# 广义零样本文本分类技术的研究与实现

**2019110739 章译文**

BUPT-CIST

指导老师：袁彩霞

**Oct 12, 2021**

# 目录

# 研究背景

# 研究背景

## 问题的提出

文本分类技术常用于问题分类、新闻归类、用户意图识别等应用，是自然语言处理领域重要的研究课题之一。

**Example (问题分类)**

在一个 FAQ 系统中，事先定义有一个标准问题集合，我们需要将对应的用户问题归类到一个标准问题中，从而给出对应的标准答案。

将标准问题视为类别，则问题分类为一个文本分类任务。

# 研究背景

## 问题的提出

在实践应用中，文本分类系统面临广义零样本（Generalized Zero-Shot Learning, GZSL）问题：

**(I)** 一方面，新类型的文本和新的业务投入导致新的未见类别不断涌现（emerge），而这些新类别没有任何标注样本，称为零样本问题。

**(II)** 另一方面，系统仍需要识别原有的类别，称为广义零样本问题。

# 研究背景

## 广义零样本分类问题

**例子：新闻文本分类**

- 训练时只有”医疗”、”教育”、”军事”、”科技”四个类别的训练样本和标签
- 测试时不仅有这四个类别，还需加入新的”育儿”共五个类别进行分类，并且其没有训练数据

1. 我们将训练时能获取的类别称为已见类别 seen class
2. 测试时才能获取的类别称为未见类别 unseen class

# 相关工作

# 相关工作

## 从任务角度和方法角度

任务角度 →

方法角度 ↓

| | 图像分类 | 意图识别 | 问题分类 |
|---|---|---|---|
| 向量嵌入方法 | 1. MMZS[1] 将图像和标签文本向量对齐<br>2. CRnet[2] 通过子网络组合缓解不平衡问题 | 1. ReCapsNet[3] 通过胶囊网络联系 query 和意图单词<br>2. RMG[4] 通过通用 KB 联系 query 和意图单词 | |
| 领域判别方法 | 1. OOD-GZSL 使用置信度判别可见和未见图像 | 1. SEG 使用 outlier detector 区别可见和未见意图 | |
| 生成方法 | 1. f-CLSWGAN[5] 通过 GAN 生成未见类别图像 | | 1. BDPG 使用标准问题生成未见虚拟样本 |
| 元学习方法 | 1. EPGN 通过元学习方法实现原型向量的生成<br>2. ZSML 通过元学习得到 GAN 用于生成未见类别图像 | | |

[1] Richard Socher et al. "Zero-shot learning through cross-modal transfer". In: *Advances in neural information processing systems* 26 (2013), pp. 935–943.

[2] Zhang et al. "Co-Representation Network for Generalized Zero-Shot Learning". In: *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019, pp. 7434–7443.

[3] Han Liu et al. "Reconstructing capsule networks for zero-shot intent classification". In: *Proceedings of the 2019 Conference on (EMNLP-IJCNLP)*. 2019, pp. 4801–4811.

[4] AB Siddique et al. "Generalized Zero-shot Intent Detection via Commonsense Knowledge". In: *arXiv preprint arXiv:2102.02925* (2021).

[5] Yongqin Xian et al. "Feature Generating Networks for Zero-Shot Learning.". In: *CVPR*. IEEE Computer Society, 2018, pp. 5542–5551.

# 相关工作

## 从任务角度和方法角度

任务角度 →

方法角度 ↓

| | 图像分类 | 意图识别 | 问题分类 |
|---|---|---|---|
| 向量嵌入方法 | 1. MMZS 将图像和标签文本向量对齐<br>2. CRnet 通过子网络组合缓解不平衡问题 | 1. ReCapsNet 通过胶囊网络联系 query 和意图单词<br>2. RMG 通过通用 KB 联系 query 和意图单词 | |
| 领域判别方法 | 1. OOD-GZSL[1] 使用置信度判别可见和未见图像 | 1. SEG[2] 使用 outlier detector 区别可见和未见意图 | |
| 生成方法 | 1. f-CLSWGAN 通过 GAN 生成未见类别图像 | | 1. BDPG[3] 使用标准问题生成未见虚拟样本 |
| 元学习方法 | 1. EPGN[4] 通过元学习方法实现原型向量的生成<br>2. ZSML[5] 通过元学习得到 GAN 用于生成未见类别图像 | | |

[1] Xingyu Chen et al. "A Boundary Based Out-of-Distribution Classifier for Generalized Zero-Shot Learning". In: *European Conference on Computer Vision*. Springer. 2020, pp. 572–588.

[2] Guangfeng Yan et al. "Unknown Intent Detection Using Gaussian Mixture Model with an Application to Zero-shot Intent Classification". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 1050–1060.

[3] Hao Fu et al. "Zero-Shot Question Classification Using Synthetic Samples". In: *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*. IEEE. 2018, pp. 714–718.

[4] Yunlong Yu et al. "Episode-Based Prototype Generating Network for Zero-Shot Learning". en. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 14032–14041.

[5] Vinay Kumar Verma, Dhanajit Brahma, and Piyush Rai. "Meta-Learning for Generalized Zero-Shot Learning.". In: *AAAI*. 2020, pp. 6062–6069.

# 相关工作

## 从任务设置角度

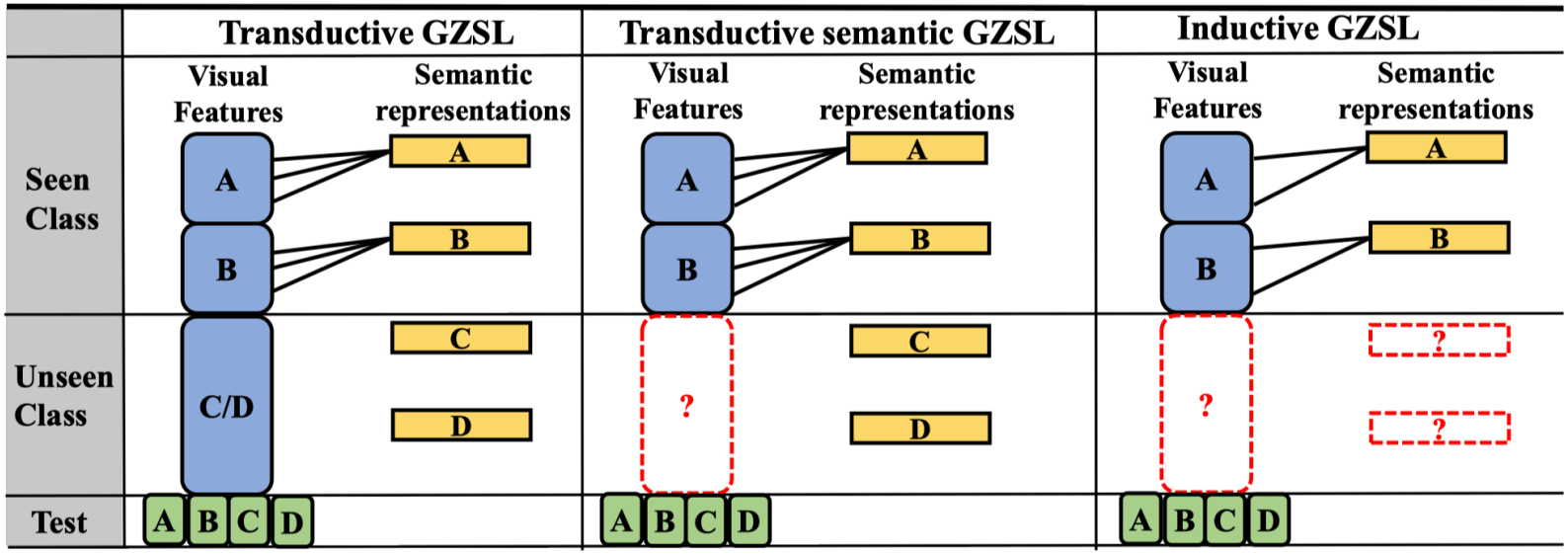



**图:** 三种不同设置的 GZSL 问题

- Transductive GZSL：训练时能获得未见标签信息和无标样本
- Transductive semantic GZSL：训练时能获得未见标签信息
- Inductive GZSL (ours)：训练时无任何未见类别信息，符合新的类别会不断出现的设置

# 相关工作

## 相关工作的对比

1. 相同任务设置的方法对比
   - 向量嵌入方法仅利用可见领域数据，未对未见类别数据进行适应，理论上达不到性能最优
   - 领域判别方法通过二分类器划分领域，同样未对未见类别进行自适应
2. 不同任务设置的方法对比
   - 生成方法需要利用未见标签生成虚拟样本，不属于 inductive 设置，无法在测试时实时响应
   - 元学习方法中，ZSML 结合生成方法，EPGN 结合向量嵌入方法，仍存在各自的局限性

**结论**：在测试过程中，模型需要具备对新类增量性的泛化能力，因此它需要动态地适应新类。

# 问题定义与前提

# 问题定义与前提

## 问题的数学定义

在 IGZSL 设置下，未见类别数目会不断涌现，是不固定的。假定对于一个具体的 GZSL 任务，形式化地有：

### GZSL 问题定义

$\mathcal{Y}^s = \{y_i^s, ..., y_{C^s}^s\}$，$\mathcal{Y}^u = \{y_i^u, ..., y_{C^u}^u\}$ 分别表示可见和未见类别，其中 $C^s$ 是可见类别个数，$C^u$ 是未见类别的数目。
可见类别和未见类别标签组成标签空间 $\mathcal{Y} = \mathcal{Y}^s \cup \mathcal{Y}^u$，并且有 $\mathcal{Y}^s \cap \mathcal{Y}^u = \emptyset$。定义可见领域数据为 $\mathcal{D}^s = \{(x_i^s, y_i^s, a_i^s) | x_i^s \in \mathcal{X}^s, y_i^s \in \mathcal{Y}^s, a_i^s \in \mathcal{A}^s\}$，未见领域的数据为 $\mathcal{D}^u = \{(x_i^u, y_i^u, a_i^u) | x_i^u \in \mathcal{X}^u, y_i^u \in \mathcal{Y}^u, a_i^u \in \mathcal{A}^u\}$，

- $x_i$ 代表文本句子
- $y_i$ 代表对应的 one-hot 标签
- $a_i$ 是 $y_i$ 对应的标签文本

对于可见领域的训练数据 $\mathcal{D}_{tr}^s$，测试时给定未见类别标签 $\mathcal{A}^u$，GZSL 需要对来自 $\mathcal{X}^s$ 或者 $\mathcal{X}^u$ 的样本进行识别。

# 问题定义与前提

## 前提

### 前提

1. 目标：对于 GZSL 任务中，最大后验估计得到数据 $\mathcal{D}^{s\cup u}$ 上的最优分类器参数 $\phi$：

$$\phi^{\star} = \arg\max_{\phi} \log p\left(\phi \mid \mathcal{D}^{s\cup u}\right) \tag{1}$$

2. $\mathcal{D}^{s\cup u}$ 不可获得：在 GZSL 设置下，训练阶段只能获得 $\mathcal{D}^{s}$，而测试阶段不能获得 $\mathcal{D}^{u}$，只有未见标签 $\mathcal{A}^{u}$，因此需要通过概率 $p(\phi \mid \mathcal{D}^{s}, \mathcal{A}^{u})$去预估最优参数：

$$\begin{aligned}\phi^{\star} &= \arg\max_{\phi} \log p\left(\phi \mid \mathcal{D}^{s\cup u}\right) \\ &\approx \arg\max_{\phi} \log p(\phi \mid \mathcal{D}^{s}, \mathcal{A}^{u})\end{aligned} \tag{2}$$

# 问题定义与前提

## 前提

### 前提

2. $\mathcal{D}^{s\cup u}$ 不可获得：在 GZSL 设置下，训练阶段只能获得 $\mathcal{D}^s$，而测试阶段不能获得 $\mathcal{D}^u$，只有未见标签 $\mathcal{A}^u$，因此需要通过概率 $p(\phi \mid \mathcal{D}^s, \mathcal{A}^u)$去预估最优参数：

$$\begin{aligned}\phi^\star &= \arg\max_{\phi} \log p\left(\phi \mid \mathcal{D}^{s\cup u}\right) \\ &\approx \arg\max_{\phi} \log p(\phi \mid \mathcal{D}^s, \mathcal{A}^u)\end{aligned} \tag{2}$$

3. $\mathcal{A}^u$ 不可获得：在 IGZSL 设置下，$\mathcal{A}^u$ 在训练阶段仍不能获得，无法得到 $p(\phi \mid \mathcal{D}^s, \mathcal{A}^u)$，并且 $\mathcal{A}^u$ 是任意的，所求从概率变为了概率分布 $p(\phi \mid \mathcal{D}^s, \mathcal{A})$：

$$\begin{aligned}&\arg\max_{\phi} \log p(\phi \mid \mathcal{D}^s, \mathcal{A}^u) \\ =&\arg\max_{\phi} \log p(\phi \mid \mathcal{D}^s, \mathcal{A} = \mathcal{A}^u)\end{aligned} \tag{3}$$

**难点**：学习分布 $p(\phi \mid \mathcal{D}^s, \mathcal{A})$

# 问题定义与前提

**难点**：学习分布 $p(\phi \mid \mathcal{D}^s, \mathcal{A})$

**解法**：受到元学习在少样本学习领域应用的启发，考虑通过构造元学习任务去学习这个分布。

1. 类比机器学习的 ERM (经验风险最小化) 原则，将一个任务视为一个样本（即元任务），使用经验分布预估真实分布
2. 精准估计得到 $\phi$ 较为困难，可以加入所有的元任务共享的起始参数 $\theta$（即元参数），降低预估的难度：$p(\phi \mid \mathcal{D}^s, \mathcal{A}, \theta)$
3. 由于只能获得 $\mathcal{D}^s$，则对 $\mathcal{D}^s$ 随机抽样元任务，而又要求 $\mathcal{D}^s$ 和 $\mathcal{A}$ 是标签无交集的 GZSL 任务，则元任务的 $\mathcal{D}$ 也是随机变量，即转化为学习分布 $p(\phi \mid \mathcal{D}, \mathcal{A})$

# 问题定义与前提

## 元任务

1. 构造元任务 $\mathcal{M}=\{\mathcal{D}_i,\mathcal{D}_i^*\}_{i=1}^N$：对已见类别数据 $\mathcal{D}^s$，每次将其随机划分为”虚拟”的可见类别 $\mathcal{Y}^{s_i}$ 和”虚拟”的未见类别 $\mathcal{Y}^{u_i}$，构成一个元任务 $\mathcal{T}_i\sim p(\mathcal{T})$，其中，$\mathcal{D}_i$ 是元训练集，$\mathcal{D}_i^*$ 是元测试集，且有 $\mathcal{Y}^s=\mathcal{Y}^{s_i}\cup\mathcal{Y}^{u_i}$，$\mathcal{Y}^{s_i}\cap\mathcal{Y}^{u_i}=\emptyset$。
   - 称”虚拟”可见类别为 memory class
   - 称”虚拟”未见类别为 novel class
2. 元任务训练过程：构造的元学习过程需要最小化：

$$-\frac{1}{N}\sum_i \mathbb{E}_{p(\theta|\mathcal{M})p(\phi|\mathcal{D}^{s_i},\mathcal{A}^{u_i},\theta)}\left[\frac{1}{K}\sum_{(x^*,y^*)\in\mathcal{D}_i^*}\log p\left(\hat{y}^*=y^*\mid x^*,\mathcal{A}^{s_i},\mathcal{A}^{u_i},\phi\right)\right] \tag{4}$$

   - $\theta$ 是所有元任务共有的分类器元参数 (meta parameters)
   - $\phi$ 是单个元任务特有的目标分类器参数 (zero-shot sensitive parameters)
   - $p\left(\phi\mid\mathcal{D}^{s_i},\mathcal{A}^{u_i},\theta\right)$ 是单个元任务的适应过程
   - $p\left(\hat{y}^*=y^*\mid x^*,\mathcal{A}^{s_i},\mathcal{A}^{u_i},\phi\right)$ 是分类器

# 问题定义与前提

## 元任务

2. 元任务训练过程：构造的元学习过程需要最小化：

$$-\frac{1}{N}\sum_{i}\mathbb{E}_{p(\theta|\mathcal{M})p(\phi|\mathcal{D}^{s_i},\mathcal{A}^{u_i},\theta)}\left[\frac{1}{K}\sum_{(x^*,y^*)\in\mathcal{D}_i^*}\log p\left(\hat{y}^*=y^*\mid x^*,\mathcal{A}^{s_i},\mathcal{A}^{u_i},\phi\right)\right] \tag{4}$$

   - $\theta$ 是所有元任务共有的分类器元参数 (meta parameters)
   - $\phi$ 是单个元任务特有的目标分类器参数 (zero-shot sensitive parameters)
   - $p\left(\phi\mid\mathcal{D}^{s_i},\mathcal{A}^{u_i},\theta\right)$ 是单个元任务的适应过程
   - $p\left(\hat{y}^*=y^*\mid x^*,\mathcal{A}^{s_i},\mathcal{A}^{u_i},\phi\right)$ 是分类器

3. 测试过程：根据大数定律，通过大量相似的元任务的学习，认为可以近似逼近总体分布 $p(\phi\mid\mathcal{D},\mathcal{A})$：

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i}p(\theta\mid\mathcal{M})p\left(\phi\mid\mathcal{D}^{s_i},\mathcal{A}^{u_i},\theta\right)\leftrightarrow p\left(\phi\mid\mathcal{D},\mathcal{A}\right) \tag{5}$$

测试时输入 $p(\phi\mid\mathcal{D}=\mathcal{D}^s,\mathcal{A}=\mathcal{A}^u,\theta)$预估目标参数 $\phi^*$，视为测试时的适应过程，满足实时的快速适应的要求。

# 问题定义与前提

## 总结

### 前提总结

1. 由于目标最优参数不可直接求得，转向估计 $p(\phi \mid \mathcal{D}^s, \mathcal{A}^u)$
2. $\mathcal{A}^u$ 训练时仍不可获得，使用元学习预估逼近总体分布 $p(\phi \mid \mathcal{D}, \mathcal{A})$
3. 测试时输入 $p(\phi \mid \mathcal{D} = \mathcal{D}^s, \mathcal{A} = \mathcal{A}^u, \theta)$，实现实时的快速适应，得到目标分类器参数

**设计目标**：如何设计 zero-shot sensitive 参数的生成函数 $p(\phi \mid \mathcal{D}, \mathcal{A}, \theta)$

1. $p(\phi \mid \mathcal{D}, \mathcal{A}, \theta)$：如何能够区分可见和未见类别
2. $p(\phi \mid \mathcal{D}, \mathcal{A}, \theta)$：如何设计 $\theta$ 和 $\phi$，即进行适应前和后的分类器
3. $p(\phi \mid \mathcal{D}, \mathcal{A}, \theta)$：如何定义生成函数

# 模型实现

# 模型实现

## 设计实现

### 设计目标

1. $p(\phi \mid \mathcal{D}, \mathcal{A}, \theta)$：如何能够区分可见和未见类别
2. $p(\phi \mid \mathcal{D}, \mathcal{A}, \theta)$：如何设计 $\theta$ 和 $\phi$，即进行适应前和后的分类器
3. $p(\phi \mid \mathcal{D}, \mathcal{A}, \theta)$：如何定义生成函数

### 设计实现

1. 设置一个可学习的向量表存储所有的可见类别，每个元任务的 memory class 信息从表中取出得到，novel class 信息用编码得到，简化了分布，将 $p\left(\phi \mid \mathcal{D}^{s_i}, \mathcal{A}^{u_i}, \theta\right)$转为求 $p\left(\phi \mid \mathcal{A}^{s_i}, \mathcal{A}^{u_i}, \theta\right)$，且不需要元训练集 $\mathcal{D}_i$ 。
2. 
   - 如果深度神经网络的参数规模较大，则为所有参数生成对应的 zero-sensitve 参数的计算复杂度较高，考虑只生成部分新参数 $\phi$，和原有的 $\theta$ 一起进行分类。
   - 分类时需要匹配标签向量和样本向量，如果要进行适应，需要对它们同时进行适应性更新。
3. 由于 $\mathcal{A}_u$可能是任意个数的类别，需要生成固定长度的 zero-sensitive 参数，考虑使用 self-attentive 机制，从多个固定角度组合类别集合信息。

# 模型实现

## GZSL: Baseline

**Input**

$T$个词的文本输入 $x$

**Encoder**

使用编码器 (*i.e.* Bi-LSTM, BERT) 得到隐向量序列 $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_T] \in \mathbb{R}^{T \times d_h}$。文本向量 $f(x)$ 通过 $T$ 个隐向量取平均得到。

**Classification**

Baseline 通过匹配函数为样本和对应标签学习一个共同空间：

$$\hat{y} = \arg\max_{y} s(f(x), f(a_y)) \tag{6}$$

其中 $s(\cdot, \cdot)$ 度量了 sample embedding $f(x)$ 和 prototype embedding $f(a_y)$ 的相似度 (*e.g.*, cosine similarity)。

分类时需要匹配标签向量和样本向量，需要对它们同时进行适应性更新：

1. prototype adaptation
2. sample adaptation

# 模型实现

## LTA: 总览

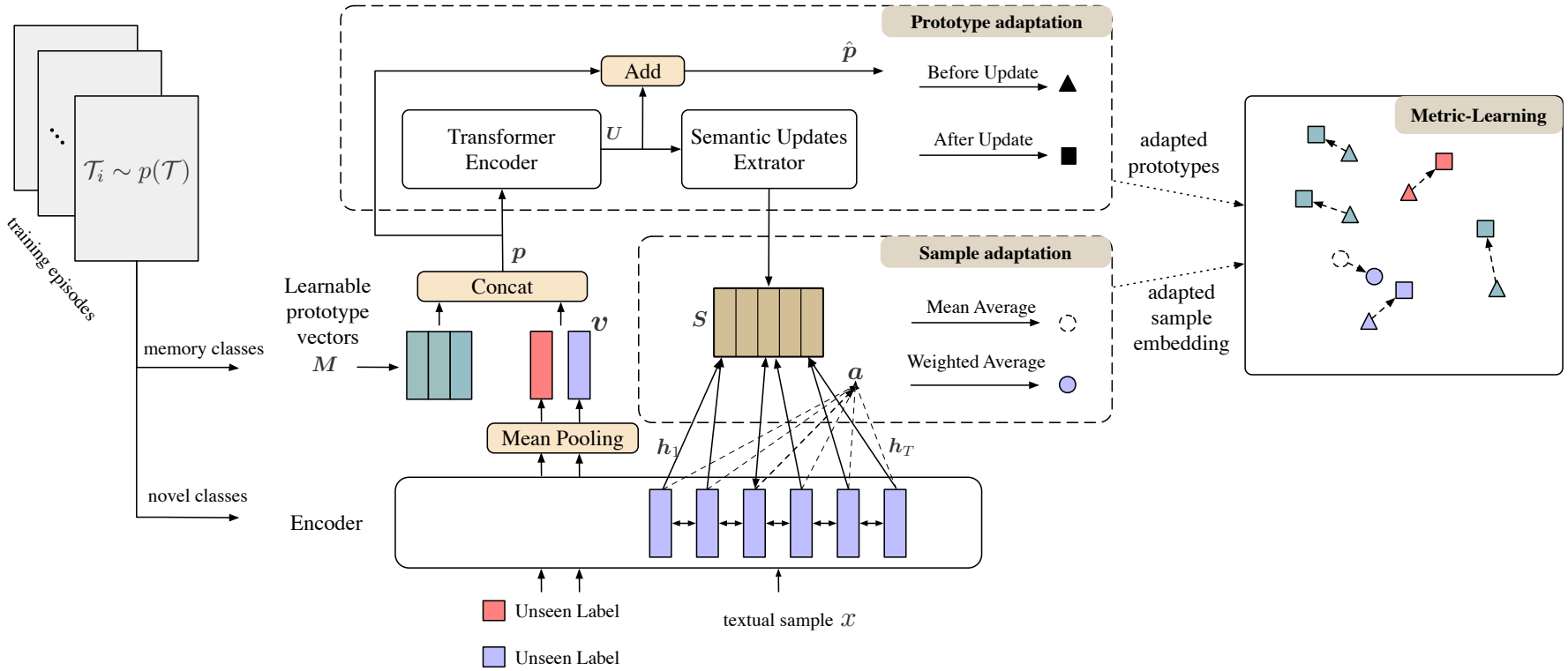



**图:** Illustration of our framework LTA. Green refers to seen classes and purple and red for unseen classes. ▲ denotes the raw prototypes before adaptation and ■ refers to the adapted prototypes ○ and ● respectively denote the example representations before and after sample adaptation.

# 模型实现

## LTA: 总览

Learning to Adapt (LTA)

1. Prototype Adapation：设置可学习矩阵 $M$存储可见 prototype，而未见 prototype 是在线编码得到的，然后将所有 prototype 输入一个 TransformerEncoder 中显式建模 prototype adaptation
2. Sample Adapation：使用上一步的 adaptation 信息生成 zero-shot sensitive 参数，用于校准 sample 词级别的隐向量权重取代平均来实现 sample adaptation
3. Metric Learning：对 adapted prototype embedding 和 adapted sample embedding 使用 metric learning 进行分类和学习

### Prototype Adaptation

对于 memory prototype 和 novel prototype，使用 self-attention 机制利用所有 prototype 之间的关联，得到更优的 prototype 表示

**(1)** 拼接 memory prototype 和 novel prototype，输入单层 TransformerEncoder 建模 prototype adaptation 的更新量

$$\boldsymbol{\Delta} = [\boldsymbol{\delta}_j]_{1..C^s} = \text{TransformerEncoder}([\boldsymbol{p}_j]_{1..C^s}) \tag{7}$$

**(2)** 原 prototype 加上更新量，得到更新后的 prototype

$$\hat{\boldsymbol{P}} = [\hat{\boldsymbol{p}}_1, \hat{\boldsymbol{p}}_2, ..., \hat{\boldsymbol{p}}_{\hat{C^s}}] = [\boldsymbol{p}_1 + \boldsymbol{\delta}_1, \boldsymbol{p}_2 + \boldsymbol{\delta}_2, ..., \boldsymbol{p}_{C^s} + \boldsymbol{\delta}_{C^s}] \tag{8}$$

# 模型实现

## LTA: Sample Adaptation

### Sample Adaptation

利用上一步 prototype adaptation 生成 sample 侧的 zero-shot sensitive 参数，用以校准词级别隐向量的权重而不是取平均来实现 sample adaptation。

**Motivation: Semantic Loss**

一些对识别未见类别重要，而对于已见类别不具有判别性的信号可能会在训练时被丢弃。利用调整词隐向量权重，希望能够重新激活这些判别性信号。

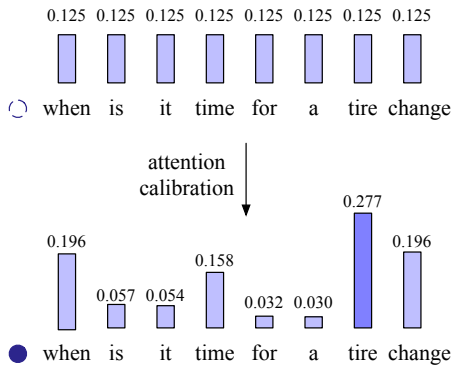



**图:** re-weight for sample adaptation

## Sample Adaptation

**(1)** 通过 self-attentive 使用 prototype adaptation 的更新量生成语义矩阵

$$\boldsymbol{F} = \boldsymbol{\Delta W}_1 \tag{9}$$

$$\boldsymbol{A} = \text{Softmax}(\boldsymbol{W}_3 \text{ReLU}(\boldsymbol{W}_2 \boldsymbol{F}^T)) \tag{10}$$

$$\boldsymbol{S} = \boldsymbol{AF} = [\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_{d_r}] \tag{11}$$

**(2)** 使用最相似的语义向量作为补充语义损失的方式，值越大代表和发掘的补充语义越相似，权重越大

$$a_i = \text{Softmax}(\beta \max_j (\frac{\boldsymbol{h}_i \boldsymbol{s}_j}{||\boldsymbol{h}_i|| ||\boldsymbol{s}_j||})) \tag{12}$$

$$f^{'}(x) = \sum_{i=1}^{T} a_i \boldsymbol{h}_i \tag{13}$$

## Metric Learning and training

**(1)** 对比 Baseline，使用 adapt 之后的 prototype $\hat{P}$ 和 adapt 之后的 sample $f^{'}(x)$ 进行分类

$$logp\left(\hat{y}=y \mid x, \mathcal{A}^{s_i}, \mathcal{A}^{u_i}, \phi, \theta\right)=log\frac{\exp(s(f^{'}(x), \hat{\boldsymbol{p}}_y))}{\sum_{\hat{y}}\exp(s(f^{'}(x), \hat{\boldsymbol{p}}_{\hat{y}}))} \tag{15}$$

**(2)** 以 Eq 4的形式，训练时最小化损失函数：

$$-\frac{1}{N}\sum_{i}\mathbb{E}_{p(\theta|\mathcal{M})p(\phi|\mathcal{A}^{s_i},\mathcal{A}^{u_i},\theta)}\left[\frac{1}{K}\sum_{(x^*,y^*)\in\mathcal{D}_i^*}\log p\left(\hat{y}^*=y^* \mid x^*, \mathcal{A}^{s_i}, \mathcal{A}^{u_i}, \phi, \theta\right)\right] \tag{16}$$

# 实验结果与分析

# 实验结果与分析

## 数据集

4 个意图识别数据集

(1) **SNIPS-SLU**：意图识别英文数据集，划分时选择 5 个已知意图，2 个未知意图

(2) **SMP-18**：意图识别中文数据集，划分时选择 24 个已知意图，6 个未知意图

(3) **ATIS**：关于航旅的意图识别英文数据集. 提取了 17 个意图，其中每个意图至少包含 5 条样本, 划分时选择 12 个已知意图，5 个未知意图。此数据集存在不平衡问题，其中 *flight* 意图占了训练数据 87% 左右

(4) **CLINC**：近期的意图识别英文数据集，包含覆盖 10 个领域 150 意图类别的 22,500 条 queries，划分时选择 120 个已知意图，30 个未知意图

1 个问题分类数据集

(1) **Quora**：从 Quora 问题匹配数据集构造而来，包含 1700 个标准问题，其中每个标准问题至少包含 5 条样本, 划分时选择 1360 个已知标准问题，340 个未知标准问题

# 实验结果与分析

## 数据集划分

1. 为了更加置信的结果，随机划分可见/未见类别，作为 10 组实验，而不是像以往工作的固定划分。
2. 对可见类别随机选取 70% 的样本作为训练集，可见类别剩余的 30% 样本和所有的未见类别样本作为测试集。

| Dataset | #classes | | #samples | | sent len | type |
|---|---|---|---|---|---|---|
| | seen | unseen | total | avg | | |
| SNIPS | 5 | 2 | 13802 | 1384 | 9.10 | BAL |
| SMP | 24 | 6 | 2460 | 60 | 4.83 | FS |
| ATIS | 12 | 5 | 4972 | 245 | 11.44 | IBAL |
| Clinc | 120 | 30 | 22500 | 105 | 8.23 | BAL |
| Quora | 1360 | 340 | 17394 | 7 | 10.46 | FS |

**表:** Dataset statistics. “FS” indicates “few-shot”, “BAL” indicates “balance”, “IBAL” indicates “imbalance”. The “avg #samples” indicates the average number of samples per class.

# 实验结果与分析

## 实验设置

实验设置：

- 使用 BiLSTM 和 Bert 作为基础编码器
- Adam 优化器 + 基础学习率 0.001，编码器参数学习率为 0.0001
- 设置元测试集为分别从 memory 和 novel 抽样 (N-way K-shot) 组合得到 batch
- 使用 metric learning 预训练得到 $M$ 的初始化

# 实验结果与分析

## 实验结果

| Model | SNIPS-NLU | | | | | | SMP-18 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Seen | | Unseen | | HM | | Seen | | Unseen | | HM | |
| | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 |
| Bi-LSTM | 98.23 | 98.23 | 0.00 | 0.00 | 0.00 | 0.00 | 93.65 | 93.43 | 0.00 | 0.00 | 0.00 | 0.00 |
| EucSoftmax | 88.95 | **89.28** | 71.58 | 71.48 | 79.33 | 79.39 | 80.81 | 82.53 | 65.48 | 55.68 | 72.34 | 66.49 |
| EucTriplet | 89.29 | 87.73 | 70.18 | 71.29 | 78.59 | 78.66 | 80.48 | 84.89 | 72.59 | 58.74 | 76.33 | 69.44 |
| CosT | 96.23 | 80.15 | 56.41 | 67.38 | 71.13 | 73.21 | **87.58** | 79.21 | 42.39 | 43.11 | 57.13 | 55.84 |
| ReCapsNet | **96.26** | 67.70 | 11.57 | 18.45 | 20.66 | 29.00 | 76.32 | 74.92 | 20.56 | 15.09 | 32.39 | 25.10 |
| + SEG | 92.11 | 73.08 | 50.29 | 62.33 | 65.06 | 67.28 | 67.10 | 67.39 | 36.65 | 32.84 | 47.70 | 44.16 |
| LTA (Ours) | 88.14 | 88.20 | **82.40** | 82.24 | **85.17** | **85.12** | 82.90 | **86.07** | **80.20** | **72.34** | **81.53** | **78.61** |
| w / o Init | 85.04 | 85.62 | 78.28 | 77.40 | 81.52 | 81.30 | 75.65 | 79.84 | 75.13 | 61.69 | 75.39 | 69.60 |
| w / o SA | 90.11 | 84.10 | 77.36 | **82.61** | 83.25 | 83.35 | 77.90 | 80.12 | 76.40 | 64.76 | 77.14 | 71.62 |
| w / o A | 88.04 | 84.75 | 75.06 | 77.89 | 81.03 | 81.17 | 80.65 | 84.85 | 75.89 | 69.20 | 78.19 | 76.23 |

| Model | ATIS | | | | | | CLINC | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Seen | | Unseen | | HM | | Seen | | Unseen | | HM | |
| | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 |
| Bi-LSTM | 93.24 | 79.51 | 0.00 | 0.00 | 0.00 | 0.00 | 92.07 | 92.06 | 0.00 | 0.00 | 0.00 | 0.00 |
| EucSoftmax | 80.32 | 39.24 | 20.18 | 24.39 | 32.26 | 30.09 | 83.83 | 75.26 | 26.42 | 28.49 | 40.18 | 41.33 |
| EucTriplet | 90.67 | 52.49 | 24.01 | 21.81 | 37.97 | 30.82 | 84.28 | 81.00 | 35.22 | 36.69 | 49.68 | 50.50 |
| CosT | 86.88 | 46.13 | 29.42 | 30.07 | 43.95 | 36.41 | **91.98** | 77.69 | 36.84 | 46.35 | 52.61 | 58.06 |
| ReCapsNet | 86.19 | 23.88 | 12.80 | 4.89 | 22.32 | 8.12 | 88.53 | 69.83 | 4.24 | 3.33 | 8.10 | 6.36 |
| + SEG | **93.75** | 40.90 | 14.78 | 6.36 | 25.53 | 11.01 | 81.04 | 78.89 | 9.07 | 5.44 | 16.31 | 10.18 |
| LTA (Ours) | 86.89 | 52.58 | **37.99** | 41.96 | **52.87** | 46.68 | 89.67 | 82.11 | **61.69** | **66.48** | **73.09** | **73.47** |
| w / o Init | 84.98 | **62.98** | 37.07 | 37.89 | 51.62 | 47.29 | 88.22 | 81.53 | 58.24 | 62.01 | 70.17 | 70.44 |
| w / o SA | 93.44 | 58.49 | 34.17 | **43.95** | 50.04 | **50.19** | 89.56 | 80.36 | 52.73 | 59.31 | 66.38 | 68.25 |
| w / o A | 84.51 | 56.79 | 33.11 | 37.42 | 47.58 | 45.12 | 91.19 | **83.01** | 54.09 | 60.44 | 67.90 | 69.95 |

**表:** Results (in %) of LSTM-based models on four intent benchmarks.

# 实验结果与分析

## 实验结果

| Model | SNIPS-NLU | | | | | | SMP-18 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Seen | | Unseen | | HM | | Seen | | Unseen | | HM | |
| | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 |
| BERT | 98.91 | 98.91 | 0.00 | 0.00 | 0.00 | 0.00 | 95.28 | 94.87 | 0.00 | 0.00 | 0.00 | 0.00 |
| EucSoftmax | 81.09 | 65.50 | 45.89 | 58.21 | 58.61 | 61.64 | 89.84 | 87.85 | 76.65 | 77.51 | 82.72 | 82.36 |
| EucTriplet | 81.09 | 65.28 | 45.91 | 58.53 | 58.63 | 61.72 | **90.97** | 87.67 | 75.38 | 77.32 | 82.44 | 82.17 |
| CosT | **91.68** | **75.76** | 47.73 | 62.84 | 62.77 | 68.70 | 90.65 | 88.41 | 72.59 | 73.89 | 80.62 | 80.50 |
| LTA (Ours) | 74.05 | 74.11 | **90.09** | **84.22** | **81.28** | **78.84** | 89.84 | **90.79** | 79.19 | 75.20 | 84.18 | 82.26 |
| w / o Init | 82.57 | 75.22 | 64.36 | 71.63 | 72.34 | 73.87 | 89.03 | 87.23 | **80.71** | **81.74** | **84.67** | **84.40** |
| w / o SA | 67.31 | 70.56 | 84.70 | 77.51 | 75.01 | 73.87 | 84.52 | 81.40 | 75.89 | 74.40 | 79.97 | 77.75 |
| w / o A | 75.26 | 71.82 | 83.85 | 80.77 | 79.33 | 76.03 | 84.35 | 86.93 | 76.90 | 73.54 | 80.72 | 80.50 |

| Model | ATIS | | | | | | CLINC | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Seen | | Unseen | | HM | | Seen | | Unseen | | HM | |
| | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 |
| BERT | 97.18 | 93.71 | 0.00 | 0.00 | 0.00 | 0.00 | 97.37 | 97.37 | 0.00 | 0.00 | 0.00 | 0.00 |
| EucSoftmax | 67.67 | 16.11 | 7.78 | 5.50 | 13.96 | 8.20 | 96.02 | 87.07 | 58.02 | 66.00 | 72.33 | 75.08 |
| EucTriplet | 63.56 | 23.12 | 8.05 | 12.02 | 14.29 | 15.82 | 95.31 | 86.65 | 58.49 | 65.89 | 72.49 | 74.68 |
| CosT | **98.02** | 59.55 | 46.04 | 45.21 | 62.66 | 51.40 | **96.31** | 87.33 | 62.73 | 70.28 | 75.98 | 77.89 |
| LTA (Ours) | 96.28 | 63.13 | 66.09 | **55.02** | 78.38 | **58.80** | 92.22 | 87.57 | 73.18 | 75.74 | 81.60 | 81.23 |
| w / o Init | 89.96 | 47.48 | **69.79** | 52.14 | **78.60** | 49.70 | 93.07 | **88.19** | **73.80** | **77.54** | **82.32** | **82.52** |
| w / o SA | 90.20 | 51.74 | 66.23 | 47.24 | 76.38 | 49.38 | 92.46 | 87.30 | 69.27 | 73.26 | 79.20 | 79.67 |
| w / o A | 94.94 | **63.25** | 57.52 | 49.19 | 71.64 | 55.34 | 93.81 | 88.12 | 70.11 | 74.58 | 80.25 | 80.79 |

**表:** Results (in %) of BERT-based models on four intent benchmarks.

# 实验结果与分析

## 实验结果

| Model | Seen | | Unseen | | HM | |
|---|---|---|---|---|---|---|
| | Acc | F1 | Acc | F1 | Acc | F1 |
| BiLSTM | 71.70 | 69.04 | 0.00 | 0.00 | 0.00 | 0.00 |
| EucSoftmax | 79.88 | 74.42 | 56.85 | 62.39 | 66.43 | 67.88 |
| EucTriplet | 72.52 | 67.42 | 48.68 | 53.27 | 58.26 | 59.52 |
| CosT | **88.50** | 81.39 | 62.21 | 73.55 | 73.06 | 77.27 |
| LTA (Ours) | 84.69 | **83.56** | 74.83 | **76.93** | **79.45** | **80.11** |
| w / o Init | 82.11 | 81.99 | **75.49** | 76.53 | 78.66 | 79.17 |
| w / o SA | 84.95 | 82.79 | 73.56 | 76.67 | 78.84 | 79.62 |
| w / o A | 84.21 | 82.40 | 72.50 | 75.23 | 77.92 | 78.65 |

**表:** Results (in %) of LSTM-based models on Quora.

结论:

- 在多个数据集和任务上验证了 LTA 的有效性
- 在可见领域性能下降很小的情况下，大幅度提升了未见领域的性能
- 消融实验表明结合使用 Prototype Adapation 和 Sample Adapation 能够明显提升性能

# 实验结果与分析

## 领域偏差结果分析

(1) 领域偏差问题

为了进一步评估性能，设置领域分类的二分类任务，判别样本是来自可见领域还是未见领域：

| Model | CLINC | | | Quora | | |
|---|---|---|---|---|---|---|
| | Seen DR | Unseen DR | HM DR | Seen DR | Unseen DR | HM DR |
| BiLSTM | 100.00 | 0.00 | 0.00 | 100.00 | 0.00 | 0.00 |
| EucSoftmax | 90.43 | 55.29 | 68.62 | 91.57 | 74.70 | 82.28 |
| EucTriplet | 85.76 | 64.11 | 73.37 | 88.16 | 71.95 | 79.23 |
| CosT | **98.17** | 42.91 | 59.72 | **98.07** | 68.23 | 80.47 |
| LTA (Ours) | 94.43 | **70.09** | **80.46** | 91.26 | **86.81** | **88.98** |
| w / o Init | 93.41 | 70.84 | 80.57 | 92.24 | 85.39 | 88.68 |
| w / o SA | 95.31 | 61.44 | 74.71 | 89.07 | 86.72 | 87.88 |
| w / o A | 95.83 | 62.64 | 75.76 | 91.64 | 85.26 | 88.34 |

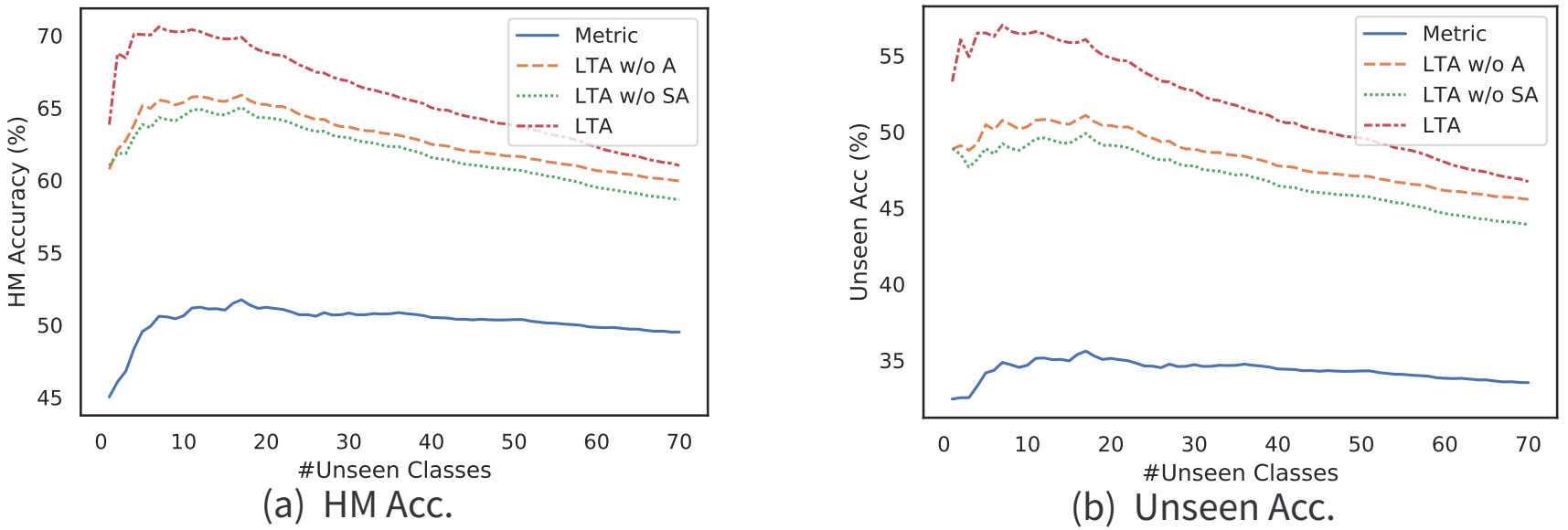
**表:** Results (in %) of LSTM-based models on CLINC and Quora for domain classification.

结论：LTA 在少量可见领域性能损失的情况下，对未见样本的识别能力显著增强，从而减小将未见样本误分类到可见领域的偏差。

# 实验结果与分析

## 不同未见类别数目的实验结果分析

（2）之前的评估方式中，未见类别数目是固定的，难以体现目标中” 不断涌现” 新类别的情况。针对这一问题，设计不同未见类别数目的实验：




(a) HM Acc.　　(b) Unseen Acc.

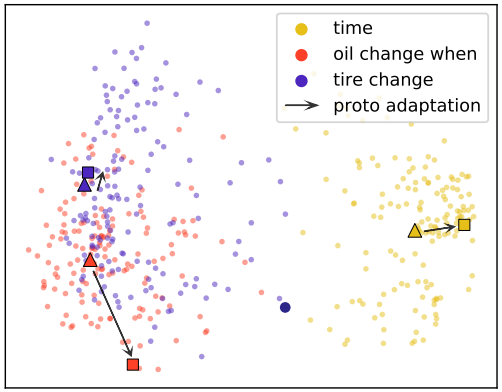
**图:** The performance with different numbers of unseen classes on CLINC dataset. Specifically, we select 70 classes as seen classes and 10 classes as validating unseen classes. The testing unseen classes are randomly sampled from the remaining 70 classes and each experiment is repeated 50 times with different sampling sets for a more stable result.

结论：不仅对于固定的未见类别数目，LTA 在不同未见类别数目的情况下都超过 Baseline，表明 LTA 能够对未见类别进行快速适应，能够处理不同规模的未见类别。

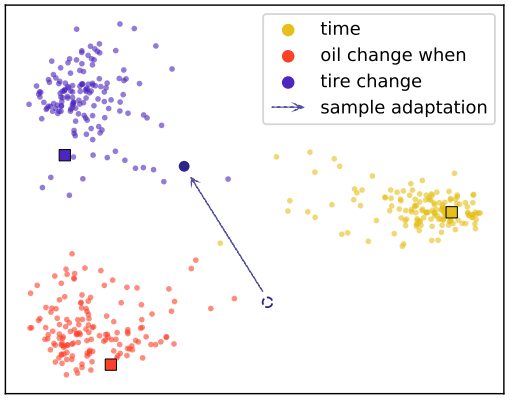
# 实验结果与分析

## 可视化结果分析

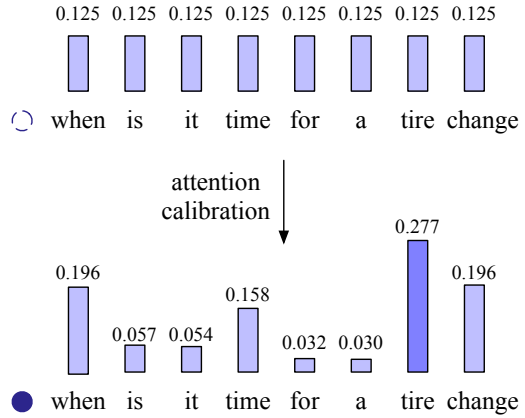
(3) 为了进一步阐述模型工作方式，进行了可视化分析 Prototype Adapation 和 Sample Adaptation




(a) LTA w/o sample adaption




(b) LTA




(c) example

**Figure:** PCA plots of encoded unseen sample representations(·) and prototype representations(■) from (a) LTA w/o sample adaption model and (b) full LTA model, where "time" and "oil change when" are two seen classes and "tire change" is an unseen class. (c) is an unseen example with sample-level raw attention and adapted attention. ▲denotes the raw prototype representations before adaptation. ○ and ● respectively denote the example representations before and after sample adaptation.

结论：LTA 通过实现 prototype adaptation 和 sample adaptation，有效补充了样本中的判别性信号，增加了其判别性使其分类正确

# 方法二

## 另一种实现：梯度方法

设计目标：$p(\phi \mid \mathcal{D}, \mathcal{A}, \theta)$ 存在的问题

1. **初始化影响较大**：由于设置了可学习原型矩阵 $M$，发现不同随机种子对性能结果的影响方差很大
2. **可解释性较差**：通过黑盒的方式推理得到新的模型参数，对于参数生成器的可控性较差

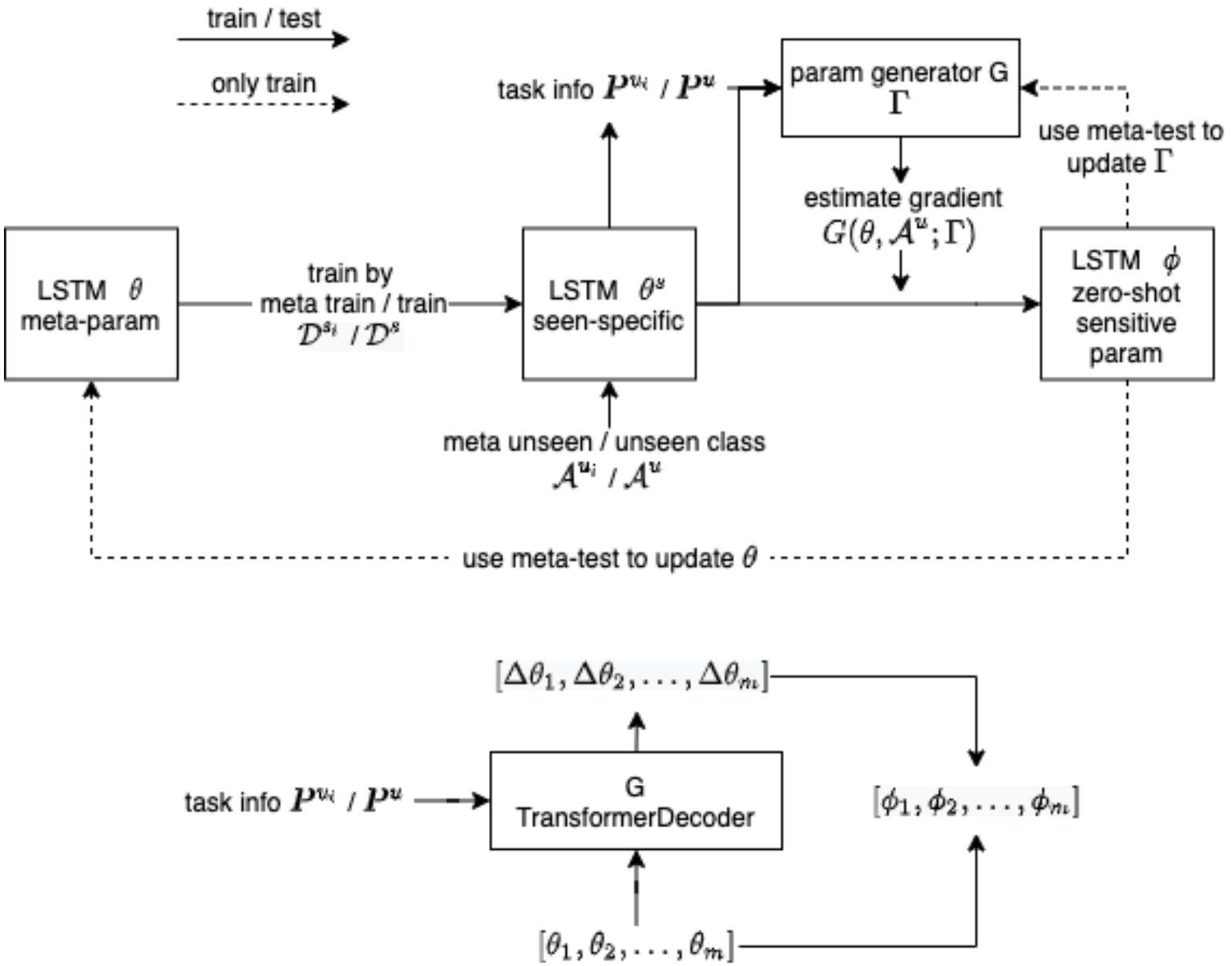
解法：

1. **初始化影响较大**：抛弃可学习矩阵 $M$，使用 $\mathcal{D}^s$ 先进行训练得到 seen-specific 参数 $\theta^{s*}$，再通过点估计得到 $\phi$
2. **可解释性较差**：对于上述点估计的方法，通过点估计梯度信息来实现 $p(\phi \mid \theta, \mathcal{A}^u)$

$$\phi = \theta^s - \eta G(\theta, \mathcal{A}^u; \Gamma)$$

# 方法二

## 流程图


train / test
only train
task info $P^{u_i}$ / $P^u$
param generator G
$\Gamma$
use meta-test to update $\Gamma$
estimate gradient
$G(\theta, \mathcal{A}^u; \Gamma)$
LSTM $\theta$
meta-param
train by
meta train / train
$\mathcal{D}^{s_i}$ / $\mathcal{D}^s$
LSTM $\theta^s$
seen-specific
LSTM $\phi$
zero-shot
sensitive
param
meta unseen / unseen class
$\mathcal{A}^{u_i}$ / $\mathcal{A}^u$
use meta-test to update $\theta$
$[\Delta\theta_1, \Delta\theta_2, \ldots, \Delta\theta_{n_i}]$
task info $P^{u_i}$ / $P^u$
G
TransformerDecoder
$[\phi_1, \phi_2, \ldots, \phi_{n_i}]$
$[\theta_1, \theta_2, \ldots, \theta_m]$

# 方法二

## 算法伪代码

**Algorithm 1:** Gradient-Based Method for GZSL.

**Input:** distribution over tasks $p(\mathcal{T})$, class set $\mathcal{Y}^s$
**Output:** encoder parameters $\theta$, gradient estimator parameters $\Gamma$
**while** *not done* **do**

**Step1: Sample Tasks**

Randomly sample meta GZSL tasks $\mathcal{T}_i \sim p(\mathcal{T})$ with two $N$-way $K$-shot meta-train $\mathcal{D}_i$ and meta-test $\mathcal{D}_i^*$, the class sets of them are $\mathcal{Y}_i = \mathcal{Y}^{s_i}$ and $\mathcal{Y}_i^* = \mathcal{Y}^{s_i} \cup \mathcal{Y}^{u_i} = \mathcal{Y}^s$, respectively, where $\mathcal{Y}_{s_i} \cap \mathcal{Y}_{u_i} = \emptyset$.

**Step2: Gradient Estimate**

**for *all*** $\mathcal{T}_i$ **do**

Evaluate $\nabla_\theta \mathcal{L}^C_{\mathcal{T}_i}(\theta)$ using meta-trains

Compute seen-specific parameters: $\theta^{s_i} = \theta - \beta \nabla_\theta \mathcal{L}^C_{\mathcal{D}_i}(\theta)$

Estimate target parameters using set2set gradient estimator: $\phi^{\mathcal{T}_i} = \theta^{s_i} - \eta_{\mathcal{T}_i} G(\theta^{s_i}, \mathcal{A}^{u_i}; \Gamma)$

**end**

**Step3: Optimization**

Update $\theta = \theta - \beta_\theta \nabla_\theta \sum_{\mathcal{D}_i^*} \mathcal{L}^C_{\mathcal{D}_i^*}(\phi^{\mathcal{T}_i})$ using meta-tests

Update $\Gamma = \Gamma - \beta_\Gamma \nabla_\Gamma \sum_{\mathcal{D}_i^*} \mathcal{L}^C_{\mathcal{D}_i^*}(\phi^{\mathcal{T}_i})$ using meta-tests

**end**

# 方法二

## 梯度估计方法

| Model | Seen | | Unseen | | HM | |
|---|---|---|---|---|---|---|
| | Acc | F1 | Acc | F1 | Acc | F1 |
| CosT | 91.98 | 77.69 | 36.84 | 46.35 | 52.61 | 58.06 |
| LTA | 89.67 | 82.11 | 61.69 | 66.48 | 73.09 | 73.47 |
| grad - 0.1 | 95.81 | 87.32 | 50.52 | 56.73 | 65.51 | 68.14 |
| grad - reset3 | 92.44 | 86.94 | 52.30 | 56.99 | 66.44 | 68.62 |
| grad - sum | 92.43 | 85.70 | 53.70 | 57.82 | 67.25 | 69.02 |

**表:** Results (in %) of LSTM-based models on CLINC.

遇到的困难:

- 相比于原有 MAML，除了需要更新元参数 $\theta$，还需要更新梯度估计器 $\Gamma$，多任务学习造成收敛困难
- 使用 seen 数据先训练的方式会加剧 bias 问题，难以解决。这说明之前方法的收益可能基本来自于两种不同的编码方式用以区分可见和未见
- 训练十分缓慢，在需要大量元任务的要求下迭代困难

# 总结与计划

# 总结

1. 建模广义零样本学习问题，为其提供一种新的解法
2. 针对广义零样本文本分类任务，提供数据集及其设置，可作为 benchmark 使用
3. 在广义零样本文本分类任务上，提出基于元学习框架的适应模型 LTA

# 中期计划

| 时间 | 研究内容 | 效果 |
| --- | --- | --- |
| 2020/12-2021/02 | 实现基于元学习的广义零样本文本分类算法，并进行领域自适应模型的初步探究 | 完成代码工作，得到初步实验结果 |
| 2021/02-2021/07 | 优化设计实现领域自适应的算法模型 | 以部分实验成果撰写小论文，于某会议收录 |
| 2021/08-2021/09 | 完成基于梯度方法的调研与实验 | 完成代码工作，得到初步实验结果 |
| 2021/10-2021/11 | 撰写毕业论文及小论文 | 完成论文 |
| 2021/11- | 优化梯度方法、实现广义零样本问题分类系统 | 预期实现一份专利，完成系统 |


BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS
北京邮电大学

# Thanks