ODIST: Open World Classification via Distributionally Shifted Instances

Lei Shu, Yassine Benajiba, Saab Mansour and **Yi Zhang**
Amazon AWS AI
{leishu, benajiy, saabm, yizhngn}@amazon.com

Abstract

In this work, we address the open-world classification problem with a method called ODIST(open world classification via distributionally shifted instances). This novel and straightforward method can create out-of-domain instances from the in-domain training examples with the help of a pre-trained language model. Experimental results show that ODIST performs better than state-of-the-art decision boundary finding method.

1 Introduction

In the supervised learning setting, it is generally assumed that test set data points will be organized along the same classes observed during training. This assumption, however, proves unreliable in many applications, especially in dynamic and open environments. For instance, Zhang et al. (2021) show that an intent classifier performs rather poorly in a dialogue system when the user expresses intents unobserved in the training dialogues. In an open environment, the ideal classifier should classify incoming data to the correct existing classes that appeared in training and detect those examples that do not belong to any existing classes. Such classifier is thus described as *open set recognition* (Scheirer et al., 2013) or *open world classification* (Fei and Liu, 2016).

The existing research to achieve this capability in natural language processing (NLP) and computer vision (CV) mainly focuses on decision boundary finding. Schölkopf et al. (2001); Tax and Duin (2004); Fei et al. (2016) use SVM to detect the negative classified examples. Scheirer et al. (2013) introduce the concept of open-space risk in CV. Jain et al. (2014); Scheirer et al. (2014) propose a series of Weibull-calibrated SVM to reduce the open space risk further. Recent research shows that it is also possible to use deep neural networks to capture advanced features from the data (Lin and Xu, 2019). In CV, Bendale and Boult (2016) train a multi-class classifier and take the outputs of the penultimate layer to fit Weibull distribution. Hendrycks and Gimpel (2017) reject the low confidence samples with the threshold based on the probability of softmax distribution. Liang et al. (2018) add a temperature scaling on the softmax function to get a calibrated softmax score.In NLP, Shu et al. (2017) adopt the sigmoid function to learn the one-vs-all classifier and calculate the confidence threshold by fitting training data to Gaussian statistics. Zhang et al. (2021) propose to learn the adaptive decision boundary (ADB). ADB performs best among all the above methods on the open text classification.

Besides adjusting the decision boundary on the feature space learned from in-domain data, a good feature space representing both in-domain and novel out-of-domain (OOD) examples is also essential for novelty detection, namely: *open representation learning*. We can illustrate this approach with the following NLP example: Let us assume that we have only learned features for “it is red" (for cherries) and “it is yellow" (for bananas) for a fruit classification task. The problem we are trying to overcome manifests when the model is exposed to a blueberry during testing. Since it has not seen the class during training, it does not possess a proper method to extract features for “blue”. Ideally, we want a representation learning approach that can compute such a representation instead of using the representation of “red” or “yellow”. The straightforward solution is to explore some examples with “blue” during model training, although a blueberry does not exist as a class for in-domain training. However, in real-world applications, we do not foresee the OOD examples that would come in the future. Similarly, for CV, recent work (Tack et al., 2020) augment distributionally shifted images by rotating/flipping the original image and pretrain an image representation space for novelty detection. Inspired by the work, we propose a novel and sim-

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 3751–3756
November 7–11, 2021. ©2021 Association for Computational Linguistics

ple distributionally shifted data creation method for NLP. And then, we train a classifier on in-domain training examples and distributionally shifted examples. Such a classifier can work with existing decision boundary finding methods for further open space risk reduction.

Related Works: Besides the works(Shu et al., 2018; Xu et al., 2019) in open-world learning, our work is also related to data augmentation. In CV, Chen et al. (2020) propose a simple image pretraining method based on data augmentation. In NLP, Wu et al. (2020); Lewis et al. (2020a) pretrain language model by contrastive learning on augmented data. Wu et al. (2020) propose word/span deletion, word reorder, word replacement and Lewis et al. (2020a) use paraphrasing method to augment examples. Differently from what we explore in this paper, these works focus on similar instances instead of OOD examples.

In this work, we take advantage of the recent success of pretrained language models. We use the sequence-to-sequence language model BART (Lewis et al., 2020b) for distributionally shifted example creation. BART can fill the masked sentences by generation. Furthermore, we use the finetuned BART [1] on MNLI (Williams et al., 2018) for predicting the relationship between the original text and augmented examples for filtering.

2 Methodology

Problem Definition: We define a training data set as $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$ composed of n examples where the i-th document $\mathbf{x}_i$ is associated with one of the m *seen* classes $y_i \in \{l_1, l_2, \dots, l_m\}$. In the canonical open-world classification setting, a model learns from the training data and either classifies the test instance to one of the m *seen* classes or reject it as *unseen* (denoted by l_0), i.e., it does not belong to any of the seen classes. Therefore, it is a $(m + 1)$-class classifier.

In our setting, we create distributionally shifted instances $D^A = \{(\mathbf{x}_1^A, l_{m+1}), \dots, (\mathbf{x}_k^A, l_{m+1})\}$ by augmenting the training set D of seen classes into a new augmented class l_{m+1}. We learn a model $f(\mathbf{x})$ using both in-domain training examples in D and the OOD examples in D^A. During prediction, a data point is classified to either: one of the m seen classes from D; or l_0 either because it is classified as l_{m+1} (from D^A) or because all m seen classes reject it. Therefore, our method is a $(m + 2)$-class classifier $f(\mathbf{x})$ with the classes $\mathcal{C} = \{l_1, l_2, \dots, l_m, \underline{l_0, l_{m+1}}\}$.

This section introduces the creation process of distributionally-shifted instances, model training, and testing procedure.

2.1 Distributionally Shifted Data Augmentation

As previously discussed, we do not have the OOD data or unseen classes' examples ready at the training time in most real-world scenarios. The goal of distributionally shifted data augmentation is to create OOD examples from the seen classes' examples. Thus, the model can learn discriminative features for OOD detection and in-domain classification. Distributionally shifted data augmentation inherits from the span replacement (Wu et al., 2020). As shown in Figure 1, there are four steps, namely: 1) **chunk** the example $\mathbf{x}$ in the in-domain training data into pieces; 2) **mask** each piece iteratively to create masked sentences; 3) **replace** the <mask> tokens with predicted tokens from the pre-trained generative language model BART to obtain the augmented examples; 4) **select** the augmented examples by predicting with the the fine-tuned BART on MNLI whether the original and augmented pair is contradiction relation as qualified OOD examples.

The outcome of this approach is a list of qualified OOD examples $\{\mathbf{x}_i^A, \dots\}$ as the pink examples in Figure 1. Our motivation to choose span replacement instead of the standard data augmentation methods: deletion, reorder, paraphrase, and word replacement is the OOD rate among the augmented examples. The reorder and paraphrase contributes in-domain examples. Word deletion and replacement have lower OOD rates than span replacement. As the example shown in Figure 1, span replacement has only $1/3$ of the augmented examples seems out-of-domain. This suggests that most tokens in the examples do not decide the semantic or class label of the examples.

2.2 Open Representation Learning

After preparing the OOD examples $\mathcal{D}^A = \{(\mathbf{x}_1^A, l_{m+1}), \dots, (\mathbf{x}_k^A, l_{m+1})\}$, we use them together with the in-domain training examples D for supervised $(m + 1)$-class classification. The class label space is $\mathcal{Y} = \{l_1, \dots, l_m, l_{m+1}\}$. Let f_E denote the encoder network, Linear$(\cdot)$ is a linear mapping function that maps a representation $\mathbf{r}$ to a $(m+1)$-dimension logits and Softmax$(\cdot)$. The

[1] https://github.com/pytorch/fairseq/tree/master/examples/bart

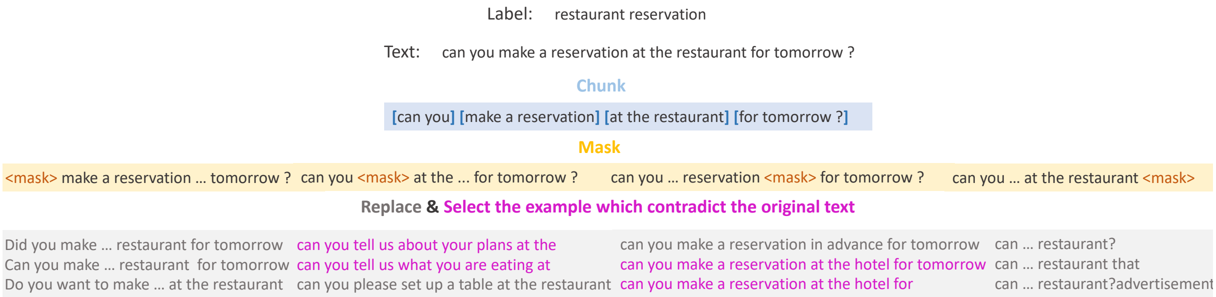
Label: restaurant reservation
Text: can you make a reservation at the restaurant for tomorrow ?
Chunk
[can you] [make a reservation] [at the restaurant] [for tomorrow ?]
Mask
<mask> make a reservation … tomorrow ?
can you <mask> at the … for tomorrow ?
can you … reservation <mask> for tomorrow ?
can you … at the restaurant <mask>
Replace & Select the example which contradict the original text
Did you make … restaurant for tomorrow
Can you make … restaurant for tomorrow
Do you want to make … at the restaurant
can you tell us about your plans at the
can you tell us what you are eating at
can you please set up a table at the restaurant
can you make a reservation in advance for tomorrow
can you make a reservation at the hotel for tomorrow
can you make a reservation at the hotel for
can … restaurant?
can … restaurant that
can … restaurant?advertisement

Figure 1: The creation process of distributionally-shifted instances: chunk, mask, replace and filter. The examples in pink color are the final distributionally-shifted instances. (Best view in color mode)

loss function $\mathcal{L}$ is the cross-entropy loss.

$$\begin{aligned} \mathbf{r}_i &= f_E(\mathbf{x}_i), \\ P(\mathbf{x}_i) &= \text{Softmax}\Big(\text{Linear}(\mathbf{r}_i)\Big), \\ \mathcal{L} &= \sum_{i=1}^{n+k} -\log P(\mathbf{x}_i). \end{aligned} \tag{1}$$

2.3 Rejection

Here we present the method for identifying unseen examples during testing/inference.

Given the class prediction $\tilde{y}$ for the example $\mathbf{x}$ from the $(m+1)$-class classifier described in Section 2.2, the method applies the decision boundary learning method upon the trained multi-class classifier to further reduce the open space risk. Here, we use the SOTA adjustable decision boundary (ADB) (Zhang et al., 2021) as the boundary finding method. The ADB method aims to learn euclidean distance decision boundaries for every seen class. After training a multi-class classifier, ADB feeds the training examples $\mathbf{x}_i$ back to the model and gets its representation $\mathbf{r}_i$. Based on the represention and class label $\{(\mathbf{r}_1, y_i), \dots, (\mathbf{r}_n, y_n))\}$, it calculates the centroids for each seen class $\{\mathbf{c}_1, \dots, \mathbf{c}_m\}$, and then learns the radius of the boundaries $\{b_1, \dots, b_m\}$ by tightening same-class's representations to its class-centroid.

Considering we use both in-domain training examples and distributionally shifted instances as input for the model, we inject them to get $(m+1)$-centroids $\{\mathbf{c}_1, \dots, \mathbf{c}_m, \mathbf{c}_{m+1}\}$ and learn $(m+1)$ boundaries $b = \{b_1, \dots, b_m, b_{m+1}\}$ for $(m+1)$ classes including m seen classes and the augmented class. The testing example is treated as a rejection example if it is out of all decision boundaries or belonging to the augmented class.

$$\hat{y} = \begin{cases} l_0 & \text{if } \tilde{y} = l_{m+1}, \\ l_0 & \text{elif } \forall j, 1 \leq j \leq m+1, \|\mathbf{r} - \mathbf{c_j}\| \geq b_j, \\ \tilde{y} & \text{otherwise .} \end{cases} \tag{2}$$

3 Experiments

We evaluate our method on three datasets: **Banking** (Casanueva et al., 2020) [2], **OOS** (Larson et al., 2019) [3] and **StackOverflow** (SO) (Xu et al., 2015) [4]. We follow ADB (Zhang et al., 2021) and split the datasets into training, validation, and testing. Furthermore, we create distributionally shifted examples on the training splits. The testing examples cover all classes in the datasets. The unseen classes' examples are treated as the rejection class l_0. The details of the datasets are in Table 4. Following (Shu et al., 2017; Zhang et al., 2021), we experiment with three portions of 25%, 50%, and 75% from all the classes as seen classes. For distributionally shifted instance creation, we use NLTK [5] to chunk the sentence and set BART to predict top-3 candidates with a beam size of 5. Regarding the model architecture and training, we keep ADB [6] setting that utilizes BERT (Devlin et al., 2019) as the base for multi-class classification. We use the NVIDIA Tesla V100 GPU. In representation learning, we use all qualified distributional-shift instances associated with the seen classes and maintain class balance in a batch. The training batch is 128, and the learning rate is 2e-5. For boundary learning, the learning rate is 0.05. We report the averaged scores and standard deviation on five random seeds.

ODIST is our proposed solution that includes the distributionally shifted instances in Sec. 2.1, open representation learning in Sec. 2.2 and decision boundaries learning in Eq. 2. Its variant

[2] https://github.com/PolyAI-LDN/task-specific-datasets
[3] https://github.com/clinc/oos-eval
[4] https://archive.org/details/stackexchange
[5] https://www.nltk.org/
[6] https://github.com/thuiar/Adaptive-Decision-Boundary

		25%			**50%**			**75%**		
Dataset	Method	Unseen	Seen	Acc	Unseen	Seen	Acc	Unseen	Seen	Acc
Banking	ADB	84.56	70.94	78.85	78.44	80.96	78.86	66.47	86.92	81.08
	ODIST	**87.11**±2.09	**72.72**±1.08	**81.69**±1.43	**81.32**±1.54	**81.79**±0.81	**80.90**±1.15	**71.95**±3.26	**87.20**±1.06	**82.79**±1.58
OOS	ADB	91.84	76.80	87.59	88.65	85.00	86.54	83.92	88.58	86.32
	ODIST	**93.42**±1.39	**79.69**±2.53	**89.79**±1.99	**90.62**±0.71	**86.52**±0.87	**88.61**±0.82	**85.86**±0.96	**89.33**±0.53	**87.70**±0.74
SO	ADB	90.88	78.82	86.72	87.34	85.68	86.4	73.86	86.80	82.78
	ODIST	**94.41**±1.36	**83.18**±2.54	**91.53**±1.96	**89.57**±1.04	**87.13**±1.41	**88.52**±1.26	**75.21**±1.23	**87.66**±0.87	**83.75**±0.94

Table 1: Results of ODIST (including standard deviation) and ADB. ADB's scores are from the original paper (Zhang et al., 2021).

		25%			**50%**			**75%**		
Dataset	Method	P	R	F_1	P	R	F_1	P	R	F_1
Banking	ODIST	94.90	80.51	87.11	85.36	77.65	81.32	64.12	81.84	71.95
	-DB	95.46	8.28	15.22	89.93	9.56	17.19	68.25	11.32	19.41
OOS	ODIST	97.15	90.00	93.42	92.63	88.70	90.62	85.59	86.14	85.86
	-DB	99.52	37.53	53.93	98.10	33.76	49.94	96.91	21.02	34.45
SO	ODIST	93.95	94.87	94.41	83.05	97.20	89.57	61.63	96.47	75.21
	-DB	95.64	19.00	31.70	90.44	17.97	29.98	73.42	10.87	18.93

Table 2: Ablation study: the precision(P), recall(R) and F_1 score of unseen examples.

	P	R	F_1
ODIST-DB	**99.52**	**37.53**	**53.93**
ODIST-DB-Select	98.43	33.95	50.48
Word Delete 50%	98.51	7.26	13.52
Word Reorder 50%	96.61	5.0	9.5

Table 3: Ablation study: different data augmentation methods' the precision(P), recall(R) and F_1 score of unseen examples on OOS 25% setting.

	Banking	OOS	SO
Class	77	150	20
Train	9003	15000	12000
Valid	1000	3000	2000
Test	3080	5700	6000
Shift	127092	186219	143831

Table 4: Details of the datasets and distributionally shifted instances (Shift).

ODIST-DB does not use any decision boundaries and treats the samples predicted to the augment class as rejected, as shown in Eq. 3.

$$\tilde{y} = \text{argmax}\Big(\text{Linear}(\mathbf{r})\Big),$$
$$\hat{y} = \begin{cases} l_0 & \text{if } \tilde{y} = l_{m+1}, \\ \tilde{y} & \text{otherwise .} \end{cases} \tag{3}$$

We compare our method to **ADB** that trains a (m)-class classifier on the in-domain training examples and learns decision boundaries for m seen classes. It is the SOTA method in open text classification. We report the F_1 score for the unseen class l_0, averaged F_1 score for seen classes, and accuracy of all test data. The unseen class's precision, recall, and F_1-score are reported regarding the ablation study between ODIST and ODIST-DB. We also compare augmentation methods without decision boundary but using Eq. 3 that directly treat the examples predicted into the class l_{m+1} as rejected. The precision, recall, and F_1-score of the unseen class on the OOS 25% setting are reported. The compared methods are: **Word Delete 50%** that randomly deletes 50% words in the original sentence, **Word Reorder 50%** that reorders 50% words in the original sentence; **ODIST-DB-Select**, which is the span-replacement proposed in Section 2.1 without the last selection step; and ODIST-DB that is our proposed data augmentation method.

Result Analysis: As shown in Table 1, we notice that ODIST performs better than the ADB in all scenarios. This supports that distributionally shifted instances can help open-world classification. It is promising to see the performance improvement on both unseen's and seen's examples. This suggests that distributionally shifted instances help model learning features for both in-domain classification and novelty detection. We notice that the performance improvement decrease with the increase of seen ratio. It is because there are more training examples for feature learning when the seen ratio is high. Distributionally shifted instances are more helpful in low seen-ratio scenarios.

In Table 2, ODIST is compared to ODIST-DB. The recall scores of ODIST-DB are low. This suggests that the diversity of distributionally shifted instances is limited, and they cannot cover all OOD test samples. It is because of the mask portion and Bart. On the other hand, the precision scores are high. This shows the OOD quality of the distributionally shifted instances. ODIST can achieve a high recall with a slight performance drop on the precision with the decision boundaries. There is a decreasing trend of precision with the increase of seen ratio. This is because our current filter mechanism compares the augmented example to the original one. With more seen classes, the augmented examples are likely similar to other classes.

We compare ODIST-DB to another three data augmentation methods: ODIST-DB-Select, Word Delete 50%, and Word Reorder 50%. In Table 3, ODIST-DB and ODIST-DB-Select have much higher recall scores of the unseen class than Word Delete 50% and Word Reorder 50%. This suggests that Word Delete 50% and Word Reorder 50% cannot produce distributional-shift points and enrich discriminative features. Span-replacement-based methods (ODIST-DB and ODIST-DB-Select) inject new text spans that help open representation learning. All methods have good performances on the precision of unseen class though some methods mix in-distribution and OOD in their augmented examples. It is because we ensure class balance in a batch during open representation learning and bad augmented examples have lower weight in the loss than gold data (in-distribution training data). However, we still can observe that ODIST-DB has the highest precision. This suggests that the 'select' step in distributional-shift data augmentation is helpful. One venue for future work is to efficiently and effectively create diverse augmented data.

4 Conclusion

In this paper, we study the open-world classification problem. Differently from existing research, we propose to learn an open representation. To achieve that goal, we propose a novel and simple method to create distributionally shifted instances from the training examples. The experimental results show that the method is effective and improves over SOTA results on three classification datasets.

References

Abhijit Bendale and Terrance E Boult. 2016. Towards open set deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1563–1572.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186.

Geli Fei and Bing Liu. 2016. Breaking the closed world assumption in text classification. In *Proceedings of NAACL-HLT*, pages 506–514.

Geli Fei, Shuai Wang, and Bing Liu. 2016. Learning cumulatively to become more knowledgeable. In *Proceedings of SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2016)*.

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*.

Lalit P Jain, Walter J Scheirer, and Terrance E Boult. 2014. Multi-class open set recognition using probability of inclusion. In *European Conference on Computer Vision*, pages 393–409. Springer.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 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.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020a. Pre-training via paraphrasing. *Advances in Neural Information Processing Systems*, 33.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020b. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Shiyu Liang, Yixuan Li, and R Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*.

Ting-En Lin and Hua Xu. 2019. A post-processing method for detecting unknown intent of dialogue system via pre-trained deep neural network classifier. *Knowledge-Based Systems*, 186:104979.

Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. 2013. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772.

Walter J Scheirer, Lalit P Jain, and Terrance E Boult. 2014. Probability models for open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2317–2324.

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471.

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.

Lei Shu, Hu Xu, and Bing Liu. 2018. Unseen class discovery in open-world classification. *arXiv preprint arXiv:1801.05609*.

Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. 2020. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In *34th Conference on Neural Information Processing Systems (NeurIPS) 2020*. Neural Information Processing Systems.

David MJ Tax and Robert PW Duin. 2004. Support vector data description. *Machine learning*, 54(1):45–66.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*.

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

Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. 2015. Short text clustering via convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 62–69.

Hanlei Zhang, Hua Xu, and Ting-En Lin. 2021. Deep open intent classification with adaptive decision boundary. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14374–14382.