Out-of-domain Detection based on Generative Adversarial Network

Seonghan Ryu[1,2], Sangjun Koo[2], Hwanjo Yu[2], and Gary Geunbae Lee[2]
[1]Language Understanding Lab, Artificial Intelligence Center, Samsung Research
[2]Computer Science and Engineering Department, Pohang University of Science and Technology
seonghan.ryu@samsung.com

Abstract

The main goal of this paper is to develop *out–of–domain* (OOD) detection for dialog systems. We propose to use only in–domain (IND) sentences to build a generative adversarial network (GAN) of which the discriminator generates low scores for OOD sentences. To improve basic GANs, we apply feature matching loss in the discriminator, use domain–category analysis as an additional task in the discriminator, and remove the biases in the generator. Thereby, we reduce the huge effort of collecting OOD sentences for training OOD detection. For evaluation, we experimented OOD detection on a multi–domain dialog system. The experimental results showed the proposed method was most accurate compared to the existing methods.

1 Introduction

Multi–domain dialog systems (Hakkani-Tur et al., 2016; Jiang et al., 2014; Lee et al., 2013; Ryu et al., 2015; Seon et al., 2014) should detect whether an input request is *out–of–domain* (OOD) because users do not know the exact coverages of those systems. One important problem of building OOD detection is the huge effort required to collect OOD sentences. This paper focuses on developing an accurate OOD detection method that requires only in–domain (IND) sentences for training, so this paper can reduce the effort of collecting OOD sentences.

For OOD detection, sentences would be represented in a continuous vector space in which IND cases are distinguished from OOD cases. Therefore, we use the existing *sentence embedding* method for OOD detection (Ryu et al., 2017). The authors train a recurrent neural network (RNN) for *domain–category analysis* task, in which one domain–category is assigned to an input sentence. Due to the similarity between OOD detection and domain–category analysis, the extracted features (i.e., representation) of the RNN contain information about domain–category. In addition, the word representations are pre–trained from a large unlabelled corpus, so the sentence embedding method has the advantage in understanding rare or unknown words that are likely to appear in OOD sentences.

Afterwards, we use the learned representations of IND sentences to train *one–class classifiers* that distinguish IND sentences from OOD sentences. We propose to use a *generative adversarial network* (GAN) (Goodfellow et al., 2014) that consists of a generator G and a discriminator D. We train D that distinguishes the IND sentences from the fake sentences generated by G, so we expect D to reject OOD sentences. We apply three modifications to improve basic GANs. To the best of our knowledge, this is the first study that uses GANs to solve OOD detection.

2 Related Work

Lane et al. (2007) proposed an *in–domain verification* method. The authors first build a basic binary classifier for each domain, and then build a meta classifier that takes the scores by the basic binary classifiers as input. However, in our experiment, many OOD sentences were misclassified into IND because OOD sentences were not in the negative examples of the classifiers. Therefore, the confidence scores of the basic binary classifiers are not sufficiently reliable evidences of OOD. Also, understanding rare or unknown words remains a problem because bag–of–words model is used.

Ryu et al. (2017) proposed an *autoencoder–* based method. The authors use neural sentence embedding that has the advantage in representing rare or unknown words. Based on those distributed sentence representations, an autoencoder

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 714–718
Brussels, Belgium, October 31 - November 4, 2018.

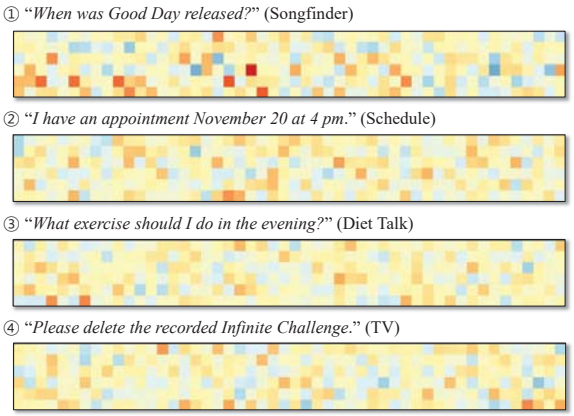

Figure 1: Distributed representations of IND sentences.

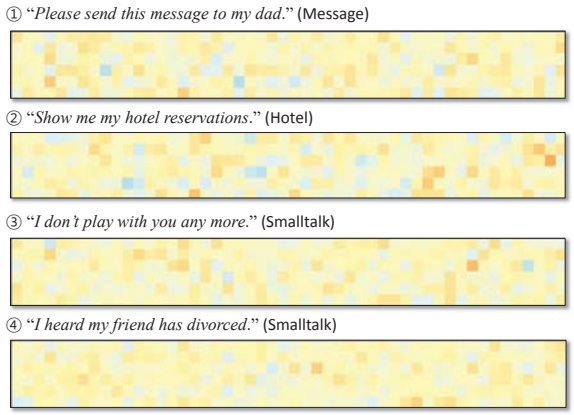

Figure 2: Distributed representations of OOD sentences.

is trained on IND sentences. The autencoder will have low reconstruction errors for IND sentences, so an input sentence can be classified into either IND or OOD. However, the autoencoder–based method has a limitation in expandability. When the weights are initialized carefully and regularization techniques are applied, the trained autoencoder reconstructs any input accurately. This result means that the reconstruction errors by the *ideal* autoencoder are not reliable evidence of OOD, so in OOD detection, autoencoders have little room for improvement.

3 Methods

As we discussed in Section 1, we use the sentence embedding to represent sentences in an 300–dimensional continuous vector space. We propose to use a GAN to OOD detection; a GAN consists of two adversarial components: generator G and discriminator D. G generates artificial data to deceive D. D distinguishes real data from the artificial data generated by G. GAN is an unsupervised algorithm because learning G and D does not require labels. Standard GANs are trained based on the objective function $V(D,G)$ as

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log 1 - D(G(\boldsymbol{z}))]. \quad (1)$$

So GAN is a minimax two–player game because G minimizes $V(D,G)$, and D maximizes $V(D,G)$.

We propose to use GANs to obtain a one–class classifier for OOD detection. When we train G to generate sentences similar to IND sentences and D to classify real IND sentences and fake sentences generated by G, we expect D to reject OOD sentences. Therefore, we use the low confidence score by D about an input sentence as the evidence that the sentence is OOD.

Let $p_{\mathbf{z}}(\boldsymbol{z})$ be a continuous uniform distribution $(-1,1)$. We define G that generates fake data $G(\mathbf{z}) \in \mathrm{R}^m$ from input noise $\mathbf{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})$, $\mathbf{f}$ that extracts features from either real data $\mathbf{x}$ of $(\mathbf{x},\mathbf{y}) \sim p_{\text{data}}(\mathbf{x},\mathbf{y})$ or $G(\mathbf{z})$, and D that measures the probability of either $\mathbf{f}(\mathbf{x})$ or $\mathbf{f}(G(\mathbf{z}))$ from the real data as

$$G(\mathbf{z}) = \sigma_h(\mathbf{W}_g\mathbf{z}), \quad (2)$$
$$\mathbf{f}(\mathbf{x}) = \sigma_h(\mathbf{W}_h\mathbf{x} + \mathbf{b}_f), \quad (3)$$
$$\mathbf{f}(G(\mathbf{z})) = \sigma_h(\mathbf{W}_h G(\mathbf{z}) + \mathbf{b}_f), \quad (4)$$
$$D(\mathbf{f}(\mathbf{x})) = \sigma_g(\mathbf{W}_d\mathbf{f}(\mathbf{x}) + \mathbf{b}_d), \quad (5)$$
$$D(\mathbf{f}(G(\mathbf{z}))) = \sigma_g(\mathbf{W}_d\mathbf{f}(G(\mathbf{z})) + \mathbf{b}_d), \quad (6)$$

where $\mathbf{W}_g$, $\mathbf{W}_f$, and $\mathbf{W}_d$ are weight matrices, $\mathbf{b}_f$ and $\mathbf{b}_d$ are bias vectors. So we define two objective functions

$$L_{\mathrm{D}} = -\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{\text{data}}(\mathbf{x},\mathbf{y})}[\log D(\mathbf{f}(\mathbf{x}))] - \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(\mathbf{f}(G(\mathbf{z}))))], \quad (7)$$
$$L_{\mathrm{G}} = -\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log D(\mathbf{f}(G(\mathbf{z})))]. \quad (8)$$

Because $G(\mathbf{z})$ continuously changes during the training, f of traditional GAN also changes. Thus we define $C \in \mathbb{R}^{|D|}$ that computes domain–category of $\mathbf{f}(\mathbf{x})$ as

$$C(\mathbf{f}(\mathbf{x})) = \mathrm{softmax}(\mathbf{W}_c\mathbf{f}(\mathbf{x}) + \mathbf{b}_c), \quad (9)$$

where $\mathbf{W}_c$ is a weight matrix and $\mathbf{b}_c$ is a bias vector. We expect f trained by the losses of both D

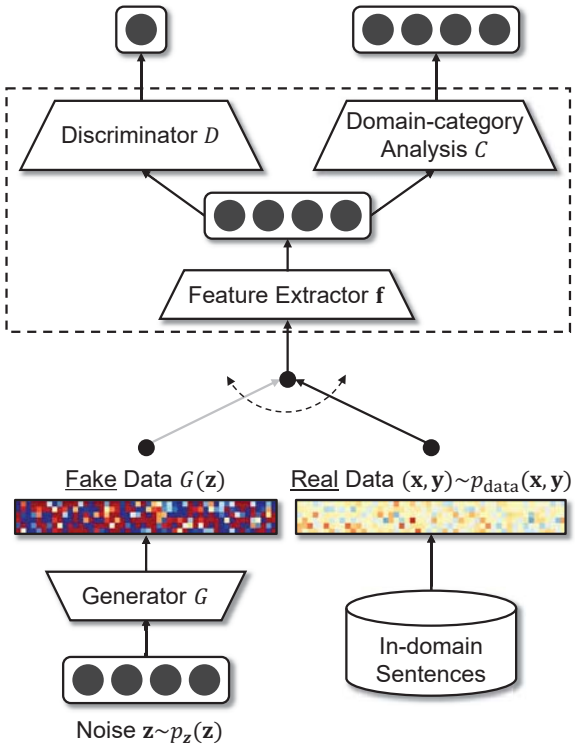

Figure 3: Generative adversarial network for out–of–domain sentence detection.

and C to be more stable than f trained by the loss of only D, so we define an objective function

$$L_{\text{C}} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{\text{data}}(\mathbf{x},\mathbf{y})}[H(C(\mathbf{f}(\mathbf{x})), \mathbf{y})], \quad (10)$$

where $H(p, q)$ is the categorical cross entropy and $\mathbf{y}$ is the true domain–category.

In addition, GAN suffers from a *mode collapse* problem in which G generates samples with a low variance. To solve the problem, we remove the biases in the generator because the G was trained to use the biases mainly instead of the weights to generate data.

Second, we apply feature matching (Salimans et al., 2016). The authors say "Instead of directly maximizing the output of the discriminator, the new objective requires the generator to generate data that matches [sic] the statistics of the real data, where we use the discriminator only to specify the statistics that we think are worth matching". So G is trained to generate high variance sentence $G(\mathbf{z})$ by additional objective function

$$L_{\text{f}} = ||\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{\text{data}}(\mathbf{x},\mathbf{y})}\mathbf{f}(\mathbf{x}) - \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}\mathbf{f}(G(\mathbf{z}))||_2^2. \quad (11)$$

Based on our design of GAN, we train D, C, $\mathbf{f}$, and G as Algorithm 1. To implement our GAN for one–class classification, we use the Tensorflow library (Abadi et al., 2015). We train our models by using Adam (Kingma and Ba, 2015) optimizer with a mini–batch size of 256 and an initial learning rate of 0.01 that is decreased linearly during 500 epochs. All weights are initialized from a zero–centered Normal distribution with standard deviation 1.0.

Algorithm 1 Training process of GAN for OOD detection.

for number of training iterations **do**
  Sample real data $(\mathbf{x}, \mathbf{y}) \sim p_{\text{data}}(\mathbf{x}, \mathbf{y})$.
  Sample noise $\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$.
  Update D, C, and $\mathbf{f}$ based on $L_{\text{D}} + L_{\text{C}}$.
  Sample noise $\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$.
  Update G based on $L_{\text{G}} + L_{\text{f}}$.
end for

4 Experiments

4.1 Data Set

We experimented on a data set of 6,268 Korean sentences. We collected 706 OOD sentences about three domains: hotel, message, and smalltalk; and 5,562 IND sentences about fourteen domains: airplane, alarm, bus, call, car navigation, diet talk, exchange, general, schedule, songfinder, time, train, and TV control. We used eighty percent of the IND sentences to train the models; we used the remaining IND sentences and all OOD sentences for testing.

4.2 Evaluation Metrics

We use *equal error rate* (EER) to represent the accuracy of OOD detection (Lane et al., 2007). EER is the error rate at which false acceptance rate

$$\text{FAR} = \frac{\text{Number of } \textit{accepted} \text{ OOD sentences}}{\text{Number of OOD sentences}} \quad (12)$$

and false rejection rate

$$\text{FRR} = \frac{\text{Number of } \textit{rejected} \text{ ID sentences}}{\text{Number of ID sentences}} \quad (13)$$

are equal.

We performed each experiment 20 times, and recorded the average EER of OOD detection.

4.3 Compared Methods

We have three variations of vanilla GAN: to remove the biases, to add domain–category analy-

Table 1: EERs [%] ± s.d. (n = 20) of OOD detection.

Method	EER
Local outlier factor	13.33
One–class SVM	13.76
Autoencoder	9.24 ± 0.43
GAN with biases	15.93 ± 5.82
GAN	9.18 ± 0.30
GAN with DCA task	9.17 ± 0.40
GAN with FM loss	9.04 ± 0.30
GAN with DCA task and FM loss	8.96 ± 0.34

sis (DCA) task, and to add feature matching (FM) loss. So we assessed five settings about GAN.

We compare our method to three one–class classifiers. (1) *Local outlier factor* (Breunig et al., 2000) compares the local density of a point to the local densities of its neighbors, and considers the point that has lower density than their neighbors as an outlier. The local density of a point is defined by the distance to its nearest neighbors. (2) *One–class support vector machines* (One–class SVMs) (Schölkopf et al., 2001) that treats the origin as a negative example to learn a decision function. (3) *Autoencoder* is explained in Section 2.

4.4 Results

In the experiments (Table 1), the best EER (8.96%) was obtained by the GAN in which all three of our variations are applied ($p < 0.05$). This result means that (1) removing the biases and using the feature matching prevented the generator from mode collapse problem and (2) using domain–category analysis as an auxiliary task stabilized the training of feature extractor. Compared to the other one–class classification methods including the autoencoder, the proposed GAN was most accurate ($p < 0.05$), so we can say that the discriminator scores of the GAN are reliable evidence for OOD detection (Table 2).

5 Conclusion

In this paper, we aimed at building OOD detection without OOD sentences for training. We proposed to use the discriminator of a GAN, which is trained on only IND sentences. The proposed method outperformed the existing methods in our data set.

Table 2: Average score [%] by the discriminator of the GAN.

Data	Score
IND training sentences	98.69 (± 3.36)
Fake sentences $G(\mathbf{z})$	0.20 (± 1.70)
IND test sentences	88.59 (± 28.09)
OOD test sentences	8.04 (± 23.10)

To train the GAN, we used the distributed sentence representations computed from the pre–trained sentence embeddings instead of symbolic sentences. However, we think the limitation of the pre–trained sentence embeddings can be overcome by building a GAN that generates symbolic sentences and discriminates them.

Acknowledgements

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2018-2015-0-00742) supervised by the IITP(Institute for Information & communications Technology Promotion)

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, ..., Devin, Moore, Vanhoucke, Warden, 2015. TensorFlow: Large–scale machine learning on heterogeneous distributed systems. Software available from tensorflow.org.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying density–based local outliers. In *Proceedings of ACM SIGMOD*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of NIPS*.

Dilek Hakkani-Tur, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi–domain joint semantic frame parsing using bi-directional RNN-LSTM. In *Proceedings of Interspeech*.

Ridong Jiang, Rafael E. Banchs, Seokhwan Kim, Kheng Hui Yeo, Arthur Niswar, and Haizhou Li. 2014. Web–based multimodal multi-domain spoken dialogue system. In *Proceedings of IWSDS*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Ian Lane, Tatsuya Kawahara, Tomoko Matsui, and Satoshi Nakamura. 2007. Out–of–domain utterance detection using classification confidences of multiple topics. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 15:150–161.

Donghyeon Lee, Minwoo Jeong, Kyungduk Kim, Seonghan Ryu, and Gary Geunbae Lee. 2013. Unsupervised spoken language understanding for a multi–domain dialog system. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 21:2451–2464.

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 Recogn. Lett.*, 88:26–32.

Seonghan Ryu, Jaiyoun Song, Sangjun Koo, Soonchoul Kwon, and Gary Geunbae Lee. 2015. Detecting multiple domains from users utterance in spoken dialog system. In *Proceedings of IWSDS*.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training GANs. In *Proceedings of NIPS*.

Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. 2001. Estimating the support of a high–dimensional distribution. *Neural Comput.*, 13:1443–1471.

Choong-Nyoung Seon, Hyunjung Lee, Harksoo Kim, and Jungyun Seo. 2014. Improving domain action classification in goal–oriented dialogues using a mutual retraining method. *Pattern Recogn. Lett.*, 45.