Discovering New Intents via Constrained Deep Adaptive Clustering with Cluster Refinement

基于深度约束聚类的对话新意图发现

林廷恩[1, 2], 徐华[1,2,*],张瀚镭[1,2,3]

[1]清华大学计算机科学与技术系 智能技术与系统国家重点实验室

[2] 北京信息科学与技术国家研究中心

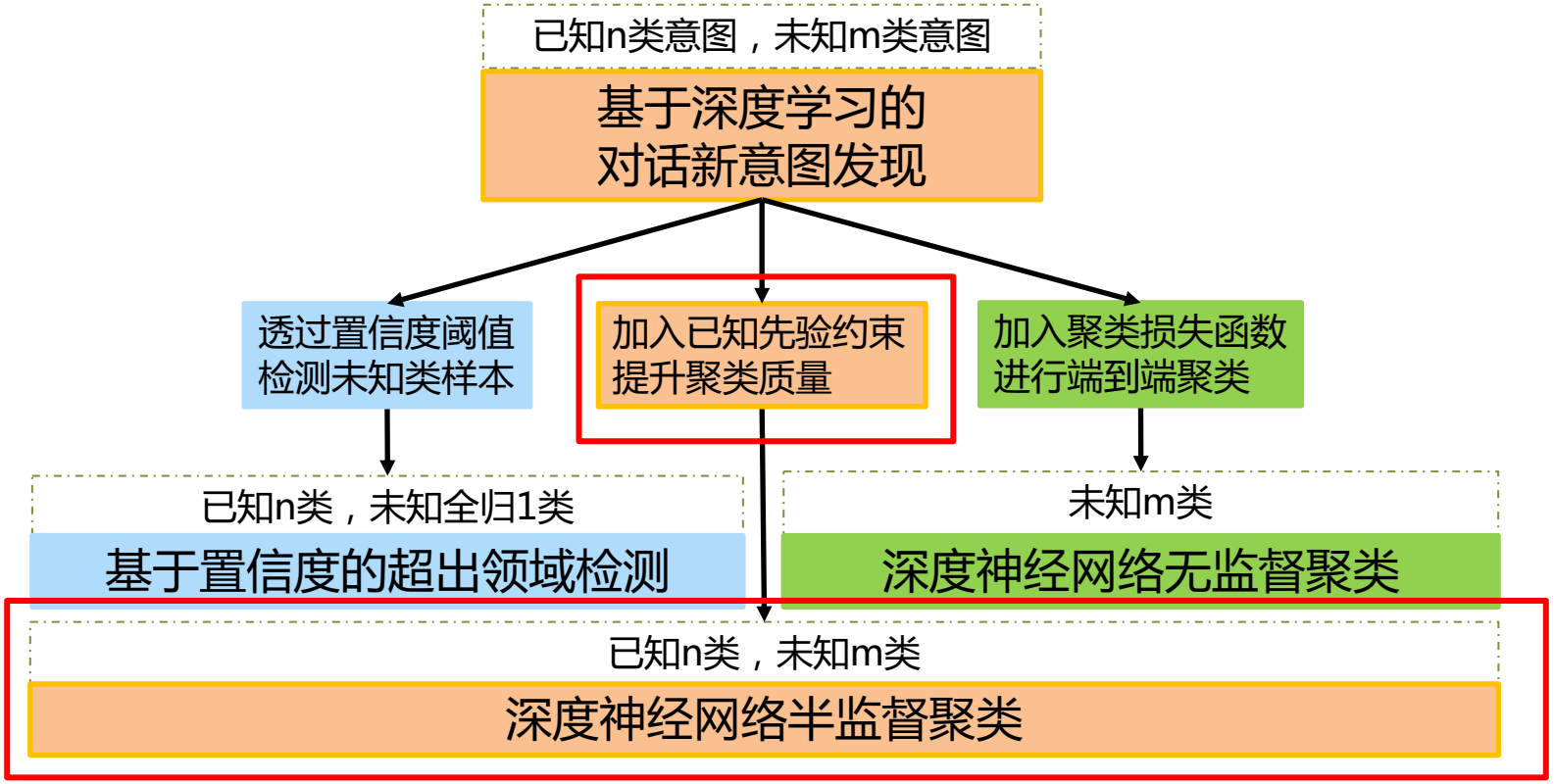
[3] 北京交通大学计算机与信息技术学院

2019.12.22

研究背景

- 智能语音助手的全面崛起，人机对话系统的需求激增
  - 国外：亚马逊 Alexa、苹果 Siri、谷歌语音助理...
  - 国内：阿里小蜜、百度小度、微软小冰、腾讯小微、小米小爱
- 对话系统构建困难，用户需求复杂多样
  - 对话系统的自然语言理解模块在设计时，难以涵盖所有的用户意图
- 如何发现尚未被满足的用户意图？ (新意图)
  - 在未被满足的意图中，有哪些需求是相似的？
  - 在未被满足的意图中，有哪些需求是最多的？

研究框架


已知n类意图，未知m类意图
基于深度学习的
对话新意图发现
透过置信度阈值
检测未知类样本
加入已知先验约束
提升聚类质量
加入聚类损失函数
进行端到端聚类
已知n类，未知全归1类
基于置信度的超出领域检测
未知m类
深度神经网络无监督聚类
已知n类，未知m类
深度神经网络半监督聚类

研究现状分析

- 新意图发现的相关学术研究，目前仍十分匮乏 [Haponchyk 2018]
  - 本文是基于DAC [Chang 2017] 的改进工作

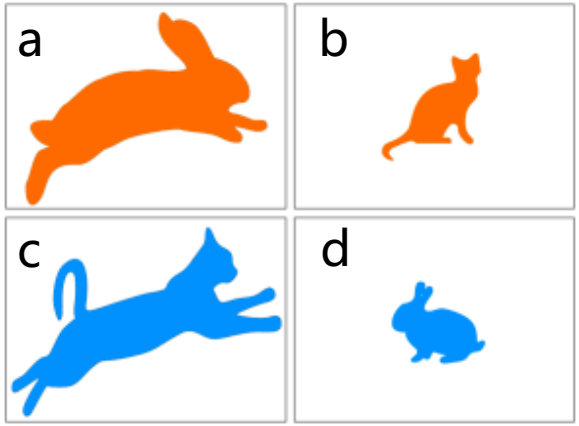
相关内容	算法	思想
基于超出领域检测 (Out-of-domain)	Joint OOD [Kim 2018]	联合学习意图分类与超出领域检测
	DeepUNK [Lin 2019]	基于度量学习的超出领域检测
基于无监督聚类	AMR [Hakkani-Tür 2018]	基于语义解析的模版聚类
	COC [Padmasundari 2018]	基于集成学习的句子表征聚类
基于半监督聚类	LSP [Haponchyk 2018]	基于局部标注的半监督聚类
	CBI [Forman 2015]	基于用户交互的半监督聚类

为什么是半监督聚类？

- 「意图」的划分方式，是人为主观定义的
  - 样本可以有多种不同的划分方式
  - 缺乏先验知识引导，很难得到理想的聚类结果

(a) What's the color of apples?
(b) When will this apple be ripe?
(c) Do you like apples?
(d) What's the color of oranges?
(e) When will this orange be ripe?
(f) Do you like oranges?

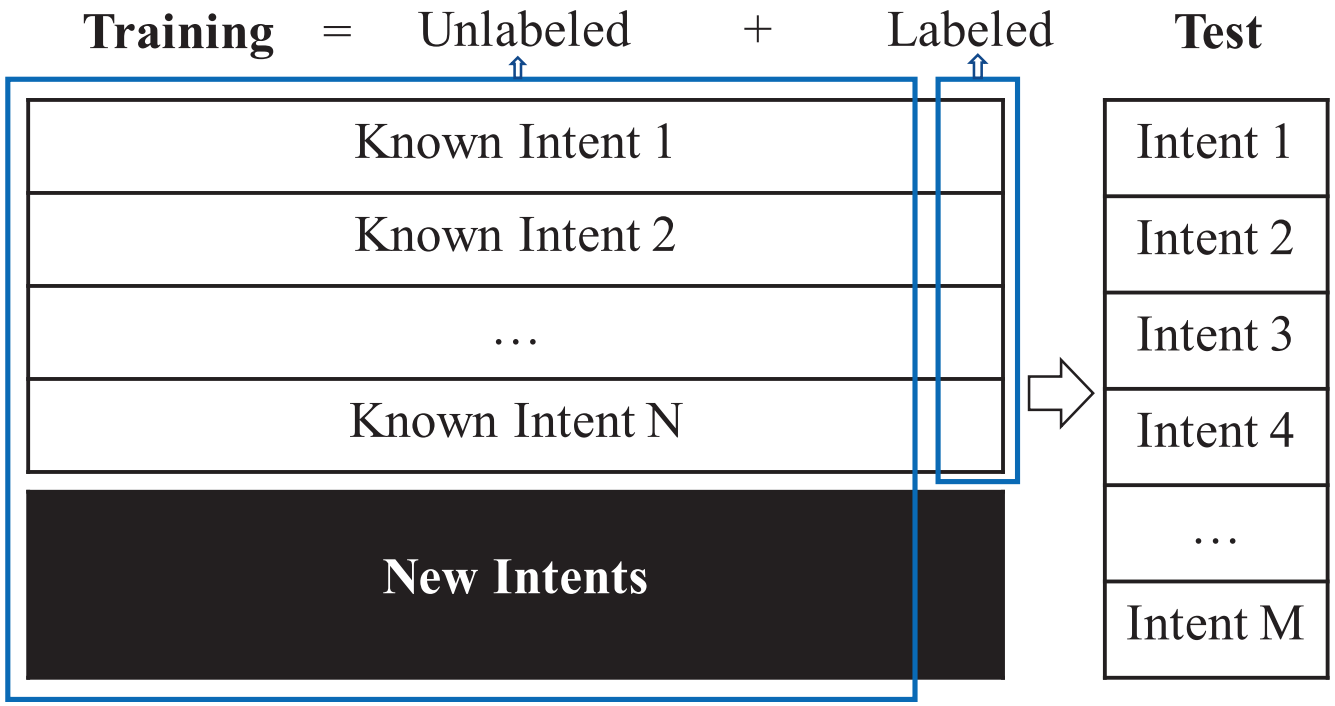
水果类型：{a,b,c}, {d, e, f}
问题类型：{a,d}, {b,e}, {c,f}



颜色类型：{a,b}, {c,d}
姿态类型：{a,c}, {b,d}
动物类型：{a,d}, {b,c}

新意图发现-范例

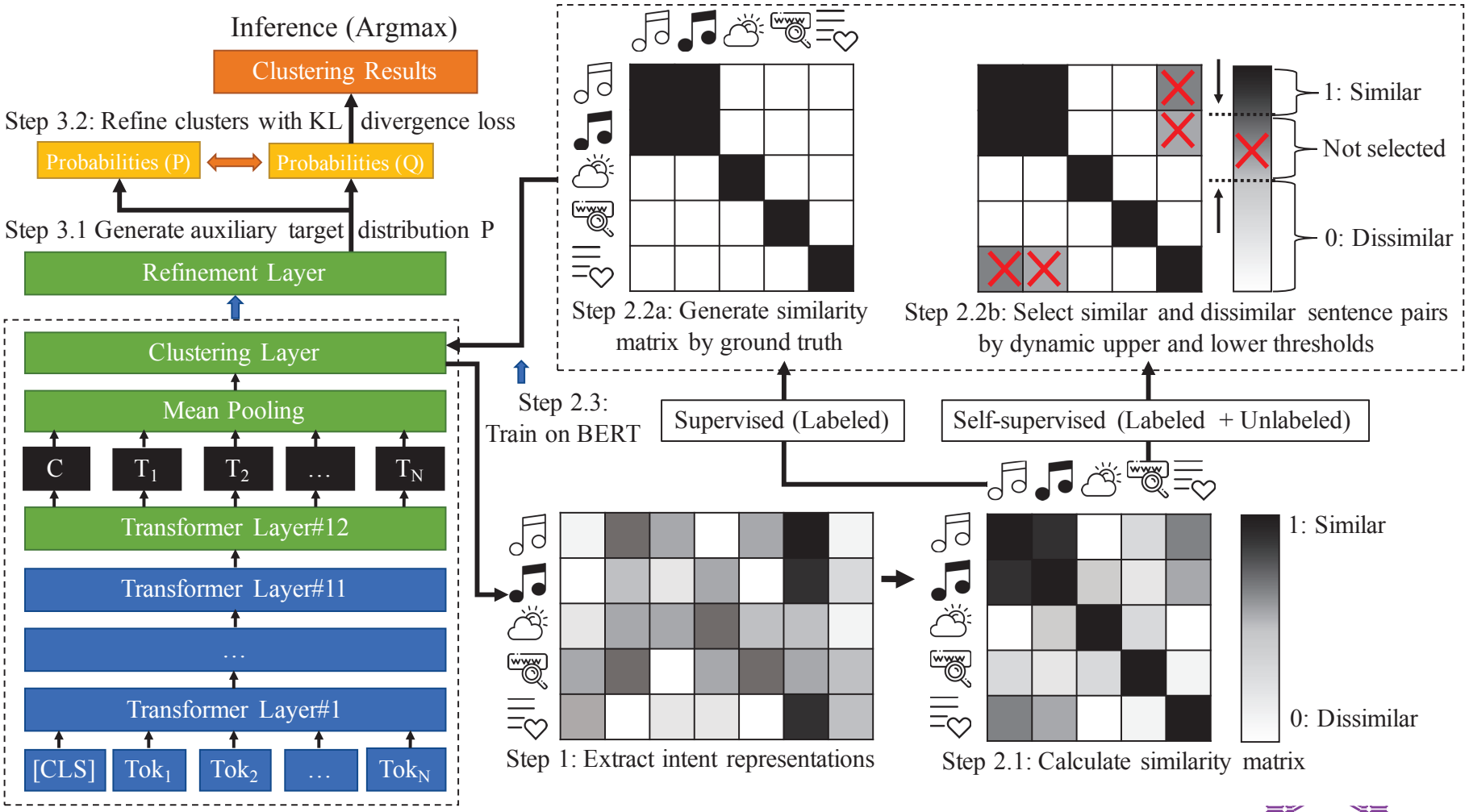
- 加入已知先验约束，提升聚类型能
  - 已知75%类别，数据10%有标注 (实验设置)



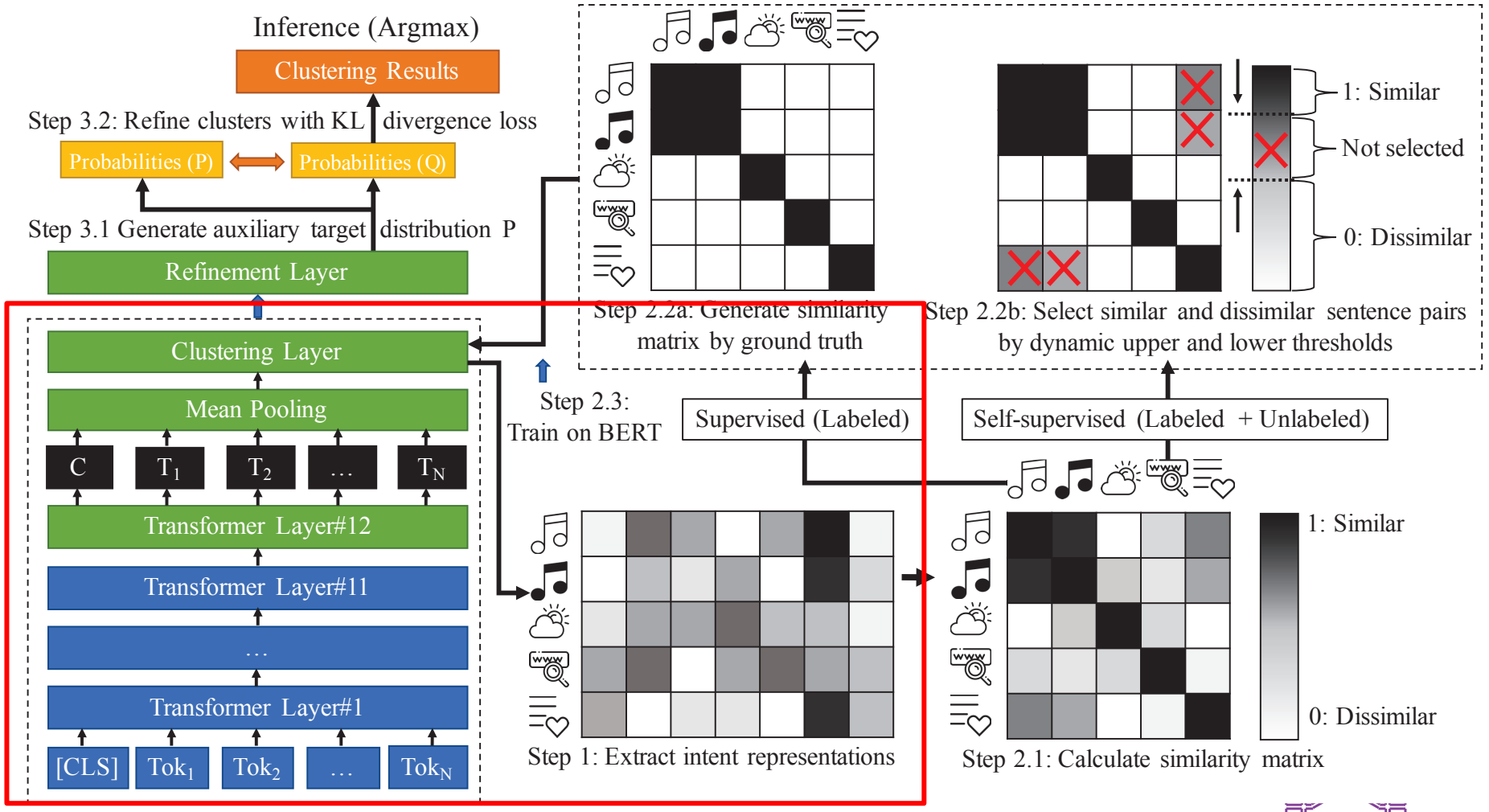
CDAC+：三个步骤

1. Intent representation
   - 基于预训练BERT来获得意图表示
2. Pairwise-Classification with Similarity Loss
   - 透过相似与否二分类任务，将先验知识 implicitly 迁移到聚类任务上
     - 先验知识：预训练BERT、少量标注数据
   - 分为 Supervised / Self-supervised Step，交替优化
3. Cluster Refinement with KLD loss
   - 透过cluster refinement 来消除低置信度 assignment

CDAC+：流程图


Inference (Argmax)
Clustering Results
Step 3.2: Refine clusters with KL divergence loss
Probabilities (P)
Probabilities (Q)
Step 3.1 Generate auxiliary target distribution P
Refinement Layer
Clustering Layer
Mean Pooling
C
T1
T2
…
TN
Transformer Layer#12
Transformer Layer#11
…
Transformer Layer#1
[CLS]
Tok1
Tok2
…
TokN
Step 2.3: Train on BERT
Step 2.2a: Generate similarity matrix by ground truth
Step 2.2b: Select similar and dissimilar sentence pairs by dynamic upper and lower thresholds
1: Similar
Not selected
0: Dissimilar
Supervised (Labeled)
Self-supervised (Labeled + Unlabeled)
Step 1: Extract intent representations
Step 2.1: Calculate similarity matrix
1: Similar
0: Dissimilar

CDAC+ (1) Intent Representation

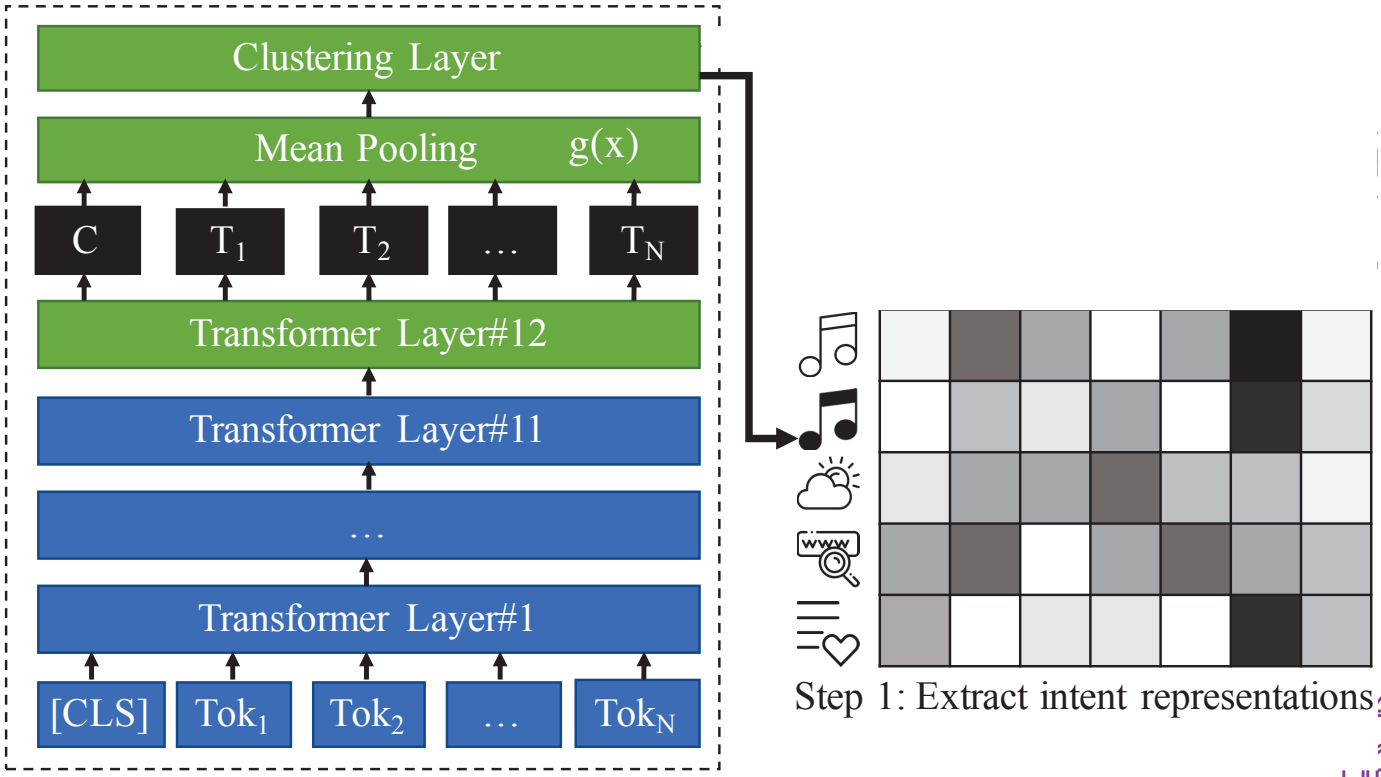


CDAC+ (1) Intent Representation

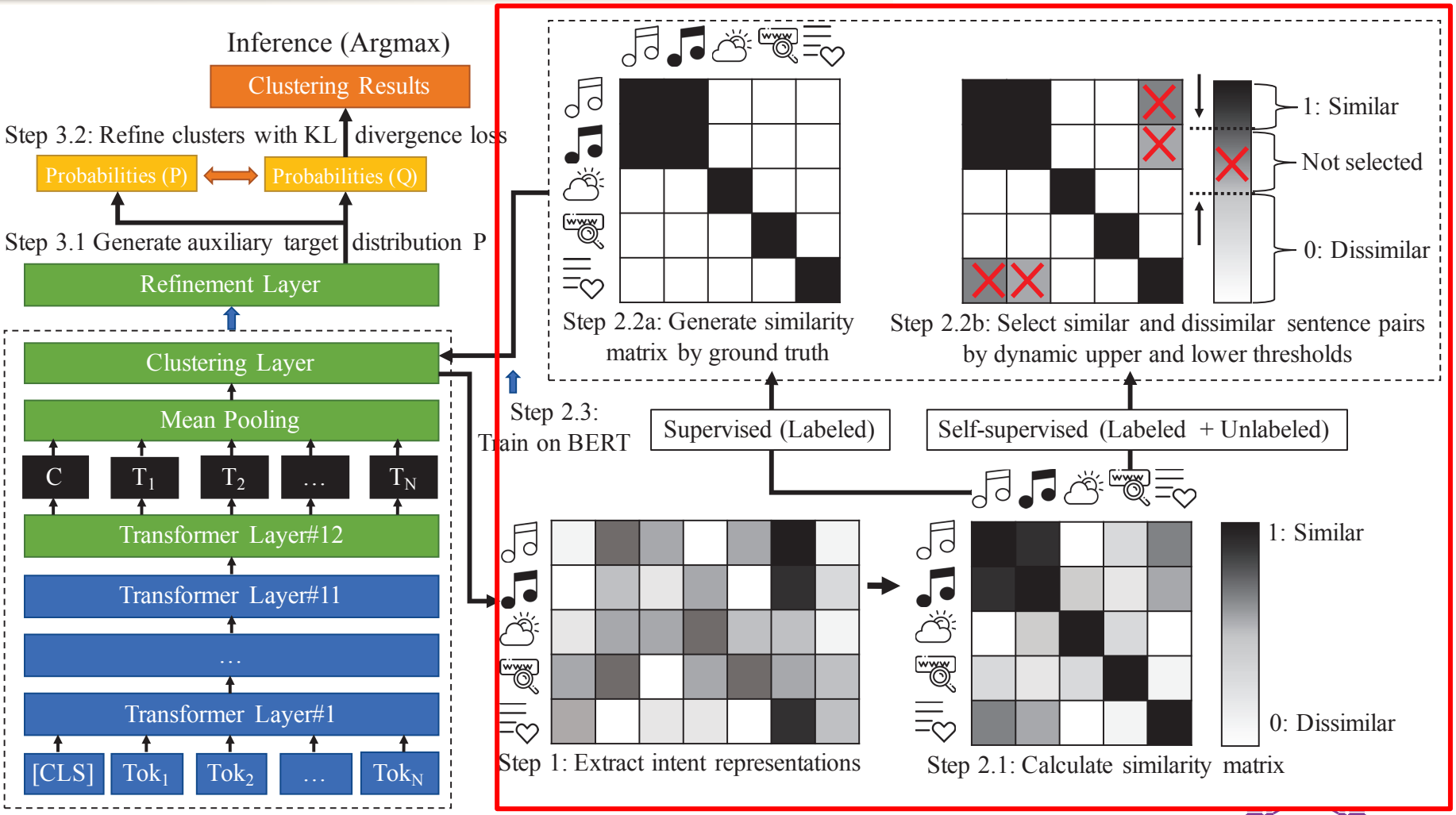
- 基于BERT的意图表示 $I_i \in \mathbb{R}^k$
  - k: 预定义的聚类中心数

$$e_i = \text{mean-pooling}([C, T_1, \cdots, T_N])$$

$$g(e_i) = I_i = W_2(\text{Dropout}(\tanh(W_1 e_i)))$$



CDAC+ (2) Pairwise-Classification

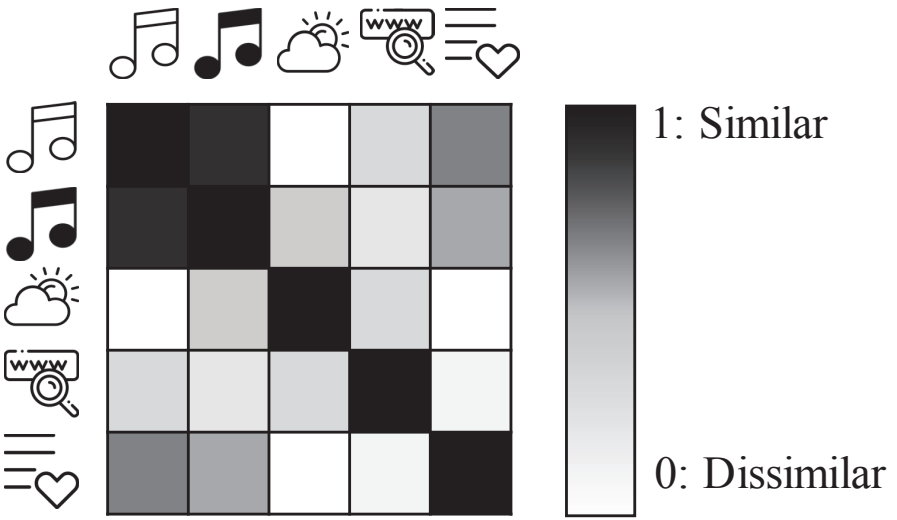


CDAC+ (2) Pairwise-Classification

- 构建相似度矩阵S

$$S_{ij} = \frac{I_i I_j^T}{\| I_i \| \| I_j \|}$$

- 交替训练
  - Supervised step
  - Self-supervised step
    - Dynamic thresholds

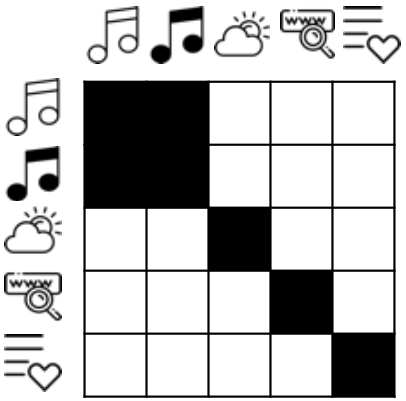


Step 2.1: Calculate similarity matrix

CDAC+ (2) Pairwise-Classification

- Supervised Step
  - 构建相似二分类ground truth矩阵R

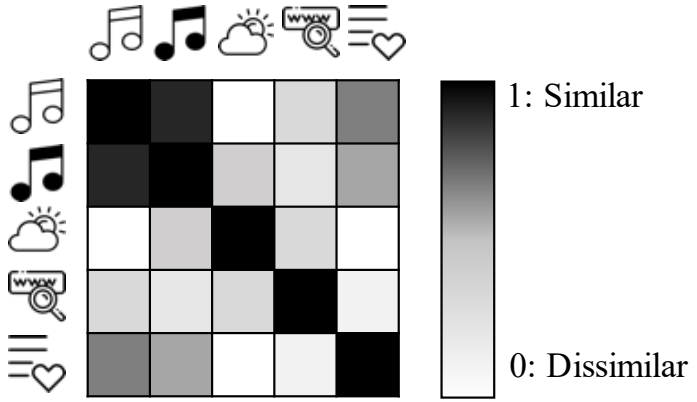
$$R_{ij} := \begin{cases} 1, & \text{if} \quad y_i = y_j, \\ 0, & \text{if} \quad y_i \neq y_j \end{cases}$$



Step 2.2a: Generate similarity matrix by ground truth

  - 矩阵R 再和相似度矩阵S 计算loss

$$\mathcal{L}_{\text{sim}}(R_{ij}, S_{ij}) = -R_{ij} \log(S_{ij}) - (1 - R_{ij}) \log(1 - S_{ij}).$$

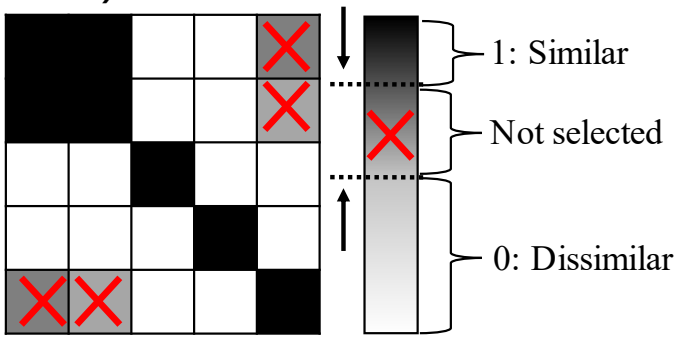


Step 2.1: Calculate similarity matrix

CDAC+ (2) Pairwise-Classification

- Self-supervised Step (Labeled + Unlabeled)
  - 构建自监督ground truth矩阵Ř

$$\hat{R}_{ij} := \begin{cases} 1, \text{if} \quad S_{ij} > u(\lambda) \quad \text{or} \quad y_i = y_j \\ 0, \text{if} \quad S_{ij} < l(\lambda) \quad \text{or} \quad y_i \neq y_j, \\ \text{Not selected , otherwise} \end{cases}$$



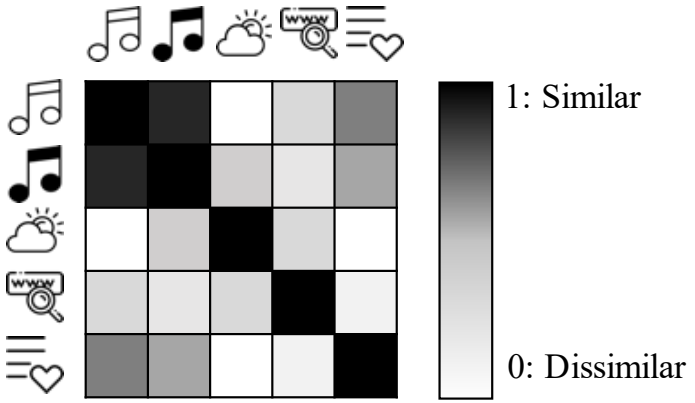
Step 2.2b: Select similar and dissimilar sentence pairs by dynamic upper and lower thresholds

  - 根据动态阈值决定相似/不相似

$$u(\lambda) = 0.95 - \lambda$$
$$l(\lambda) = 0.455 + 0.1 \cdot \lambda$$
$$\min_{\lambda} \mathbf{E}(\lambda) = u(\lambda) - l(\lambda)$$
$$\lambda := \lambda - \eta \cdot \frac{\partial \mathbf{E}(\lambda)}{\partial \lambda}$$

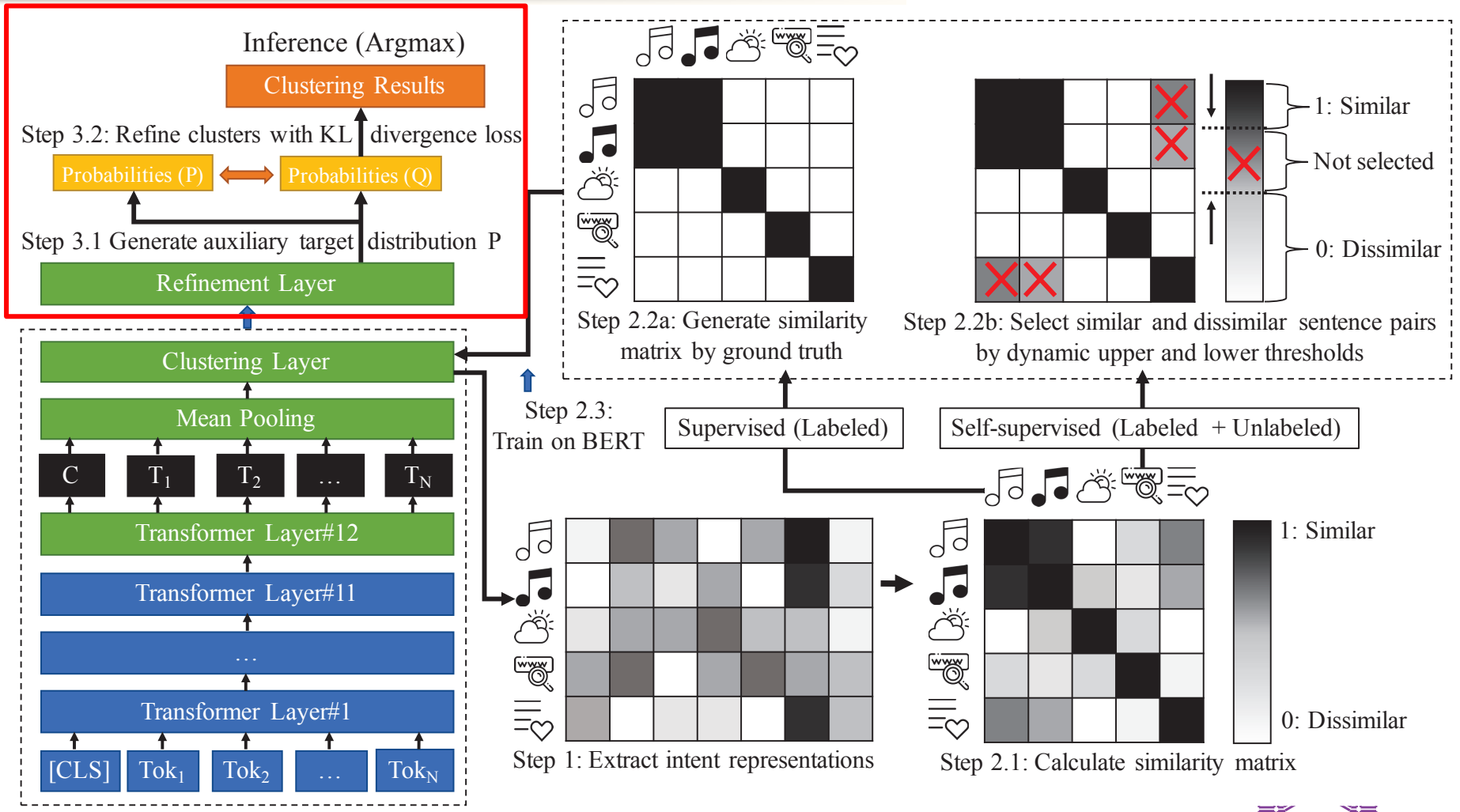
    - 终止条件：$u(\lambda) \leq l(\lambda) \sim 0.5$
  - 矩阵Ř 再和相似度矩阵S 计算loss

$$\hat{\mathcal{L}}_{\text{sim}}(\hat{R}_{ij}, S_{ij}) = -\hat{R}_{ij} \log(S_{ij}) - (1 - \hat{R}_{ij}) \log(1 - S_{ij})$$



Step 2.1: Calculate similarity matrix

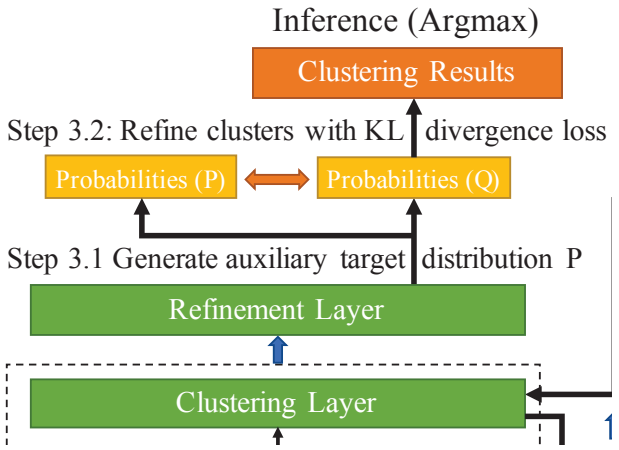
CDAC+ (3) Cluster Refinement



CDAC+ (3) Cluster Refinement

- 透过KM初始化Refinement layer U$\in \mathbb{R}^{k\times k}$
- 透过学生t-分布估计样本i属于和聚类中心j的概率

$$Q_{ij} = \frac{(1+\| I_i - U_j \|^2)^{-1}}{\sum_{j'}(1+\| I_i - U_j \|^2)^{-1}}$$



- 构建辅助分布P

$$P_{ij} = \frac{Q_{ij}^2 / f_i}{\sum_{j'} Q_{ij'}^2 / f_{j'}}$$

- 透过KL散度优化，迫使Q向高置信度样本分布(P)学习

$$\mathcal{L}_{\mathrm{KLD}} = KL(P\|Q) = \sum_i \sum_j P_{ij} \log \frac{P_{ij}}{Q_{ij}}$$

数据集

- 在三个公开数据集上进行实验 (对话意图/短文本)

Dataset	#Classes (Known + Unknown)
SNIPS	7 (5 + 2)
DBPedia	14 (11 + 3)
StackOverflow	20 (15 + 5)

#Training	#Validation	#Test	Vocabulary	Length (max / mean)
13,084	700	700	11,971	35 / 9.03
12,600	700	700	45,077	54 / 29.97
18,000	1,000	1,000	17,182	41 / 9.18

实验结果-主实验

- CDAC+的性能显著优于所有无监督/半监督baselines
  - 最具竞争力无监督baseline：DEC
  - 最具竞争力半监督baseline：BERT-Semi
  - 性能最差的baseline：BERT-KM

		SNIPS			DBPedia			StackOverflow		
	Method	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC
Unsup.	KM	71.42	67.62	84.36	67.26	49.93	61.00	8.24	1.46	13.55
	AG	71.03	58.52	75.54	65.63	43.92	56.07	10.62	2.12	14.66
	SAE-KM	78.24	74.66	87.88	59.70	31.72	50.29	32.62	17.07	34.44
	DEC	84.62	82.32	91.59	53.36	29.43	39.60	10.88	3.76	13.09
	DCN	58.64	42.81	57.45	54.54	32.31	47.48	31.09	15.45	34.26
	DAC	79.97	69.17	76.29	75.37	56.30	63.96	14.71	2.76	16.30
	BERT-KM	52.11	43.73	70.29	60.87	26.6	36.14	12.98	0.51	13.9
Semi-sup.	PCK-means	74.85	71.87	86.92	79.76	71.27	83.11	17.26	5.35	24.16
	BERT-KCL	75.16	61.90	63.88	83.16	61.03	60.62	8.84	7.81	13.94
	BERT-Semi	75.95	69.08	78.00	86.35	72.49	75.31	65.07	47.48	65.28
	CDAC+	**89.30**	**86.82**	**93.63**	**94.74**	**89.41**	**91.66**	**69.84**	**52.59**	**73.48**

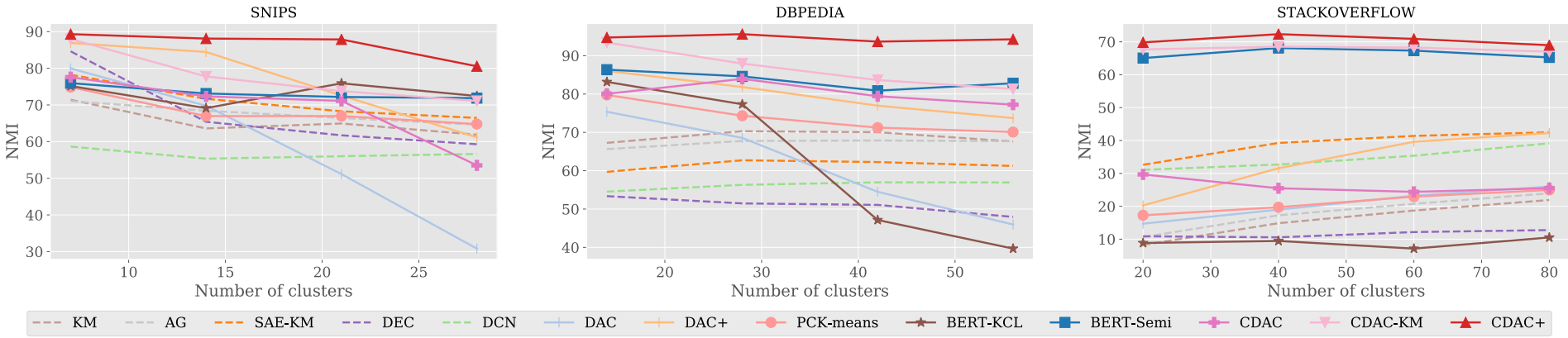
实验结果-对比实验

- 与DAC相比，DAC+ 性能明显提升
- 最具竞争力半监督baseline(对比)：CDAC-KM
  - 与CDAC+相比， CDAC-KM也有不错性能
  - 但鲁棒性远不如CDAC+（见辅助实验）

		SNIPS			DBPedia			StackOverflow		
	Method	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC
Unsup.	DAC	79.97	69.17	76.29	75.37	56.30	63.96	14.71	2.76	16.30
	DAC-KM	86.29	82.58	91.27	84.79	74.46	82.14	20.28	7.09	23.69
	DAC+	86.90	83.15	91.41	86.03	75.99	82.88	20.26	7.10	23.69
Semi-sup.	CDAC	77.57	67.35	74.93	80.04	61.69	69.01	29.69	8.00	23.97
	CDAC-KM	87.96	85.11	93.03	93.42	87.55	89.77	67.71	45.65	71.49
	CDAC+	**89.30**	**86.82**	**93.63**	**94.74**	**89.41**	**91.66**	**69.84**	**52.59**	**73.48**

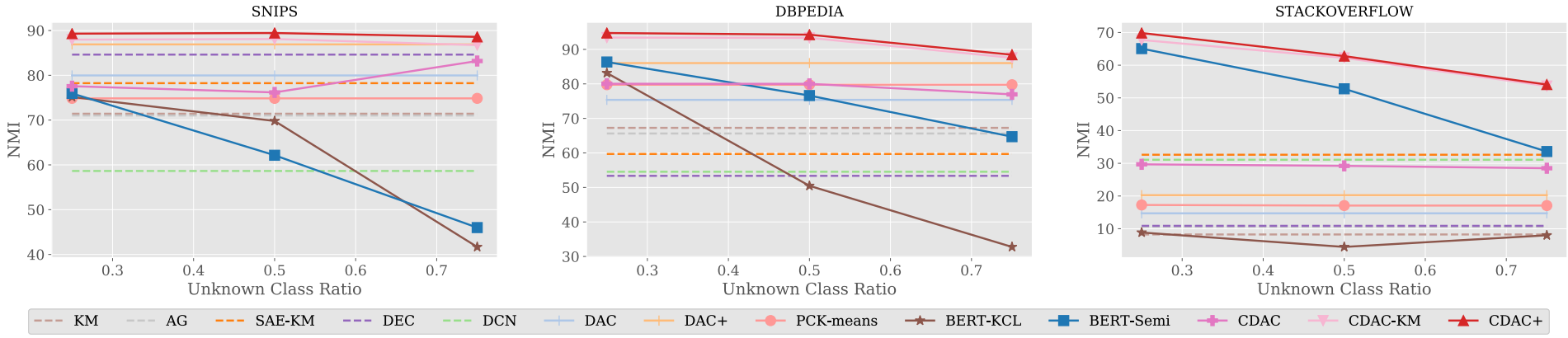
实验结果-辅助实验(1)

- 聚类中心数的影响 (1x, 2x, 3x, 4x)
  - CDAC+性能保持鲁棒
  - CDAC / CDAC-KM 性能在 SNIPS, DBPedia 大幅下降
  - PCK-means 性能在 DBPedia大幅下降



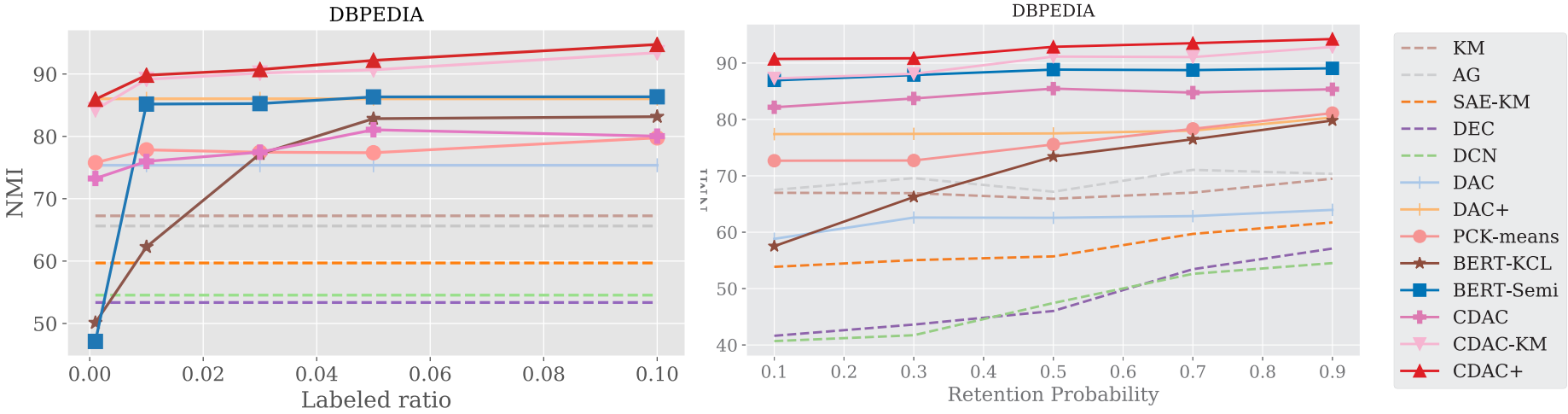
实验结果-辅助实验(2)

- 未知意图比例的影响 (25%, 50%, 75%)
  - CDAC+性能保持相对鲁棒
  - CDAC-Semi 性能在所有数据集都大幅下降
  - BERT-KCL 性能在 SNIPS, DBPedia 大幅下降



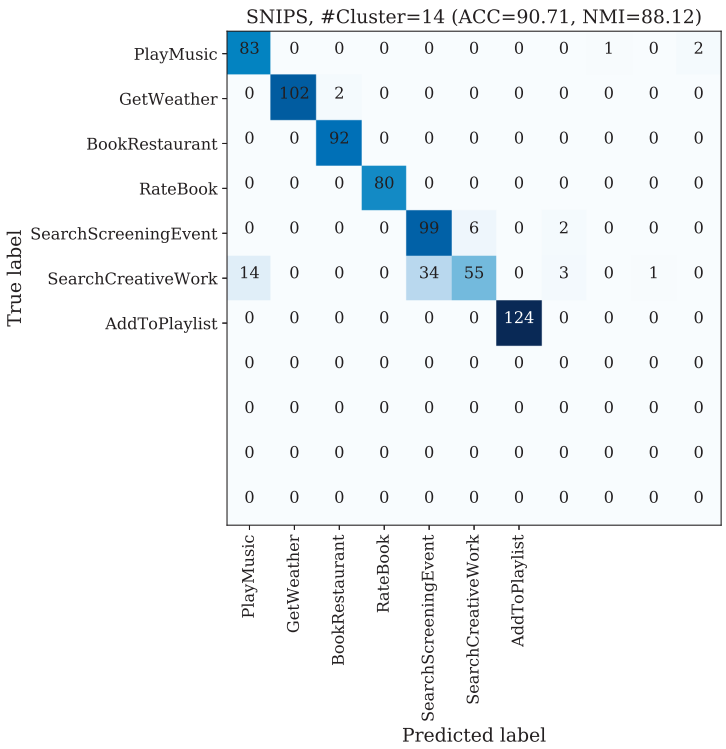
实验结果-辅助实验(3)

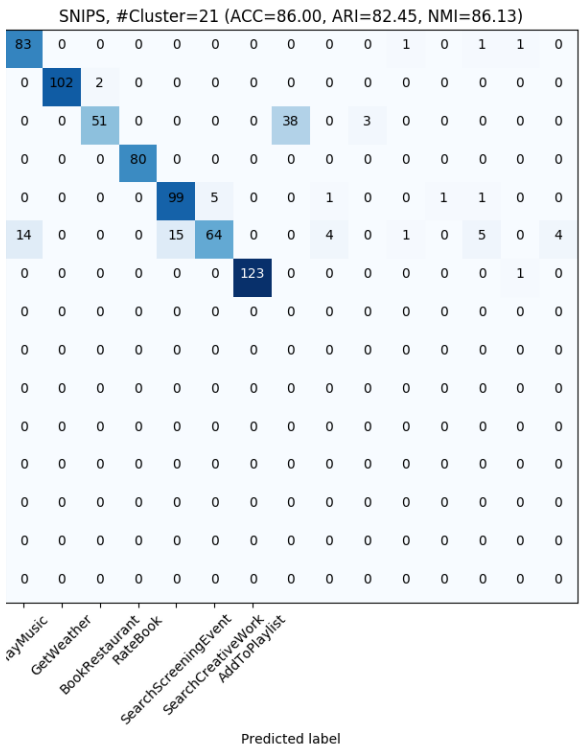
- 左：有标注数据比例的影响 (0.1%, 1%, 3%, 5%, 10%)
  - BERT-Semi, BERT-KCL 性能受有标注数据比例的影响大
- 右：数据不均衡的影响 (10%, 30%, 50%, 70%, 90%)
  - 数字越小，数据越不均衡
  - BERT-KCL 性能受数据不均衡的影响大

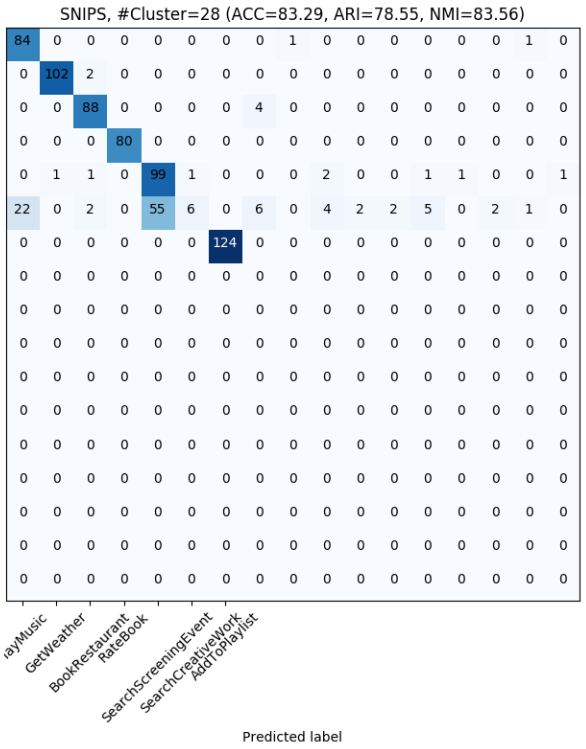


实验结果-案例分析(1)

- CDAC+对聚类中心数不敏感

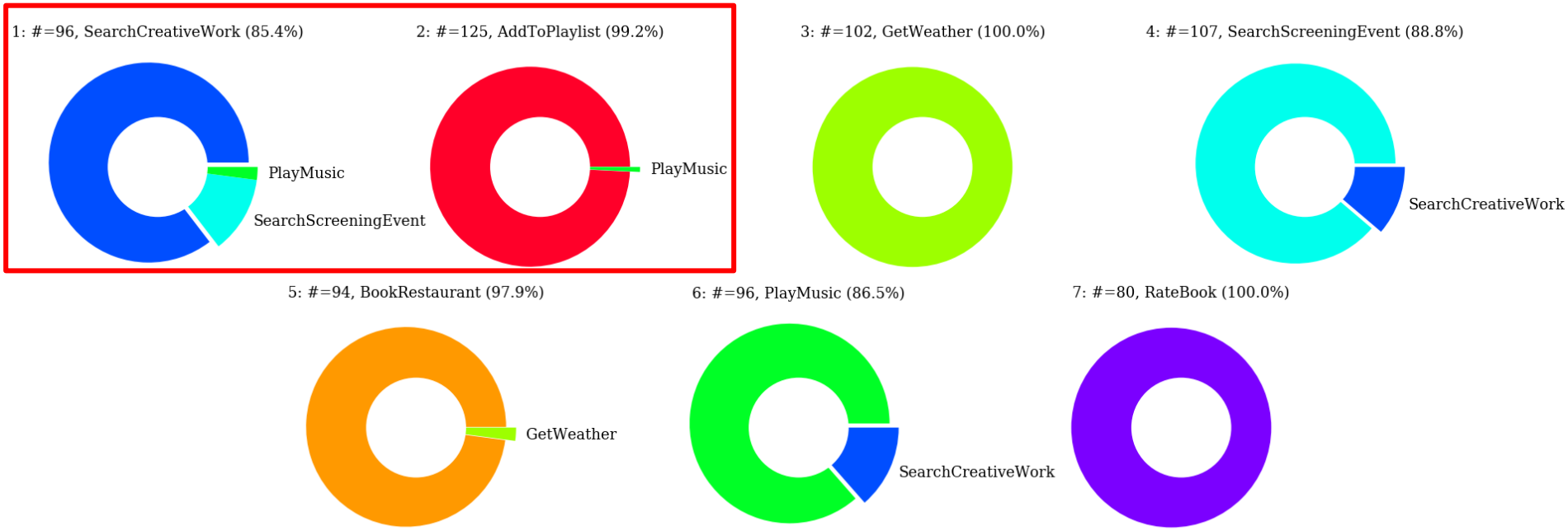






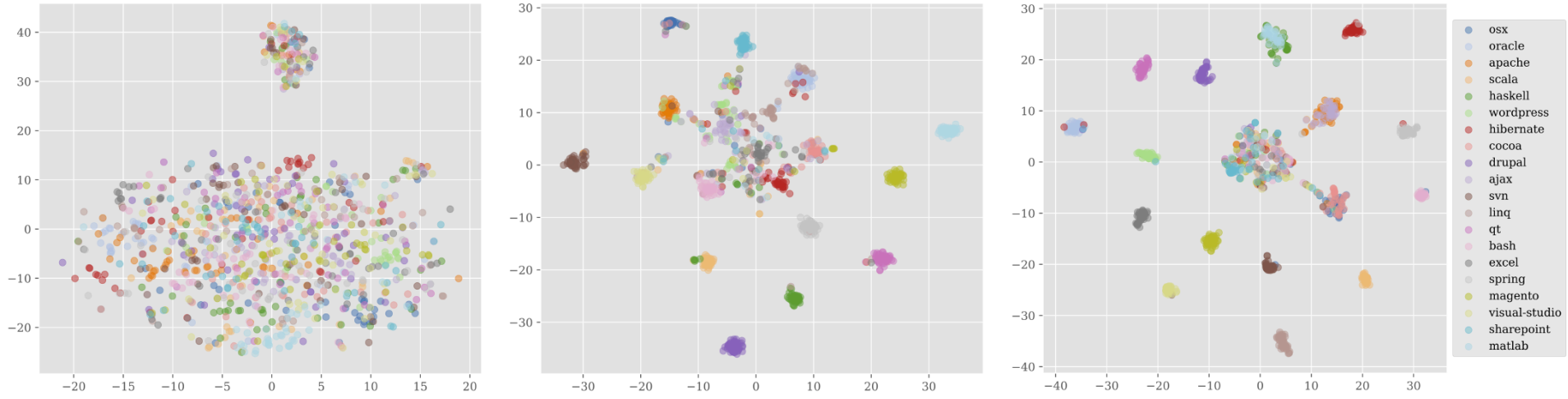
实验结果-案例分析(2)

- CDAC+能够发现潜在的新意图（红框）



实验结果-可视化

- CDAC+可以学到对聚类友好的特征表示 (t-SNE降维)
  - 透过加入标注数据作为先验知识&引导聚类，有效提升性能



(a) DAC (b) BERT-Semi (c) CDAC+

结论

- 我们提出了一个端到端的深度约束聚类算法CDAC+
- CDAC+能充分利用先验知识来引导聚类过程，发现新的类别
  - 透过预训练BERT、少量标注样本
- CDAC+透过聚类精炼，大幅提升了算法的鲁棒性
  - 对聚类中心数不敏感
- 未来展望：在聚类过程中融入更多先验知识作引导
  - 知识图谱、图卷积网络