\mathbf{W}_K^l, \\ \mathbf{V} &= \mathbf{H}^{l-1}\mathbf{W}_V^l, \end{aligned} \qquad (2)$$

$$\mathbf{A}_l = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \qquad (3)$$

where the output of the previous layer $\mathbf{H}^{l-1} \in \mathbb{R}^{T \times d_h}$ is linearly projected to a triple of queries, keys and values parameterized by matrices $\mathbf{W}_l^Q, \mathbf{W}_K^l, \mathbf{W}_V^l \in \mathbb{R}^{d_h \times d_k}$. Following BERT, the embeddings for the [CLS] token in the 6-th transformer block $\mathbf{h}_1^6$ is regarded as sentence-level representation.

By modeling the true distribution $p(z|x, y)$ using a known distribution that is easy to sample from (Kingma et al., 2014), we encode the sentence-level representation $\mathbf{h}_1^6$ into a latent variable z whose prior distribution $p(z|y)$ is a multivariate standard Gaussian distribution. We use the reparametrization trick (Kingma and Welling, 2013) to generate the latent vector z. The encoder predicts the parameters μ and σ in the Gaussian distribution such that the proxy posterior $q(z|x, y) = \mathcal{N}(\mu, \sigma)$:
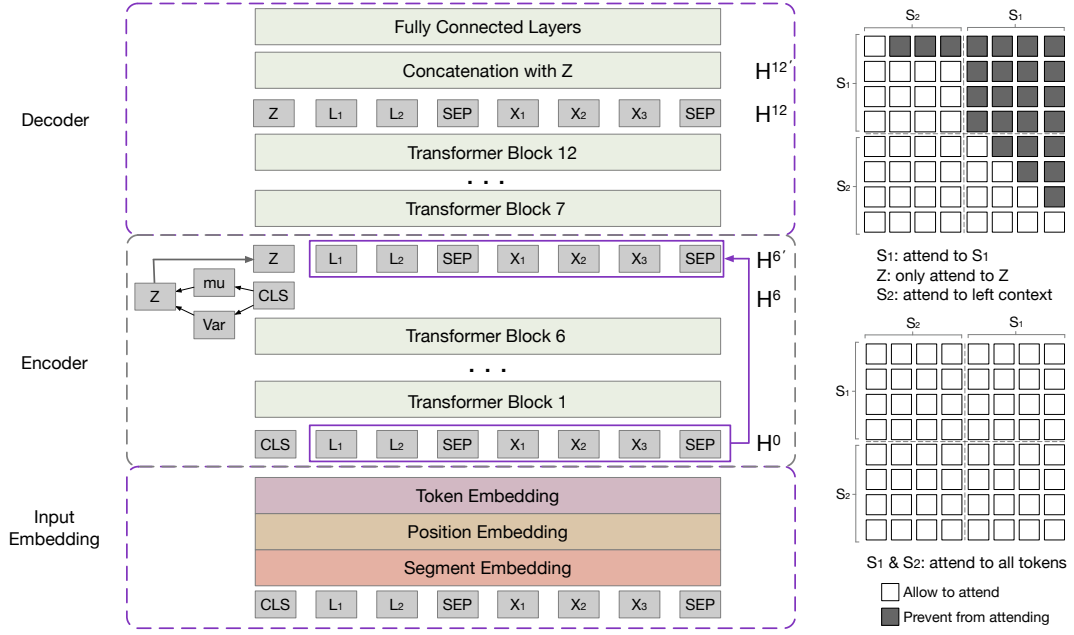
Figure 2: The overall framework of CG-BERT: Conditional Text Generation with BERT.

$$\mu = \mathbf{h}_1^6 \mathbf{W}_\mu + b_\mu,$$
$$\log(\sigma^2) = \mathbf{h}_1^6 \mathbf{W}_\sigma + b_\sigma, \qquad (4)$$
$$z = \mu + \sigma\varepsilon,$$

where μ and $\log(\sigma^2)$ are projected from $\mathbf{h}_1^6$ through parameters $\mathbf{W}_\mu \in \mathbb{R}^{d_h \times d_h}$, $\mathbf{W}_\sigma \in \mathbb{R}^{d_h \times d_h}$, $b_\mu \in \mathbb{R}^{d_h}$, $b_\sigma \in \mathbb{R}^{d_h}$, $\varepsilon \in \mathcal{N}(0,1)$ are used to sample z.

3.3 The Decoder

The decoder aims to reconstruct the input utterance x using the latent variable z and the intent y. As shown in Figure 2, we add a residual connection from the input representation $\mathbf{H}^0$ to the input of the decoder $\mathbf{H}^{6'}$. The input of the decoder, $\mathbf{H}^{6'} = [z, \mathbf{h}_2^0, ..., \mathbf{h}_T^0]$, is constructed by concatenating the latent variable z and the input representation $\mathbf{H}^0$ which represents the input tokens. The input representation for the [CLS] token, $\mathbf{h}_1^0$, is ignored since its position is taken by z.

The input for the decoder $\mathbf{H}^{6'}$ is fed into the last six transformer blocks of BERT-base, as shown in Figure 2. To reconstruct the input utterance, the decoder is trained to generate the utterance in a left-to-right manner, since we can only use the left context to predict the next token. The original BERT model is trained on a large corpus to predict a randomly masked token given its contexts in the sentence. We try to leverage the conditional prediction ability and modify it for text generation in a left-to-right manner.

Inspired by the Unified Language model (Dong et al., 2019), which utilizes specific self-attention masks to control what context the prediction conditions on, we adopt the attention mask which helps the transformer blocks fit into the conditional text generation task. Instead of applying the whole bidirectional attention to the input, a mask matrix $\mathbf{M} \in \mathbb{R}^{T \times T}$ is added to determine whether a pair of tokens can be attended to each other. We update the calculation for the attentions in transformer blocks from Equation 3 to:

$$\mathbf{A}_l = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} + \mathbf{M}\right)\mathbf{V}, \qquad (5)$$

where

$$\mathbf{M}_{ij} = \begin{cases} 0, & \text{allow to attend}; \\ -\infty, & \text{prevent from attending}. \end{cases} \qquad (6)$$

The proposed attention mask matrix for the transformer blocks in the decoder is shown in the upper right corner of Figure 2. To keep the transformers in the encoder having the same structure, we apply the bidirectional self-attention mask for the encoder as shown in the lower right corner of Figure 2. The bidirectional self-attention mask allows all the tokens in S_1 and S_2 to attend to all the tokens.

In the attention mask for the decoder, the tokens in the first intent sentence S_1 can only attend to all the tokens in S_1, while the tokens in the second sentence S_2 can attend to both S_1 and all the left tokens in S_2. With this attention matrix, we are able to control the decoder to generate the output from left to right conditioned on the given intent.

A special token here is the first token which holds the latent variable z, it is only allowed to attend to itself. This is due to the vanishing latent variable problem (Zhao et al., 2017) when adapting VAE/CVAE to natural language generators. Previous works (Zhao et al., 2017) reported the problem that the LSTM decoder tends to ignore the latent variable. We met the same issue when applying the variational autoencoder with transformers. To alleviate this problem, we only allow the latent variable z to attend to itself to avoid it been overwhelmed by the information of other tokens.

The output of the 12-th transformer block in the decoder is represented as $\mathbf{H}^{12} = [\mathbf{h}_1^{12}, \mathbf{h}_2^{12}, ..., \mathbf{h}_T^{12}]$, where $\mathbf{h}_1^{12}$ is the embeddings for the latent variable z. To further increase the impact of z and alleviate the vanishing latent variable problem, we concatenate the embedings of z with all the tokens: $\mathbf{H}^{12'} = [\mathbf{h}_1^{12} \| \mathbf{h}_1^{12}, \mathbf{h}_2^{12} \| \mathbf{h}_1^{12}, ..., \mathbf{h}_T^{12} \| \mathbf{h}_1^{12}]$. Two fully-connected layers with a layer normalization (Ba et al., 2016) are applied to get the final representation:

$$\mathbf{H}^f = g(f(f(\mathbf{H}^{12'}\mathbf{W_1} + b_1)\mathbf{W_2} + b_2)), \quad (7)$$

where $\mathbf{W}_1 \in \mathbb{R}^{2d_h \times d_h}$, $\mathbf{W}_2 \in \mathbb{R}^{d_h}$, f is a Gelu (Hendrycks and Gimpel, 2016) activation function and g is for layer normalization. The embeddings in $\mathbf{H}^f$ at position at position t is used to predict the next token at position $t + 1$:

$$p(X_{t+1}) = f(\mathbf{H}_t^f \mathbf{W_e}^\top + b_e), \quad (8)$$

where $\mathbf{W_e} \in \mathbb{R}^{V \times d_h}$ is the token embedding in the input representation, $b_e \in \mathbb{R}^V$, V is vocabulary size.

3.4 Loss Function

We train this model to maximize $\log p(x|y)$, the conditional log-likelihood of the utterance x given an intent y, which involves a marginalization over the latent variable z. As proposed in (Kingma et al., 2014), CVAE can be effectively trained with the Stochastic Gradient Variational Bayes framework by maximizing the evidence lower bound (ELBO) of the log-likelihood. The loss function for the proposed model is:

$$\mathcal{L} = -KL\left[q\left(z|x,y\right), p\left(z|y\right)\right] + \\ E_{q(z|x,y)}\left[\log p\left(x|z,y\right)\right], \quad (9)$$

which consists of the KL divergence loss and the reconstruction loss.

3.5 Generating Utterances for Novel Intents

The distributions of the intents are learned after training the CG-BERT. Through sampling from these distributions, we are able to generate utterances for novel intents. As shown in Figure 2, we use the modules in the purple boxes (Input Embedding and Decoder) to generate new utterances. For a given novel intent y, an intent sentence S_1 is fed into the Input Embedding module to get the input representation $\mathbf{H}^0$. The input for the decoder is constructed through $\mathbf{H}^{6'} = [z, \mathbf{h}_2^0, ..., \mathbf{h}_{T_1}^0]$, where z is sampled from a multivariate standard Gaussian distribution instead of encoded from the input. With the decoder, the new utterance is generated sequentially from left to right.

To generate more utterances, we sample z for s times, and for each z, we get the top 20 results using beam search. To increase diversity, we will delete the generated sentences which already exist in the training data. The procedure for generating new utterances for the novel intents is illustrated as:

Algorithm 1 Text Generation for Novel Intents

1: **procedure** TEXT GENERATION(y, s, k)
2:     **for** y all novel intents **do**
3:         **for** s sample_times **do**
4:             $z_s \in \mathcal{N}(0,1)$
5:             $\mathcal{G}_y$.add(top$k(decode(z_s, y))) \triangleright \mathcal{G}_y$
    is an utterance set with intent y.
6:         $\mathcal{G}_y = \mathcal{G}_y$.remove($\mathcal{D}_{novel}$) $\triangleright$ Remove the provided few-shot examples in $\mathcal{G}_y$.
7:         $\mathcal{D}_{ge}$.add(y, $\mathcal{G}_y$) $\triangleright \mathcal{D}_{ge}$ is the generated dataset.
8:     Return $\mathcal{D}_{ge}$

The generated dataset $\mathcal{D}_{ge}$ are merged with the original training data $\mathcal{D}_{seen}, \mathcal{D}_{novel}$ to do intent detection with fine-tuning on BERT.

4 Experiments

4.1 Datasets

To demonstrate the effectiveness of our proposed model, we evaluate CG-BERT on two real-word datasets for the generalized few-shot intent detection task: SNIPS-NLU (Coucke et al., 2018) and NLU-Evaluation-Data (NLUED) (Xingkun Liu and Rieser, 2019). These two datasets were collected to benchmark the performance of natural language understanding services offering customized solutions. The statistical information of these two datasets is shown in Table 2.

SNIPS-NLU[1] Following Xia et al. (2018), we select two intents (RateBook and AddToPlaylist) as

---

[1] https://github.com/snipsco/nlu-benchmark/

	Overall	Seen	Novel	H-Mean	Overall	Seen	Novel	H-Mean
	SNIPS-NLU 1-shot				SNIPS-NLU 5-shot			
IN+	18.02 ± 2.92	20.16 ± 4.20	12.38 ± 8.34	13.44 ± 7.15	28.91 ± 1.44	30.34 ± 1.42	25.15 ± 3.49	27.39 ± 2.21
BERT-SMOTE	74.65 ± 0.80	96.24 ± 0.35	17.66 ± 2.71	29.77 ± 3.94	83.56 ± 1.42	95.84 ± 1.08	51.13 ± 4.04	66.61 ± 3.54
BERT	83.42 ± 1.18	98.20 ± 0.06	44.42 ± 4.35	57.74 ± 7.50	93.80 ± 1.74	98.34 ± 0.10	81.82 ± 6.16	89.22 ± 3.74
SVAE	83.65 ± 1.55	$\mathbf{98.24 \pm 0.09}$	45.15 ± 5.54	61.67 ± 5.11	93.88 ± 1.11	$\mathbf{98.34 \pm 0.06}$	82.10 ± 4.06	89.49 ± 2.47
BERT-PN+	83.83 ± 4.13	92.66 ± 4.49	60.52 ± 7.58	72.99 ± 5.97	93.23 ± 0.94	95.96 ± 1.13	86.03 ± 2.00	90.71 ± 1.19
CG-BERT	$\mathbf{88.49 \pm 1.55}$	98.13 ± 0.15	$\mathbf{63.04 \pm 5.49}$	$\mathbf{76.65 \pm 4.24}$	$\mathbf{95.16 \pm 1.12}$	98.30 ± 0.17	$\mathbf{86.89 \pm 4.05}$	$\mathbf{92.20 \pm 2.32}$
	NLUED 1-shot				NLUED 5-shot			
BERT-SMOTE	52.44 ± 0.96	85.89 ± 1.26	3.35 ± 1.37	6.40 ± 2.51	64.17 ± 1.25	86.03 ± 1.32	32.11 ± 1.89	46.74 ± 2.06
BERT	72.11 ± 1.27	94.00 ± 0.93	7.88 ± 3.28	14.39 ± 5.66	83.33 ± 0.74	$\mathbf{94.12 \pm 0.89}$	51.69 ± 3.19	66.67 ± 2.51
SVAE	72.22 ± 1.07	93.80 ± 0.70	8.88 ± 3.66	16.01 ± 6.06	83.54 ± 0.88	93.60 ± 0.63	54.03 ± 3.91	68.42 ± 3.06
BERT-PN+	71.49 ± 2.40	81.24 ± 2.76	18.95 ± 4.42	30.67 ± 5.53	78.29 ± 2.62	83.41 ± 2.62	60.28 ± 4.19	69.93 ± 3.49
CG-BERT	$\mathbf{75.29 \pm 1.75}$	$\mathbf{94.01 \pm 0.70}$	$\mathbf{20.39 \pm 5.77}$	$\mathbf{33.12 \pm 7.92}$	$\mathbf{85.48 \pm 0.88}$	93.80 ± 0.60	$\mathbf{61.06 \pm 4.29}$	$\mathbf{73.88 \pm 3.10}$

Table 1: Generalized few shot experiments with 1-shot/5-shot setting on SNIPS-NLU and NLUED.

emerging intents (the few-shot classes), while the other five intents are regarded as existing intents. For the GFSID task, we need to do classification among the whole 7 intents. We randomly choose 80% of the whole data as the training data and 20% as the test data. For the few-shot intents, we randomly sample 1 or 5 examples as the few-shots. **NLUED** Following Xingkun Liu and Rieser (2019), a sub-corpus of 11, 036 utterances covering all the 64 intents are used. We randomly choose 16 intents as the few-shot ones. We perform the generalized intent detection task among the whole 64 intents over 10-Folds[2].

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 2: Data Statistics for SNIPS-NLU and NLUED. #Few-shot examples are excluded in the #Training Examples. For NLUED, we report the statistics in KFold_1.

4.2 Baselines

To the best of our knowledge, we are the first one to study the GFSID task. Following Shi et al. (2019), we extend two state-of-the-art FSL methods to the GFSID setup as our baselines. From the data augmentation perspective, our model is an over-sampling model which samples sentence level examples from latent distributions. To show the effectiveness of our proposed model, we also compare it with other over-sampling methods.

Five baselines are considered in total: 1) IN+. Induction Networks (Geng et al., 2019) (IN) is a state-of-the-art few-shot text classification model.

It can take any-way any-shot inputs, hence we can easily apply a readily trained IN model to solve the GFSL task. 2) BERT-PN+. Prototypical Networks (Snell et al., 2017) (PN) is a simple but effective FSL model. To provide a fair comparison, we use BERT as the encoder in PN and fine-tune BERT together with the PN model. 3) BERT. BERT (Devlin et al., 2018) is the state-of-the-art text classification model. We simply over sample the few-shots by copying the few-shot examples to the maximum training examples per class and use the over-sampled dataset to do intent detection with fine-tuning on BERT. 4) BERT-SMOTE. SMOTE (Chawla et al., 2002) is an over-sampling method that samples features based on the nearest neighbors. BERT-SMOTE uses fixed pre-trained BERT features. A fully connected layer is used for intent detection with the over-sampled features. 5) SVAE. SVAE (Bowman et al., 2015) is a text generation model that utilizes Variational Auto-Encoder (VAE) to model the distribution of the utterances. It can generate similar sentences by sampling from the posterior or get intermediate sentences between two sentences.

4.3 Implementation Details

We use BERT-base and Adam optimizer (Kingma and Ba, 2014) for all the experiments. For CG-BERT, the hidden dimension d_h is 768 and the batch size is 16. CG-BERT is trained with learning rate equals 1e-5 in 100 epochs and each epoch has 1000 steps. We sample the latent variable $s = 10$ times and choose the top $k = 20$ utterances when generating new utterances. For fine-tuning on BERT, we set batch size as 32, learning rate as 2e-5 and the number of the training epochs as 3. For the FSL baselines, we train these models in the episode training method. For SNIPS-NLU, we train these models in 3-way for 1000 episodes. For

---

[2]https://github.com/xliuhw/NLU-Evaluation-Data

NLUED, we train these models in 10-way for 1500 episodes.

4.4 Experiment Results

The generalized few-shot intent detection results on two datasets with 1-shot/5-shot settings are reported in Table 1. In addition to the overall accuracy (Overall) measured on all intents, we follow the convention in GFZL (Xian et al., 2017) and report the accuracy on seen intents (Seen), the accuracy on novel intents (Novel) together with their harmonic mean (H-Mean). We report the average and the standard deviation for SNIPS-NLU over 5 runs. The results on NLUED are reported over 10 Folds.

On both two datasets with 1-shot and 5-shot settings, our proposed model CG-BERT achieves state-of-the-art performance on overall accuracy, novel accuracy, and harmonic mean. The improvement mainly stems from the high quality of the generated examples for novel intents, which leads to significantly increased novel accuracy and harmonic mean. CG-BERT achieves the best performance for the seen accuracy on the NLUED 1-shot setting, while obtaining comparable results with BERT on three other settings.

Among the extended few-shot learning models, BERT-PN+ achieves decent performance on the novel accuracy compared to other baselines while it sacrifices the performance on the seen intents. Compared to BERT-PN+, CG-BERT achieves good performance consistently by training the model on both seen intents and novel intents altogether. IN+ lacks the ability to extend from the FSL training to the GFSID setting since it uses a parametric method other than distance-based methods like PN to do classification. It performs poorly on the NLUED dataset ($< 20\%$ on the overall accuracy, thus not reported in Table 1).

For the text generation baseline, SVAE performs worse than CG-BERT on the novel intents due to the generated sentences are too similar to the input few-shots. BERT-SMOTE also does not perform well since it uses fixed BERT features and does over-sampling by exploring the interpolation among the few-shots in the feature space.

4.5 Evaluation of the Generated Utterances

We show some examples generated by CG-BERT for two novel intents in Table 3. The generated utterances for Rate Book show that CG-BERT is able to generate similar utterances to the few-shots by adding expressions like "can you" or replacing

words ("like" and "want"). The cases in Alarm Query are pretty diverse. It learns from other intents like "Calendar Query" that expressions like "show me" can be used for request information. It also generates utterances that are totally different from the few-shots, like "do i have any alarms".

Rate Book (Real examples)
rate the current essay 0
rate this textbook four stars
i d like to rate my beloved world two points
Rate Book (Generated Examples)
can you rate the current essay 0
rate that textbook four stars
i d want to rate my beloved world two points
Alarm Query (Real examples)
my alarms
what alarms do i have set
tell me what alarms are set for me
what are my alarms
Alarm Query (Generated Examples)
show me my alarms
what alarms do you have set
i need to see what alarms are there
do i have any alarms
remind me what alarms are set for me

Table 3: Generated examples for two novel intents. Rate Book is an novel intent in SNIPS-NLU and Alarm Query is an novel intent in NLUED.

There are also some bad cases when the intent is not fully conditioned as expected. For example, "am i going to need a jacket tomorrow" which has the intent of Query Weather, but generated for Alarm Query. In other cases, sentences are not well generated and may have syntax errors, like "alarms me are there any alarms for me".

We further evaluate the quality of the generated utterances both quantitatively and qualitatively. Quantitative metrics like the percentage of unique sentences and n-grams are used. Sentences are manually annotated for qualitative analysis. We use the Fold 7 of the NLUED dataset with the 5 shot setting as an example. There are 16 novel intents in total and each novel intent has 5 examples. We randomly sample the hidden variable z for once and get the top 20 generated sentences for each intent. After deduplication, we got 257 unique sentences out of 320, which is around 80%.

Quantitatively, we compare the average percentage of new n-grams in the generated sentences to a text generation model SVAE (Bowman et al., 2015). As shown in Table 4, CG-BERT is able to generate

	unigram	bigram	trigram
SVAE	3.57%	19.51%	22.90%
CG-BERT	31.74%	43.29%	47.36%

Table 4: The average percentage of new n-grams in the generated sentences.

more new n-grams which present the diversity of the generated sentences. Qualitatively, we asked human annotators to annotate the syntax quality (result: 4.3 out of 5) and intent coherency (result: 4.7 out of 5) of the generated utterances.

4.6 Visualization

We visualize the embedding space of the generated utterances using t-SNE for BERT-SMOTE and CG-BERT. As shown in Figure 3, the generated examples from CG-BERT for novel intents are diverse and well separated from the seen intents, while BERT-SMOTE tends to generate similar examples on the feature level as in Figure 4. SMOTE only generates new features within the few-shots, while CG-BERT is able to generate diverse examples beyond these five shots by transfer expressions from existing intents.
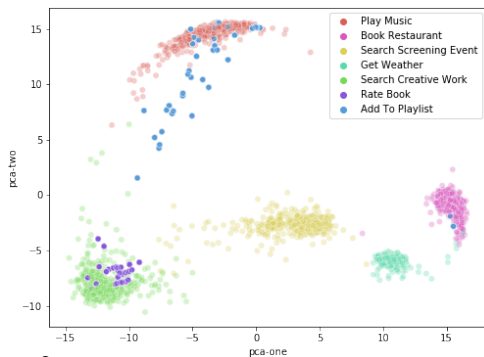

Figure 3: t-SNE visualization of the embedding space for SNIPS-NLU with generated utterances from CG-BERT.
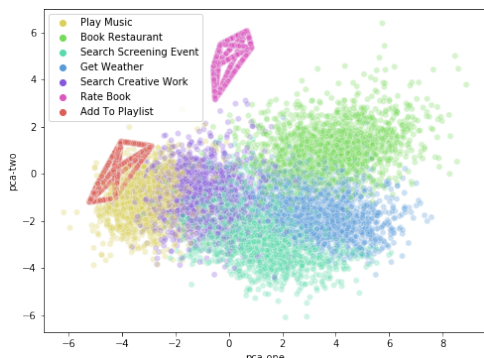

Figure 4: t-SNE visualization of the embedding space for SNIPS-NLU with over-sampled features from BERT-SMOTE.

5   Related Work

Few Shot Learning (Miller et al., 2000; Li et al., 2006) is a task that aims to learn classifiers for new classes with only a few training examples per class. Recent deep learning-based FSL approaches mainly fall into two categories: (1) metric-based approaches, such as Matching Networks (Vinyals et al., 2016) and Prototypical Networks (Snell et al., 2017), which tries to learn one generalizable metric to separate the classes based on the distance for all the tasks; (2) optimization-based approaches, e.g. Meta Network (Munkhdalai and Yu, 2017) and MAML (Finn et al., 2017), which aims to optimize model parameters based the gradients computed from few-shot examples.

Recently, some few-shot learning studies are presented with a special focus on few-shot text classification problems. Rios and Kavuluru (2018) develop a few-shot text classification model for multi-label text classification where there is a known structure of the label space. Geng et al. (2019) propose Induction Networks that use dynamic routing induction method to encapsulate the abstract class representation from a few examples. Xu et al. (2019) proposes an open-world learning model to deal with the unseen classes in the product classification problem. Zhang et al. (2019) is the first work which utilizes the pre-trained language model for few-shot text classification task.

Generalized FSL extends the setup from FSL, where the model is required to perform classification on the joint label space consisting of both previously seen and novel classes. This setup is not yet well-studied, especially for the intent detection task. There are only several works in the field of computer vision which try to solve this problem. Gidaris and Komodakis (2018) utilizes an attention-based weight generator for novel classes to extend the classifier from seen classes to the joint label space. Shi et al. (2019) incorporates inter-class relations using graph convolution to embed novel class representations into the same space with seen classes. Schonfeld et al. (2019) utilizes Variational Autoencoders to align the distributions learned from images and use side-information to construct latent features that contain the essential multi-modal information associated with unseen classes. Ye et al. (2019) proposes a learning framework, Classifier Synthesis Learning (CAS-TLE), which learns how to synthesize calibrated few-shot classifiers in addition to the classifiers of

seen classes, leveraging a shared neural dictionary between seen classes and novel classes.

Conditional Text Generation. As text generation is an important and difficult task in natural language processing, a lot of works have been presented to solve this task. Seq2seq models are standard encoder-decoder models widely used in text applications like machine translation (Luong et al., 2015). Variational Auto-Encoder (VAE) models are another important family (Kingma and Welling, 2013) and they consist of an encoder that maps each sample to a latent representation and a decoder that generates samples from the latent space. The advantage of these models is the variational component and its potential to add diversity to the generated data. They have been shown to work well for text generation (Bowman et al., 2015).

Conditional VAE (CVAE) (Kingma et al., 2014) is proposed to improve over seq2seq models for generating more diverse and relevant text. CVAE based models (Rajeswar et al., 2017; Zhao et al., 2017) incorporate stochastic latent variables that represent the generated text. Malandrakis et al. (2019) is the most relevant work, which investigates the use of text generation techniques to augment the training data for intelligent artificial agents. However, their generation method is template-based and they don't leverage the power of pre-trained language models (Devlin et al., 2018). This is the first work that employs a text generation model to solve the generalized few-shot learning problem.

6 Conclusions

A new task named Generalized Few-Shot Intent Detection is studied in this paper. It aims at discriminating a joint label space of existing intents with enough annotation and novel intents with only a few examples. A novel model, Conditional Text Generation with BERT, is proposed to solve this task by generating new utterances conditioned on a given novel intent. To the best of our knowledge, this is the first work that alleviates the scarce annotation problem in intent detection by generating sentence level examples. The proposed model achieves state-of-the-art performance on two-real word intent detection datasets.

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*.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

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*.

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*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3895–3904.

Spyros Gidaris and Nikos Komodakis. 2018. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana. Association for Computational Linguistics.

E Haihong, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5467–5471.

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

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.

Fei-Fei Li, Rob Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

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*.

Erik G Miller, Nicholas E Matsakis, and Paul A Viola. 2000. Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 464–471. IEEE.

Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2554–2563. JMLR. org.

Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Pal, and Aaron Courville. 2017. Adversarial generation of natural language. *arXiv preprint arXiv:1705.10929*.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3132–3142, Brussels, Belgium. Association for Computational Linguistics.

Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. 2019. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8247–8255.

Xiahan Shi, Leonard Salewski, Martin Schiegg, Zeynep Akata, and Max Welling. 2019. Relational generalized few-shot learning. *CoRR*, abs/1907.09557.

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

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

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.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638.

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, Xiaohui Yan, Yi Chang, and Philip Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3090–3099.

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.

Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. 2019. Learning classifier synthesis for generalized few-shot learning. *arXiv preprint arXiv:1906.02944*.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S Yu. 2018. Joint slot filling and intent detection via capsule neural networks. *arXiv preprint arXiv:1812.09471*.

Ningyu Zhang, Zhanlin Sun, Shumin Deng, Jiaoyan Chen, and Huajun Chen. 2019. Improving few-shot text classification via pretrained language representations. *arXiv preprint arXiv:1908.08788*.

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*.