INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Data Augmentation for Spoken Language Understanding via Pretrained Language Models

Baolin Peng, Chenguang Zhu*, Michael Zeng, Jianfeng Gao*

Microsoft Research, Redmond
{bapeng,chezhu,nzeng,jfgao}@microsoft.com

Abstract

The training of spoken language understanding (SLU) models often faces the problem of data scarcity. In this paper, we put forward a data augmentation method using pretrained language models to boost the variability and accuracy of generated utterances. Furthermore, we investigate and propose solutions to two previously overlooked semi-supervised learning scenarios of data scarcity in SLU: i) *Rich-in-Ontology*: ontology information with numerous valid dialogue acts is given; ii) *Rich-in-Utterance*: a large number of unlabelled utterances are available. Empirical results show that our method can produce synthetic training data that boosts the performance of language understanding models in various scenarios.

Index Terms: Spoken language understanding, pretraining, data augmentation, rich-in-ontology, rich-in-utterance

1. Introduction

Spoken Language Understanding (SLU) is widely applied in human-machine dialogue systems to convert natural utterances into predefined semantic frames, i.e. dialogue acts, for further processing. For example, an SLU component in a virtual assistant or robot outputs its prediction of intents and slot labels detected within a user's utterance [1]. Nevertheless, as a supervised learning task, SLU suffers from the problem of data scarcity. The problem becomes more prevalent in face of new LU domains with novel definitions of intents and slot labels. Even with an existing domain, the data correlated with a certain intent or slot is often not sufficient. These problems significantly limit the applicability of SLU systems.

Recently, various successful use cases of synthetic datasets have stimulated the growth of the area of Data Augmentation (DA) [2, 3]. The typical approach is to learn a model to mimic the language style in the training data, leveraging the relationship between semantic units and their natural representations. Then, a non-generative model can modify utterances and replace slot labels from existing data [4], while a generative model can produce synthetic utterances in the same distribution space of the training data [5]. However, these approaches usually train the DA model on domain-specific data, which is of a small scale by itself. It is thus questionable whether the augmented data contains rich language expressibility beyond the scope of the given data.

On the other hand, the rapid development of large-scale pretrained language models has significantly improved the capacity of language understanding and generation models [6, 7]. With a modest amount of domain-specific data, a pretrained model can quickly adapt to a new domain. For instance, SC-GPT [8] finetunes the GPT-2 language model [9] with dialogue data. It can efficiently adapt to new dialogue domains with only a couple of labelled data samples.

In this paper, we propose to frame data augmentation as a semantically controlled generation problem. Given dialogue act, we leverage the pretrained SC-GPT model to generate corresponding utterances as synthetic training data. In the process, the general language syntax and semantics learned during the pretraining phase are fused into the generation of domain-specific utterances to increase variability and accuracy of SLU.

Furthermore, previous literature on SLU data augmentation focus on the case where only a scant number of pairs of utterance and corresponding semantic labels are given, which we denote as *Paired-Data-Only*. However, there are two other overlooked semi-supervised learning scenarios that commonly arise in application.

- *Rich-in-Ontology*: The full ontology for the dialogue domain is also given, including the definitions of intents, slot lists and valid combinations of slots and values. In other words, the model is given a variety of valid combinations of semantic labels. What lacks is the corresponding natural language utterances.
- *Rich-in-Utterance*: Apart from the labelled data, there are abundant unlabelled utterances without annotated intents, slots and values.

In this paper, we also delve into these two scenarios and propose corresponding data augmentation solutions.

For *Rich-in-Ontology*, we first finetune the pretrained model SC-GPT on the paired training data, and then apply it to the valid combination of intents and slots in the ontology information to generate additional training data.

For *Rich-in-Utterance*, following the idea of the NLG model SC-GPT, we propose SC-GPT-NLU, which is pretrained on the same corpus of SC-GPT with flipped sources and targets. In detail, we feed the utterances into the model and let it generate intent and slots in a text sequence. Therefore, SC-GPT-NLU can act as a language understanding module and produce semantic labels for the unlabelled utterances available.

In the experiments, we evaluate the slot tagging and intent classification accuracies of a Bi-LSTM seq2seq SLU model, using various data augmentation methods. Results show that on ATIS and Snips datasets, our proposed method outperforms other baseline systems. For instance, compared with baseline methods, the data augmented by our system can help the underlying SLU model achieve 0.5 points higher slot F1 and 3.02 points higher intent accuracy in ATIS-Small dataset. Furthermore, when ontology information or unlabelled utterances are available, i.e. Rich-in-Ontology and Rich-in-Utterance, our method can produce synthetic data that significantly boosts the performance of SLU models.

*Equal contribution

http://dx.doi.org/10.21437/Interspeech.2021-117

2. Related Work

2.1. SLU Data Augmentation

Many previous approaches to SLU data augmentation target to increase variability of generated utterances. [10] proposes to add noise to perturb the decoder states to generate variants of an utterance. Variational autoencoder (VAE) and conditional variational autoencoder (CVAE) are used to generate utterances with diversified expressions [11]. [4] uses both non-generative models like word substitution and generative models like paraphrasing and back-translation to augment training data. [5] proposes a multi-stage framework to generate, filter, and rank augmented utterances. [12] uses reinforcement learning to learn a generator that facilitates dialogue state tracking. [13] employs atomic templates to guide the model to generate more utterances given combination of dialogue acts. [14] proposes to select sentences from unlabeled utterances and apply pseudo-labels. The two additional scenarios we propose in this paper are also related to semi-supervised learning [15]. But we focus on data augmentation, which is independent of the downstream learning models.

Similar to our work, [16, 17] uses pretrained language models to generate synthetic training data for data augmentation. However, their approach blends multiple labels and input sentences together during training, so it is hard to control the amount of generated synthetic data per class.

2.2. Pretraining

Pretrained models leverage the large amount of unlabelled text corpora to improve the capability of language understanding. ELMo [18] applies two unidirectional RNNs for language modeling. GPT-2 [9] utilizes the transformer architecture [19] for the task. BERT [6] employs a masking technique and next-sentence-prediction task to train a bidirectional language model. UniLM [20] uses different masking patterns to unify the model structure for NLU and NLG. These pretrained language models have been widely used with considerable success in various NLP applications such as question answering [21] and summarization [22].

Furthermore, pretrained language models have been leveraged in speech language processing to provide rich contextual embeddings [23]. Specifically, SC-GPT [8], i.e. Semantically Conditioned Generative Pre-training Transformer, builds upon GPT-2 and is further pretrained on a large-scale dialogue corpus. The resulting model outperforms many baselines in few-shot language generation for task-oriented dialogue.

3. Data Augmentation

3.1. Traditional Augmentation Scenario

We describe the traditional augmentation scenario in SLU as **Paired-Data-Only**, as the training data consists of N instance pairs. Each pair contains the input tokenized utterance $\boldsymbol{x} = (x_1, x_2, ..., x_T)$ and the corresponding output *dialogue act* $\mathcal{A}$. $\mathcal{A}$ includes the intent label $\mathbf{I}$ and P slot-value pairs:

$$\mathcal{A} = [\underbrace{\mathbf{I}}_{\text{Intent}}, \underbrace{(s_1 = v_1, \cdots, s_P = v_P)}_{\text{Slot-value pairs}}] \quad (1)$$

Thus, the training data $\mathcal{D} = \{(\boldsymbol{x}_1, \mathcal{A}_1), ..., (\boldsymbol{x}_N, \mathcal{A}_N)\}$. However, due to high labeling costs, the size of labeled data N is usually small. In such cases, data augmentation (DA) is needed. An augmenter $\mathbf{S}$ is a language generation model, which is trained on $\mathcal{D}$ to be able to produce a corresponding utterance $\tilde{x}$ given an input dialogue act $\tilde{\mathcal{A}}$. For example, suppose $\tilde{\mathcal{A}}$ = [*hotel-inform*,$(name = Hyatt, area = center, star = 5$], $\mathbf{S}$ can generate $\tilde{x}$ =*I have booked the 5-star Hyatt hotel in the center area for you.*

Table 1: *The input and output of SC-GPT [8] and SC-GPT-NLU models. Both are initialized with GPT-2 [9] but further pretrained on different data with swapped inputs and outputs.*

Model	Input	Output
SC-GPT	Dialogue act	Utterance
SC-GPT-NLU	Utterance	Dialogue act

Then, during augmentation, we first augment the dialogue acts in the training data by replacing/inserting/deleting slot values to create more combinations. The augmenter $\mathbf{S}$ then generates candidate utterances for the dialogue acts. As the generated utterances may not always contain the required slot-value labels, we filter them to make sure that each utterance has all the required input slot-values.

However, the data augmenter itself requires a considerable amount of training data. As a result, augmenters directly trained on $\mathcal{D}$ may have limited model capacity and expressibility. Thus, we adopt the pretrained model SC-GPT [8], which is a language model to produce utterances given a dialogue act. SC-GPT is initialized with GPT-2 [9], further pretrained on a large corpus of 400K dialogue-act-utterance pairs and then fine-tuned on the training data $\mathcal{D}$. It has been shown that SC-GPT can quickly adapt to new domains with only a few domain-specific data samples [8].

3.2. More Data Augmentation Scenarios

We note that in many real applications, there is often additional available information beyond the paired training data. Here, we specify two semi-supervised scenarios that commonly arise in applications but have been overlooked by previous approaches.

3.2.1. Rich In Ontology

In many dialogue domains, a detailed description of the ontology is given, which is a list of valid dialogue acts. Formally, the training data consists of both labelled pairs and many dialogue acts: $\mathcal{D} = \{(\boldsymbol{x}_1, \mathcal{A}_1), ..., (\boldsymbol{x}_N, \mathcal{A}_N), \mathcal{A}_{N+1}, ..., \mathcal{A}_M\}$.

To work with this scenario, we finetune SC-GPT on the paired part of $\mathcal{D}$, i.e. $\{(\boldsymbol{x}_1, \mathcal{A}_1), ..., (\boldsymbol{x}_N, \mathcal{A}_N)\}$, and then generate utterances for the other dialogue acts $\{\mathcal{A}_{N+1}, ..., \mathcal{A}_M\}$. The utterances are then filtered to make sure that each utterance has all the corresponding slot-values.

3.2.2. Rich In Utterance

It is common in practice that a large number of unlabelled dialogue utterances are available, usually collected from history data. Formally, the training data consists of both labelled pairs and many unlabeled utterances: $\mathcal{D} = \{(\boldsymbol{x}_1, \mathcal{A}_1), ..., (\boldsymbol{x}_N, \mathcal{A}_N), \boldsymbol{x}_{N+1}, ..., \boldsymbol{x}_M\}$.

To utilize these utterances, we need to produce corresponding dialogue acts. We propose to finetune GPT-2 in the reverse way: feed an utterance as input and let the model generate the dialogue act as output. In other words, we leverage a language generation model to act as a language understanding module, denoted as SC-GPT-NLU (Table 1).

Like SC-GPT, SC-GPT-NLU is initialized with GPT-2 and

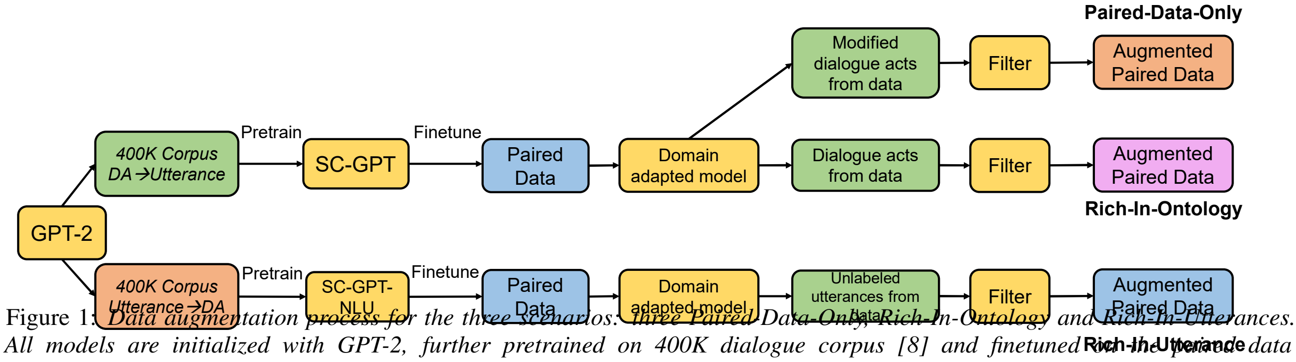

Figure 1: *Data augmentation process for the three scenarios: three Paired-Data-Only, Rich-In-Ontology and Rich-In-Utterances. All models are initialized with GPT-2, further pretrained on 400K dialogue corpus [8] and finetuned on the paired data* $\{(\boldsymbol{x}_1, \mathcal{A}_1), ..., (\boldsymbol{x}_N, \mathcal{A}_N)\}$.

Table 2: *Slot F1 and intent accuracy scores on ATIS and Snips dataset. The overall highest score is in bold, and the best result in Paired-Data-Only category is underlined. *: Statistically significant with p-value less than 0.05.*

Dataset	ATIS				Snips			
Split	Small		Medium		Small		Medium	
Model	Slot F1	Intent Acc.	Slot	Intent	Slot	Intent	Slot	Intent
				No Data Augmentation				
No-DA	68.91	84.99	87.30	90.15	61.30	93.43	79.83	97.29
				Paired-Data-Only				
Seq2Seq	73.71	-	88.72	-	-	-	-	-
VAE	74.92	83.65	<u>89.27</u>	<u>90.95</u>	-	-	-	-
Ours	<u>75.42</u>	<u>86.67</u>	88.61	90.71	64.96	93.43	80.62	97.57
				Rich-in-Ontology				
Ours	**82.42***	**89.03***	**89.81***	**92.27***	**67.06***	**94.14***	**82.54***	97.86
				Rich-in-Utterance				
Ours	78.45	87.46	88.23	91.94	63.46	93.43	80.54	**98.14***

further pretrained on the 400K dialogue-act-utterance data and finetuned on the paired part of $\mathcal{D}$. But SC-GPT-NLU treats the utterance as input and dialogue acts as output. So both SC-GPT and SC-GPT-NLU are language generation models with a softmax-based output layer that produces utterance/dialogue acts token by token.

During augmentation, SC-GPT-NLU generates dialogue acts for the unlabeled utterances $\boldsymbol{x}_{N+1}, ..., \boldsymbol{x}_M$. Here, the generated names of intents, slots and values are mapped to the pre-defined ontology by string matching. The augmented data is filtered to make sure that each input slot-value appears in the utterance.

Figure 1 illustrates our SLU data augmentation process for all three scenarios.

4. Experiments

4.1. Datasets and Metrics

We employ the widely used SLU benchmark dataset ATIS [24] and Snips [25]. ATIS contains around 5.8K utterances from flight reservation dialogues. It includes 120 slot labels and 21 intent types. Snips contains 14K utterances from the Snips personal voice assistant. It includes 123 slot labels and 7 intent types.

To simulate the few-shot data situations, we follow [26] to use two small portions of the ATIS training set as training data: Small (∼1/40 of the original training set) and Medium (∼1/10 of the original training set). A development set of 500 instances is used. Following the same split ratio, we sampled 327 and 1308 instances in Snips for Small and Medium respectively.

We use F1 score to measure slot tagging quality and use accuracy score to evaluate intent classification, in accordance with [26].

4.2. Model Details

SLU Model. For fair comparison, we use the same SLU model that is trained on the training data and the data augmented by our model and baseline systems.

We adopt the same setting for the SLU model as in [5]. It has two layers of bi-directional LSTM with a hidden dimension of 200 and a dropout probability of 0.5. We choose the Adam optimizer [27] with a learning rate of 0.001. Gradients with a 2-norm greater than 5 are clipped. The best model is selected based on performances on the validation set. The number of training epochs is 50 and the batch size is 20.

Data augmentation. For the Paired-Data-Only case, we modify the dialogue acts in the training split to construct around 300 additional combinations of DAs via dropping/inserting/replacing

Table 3: *Example utterances generated by SC-GPT given dialogue acts (DA) and dialogue acts generated by SC-GPT-NLU given unlabelled utterances in Snips.*

SC-GPT	
DA	RateBook (best_rating = 6; object_select = current; object_type = textbook; rating_value = 3)
Utterance 1	Give 3 out of 6 to current textbook
Utterance 2	The current textbook gets a 3 out of 6
Utterance 3	I think that the current textbook should be rated 3 out of 6
DA	BookRestaurant (country = Honduras; facility = indoor; restaurant_type = restaurant)
Utterance 1	Book me a reservation for an indoor restaurant in Honduras
Utterance 2	Book an indoor restaurant in Honduras
Utterance 3	I need to book an indoor restaurant in Honduras
SC-GPT-NLU	
Utterance	2 of us want to eat at a restaurant that serves meatballs in VT
DA	BookRestaurant (party_size_number = 2; restaurant_type = restaurant; served_dish = meatballs; state = VT)
Utterance	Add the track to the Metal Talks Metallica playlist.
DA	AddToPlaylist (music_item = track; playlist = metal talks Metallica)

slots and values. For each dialogue act, we sample three utterances produced by SC-GPT. After filtering out utterances which do not contain all the slot-values, we collect around 500 synthetic utterances and add them into the original training split.

We simulate the Rich-in-Ontology scenario by making the dialogue acts in the whole training set available, from which 500 dialogue acts are sampled and added to the training split.

For the Rich-in-Utterance scenario, we sample 1,000 utterances in the training corpus and use SC-GPT-NLU to produce the most probable dialogue act. After filtering, around 500 utterance-DA pairs are added to the original training split.

Implementation details. Both SC-GPT and SC-GPT-NLU are finetuned for 5 epoches with a learning rate as 5e-5. Nucleus sampling [28] is used for decoding, where the sampling top-p is 0.9, and the temperature is 1. Details on SC-GPT including the number of parameters and pretraining procedure can be found at [8]. The finetuning takes about half an hour on a V100 GPU machine 64GB memory.

Baselines. The baseline data augmentation systems include the seq2seq [5] and variational autoencoder (VAE) data augmentation model [29]. We also report the results for the case without data augmentation, denoted by No-DA.

4.3. Results

Table 2 shows the accuracy of slot tagging and intent classification for various models. Based on the results, we make the following observations.

Firstly, our data augmentation method can considerably boost the model accuracy (comparing No-DA and Ours), especially when the training data size is small. For instance, in ATIS, when only paired data is available, the slot F1 increases by 6.51 (Small) and 1.31 (Medium) points, while the intent accuracy increases by 1.68 (Small) and 0.56 (Medium) points.

Secondly, under Rich-in-Ontology and Rich-in-Utterance scenarios, our method further boosts the slot F1 by up to 7 points and intent accuracy by up to 2.4 points. Overall, the accuracy scores are the highest when the ontology information is available. This shows that our method can take advantage of additional information and produce better synthetic training data for downstream models. We conduct statistical paired t-tests and find that the best model's performance is all statistically significant with p-value less than 0.05.

Thirdly, under the traditional Paired-Data-Only scenario, our data augmentation method outperforms all baselines in ATIS-Small, and achieves comparable results in ATIS-Medium. This shows that our method is better suited when training data is scarce.

4.4. Examples of Augmented Data

In Table 3, we show examples of generated utterances and dialogue acts by SC-GPT and SC-GPT-NLU in Snips. As shown, after pretraining and domain finetuning, SC-GPT can produce coherent utterances with a high variability, while covering all required intent, slots and values. SC-GPT-NLU can generate dialogue acts in the same format as input to SC-GPT, which captures the important semantic information in the input utterances. This demonstrates that pretrained models can quickly adapt to target domains with a small amount of labeled data. This facilitates the generation of high-quality synthetic data for SLU.

5. Conclusion

In this paper, we approach the problem of data scarcity in SLU with pretrained language models. After finetuning on domain-specific dialogue data, our model can produce high-quality synthetic data which boosts the performance of the downstream SLU model. Moreover, we provide solutions to two semi-supervised scenarios in SLU overlooked by previous literature: *Rich-in-Ontology* and *Rich-in-Utterance*. In experiments on the benchmark datasets ATIS and Snips, we demonstrate that our solution can effectively leverage auxiliary unlabeled data to produce high-quality synthetic training data for building SLU models with a higher accuracy.

As future work, we aim to extend the idea of data augmentation based on pretrain language models to other speech language processing tasks, such as information retrieval and summarization.

6. References

[1] J. R. Bellegarda, "Spoken language understanding for natural interaction: The siri experience," in *Natural interaction with robots, knowbots and smartphones*. Springer, 2014, pp. 3–14.

[2] X. Lu, B. Zheng, A. Velivelli, and C. Zhai, "Enhancing text categorization with semantic-enriched representation and training data augmentation," *Journal of the American Medical Informatics Association*, vol. 13, no. 5, pp. 526–535, 2006.

[3] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[4] J. Quan and D. Xiong, "Effective data augmentation approaches to end-to-end task-oriented dialogue," *arXiv preprint arXiv:1912.02478*, 2019.

[5] Y. Hou, Y. Liu, W. Che, and T. Liu, "Sequence-to-sequence data augmentation for dialogue language understanding," *arXiv preprint arXiv:1807.01554*, 2018.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[7] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[8] B. Peng, C. Zhu, C. Li, X. Li, J. Li, M. Zeng, and J. Gao, "Few-shot natural language generation for task-oriented dialog," *arXiv preprint arXiv:2002.12328*, 2020.

[9] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[10] G. Kurata, B. Xiang, and B. Zhou, "Labeled data generation with encoder-decoder lstm for semantic slot filling." in *Interspeech*, 2016.

[11] J. Li, L. Qiu, B. Tang, D. Chen, D. Zhao, and R. Yan, "Insufficient data can also rock! Learning to converse using smaller data with augmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6698–6705.

[12] Y. Yin, L. Shang, X. Jiang, X. Chen, and Q. Liu, "Dialog state tracking with reinforced data augmentation," *arXiv preprint arXiv:1908.07795*, 2019.

[13] Z. Zhao, S. Zhu, and K. Yu, "Data augmentation with atomic templates for spoken language understanding," *arXiv preprint arXiv:1908.10770*, 2019.

[14] E. Cho, H. Xie, J. P. Lalor, V. Kumar, and W. M. Campbell, "Efficient semi-supervised learning for natural language understanding by optimizing diversity," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 1077–1084.

[15] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in neural information processing systems*, 2014, pp. 3581–3589.

[16] A. Anaby-Tavor, B. Carmeli, E. Goldbraich, A. Kantor, G. Kour, S. Shlomov, N. Tepper, and N. Zwerdling, "Do not have enough data? Deep learning to the rescue!" in *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[17] V. Kumar, A. Choudhary, and E. Cho, "Data augmentation using pre-trained transformer models," *arXiv preprint arXiv:2003.02245*, 2020.

[18] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[20] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, "Unified language model pre-training for natural language understanding and generation," *arXiv preprint arXiv:1905.03197*, 2019.

[21] C. Zhu, M. Zeng, and X. Huang, "Sdnet: Contextualized attention-based deep network for conversational question answering," *arXiv preprint arXiv:1812.03593*, 2018.

[22] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," *EMNLP*, 2019.

[23] Y.-A. Chung, C. Zhu, and M. Zeng, "Semi-supervised speech-language joint pre-training for spoken language understanding," *arXiv preprint arXiv:2010.02295*, 2020.

[24] G. Tur, D. Hakkani-Tür, and L. Heck, "What is left to be understood in atis?" in *2010 IEEE Spoken Language Technology Workshop*. IEEE, 2010, pp. 19–24.

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

[26] Y.-N. Chen, D. Hakanni-Tür, G. Tur, A. Celikyilmaz, J. Guo, and L. Deng, "Syntax or semantics? Knowledge-guided joint semantic frame parsing," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 348–355.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *arXiv preprint arXiv:1904.09751*, 2019.

[29] K. M. Yoo, Y. Shin, and S.-g. Lee, "Data augmentation for spoken language understanding via joint variational generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 7402–7409.