# Likelihood Ratios and Generative Classifiers for Unsupervised Out-of-Domain Detection in Task Oriented Dialog

**Varun Gangal,**[1,2*] **Abhinav Arora,**[2] **Arash Einolghozati,**[2] **Sonal Gupta**[2]
[1]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213
[2]Facebook Conversational AI, Menlo Park, CA 94303
vgangal@andrew.cmu.edu {abhinavarora, arashe, sonalg}@fb.com

## Abstract

The task of identifying out-of-domain (*OOD*) input examples directly at test-time has seen renewed interest recently due to increased real world deployment of models. In this work, we focus on OOD detection for natural language sentence inputs to task-based dialog systems. Our findings are three-fold:

First, we curate and release ROSTD (**R**eal **O**ut-of -Domain **S**entences From **T**ask-oriented **D**ialog) - a dataset of $4K$ *OOD* examples for the publicly available dataset from (Schuster et al. 2019). In contrast to existing settings which synthesize *OOD* examples by holding out a subset of classes, our examples were authored by annotators with apriori instructions to be out-of-domain with respect to the sentences in an existing dataset.

Second, we explore likelihood ratio based approaches as an alternative to currently prevalent paradigms. Specifically, we reformulate and apply these approaches to natural language inputs. We find that they match or outperform the latter on all datasets, with larger improvements on non-artificial *OOD* benchmarks such as our dataset. Our ablations validate that specifically using likelihood ratios rather than plain likelihood is necessary to discriminate well between *OOD* and in-domain data.

Third, we propose learning a generative classifier and computing a marginal likelihood (ratio) for *OOD* detection. This allows us to use a principled likelihood while at the same time exploiting training-time labels. We find that this approach outperforms both simple likelihood (ratio) based and other prior approaches. We are hitherto the first to investigate the use of generative classifiers for *OOD* detection at test-time.

## 1 Introduction

With increased use of ML models in real life settings, it has become imperative for them to self-identify, at test-time, examples on which they are likely to fail due to them differing significantly from the model's training time distribution.

In particular, for state-of-the-art deep classifiers such as those used in vision and language tasks, it has been observed that the raw probability value is over calibrated (Guo et al. 2017) and can have high values even for *OOD* inputs. This necessitates having an auxiliary mechanism to detect them.

This task is not in entirety novel, and has historically been explored in related forms under various names such as *one class classification*, *open classification* etc. The recent stream of work on this started with (Hendrycks and Gimpel 2017), which proposed benchmark datasets for doing this on vision problems. (Liang, Li, and Srikant 2018) find that increasing the softmax temperature $\tau$ makes the resultant probability more discriminative for *OOD* Detection. (Lee et al. 2018b) propose using distances to per-class Gaussians in the intermediate representation learnt by the classifier. Specifically, a Gaussian is fit for each training class from all training points in that class . (Ren et al. 2019) show that "correcting" likelihood with likelihood from a "background" model trained on noisy inputs is better at discriminating out of distribution examples. Recently, (Lin and Xu 2019) propose using an old measure from the data mining literature named LOF (Breunig et al. 2000) in the space of penultimate activations learnt by a classifier.

Apart from (Lin and Xu 2019) and few others, a majority of the prior work uses vision problems and datasets, often image classification as the setting in which to perform *OOD* Detection. Certain methods, such as input gradient reversal from (Liang, Li, and Srikant 2018) or an end-to-end differentiable Generative Adversarial Network (GAN) as in (Lee et al. 2018a) are not directly applicable for natural language inputs. Furthermore, image classification has available several benchmarks with a similar label space (digits and numbers) but differing input distributions, such as *MNIST*, *CIFAR* and *SVHN*. Most of these works exploit this fact for their experimental setting by picking one of these datasets as *ID* and the other as *OOD*. In this work, we attempt to address these lacunae and specifically explore which *OOD* detection approaches work well on natural language, in particular, intent classification.

This problem is greatly relevant for task oriented dialog systems since intent classification can receive user intents which are sometimes not in any of the domains defined by the current ontology or downstream functions. In particular, **unsupervised** $OOD$ detection approaches are important as it is difficult to curate this kind of data for training because

[*] Work done by author while interning at Facebook Conv AI

1. The size of in-domain data to train on can become arbitrarily large as the concerned dialog system gets more users and acquires the ability to handle newer intent classes. After a point, it becomes impractical to continue curating newer $OOD$ examples for training in proportion to the in-domain data. From there on, class imbalance would keep increasing.
2. By definition, *OOD* is an open class. For natural language intents, utterances can demonstrate diverse sentence phenomenena such as slang, rhetorical questions, code mixed language, etc. User data can exhibit a large range of *OOD* behaviours, all of which may be difficult to encapsulate using a limited set of OOD examples at training time.

To the best of our knowledge, this is the first application of likelihood ratios approach for *OOD* Detection in natural language. Overall, our contributions are as follows:

1. We release ROSTD, a novel dataset[1] of $4500$ *OOD* sentences for intent classification. We observe that existing datasets for *OOD* intent classification are
   - too small (<1000 examples)
   - create *OOD* examples synthetically

   We show that performing *OOD* detection on ROSTD is more challenging than the synthetic setting where *OOD* examples are created by holding out some fraction of intent classes. We further describe this dataset in §4 .
2. We show that using the marginal likelihood of a generative classifier provides a principled way of both incorporating the label information [like classifier uncertainty based approaches] while at the same time testing for *ID* vs *OOD* using a likelihood function.
3. We show that using likelihood with a correction term from a "background" model, based on the formalism proposed in (Ren et al. 2019), is a much more effective approach than using the plain likelihood. We propose multiple ways of training such a background model for natural language inputs.
4. Our improvements hold on multiple datasets - both for our dataset as well as the existing SNIPS dataset (Coucke et al. 2018).

## 2 Methods

All our methods attempt to estimate a score which is indicative of the data point being *OOD*. For an input $x$, we refer to this function as $\eta(x)$. This function may additionally be parametrized by the classifier distribution $\hat{P}$ or the set of nearest neighbors $N_k(x)$, based on the specific method in use.

Some of our evaluation measures are threshold independent. In this case, $\eta(x)$ can be directly evaluated for its goodness of detecting OOD points. For measures which are threshold dependent, an optimal threshold which maximizes macro-F1 is picked using the values of $\eta(x)$ on a validation set.

### Maximum Softmax Probability

*Maximum Softmax Probability*, or MSP is a simple and intuitive baseline proposed by (Hendrycks and Gimpel 2017). MSP uses the maximum probability 1-$\max_y \hat{P}(y|x)$ as $\eta(x, \hat{P})$. The lesser "confident" the classifier $\hat{P}$ is about its predicted outcome i.e the argmax label, the greater is $\eta(x, \hat{P})$. Typically,

$$\hat{P}(y|x) = \frac{e^{\frac{z_y}{\tau}}}{\sum_y e^{\frac{z_y}{\tau}}}$$

Here, $z_y$ denotes the logit for label $y$ while $\tau$ denotes the softmax temperature. Increasing $\tau$ smoothens the distribution while decreasing $\tau$ makes it peakier. We also try increased values of $\tau$ as they were shown to work better by (Liang, Li, and Srikant 2018).

### Softmax Entropy

Alternatively, both (Lee et al. 2018a) and (Hendrycks, Mazeika, and Dietterich 2019) propose using either of the following[2]:

1. Entropy $H_Y \hat{P}(y|x) = -\sum_{y \in Y} \hat{P}(y|x) \log \hat{P}(y|x)$ as $\eta(x, \hat{P})$
2. Negative KL Divergence $-KL(\hat{P}|\mathbf{U})$ w.r.t the uniform distribution over labels $\mathbf{U}$.

We refer to this method as $-KL(\hat{P}|\mathbf{U})$ in our experiments[3]. Here, we also experiment with a variant of this method which replaces $\mathbf{U}$ with $\mathbf{R}$, where $\mathbf{R}(y) = \frac{\sum_{i=1}^{i=m} \mathbf{1}(y_i = y)}{|m|}$, or the fraction of class $y$ in the training set. We expect this variant to do better when *ID* classes are not distributed equally.

### Local Outlier Factor

LOF, proposed by (Breunig et al. 2000), is a measure based on local density defined with respect to nearest neighbours. Recently, (Lin and Xu 2019) effectively used LOF in the intermediate representation learnt by a classifier for *OOD* detection of intents. The LOF measure can be defined in three steps:

1. First, $reachdist_k(A, B) = max(kdist(B), d(A, B))$ . Here, $kdist(B)$ is the distance to the kth nearest neighbor, while $d$ is the distance measure being used. Intuitively, $reachdist_k(A, B)$ is lower-bounded by $kdist(B) \forall A$, but can become arbitrarily large.
2. Next, define a measure named local reachability density or *lrd*. This is simply the reciprocal of the average $reachdist_k$ for $A$

$$lrd(A) = \frac{|N_k(A)|}{\sum_{B \in N_k(A) reachdist_k(A,B)}}$$

[1] Our dataset is available at github.com/vgtomahawk/LR_GC_OOD/blob/master/data/fbrelease/OODrelease.tsv

[2] They differ only by a constant and have the exact same minimas, as we show in Appendix 1.1. Appendix can be read at github.com/vgtomahawk/LR_GC_OOD/tree/master/appendix

[3] One distinction from the two cited papers is that they use $-KL(\hat{P}|\mathbf{U})$ merely as an auxiliary training objective, and end up using MSP as the $\eta$ at test time. In contrast, we explicitly use $-KL(\hat{P}|\mathbf{U})$ as $\eta$

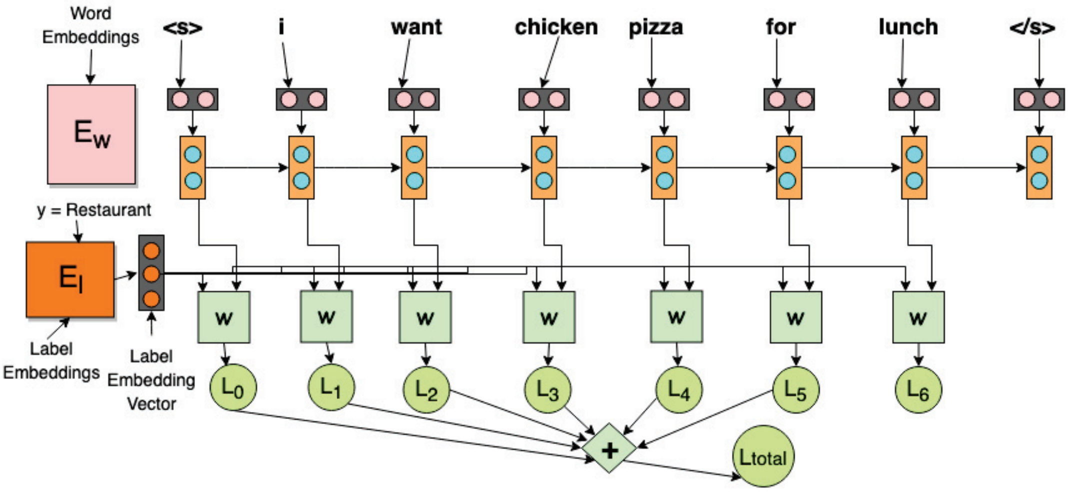


Figure 1: We illustrate the architecture of our generative classifier. $E_w$ and $E_l$ are the word embeddings and the label embeddings respectively. The hidden state is concatenated with the label embedding for *"Restaurant"* before passing through the output layer $W$. Best seen in color.

3. Lastly, LOF is defined as

$$LOF_k(A) = \frac{\sum_{B \in N_k(A)} \frac{lrd(B)}{lrd(A)}}{|N_k(A)|}$$

Intuitively, if the "density" around a point's nearest neighbours is higher than its own "density", the point will have a higher LOF. Points with a higher $LOF$ score are more likely to be *OOD*.

(Lin and Xu 2019) further show that using the *large margin cosine* loss or *LMCL*, works better than the typical combination of softmax + cross entropy.

$$\hat{P}(y|x) = \frac{e^{\frac{\overline{w_y}^T \overline{x} - m}{\tau}}}{\sum_y e^{\frac{\overline{w_y}^T \overline{x} - m}{\tau}}}$$

Here, $m$ denotes the margin. $w_y$ denotes the row in the final linear layer weight matrix corresponding to the label $y$. $x$ denotes the penultimate layer activations which are input to the final layer. We use $\overline{v}$ to denote the normalized $v$, i.e $\frac{v}{||v||}$.

We denote this approach as LOF+LMCL. We directly use the author's implementation [4] for this approach.

## Likelihood

Here, $\eta(x)$ is the likelihood $\mathbf{L}_{simple} = \hat{P}_{\mathbf{M}}(x)$ according to a model $\mathbf{M}$ trained on the *ID* training points. In the simplest case, $\mathbf{M}$ is simply a left-to-right language model learnt on all our training sentences. Later, we discuss another class of models which can give a valid likelihood, which we name $\mathbf{L}_{gen}$.

## Likelihood Ratio

(Nalisnick et al. 2019; Choi and Jang 2018) found that likelihood is poor at separating out *OOD* examples; in some cases even assigning them higher likelihood than the *ID* test split. (Ren et al. 2019) make similar observations for detecting *OOD* DNA sequences. They then propose the concept of *likelihood ratio* or LLR based methods, which we briefly revisit here.

Let $\hat{P}_{\mathbf{M}}(x)$ and $\hat{P}_{\mathbf{B}}(x)$ denote the probability of $x$ according to the model $\mathbf{M}$ and a background model $\mathbf{B}$ respectively. $\mathbf{M}$ is trained on the training set $\{x_i\}_{i=1}^{i=m}$, while $\mathbf{B}$ is trained on noised samples from the training set. Let ${x_1}^i$ denote the prefix $x_1 x_2 \ldots x_{i-1}$. The LLR is derived as in Equation 2

$$LLR_{\mathbf{M},\mathbf{B}}(x) = \frac{\hat{P}_M(x)}{\hat{P}_B(x)}$$

$$LLR_{\mathbf{M},\mathbf{B}}(x) = \frac{\Pi_{i=1}^{i=|S|} \hat{P}_{\mathbf{M}}(x_i|{x_1}^i)}{\Pi_{i=1}^{i=|S|} \hat{P}_{\mathbf{B}}(x_i|{x_1}^i)}$$

$$\log LLR_{\mathbf{M},\mathbf{B}}(x) = \sum_{i=1}^{i=|S|} \log \hat{P}_{\mathbf{M}}(x_i|x_1^i) - \log \hat{P}_{\mathbf{B}}(x_i|x_1^i)$$

The intuition was that "surface-level" features might be causing the *OOD* points to be assigned a reasonable probability. The hypothesis is that the background model would capture these "surface-level" features which also persist after noising and remove their influence on being divided out. If it were so, $LLR_{\mathbf{M},\mathbf{B}}(x)$ would be a better choice for $\eta(x)$.

**How to introduce noise?** It is a common practice in vision to add noise or perturb images slightly by adding a Gaussian noise vector of small magnitude. Since natural language utterances are sequences of discrete words, this does not extend directly to them.

A simple alternative to introduce noise into natural language inputs is by random word substitution. Word substitution based noise has a long precedent of use, from negative sampling as in *word2vec* (Mikolov et al. 2013; Goldberg and Levy 2014) to autoencoder-like objectives like BERT (Devlin et al. 2019).

[4] https://github.com/thuiar/DeepUnkID

More specifically, with probability $p_{noise}$, we substitute each word $w$ with word $w'$ sampled from the distribution $\mathbf{N}(w')$. $p_{noise}$ is a hyperparameter to be tuned. We experiment with 3 different choices of $\mathbf{N}$:

1. UNIFORM: $\mathbf{N}(w') = \frac{1}{|W|}$ i.e each word $w' \in W$ is equally likely.
2. UNIGRAM: $\mathbf{N}(w') = \frac{f(w')}{\sum_{w \in W} f(w)}$ i.e a word $w'$ is sampled with probability proportional to its frequency $f(w')$. Using unigram frequency for the noise distribution is common practice in *noise contrastive estimation* (Dyer 2014).
3. UNIROOT: $\mathbf{N}(w') = \frac{\sqrt{f(w')}}{\sum_{w \in W} \sqrt{f(w)}}$ i.e a word $w'$ is sampled with probability proportional to the square root of its frequency. Using such smoothed versions of the unigram frequency distribution has precedent in other NLP tasks. For instance, in (Goldberg and Levy 2014)[5], the word2vec negative sampling uses $P(c) = \frac{{f_c}^{3/4}}{Z}$ to sample negative contexts, $c$, proportional to frequency, $f_c$.

**Choice of $\mathbf{B}$ architecture** Since this is hitherto the first work to extend LLR method for NLP, we try to use a simple and standard architecture for $\mathbf{B}$. We use a left-to-right LSTM language model (Sundermeyer, Schlüter, and Ney 2012) with a single layer. We vary the hidden state $\mathbf{B}_h \in \{64, 128, 256\}$. Note that $\mathbf{B}$ is non-class conditional - it does not use the labels in any way.

An additional point of consideration is that $\mathbf{B}$ should not have a very large number of parameters, or have large time complexity at test time. Even in this regard, a *LSTM* language model with a small state size is apt. We refer to approaches which use this architecture for $\mathbf{B}$ with +BACKLM

### Generative Classifier

Typical classification models estimate the conditional probability of the label given the input, i.e $P(y|x)$. An alternative paradigm learns to estimate $P(x|y)$, additionally estimating $P(y)$ from the training set label ratios. Using Bayes rule,

$$\begin{aligned} \text{argmax}\, P(y|x) &= \text{argmax}\, \frac{P(x|y)P(y)}{P(x)} \\ &= \text{argmax}\, P(x|y)P(y) \end{aligned}$$

Classifiers of this paradigm are called *generative classifiers*, in contrast to the typical *discriminative classifiers*.

(Yogatama et al. 2017) compare the two paradigms and found generative classifiers useful for a) High sample efficiency b) Continual Learning c) Explicit Marginal Likelihood. The last point is particularly useful for us since we can use the explicit marginal likelihood from the classifier $P(x)$ as our $\eta$ function. Specifically, we use the $P(x) = \sum_{y \in Y} P(x|y)P(y)$ term which is directly available from a trained generative classifier. Hereon, we refer to this as $L_{gen}$.

(Yogatama et al. 2017) also propose a deep architecture for generative text classifiers that consists of a shared unidirectional LSTM across classes and a label embedding matrix. The respective label embedding is concatenated to the current hidden state and a final layer is then applied on this vector to give the distribution over the next word. The per-word cross-entropy loss serves as the loss function. We use a similar architecture as illustrated in Figure 1.

## 3 Evaluation

For the threshold dependent measures, we tune our $\eta()$ function on the validation set. We use the following metrics to measure *OOD* detection performance:

- $FPR@k\%TPR$: On picking a threshold such that the *OOD* recall is k%, what fraction of the predicted *OOD* points are *ID*? FPR denotes False Positive Rate and TPR denotes True Positive Rate. Note that *Positive* here refers to the *OOD* class. We choose a high values of $k$, i.e 95. Note that lower this value, the better is our *OOD* Detection.
- $AUROC$ : Measures the area under the Receiver Operating Characteristic, also known as the *ROC* curve. Note that this curve is for the *OOD* class. (Hendrycks and Gimpel 2017) first proposed using this. Higher the value, better is our *OOD* Detection. This metric is threshold independent.
- $AUPR_{OOD}$ : Area under the Precision Recall Curve is another threshold independent metric, based on the Precision-Recall Curve. Unlike $AUROC$, $AUPR$ is insensitive to class imbalance (Davis and Goadrich 2006). The $AUPR_{OOD}$ and $AUPR_{ID}$ correspond to taking *ID* and *OOD* respectively as the positive class.

## 4 Datasets

We use two datasets for our experiments. The first, SNIPS, is a widely used, publicly available dataset, and does not contain actual *OOD* intents. The second, ROSTD, is a combination of a dataset released earlier (Schuster et al. 2019), with new *OOD* examples collected by us. We briefly describe both of these in order. Table 1 also provides useful summary statistics about these datasets:

### SNIPS

Released by (Coucke et al. 2018), SNIPS consists of $\approx 15,000$ sentences spread through 7 intent classes such as *GetWeather*, *RateBook* etc. As discussed previously, it does not explicitly include *OOD* sentences.

We follow the procedure described in (Lin and Xu 2019) to synthetically create OOD examples. Intent classes covering atleast $K\%$ of the training points in combination are retained as $ID$. Examples from the remaining classes are treated as *OOD* and removed from the training set. In the validation and test sets, examples from these classes are relabelled to the single class label *OOD*. Besides not being genuinely *OOD*, another issue with this dataset is that the validation and test splits are quite small in size at 700 each.

[5] Specifically, see the footnote concluding Page 2 in the paper

| Category | Example | % |
|---|---|---|
| Overtly Powerful Action | 1. send Ameena $ 25 from Venmo account<br>2. fix a pot of coffee | 20.55 |
| Action Memory | 1. What's the color of the paint I bought off Amazon<br>2. how much did I spend yesterday | 12.24 |
| Declarative Statement | 1. I learned some good words.<br>2. I always bookmark my favorite website to go back in it anytime.<br>3. all Star Wars movie are great | 8.74 |
| Underspecified Query | 1. On what website can I order medication?<br>2. how many jobs is having been lost | 33.94 |
| Speculative Question | 1. Can I do all of my Amazon shopping through the app?<br>2. when is the next episode of General Hospital | 6.91 |
| Subjective Question | 1. What color goes well with navy blue?<br>2. where can I learn something new every day? | 27.99 |

Table 1: We manually classify each *OOD* sentence in ROSTD into [1 or more] of 6 qualitative categories named self-explanatorily. More examples per category can be seen in Table 1 of the appendix.

| Statistic | ROSTD | SNIPS |
|---|---|---|
| Train-ID | 30521 | 13084 |
| Valid-ID | 4181 | 700 |
| Test-ID | 8621 | 700 |
| Actual OOD | 4590 | None |
| Unique Word Types | 11.5K | 11.4K |
| Unique Bigrams | 47.3K | 36.3K |
| Unique Trigrams | 80.8K | 52.2K |
| Mean Utterance Length | 6.85 | 6.79 |
| Number of *ID* classes | 12/3 (Coarse) | 7 |

Table 2: Dataset and Vocabulary Statistics contrasting ROSTD and SNIPS. Note that the *ID* part of ROSTD comes from the English portion of the publicly available data from (Schuster et al. 2019)

In §5, we report experiments on $K = 75$ and $K = 25$[6], both of which ratios were used in (Lin and Xu 2019). We refer to these datasets as SNIPS,75% and SNIPS,25% respectively. Since multiple *ID-OOD* splits of the classes satisfying these ratios are possible, our results are averaged across 5 randomly chosen splits.

## ROSTD

We release a dataset of $\approx 4590$ *OOD* sentences . These sentences were curated to be explicitly *OOD* with respect to the English split of the recently released dataset of intents from (Schuster et al. 2019) as the *ID* dataset. This dataset contained $43,000$ intents from 13 intent classes. We chose this dataset over *SNIPs* owing to its considerably larger size ($\approx 2.3$ times larger). The sentences were authored by human annotators with the instructions as described in the subsection **Annotation Guidelines**.

**Annotation Guidelines** We use human annotators to author intents which are explicity with respect to the English split of (Schuster et al. 2019). The requirements and instructions for annotation were as follows:

1. The *OOD* utterances were authored by several distinct English-speaking annotators from Anglophone countries.
2. The annotators were asked to author sentences which were both grammatical and semantically sensible as English sentences. This was to prevent our *OOD* data from becoming trivial by inclusion of ungrammatical sentences, gibberish and nonsensical sentences.
3. The annotators were well informed of existing intent classes to prevent them from authoring intents . This was done by presenting the annotators with 5 examples from the training split of each intent class, with the option to scroll for more through a dropdown.
4. After the first round of annotators had authored such intents, each intent was post-annotated as in-domain vs out-of-domain by two fresh annotators who were not involved in the authoring stage.
5. If both annotators agreed that the example was *OOD*, it was retained. If both agreed it was *ID*, it was discarded. In the event the two annotators disagreed, an additional, third annotator was asked to label the example and adjudicate the disagreement.
6. During post-processing, we removed utterances which were shorter than three words.

**Qualitative Analysis** We identify six qualitative categories which might be making the sentences *OOD*. We then manually assign each example into these categories. We summarize their distribution in Table 1. Note that since these categories are not mutually exhaustive, an example may get assigned multiple categories.

## Coarsening Labels

The *ID* examples from (Schuster et al. 2019), which we also use as the *ID* portion of ROSTD has hierarchical class labels [e.g $alarm/set\_alarm$, $alarm/cancel\_alarm$ and $weather/find$][7]. Hence, ROSTD has a large number of classes (12), not all of which are equally distinct from each other. To ensure that our results are not specific only to settings with this kind of hierarchical label structure, we also experiment with retaining only the topmost or most *"coarse"* label on each example. We refer to this variant with "coarsened" labels as ROSTD-COARSE.

# 5 Experiments

We compile the results of all our experiments in Table 3.

## Implementation

All experiments are averaged across 5 seeds. We use Pytorch 1.0 (Paszke et al. 2017) to implement models[8].

The checkpoint with highest validation F1 on the *ID* subset of the validation set is chosen as the final checkpoint for

[6] See appendix for the experimental results with $K = 25$ i.e SNIPS,25%.

[7] See (Schuster et al. 2019) for full list

[8] Code available at github.com/vgtomahawk/LR_GC_OOD

| Dataset | Model | $F_1 \uparrow$ | $FPR@95\%TPR \downarrow$ | AUROC ↑ | $AUPR_{OOD} \uparrow$ |
|---|---|---|---|---|---|
| ROSTD | MSP | 54.22 ± 4.01 | 100.00 ± 0.00 | 70.75 ± 3.70 | 55.68 ± 6.36 |
| | MSP,$\tau = 1e^3$ | 55.45 ± 4.19 | 60.48 ± 3.17 | 76.94 ± 4.01 | 59.64 ± 6.49 |
| | $-KL(P\|\mathbf{U})$ | 55.31 ± 3.89 | 60.15 ± 3.11 | 76.86 ± 3.85 | 59.54 ± 6.10 |
| | $-KL(P\|\mathbf{R})$ | 83.24 ± 2.78 | 21.31 ± 8.38 | 95.78 ± 1.30 | 90.32 ± 1.97 |
| | LOF | 64.46 ± 2.57 | 42.49 ± 3.49 | 81.39 ± 2.38 | 46.89 ± 3.50 |
| | LOF+LMCL | 85.97 ± 2.00 | 15.03 ± 5.42 | 95.60 ± 0.75 | 82.71 ± 9.17 |
| | $\mathbf{L}_{simple}$ | 81.38 ± 0.19 | 18.92 ± 0.56 | 95.42 ± 0.11 | 87.38 ± 0.41 |
| | $\mathbf{L}_{simple}$+BackLM+UNIFORM | 85.25 ± 0.72 | 36.65 ± 6.87 | 94.71 ± 0.49 | 91.10 ± 0.63 |
| | $\mathbf{L}_{simple}$+BackLM+UNIGRAM | 82.27 ± 0.74 | 42.16 ± 3.62 | 93.62 ± 0.43 | 89.30 ± 0.50 |
| | $\mathbf{L}_{simple}$+BackLM+UNIROOT | 87.42 ± 0.45 | 20.10 ± 5.25 | 96.35 ± 0.41 | 93.44 ± 0.37 |
| | $\mathbf{L}_{gen}$ | 86.25 ± 0.71 | 10.86 ± 1.08 | 97.42 ± 0.28 | 92.30 ± 0.99 |
| | $\mathbf{L}_{gen}$+BackLM+UNIFORM | 89.60 ± 0.56 | 13.71 ± 5.64 | 97.67 ± 0.35 | 95.49 ± 0.42 |
| | $\mathbf{L}_{gen}$+BackLM+UNIGRAM | **91.35 ± 2.62** | 10.55 ± 4.11 | 97.87 ± 0.49 | 95.86 ± 0.68 |
| | $\mathbf{L}_{gen}$+BackLM+UNIROOT | 91.17 ± 0.32 | **7.41 ± 1.88** | **98.22 ± 0.26** | **96.47 ± 0.29** |
| ROSTD-COARSE | MSP | 59.99 ± 19.01 | 26.00 ± 34.32 | 71.63 ± 15.55 | 64.32 ± 19.36 |
| | MSP,$\tau = 1e^3$ | 64.62 ± 15.31 | 64.46 ± 9.84 | 78.39 ± 11.92 | 66.89 ± 11.76 |
| | $-KL(P\|\mathbf{U})$ | 65.36 ± 15.49 | 65.39 ± 4.84 | 79.05 ± 11.40 | 67.79 ± 19.43 |
| | $-KL(P\|\mathbf{R})$ | 81.56 ± 8.51 | 17.78 ± 15.70 | 93.47 ± 6.25 | 87.49 ± 8.94 |
| | LOF | 62.39 ± 9.01 | 46.55 ± 17.56 | 78.07 ± 12.23 | 45.80 ± 12.21 |
| | LOF+LMCL | 84.28 ± 3.44 | 15.24 ± 4.70 | 95.19 ± 1.03 | 76.63 ± 2.53 |
| | $\mathbf{L}_{simple}$ | 80.48 ± 0.27 | 20.78 ± 0.71 | 95.2 ± 0.07 | 86.87 ± 0.13 |
| | $\mathbf{L}_{simple}$+BackLM+UNIFORM | 85.97 ± 0.65 | 30.65 ± 4.51 | 95.27 ± 0.47 | 91.98 ± 0.61 |
| | $\mathbf{L}_{simple}$+BackLM+UNIGRAM | 84.46 ± 0.62 | 31.79 ± 3.04 | 94.93 ± 0.32 | 91.22 ± 0.50 |
| | $\mathbf{L}_{simple}$+BackLM+UNIROOT | 88.25 ± 0.50 | 16.35 ± 1.32 | 96.82 ± 0.12 | 94.10 ± 0.20 |
| | $\mathbf{L}_{gen}$ | 86.67 ± 0.34 | 9.88 ± 0.44 | 97.58 ± 0.08 | 92.74 ± 0.29 |
| | $\mathbf{L}_{gen}$+BackLM+UNIFORM | 89.32 ± 0.30 | 8.04 ± 0.69 | 97.83 ± 0.15 | 95.27 ± 0.32 |
| | $\mathbf{L}_{gen}$+BackLM+UNIGRAM | 90.05 ± 0.73 | **6.69 ± 0.82** | 98.16 ± 00.15 | 95.61 ± 0.50 |
| | $\mathbf{L}_{gen}$+BackLM+UNIROOT | **90.14 ± 0.39** | 6.78 ± 0.60 | **98.30 ± 0.09** | **95.96 ± 00.37** |
| SNIPS, 75% | MSP | 81.58 ± 7.68 | **16.68 ± 18.06** | 93.51 ± 4.49 | 85.03 ± 6.19 |
| | MSP,$\tau = 1e^3$ | 83.94 ± 6.82 | 31.32 ± 30.25 | 94.30 ± 4.50 | 88.44 ± 5.72 |
| | $-KL(P\|\mathbf{R}) \approx -KL(P\|\mathbf{U})$ | 84.23 ± 7.22 | 29.28 ± 27.04 | 94.51 ± 4.38 | 88.71 ± 6.15 |
| | LOF | 66.07 ± 8.82 | 49.56 ± 13.49 | 79.65 ± 7.81 | 51.69 ± 13.08 |
| | LOF+LMCL | 76.24 ± 9.34 | 42.27 ± 20.64 | 90.37 ± 6.54 | 77.81 ± 10.53 |
| | $\mathbf{L}_{simple}$ | 63.51 ± 6.33 | 54.56 ± 12.13 | 81.72 ± 5.90 | 62.12 ± 13.28 |
| | $\mathbf{L}_{simple}$+BackLM+UNIFORM | 74.74 ± 3.25 | 44.60 ± 12.01 | 90.02 ± 2.24 | 80.08 ± 3.31 |
| | $\mathbf{L}_{simple}$+BackLM+UNIGRAM | 81.19 ± 3.53 | 27.00 ± 8.71 | 93.97 ± 1.85 | 87.57 ± 3.38 |
| | $\mathbf{L}_{simple}$+BackLM+UNIROOT | 78.75 ± 3.25 | 35.24 ± 11.08 | 92.66 ± 1.89 | 84.84 ± 3.36 |
| | $\mathbf{L}_{gen}$ | 67.31 ± 7.06 | 44.60 ± 18.53 | 85.17 ± 7.18 | 68.11 ± 14.29 |
| | $\mathbf{L}_{gen}$+BackLM+UNIFORM | 78.37 ± 6.60 | 29.28 ± 4.23 | 92.35 ± 2.77 | 82.84 ± 7.33 |
| | $\mathbf{L}_{gen}$+BackLM+UNIGRAM | **85.47 ± 6.90** | 18.48 ± 11.26 | **95.79 ± 2.67** | **90.98 ± 6.73** |
| | $\mathbf{L}_{gen}$+BackLM+UNIROOT | 81.91 ± 6.83 | 22.24 ± 6.26 | 94.15 ± 2.59 | 86.60 ± 7.03 |

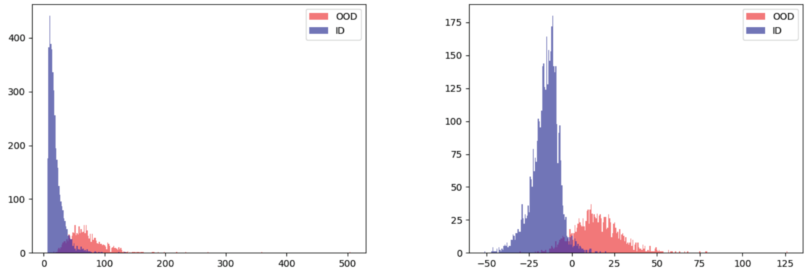
Table 3: Performance of the baseline methods and our proposed models on ROSTD, ROSTD-COARSE and SNIPS. ↓ (↑) indicates lower (higher) is better. We can see that the $\mathbf{L}_{gen}$+BackLM+<*Noise*> (where <*Noise*> is one of three noising schemes) approaches outdo their non LLR counterparts on most measures. For SNIPS, $-KL(P|\mathbf{R}) \approx -KL(P|\mathbf{U})$ since the training set is almost evenly distributed between the *ID* classes. We can also observe that the differences in performance between different approaches are much more observable on ROSTD as compared to SNIPS.

computing the other *OOD* evaluation metrics. For the label-agnostic approaches ($L_{simple}$), the checkpoint with lowest validation perplexity is chosen. For the +BackLM approaches, we use $p_{noise} = 0.5$. We also experimented with $p_{noise} \in \{0.1, 0.3, 0.7\}$, but find 0.5 works best.

**Base Classifier Architectures** For the discriminative classifier, we use a bidirectional LSTM [1-layer] with embedding size 100, projection layer of 100×300 (to project up embeddings), hidden size 300 and embeddings initialized with Glove (*glove.6B.100D*) (Pennington, Socher, and Manning 2014). Generative classifier approaches have similar architecture except that they are unidirectional and have additional label embeddings of dimension 20

**LOF implementation** We use the scikit-learn 0.21.2 implementation[9] (Pedregosa et al. 2011) of *LOF*. We fix the number of nearest neighbors to 20 but tune the contamination rate as a hyperparameter. We also corroborated over email correspondence with the authors of (Lin and Xu 2019) that they had used a similar hyperparameter setting for LOF.

[9] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html



(a) $\mathbf{L}_{gen}$

(b) $\mathbf{L}_{gen}$+BACKLM+UNIROOT

Figure 2: Effect of BACKLM+UNIROOT. In the right graph, we can see that the *OOD* has shifted considerably to the right then before and overlaps less with the *ID* set

## Observations

From Table 3, we see that $\mathbf{L}_{gen}$ outperforms uncertainty based and nearest neighbour approaches by a reasonable margin on both datasets. It is also significantly better than the language model likelihood based $\mathbf{L}_{simple}$. This validates our hypothesis that generative classifiers effectively combine the benefits of likelihood-based and uncertainty-based approaches.

Furthermore, LLR based approaches always outperform the respective likelihood-only approach, whether $\mathbf{L}_{simple}$ or $\mathbf{L}_{gen}$. Amongst different noising methods, the performance improvement is typically largest using the UNIROOT approach we proposed. For instance, on ROSTD,

$$\mathbf{L}_{gen} + \text{BACKLM} + \text{UNIROOT} > \mathbf{L}_{gen} + \text{BACKLM} + \text{UNIFORM}$$

$$\mathbf{L}_{gen} + \text{BACKLM} + \text{UNIFORM} > \mathbf{L}_{gen}$$

$$\mathbf{L}_{simple} + \text{BACKLM} + \text{UNIROOT} > \mathbf{L}_{simple} + \text{BACKLM} + \text{UNIFORM}$$

$$\mathbf{L}_{simple} + \text{BACKLM} + \text{UNIFORM} > \mathbf{L}_{simple}$$

A clear advantage of ROSTD which is clear from the experiments is that differences in performance between the various methods are much more pronounced when tested on it, as compared to SNIPS. On SNIPS, the simple MSP, $\tau = 1e^3$ baseline is itself able to reach $\approx$ 80-90% of the best performing approach on most metrics.

## 6 Conclusion

To the best of our knowledge, we are hitherto the first work to use an approach based on generative text classifiers for *OOD* detection. Our experiments show that this approach can outperform existing paradigms significantly on multiple datasets.

Furthermore, we are the first to flesh out ways to use likelihood ratio based approaches first formalized by (Ren et al. 2019) for *OOD* detection in NLP. The original work had tested these approaches only for DNA sequences which have radically smaller vocabulary than NL sentences. We propose UNIROOT, a new way of noising inputs which works better for NL. Our method improves two different likelihood based approaches on multiple datasets.

Lastly, we curate and plan to publicly release ROSTD, a novel dataset of *OOD* intents w.r.t the intents in (Schuster et al. 2019). We hope ROSTD fosters further research and serves as a useful benchmark for *OOD* Detection

## 7 Acknowledgements

We thank Tony Lin and co-authors for promptly answering several questions about their paper, and Sachin Kumar for valuable discussion on methods. We also thank Hiroaki Hayashi and 3 anonymous reviewers for valuable comments.

## References

Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; and Sander, J. 2000. LOF: identifying density-based local outliers. In *ACM sigmod record*, volume 29, 93–104. ACM.

Choi, H., and Jang, E. 2018. Generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*.

Coucke, A.; Saade, A.; Ball, A.; Bluche, T.; Caulier, A.; Leroy, D.; Doumouro, C.; Gisselbrecht, T.; Caltagirone, F.; Lavril, T.; et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Davis, J., and Goadrich, M. 2006. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, 233–240. ACM.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dyer, C. 2014. Notes on noise contrastive estimation and negative sampling. *arXiv preprint arXiv:1410.8251*.

Goldberg, Y., and Levy, O. 2014. Word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *Proceedings* *of the 34th International Conference on Machine Learning-Volume 70*, 1321–1330. JMLR. org.

Hendrycks, D., and Gimpel, K. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Hendrycks, D.; Mazeika, M.; and Dietterich, T. G. 2019. Deep anomaly detection with outlier exposure. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Lee, K.; Lee, H.; Lee, K.; and Shin, J. 2018a. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Lee, K.; Lee, K.; Lee, H.; and Shin, J. 2018b. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, 7167–7177.

Liang, S.; Li, Y.; and Srikant, R. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

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

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Nalisnick, E. T.; Matsukawa, A.; Teh, Y. W.; Görür, D.; and Lakshminarayanan, B. 2019. Do deep generative models know what they don't know? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. *OpenReview*.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12(Oct):2825–2830.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Ren, J.; Liu, P. J.; Fertig, E.; Snoek, J.; Poplin, R.; DePristo, M. A.; Dillon, J. V.; and Lakshminarayanan, B. 2019. Likelihood ratios for Out-of-Distribution Detection. *arXiv preprint arXiv:1906.02845*.

Schuster, S.; Gupta, S.; Shah, R.; and Lewis, M. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 3795–3805. Minneapolis, Minnesota: Association for Computational Linguistics.

Sundermeyer, M.; Schlüter, R.; and Ney, H. 2012. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

Yogatama, D.; Dyer, C.; Ling, W.; and Blunsom, P. 2017. Generative and discriminative text classification with recurrent neural networks. *arXiv preprint arXiv:1703.01898*.