Semi-supervised New Slot Discovery with Incremental Clustering

Yuxia Wu
Xi'an Jiaotong University
wuyuxia@stu.xjtu.edu.cn

Lizi Liao
Singapore Management University
lzliao@smu.edu.sg

Xueming Qian
Xi'an Jiaotong University
qianxm@mail.xjtu.edu.cn

Tat-Seng Chua
Sea-NExT Joint Lab, NUS
dcscts@nus.edu.sg

Motivation: Slot Filling

- Identify contiguous word spans in an utterance based on slots to represent the meaning of the user.

I would like a moderately priced restaurant in the north part of town.

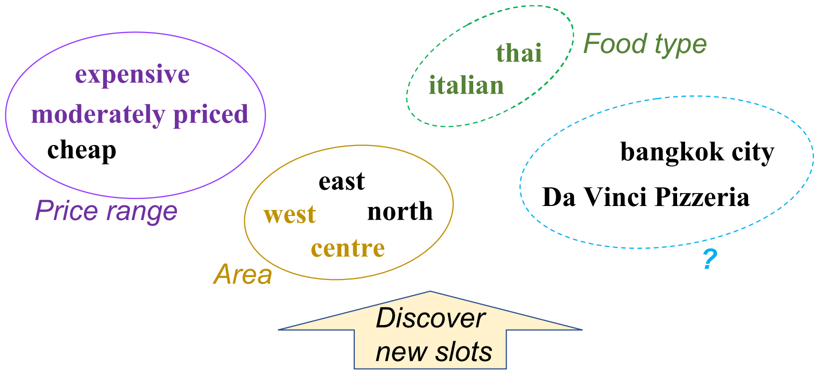
Slot: Price range

Slot: area

- Supervised methods can only recognize pre-defined entity types from a limited slot set and require a significant amount of labeled training data

Motivation: Slot Filling

- In practical settings, new unseen slots may emerge after the deployment of the dialogue system, rendering these supervised models ineffective.

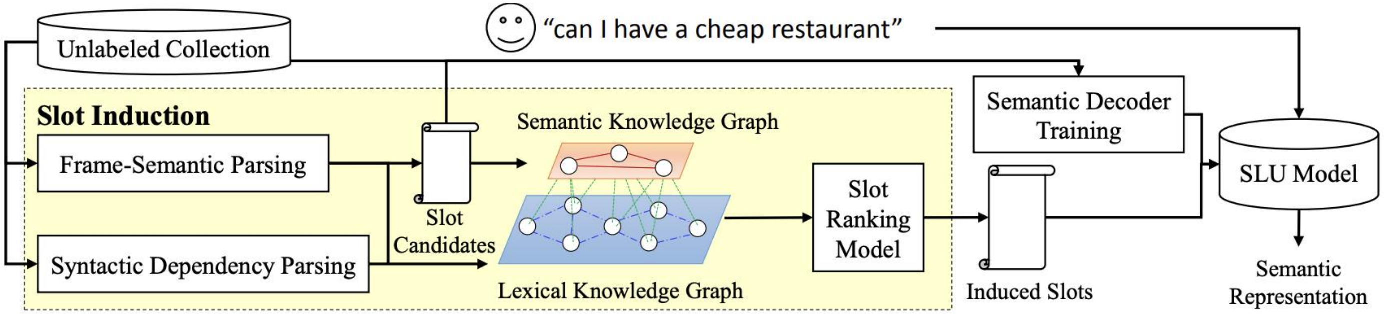


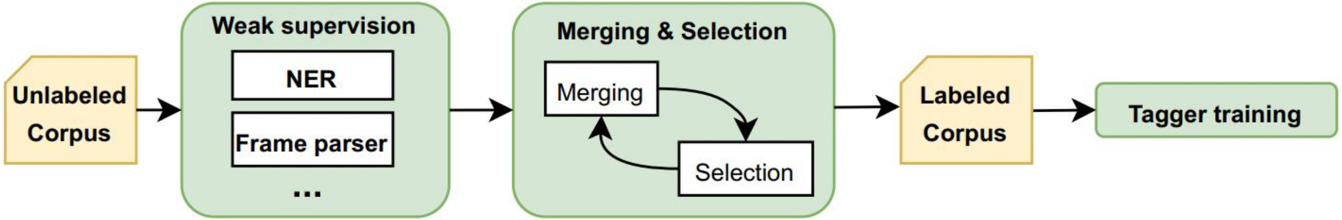
Utterances:

I'd like to find a **west** side restaurant that is **expensive**.
I want a **moderately priced** restaurant in the **east** part of town.
Is there a **cheap** restaurant in the **north** part of town?
bangkok city serves **thai** food in the **centre** of town.
Da Vinci Pizzeria is a **cheap** **italian** restraunt in the **north** area.
…

Existing Methods: Automatic Slot Induction

1) Extract candidate slots and values
2) Obtain slots via ranking. (chen 2014., chen 2015., Hudeˇcek et al., 2021)



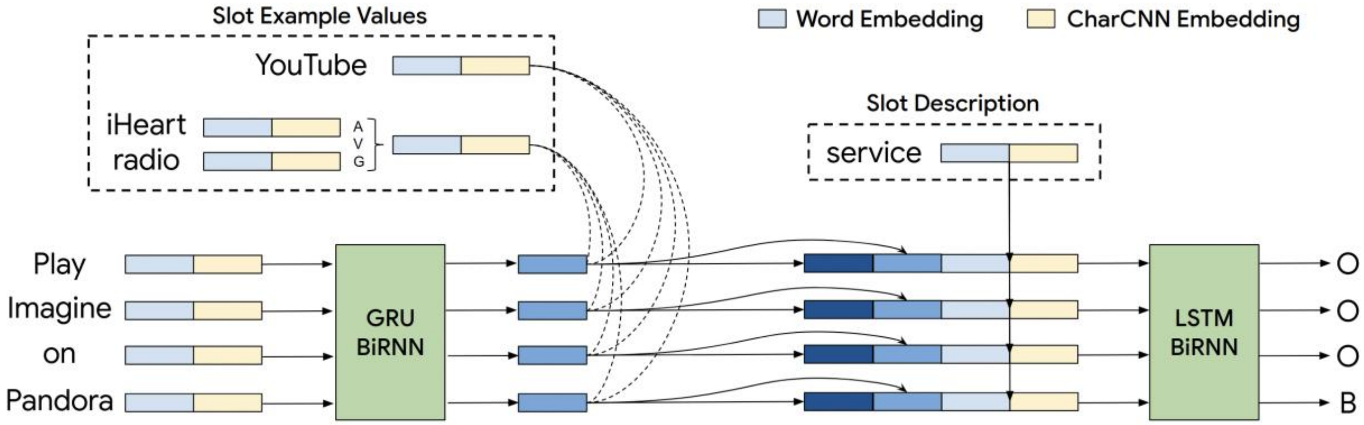


Limitations:

- The ranking process needs deliberate human intervention and largely affects the final results.
- Instead of entirely without labeled data, we often have access to a small amount of that in real practice.

Existing Methods: Cross-domain Adaptation

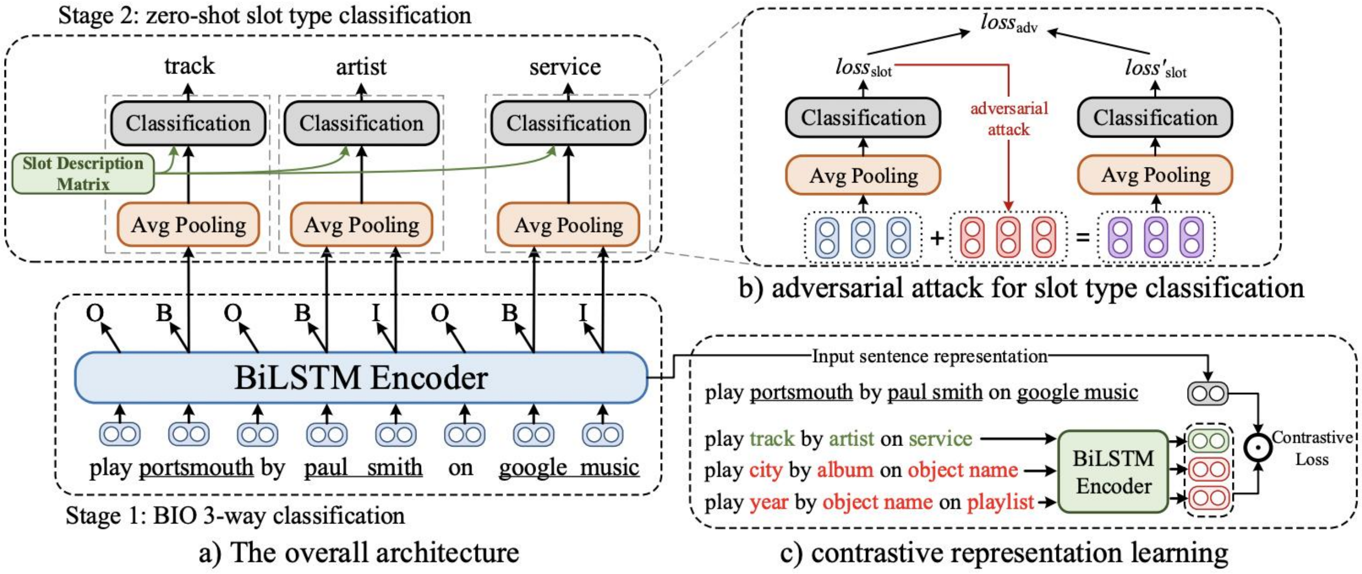
- Identify unseen slots in the target domain by leveraging evidence from labeled data in the source domain
- One stage methods: (Bapna et al., 2017; Shah et al., 2019; Lee and Jha, 2019; Hou et al., 2020;Oguz and Vu, 2021).



Rely on prior knowledge: slot description, example values

Existing Methods: Cross-domain Adaptation

- Two or more stages methods: (Liu et al., 2020; He et al., 2020; Siddique et al., 2021).
    1) slot values identification by sequence labeling
    2) slot type classification



Drawbacks of Existing Methods

- Pay less attention to slot value identification
- Heavily rely on auxiliary information
- Fail to provide proper guidance for clustering-friendly features

Problem Setting

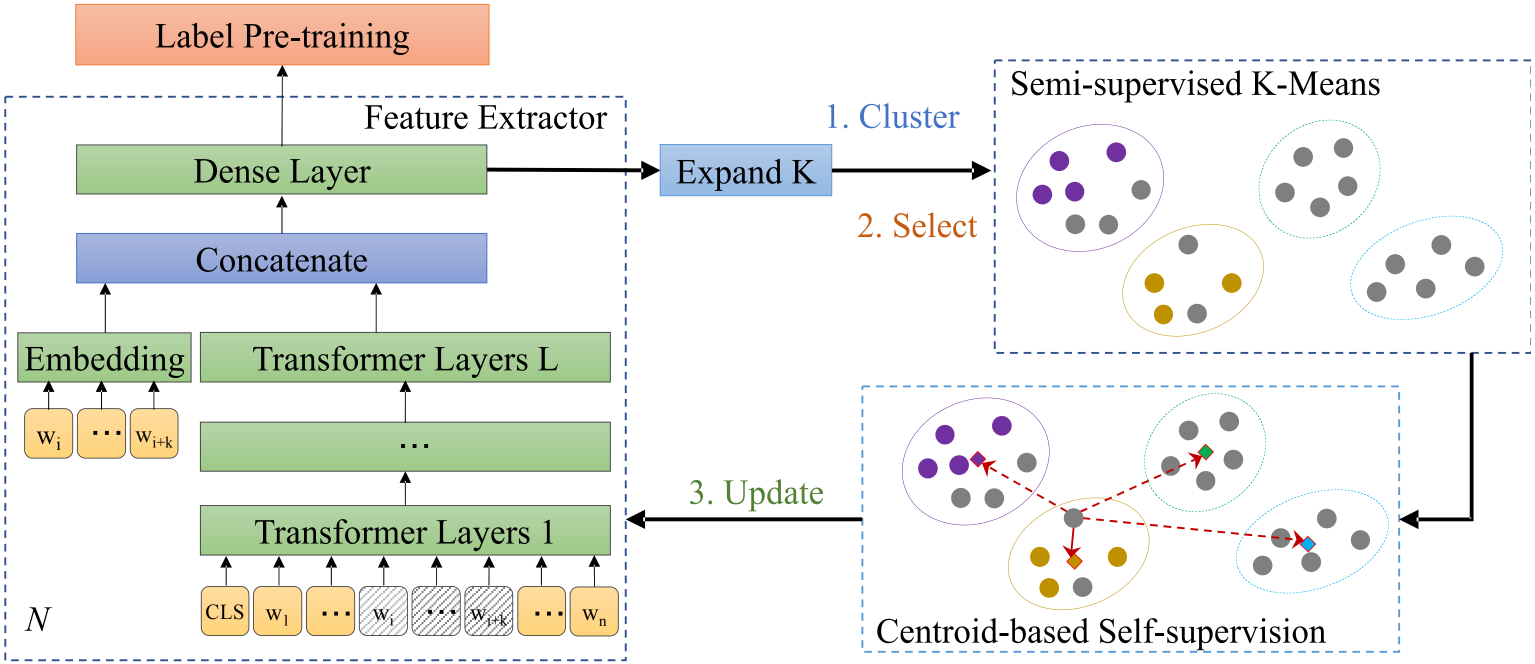
- Suppose there is a candidate value $x = \langle w_i, \cdots, w_{i+k} \rangle$ of length $|x| = k + 1$

  identified from the utterance U

- The whole dataset $\mathcal{D}$ with N candidate values
- Limited labeled data: $\mathcal{D}^{\mathcal{L}} = \{x_i, y_i\}_{i=1}^{M} \in \mathcal{X} \times \mathcal{Y}_{\mathcal{L}}$
- Unlabeled data: $\mathcal{D}^{\mathcal{U}} = \{x_i, y_i\}_{i=1}^{N-M} \in \mathcal{X} \times \mathcal{Y}_{\mathcal{U}}$ (N-M is not given)

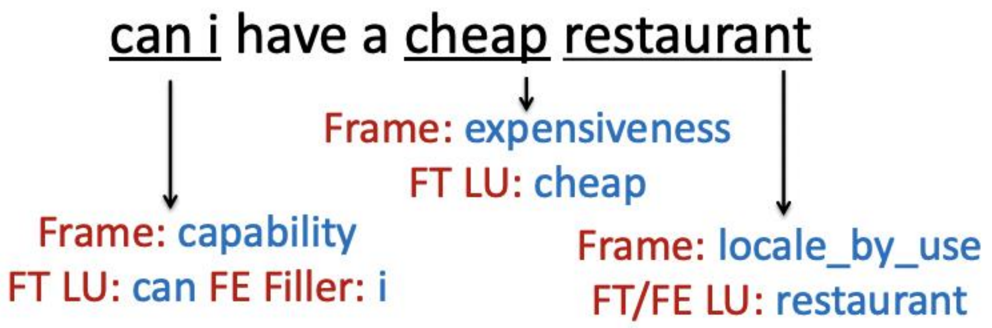
Proposed Method: SIC

- SIC: Semi-supervised New Slot Discovery with Incremental Clustering



Proposed Method: Candidate Value Extraction and Filtering

- Extracting: frame semantic parser SEMAFOR (Das et al.,2010, 2014) and NER
- Filtering:
  - remove the stop words by the NLTK
  - remove these spans that appeared less than a certain number of times
  - delete these frequently appeared but meaningless values



Frame semantic parser

https://github.com/swabhs/open-sesame

Proposed Method: Feature Extractor Pre-training

Candidate value $x = \langle w_i, \cdots, w_{i+k} \rangle$ with n tokens ($0 \leq k \leq n$)

- Inner representation

$$\langle \mathbf{e}_i, \cdots, \mathbf{e}_{i+k} \rangle = BERT(\langle w_i, \cdots, w_{i+k} \rangle),$$
$$\mathbf{v}^{inner} = mean\_pooling(\langle \mathbf{e}_i, \cdots, \mathbf{e}_{i+k} \rangle),$$

- Context representation

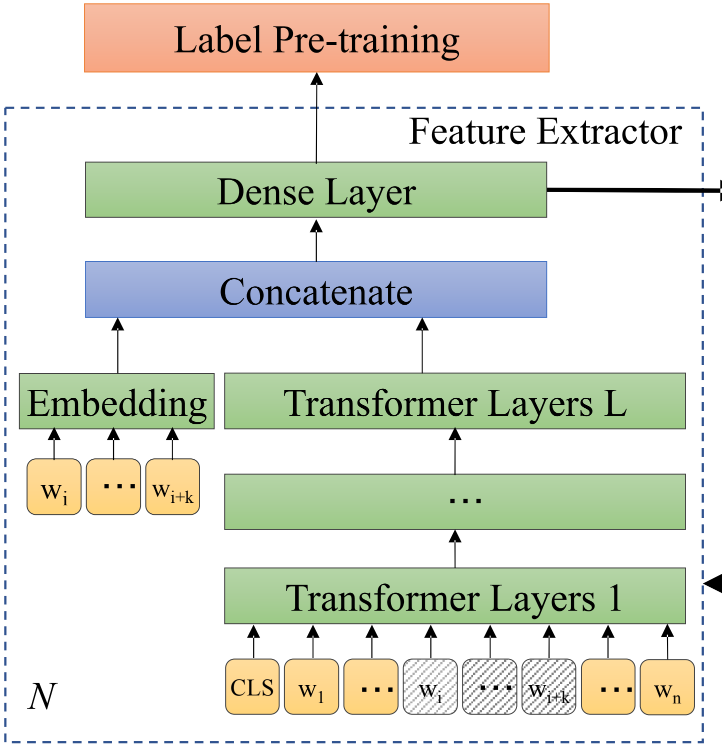
$$U' = \langle w_1, \cdots, [mask]_i, \cdots, [mask]_{i+k}, \cdots, w_n \rangle$$
$$\langle \mathbf{h}_1, \cdots, \mathbf{h}_n \rangle = BERT(U'),$$
$$\mathbf{v}^{context} = mean\_pooling(\langle \mathbf{h}_i, \cdots, \mathbf{h}_{i+k} \rangle),$$

- Final representation

$$\mathbf{v} = tanh(\mathbf{W}_1[\mathbf{v}^{inner}; \mathbf{v}^{context}]^T + \mathbf{b}_1),$$

- Pre-training

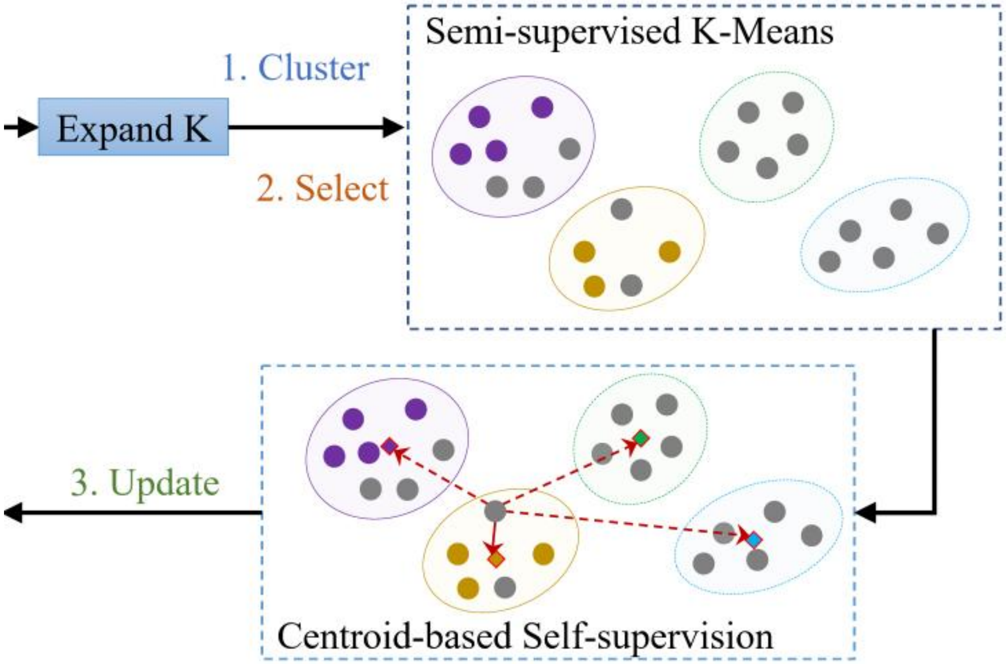
$$\mathbf{y} = Softmax(\mathbf{W}_2\mathbf{v}^T + \mathbf{b}_2),$$



Proposed Method: Incremental Clustering

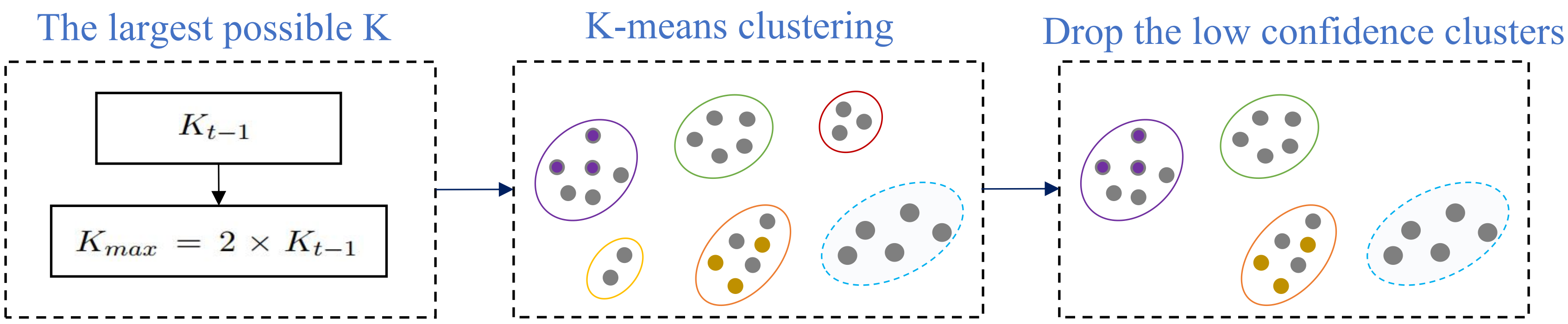
Iteratively clustering

- Expand clustering number K
- Semi-supervised clustering
- Centroid-based self-supervision



Proposed Method: Incremental Clustering

- Expand K



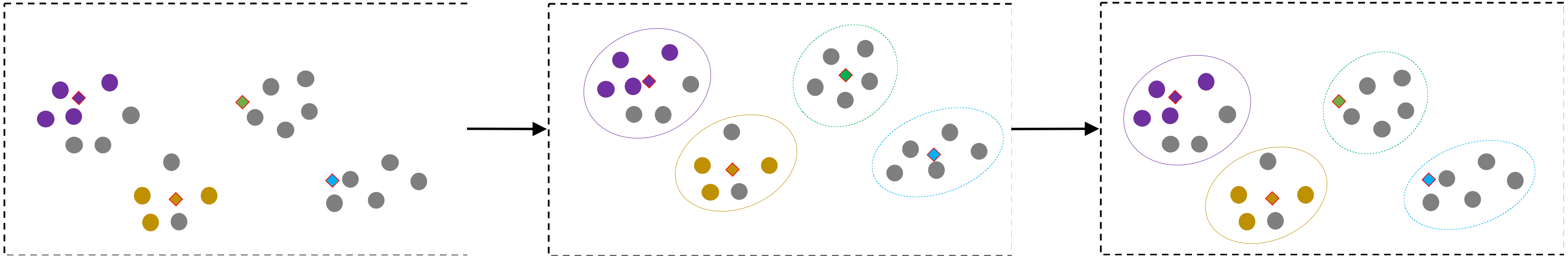
$$K_t = \sum_{i=1}^{K_{max}} \mathbb{1}(|C_i| > \epsilon),$$

the size of the *i*-th cluster

Proposed Method: Incremental Clustering

- Semi-supervised clustering

Initialize the K_t centroids — Modify the assignments — Update the centroids



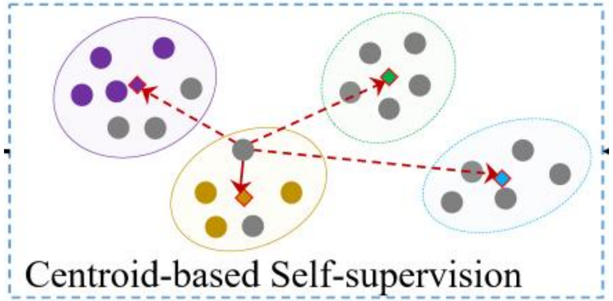
Labeled data: the average representations of the candidate values belonging to each slot.

Unlabeled data: kmeans++

Force the assignments of the labeled data unchanged

Proposed Method: Incremental Clustering

- Centroid-based Self-Supervision



- Select samples with high confidence: $\mathcal{D}^{\mathcal{S}} = \{x_i, y_i : s_i \geq \gamma\}$

  s_i : the similarity score of sample v_i to its centroid

  γ: threshold on the score

- Expand the labeled set: $\mathcal{D}^{\mathcal{L}} = \mathcal{D}^{\mathcal{L}} \cup \mathcal{D}^{\mathcal{S}}$,
- Update the centroids: $\{c_1, \cdots, c_{K_t}\}$
- Update the feature extractor:

$$L_s = -\sum_{i=1}^{|\mathcal{D}^{\mathcal{L}}|} \log \frac{exp(\mathbf{v}_i \cdot \mathbf{c}_i / \tau)}{\sum_{\mathbf{c}_j \neq \mathbf{c}_i} exp(\mathbf{v}_i \cdot \mathbf{c}_j / \tau)},$$

Experiments:

Datasets:

Dataset	Domain	# of utterances	# of slots
CamRest676 (CR)	Restaurant	2,744	4
WOZ-hotel(WH)	Hotel	14,435	9
WOZ-attr (WA)	Attraction	7524	8
Cambridge SLU (CS)	Restaurant	10,569	5
ATIS (AT)	Flight	4,978	79

We randomly select 75% of all slots as the known slots and choose 10% data for each slot as labeled data.

Experiment Results: F1 Score

		CR		CS		WH		WA		AT	
		Extr	*GT*	*Extr*	*GT*	*Extr*	*GT*	*Extr*	*GT*	*Extr*	*GT*
Sup.	*Tag-supervised*	**0.778**	-	0.724	-	**0.742**	-	0.731	-	**0.848**	-
	Dict-supervised	0.705	-	0.753	-	0.750	-	0.665	-	0.678	-
Unsup.	*Chen et al.*	0.535	-	0.590	-	0.382	-	0.375	-	0.616	-
	WeakS-notag	0.552	-	0.664	-	0.388	-	0.383	-	0.648	-
Weakly-sup.	*WeakS-full*	0.665	-	0.692	-	0.548	-	0.439	-	0.710	-
Semi-sup.	*BERT-KCL**	0.189	0.224	0.131	0.188	0.178	0.346	0.560	0.731	0.492	0.584
	*BERT-MCL**	0.188	0.321	0.129	0.210	0.179	0.332	0.532	0.729	0.504	0.591
	*BERT-DTC**	0.131	0.303	0.138	0.206	0.170	0.334	0.545	0.670	0.543	0.578
	*CDAC+**	0.204	0.270	0.178	0.221	0.174	0.332	0.552	0.641	0.582	0.588
	DeepAligned	0.663	0.901	0.633	0.899	0.378	0.750	0.644	0.719	0.629	0.676
	SIC(Ours)	**0.706**	**0.913**	**0.770**	**0.969**	**0.588**	**0.824**	**0.761**	**0.851**	**0.638**	**0.721**

Table 1: Results compared with baselines on F1. * indicates that the method uses the ground truth slot number. *Extr* and *GT* represent that we use extracted candidate values or ground truth values respectively.

- SIC performs consistently better than all the baseline methods on nearly all five datasets.
- Compared with Unsup. and Weakly-sup. methods, SIC maintains a large performance gap.

Experiment Results: Clustering Metrics

	CR		CS		WH		WA		AT	
	NMI	*ARI*	*NMI*	*ARI*	*NMI*	*ARI*	*NMI*	*ARI*	*NMI*	*ARI*
*BERT-KCL**	22.06	12.48	12.56	6.57	12.10	8.99	63.27	61.27	29.02	54.42
*BERT-MCL**	64.21	63.03	10.60	3.77	11.49	9.19	63.25	61.20	30.65	55.43
*BERT-DTC**	64.08	34.25	11.35	2.61	11.67	8.83	64.64	65.51	27.61	52.43
*CDAC+**	21.22	13.55	31.12	26.27	11.71	9.05	69.07	71.04	30.69	55.90
DeepAligned	82.05	80.01	88.77	90.20	81.53	76.21	70.84	68.59	71.44	78.78
SIC(Ours)	**82.61**	**81.23**	**90.62**	**92.88**	**87.71**	**86.86**	**71.36**	**72.87**	**73.70**	**78.85**

Table 2: Results compared with baselines via cluster-based metrics. * indicates that the method uses the ground truth slot number.

- SIC achieves the best performance on all datasets.
- DeepAligned shows comparable performance. However, the fixed cluster number and representations of clusters hinder its adaptability during the learning process.

Summary:

- We design a semi-supervised learning scheme for new slot discovery, which does not require any prior knowledge about new slots.
- We perform clustering and feature extractor training iteratively to harvest high-quality self-supervised signals and learn discriminative features for grouping values to slots.
- Thank you for your listening!
- Q & A (Email: wuyuxia@stu.xjtu.edu.cn)