Slot Induction via Pre-trained Language Model Probing and Multi-level Contrastive Learning

Hoang H. Nguyen[1], **Chenwei Zhang**[2], **Ye Liu**[3], **Philip S. Yu**[1]
[1] Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA
[2] Amazon, Seattle, WA, USA
[3] Salesforce Research, Palo Alto, CA, USA
{hnguy7,psyu}@uic.edu, cwzhang@amazon.com, yeliu@salesforce.com

Abstract

Recent advanced methods in Natural Language Understanding for Task-oriented Dialogue (TOD) Systems (e.g., intent detection and slot filling) require a large amount of annotated data to achieve competitive performance. In reality, token-level annotations (slot labels) are time-consuming and difficult to acquire. In this work, we study the Slot Induction (SI) task whose objective is to induce slot boundaries without explicit knowledge of token-level slot annotations. We propose leveraging Unsupervised Pre-trained Language Model (PLM) Probing and Contrastive Learning mechanism to exploit (1) unsupervised semantic knowledge extracted from PLM, and (2) additional sentence-level intent label signals available from TOD. Our approach is shown to be effective in SI task and capable of bridging the gaps with token-level supervised models on two NLU benchmark datasets. When generalized to emerging intents, our SI objectives also provide enhanced slot label representations, leading to improved performance on the Slot Filling tasks. [1]

1 Introduction

Natural Language Understanding (NLU) has become a crucial component of the Task-oriented Dialogue (TOD) Systems. The goal of NLU is to extract and capture semantics from users' utterances [2]. There are two major tasks in NLU framework, including intent detection (ID) and slot filling (SF) (Tur and De Mori, 2011). While the former focuses on identifying overall users' intents, the latter extracts semantic concepts from natural language sentences. In NLU tasks, intents denote sentence-level annotations while slot types represent token-level labels.

Despite recent advances, state-of-the-art NLU methods (Haihong et al., 2019; Goo et al., 2018) require a large amount of annotated data to achieve competitive performance. However, the fact that annotations, especially token-level labels, are expensive and time-consuming to acquire severely inhibits the generalization capability of traditional NLU models in an open-world setting (Louvan and Magnini, 2020; Xia et al., 2020). Recent works attempt at tackling the problems in low-resource settings on both intent level (Xia et al., 2018; Nguyen et al., 2020; Siddique et al., 2021) and slot level (Yu et al., 2021; Glass et al., 2021). However, most approaches remain restricted to closed-world settings where there exist pre-defined sets of seen and emerging sets of classes. Some approaches even require additional knowledge from related token-level tasks that might not be readily available.

Additionally, with increasing exposure to the ever-growing number of intents and slots, TOD systems are expected to acquire task-oriented adaptation capability by leveraging both inherent semantic language understanding and task-specific knowledge to identify the crucial emerging concepts in the users' utterances. This ability can be referred to as **Slot Induction** in TOD Systems.

Recently, Pre-trained Contextualized Language Models (PLM) such as BERT (Devlin et al., 2019) have shown promising capability of capturing semantic and syntactic structure without explicit linguistic pre-training objectives (Jawahar et al., 2019; Rogers et al., 2020; Wu et al., 2020b). Despite imperfections, the captured semantics from PLM via unsupervised probing mechanisms could be leveraged to induce important semantic phrases covering token-level slot labels.

Additionally, as an effective unsupervised representation learning mechanism (Wei and Zou, 2019; Gao et al., 2021), Contrastive Learning (CL) is capable of refining the imperfect PLM semantic phrases in a self-supervised manner to mitigate biases existent in the PLM. In specific, given a sample phrase *in the same area* corresponding to

[1] Our code and datasets are publicly available at https://github.com/nhhoang96/MultiCL_Slot_Induction

[2] In our work, we use the term **utterance** and **sentence** interchangeably.

Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue, pages 470–481
September 11–15, 2023.

spatial_relation slot type, as a presumed structural knowledge, PLM tends to split the preposition and determiner from the noun phrase during segmentation, resulting in *in the* and *same area*. Despite its structural correctness, the identified segments fail to align with ground truth slots due to the lack of knowledge from the overall utterance semantics.

On the other hand, CL can also be leveraged on a sentence level when intent labels are available. In fact, there exist strong connections between slot and intent labels (Zhang et al., 2019; Wu et al., 2020a). For instance, utterances with *book_restaurant* intent tend to contain *location* slots than those from *rate_book* intent. Therefore, as intent labels are less expensive to acquire, they could provide additional signals for CL to induce slot labels more effectively when available.

In this work, we propose leveraging PLM probing together with CL objectives for Slot Induction (SI) task. Despite imperfections, PLM-derived segmentations could produce substantial guidance for SI when slot labels are not readily available. We introduce CL to further refine PLM segmentations via (1) segment-level supervision from unsupervised PLM itself, and (2) sentence-level supervision from intent labels to exploit the semantic connections between slots and intents. Our refined BERT from SI objectives can produce effective slot representations, leading to improved performance in slot-related tasks when generalized towards emerging intents.

Our contributions can be summarized as follows:

• We propose leveraging semantic segments derived from Unsupervised PLM Probing (UPL) to induce phrases covering token-level slot labels. We name the task as Slot Induction.

• We propose enhancing the quality of PLM segments with Contrastive Learning refinement to better exploit (1) unsupervised segment-level signals from PLM, (2) sentence-level signals from intent labels to improve SI performance.

• We showcase the effectiveness of our proposed SI framework and its ability to produce refined PLM representations for token-level slots when generalized to emerging intents.

2 Related Work

Pre-trained Language Model Probing Pre-trained Language Models (PLMs) have been shown to possess inherent syntactic and semantic information. Different probing techniques are developed

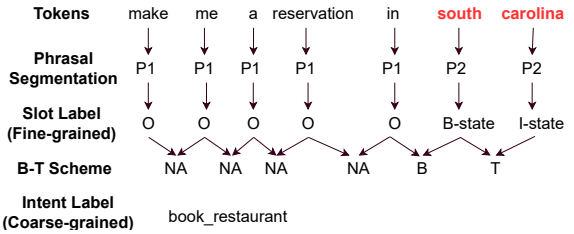


Figure 1: Illustration of connections between Phrasal Segmentation (PS), Beginning-Inside-Outside (BIO) Tagging Slot Label and Break-Tie (B-T) Labeling Schema based on Golden Slot Labels (Red: denotes Golden Slot Labels for the utterance, **P1,P2** denote identified phrases, **NA, B,T** denote Not-Relevant, Break, Tie Labels in B-T Labeling Scheme)

to investigate the knowledge acquired by PLMs, either from output representations (Wu et al., 2020b), intermediate representations (Sun et al., 2019), or attention mapping (Clark et al., 2019; Yu et al., 2022). Unlike previous probing techniques that focus on deriving syntactic tree structure, we leverage semantically coherent segments recognized by PLMs to induce phrases containing token-level slot labels in NLU tasks for TOD Systems.

Contrastive Learning Contrastive Learning (CL) has been widely leveraged as an effective representation learning mechanism (Oord et al., 2018). The goal of CL is to learn the discriminative features of instances via different augmentation methods. In Natural Language Processing (NLP), CL has been adopted in various contexts ranging from text classification (Wei and Zou, 2019), embedding representation learning (Gao et al., 2021) to question answering (Xiong et al., 2020; Liu et al., 2021). CL has also been integrated with PLM as a more effective fine-tuning strategy for downstream tasks (Su et al., 2021). In our work, we propose an integration of CL with PLM probing techniques to further refine imperfect PLM-derived segments via (1) unsupervised signals from PLM itself, and (2) less expensive sentence-level intent label supervision for improved SI performance.

3 Problem Formulation

Slot Induction We introduce the task of Slot Induction (SI) whose objective is to identify phrases containing token-level slot labels. Unlike traditional SF and previously proposed AISI framework (Zeng et al., 2021), in our SI task, both slot boundaries and slot types are unknown during training. The task is also related to Phrasal Segmentation/ Tagging (PS) methods (Shang et al., 2018a; Gu et al., 2021). However, there are three key distinc-

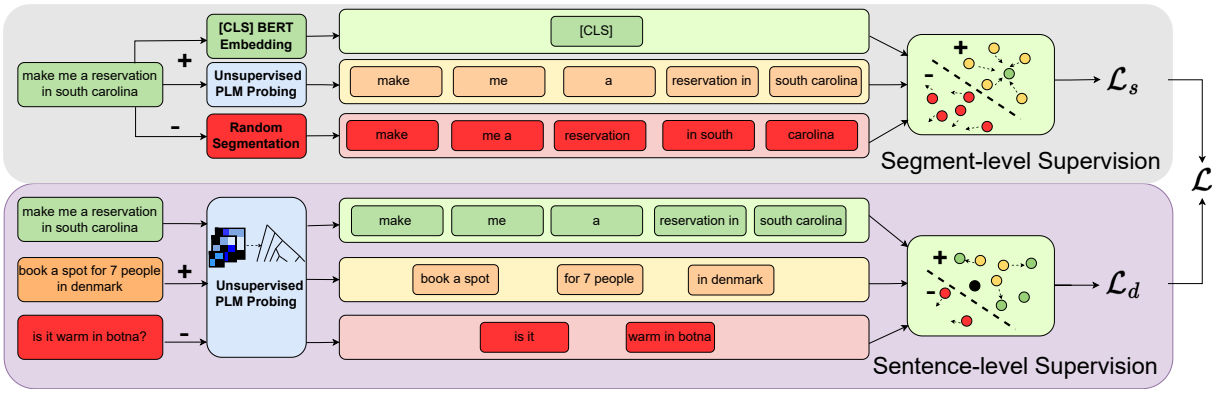


Figure 2: Illustration of the Proposed Model Overview. The model is made up of two-level Contrastive Learning depicted by two modules: (1) **Segment-level Supervision (SegCL)** via Unsupervised PLM Probing (UPL), (2) **Sentence-level Supervision (SentCL)** via intent labels. Green, Orange, Red denote Anchor, Positive, Negative samples respectively. **Black circle** denotes the representation of the **cropped segment** from Augmentation.

tions: (1) utterances and intent labels (if available) are the only sources of information for the task, (2) slot phrases (i.e. close by (*spatial_relation*), most expensive (*cost_relative*)), are not restricted to noun phrases, (3) slot phrases (i.e. strauss is playing today (*movie_name*)) might be more sophisticated and harder to identify than typical noun phrases (i.e. chicago (*city*)). These differences explain why PS methods do not consistently perform well in our proposed SI task (Section 6).

Specifically, given an utterance with the length of T tokens $x = [x_1, x_2..., x_T]$, SI task aims to make decisions at $T-1$ positions whether to (1) tie the current token with the previous one to extend the current phrase [3], or (2) break away from the previous token/ phrase to form a new phrase.

Evaluation Metric We adopt the Break-Tie (B-T) schema (Shang et al., 2018b) to evaluate SI task. The metric allows for direct comparison between supervised Sequential Labeling and unsupervised PS methods. In SI setting, *Tie* represents the connection between tokens of the same slot type while *Break* denotes the separation between (1) tokens from different slot types, and (2) tokens from a slot type and non-slot tokens. As the objective of SI is on slot tokens, consecutive non-slot tokens should not contribute to the overall performance. Therefore, additional *NA* labels are introduced to guarantee that evaluations are only conducted on slot tokens and their adjacent tokens.

Figure 1 depicts the connections of SF and PS labels with B-T schema. For PS, Break denotes the separation of two consecutive phrases. If no phrase is identified by PS methods, every token is considered as *Tie* to one another. In the Figure 1 example, as "south carolina" is the only identified phrase, the given sentence is simply split into two phrases where *Break* denotes their junction. Precision, Recall and F-1 Metrics are reported for individual labels, namely B-P,B-R,B-F1 for *Break* and T-P,T-R,T-F1 for *Tie*.

[3] In our work, we use the term **segment** and **phrase** interchangeably.

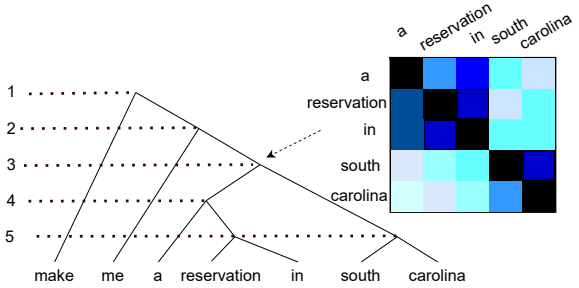


Figure 3: Illustration of UPL Segmentation Tree for sentence *"make me a reservation in south carolina"* with sample Impact Matrix at depth $d = 3$ (**Lighter** color denotes **lower** impact score). $d = 0$ corresponds to the sentence-level representation (no segmentation).

Given an utterance, an optimal SI model makes correct decisions to either break and tie at every token index. Therefore, **H-Mean**, denoting the harmonic mean between F-1 Scores of *Tie* and *Break* label predictions, is considered the golden criteria for SI model comparison.

4 Proposed Framework

In this section, we introduce our proposed Multi-level Contrastive Learning framework for SI task with 2 major components: **Segment-level Contrastive Learning (SegCL)** and **Sentence-level Contrastive Learning (SentCL)** as depicted in Figure 2. We first introduce the backbone Unsupervised PLM Probing (UPL) for both components.

4.1 Unsupervised PLM Probing (UPL)

We adopt Token-level Perturbed Masking mechanism (Wu et al., 2020b) to construct semantic segments by leveraging PLM in an unsupervised manner. Due to its operations on the output layers of PLM, UPL is flexible with the choices of PLM and avoids local sub-optimal structure from pre-selected PLM layers (Clark et al., 2019). In our study, we use BERT (Devlin et al., 2019) as an exemplar PLM. Specifically, given a sentence $x = [x_1, \cdots, x_T]$, the Impact Matrix $\mathcal{F} \in \mathbb{R}^{T \times T}$ is constructed by calculating the Impact Score between every possible pair of tokens (including with itself) in the given sentence based on BERT's embedding and a specified distance metric (Wu et al., 2020b). Leveraging $\mathcal{F}$, UPL derives the structural tree by recursively finding the optimal cut position k with the following objective:

$$\begin{aligned} \underset{k}{argmax}(&\mathcal{F}^{i..k}_{i..k} + \mathcal{F}^{k+1..j}_{k+1..j} \\ &-\mathcal{F}^{k+1..j}_{i..k} - \mathcal{F}^{i..k}_{k+1..j}) \end{aligned} \tag{1}$$

where $i, j \in [0, T-1]$ denotes the start and end indexes of the segment considered for splitting.

At every tree depth, sets of combined tokens are considered semantic segments since they preserve certain meanings within utterances. Segments at a deeper level include (1) all segments obtained from previous levels and (2) new segments obtained at the current level. For instance, at depth $d = 3$ of the given example in Figure 3, the obtained segments are *"make"*, *"me"*, *"a reservation in"*, *"south carolina"*. As PLM parameters are updated during training, the derived UPL trees from the same utterance can vastly change. For simplicity, we set the tree depth d as a tunable hyperparameter.

Formally, at a specified depth d with m semantic segments acquired from UPL, the final representation of the input sentence x is defined as follows:

$$\mathbf{h_U} = [\overrightarrow{s_0}, ...\overrightarrow{s_{m-1}}], \ \overrightarrow{s_i} = \frac{\sum_{j=c}^{d} \overrightarrow{h_j}}{d - c + 1} \tag{2}$$

where $\mathbf{h_U} \in \mathbb{R}^{\mathbf{m} \times \mathbf{d_h}}$, d_h is hidden dimensions of BERT representations, c,d are the start and end indexes of the corresponding segment s_i and $\overrightarrow{h_j}$ represents the BERT embedding of j-th token.

4.2 Multi-level Contrastive Learning

As UPL only considers token interactions for segment formation, its semantic segments are far from perfect. Additional refinements are needed to enhance the quality of the extracted segments via (1) semantic signals captured in segment-level PLM representations, (2) sentence-level intent labels.

Our overall learning objective is summarized as $\mathcal{L} = \delta\mathcal{L}_s + \gamma\mathcal{L}_d$, where $\mathcal{L}_s, \mathcal{L}_d$ denote SegCL Loss and SentCL Loss, and γ, δ are their corresponding loss coefficient hyperparameters for aggregation. For each CL level, positive and negative samples are drawn separately based on (1) the same batch of sampled anchor samples, (2) different selection criteria detailed below.

Segment-level Contrastive Learning (SegCL) UPL produces semantic segments by purely considering the exhaustive word-pair interactions within given sentences. However, it does not take into consideration the overall semantic representation produced by the PLM BERT via special [CLS] tokens. Therefore, we propose leveraging [CLS] representations to guide UPL towards more discriminative segment representations via SegCL objectives. Specifically, SegCL aims to minimize the distance between [CLS] representation and UPL segment representations while maximizing the distance between representations of [CLS] and random segments of the corresponding utterance.

Given a sample utterance, segment representation obtained from UPL is considered a positive sample while negative samples are represented as segments produced by randomly chosen indexes within the given utterance. The number of segments for both positive and negative samples are kept similar (m) so that SegCL focuses on learning the optimal locations of segmentation indexes. We adopt InfoNCE contrastive loss (Oord et al., 2018):

$$\mathcal{L}_s = -log \frac{\exp^{cos(\overrightarrow{h_C}, \mathbf{h_U})/\tau_\mathbf{s}}}{\exp^{cos(\overrightarrow{h_C}, \mathbf{h_U})/\tau_\mathbf{s}} + \exp^{cos(\overrightarrow{h_C}, \mathbf{h_r})/\tau_\mathbf{s}}} \tag{3}$$

where $\overrightarrow{h_C} \in \mathbb{R}^{1 \times d_h}$ denotes [CLS] representation from BERT, and $\mathbf{h_U}, \mathbf{h_r} \in \mathbb{R}^{\mathbf{m} \times \mathbf{d_h}}$ denote the representations from UPL and random segmentation. m is the number of extracted segments from UPL as defined in Equation 2. τ_s is the soft segment-level temperature hyperparameter.

Sentence-level Contrastive Learning (SentCL) Besides relying on UPL, we propose leveraging sentence-level intent labels to further improve the quality of segment representations derived from UPL. Specifically, we randomly draw positive and negative samples based on the intent labels of the given anchor samples. As utterances with similar intents tend to share common slot phrases, our SentCL aims to learn discriminative segments for better alignment between utterances from the same

intents. We adopt InfoNCE loss for SentCL:

$$\mathcal{L}_d = -log\frac{\exp^{cos(\mathbf{h_a},\mathbf{h_+})/\tau_\mathbf{d}}}{\exp^{cos(\mathbf{h_a},\mathbf{h_+})/\tau_\mathbf{d}} + \exp^{cos(\mathbf{h_a},\mathbf{h_-})/\tau_\mathbf{d}}} \quad (4)$$

where $\mathbf{h_a} \in \mathbb{R}^{\mathbf{m}\times\mathbf{d_h}}, \mathbf{h_+} \in \mathbb{R}^{\mathbf{a}\times\mathbf{d_h}}, \mathbf{h_-} \in \mathbb{R}^{\mathbf{b}\times\mathbf{d_h}}$ denote the representations of anchor, positive and negative samples respectively and m, a, b denote the number of extracted segments from UPL for the respective samples. τ_d is the soft sentence-level temperature hyperparameter.

To further encourage the model to identify discriminative segments from the same sentence-level intent label, we adopt random segment cropping as an augmentation strategy. As UPL could generate a vastly different number of segmentation based on the the cut_score (Equation 1) from the updated BERT parameters at each step, we conduct random segmentation cropping by a percent ratio (β) so that it could be adapted to individual input utterances and segmentation trees. The remaining segments after cropping are utilized to compute $\mathcal{L}_d$.

5 Experiments

5.1 Datasets & Evaluation Tasks

We evaluate our proposed work on the two publicly available NLU benchmark datasets ATIS (Tur et al., 2010) and SNIPS (Coucke et al., 2018) with the previously proposed data splits (Zhang et al., 2019).

To evaluate the generalization of the refined representations from our proposed work, we conduct additional splits of each dataset into 2 parts (P1 and P2). For each benchmark dataset, we construct P1 for SI evaluation by reserving samples from randomly chosen 60% of available intents. The remaining samples (P2) are used as test sets for evaluating SF task when generalized towards emerging intents. The objective of this splitting strategy is two-fold: (1) Since there is no overlapping intent between P1 and P2, there exists no information leakage of intents leveraged in SI training (P1) while evaluating SF (P2). (2) We can validate the generalization capability of representations learned from our SI framework in other slot-related tasks. Statistics for both parts of each dataset are reported in Table 1.

Evaluation Task 1: Slot Induction (P1) We conduct evaluation of Unsupervised SI task on P1 of both SNIPS and ATIS datasets. B-T evaluation metrics are adopted as introduced in Section 3. Implementation details of our SI model, including hyperparameters, are discussed in Appendix B.

Table 1: Details of SNIPS and ATIS datasets.

	SNIPS_P1	SNIPS_P2	ATIS_P1	ATIS_P2
# Intents	5	2	14	7
# Slots	31	16	68	63
# Train Samples	9356	–	3811	–
# Validation Samples	500	–	414	–
# Test Samples	501	4127	750	895
Avg Train Sent Length	8.65	–	11.67	–
Avg Valid Sent Length	8.72	–	11.82	–
Avg Test Sent Length	8.71	9.87	10.68	8.92

Evaluation Task 2: Generalization towards Emerging Intents (P2) To evaluate the generalization of SI refinement, we conduct SF training on P1 datasets with different BERT initializations (Original vs Refined BERT) and evaluation on emerging intents and slots in P2. Slot Precision (S-P), Recall (S-R), F1 (S-F1) are reported on P2. Implementation is detailed in Appendix C.

5.2 Slot Induction Baseline

We conduct a comprehensive study that evaluates our SI approach with both *Upper Bound* and *Comparable* Methods. For fair comparisons, we leverage the same "bert-base-uncased" PLM (Devlin et al., 2019) across all applicable baselines. The *Upper Bound* includes methods that leverage directly **token-level labels** such as Golden Slot Labels, Named Entity Recognition (NER) Labels, Part-of-Speech (POS) Tagging or Noun Phrase (NP) Labels during training and/or pre-training process, including **Joint BERT FT**, **SpaCy** (Honnibal et al., 2020), **FlairNLP** (Akbik et al., 2018).

In addition, we compare with other **unsupervised** PS methods that do not require any token-level labels as *Comparable* Baselines, including: **Dependency Parsing (DP-RB/DP-LB)**, **AutoPhrase** (Shang et al., 2018a), **UCPhrase** (Gu et al., 2021), **USSI** (Yu et al., 2022). For fair comparisons with *Comparable* baselines, we also report results from our model's variants with similar prior knowledge assumption, namely **Ours (w/o CL)**, **Ours (w/o SentCL)** . Due to space constraints, details of *Upper Bound* and *Comparable* baselines are provided in Appendix A.1, A.2 respectively.

6 Result & Discussion

6.1 Slot Induction

From our experimental results in Table 2 and 3, for SI task, our proposed framework outperforms the *Comparable* Methods in H-Mean evaluation metric for B-T schema on both datasets. We achieve significant gains in SNIPS dataset (+6.28 points in H-Mean as compared to the next *Comparable* Methods). Despite lack of access to any types of token-level labels, our method is also closely on

Table 2: Experimental performance result on SNIPS dataset over 3 runs (**H-Mean** is considered the golden criteria for SI (Section 3)). ¶ denotes models that do not require random initializations.

	Model	Prior Knowledge	Break			Tie			H-Mean
			B-P	B-R	B-F1	T-P	T-R	T-F1	
Upper Bound	Joint BERT FT	Slot + Intent	96.91 ± 0.17	96.62 ± 0.69	96.76 ± 0.26	73.55 ± 0.38	73.39 ± 1.03	73.47 ± 0.38	83.52 ± 0.16
	FlairNLP ¶	POS & NER	80.04	62.81	70.38	48.25	63.31	54.77	61.60
	SpaCy ¶	POS	75.73	50.29	60.45	41.71	62.97	50.18	54.84
Comparable	DP-LB ¶	–	59.68	34.27	43.54	21.69	38.53	27.76	33.90
	DP-RB ¶	–	66.53	52.56	58.73	33.97	52.24	41.17	48.40
	AutoPhrase	External KB	65.51 ± 0.23	57.16 ± 2.59	61.05 ± 1.15	33.39 ± 0.74	36.62 ± 1.67	34.93 ± 1.50	44.43 ± 1.64
	UCPhrase	PLM	42.25 ± 4.90	20.26 ± 2.71	27.39 ± 1.95	36.06 ± 2.42	**73.53 ± 3.33**	**48.39 ± 2.91**	34.98 ± 2.35
	USSI ¶	PLM	**83.21**	62.12	71.14	33.96	49.93	40.42	51.55
	Ours (w/o CL) ¶	PLM	75.36	66.70	70.76	38.51	45.81	41.84	52.59
	Ours (w/o SentCL)	PLM	76.09 ± 0.73	66.43 ± 0.29	70.94 ± 0.49	39.15 ± 0.60	47.9 ± 0.91	43.09 ± 0.73	53.61 ± 0.71
	Ours (full)	**PLM + Intent**	76.87 ± 0.25	**67.77 ± 0.26**	**72.00 ± 0.24**	**40.39 ± 0.16**	48.49 ± 0.19	44.07 ± 0.04	**54.68 ± 0.08**

Table 3: Experimental performance result on ATIS dataset over 3 runs (**H-Mean** is considered the golden criteria for SI (Section 3)). ¶ denotes models that do not require random initializations.

	Model	Prior Knowledge	Break			Tie			H-Mean
			B-P	B-R	B-F1	T-P	T-R	T-F1	
Upper Bound	Joint BERT FT	Slot + Intent	98.49 ± 0.24	99.33 ± 0.08	98.91 ± 0.09	59.07 ± 0.36	58.27 ± 0.89	58.67 ± 0.63	73.65 ± 0.54
	FlairNLP ¶	POS & NER	95.44	77.90	85.78	41.34	61.91	49.58	62.84
	SpaCy ¶	POS	94.45	69.64	80.17	35.33	61.17	44.79	57.47
Comparable	DP-LB ¶	–	80.80	36.38	50.17	12.32	38.51	18.67	27.21
	DP-RB ¶	–	84.24	**66.84**	**74.54**	14.81	30.52	19.94	31.46
	AutoPhrase	External KB	75.96 ± 0.04	40.06 ± 0.28	52.46 ± 0.18	19.75 ± 0.21	49.33 ± 0.38	**28.20 ± 0.28**	36.68 ± 0.21
	UCPhrase	PLM	47.25 ± 0.04	17.27 ± 0.72	25.29 ± 0.78	17.36 ± 0.16	**58.21 ± 0.68**	26.75 ± 0.11	26.00 ± 0.47
	USSI ¶	PLM	**95.06**	56.36	70.77	14.78	45.22	22.28	33.89
	Ours (w/o CL) ¶	PLM	86.40	61.53	71.87	18.23	35.27	24.04	36.03
	Ours (w/o SentCL)	PLM	87.29 ± 0.15	64.21 ± 0.27	73.99 ± 0.13	20.09 ± 0.08	35.86 ± 0.35	25.75 ± 0.08	38.20 ± 0.08
	Ours (full)	**PLM + Intent**	87.80 ± 0.27	63.27 ± 0.67	73.54 ± 0.36	**20.53 ± 0.14**	37.89 ± 0.99	26.63 ± 0.26	**39.10 ± 0.24**

Table 4: Ablation study of effectiveness of SegCL and SentCL on SNIPS and ATIS in terms of H-Mean

	SNIPS	**ATIS**
Ours (w/o CL)	52.59	36.03
+ SegCL	53.61 ± 0.71	38.20 ± 0.08
+ SentCL (w/o aug)	53.44 ± 0.22	37.59 ± 0.81
+ SentCL (w aug)	54.23 ± 0.10	38.12 ± 0.36
Ours (full)	**54.68 ± 0.08**	**39.10 ± 0.24**

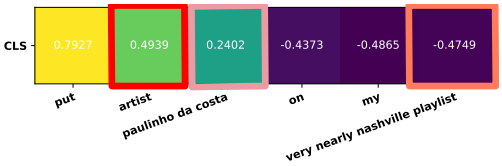
par with some of the *Upper Bound* methods that have been pre-trained with token-level labels (0.16 point difference from SpaCy in H-Mean). Despite promising achievements, most unsupervised PS methods only achieve competitive Break performance as compared to supervised methods but fall behind more significantly in terms of Tie performance. This implies unsupervised methods are able to differentiate non-slot tokens from slot tokens but tend to fragment slot tokens of the same type into multiple slot phrases due to the missing knowledge of token-level slot label spans.

UCPhrase is an exceptional baseline as it achieves significant better Tie but worse Break performance as compared to other *Comparable* baselines. This roots from the lack of keyphrases predicted from the model, leading to higher tendency to "tie" tokens. We speculate that its core phrase miner's dependency on frequency is not effective for extracting slots in NLU tasks. Phrases with high frequency in utterances are typically non-slot tokens (i.e. add, reserve), leading to limited meaningful core phrases for phrase-tagging training.

On ATIS dataset, the gap between *Comparable* Methods and *Upper Bound* is more significant as utterances tend to be longer and contain a wider variety of slot types than SNIPS dataset. This leads to a significant reduction in T-P across all of the *Comparable* Methods, resulting in a larger gap in H-Mean for ATIS dataset (approximately 18.37 points in comparison with 0.16 points in SNIPS dataset). Additionally, in comparison with SNIPS dataset, ATIS dataset contains more domain-independent slot types such as *city_name* (New York), *country_name* (United States). Therefore, methods leveraging either relevant token-level labels (i.e. POS, NER tags) or additional large-scaled external Knowledge Base (i.e. Wikipedia) achieve considerable performance gains. For instance, *FlairNLP* is only 10.81 points below the Fully Supervised *Joint BERT FT* on ATIS dataset (as compared to 21.92 points below on SNIPS) in terms of H-Mean.

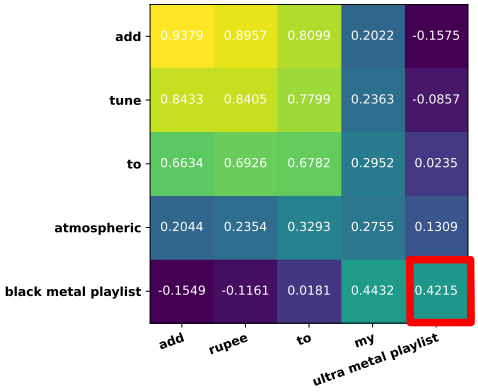
Compared with USSI, *Ours (w/o CL)* consistently achieves better H-Mean performance on both ATIS and SNIPS datasets (1.04% and 2.14% respectively). We hypothesize USSI might suffer from the local sub-optimality of pre-selected layers within deep PLM architecture. As the attention distribution across different layers varies (Clark et al., 2019), the pre-selected layers can significantly impact the unsupervised semantic probing of PLM.

Table 4 demonstrates that both SegCL and SentCL (w aug) objectives provide valuable in-

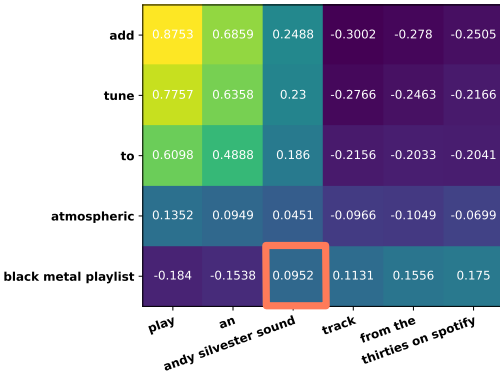


(a) Segment-level Supervised Positive-Anchor Pair

(b) Segment-level Supervised Negative-Anchor Pair



(c) Sentence-level Supervised Positive-Anchor Pair



(d) Sentence-level Supervised Negative-Anchor Pair

Figure 4: Similarity Matrices between positive/negative and anchor samples from SegCL and SentCL. For SegCL ((a), (b)), positive-anchor pair is more aligned as the sum of similarity scores between positive segments and [CLS] representation (i.e. sum of row-wise cell values) is higher than the negative counterpart. Boundaries of all slot types (presented by red, pink, orange boxes) are correctly recognized in the positive sample in contrast to the negative counterpart. For SentCL ((c), (d)), positive-anchor pair assigns a higher similarity score to the aligned slot phrase (red box) while negative-anchor pair reduces similarity scores between potential relevant slot phrase (orange box).

formation for SI task, leading to improved performance on both datasets beyond *Ours (w/o CL)*.

Segment-level Supervision (SegCL) As observed in Figure 4a, 4b, semantic representation of the given utterance via [CLS] token is closer to the UPL-derived segments as compared to random segment counterparts due to the higher sum of similarity score (0.1281 > -0.6304). UPL segments also correctly identify nearly all of the slot ground truth labels (i.e. artist (*music_item*), paulinho da costa (*artist*), my (*playlist_owner*), very nearly nashville (*playlist*)) in the given utterance while random segmentations truncate the slot phrases incorrectly.

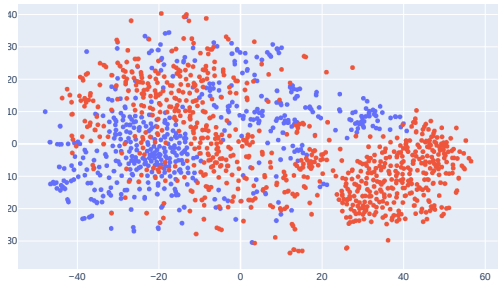
Sentence-level Supervision (SentCL) On the sentence level, besides the commonly aligned phrases (i.e. *add tune to* vs *add rupee to*), the model recognizes corresponding playlists in anchor and positive samples (i.e. *black metal playlist* vs *ultra metal playlist*) and assign competitive similarity score between them. On the other hand, potential relevant noun phrases (i.e. ultra metal playlist (*playlist*) and andy silvester sound track (*sound track*)) between anchor and negative samples are assigned low similarity score. This showcases the model's capability in (1) correctly recognizing and bringing the important slot phrases in positive-anchor pair closer together, (2) reducing

`[Golden]:`	**play me a top-ten song by | phil ochs | on | groove shark add | spirit touches ground | to my | leche con chocolate list**
`[DP-LB]:`	play me a top-ten song by phil ochs on | groove shark add spirit touches ground to my leche con | chocolate list
`[DP-RB]:`	play me a top-ten | song by phil ochs on groove shark add spirit touches ground | to my leche con chocolate list
`[Autophrase]:`	|play | me a top-ten | song | by | phil ochs| on | groove shark add | spirit | touches ground to my leche con | chocolate list
`[UCPhrase]:`	play me a top-ten | song by phil ochs on | groove shark add spirit touches | ground to my leche con | chocolate list
`[USSI]:`	play | me | a | top-ten | song | by phil ochs on groove shark add spirit | touches | ground | to my | leche con chocolate list
`[Ours(w/o CL]:`	play me | a top-ten | song | by phil | ochs on groove shark add spirit touches ground to | my leche con chocolate list
`[Ours(with CL)]:`	**play me | a top-ten | song | by | phil ochs | on groove shark add | spirit touches ground | to my | leche con chocolate list**

Figure 5: Sample Segmentation Results from *Comparable* Methods in comparison with **Golden Slot Labels** on SNIPS dataset where "|" denotes the *Break* as introduced in Figure 1. Red, Blue denote distinct slot label segments. The colors are repeated in *Comparable* Methods to showcase the consistency of models' predictions with ground truth labels under the condition no more than 2 tokens in the segments are mispredicted.

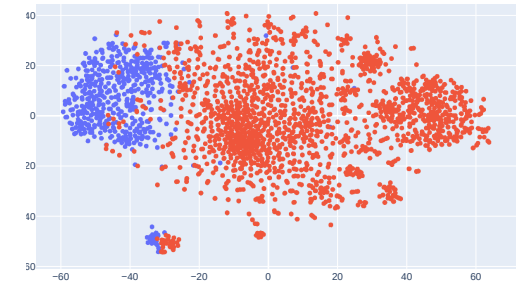
the importance of potential relevant slot phrases across samples with different intents. The Similarity Matrix presented in Figure 4c also indicates the strong segment alignment between positive and anchor samples as the diagonal cells receive higher similarity score than most of the other cells within the same column or row.

Qualitative Case Study Additional Case Studies presented in Figure 5 demonstrate the effec-

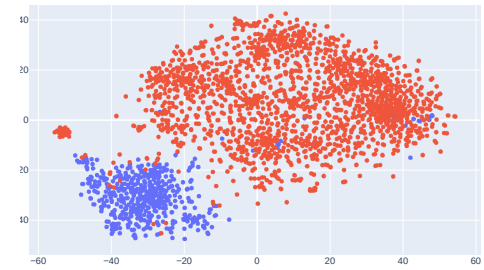


(a) Pre-trained BERT (Train Slots)

(b) Refined BERT (Train Slots)



(c) Pre-trained BERT (Test Slots)



(d) Refined BERT (Test Slots)

Figure 6: Slot Value Representation Visualization of the raw original pre-trained BERT and raw Refined BERT via SI on sample slot types from training set SNIPS_P1 ((a), (b)) and testing set SNIPS_P2 ((c), (d)). Blue and Red denotes slot values from randomly sampled ground truth slot types.

Table 5: Evaluation of SF task over 3 runs on Emerging Intents in SNIPS_P2 and ATIS_P2 datasets.

	SNIPS_P2		
	S-P	S-R	S-F1
Original BERT	14.11 ± 0.47	17.78 ± 0.82	15.73 ± 0.62
Refined BERT	**15.08 ± 0.48**	**19.61 ± 0.23**	**17.05 ± 0.38**
	ATIS_P2		
Original BERT	66.67 ± 0.82	63.35 ± 1.35	64.96 ± 0.74
Refined BERT	**70.12 ± 0.85**	**63.64 ± 0.48**	**66.72 ± 0.66**

tiveness of our proposed framework in capturing slot phrases. Despite the imperfect segmentations, *Ours* captures phrases closer to the ground truth slot labels than other *Comparable* baselines. In fact, our identified phrases "spirit touches ground" and "leche con chocolate list" are exact matches for the golden slot labels. Our proposed multi-level CL refining mechanism is also shown to correct mistakes of the original model. (from "by phil" in *Ours (w/o CL)* to "phil och" in *Ours (with CL)*.

6.2 Generalization towards Emerging Intents

Visual Representation We first visualize the representations of two randomly sampled slot types produced by the raw original BERT and our Refined BERT (via SI objectives). As observed in Figure 6, our Refined BERT clusters the representations of samples with the same slot types for both training and testing sets more effectively than the original BERT in the embedding space, leading to far clearer separation boundaries between the sampled slot types. For Train Slots, embeddings of slot values from each slot type are nearly disentangled, implying our Refined BERT is capable of recognizing slot types without explicit slot training objectives and token-level label access. In addition, when applied to new intents and slots in P2 dataset, our SI framework produces refined BERT with better semantic representations for tokens from the same slot types as observed in Figure 6c,6d.

Quantitative Evaluation As observed in Table 5, when generalized to emerging intents and slots, our Refined BERT outperforms the traditional BERT while fine-tuning on both datasets in all slot evaluation metrics. This showcases the generalization capability of our model across different sentence-level intent labels. In addition, the consistent improvement in SF evaluation implies that SI training objectives via UPL and CL refinement provide more guidance to the PLM for the downstream token-level task without explicit training objectives and label requirements.

7 Conclusion

In our work, we propose the study of token-level Slot Induction (SI) via an Unsupervised Pre-trained Language Modeling (PLM) Probing in conjunction with Contrastive Learning (CL) objectives. By leveraging both unsupervised signals from PLM and sentence-level signals from intent labels via CL objectives, our proposed framework not only

achieves competitive performance in comparison with other unsupervised phrasal segmentation baselines but also bridges the gap in performance with *Upper Bound* methods that require additional token-level labels on two NLU benchmark datasets. We also demonstrate that our proposed SI training is capable of refining the original PLM, resulting in more effective slot representations and benefiting downstream SF tasks when generalized towards emerging intents. Further studies of better exploitation of full-depth segmentation trees, enhanced segment augmentation mechanisms and better semantic alignment extraction between slots and intents are promising directions for our future work. We also seek to extend the current SI studies beyond English and towards multilingual NLU systems. (Nguyen and Rohrbaugh, 2019; Qin et al., 2022; Nguyen et al., 2023)

Limitations

Our proposed framework assumes a fixed hyperparameter depth d for UPL segmentation tree. In other words, only segments extracted at the depth d are considered for CL objectives. d is tuned with each dataset's validation set. However, as our main objective is to investigate the effects of UPL and CL objectives, we leave the full tree exploitation as future extensions for our work.

Secondly, the goal of our SI is to identify the slot phrase boundaries. The label type predictions for recognized slot phrases are beyond the scope of our investigation. Therefore, direct end-to-end evaluation of SI in mitigating slot label scarcity issues cannot be directly evaluated. Our rationale for dividing the task into 2 separate steps (i.e. slot boundary induction and slot label prediction) is as follows: As the complete SI is a complex task, breaking it down not only allows for direct and focused evaluation of the proposed framework's contribution at individual steps but also minimizes error propagation from intermediate steps to a single end-task metric. This rationale is further supported by our empirical study in Section 6. The proposed *USSI* whose objective unifies both aforementioned steps underperforms *Ours(w/o CL)* and *Ours(full)* when evaluated at the slot boundary induction step.

Acknowledgement

This work is supported in part by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

We would like to acknowledge the use of the facilities of the High Performance Computing Division and High Performance Research and Development Group at the National Center for Atmospheric Research and the use of computational resources (doi:10.5065/D6RX99HX) at the NCAR-Wyoming Supercomputing Center provided by the National Science Foundation and the State of Wyoming, and supported by NCAR's Computational and Information Systems Laboratory.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

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*, pages 12–16.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, and Alfio Gliozzo. 2021. Robust retrieval augmented generation for zero-shot slot filling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1939–1949.

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.

Xiaotao Gu, Zihan Wang, Zhenyu Bi, Yu Meng, Liyuan Liu, Jiawei Han, and Jingbo Shang. 2021. *UCPhrase: Unsupervised Context-Aware Quality Phrase Tagging*, page 478–486. Association for Computing Machinery, New York, NY, USA.

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.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sang goo Lee. 2020. Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. In *International Conference on Learning Representations*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Ye Liu, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and S Yu Philip. 2021. Dense hierarchical retrieval for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 188–200.

Samuel Louvan and Bernardo Magnini. 2020. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 480–496.

Hoang Nguyen and Gene Rohrbaugh. 2019. Cross-lingual genre classification using linguistic groupings. *Journal of Computing Sciences in Colleges*, 34(3):91–96.

Hoang Nguyen, Chenwei Zhang, Congying Xia, and S Yu Philip. 2020. Dynamic semantic matching and aggregation network for few-shot intent detection. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1209–1218.

Hoang Nguyen, Chenwei Zhang, Tao Zhang, Eugene Rohrbaugh, and Philip Yu. 2023. Enhancing cross-lingual transfer via phonemic transcription integration. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9163–9175, Toronto, Canada. Association for Computational Linguistics.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Libo Qin, Qiguang Chen, Tianbao Xie, Qixin Li, Jian-Guang Lou, Wanxiang Che, and Min-Yen Kan. 2022. GL-CLeF: A global–local contrastive learning framework for cross-lingual spoken language understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2677–2686, Dublin, Ireland. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018a. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1825–1837.

Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018b. Learning named entity tagger using domain-specific dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2054–2064.

AB Siddique, Fuad Jamour, Luxun Xu, and Vagelis Hristidis. 2021. Generalized zero-shot intent detection via commonsense knowledge. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1925–1929.

Yusheng Su, Xu Han, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. 2021. Css-lm: A contrastive framework for semi-supervised fine-tuning of pre-trained language models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2930–2941.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. 2010. What is left to be understood in atis? In *2010 IEEE Spoken Language Technology Workshop*, pages 19–24. IEEE.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388.

Di Wu, Liang Ding, Fan Lu, and Jian Xie. 2020a. SlotRefine: A fast non-autoregressive model for joint intent detection and slot filling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1932–1937, Online. Association for Computational Linguistics.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. 2020b. Perturbed masking: Parameter-free probing for analyzing and interpreting bert. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176.

Congying Xia, Chenwei Zhang, Hoang Nguyen, Jiawei Zhang, and Philip Yu. 2020. Cg-bert: Conditional text generation with bert for generalized few-shot intent detection. *arXiv preprint arXiv:2004.01881*.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and S Yu Philip. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3090–3099.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Dian Yu, Mingqiu Wang, Yuan Cao, Izhak Shafran, Laurent Shafey, and Hagen Soltau. 2022. Unsupervised slot schema induction for task-oriented dialog. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1174–1193.

Mengshi Yu, Jian Liu, Yufeng Chen, Jinan Xu, and Yujie Zhang. 2021. Cross-domain slot filling as machine reading comprehension. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, Montreal, QC, Canada*, pages 19–26.

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*, pages 2578–2589.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and S Yu Philip. 2019. Joint slot filling and intent detection via capsule neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5259–5267.

Table 6: Hyperparameters for SNIPS and ATIS datasets (SI task)

	d	β	τ_s	τ_d	δ	γ
SNIPS	3	0.2	0.1	0.05	0.3	0.7
ATIS	4	0.2	0.05	0.1	1.0	0.2

A Slot Induction Baselines

For fair comparisons across all baselines, we leverage BERT (Devlin et al., 2019) as the backbone PLM architecture (if applicable).

A.1 Upper Bound Baselines

• **Joint BERT FT**: Fully Supervised Joint Sequence Labeling and Sentence Classification model is trained on top of fine-tuning BERT embeddings with available golden training slot and intent labels.
• **SpaCy** (Honnibal et al., 2020): Industrial-strength NLP tagging methodology that leverages pre-trained NP chunking model.
• **FlairNLP** (Akbik et al., 2018): Neural Language Modeling in junction with pre-trained Sequential Labeling (NER and POS).

A.2 Comparable Baselines

• **Dependency Parsing** (Right/Left-branching (RB/LB): Parameter-free methods for sentence segmentation. Result from the best depth is reported.
• **AutoPhrase** (Shang et al., 2018a): Statistical phrase tagging method utilizing high quality massive corpus as additional Knowledge Base (KB).
• **UCPhrase** (Gu et al., 2021): Phrase tagging method leveraging co-occurrence word frequency and PLM attention maps.
• **USSI** (Yu et al., 2022): Unsupervised Slot Schema Induction method leveraging attention distribution of PLM and additional constraints from Probabilistic Context-free Grammar (PCFG) (Kim et al., 2020). For completeness, additional experiments in leveraging the proposed in-domain training objectives with SpanBERT PLM (Joshi et al., 2020) are provided in Appendix D.
• **Ours (w/o CL)**: Fixed UPL is directly used for inference without additional CL refinement. Same depth d is used as our proposed model **Ours (full)** and its variant **Ours (w/o SentCL)**.
• **Ours (w/o SentCL)**: Our model variant that is trained only with SegCL objectives ($\mathcal{L}_s$). The model does not leverage sentence-level intent label information (SentCL) during training.

Table 7: Ablation study of SpanBERT PLMs with in-domain training objectives on SNIPS and ATIS datasets in terms of H-Mean over 3 runs. ¶ denotes models that do not require random initializations.

	SNIPS	ATIS
SpanBERT ¶	43.15	35.05
USSI (Yu et al., 2022)	48.61 ± 0.69	36.63 ± 1.93
Ours (SpanBERT w CL)	**53.25 ± 0.29**	**40.07 ± 2.34**

B Slot Induction Implementation (P1)

We train our proposed SI model with batch size of 16, learning rate 1e-5 for 10 epochs. The remaining hyperparameters for individual datasets are reported in Table 6 respectively for SI task. We tune our hyperparameters based on each dataset's P1 validation set via grid search for $\beta, \tau_s, \tau_d, \delta, \gamma$, except for d. For depth d, we conduct inference of PLM probing (i.e. Ours (w/o CL)) on P1 validation sets and select d with the highest H-Mean performance. The same depth d is used consistently across different variants of our proposed framework in the empirical study. Our reported results are reported based on 3 runs with different seeds.

C Slot Filling Implementation (P2)

As the objective of SF is to compare different BERT models (i.e. Original BERT vs Refined BERT via SI objectives), we keep the Sequence Labelling architecture simple and similar between the two models. Specifically, we stack the traditional CRF layer (Lafferty et al., 2001) on top of the corresponding BERT models. The overall model is fine-tuned on SF task with available training slot labels in P1 training data. The model is fine-tuned with batch size of 16, learning rate of 0.01 for CRF and Linear layer, BERT learning rate of 1e-5 for 10 epochs. The testing results (Table 5) are reported on P2 of each dataset as an average over 3 runs. Both training and inference for Appendix B and C are conducted on NVIDIA Titan RTX GPU.

D SpanBERT-based Model

Yu et al. (2022) proposed additional self-supervised in-domain training on Task-oriented Dialogue datasets. For fair comparisons with (Yu et al., 2022), we conduct additional studies training the same backbone SpanBERT PLM architecture (Joshi et al., 2020) with their proposed self-supervised in-domain training objectives on our training SNIPS_P1 and ATIS_P1 datasets and report test results in Table 7. To evaluate the effectiveness of our multi-level CL objectives, in Table 7, **Ours (SpanBERT w CL)** follows the induction mechanisms proposed by Yu et al. (2022) instead of UPL mentioned in Section 4.1. The only difference between **Ours (SpanBERT w CL)** and USSI is our proposed multi-level CL objectives

As demonstrated in Table 7, **Ours (SpanBERT w CL)** achieves consistent improvements over USSI on both SNIPS and ATIS datasets (4.64% and 3.44% respectively) under the same training architecture and in-domain training objectives. This observation implies the effectiveness of our multi-level CL objectives (SegCL and SentCL).