Adversarial Semantic Decoupling for Recognizing Open-Vocabulary Slots

Yuanmeng Yan[1*], Keqing He[1*], Hong Xu[1], Sihong Liu[1], Fanyu Meng[2], Min Hu[2], Weiran Xu[1]
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]China Mobile Research Institute, Beijing, China
{yanyuanmeng,kqin,xuhong,liusihong,xuweiran}@bupt.edu.cn
{mengfanyu,humin}@chinamobile.com

Abstract

Open-vocabulary slots, such as file name, album name, or schedule title, significantly degrade the performance of neural-based slot filling models since these slots can take on values from a virtually unlimited set and have no semantic restriction nor a length limit. In this paper, we propose a robust adversarial model-agnostic slot filling method that explicitly decouples local semantics inherent in open-vocabulary slot words from the global context. We aim to depart entangled contextual semantics and focus more on the holistic context at the level of the whole sentence. Experiments on two public datasets show that our method consistently outperforms other methods with a statistically significant margin on all the open-vocabulary slots without deteriorating the performance of normal slots.

1 Introduction

Slot filling is a critical component of spoken language understanding (SLU) in task-oriented dialogue systems. It aims at extracting semantic constituents from the user queries. Given an immense amount of labeled training data, recent neural networks (Mesnil et al., 2015; Liu and Lane, 2015, 2016; Goo et al., 2018; Haihong et al., 2019; Chen et al., 2019; He et al., 2020a,b) have been actively applied to slot filling task and achieved good results.

Although most previous neural-based models achieve state-of-the-art performance across a wide range of slot filling datasets, they often suffer from poor slot filling accuracy while dealing with 'open-vocabulary' slots. Open-vocabulary slots signify slot types that can take on values from a virtually unlimited set, such as file name, album name, text body, or schedule title. Typically, these slot values

Figure 1: An error case of open-vocabulary slot "playlist" in Snips dataset (Coucke et al., 2018). Here "water" is mistakenly recognized as "entity_name" type by the baseline model (Liu and Lane, 2016) due to the local context "don't drink the water". However, it represents a playlist at the level of the whole sentence.

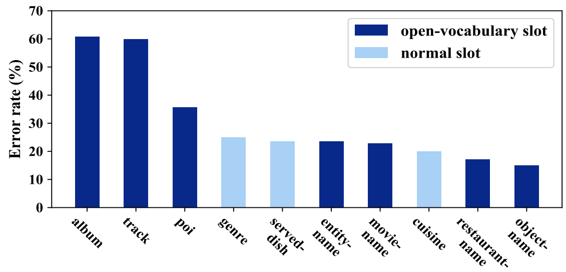

Figure 2: Error rates of open-vocabulary slots compared to normal slots in Snips from Baseline (Liu and Lane, 2016). We display the top10 slot types of the highest error rates.

have no constraints on the length and specific semantic patterns of content. Besides, these words are employed differently from the meaning inherent in themselves, as Fig 1 shows. Intrinsically, the complexity of recognizing open-vocabulary slots comes from the inconsistent context with different granularity. For example, consider the utterance "add the song don't drink the water to my playlist" in Fig 1. While identifying the slot type of the word "water", the slot filling model will mistakenly recognize the word "water" as "entity_name" slot type if it only focuses on the local context "don't drink the water". By contrast, it should instead focus on the global context "add the song ... to my playlist" to recognize the "don't drink the water" as the correct "playlist" slot type. Therefore, these characteristics of open-vocabulary slots confuse

*The first two authors contribute equally. Weiran Xu is the corresponding author.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6070–6075,
November 16–20, 2020.

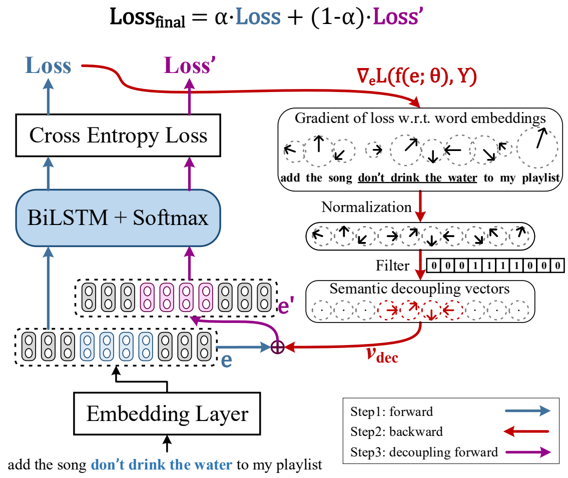

Figure 3: The overall architecture of our approach, including three core steps: forward, backward, and decoupling forward. Forward calculates the traditional classification loss and backward adds adversarial decoupling perturbations. Then decoupling forward calculates a new adversarial loss. Finally, the model is updated by the weighted sum of two losses.

the models to recognize the correct slot type. Fig 2 displays slot error rates of open-vocabulary slots are generally higher than normal slots. The results confirm that traditional neural networks can not adequately handle issues caused by open-vocabulary slots.

Kim et al. (2018) exploits a long-term aware attention structure and positional encoding with multi-task learning to capture global information. Kim et al. (2019) focuses on data augmentation by adding random noise in the embeddings of all slot words. Ray et al. (2019) proposes an iterative delexicalization algorithm that utilizes model uncertainty to improve delexicalization for open-vocabulary slots. One major limitation is that these methods can't explicitly distinguish semantic representation inherent in open-vocabulary slot words from the holistic context.

In this paper, we propose a robust adversarial slot filling approach that explicitly decouples local semantic representation inherent in open-vocabulary slot words from the global context. Our approach aims to focus more on the holistic semantics at the level of the whole sentence, not only the vicinity of the local context within open-vocabulary slots. Specifically, our approach generates model-agnostic adversarial worst-case perturbations to the inputs in the direction that significantly increases the model's loss. Our main contributions are threefold: (1) We dive into the issues of open-vocabulary slots in slot filling task and propose a novel adversarial semantic decoupling method which distinguishes local semantics from the global context. (2) Our method can be easily applied to all the previous slot filling neural-based models. (3) Experiments show that our proposed method consistently outperforms various SOTA baselines, especially in open-vocabulary slot f1.[1]

2 Approach

Problem Formulation Given a sentence $X = \{x_1, ..., x_n\}$ with n tokens, the slot filling task is to predict a corresponding tag sequence $Y = \{y_1, ..., y_n\}$ in BIO format, where each y_i can take three types of values: B-slot_type, I-slot_type and O.

Fig 3 shows the overall architecture of our method. Here we adopt BiLSTM (Liu and Lane, 2016) as our backbone.[2] Our method includes three core steps: forward, backward, and decoupling forward. We first feed each word to an embedding layer to get word embeddings $e_i = \mathrm{E}(x_i)$. Then in the forward step, we adopt a BiLSTM layer and softmax output layer to calculate the classification cross-entropy loss $\mathcal{L}(f(\boldsymbol{e};\theta), Y)$ for each word.

Then in the second backward step, we perform adversarial attacks (Goodfellow et al., 2015; Kurakin et al., 2016; Miyato et al., 2016; Jia and Liang, 2017; Zhang et al., 2019; Ren et al., 2019) to explicitly shift the local semantics of open-vocabulary slot words and decouple them from the global context. Theoretically, we need to compute a decoupling vector $\widetilde{\boldsymbol{v}}_{\text{dec}}$ that effectively degrades the current model's performance (i.e., maximum the loss function):

$$\widetilde{\boldsymbol{v}}_{\text{dec}} = \underset{||\boldsymbol{v}_{\text{dec}}|| \le \epsilon}{\arg\max}\, \mathcal{L}(f(\boldsymbol{e} + \boldsymbol{v}_{\text{dec}}; \theta), Y) \quad (1)$$

where $\mathcal{L}$ indicates the loss function and ϵ is the norm bound of the decoupling vector. However, due to model complexity, accurate computation for $\widetilde{\boldsymbol{v}}_{\text{dec}}$ is costly and inefficient. Similar to Vedula et al. (2020) and Ru et al. (2020), we apply Fast Gradient Value (FGV) (Rozsa et al., 2016) to approximate a worst-case perturbation as our decoupling vector:

$$\widetilde{\boldsymbol{v}}_{\text{dec}} = \epsilon \frac{g}{||g||}; \text{where } g = \nabla_{\boldsymbol{e}} \mathcal{L}(f(\boldsymbol{e};\theta), Y) \quad (2)$$

[1] Our code is available at `https://github.com/yym6472/OVSlotTagging`

[2] Since our method is model-agnostic, we also apply our method to BERT (Devlin et al., 2019) in the experiments.

Here, the gradient g is the first-order differential of the loss function $\mathcal{L}$ w.r.t. $\boldsymbol{e}$, representing the direction that rapidly increases the loss function. We perform normalization to g and then use a small ϵ to ensure the approximate is reasonable. Finally, we perform a mask operation to filter out normal words and add the decoupling vector to the original token embeddings $\boldsymbol{e}$. Hence, the updated word embeddings are $\boldsymbol{e}' = \boldsymbol{e}+\widetilde{\boldsymbol{v}}_{\text{dec}}$ while other model parameters are fixed. Ablation study proves that only adding the decoupling vector to open-vocabulary slot words achieves better improvement.

In the third decoupling forward step, we feed $\boldsymbol{e}'$ to the same BiLSTM model and calculate a new adversarial loss $\mathcal{L}'$. The final loss is a weighted sum of $\mathcal{L}$ and $\mathcal{L}'$ controlled by a hyperparameter α[3]:

$$\mathcal{L}_{\text{final}} = \alpha \cdot \mathcal{L} + (1 - \alpha) \cdot \mathcal{L}' \tag{3}$$

Finally, we use $\mathcal{L}_{\text{final}}$ to update all the model parameters.

By adding those decoupling vectors to open-vocabulary slot words, we break the semantics inherent in open-vocabulary slots and thus force the model to pay more attention to global context (e.g. "add the song ... to my playlist") when identifies types of open-vocabulary slots.

3 Experiment

3.1 Setup

Datasets To evaluate our approach, we conduct experiments on two public benchmark datasets, Snips (Coucke et al., 2018) and MIT-restaurant (MR)[4]. Snips contains user utterances from various domains resulting in relatively extensive open-vocabulary slots, such as album and movie_name. MR is a single-domain dataset associated with restaurant reservations, which contains open-vocabulary slots, such as restaurant_name and amenity.[5] Table 1 shows the full statistics and Table 2 shows all the open-vocabulary slots of Snips and MR datasets. Note that we identify the open-vocabulary slots according to the diversity of different slot values as well as the average length of slot values.

[3] In the experiments, we set α to 0.5.

[4] https://groups.csail.mit.edu/sls/downloads/restaurant/

[5] Similar to (Ray et al., 2019), we do not consider the ATIS (Hemphill et al., 1990) dataset since it lacks open-vocabulary slots, hence not suited for our evaluation. And we only focus on the main slot filling task instead of intent detection.

	Snips	MR
Vocabulary size	11,241	3,804
Percentage of OOV words	5.95%	2.76%
Number of all slots	39	8
Number of open-vocabulary slots	9	4
Train set size	13,084	6,894
Development set size	700	766
Test set size	700	1,521

Table 1: Statistics of Snips and MR datasets.

Dataset	Open-vocabulary Slots	Normal Slots
Snips	playlist, object_name, entity_name, album, movie_name, track, poi, geographic_poi, restaurant_name	served_dish, cuisine, sort, best_rating, genre, service, movie_type, ...
MR	restaurant_name, dish, amenity, location	rating, hours, cuisine, price

Table 2: The lists of all the open-vocabulary slots and normal slots in Snips and MR datasets. We only show a part of normal slots in Snips dataset for clarity.

Baselines For a fair comparison, we use the same slot filling architecture BiLSTM (Liu and Lane, 2016) as (Kim et al., 2019; Ray et al., 2019). Kim et al. (2019) proposes two model variants, where *random noise* means adding random noise in the embeddings of all slot words and *cw* represents concatenating the context word window as input. Note that the random noise in (Kim et al., 2019) is independently sampled regardless of the global context, which is significantly different from our method. Our adversarial semantic decoupling method can take into account the impact of different contexts (global semantics) on local semantics, thereby enabling more accurate decoupling. Ray et al. (2019) proposes *greedy delex* and *iterative delex* methods for open-vocabulary slots. We also validate our method in the BERT-based models (Devlin et al., 2019) for comprehensive analysis.

Evaluation We evaluate the performance of slot filling using the F1 metric (Sang and Buchholz, 2000). Specially, we report the F1 score over all open-vocabulary slots, noted as F1-ov. We followed the set-ups in (Liu and Lane, 2016; Kim et al., 2019), and re-implement the baseline *BiLSTM*, *+random noise* and *+random noise,cw* based on the same settings. We report the original results of *greedy delex* and *iterative delex* from (Ray et al., 2019).

3.2 Main Results

We display the experiment results in Table 3. Compared to the previous state-of-the-arts , our method

Model	Snips				MR			
	Valid		Test		Valid		Test	
	F1	F1-ov	F1	F1-ov	F1	F1-ov	F1	F1-ov
BiLSTM (Liu and Lane, 2016)	91.63	78.91	88.99	71.78	73.67	71.44	72.07	70.39
+CRF	93.37	83.55	92.28	79.71	76.51	75.63	75.78	75.45
+random noise (Kim et al., 2019)	92.94	81.92	92.46	82.35	76.43	75.61	75.81	75.51
+random noise,cw (Kim et al., 2019)	93.52	82.06	92.89	82.58	76.51	75.78	75.92	75.60
+greedy delex (Ray et al., 2019)	-	-	92.56	-	-	-	-	-
+iterative delex (Ray et al., 2019)	-	-	93.24	-	-	-	-	-
ours	94.33	85.57	**94.55***	**86.09***	78.94	77.89	**77.96***	**77.48***
BERT (Devlin et al., 2019)	94.61	84.09	93.31	79.77	76.80	75.35	76.07	75.40
+CRF	95.93	88.05	94.70	84.99	79.66	79.43	79.39	79.55
+random noise	95.99	88.05	95.63	87.32	79.67	79.39	79.59	79.68
+random noise,cw	95.90	87.92	95.57	87.18	79.59	78.84	79.49	79.56
ours	95.88	88.24	**95.87**	**88.06***	81.54	80.97	**81.61***	**81.78***

Table 3: Slot filling performance on Snips and MR datasets. F1 is the overall score on all slot types and F1-ov is the score on all the open-vocabulary slots. The numbers with * indicate the significant improvement over all baselines with $p < 0.05$ under t-test.

achieves significantly superior performance for both datasets, both in F1-ov and overall slot F1. In the Snips dataset, our BiLSTM-based method outperforms the SOTA model by 3.51% in F1-ov and 1.31% in F1. In the MR dataset, our method gets improvements of 1.88% in F1-ov and 2.04% in F1. The results demonstrate that explicitly decoupling local semantics inherent in open-vocabulary slot words from the global context can effectively benefit open-vocabulary slot filling. We observe that in the Snips dataset F1-ov is extremely lower than F1, which shows the previous slot filling methods cannot tackle the critical issues of open-vocabulary slots. There is no such clear performance drop in the MR dataset. The probable reason is that open-vocabulary slots account for a large proportion(70%) of all samples on MR.

We also show the results of BERT models. Table 3 displays that our method still achieves an improvement of 8.29% in F1-ov over the original BERT model and 0.74% over the previous SOTA, which substantiates our method is model-agnostic and can be easily integrated into different slot filling architectures. Meanwhile, the F1-ov scores in BERT-based models are consistently higher than BiLSTM-based models, which indicates that BERT can effectively capture the global context semantics and tackle long-term dependency than BiLSTM.

3.3 Qualitative Analysis

Results of all open slot categories Fig 4 shows test F1 scores of five open-vocabulary slot types to verify the improvement of each type. We choose BiLSTM and random noise as our baseline models. The results demonstrate that our method consistently outperforms other methods on each open-vocabulary slot type, which confirms our method is not specific to several slot types. For the *restaurant_name* type, the random noise model suffers from a performance drop of 7.62% compared to BiLSTM. It illustrates simply adding random noise is not constrained and has no guarantee of semantics decoupling. Conversely, our method employs adversarial deliberate disturbance and outperforms BiLSTM by 9.58%.

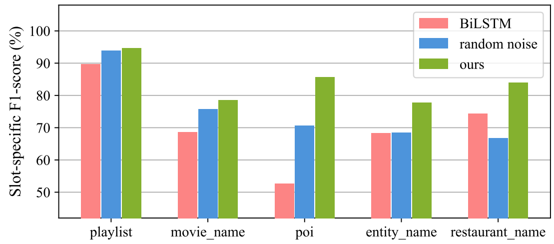

Figure 4: Test F1 scores of each open-vocabulary slot type on Snips. We show the results of five slots for clarity.

Model	F1	F1-ov	F1-normal
BiLSTM	88.99	71.78	94.50
random noise	92.46	82.35	95.51
ours	**94.55**	**86.09**	**97.10**

Table 4: Performance comparison between open-vocabulary slots and normal slots on Snips.

Open-vocabulary slots vs normal slots We also show overall test F1, F1-ov on all the open-vocabulary slots, and F1-normal on all the normal slots in Table 4 to compare the comprehensive performance. The results show that our method significantly outperforms BiLSTM by 14.31% on F1-ov and 2.6% on F1-normal, which proves our method gets notable improvement on open-vocabulary slots without harm to the performance of normal slots. We hypothesize the improvement on normal slots

Filter	Space	ϵ	α	F1-ov
OV slots	Embedding	1.5	0.5	**86.09**
All slots	Embedding	1.5	0.5	84.44
OV slots	BiLSTM	1.5	0.5	82.86
OV slots	Embedding	1.0	0.5	84.44
OV slots	Embedding	3.0	0.5	82.05
OV slots	Embedding	1.5	0.4	85.20
OV slots	Embedding	1.5	0.6	85.34

Table 5: Effects of different hyperparameters on Snips dataset for the BiLSTM-based model. Filter indicates whether the perturbation is applied to the open-vocabulary slots or all slots. Space indicates which space the perturbation is added to, where *Embedding* means the space after the word embedding layer and *BiLSTM* means the space after the BiLSTM layer. ϵ indicates the norm of perturbation and α is a hyperparameter to balance two training objectives.

is mainly because our method can effectively alleviate contextual semantic noise caused by open-vocabulary slots.

Analysis of generalization capability Table 3 shows there exists clear overfitting for BiLSTM and BERT models on open-vocabulary slots. For example, BiLSTM gets a performance drop of 7.13% comparing test F1-ov with valid F1-ov, and BERT gets a drop of 4.32%. The overfitting illustrates these baselines cannot capture contextual patterns, resulting in poor generalization capability to new slot values. By contrast, our method achieves comparable performance on valid and test sets both for BiLSTM(85.57 vs 86.09) and BERT(88.24 vs 88.06) architectures. The results demonstrate our method has a strong generalization capability for open-vocabulary slots.

Ablation studies To study the effects of different hyperparameters of our method, we conduct ablation analysis under BiLSTM architecture (Table 5). We can see that adding perturbation to the embedding layer of open-vocabulary slots gets significant improvement. Specifically, for the Filter setting, adding perturbation to open-vocabulary slots outperforms all slots by 1.65%. For the Space setting, adding perturbation to the word embedding layer is superior to the RNN layer. For the hyperparameters ϵ and α, $\epsilon = 1.5$ and $\alpha = 0.5$ achieves the best performance.

Case study Table 6 gives three examples from the Snips dataset: (1) the baseline model identifies a partial word "one" in "the sound of one hand clipping" as "rating_value" due to overfitting. (2) the baseline model fails to identify "look to you" since it is heavily coupled with "put" in local semantics.

Example 1	search for *the sound of one hand clipping*
Baseline Pred.	O O B-obj_nm I-obj_nm O B-rating_value O B-obj_nm
Proposed Pred.	O O B-obj_nm I-obj_nm I-obj_nm I-obj_nm I-obj_nm I-obj_nm
Example 2	i want to put *look to you* on the playlist named *80s classic hits*
Baseline Pred.	O O O O O O O O O O O B-plist I-plist I-plist
Proposed Pred.	O O O O B-ent_nm I-ent_nm I-ent_nm O O O O B-plist I-plist I-plist
Example 3	*a day no pigs would die* deserves a best rating of 6 and a value of 4
Baseline Pred.	B-obj_nm I-obj_nm I-obj_nm I-obj_nm O O O O O O O B-best_rt O O O O B-rt_value
Proposed Pred.	B-obj_nm I-obj_nm I-obj_nm I-obj_nm I-obj_nm I-obj_nm O O O O O B-best_rt O O O O B-rt_value
Abbreviation	'object': 'obj', 'name': 'nm', 'entity': 'ent', 'playlist': 'plist', 'rating': 'rt'

Table 6: Three examples from the Snips dataset. The *italic spans* are open-vocabulary slots and should be viewed as a whole. We use RED and GREEN text to represent wrong and correct slot filling results, respectively. For brevity, we abbreviate some slot type words.

(3) the predicate "would die" in open-vocabulary slots are identified as the predicate of the whole sentence and thus are mistakenly labeled as "O" by the baseline model. In all cases, the baseline model focuses too much on local semantics and neglects the hints in global. With our proposed approach, the model is trained to pay more attention to global semantics and succeeds to identify open-vocabulary slots.

4 Conclusion

In this paper, we dive into the issues of open-vocabulary slots in slot filling task and propose a novel model-agnostic adversarial semantic decoupling method which distinguishes local semantics inherent in open-vocabulary slot words from the global context. Experiments confirm the effectiveness of semantic decoupling. We hope to provide new guidance for the future slot filling work.

Acknowledgments

We thank Pengda Qin and all anonymous reviewers for their constructive feedback. This work was partially supported by MoE-CMCC "Artifical Intelligence" Project No. MCM20190701, National Key R&D Program of China No. 2019YFF0303300 and Subject II No. 2019YFF0303302, DOCOMO Beijing Communications Laboratories Co., Ltd.

References

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv* *preprint arXiv:1902.10909*.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

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

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572.

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

Keqing He, Weiran Xu, and Yuanmeng Yan. 2020a. Multi-level cross-lingual transfer learning with language shared and specific knowledge for spoken language understanding. *IEEE Access*, 8:29407–29416.

Keqing He, Yuanmeng Yan, and XU Weiran. 2020b. Learning to tag oov tokens by integrating contextual representation and background knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 619–624.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Hwa-Yeon Kim, Yoon-Hyung Roh, and Young-Kil Kim. 2019. Data augmentation by data noising for open-vocabulary slots in spoken language understanding. In *NAACL-HLT*.

Junseong Kim, Junghoe Kim, SeungUn Park, Kwangyong Lee, and Yoonju Lee. 2018. Modeling with recurrent neural networks for open vocabulary slots. In *COLING*.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Bing Liu and Ian Lane. 2015. Recurrent neural network structured output prediction for spoken language understanding. In *Proc. NIPS Workshop on Machine Learning for Spoken Language Understanding and Interactions*.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Z. Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, and Geoffrey Zweig. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23:530–539.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

Avik Ray, Yilin Shen, and Hongxia Jin. 2019. Iterative delexicalization for improved spoken language understanding. In *INTERSPEECH*.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097.

Andras Rozsa, Ethan M Rudd, and Terrance E Boult. 2016. Adversarial diversity and hard positive generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 25–32.

Dongyu Ru, Yating Luo, Lin Qiu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2020. Active sentence learning by adversarial uncertainty sampling in discrete space. *arXiv preprint arXiv:2004.08046*.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task chunking. *ArXiv*, cs.CL/0009008.

Nikhita Vedula, Nedim Lipka, Pranav Maneriker, and Srinivasan Parthasarathy. 2020. Open intent extraction from natural language interactions. In *Proceedings of The Web Conference 2020*, pages 2009–2020.

Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2019. Generating fluent adversarial examples for natural languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5569.