Toward Open-domain Slot Filling via Self-supervised Co-training

Adib Mosharrof
University of Kentucky
Lexington, KY, USA
amo304@g.uky.edu

Moghis Fereidouni
University of Kentucky
Lexington, KY, USA
mfe261@uky.edu

A.B. Siddique
University of Kentucky
Lexington, KY, USA
siddique@cs.uky.edu

ABSTRACT

Slot filling is one of the critical tasks in modern conversational systems. The majority of existing literature employs supervised learning methods, which require labeled training data for each new domain. Zero-shot learning and weak supervision approaches, among others, have shown promise as alternatives to manual labeling. Nonetheless, these learning paradigms are significantly inferior to supervised learning approaches in terms of performance. To minimize this performance gap and demonstrate the possibility of open-domain slot filling, we propose a **S**elf-supervised **Co-t**raining framework, called SCot, that requires zero in-domain manually labeled training examples and works in three phases. Phase one acquires two sets of complementary pseudo labels automatically. Phase two leverages the power of the pre-trained language model BERT, by adapting it for the slot filling task using these sets of pseudo labels. In phase three, we introduce a self-supervised co-training mechanism, where both models automatically select high-confidence soft labels to further improve the performance of the other in an iterative fashion. Our thorough evaluations show that SCot outperforms state-of-the-art models by 45.57% and 37.56% on SGD and `MultiWoZ` datasets, respectively. Moreover, our proposed framework SCot achieves comparable performance when compared to state-of-the-art fully supervised models.

CCS CONCEPTS

• **Computing methodologies → Information extraction**; *Lexical semantics*; **Transfer learning**.

KEYWORDS

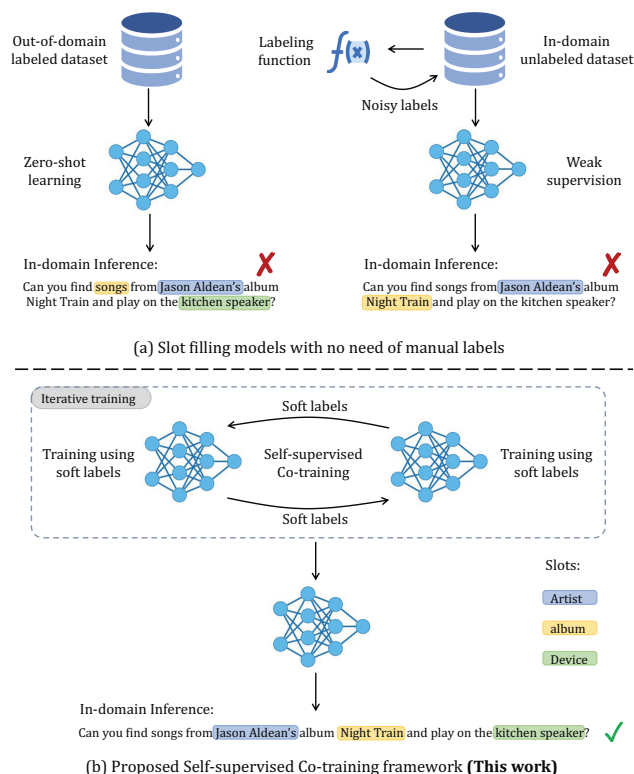open-domain slot filling, co-training, weak supervision.

1 INTRODUCTION

Conversational Systems empower users by facilitating interactions through expressive and accessible natural language. These systems

(a) Slot filling models with no need of manual labels

(b) Proposed Self-supervised Co-training framework **(This work)**

Figure 1: (a) State-of-the-art approaches that do not require manual labels show poor performance as compared to supervised learning models. (b) The proposed self-supervised co-training framework shows comparable performance to state-of-the-art supervised learning models.

need to accurately perform *slot filling*, among other tasks, in order to fulfill users' requests in task-oriented dialog systems or make meaningful recommendations in conversational recommendation systems. Slot filling is the task of detecting contiguous spans of text that correspond to a slot of interest, usually referred to as a query parameter. Figure 1 highlights values for various slots. Due to the vast diversity in users' expressions, slot filling is a challenging task that has been studied extensively in the supervised setting [2, 15, 16, 23, 30, 50, 51, 54]. The critical drawback of the supervised learning slot filling methods is that they require a massive amount of labeled training data for each new domain (and slot type) which can be costly and time-consuming. This *unscalable* requirement has become a major bottleneck and renders supervised learning methods ineffective because practical conversational systems need to continuously expand their conversational capabilities (e.g., `Alexa Skills`) and be able to converse in new emerging domains.
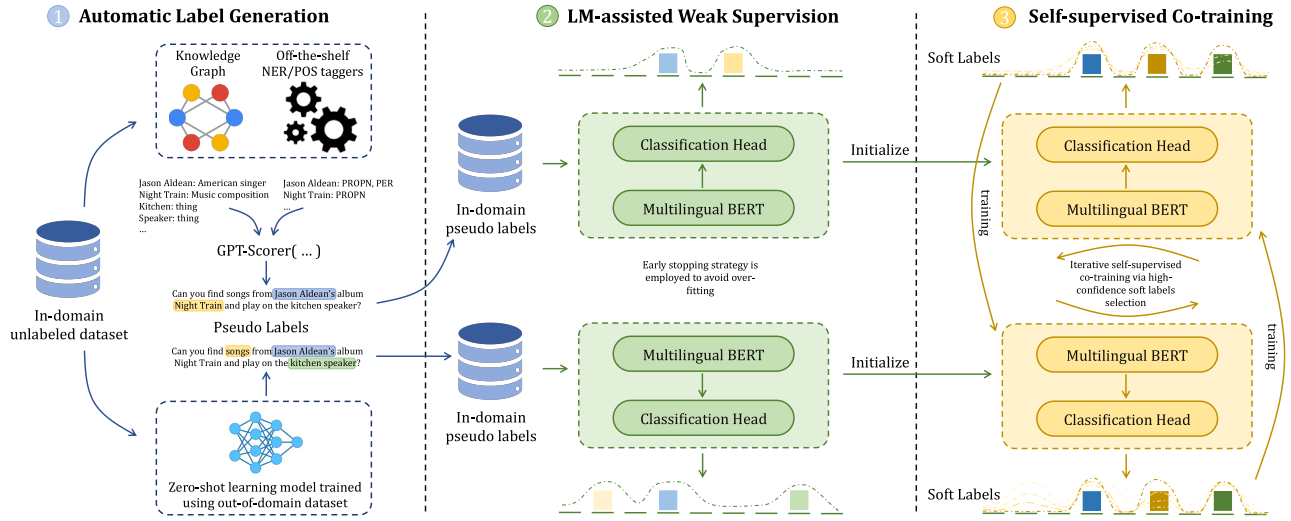
Figure 2: Overview of SCot. The self-supervised co-training phase facilitates significant performance gains.

Recently, researchers have proposed a wide range of approaches to address the label scarcity issue for the slot filling task, such as zero-shot learning [27, 43] and weak supervision [8, 20]. Zero-shot learning methods [44, 45] are capable of classifying instances of new unseen classes at inference time, they had not encountered during training, and thus do not require training data for each new domain. Weak supervision approaches eliminate the need for manually labeled data by automatically generating noisy labels using a heuristic labeling function, powered by almost freely-available external knowledge bases, off-the-shelf models (e.g., NER models), or their combination. Both learning paradigms have emerged as promising alternatives to manual labeling of the data for every single domain. However, both approaches suffer from significantly poor performance when compared to the supervised methods. Motivated by this huge performance gap and the pressing need for slot filling models with no restriction on the domain, we explore the possibility of developing an open-domain slot filling model with comparable performance to the supervised models without incurring the cost and effort associated with manual annotations.

We introduce a **S**elf-supervised **Co-t**raining framework, SCot, that automatically generates pseudo labels for initial supervision, then it selects high-confidence soft labels to gradually guide the framework to superior prediction performance by leveraging the power of pre-trained language models. The proposed framework is also presented in Figure 2. SCot employs the pre-trained multilingual BERT and works in three phases. Phase one acquires two sets of complementary pseudo labels automatically, with no need for human intervention. The first set of pseudo labels is generated using a combination of knowledge graphs, off-the-shelf taggers (e.g., NER/POS taggers), and a GPT-2-based [34] scorer. The other set of pseudo labels is obtained using a zero-shot slot filling model that has been trained on the out-of-domain dataset and does not require a single in-domain training example. Essentially, the first set of labels exploits freely available and domain-independent knowledge sources to acquire labels, whereas the other set of labels leverages the power of zero-shot learning to pick up generalizable patterns of how slot values are mentioned in other domains.

Phase two leverages the power of the pre-trained language models by adapting the multilingual BERT to the task of slot filling. This phase fine-tunes the model using pseudo labels from phase one with the early stopping strategy to avoid overfitting to the noisy labels. This phase overcomes the challenge of incomplete and noisy labels to some extent. Using these sets of pseudo labels from phase one, both models are trained that are capable of making better quality predictions, so pseudo labels from phase one are dropped. Phase three introduces an iterative self-supervised co-training mechanism that further trains the models on the soft labels. This phase also proposes a peer training approach where one peer automatically generates soft labels for training the other peer in the next iteration. Using this progressive self-supervised training strategy, we can exploit the patterns of slot values in other domains as well as information from external knowledge sources. Moreover, the high-confidence soft label selection mechanism facilitates efficient learning and enables better model fitting through quality soft labels. Phase three is the most critical phase that facilitates significant performance gains for the open-domain slot filling task.

Our evaluations using two benchmark datasets, SGD [36] and MultiWoZ [52], that span up to 20 domains, demonstrate that SCot achieves significant performance gains as compared to the learning paradigms [8, 20, 27, 31, 43, 53] that do not require in-domain labeled data for training. Moreover, SCot with zero in-domain labeled training examples [29], achieves *competitive* performance as compared to *fully* supervised slot filling models [7, 11].

The contributions of this work are summarized as follows:

- We propose an iterative self-supervised co-training framework to enable open-domain slot filling that requires zero in-domain manually labeled training examples.
- We demonstrate that combining the power of zero-shot learning with external knowledge sources can lead to superior performance and better model fitting.
- Our comprehensive experimental analysis using two public datasets SGD and MultiWoZ shows that SCot outperforms state-of-the-art models consistently and achieves comparable performance to fully supervised models.

2 PRELIMINARIES

2.1 Problem Formulation

Given a natural language text $\mathcal{X}_i = (x_1, x_2, \cdots, x_{|\mathcal{X}_i|})$, the value for a slot $\mathcal{S}_r$ is a contiguous segment of tokens $(x_j, \cdots, x_k)$ such that $0 \leq j \leq k \leq |\mathcal{X}_i|$. Slot filling is generally formulated as a token classification task that predicts the labels $\mathcal{Y}_i = (y_1, y_2, \cdots, y_{|\mathcal{X}_i|})$ for the input $\mathcal{X}_i$ using the IOB labeling scheme [35]. Specifically, the first token that points to the value of a slot $\mathcal{S}_r$ is provided the label B-$\mathcal{S}_r$, the other tokens inside the slot value are given the label I-$\mathcal{S}_r$, whereas non-slot tokens are labeled as O. In the traditional supervised setting, a dataset with n token-level labeled training examples $\{(\mathcal{X}_i, \mathcal{Y}_i)\}_{i=1}^n$ is provided to train the model, whereas acquiring such dataset for each new slot is expensive and labor-intensive. In this work, we focus on a challenging problem setup where we are provided with unlabeled training examples $\{\mathcal{X}_i\}_{i=1}^n$ only, and corresponding labels $\{\mathcal{Y}_i\}_{i=1}^n$ are not available. In contrast to the named entity recognition (NER) task where only a limited set of entities (e.g., LOC, PER, ORG) is considered, the slot filling task is more demanding because a wide range of slot types (e.g., price, destination, departure_time in the Buses domain) may appear in the natural language text.

2.2 Pre-trained NLP Models

This work uses several pre-trained and readily-available natural language processing (NLP) models. In the following, we provide a brief overview of these models.

Pre-trained POS tagger. The POS tagger is trained to predict part-of-speech tags for each token in a given natural language text, such as PROPN, VERB, and ADJ. The POS tags provide useful syntactic cues for slot value identification from natural language texts. Since POS tagger is domain-independent, it facilitates automatic label generation for the task of open-domain slot filling. For example, the values for slots are generally tagged as proper nouns by the POS tagger. We use the production-ready pre-trained POS tagger[1] from SpaCy that has produced a high performance for the task.

Pre-trained NER model. The NER model tags each token in the natural language text for the mentions of the four entities: LOC, PER, ORG, and MISC. Although a limited number of entities are tagged by the NER model, these entities can be used to provide cues for many slots (though not all). For example, LOC can be used to narrow down the search to source or destination slots in the Buses domain. Nevertheless, it still proves challenging to automatically assign labels to the given natural language text for the slot filling task. Furthermore, open-domain NER models are still evolving and may miss mentions of the entities, especially slots of interest in the target domain, which results in a low recall. Nonetheless, the NER model facilitates automatic labeling and reduces the task's complexity. We leverage the pre-trained NER model[2] from SpaCy.

Pre-trained GPT-2. GPT-2 [34] is trained for predicting the next token, i.e., autoregressive objective, using 40 GB textual data of WebText dataset in an unsupervised setting. It also follows transformer architecture [48]. For a given natural language sequence $(x_1, \cdots, x_{|x|})$ where token x_i is sampled from a fixed vocabulary

of tokens, factorizing the joint probabilities over tokens as a product of conditional probabilities [3] leads to $\prod_{i=1}^{|x|} p(x_i | x_1, \cdots, x_{i-1})$. This approach not only enables tractable sampling, but also allows for computing $p(x)$, and estimating any conditionals of the form: $p(x_{i-k}, \cdots, x_i | x_1, \cdots, x_{i-k-1})$. In this work, we use the pre-trained GPT-2 model[3] to score a given natural language sequence by computing the log probabilities of the tokens to select the best hypothesis for automatic label generation in phase one of SCot.

Pre-trained BERT. BERT [10] is an unsupervised model that follows the transformer architecture [48] by conditioning on bidirectional contexts. In contrast to traditional unidirectional language modeling [3], BERT has been trained to condition on both left and right contexts simultaneously across all layers to learn deep bidirectional representations from the unlabeled text. It is trained for masked language model (MLM) and the next sentence prediction (NSP) task. In the MLM task, for a given natural language sequence $(x_1, x_2, \cdots, [MASK], \cdots, x_{|x|})$, the goal is to predict $[MASK]$ tokens successfully, whereas the next sentence prediction task focuses on whether sentence B naturally follows sentence A, given two natural language sequences $(x_1, \cdots, x_{|x|})$ and $(x'_1, \cdots, x'_{|x'|})$, using the BooksCorpus (i.e., 800 million words) [55] and textual data from Wikipedia English articles (i.e., 2,500 million words). The pre-trained BERT has produced state-of-the-art results on many natural language understanding benchmarks, including slot filling task [7] in the supervised setting. In our proposed framework SCot, we employ the pre-trained multilingual BERT[4] as the backbone model and adapt it for open-domain slot filling task.

2.3 Knowledge Graphs

Knowledge graphs (KG) are used to store information about entities or concepts and are often used as knowledge sources for a range of NLP tasks. In this work, we use Google Knowledge Graph Search API [5] to retrieve type information about entity mentions in the natural language. Information from KG facilitates the generation of several hypotheses about slot labels in phase one of the framework. Moreover, matching the n-grams in the natural language text to the KG entities also assists in improving recall for the cases where entities or concepts are missed by POS and NER taggers.

2.4 Zero shot Learning

Zero-shot learning models have the capability to generalize to new unseen classes, not encountered during training. Zero-shot learning is one of the promising approaches to address the data scarcity issue. A few zero-shot learning models [1, 27, 39, 43] have been proposed for the slot filling task in the literature that has shown encouraging results. Yet, there exists a huge performance gap compared to the supervised slot filling models. Nonetheless, zero-shot slot filling models can generate decent quality labels (though noisy) for new unseen domains and slots at *zero cost*. In this work, we pre-train a zero-shot slot filling model [27] on the out-of-domain slot filling dataset. Since the zero-shot slot filling model can infer labels for unseen classes, we use the trained model to generate pseudo labels for the target domain/dataset in phase one.

---

[1]https://spacy.io/api/annotation#pos-tagging
[2]https://spacy.io/api/annotation#named-entities

[3]https://huggingface.co/gpt2
[4]https://huggingface.co/bert-base-multilingual-cased
[5]https://developers.google.com/knowledge-graph

3 PROPOSED FRAMEWORK: SCOT

In this section, we introduce a novel self-supervised co-training framework SCot. The proposed framework is illustrated in Figure 2 and works in three phases. Phase one automatically generates two sets of pseudo labels. We use a combination of off-the-shelf pre-trained POS and NER taggers, knowledge graph, and GPT-2 scorer for generating the first set of pseudo labels automatically without any hand-crafted rules for matching the slot values. The other set of pseudo labels is acquired through a zero-shot slot filling model [27], trained on the out-of-domain dataset. It is critical to emphasize that both sets of labels are noisy and incomplete which poses serious challenges to training effective models for the task of open-domain slot filling. Phase two fine-tunes the pre-trained BERT to the slot filling task that effectively transfers the knowledge from the pre-trained language model (LM) to overcome the issue of label incompleteness to some extent. Further, we employ the early stopping technique to minimize the noise in the labels. The output of this phase is two BERT models that can generate soft labels for self-supervision during co-training in phase three. Phase three leverages the fine-tuned models and further trains them in an iterative fashion. Specifically, the proposed peer training approach facilitates high-confidence soft label selection for the other peer to perform training. This phase progressively reduces the noise in the labels and enables effective model fitting.

3.1 Phase One: Automatic Label Generation

To acquire the first set of labels, we perform the following steps. First of all, off-the-shelf trained POS and NER taggers are used to predict initial estimates of the slot values irrespective of the slot types. Then, the type information of the slot values is queried from the KG and the slot value is tagged for the most appropriate slot in the target domain. This approach, however, produces low recall. To expand the candidate slot values, we generate n-grams of the natural language text and employ a partial matching scheme to query the KG for type information (e.g., Jason Aldean = American singer) of the n-grams if the entry exists. This process generates multiple overlapping hypotheses about the slot values. We replace a span of text that corresponds to a slot value by its type information and a GPT-2 based scorer (see Section 2.2) is used to select the best candidate based on the fluency of the text. Naturally, if a token (or span of tokens) is replaced by its type, the sentence should score higher as compared to the case where an inappropriate substitution is performed. We select the best hypothesis if the score is greater than the threshold. Intuitively, the candidate selection threshold can automatically be searched based on a small validation set from the target domain, making the label generation process fully automatic. The other set of noisy labels is acquired by the zero-shot slot filling model [27] that has been trained using an out-of-domain dataset. It is important to highlight that the zero-shot slot filling model does not require any labeled in-domain training example. To summarize the automatic label generation phase, both sets of labels are acquired in a fully automatic fashion without any hand-crafting.

In contrast to previous work in weak supervision [12, 14, 17, 38] that obtains a single set of noisy labels and then propose techniques to overcome the challenge of fitting an effective model to the noisy labels, we acquire two sets of complementary labels. The choice

of these two sets of labels is guided by the intuition that they should be complementary and the models trained on these sets of labels should be able to share complementary information with the other to improve the performance in the later phases of the framework. Essentially, the first set of labels carries information from external knowledge sources, whereas the labels generated through the pre-trained zero-shot slot filling model capture how the slot values are mentioned in other domains. To further elaborate on the motivation and our process for the first set of labels (i.e., labels using KG and other NLP models), the pre-trained LMs have been shown to have a great deal of knowledge [33], thus should be capable of generating automatic labels with no need of external KG. To the best of our knowledge, there exists no work that shows that accurate token-level automatic labeling (e.g., slot filling task) is possible with pre-trained LMs. Moreover, such approaches would require heavy prompting in each new target domain, whereas our label generation process is fully automatic and only relies on the readily-available pre-trained NLP models and external KG.

3.2 Phase Two: LM-assisted Weak Supervision

Since we do not have access to dataset $\{(\mathbf{X}_n, \mathbf{Y}_n)\}_{n=1}^{N}$ with true ground-truth labels. We use pseudo labels generated in phase one, $\{(\mathbf{X}_n, \mathbf{D}_n)\}_{n=1}^{N}$, to learn $f_{m,c}(\cdot; \cdot)$ that outputs the probability of the m-th token to take on class c. We learn $f_{m,c}(\cdot; \cdot)$ by minimizing the following loss over the noisy dataset $\{(\mathbf{X}_n, \mathbf{D}_n)\}_{n=1}^{N}$:

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{n=1}^{N} \ell(\mathbf{D}_n, f(\mathbf{X}_n; \theta)),$$

where $\ell(\mathbf{D}_n, f(\mathbf{X}_n; \theta)) = \frac{1}{M} \sum_{m=1}^{M} -\log f_{m,d_{n,m}}(\mathbf{X}_n; \theta)$. We employ the pre-trained multilingual BERT with token-level classification head that uses Adam optimizer [21, 26] with early stopping and multiple random initializations.

Since slot filling task is similar to the MLM training objective of the BERT, we employ pre-trained BERT as the backbone model. That is, MLM's goal is to predict the masked tokens using bidirectional contexts. Similarly, slot filling tries to predict the label for a token leveraging both left and right contexts simultaneously, which makes the pre-trained BERT an ideal model of choice that greatly facilitates minimizing incomplete labels. It is important to highlight that our automatically generated labels are not only incomplete but also potentially wrong. The training strategies employed in this phase minimize the noise in the label to some extent. Specifically, early stopping can provide a strong regularization and would not let the model overfit to the noisy labels, especially wrong labels. Moreover, early stopping does not let the model forget the knowledge in the pre-trained model. Similarly, multiple random initializations enforce robustness. Since the model is fine-tuned on the noisy labels, averaging the predictions of multiple models for each token ensures that wrong labels end up with low probabilities and true labels consistently achieve high probabilities. Using the above-mentioned strategies, we train two slot filling models, which we call the peers. The peer one is trained on the first set of pseudo labels that were generated using POS and NER taggers, KG, and the GPT-2 scorer in phase one. Similarly, peer two is trained using the predictions of the zero-shot slot filling model [27]. Both models have the same architecture and follow the same training procedures.

Table 1: Dataset statistics.

Dataset	Dataset Size	Vocab. Size	Avg. Length	# of Domains	# of Slots
SGD	188K	33.6K	13.8	20	240
MultiWoZ	67.4K	10.5K	13.3	8	61

3.3 Phase Three: Self-supervised Co-training

We introduce an iterative peer training algorithm where both peers generate high-confidence soft labels for training the other peer in the next iteration. Theoretically, these peers can be anything, but in this work, we explore two of the most promising directions that have shown the promise to minimize the need for manual labeling for the task: zero-shot learning and distant supervision. This phase uses a self-supervised co-training scheme to exploit the patterns of slot values from other domains through the labels generated by the zero-shot filling model (i.e., peer two) [27] as well as utilize the knowledge in external KGs and pre-trained models via labels provided by the peer one. Specifically, we initialize the peers trained in phase two and use their pseudo labels to kick-start training in this phase. Specifically, peer one $f_{m,c}(\cdot; \theta_{\text{p1}})$ would generate labels $\{\tilde{\mathbf{Y}}_n^{(t)} = [\tilde{y}_{n,1}^{(t)}, ..., \tilde{y}_{n,m}^{(t)}]\}_{n=1}^N$ for peer two $f_{m,c}(\cdot; \theta_{\text{p2}})$ at the t-th iteration by:

$$\tilde{y}_{n,m}^{(t)} = \arg\max_c f_{m,c}(\mathbf{X}_n; \theta_{\text{p1}}^{(t)}).$$

Based on these labels, the peer two can be fine-tuned by:

$$\hat{\theta}_{\text{p2}}^{(t+1)} = \arg\min_\theta \frac{1}{N} \sum_{n=1}^N \ell(\tilde{\mathbf{Y}}_n^{(t)}, f(\mathbf{X}_n; \theta)).$$

Similarly, peer two $f_{m,c}(\cdot; \theta_{\text{p2}})$ would generate pseudo labels for peer one $f_{m,c}(\cdot; \theta_{\text{p1}})$ that are used to fine-tune peer one. We also notice that it is beneficial to stop early during this phase as well, to improve the model fitting and gradually reduce the noise associated with the automatically generated labels. Since pseudo labels are refined gradually in an iterative way, both peers can benefit from the knowledge contained within the labels of the other while avoiding overfitting. Furthermore, as an alternative to pseudo labels, we also generate soft labels that are used for confidence re-weighting. The high-confidence soft label selection strategy enables better model fitting and efficient learning via better quality of the automatic labels. Specifically, for the given m-th token in the n-th training example, the probability for all classes C is $[f_{m,1}(\mathbf{X}_n; \theta), ..., f_{m,C}(\mathbf{X}_n; \theta)]$. Following [49], at t-th iteration, peer one generates soft labels, $\{\mathbf{S}_n^{(t)} = [\mathbf{s}_{n,m}^{(t)}]_{m=1}^M\}_{n=1}^N$, as given below:

$$\mathbf{s}_{n,m}^{(t)} = [s_{n,m,c}^{(t)}]_{c=1}^C = \left[\frac{f_{m,c}^2(\mathbf{X}_n; \theta_{\text{peer1}}^{(t)})/p_c}{\sum_{c'=1}^C f_{m,c'}^2(\mathbf{X}_n; \theta_{\text{peer1}}^{(t)})/p_{c'}} \right]_{c=1}^C$$

where $p_c = \sum_{n=1}^N \sum_{m=1}^M f_{m,c}(\mathbf{X}_n; \theta_{\text{p1}}^{(t)})$ computes the frequency of the tokens for the c-th class. Then, peer two $f(\cdot; \theta_{\text{p2}}^{(t+1)})$ is fine-tuned by:

$$\theta_{\text{p2}}^{(t+1)} = \arg\min_\theta \frac{1}{N} \sum_{n=1}^N \ell_{\text{KL}}(\mathbf{S}_n^{(t)}, f(\mathbf{X}_n; \theta)),$$

where $\ell_{\text{KL}}(\cdot, \cdot)$ is the KL-divergence-based loss:

$$\ell_{\text{KL}}(\mathbf{S}_n^{(t)}, f(\mathbf{X}_n; \theta)) = \frac{1}{M} \sum_{m=1}^M \sum_{c=1}^C -s_{n,m,c}^{(t)} \log f_{m,c}(\mathbf{X}_n; \theta).$$

Moreover, we also investigate selecting tokens that have high confidence. For instance, we pick high-confidence tokens from the m-th input example at the t-th iteration by $H_n^{(t)} = \{m : \max_c s_{n,m,c}^{(t)} > \epsilon\}$, where $\epsilon \in [0, 1]$ is a threshold that can be searched based on a small validation set. Then, peer two $f(\cdot; \theta_{\text{p2}}^{(t+1)})$ is fine-tuned by:

$$\theta_{\text{p2}}^{(t+1)} = \arg\min_\theta \frac{1}{N|H_n^{(t)}|} \sum_{n=1}^N \sum_{m \in H_n^{(t)}} \sum_{c=1}^C -s_{n,m,c}^{(t)} \log f_{m,c}(\mathbf{X}_n; \theta).$$

This phase improves the robustness to effectively fit the model for tokens with high confidence. Both peers keep sharing information and their confidence by producing soft labels for their counterparts until they approximate to the true labels while employing early stopping and scheduled learning rates. It is important to remind that phase three is the most important phase that progressively reduces noise from the labels to a great extent and enables superior performance for the task of open-domain slot filling.

4 EXPERIMENTAL SETUP

4.1 Datasets

We conducted extensive experiments to evaluate the performance of our model SCot using two well-known public benchmark datasets: Schema Guided Dialogue (SGD) [36] and Multi-Domain Wizard-of-Oz (MultiWoZ) [52]. The important statistics about datasets are presented in Table 1.

SGD [36]: SGD is one of the most challenging and comprehensive publicly available datasets for evaluating tasks related to dialog systems. It consists of dialog utterances from 20 domains and covers 240 slots. Following [43], we pre-processed the dataset for the task of slot filling. Moreover, all the domains that have less than 1850 utterances (i.e., < 1% of the dataset) have been merged into a single domain, named "Others" in our experiments.

MultiWoZ [52]: MultiWoZ has been extensively used to evaluate a wide range of tasks in dialog systems. In our experiments, we used MultiWoZ 2.2 which consists of dialog utterances for 61 slots in 8 domains. We followed the experimental setup of [43] and merged all the utterances into a single domain, called "Others", which accounted for < 1% of the dataset.

4.2 Evaluation Methodology

The Slot F1 score is the standard metric for evaluating the slot filling task, that has been used in our experiments. We present our evaluations for the following settings.

Evaluation using a single domain in the dataset. We evaluate using each domain separately in both datasets. The zero-shot learning model is trained using all the domains in the dataset except

Table 2: Slot F1 score for all the domains in SGD dataset. The zero-label proposed framework SCot **achieves competitive performance to the fully supervised methods for each domain in the dataset.**

Domains ↓	Unsupervised Learning		Weak Supervision		Zero-shot Learning		Few-shot	Supervised Learning		Self-supervised
	DSI	RCAP	Inter-Slot	Merge-Select	Coach	LEONA	GPT-3	JointBERT	Seq2Seq-DU	SCot (this work)
Buses	0.2533	0.3705	0.3006	0.2860	0.6281	0.6978	0.5054	0.9161	**0.9318**	0.7701
Calendar	0.3963	0.5253	0.4810	0.5134	0.6023	0.7436	0.6269	**0.9862**	0.9605	0.9725
Events	0.2778	0.5162	0.2562	0.3922	0.5486	0.7619	0.6658	**0.9403**	0.9346	0.8773
Flights	0.3605	0.4154	0.4240	0.5220	0.4898	0.5901	0.7166	**0.9612**	0.9286	0.9026
Homes	0.4356	0.4747	0.3558	0.5586	0.6235	0.7698	0.7734	0.9498	0.9860	**0.9862**
Hotels	0.2630	0.3584	0.1566	0.4252	0.7216	0.7677	0.6825	**0.9219**	0.9096	0.8542
Movies	0.3407	0.3631	0.1414	0.5420	0.5537	0.7285	0.5377	**0.9423**	0.9281	0.8728
Music	0.4146	0.5613	0.3134	0.4844	0.5786	0.7613	0.6098	0.9387	0.9373	**0.9419**
RentalCars	0.3661	0.4964	0.1664	0.4114	0.6576	0.7389	0.5339	0.8753	**0.9232**	0.9123
Restaurants	0.3537	0.4457	0.1316	0.6534	0.7195	0.7574	0.5513	**0.9534**	0.9345	0.8928
RideSharing	0.3642	0.5164	0.2860	0.6628	0.7273	0.8172	0.7589	0.9490	0.9429	**0.9712**
Services	0.3902	0.5834	0.3342	0.6598	0.7607	0.8182	0.5819	**0.9714**	0.9151	0.9688
Travel	0.3606	0.5894	0.1950	0.4618	0.8403	0.9234	0.6715	0.9344	0.9226	**0.9552**
Weather	0.3498	0.5363	0.1484	0.6390	0.6003	0.8223	0.7040	0.9820	0.8981	**0.9821**
Others	0.3305	0.4323	0.2824	0.5026	0.4921	0.5592	0.6660	**0.9662**	0.9353	0.9291
Average	0.3519	0.4823	0.2636	0.5151	0.6466	0.7642	0.6371	**0.9444**	0.9324	0.9186

the target domain (i.e., being used for evaluation) and the model's predictions for the target domain are used as one set of automatic labels. Since there exists a wide range of domains in both datasets, the evaluation is used to validate the robustness of the proposed framework across domains.

Evaluation using the full dataset. We also perform evaluations using full datasets. In this setup, the zero-shot learning model is trained using the dataset that is not being used for evaluation. For example, when we conduct an evaluation on SGD dataset, the zero-shot slot filling model is trained on MultiWoZ, and its predictions for the SGD dataset are considered as automatic labels.

4.3 Competing Methods

We compare SCot with state-of-the-art (SOTA) methods that employ a variety of learning paradigms, including supervised learning for the slot filling task.

DSI [31]: A SOTA unsupervised learning method that automatically mines the slot values from unlabeled data by building two neural latent variable models.

RCAP [53]: An unsupervised pipeline that works in coarse-to-fine fashion for automatically inferring intents as well as slots labels via sequence labeling, clustering, and Apriori algorithm.

Inter-Slot [8]: An automatic labeling approach that leverages a random walk inference algorithm for jointly capturing token-to-token, token-to-slot, and slot-to-slot relationships by building word-based lexical and slot-based semantic graphs and combining them using dependency grammar.

Merge-Select [20]: A weak supervision-based approach, which automatically identifies candidate slot values relevant to a given domain using clustering algorithms, then these candidates are used to train a neural network-based slot tagger.

Coach [27]: A zero-shot slot filling model that adapts a coarse-to-fine approach. First, it identifies potential slot values irrespective of the slot type and then matches the slot values to appropriate slots using the representation of the slot description. In this work, we use this model to generate one set of pseudo labels.

LEONA [43]: A zero-shot learning approach, that exploits the embeddings of POS and NER taggers as well as learns a contextual tokens-slot similarity function from seen domains to make predictions about unseen domains.

GPT-3 [5]: A SOTA model that has shown zero-shot learning capabilities for many NLP tasks. In this work, we use this model in the context of few-shot learning and provide five labeled training examples as prompts for each domain (or dataset) and expect the model to make inferences for the given domain.

JointBert [7]: A supervised learning method that fine-tunes BERT for intent detection and slot filling tasks jointly. We perform this comparison to understand the performance gap between supervised models that use expensive labeled training data and our proposed zero-label approach.

Seq2Seq-DU [11]: a supervised learning approach that formulates the task of dialog understanding as sequence-to-sequence and fine-tunes two BERT encoders, one for utterance and the other for schema descriptions, and jointly models intents and slots.

5 RESULTS

5.1 Quantitative Analysis

Evaluation using a single domain in the dataset. Tables 2 and 3 present F1 scores for each domain in SGD and MultiWoZ datasets, respectively. The best F1 score is shown as bold including the supervised learning methods that were trained using labeled training data for all the domains. We also highlight the best model by underlining the results excluding the supervised models. Our proposed framework SCot achieves significant performance gains against all the competing models (excluding supervised models) including few-shot GPT-3 on both datasets for every single domain consistently. Specifically, SCot outperforms SOTA models by 20.20% and 37.05% on average for all the domains on SGD and MultiWoZ datasets, respectively. This performance improvement against all the unsupervised learning, weak supervision, zero-shot learning, and few-shot learning approaches can be attributed to the iterative refinement of the noisy labels throughout all the phases of the framework, especially the self-supervised co-training phase. Moreover, we also notice that our proposed approach which does not use any in-domain labeled data achieves competitive performance to fully supervised models that were trained using labeled training data. Specifically, the performance gap is only 2.58 percentage points on average for all the domains in SGD dataset and 6.31 percentage points for MultiWoZ. Interestingly, SCot achieves slightly

Table 3: Slot F1 score for all the domains in MultiWoZ dataset.

Domains	Unsupervised Learning		Weak Supervision		Zero-shot Learning		Few-shot	Supervised Learning		Self-supervised
	DSI	RCAP	Inter-Slot	Merge-Select	Coach	LEONA	GPT-3	JointBERT	Seq2Seq-DU	SCot (this work)
Hotels	0.6007	0.7182	0.4904	0.5801	0.6481	0.7319	0.6967	0.9292	**0.9236**	0.9012
Restaurants	0.5162	0.7318	0.4548	0.5512	0.6137	0.7387	0.7256	**0.9687**	0.9475	0.8991
Trains	0.5994	0.6637	0.4288	0.5612	0.6804	0.7926	0.5493	**0.9524**	0.8788	0.8741
Attractions	0.5441	0.6446	0.3735	0.4305	0.3029	0.3834	0.6057	0.9316	**0.9880**	0.8393
Taxi	0.5903	0.4982	0.2968	0.3827	0.1260	0.1824	0.5175	**0.9496**	0.7991	0.8471
Others	0.5148	0.5990	0.2638	0.3132	0.1201	0.1721	0.6689	**0.9314**	0.9183	0.9236
Average	0.5609	0.6426	0.3847	0.4698	0.4152	0.5002	0.6273	**0.9438**	0.9092	0.8807

Table 4: Slot F1 score for the full datasets: SGD and MultiWoZ.

Datasets	Unsupervised			Weak Supervision		Zero-shot Learning		Few-shot	Supervised Learning		Self-supervised
	DSI	RCAP	KG	Inter-Slot	Merge-Select	Coach	LEONA	GPT-3	JointBERT	Seq2Seq-DU	SCot (this work)
SGD	0.3352	0.4539	0.4591	0.3274	0.5512	0.1102	0.1621	0.6015	**0.9156**	0.9028	0.8756
MultiWoZ	0.4961	0.5937	0.4839	0.4483	0.5219	0.1903	0.2884	0.6123	**0.9323**	0.8923	0.8423
Average	0.4157	0.5238	0.4715	0.3879	0.5366	0.1503	0.2253	0.6069	**0.9240**	0.8976	0.8590

better performance than supervised models for five domains in SGD dataset. This incredible performance improvement is due to the outstanding co-training approach of SCot that progressively allows the peers to refine the noisy labels. Specifically, the zero-shot learning labels carry information from other (seen) domains about how slot values are mentioned across domains, whereas the other set of labels carries the knowledge from KGs and other NLP models. The self-supervised co-training phase facilitates combining this information progressively by sharing high-confidence information among peers. For example, SCot achieves a better F1 score than supervised models in the domain Music. Upon further analysis, we noticed that most of the values for slots song_name and artist are present in the KG, whereas the domain Movies is quite similar to Music and the zero-shot model successfully transferred knowledge to SCot.

Evaluation using the full dataset. Table 4 presents the results for the full datasets. We note that the proposed zero-label framework has, once more, comparable performance to the fully supervised learning methods. For example, the average F1 score is only 6.5 percentage points lower than the best-performing supervised learning model. It is important to highlight that this evaluation setup is more challenging because SCot automatically generates labels for the whole dataset (i.e., SGD dataset has 240 slot types) with no human intervention and labeling through heuristic functions or zero-shot learning models is extremely error-prone because of the big number of classes, i.e., 240 in case of SGD dataset. To highlight the importance of the gradual noise reduction approach of SCot, we also experiment with a KG baseline that uses our approach from phase one to generate pseudo labels and then uses Merge-Select [20] to further refine the labels. If we compare our proposed approach to the methods that do not have access to labeled data (or only a few labeled training examples), there is a significant performance gain for SCot. Specifically, SCot outperforms unsupervised models by 92.91% and 41.87% on SGD and MultiWoZ datasets, respectively. Similarly, other learning paradigms show inferior performance when compared to SCot. For example, the F1 score of the weak supervision models is 32.44 and 32.04 percentage points lower than our proposed approach for SGD and MultiWoZ datasets, respectively. Last but not least, our performance improvement over SOTA results is statistically significant, with a P-value < 0.01 (i.e., highly significant) for all evaluation settings.

Table 5: Ablation Study: slot F1 score when certain components are removed from SCot

Configuration	SGD	MultiWoZ
w/o KG Labels	0.8231	0.7838
w/o POS / NER Taggers	0.7954	0.7762
w/o KG, POS/NER Taggers	0.5767	0.5923
w/o Zero-shot Labels	0.5091	0.4786
w/o Co-training	0.5712	0.5681
w/o Soft labels	0.7145	0.6992

5.2 Ablation study

Effect of removing a certain component. Since our proposed framework SCot uses several mature components, it is critical to investigate the role of each component. Table 5 presents the results on both datasets by removing a certain component. For example, if we remove KG from phase one and generate automatic labels, the performance slightly declined. Similarly, removing the pre-trained POS and NER taggers also makes the performance worse. Moreover, we also note that the effect of removing POS and NER taggers is somewhat greater than KG. Upon further investigation, we observed that many of the KG entries are eventually automatically labeled in phase two of the framework by the pre-trained BERT. Moreover, we also look into the result of removing one set of labels entirely from the framework. We notice that with only one set of labels, our proposed framework can not utilize phase three effectively and, as a result, shows significantly poor performance. Moreover, we also note that the effect of removing the zero-shot learning labels is more notable quantitatively as compared to the labels from KG, POS/NER taggers. The significance of phase three (i.e., self-supervised co-training) is also highlighted by skipping this phase. In fact, by skipping this phase or not being able to use this phase properly, our proposed framework shows poor performance as compared to other unsupervised or weak supervision learning approaches. Finally, we also study the effect of high-confidence soft label selection in phase three. As expected, removing the high-confidence soft label selection mechanism in phase three significantly reduces the performance on both datasets. To summarize, SCot can still outperform other models that do not use labeled data even after removing these individual components: KG labels, POS/NER taggers, and soft labels.

Adib Mosharrof, Moghis Fereidouni, and A.B. Siddique

Table 6: Slot F1 score for different percentages of out-of-domain training data for zero-shot learning model.

Dataset →	SGD			MultiWoZ		
OOD Seen % →	25%	50%	75%	25%	50%	75%
Coach	0.5888	0.6419	0.6725	0.4408	0.4505	0.6522
LEONA	0.7181	0.7925	0.8324	0.5248	0.5533	0.8581
SCot (this work)	0.8592	0.9324	0.9512	0.7937	0.8282	0.9472

Effect of increasing the out-of-domain data for zero-shot slot filling model. The zero-shot learning models have been shown to improve their performance on new unseen (target) domains, when they are provided with more (out-of-domain) data [27, 43]. We also investigated whether increasing out-of-domain training data for the zero-shot slot filling model improves the performance of our model. Table 6 presents the F1 score of our proposed framework for both datasets when we vary the percentage of out-of-domain data to train the zero-shot slot filling model. It can be observed from the results that increasing the out-of-domain training data for the zero-shot learning model improves its prediction accuracy for the target domains, which eventually improves the performance of SCot. Following [43], we randomly selected domains for out-of-domain training of the zero-shot model and reported average results of five runs.

6 RELATED WORK

Supervised Slot Filling. Supervised approaches for the slot filling task have been extensively studied. Recurrent neural networks have been leveraged to learn the temporal interactions of the tokens within a sentence using Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) networks and detect the spans of the text associated with certain slots [23, 30]. Moreover, deep learning-based approaches have employed Conditional Random Fields (CRFs) to learn global interactions along with LSTMs and GRUs [19, 37]. The self-attention technique has also been used for token classification tasks, such as slot filling in the supervised setting [41, 47]. Moreover, formulations exist to jointly optimize the tasks of slot filling and intent detection in the supervised setting [15, 16, 25, 50, 54]. The capsule neural network was trained to jointly model the tasks of slot filling and intent detection in [54], pre-trained BERT was fine-tuned in [7], and both tasks were formulated as sequence-to-sequence labeling task in [11]. Although supervised approaches have proven effective for the task of slot filling, they require labeled training data for every new domain as well as slot, which prohibits their scalability. In this work, we focus on the open-domain slot filling task and propose a zero-label slot filling framework that can automatically generate labels with no human intervention.

Few-shot and Zero-shot Slot Filling. To minimize the human labor of annotating data for each new slot, zero-shot learning along with few-shot learning methods have shown promising results for the slot filling task. Few-shot learning methods rely on a small set of in-domain labeled data to nudge the model for accurate predictions. Recently, employing regular expressions [28], the prototypical network [13], and extensions to CRFs [18] have been proposed for the few-shot slot filling task. Furthermore, several works [4, 22] have leveraged model agnostic meta-learning to jointly model the slot filling and intent classification tasks. In a similar line of work, in the zero-shot setting, no in-domain labeled training data is required, which has shown encouraging results to eliminate the need for manual labeling. Using LSTM-based models along with the natural language descriptions of the slots has been shown to overcome the challenge of different names of the slot for apparently similar slots (e.g., `city` vs `destination` in `Travel` domain) for the zero-shot adaptation of slot filling task [1]. Similarly, attention mechanisms have further improved the performance [24, 32]. The concatenation of character and word level embeddings have been employed along with bidirectional LSTMs [39]. Moreover, the coarse-to-fine approach in [27, 43] was proposed to improve the robustness of the zero-shot slot filling task. In spite of promising results, there is a huge performance gap between zero-shot learning methods and supervised learning approaches that limit their applicability. In this work, we employ a zero-shot slot filling model to automatically generate a set of labels in the target domain and further improve the performance of the slot filling task in an open-domain setting.

Unsupervised and Weak Supervision Slot Filling. The other line of work to eliminate the need for manual labeling is weak supervision and unsupervised learning. Unsupervised learning methods [46] have used mining to build latent variable models [31]. Moreover, a coarse-to-fine approach using clustering has been explored in [53]. Furthermore, extensions to the DBSCAN [6], automatic induction of slot values using frame semantics theory [9], and autoencoders along with hierarchical clustering [42] have been proposed in the unsupervised learning setup. The weak supervision approaches generally employ a heuristic function [12, 17, 38] to automatically generate labels (i.e., no human intervention) and overcome the challenge of modeling on the noisy labels by employing methods, such as distant-LSTM-CRF [14] and dictionaries [40]. Recently, weak supervision has also been leveraged using clustering algorithms for the automatic selection of candidate values [20]. Furthermore, random walk inference algorithms were used in [8] to automatically generate supervision data. It is important to recall that weak supervision and unsupervised learning models show inferior performance as compared to supervised learning models. In this work, we employ a GPT-2-based heuristic function to automatically generate one set of labels using KG and other pre-trained NLP models and our proposed framework SCot combines this set of labels with the labels generated through the zero-shot model by introducing a self-supervised co-training mechanism.

7 CONCLUSION

We have presented a self-supervised co-training slot filling framework that automatically generates two sets of complementary pseudo labels for co-training two peers in a self-supervised fashion, progressively reduces the amount of noise in the labels by automatically selecting high-confidence soft labels, and improves their prediction power. In contrast to zero-shot learning and weak supervision approaches that produce inferior quality results as compared to supervised learning approaches, the proposed framework SCot achieves competitive performance to supervised slot filling models and shows the promise of open-domain slot filling without any human labeling. Our extensive evaluations using SGD and MultiWoZ datasets that contain dialogs across 20 domains and cover 240 slots, demonstrate that SCot outperforms state-of-the-art few-shot GPT-3 by 44.18% and 40.40% on average for SGD and MultiWoZ datasets, respectively.

REFERENCES

[1] Ankur Bapna, Gokhan Tür, Dilek Hakkani-Tür, and Larry Heck. 2017. Towards Zero-Shot Frame Semantic Parsing for Domain Scaling. In *Proc. Interspeech 2017*. 2476–2480. https://doi.org/10.21437/Interspeech.2017-518

[2] Jerome R Bellegarda. 2014. Spoken language understanding for natural interaction: The siri experience. In *Natural interaction with robots, knowbots and smartphones*. Springer, 3–14.

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research* 3 (2003), 1137–1155.

[4] Hemanthage S Bhathiya and Uthayasanker Thayasivam. 2020. Meta Learning for Few-Shot Joint Intent Detection and Slot-Filling. In *Proceedings of the 2020 5th International Conference on Machine Learning Technologies*. 86–92.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[6] Ajay Chatterjee and Shubhashis Sengupta. 2020. Intent mining from past conversations for conversational agent. *arXiv preprint arXiv:2005.11014* (2020).

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

[8] Yun-Nung Chen, William Yang Wang, and Alexander Rudnicky. 2015. Jointly modeling inter-slot relations by random walk on knowledge graphs for unsupervised spoken language understanding. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 619–629.

[9] Yun-Nung Chen, William Yang Wang, and Alexander I Rudnicky. 2013. Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 120–125.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[11] Yue Feng, Yang Wang, and Hang Li. 2020. A Sequence-to-Sequence Approach to Dialogue State Tracking. *arXiv preprint arXiv:2011.09553* (2020).

[12] Jason Fries, Sen Wu, Alex Ratner, and Christopher Ré. 2017. Swellshark: A generative model for biomedical named entity recognition without labeled data. *arXiv preprint arXiv:1704.06360* (2017).

[13] Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in named entity recognition task. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. 993–1000.

[14] Athanasios Giannakopoulos, Claudiu Musat, Andreea Hossmann, and Michael Baeriswyl. 2017. Unsupervised Aspect Term Extraction with B-LSTM & CRF using Automatically Labelled Datasets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. 180–188.

[15] 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)*. 753–757.

[16] Dilek Hakkani-Tür, Gökhan Tür, 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 *Interspeech*. 715–719.

[17] Wenqi He. 2017. Autoentity: automated entity detection from massive text corpora. (2017).

[18] Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot Slot Tagging with Collapsed Dependency Transfer and Label-enhanced Task-adaptive Projection Network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 1381–1393.

[19] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).

[20] Vojtěch Hudeček, Ondřej Dušek, and Zhou Yu. 2021. Discovering dialogue slots with weak supervision. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2430–2442.

[21] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[22] Jason Krone, AI Amazon, Yi Zhang, and Mona Diab. 2020. Learning to Classify Intents and Slot Labels Given a Handful of Examples. *ACL 2020* (2020), 96.

[23] Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu. 2016. Leveraging Sentence-level Information with Encoder LSTM for Semantic Slot Filling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 2077–2083. https://doi.org/10.18653/v1/D16-1223

[24] Sungjin Lee and Rahul Jha. 2019. Zero-shot adaptive transfer for conversational language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6642–6649.

[25] Bing Liu and Ian Lane. 2016. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. In *Interspeech 2016*. 685–689. https://doi.org/10.21437/Interspeech.2016-1352

[26] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the Variance of the Adaptive Learning Rate and Beyond. In *ICLR*.

[27] Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. 2020. Coach: A Coarse-to-Fine Approach for Cross-domain Slot Filling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 19–25. https://doi.org/10.18653/v1/2020.acl-main.3

[28] Bingfeng Luo, Yansong Feng, Zheng Wang, Songfang Huang, Rui Yan, and Dongyan Zhao. 2018. Marrying up Regular Expressions with Neural Networks: A Case Study for Spoken Language Understanding. In *56th Annual Meeting of the Association for Computational Linguistics*. 2083–2093.

[29] Muhammad Hasan Maqbool, Luxun Xu, AB Siddique, Niloofar Montazeri, Vagelis Hristidis, and Hassan Foroosh. 2022. Zero-label Anaphora Resolution for Off-Script User Queries in Goal-Oriented Dialog Systems. In *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*. IEEE, 217–224.

[30] Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2014. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23, 3 (2014), 530–539.

[31] Qingkai Min, Libo Qin, Zhiyang Teng, Xiao Liu, and Yue Zhang. 2020. Dialogue state induction using neural latent variable models. *arXiv preprint arXiv:2008.05466* (2020).

[32] Subhabrata Mukherjee and Ahmed Awadallah. 2020. Uncertainty-aware self-training for few-shot text classification. *Advances in Neural Information Processing Systems* 33 (2020), 21199–21212.

[33] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066* (2019).

[34] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[35] Lance Ramshaw and Mitch Marcus. 1995. Text Chunking using Transformation-Based Learning. In *Third Workshop on Very Large Corpora*. https://www.aclweb.org/anthology/W95-0107

[36] Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards Scalable Multi-Domain Conversational Agents: The Schema-Guided Dialogue Dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8689–8696.

[37] Nils Reimers and Iryna Gurevych. 2017. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799* (2017).

[38] Xiang Ren, Ahmed El-Kishky, Chi Wang, Fangbo Tao, Clare R Voss, and Jiawei Han. 2015. Clustype: Effective entity recognition and typing by relation phrase-based clustering. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 995–1004.

[39] Darsh Shah, Raghav Gupta, Amir Fayazi, and Dilek Hakkani-Tur. 2019. Robust Zero-Shot Cross-Domain Slot Filling with Example Values. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 5484–5490. https://doi.org/10.18653/v1/P19-1547

[40] Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. Learning Named Entity Tagger using Domain-Specific Dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2054–2064.

[41] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[42] Chen Shi, Qi Chen, Lei Sha, Sujian Li, Xu Sun, Houfeng Wang, and Lintao Zhang. 2018. Auto-dialabel: Labeling dialogue data with unsupervised learning. In *Proceedings of the 2018 conference on empirical methods in natural language processing*. 684–689.

[43] AB Siddique, Fuad Jamour, and Vagelis Hristidis. 2021. Linguistically-enriched and context-awarezero-shot slot filling. In *Proceedings of the Web Conference 2021*. 3279–3290.

[44] A.B. Siddique, Fuad Jamour, Luxun Xu, and Vagelis Hristidis. 2021. Generalized Zero-shot Intent Detection via Commonsense Knowledge. *arXiv preprint arXiv:2102.02925* (2021). arXiv:2102.02925

[45] AB Siddique, MH Maqbool, Kshitija Taywade, and Hassan Foroosh. 2022. Personalizing Task-oriented Dialog Systems via Zero-shot Generalizable Reward Function. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 1787–1797.

[46] Muhammad Abu Bakar Siddique. 2021. *Unsupervised and Zero-Shot Learning for Open-Domain Natural Language Processing*. University of California, Riverside.

[47] Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep semantic role labeling with self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[49] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *ICML*. 478–487.

[50] Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *2013 ieee workshop on automatic speech recognition and understanding*. IEEE, 78–83.

[51] Steve Young. 2002. Talking to machines (statistically speaking). In *Seventh International Conference on Spoken Language Processing*.

[52] Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A Dialogue Dataset with Additional Annotation Corrections and State Tracking Baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*. Association for Computational Linguistics, Online, 109–117. https://doi.org/10.18653/v1/2020.nlp4convai-1.13

[53] Zengfeng Zeng, Dan Ma, Haiqin Yang, Zhen Gou, and Jianping Shen. 2021. Automatic intent-slot induction for dialogue systems. In *Proceedings of the Web Conference 2021*. 2578–2589.

[54] Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip Yu. 2019. Joint Slot Filling and Intent Detection via Capsule Neural Networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 5259–5267. https://doi.org/10.18653/v1/P19-1519

[55] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*. 19–27.