Energy-based Unknown Intent Detection with Data Manipulation

Yawen Ouyang, Jiasheng Ye, Yu Chen, Xinyu Dai[*],
Shujian Huang and **Jiajun Chen**
National Key Laboratory for Novel Software Technology, Nanjing University, China
{ouyangyw,yejiasheng,cheny98}@smail.nju.edu.cn
{daixinyu,huangsj,chenjj}@nju.edu.cn

Abstract

Unknown intent detection aims to identify the out-of-distribution (OOD) utterance whose intent has never appeared in the training set. In this paper, we propose using energy scores for this task as the energy score is theoretically aligned with the density of the input and can be derived from any classifier. However, high-quality OOD utterances are required during the training stage in order to shape the energy gap between OOD and in-distribution (IND), and these utterances are difficult to collect in practice. To tackle this problem, we propose a data manipulation framework to **G**enerate high-quality **O**OD utterances with importance weigh**T**s (**GOT**). Experimental results show that the energy-based detector fine-tuned by GOT can achieve state-of-the-art results on two benchmark datasets.

1 Introduction

Unknown intent detection is a realistic and challenging task for dialogue systems. Detecting out-of-distribution (OOD) utterances is critical when employing dialogue systems in an open environment. It can help dialogue systems gain a better understanding of what they do not know, which prevents them from yielding unrelated responses and improves user experience.

A simple approach for this task relies on the softmax confidence score and achieves promising results (Hendrycks and Gimpel, 2017). The softmax-based detector will classify the input as OOD if its softmax confidence score is smaller than the threshold. Nevertheless, further works demonstrate that using the softmax confidence score might be problematic as the score for OOD inputs can be arbitrarily high (Louizos and Welling, 2017; Lee et al., 2018).

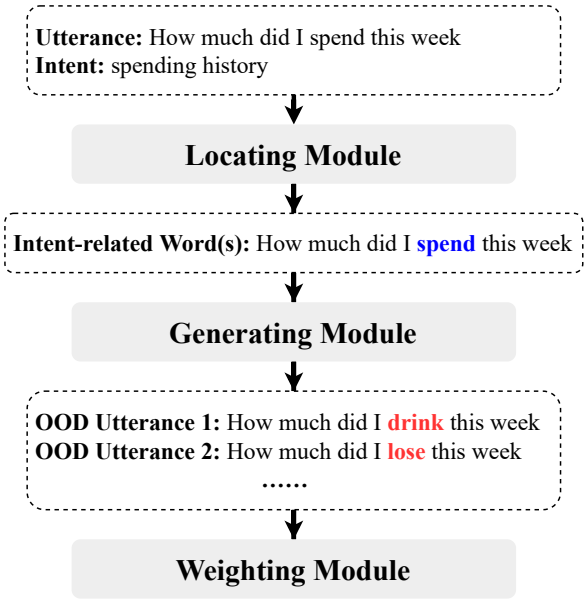

Figure 1: An overview of our framework GOT. For the utterance "How much did I spend this week" from CLINC150 dataset (Larson et al., 2019). Our locating module locates the intent-related word "spend". And then our generating module generates words "drink", "lose" to replace it and obtains OOD utterances. Finally, our weighting module assigns a weight for each of OOD utterances.

Another appealing approach is to use generative models to approximate the distribution of in-distribution (IND) training data and use the likelihood score to detect OOD inputs. However, Ren et al. (2019) and Gangal et al. (2019) find that likelihood scores derived from such models are problematic for this task as they can be confounded by background components in the inputs.

In this paper, we propose using *energy scores* (Liu et al., 2020) for unknown intent detection. The benefit is that energy scores are theoretically well aligned with the density of the inputs, hence more suitable for OOD detection. Inputs with higher energy scores mean lower densities, which can be

[*] Corresponding author.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2852–2861
August 1–6, 2021.

classified as OOD by the energy-based detector. Moreover, energy scores can be derived from any pre-trained classifier without re-training. Nevertheless, the energy gap between IND and OOD utterances might not always be optimal for differentiation. Thus we need auxiliary OOD utterances to explicitly shape the energy gap between IND and OOD utterances during the training stage (Liu et al., 2020). This poses a new challenge in that the variety of possible OOD utterances is almost infinite. It is impossible to sample all of them to create the gap. Zheng et al. (2019) demonstrate that OOD utterances akin to IND utterances, such as sharing the same phrases or patterns, are more effective, whereas these high-quality OOD utterances are difficult and expensive to collect in practice.

To tackle this problem, we propose a data manipulation framework **GOT** to generate high-quality OOD utterances as well as importance weights. GOT generates OOD utterances by perturbing IND utterances locally, which allows the generated utterances to be closer to IND. Specifically, GOT contains three modules: (1) a locating module to locate intent-related words in IND utterances; (2) a generating module to generate OOD utterances by replacing intent-related words with desirable candidate words, evaluated in two aspects: whether the candidate word is suitable given the context, and whether the candidate word is irrelevant to IND; (3) a weighting module to reduce the weights of potential harmful generated utterances. Figure 1 illustrates the overall process of GOT. Experiments show that the generated weighted OOD utterances can further improve the performance of the energy-based detector in unknown intent detection. Our code and data will be available at: https://github.com/yawenouyang/GOT.

To summarize, the key contributions of the paper are as follows:

- We propose using energy scores for unknown intent detection. We conduct experiments on real-world datasets including CLINC150 and SNIPS to show that the energy score can achieve comparable performance as strong baselines.

- We put forward a new framework GOT to generate high-quality OOD utterances and reweight them. We demonstrate that GOT can further improve the performance of the energy score by explicitly shaping the energy gap and achieves state-of-the-art results.

- We show the generality of GOT by applying generated weighted OOD utterances to fine-tune the softmax-based detector, and the fine-tuned softmax-based detector can also yield significant improvements.

2 Related Work

Lane et al., 2006, Manevitz and Yousef, 2007 and Dai et al., 2007 address OOD detection for the text-mining task. Recently, this problem has attracted growing attention from researchers (Tur et al., 2014; Fei and Liu, 2016; Fei et al., 2016; Ryu et al., 2017; Shu et al., 2017). Hendrycks and Gimpel (2017) present a simple baseline that utilizes the softmax confidence score to detect OOD inputs. Shu et al. (2017) create a binary classifier and calculate the confidence threshold for each class. Some distance-based methods (Oh et al., 2018; Lin and Xu, 2019; Yan et al., 2020) are also used to detect unknown intents as OOD utterances highly deviate from IND utterances in their local neighborhood. Simultaneously, with the advancement of deep generative models, learning such a model to approximate the distribution of training data is possible. However, Ren et al. (2019) find that likelihood scores derived from these models can be confounded by background components, and propose a likelihood ratio method to alleviate this issue. Gangal et al. (2019) reformulate and apply this method to unknown intent detection.

Different from these methods, we introduce the energy score for this task. Liu et al. (2020) prove that the energy score is theoretically aligned with the density of the input, and can be derived from any classifier without re-training, hence desirable for our task. We further propose a data manipulation framework to generate high-quality OOD utterances to shape the energy gap between IND and OOD utterances.

Note that there are some related works that also generate OOD samples to improve OOD detection performance. Lee et al. (2017) generate OOD samples with Generative Adversarial Network (GAN) (Goodfellow et al., 2014), and Zheng et al. (2019) explore this method for unknown intent detection. However, there are two major distinctions between our study and these works. First, they generate OOD utterances according to continuous latent variables, which cannot be easily interpreted. In contrast, our framework generates utterances by performing local replacements to IND utterances,

which is more interpretable to human. Second, our framework additionally contains a weighting module to reform the generated utterances. Our work is also inspired by Cai et al. (2020), which proposes a framework to augment the IND data, while our framework aims to generate OOD data.

3 Preliminary

In this section, we formalize unknown intent detection task. Then we introduce the energy score, and its superiority and limitations for this task.

3.1 Problem Formulation

Given a training dataset $\mathcal{D}_{\text{in}}^{\text{train}} = \{(\mathbf{u}^{(i)}, y^{(i)})\}_{i=1}^{N}$ where $\mathbf{u}^{(i)}$ is an utterance and $y^{(i)} \in Y_{\text{in}} = \{y_1, y_2, ..., y_K\}$ is its intent label. In testing, given an utterance, unknown intent detection aims to detect whether its intent belongs to existing intents Y_{in}. In general, unknown intent detection is an OOD detection task. The essence of all methods is to learn a score function that maps each utterance $\mathbf{u}$ to a single scalar that is distinguishable between IND and OOD utterances.

3.2 Energy-based OOD Detection

An energy-based model (LeCun et al., 2006) builds an energy function $E(\mathbf{u})$ that maps an input $\mathbf{u}$ to a scalar called energy score (i.e., $E : \mathbb{R}^D \to \mathbb{R}$). Using the energy function, probability density $p(\mathbf{u})$ can be expressed as:

$$p(\mathbf{u}) = \frac{\exp(-E(\mathbf{u})/T)}{Z}, \tag{1}$$

where $Z = \int_{\mathbf{u}} \exp(-E(\mathbf{u})/T)$ is the normalizing constant also known as the partition function and T is the temperature parameter. Take the logarithm of both side of (1), we can get the equation:

$$\log p(\mathbf{u}) = -E(\mathbf{u})/T - \log Z. \tag{2}$$

Since Z is constant for all input $\mathbf{u}$, we can ignore the last term $\log Z$ and find that the energy function $-E(\mathbf{u})$ is in fact linearly aligned with the log likelihood function, which is desirable for OOD detection (Liu et al., 2020).

The energy-based model has a connection with a softmax-based classifier. For a classification problem with K classes, a parametric function f maps each input $\mathbf{u}$ to K real-valued numbers (i.e., $f : \mathbb{R}^D \to \mathbb{R}^K$), known as logits. Logits are used to parameterize a categorical distribution using a softmax function:

$$p(y|\mathbf{u}) = \frac{\exp[f_y(\mathbf{u})/T]}{\sum_{y'} \exp[f_{y'}(\mathbf{u})/T]}, \tag{3}$$

where $f_y(\mathbf{u})$ indicates the y^{th} index of $f(\mathbf{u})$, i.e., the logit corresponding the y^{th} class label. And these logits can be reused to define an energy function without changing function f (Liu et al., 2020; Grathwohl et al., 2020):

$$E(\mathbf{u}) = -T \cdot \log \sum_{y'} \exp\left[f_{y'}(\mathbf{u})/T\right]. \tag{4}$$

According to the above, a classifier can be reinterpreted as an energy-based model. It also means the energy score can be derived from any classifier.

Due to its consistency with density and accessibility, we introduce the energy score for unknown intent detection, and utterances with higher energy scores can be viewed as OOD. Mathematically, the energy-based detector G can be described as:

$$G(\mathbf{u}; \delta, E) = \begin{cases} \text{IND} & E(\mathbf{u}) \leq \delta, \\ \text{OOD} & E(\mathbf{u}) > \delta, \end{cases} \tag{5}$$

where δ is the threshold.

Although the energy score can be easily computed from the classifier, the energy gap between IND and OOD samples might not always be optimal for differentiation. To solve this problem, Liu et al. (2020) propose an energy-bounded learning objective to further widen the energy gap. Specifically, the training objective of the classifier combines the standard cross-entropy loss with a regularization loss:

$$\mathcal{L} = \mathbb{E}_{(\mathbf{u},y)\sim\mathcal{D}_{\text{in}}^{\text{train}}}[-\log F_y(\mathbf{u})] + \lambda \cdot \mathcal{L}_{\text{energy}}, \tag{6}$$

where $F(\mathbf{u})$ is the softmax output, λ is the auxiliary loss weight. The regularization loss is defined in terms of energy:

$$\begin{aligned} \mathcal{L}_{\text{energy}} &= \mathbb{E}_{(\mathbf{u},y)\sim\mathcal{D}_{\text{in}}^{\text{train}}}(\max(0, E(\mathbf{u}) - m_{\text{in}}))^2 \\ &+ \mathbb{E}_{\hat{\mathbf{u}}\sim\mathcal{D}_{\text{out}}^{\text{train}}}(\max(0, m_{\text{out}} - E(\hat{\mathbf{u}})))^2, \end{aligned} \tag{7}$$

which utilizes both labeled IND data $\mathcal{D}_{\text{in}}^{\text{train}}$ and auxiliary unlabeled OOD data $\mathcal{D}_{\text{out}}^{\text{train}}$. This term differentiates the energy scores between IND and OOD samples by using two squared hinge loss with the margin hyper-parameters m_{in} and m_{out}.

Ideally, one has to sample all types of OOD utterances to create the gap, which is impossible in practice. Zheng et al. (2019) demonstrate that OOD utterances akin to IND utterances could be more effective, but more difficult to collect. To address this problem, we propose a data manipulation framework, which can generate these high-quality OOD utterances and assign each generated utterance an importance weight to reduce the impact of potential bad generation.

4 Approach

In this section, we will introduce our data manipulation framework GOT in detail. GOT aims to generate high-quality OOD utterances by replacing intent-related words in IND utterances, and then assign a weight to each generated OOD utterance. Eventually, the weighted OOD utterances can be used to shape the energy gap.

4.1 Locating Module

Since not all words in utterances are meaningful, such as stop words, when generating OOD utterances, replacing these words may not change the intent. It is more efficient and effective to replace those intent-related words. Hence, we design an intent-related score function S to measure how a word w related to an intent y:

$$S(w,y) = \sum_{\mathbf{u}\in\mathcal{D}_y^{\text{train}}} \sum_{w_j\in\mathbf{u}} \mathbb{I}(w_j = w)[\log p(w_j|\boldsymbol{w}_{<j}, y) - \log p(w_j|\boldsymbol{w}_{<j})], \quad (8)$$

where $\mathcal{D}_y^{\text{train}}$ is the subset of $\mathcal{D}_{\text{in}}^{\text{train}}$, which contains utterances with intent y, $\mathbb{I}$ is the indicator function, w_j is the j^{th} word in $\mathbf{u}$, and $\boldsymbol{w}_{<j} = w_1, ..., w_{j-1}$.

Given w and y, the intent-related score function is the sum of the log-likelihood ratios for all w in $\mathcal{D}_y^{\text{train}}$. If w is related to y, w tends to occur more frequently in $\mathcal{D}_y^{\text{train}}$ than other words. For each occurrence of w, i.e., w_j equals w, $p(w|\boldsymbol{w}_{<j}, y)$ should be higher than $p(w|\boldsymbol{w}_{<j})$ as the former is additionally conditioned on the related y, while the latter is not, hence resulting in a higher $S(w, y)$. In contrast, if w is not related to y, $p(w|\boldsymbol{w}_{<j}, y)$ is much less likely to be higher than $p(w|\boldsymbol{w}_{<j})$, or w tends to have a lower frequency in $\mathcal{D}_y^{\text{train}}$, hence $S(w, y)$ is likely to be small. Therefore, $S(w, y)$ can serve as a valid score function to measure how a word w is related to an intent y.

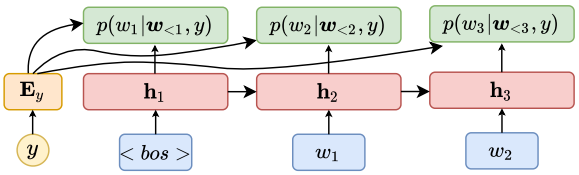

Figure 2: A class-conditional language model to estimate $p(w_j|\boldsymbol{w}_{<j}, y)$.

With the help of S, given an utterance to be replaced and its intent label, the locating module calculates the intent-related score for each word in this utterance, and a word with a higher score (i.e., larger than a given threshold) can be viewed as an intent-related word.

Implementation: We use two generative models to estimate $p(w_j|\boldsymbol{w}_{<j}, y)$ and $p(w_j|\boldsymbol{w}_{<j})$ separately. Specifically, we train a class-conditional language model (Yogatama et al., 2017) with $\mathcal{D}_{\text{in}}^{\text{train}}$ to estimate $p(w_j|\boldsymbol{w}_{<j}, y)$, shown in Figure 2. To predict the word w_j, we can combine the hidden state h_j with the intent embedding from a learnable label embedding matrix E_y, then pass it through a fully connected (FC) layer and a softmax layer to estimate the word distribution. In the training process, the input is the utterance with its intent from $\mathcal{D}_{\text{in}}^{\text{train}}$, and the training objective is to maximize the conditional likelihood of utterances. To estimate $p(w_j|\boldsymbol{w}_{<j})$, we directly use pre-trained GPT-2 (Radford et al., 2019) without tuning. Note that the whole training process only needs $\mathcal{D}_{\text{in}}^{\text{train}}$, and does not need auxiliary supervised data.

4.2 Generating Module

After detecting intent-related words in the utterance $\mathbf{u}$, for each of the intent-related words w_t, the generating module aims to generate the replacement words from the vocabulary set to replace w_t and obtain OOD utterances. We design a candidate score function Q to measure the desirability of the candidate word c:

$$Q(c; \mathbf{u}, w_t) = \log p(c|\boldsymbol{w}_{<t}, \boldsymbol{w}_{>t}) - \log \sum_{y\in Y_{\text{in}}} p(c|\boldsymbol{w}_{<t}, y)p(y). \quad (9)$$

The first term of the right hand side is the log-likelihood of c conditioned on the context of w_t; the higher it is, the more suitable c is given the context. The second term of the right hand side is the negative log of the average likelihoods of c conditioned on the IND label and previous context;

the higher it is, the less relevant c is to IND utterances. Therefore, if c has a higher candidate score, that means it fits the context well and has a low density under the IND utterance distribution, thus can be selected as the replacement word to replace w_t. The resulting generated OOD utterance is:

$$\hat{\mathbf{u}} = \{\boldsymbol{w}_{<t}, c, \boldsymbol{w}_{>t}\}. \quad (10)$$

Implementation: Similar with the locating module, we also do not need auxiliary supervised data to train the generating module. We use the same class-conditional language model mentioned in Section 4.1 to estimate $p(c|\boldsymbol{w}_{<t}, y)$. $p(y)$ is the training set label ratios. To estimate $p(c|\boldsymbol{w}_{<t}, \boldsymbol{w}_{>t})$, we use pre-trained BERT (Devlin et al., 2018) without tuning.

4.3 Weighting Module

Since we cannot ensure the generation process is perfect, given a generated OOD utterance set $\mathcal{D}_{\text{out}}^{\text{gen}} = \{\hat{\mathbf{u}}^{(i)}\}_{i=1}^{M}$, there might be some unfavorable utterances that are useless or even harmful for tuning the classifier. To fit these utterances, the generalization ability of the classifier will decrease. The weighting module aims to assign these utterances small weights.

We first use Equation 6 as the loss function to train a classifier by taking $\mathcal{D}_{\text{out}}^{\text{gen}}$ as $\mathcal{D}_{\text{out}}^{\text{train}}$. Then we calculate the influence value $\phi \in \mathbb{R}$ (Wang et al., 2020) for each generated utterance $\hat{\mathbf{u}}$. The influence value approximates the influence of removing this utterance on the loss at validation samples. An utterance with positive ϕ implies that its removal will reduce the validation loss and strengthen the classifier's generalization ability, thus we should assign it a small weight. * In particular, given ϕ, we calculate weight α as follows:

$$\alpha = \frac{1}{1 + e^{\frac{\gamma\phi}{\max_\phi - \min_\phi}}}, \quad (11)$$

where $\gamma \in \mathbb{R}^+$ is used to make the weight distribution flat or steep, $\max_\phi$ and $\min_\phi$ are the maximum and minimum influence value of utterances in $\mathcal{D}_{\text{out}}^{\text{gen}}$.

Implementation: We still do not need auxiliary supervised data for this module. The validation loss is the cross-entropy loss on the validation set.

*Details about how to calculate the influence can be found in (Koh and Liang, 2017; Wang et al., 2020).

Algorithm 1 Data Manipulation Process

Input: Training set $\mathcal{D}_{\text{in}}^{\text{train}}$, intent-related score function S, candidate score function Q, intent-related word threshold ϵ, candidate number K, weight term γ
Output: Generated weighted OOD utterances set $\mathcal{D}_{\text{out}}^{\text{gw}}$

```
1: D_out^gen = {} # generated OOD utterances without weights
2: for (u, y) ∈ D_in^train do
3:     for w_j ∈ u do
4:         if S(w_j, y) > ε then
5:             C = top−K_c Q(c; u, w_j)
6:             for c ∈ C do
7:                 û = {w_<j, c, w_>j}
8:                 Add û into D_out^gen
9:             end for
10:        end if
11:    end for
12: end for
13: D_out^gw = {}        # generated weighted OOD utterances
14: for û ∈ D_out^gen do
15:    Calculate the weight α by Equation 11
16:    Add (û, α) into D_out^gw
17: end for
18: return D_out^gw
```

4.4 Overall Data Manipulation Process

We summarize the process of GOT in Algorithm 1. Line 4 shows that w_j can be viewed as an intent-related word for y if $S(w_j, y)$ is greater than the intent-related word threshold ϵ. Line 5 shows that we generate K replacement words with the top-K $Q(c; \mathbf{u}, w_t)$.

4.5 Shape the energy gap with GOT

After obtaining weighted OOD utterances set $\mathcal{D}_{\text{out}}^{\text{gw}}$, we can explicitly shape the energy gap with them, resulting in IND utterances with smaller energy scores and OOD utterances with higher energy scores. Specifically, we redefine the regularization loss in Equation 6 as follows and use it to re-train the classifier:

$$\mathcal{L}_{\text{energy}} = \mathbb{E}_{(\mathbf{u},y)\sim\mathcal{D}_{\text{in}}^{\text{train}}}(\max(0, E(\mathbf{u}) - m_{\text{in}}))^2 + \mathbb{E}_{(\hat{\mathbf{u}},\alpha)\sim\mathcal{D}_{\text{out}}^{\text{gw}}}\alpha(\max(0, m_{\text{out}} - E(\hat{\mathbf{u}})))^2. \quad (12)$$

In the testing process, we can calculate the energy score for the utterance by Equation 4, and identify whether it is OOD by Equation 5.

5 Experimental Setup

5.1 Datasets

To evaluate the effectiveness of the energy score and our proposed framework, we conducted experiments on two public datasets:

- CLINC150† (Larson et al., 2019): this dataset

†https://github.com/clinc/oos-eval

Statistic	CLINC150	SNIPS
Train	15000	9385
validation	3000	500
Test-IND	4500	486
Test-OOD	1000	214
Test-IND: Test-OOD	4.5: 1	2.3: 1
Number of IND classes	150	5

Table 1: Statistics of CLINC150 and SNIPS datasets.

covers 150 intent classes over ten domains. It supports some OOD utterances that do not fall into any of the system's supported intents to avoid splitting unknown intents manually.

- SNIPS[‡] (Coucke et al., 2018): this dataset is a personal voice assistant dataset that contains seven intent classes. SNIPS does not explicitly include OOD utterances. We kept two classes *SearchCreativeWork* and *SearchScreeningEvent* as unknown intents.

Table 1 provides summary statistics about these two datasets. Note that the training set and validation set do not include OOD utterances.

5.2 Metrics

We used four common metrics for OOD detection to measure the performance. AUROC (Davis and Goadrich, 2006), AUPR In and AUPR Out (Manning et al., 1999) are threshold-independent performance evaluations and higher values are better. FPR95 is the false positive rate (FPR) when the true positive rate (TPR) is 95%, and lower values are better.

Considering the smaller proportion of OOD utterances in the test set on two datasets, AUPR Out is more informative here.

5.3 Baselines

We introduce the following classifier-based methods as baselines:

- MSP (Hendrycks and Gimpel, 2017) trains a classifier with IND utterances and uses the softmax confidence score to detect OOD utterances.
- DOC (Shu et al., 2017) trains a binary classifier for each IND intent and uses maximum binary classifier output to detect OOD utterances.
- Mahalanobis (Lee et al., 2018) trains a classifier with softmax loss and uses Mahalanobis distance of the input to the nearest class-conditional Gaussian distribution to detect OOD utterances.
- LMCL (Lin and Xu, 2019) uses LOF (Breunig et al., 2000) in the utterance representation learned by a classifier. In training, they replace the softmax loss with LMCL (Wang et al., 2018).
- SEG (Yan et al., 2020) also uses LOF in the utterance representation. In training, they use semantic-enhanced large margin Gaussian mixture loss.

5.4 Implementation Details

For a fair comparison, all classifiers used in the above methods and ours are pre-trained BERT (Devlin et al., 2018) with a multi-layer perceptron (MLP). We select parameter values based on validation accuracy. For energy score, we follow Liu et al. (2020) to set T as 1, λ as 0.1, m_{in} as -8 and m_{out} as -5. For influence value, we focus on changes on MLP parameters and use stochastic estimation (Koh and Liang, 2017) with the scaling term 1000 and the damping term 0.003. For LMCL implementation, we set nearest neighbor number as 20, scaling factor s as 30 and cosine margin m as 0.35, which is recommended by Lin and Xu (2019). For SEG, we follow Yan et al. (2020) to set margin as 1 and trade-off parameter as 0.5.

For our framework, we set candidate number K as 2, weight term γ as 20. In particular, for CLINC150, we set threshold ϵ as 150 and generate 100 weighted utterances for each intent. For SNIPS, we set threshold ϵ as 1500 and generate 1800 weighted utterances for each intent. The difference in settings between two datasets is due to the different sizes of per intent in the training set.

6 Results and Analysis

In this part, we will show the results of different methods on two datasets and offer some further analysis.

6.1 Overall Results

As shown in Table 2, we can observe that:

- The energy score can achieve comparable results on two datasets. Note that on SNIPS dataset, the advantages of the energy score are not as obvious. The reason is that SNIPS dataset is not as challenging as CLINC150 dataset, most methods can achieve good results, such as AUPR out is greater than 0.9.

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

Method	CLINC150				SNIPS			
	AUROC ↑	**FPR95↓**	**AUPR In↑**	**AUPR Out↑**	**AUROC ↑**	**FPR95↓**	**AUPR In↑**	**AUPR Out↑**
MSP	0.955	0.164	0.990	0.814	0.951	0.370	0.970	0.922
DOC	0.943	0.221	0.985	0.790	0.938	0.493	0.956	0.910
Mahalanobis	0.969	0.118	0.993	0.871	0.979	0.088	0.989	0.964
LMCL	0.962	0.124	0.992	0.810	0.976	0.087	0.987	0.960
SEG	0.959	0.152	0.991	0.823	0.974	0.074	0.986	0.948
Energy	0.967	0.143	0.991	0.897	0.944	0.497	0.964	0.924
Energy + GOT	**0.973**	**0.114**	**0.993**	**0.914**	**0.989**	**0.039**	**0.995**	**0.972**
Energy + GOT w/o weighting	0.972	0.123	0.992	0.909	0.979	0.083	0.989	0.969

Table 2: AUROC, FPR95, AUPR In, AUPR Out on CLINC150, SNIPS datasets. Best results are in bold. All results are averaged across five seeds.

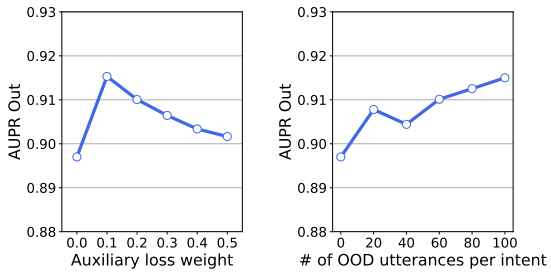

Figure 3: Effect of the auxiliary loss weight (left) and the number of generated weighted OOD utterances per intent (right).

Method	**AUROC ↑**	**FPR95↓**	**AUPR In↑**	**AUPR Out↑**
Energy	0.967	0.143	0.991	0.897
Energy + Wiki	0.961	0.170	0.988	0.889

Table 3: Effect of using Wikipedia sentences to shape the energy gap.

Method	**AUROC ↑**	**FPR95↓**	**AUPR In↑**	**AUPR Out↑**
MSP	0.955	0.164	0.990	0.814
MSP + GOT	0.972	0.118	0.993	0.903

Table 4: Effect of using GOT to fine-tune the softmax-based detector.

- Energy + GOT achieves better results on two datasets as compared to the raw energy score. It indicates that our generated weighted OOD utterances can effectively shape the energy gap, resulting in more distinguishable between IND and OOD utterances.
- We also report ablation study results. "w/o weighting" is the energy score tuned by OOD utterances without reweighting. We can see that there is a decrease in performance on both datasets, which shows the advantage of the weighting module (p-value < 0.005).

6.2 Effect of Hyper-parameters

During the training process, we find that the method performance is sensitive to two hyper-parameters: auxiliary loss weight λ and the number of generated weighted OOD utterances per intent. We conduct two experiments to demonstrate their effects separately. We choose CLINC150 dataset as it is more challenging as mentioned before.

Auxiliary Loss Weight: We set the auxiliary loss weight λ from 0 to 0.5 with an interval of 0.1 to observe its impact.

Results are shown in Figure 3 (left). With the increase of auxiliary loss weight, the performance increases first and then decreases. $\lambda = 0.1$ achieves the highest AUPR Out 0.914 and outperforms $\lambda = 0$ with an improvement of 1.7% (AUPR Out). The results suggest that although shaping the energy gap can improve the performance, there exists a trade-off between optimizing the regularization loss and optimizing cross-entropy loss.

Number of OOD Utterances: we compare the performance of generated weighted utterance numbers for each intent by adjusting the number from 0 to 100 with an interval of 20.

Results are shown in Figure 3 (right). As a whole, AUPR Out increases as more OOD utterances are incorporated into training. We can see that the performance is also improved even with a small generated number, which indicates the necessity of explicitly shaping the energy gap.

6.3 Compare with Wikipedia Sentences

An easy way to obtain OOD utterances is from the Wikipedia corpus. We investigate the effect of regarding Wikipedia sentences as OOD utterances to shape the energy gap on CLINC150 dataset. The Wikipedia sentences are from Larson et al. (2019) and the number is 14750.

As shown in Table 3, we can observe that these sentences cannot improve the performance and even have a negative effect (We experimented with several hyper-parameters, this is the best result we could get). After observing these Wikipedia sentences, we find that they have little relevance to

Intent	IND utterance and intent-related word [w]	Replacement word	Weight
Insurance	i need to know more about my health [**plan**]	problems	0.50
	what [**benefits**] are provided by my insurance	services	0.19
Credit Limit Change	can i get a higher limit on my american express [**card**]	ticket	0.46
	can you [**increase**] how much i can spend on my visa	guess	0.54
Reminder	can you list each item on my [**reminder**] list	contacts	0.50
	what's on the [**reminder**] list	agenda	0.78
Redeem Rewards	walk me through the process of cashing in on [**credit**] card points	those	0.22
	i have credit card [**points**] but don't know how to use them	privileges	0.50

Table 5: Weighted OOD utterances generated by GOT on CLINC150 dataset.

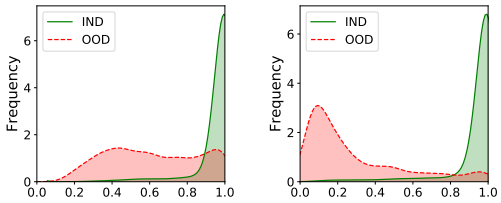

Figure 4: Histogram of the softmax confidence score from MSP (left) and MSP + GOT (right).

IND utterances. Therefore, simply using Wikipedia sentences is unrepresentative and ineffective for shaping the energy gap.

6.4 GOT for Softmax-based Detector

As mentioned in Section 1, when using the softmax-based detector, OOD inputs may also receive a high softmax confidence score. To tackle this problem, Lee et al. (2017) replace the cross entropy loss with the confidence loss. The confidence loss adds the Kullback-Leibler loss (KL loss) on the original cross entropy loss, which forces OOD inputs less confident by making their predictive distribution to be closer to uniform.

To verify the generality of GOT, we directly use the generated weighted OOD utterances to fine-tune the softmax-based detector with the confidence loss. The results are shown in Table 4. Our MSP + GOT has a significant improvement and outperforms MSP by 8.9% (AUPR Out). Figure 4 provides an intuitive presentation. The softmax confidence scores of OOD from MSP form smooth distributions (see Figure 4 (left)). In contrast, the softmax confidence scores of OOD from MSP + GOT concentrate on small values (see Figure 4 (right)). Overall the softmax confidence score is more distinguishable between IND and OOD after tuning by GOT.

6.5 Case Study for GOT

We sample some intents and showcase generated weighted OOD utterances in Table 5. We can observe that intent-related words that located by our locating module are diverse, containing not only words appeared in the intent label. The replacement word fits the context well, and the intent of the generated utterance is exactly changed in most conditions. Admittedly, GOT may have a bad generation, like replace "benefits" with "services" in the second utterance, which leads the generated utterance is still in-domain. Fortunately, the weighting module assigns these utterances a lower weight to reduce their potential harm.

7 Conclusion and Future Work

In this paper, we propose using energy scores for unknown intent detection and provide empirical evidence that the energy-based detector is comparable to strong baselines. To shape the energy gap, we propose a data manipulation framework GOT to generate high-quality OOD utterances and assign their importance weights. We show that the energy-based detector tuned by GOT can achieve state-of-the-art results. We further employ generated weighted utterances to fine-tune the softmax-based detector and also achieve improvements.

In the future, we will explore more operations, such as insertion, drop, etc., to enhance the diversity of generated utterances.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments. This work was supported by NSFC Projects (Nos. 61936012 and 61976114), the National Key R&D Program of China (No. 2018YFB1005102).

References

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104.

Hengyi Cai, Hongshen Chen, Yonghao Song, Cheng Zhang, Xiaofang Zhao, and Dawei Yin. 2020. Data manipulation: Towards effective instance learning for neural dialogue generation via learning to augment and reweight. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

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*.

Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. 2007. Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 210–219.

Jesse Davis and Mark Goadrich. 2006. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240.

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*.

Geli Fei and Bing Liu. 2016. Breaking the closed world assumption in text classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 506–514, San Diego, California. Association for Computational Linguistics.

Geli Fei, Shuai Wang, and Bing Liu. 2016. Learning cumulatively to become more knowledgeable. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1565–1574.

Varun Gangal, Abhinav Arora, Arash Einolghozati, and Sonal Gupta. 2019. Likelihood ratios and generative classifiers for unsupervised out-of-domain detection in task oriented dialog. *arXiv preprint arXiv:1912.12800*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. 2020. Your classifier is secretly an energy based model and you should treat it like one. *International Conference on Learning Representatinos*.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR.

Ian Lane, Tatsuya Kawahara, Tomoko Matsui, and Satoshi Nakamura. 2006. Out-of-domain utterance detection using classification confidences of multiple topics. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):150–161.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. 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 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. 2006. A tutorial on energy-based learning. *Predicting structured data*, 1(0).

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2017. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7167–7177. Curran Associates, Inc.

Ting-En Lin and Hua Xu. 2019. Deep unknown intent detection with margin loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5491–5496, Florence, Italy. Association for Computational Linguistics.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33.

Christos Louizos and Max Welling. 2017. Multiplicative normalizing flows for variational bayesian neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2218–2227. JMLR. org.

Larry Manevitz and Malik Yousef. 2007. One-class document classification via neural networks. *Neurocomputing*, 70(7-9):1466–1481.

Christopher D Manning, Christopher D Manning, and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT press.

Kyo-Joong Oh, DongKun Lee, Chanyong Park, Young-Seob Jeong, Sawook Hong, Sungtae Kwon, and Ho-Jin Choi. 2018. Out-of-domain detection method based on sentence distance for dialogue systems. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 673–676. IEEE.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. 2019. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, pages 14680–14691.

Seonghan Ryu, Seokhwan Kim, Junhwi Choi, Hwanjo Yu, and Gary Geunbae Lee. 2017. Neural sentence embedding using only in-domain sentences for out-of-domain sentence detection in dialog systems. *Pattern Recognition Letters*, 88:26–32.

Lei Shu, Hu Xu, and Bing Liu. 2017. DOC: Deep open classification of text documents. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2911–2916, Copenhagen, Denmark. Association for Computational Linguistics.

Gokhan Tur, Anoop Deoras, and Dilek Hakkani-Tür. 2014. Detecting out-of-domain utterances addressed to a virtual personal assistant. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. 2018. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930.

Zifeng Wang, Hong Zhu, Zhenhua Dong, Xiuqiang He, and Shao-Lun Huang. 2020. Less is better: Unweighted data subsampling via influence function. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6340–6347.

Guangfeng Yan, Lu Fan, Qimai Li, Han Liu, Xiaotong Zhang, Xiao-Ming Wu, and Albert Y.S. Lam. 2020. Unknown intent detection using Gaussian mixture model with an application to zero-shot intent classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1050–1060, Online. Association for Computational Linguistics.

Dani Yogatama, Chris Dyer, Wang Ling, and Phil Blunsom. 2017. Generative and discriminative text classification with recurrent neural networks. *arXiv preprint arXiv:1703.01898*.

Yinhe Zheng, Guanyi Chen, and Minlie Huang. 2019. Out-of-domain detection for natural language understanding in dialog systems. *arXiv preprint arXiv:1909.03862*.