Few-shot text classification

Introduction

Few-shot Learning 是 Meta Learning 在监督学习领域的应用。在 meta training 阶段将数据集分解为不同的 meta episode，去学习类别变化的情况下模型的泛化能力，在 meta testing 阶段，面对全新的类别，不需要变动已有的模型，就可以完成分类。

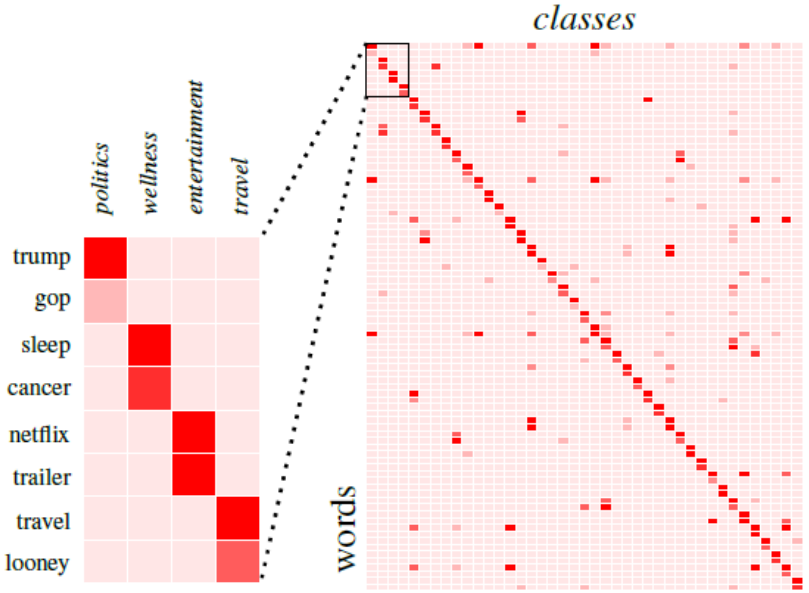
形式化来说，few-shot 的训练集中包含了很多的类别，每个类别中有多个样本。在训练阶段，会在训练集中随机抽取 N个类别，每个类别 K 个样本 (总共 NK 个数据), 作为每个episode的支撑集（support set）输入;再从这 N个类中剩余的数据中抽取一批样本作为模型的预测对象(Query Set)。即要求模型从 N*K 个数据中学会如何区分这 N 个类别，这样的任务被称为 **N-way K-shot** 问题。

FEW-SHOT TEXT CLASSIFICATION WITH DISTRIBUTIONAL SIGNATURES

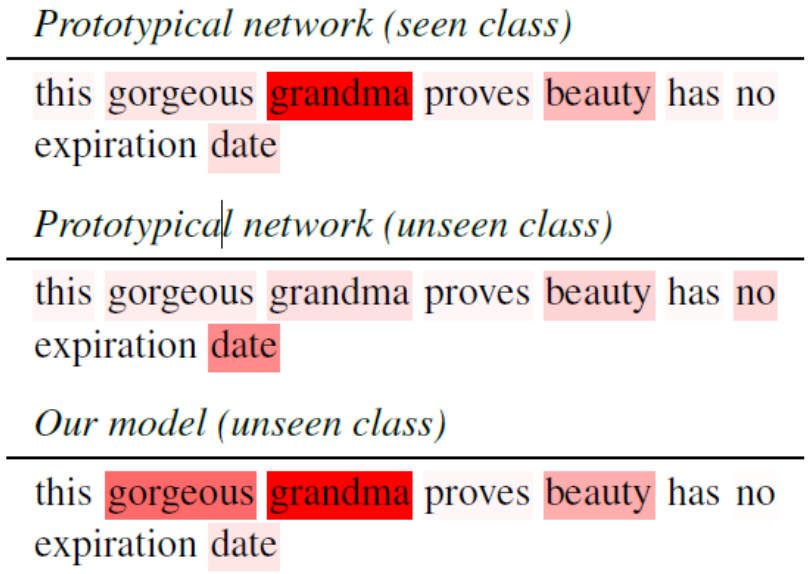
Yujia Bao[†*], **Menghua Wu**[†*], **Shiyu Chang**[‡], **Regina Barzilay**[†]
[†]Computer Science and Artificial Intelligence Lab, MIT
[‡]MIT-IBM Watson AI LAB, IBM Research
{yujia,rmwu,regina}@csail.mit.edu, {shiyu.chang}@ibm.com

Introduction

目前元学习主要在CV上取得了较大成功，但是在NLP领域还未有广泛的成功。元学习不易处理NLP任务的主要原因还是在于：CV的低级别模型，如边、颜色、点在不同任务上都是通用的；而NLP的不同任务，即使是同一个句子，其重点都是不同的。

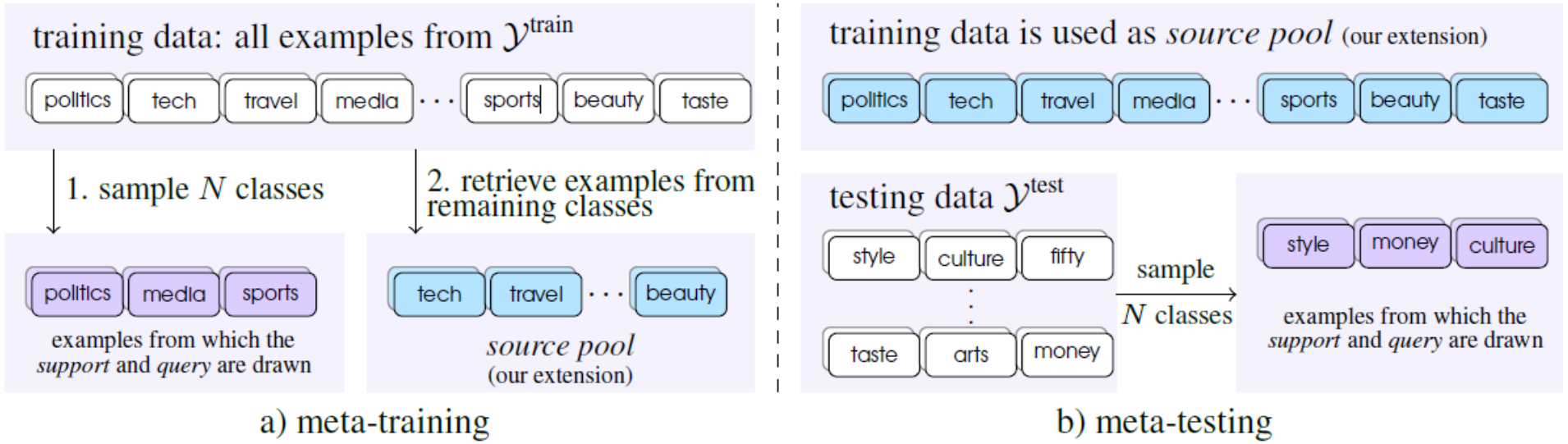


HuffPost 数据集中共41个类别不同词的相关度



HuffPost 数据集中对于类别fifty每个单词的重要性

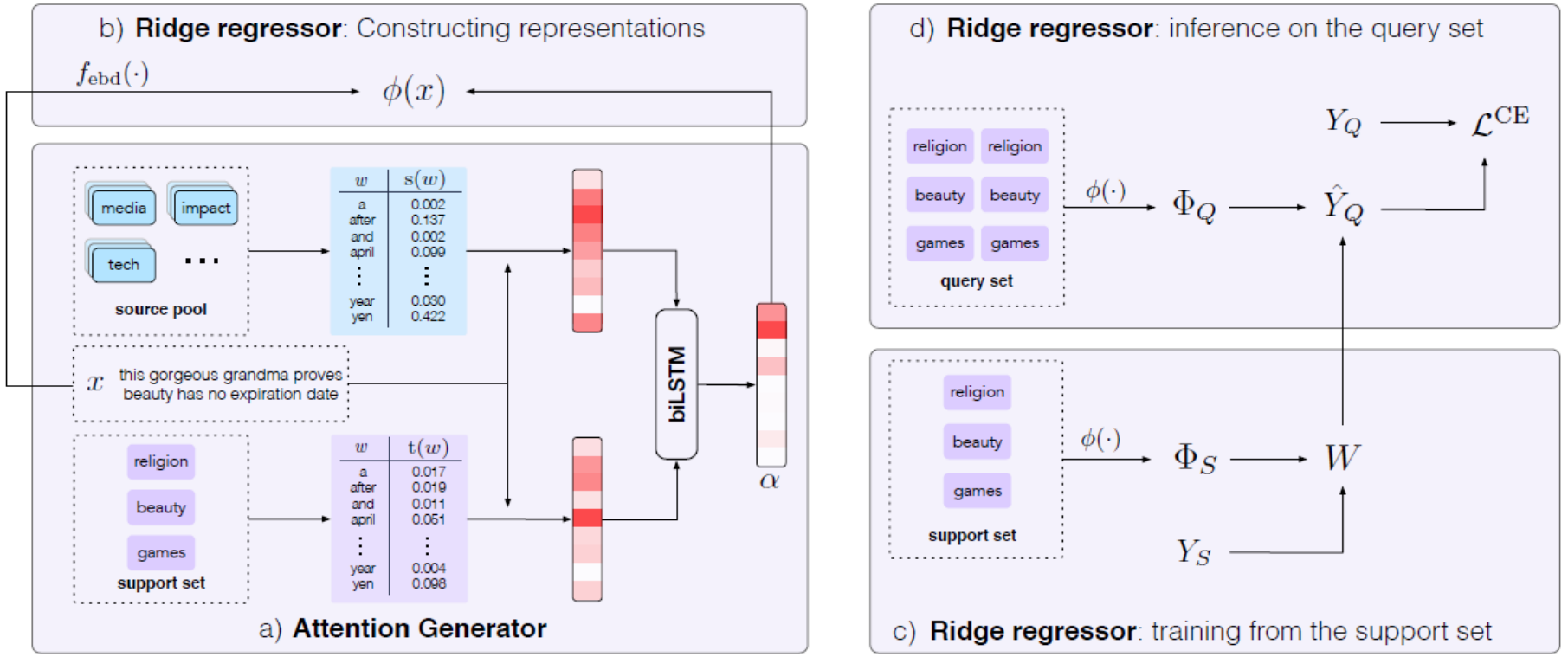
Meta-Learning



Our extension

在元学习训练时，对每个训练段，我们把所有没被选择的类的数据作为source pool；在元学习测试阶段，source pool包括所有类的训练数据。

Our Model


b) Ridge regressor: Constructing representations
$f_{\text{ebd}}(\cdot)$
$\phi(x)$
media
impact
tech
source pool
w | $s(w)$
a 0.002
after 0.137
and 0.002
april 0.099
year 0.030
yen 0.422
x
this gorgeous grandma proves beauty has no expiration date
religion
beauty
games
support set
w | $t(w)$
a 0.017
after 0.019
and 0.011
april 0.051
year 0.004
yen 0.098
biLSTM
α
a) Attention Generator
d) Ridge regressor: inference on the query set
religion
religion
beauty
beauty
games
games
query set
$\phi(\cdot)$
Φ_Q
$\hat{Y}_Q$
Y_Q
$\mathcal{L}^{\text{CE}}$
religion
beauty
games
support set
$\phi(\cdot)$
Φ_S
W
Y_S
c) Ridge regressor: training from the support set

Attention Generator

注意力生成器的目标是评估词的重要度，我们使用source pool来得到词的**一般**重要度，使用支持集得到**类相关**重要度。

由于出现越频繁的词重要度越低，所以用下式得到一般重要度:

$$\mathrm{s}(x_i) := \frac{\varepsilon}{\varepsilon + \mathrm{P}(x_i)}$$

其中x_i是输入句子x的第i个词，$P(x_i)$ 是source pool中x_i的unigram的似然概率。然后用下式得到**类相关**重要度:

$$\mathrm{t}(x_i) := \mathcal{H}(\mathrm{P}(y \mid x_i))^{-1}$$

这里条件似然 $P(y|x_i))$ 是在支持集中使用正则线性分类器得到的，H 为求熵。显然，$t(xi)$ 得到的是词x_i在类别y中的不确定度，从而也是重要度.

Attention Generator

但是只用这些数据效果不好，原因有(1)source pool和支持集包含的是互补的数据，模型无法确定如何结合;(2)这些数据只是词重要度在分类问题上的估计。于是，使用BiLSTM结合这些数据：$h = b_i LSTM([s(x); t(x)])$，然后得到下面的注意力值：

$$\alpha_i := \frac{\exp\left(v^T h_i\right)}{\sum_j \exp\left(v^T h_j\right)}$$

其中h_i是biLSTM在位置i的输出，是 v是一个可训练参数.

Ridge Regressor

对于岭回归器，我们首先定义样本的表示：

$$\phi(x) := \sum_i \alpha_i \cdot f_{\text{ebd}}(x_i)$$

其中$f_{ebd}(x_i)$ 是词x_i对应的词向量。

Training from the support set 然后令$\mathbf{\Phi}_S \in \mathbb{R}^{NK \times E}$ 是支持集的表示，$Y_S \in \mathbb{R}^{NK \times N}$ 是所有类别的one-hot表示。然后最小化下面的loss：

$$\mathcal{L}^{RR}(W) = \|\mathbf{\Phi}_S W - Y_S\|_F^2 + \lambda \|W\|_F^2, \quad W \in \mathbb{R}^{E \times N}$$

可以得到封闭形式解为（I为单位矩阵）：

$$W = \Phi_S^T(\Phi_S \Phi_S^T + \lambda I)^{-1} Y_S$$

Ridge Regressor

Inference on the query set 对询问集，令Φ_Q是其表示，于是我们直接预测其标签

$$\hat{Y}_Q = a\Phi_Q W + b, \quad a \in \mathbb{R}^+, b \in \mathbb{R}$$

最后我们用softmax函数得到$\hat{P}_Q$，计算$\hat{P}_Q$和query set中的数据交叉熵L^{CE}，进行反向传播。

Experiments

数据集：20 Newsgroups，RCV1，Reuters-21578，Amazon product data，HuffPost headlines和FewRel(关系分类数据集)：

Dataset	# tokens / example	vocab size	# examples / cls	# train cls	# val cls	# test cls
20 Newsgroups	340	32137	941	8	5	7
RCV1	372	7304	20	37	10	24
Reuters	168	2234	20	15	5	11
Amazon	140	17062	1000	10	5	9
HuffPost	11	8218	900	20	5	16
FewRel	24	16045	700	65	5	10

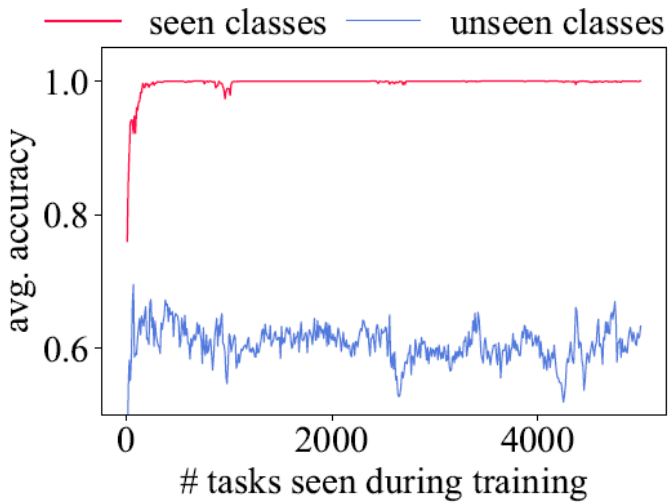
Experiments

下表是6个数据集在不同表示（Rep.）和不同学习算法（Alg.）上的表现：

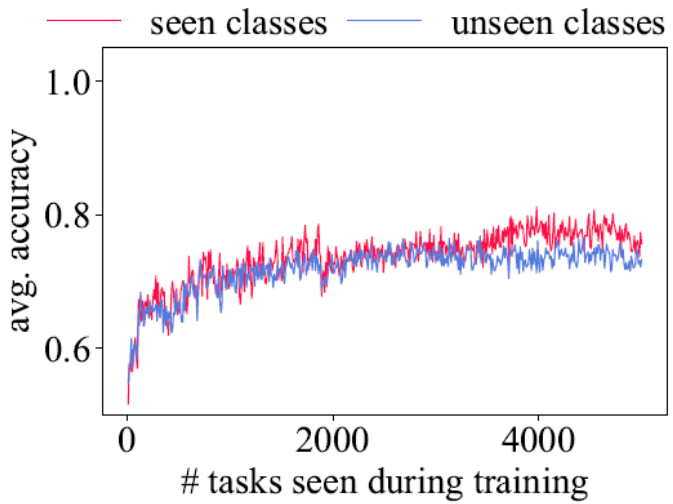
Method		20 News		Amazon		HuffPost		RCV1		Reuters		FewRel		Average	
Rep.	Alg.	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot
AVG	NN	33.9	45.8	46.7	60.3	31.4	41.5	43.7	60.8	56.5	80.5	47.5	60.6	43.3	58.2
IDF	NN	38.8	51.9	51.4	67.1	31.5	42.3	41.9	58.2	57.8	82.9	46.8	60.6	44.7	60.5
CNN	FT	33.0	47.1	45.7	63.9	32.4	44.1	40.3	62.3	70.9	91.0	54.0	71.1	46.0	63.2
AVG	PROTO	36.2	45.4	37.2	51.9	35.6	41.6	28.4	31.2	59.5	68.1	44.0	46.5	40.1	47.4
IDF	PROTO	37.8	46.5	41.9	59.2	34.8	50.2	32.1	35.6	61.0	72.1	43.0	61.9	41.8	54.2
CNN	PROTO	29.6	35.0	34.0	44.4	33.4	44.2	28.4	29.3	65.2	74.3	49.7	65.1	40.1	48.7
AVG	MAML	33.7	43.9	39.3	47.2	36.1	49.6	39.9	50.6	54.6	62.5	43.8	57.8	41.2	51.9
IDF	MAML	37.2	48.6	43.6	62.4	38.9	53.7	42.5	54.1	61.5	72.0	48.2	65.8	45.3	59.4
CNN	MAML	28.9	36.7	35.3	43.7	34.1	45.8	39.0	51.1	66.6	85.0	51.7	66.9	42.6	54.9
AVG	RR	37.6	57.2	50.2	72.7	36.3	54.8	48.1	72.6	63.4	90.0	53.2	72.2	48.1	69.9
IDF	RR	44.8	64.3	60.2	79.7	37.6	59.5	48.6	72.8	69.1	93.0	55.6	75.3	52.6	74.1
CNN	RR	32.2	44.3	37.3	53.8	37.3	49.9	41.8	59.4	71.4	87.9	56.8	71.8	46.1	61.2
OUR		**52.1**	**68.3**	**62.6**	**81.1**	**43.0**	**63.5**	**54.1**	**75.3**	**81.8**	**96.0**	**67.1**	**83.5**	**60.1**	**78.0**
OUR w/o $t(\cdot)$		50.1	67.5	61.7	80.5	42.0	60.8	51.5	75.1	76.7	93.7	66.9	83.2	58.1	76.8
OUR w/o $s(\cdot)$		41.9	60.7	51.1	75.3	40.1	60.2	48.5	72.8	78.1	94.8	65.8	82.6	54.2	74.4
OUR w/o biLSTM		50.3	66.9	61.9	80.9	42.2	63.0	51.8	74.1	77.2	95.4	66.4	82.9	58.3	77.2
OUR w EBD		39.7	57.5	56.5	76.3	40.6	58.6	48.6	71.5	81.7	95.8	61.5	80.9	54.8	73.4

Contrast with lexicon-aware meta-learner

下图说明了本文模型可以有效地避免过拟合



(a) CNN+PROTO



(b) OUR

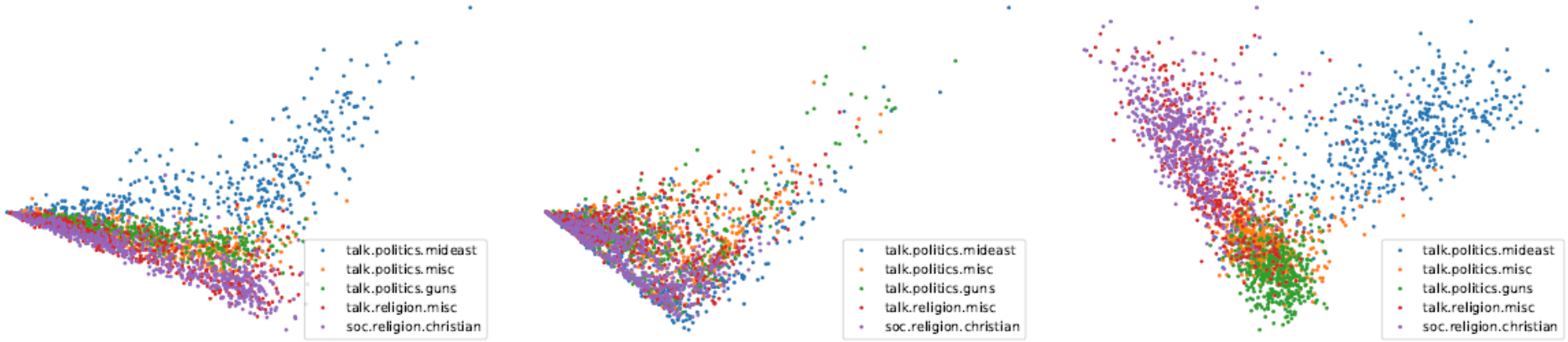
Contextualized representations

对于句子级的数据集（FewRel、HuffPost）还使用BERT提供的上下文表示进行了实验，BERT在FewRel上显著提高了分类性能，另一数据集上则没有。这种差异的产生可能是因为关系分类本身时高度语境化的，新闻分类则多基于关键词。

Method		HuffPost		FewRel	
Rep.	Alg.	1 shot	5 shot	1 shot	5 shot
AVG	NN	23.23 ±0.20	34.22 ±0.28	52.37 ±0.27	64.85 ±0.25
IDF	NN	33.49 ±0.18	45.71 ±0.17	48.65 ±0.18	62.35 ±0.18
CNN	FT	37.30 ±1.08	51.56 ±1.28	61.10 ±2.54	80.04 ±0.69
AVG	PROTO	34.21 ±0.56	49.77 ±1.90	50.27 ±0.98	66.24 ±2.01
IDF	PROTO	36.06 ±0.84	54.58 ±0.99	48.23 ±0.58	67.82 ±0.72
CNN	PROTO	36.17 ±1.00	50.55 ±0.96	57.08 ±5.52	75.01 ±2.21
AVG	MAML	38.58 ±1.56	55.32 ±1.42	47.18 ±3.49	64.50 ±2.72
IDF	MAML	34.22 ±0.74	56.50 ±1.50	50.06 ±2.88	68.43 ±2.50
CNN	MAML	38.39 ±1.68	53.86 ±0.76	47.68 ±1.66	71.56 ±4.75
AVG	RR	25.34 ±0.14	51.52 ±0.14	55.65 ±0.27	73.91 ±0.77
IDF	RR	40.38 ±0.11	61.72 ±1.03	54.48 ±0.26	73.48 ±0.72
CNN	RR	41.37 ±0.54	53.10 ±0.76	65.65 ±5.70	78.65 ±4.24
OUR		**42.12** ±0.15	**62.97** ±0.67	**70.08** ±0.56	**88.07** ±0.27

Table 2: 5-way 1-shot and 5-way 5-shot classification on HuffPost and FewRel using BERT.

PCA visualization of our attention-weighted representation



(a) $s(\cdot)$ (b) $t(\cdot)$ (c) OUR

conclusion

本文提出了一种少样本下的元学习方法，利用词的分布式特征得到注意力值，从而与词重要度建立联系，在多个数据集上取得较好效果。在测试的时候，不需要经过复杂网络，只需用一个线性变换即可进行预测。

Dynamic Memory Induction Networks for Few-Shot Text Classification

Ruiying Geng[1], Binhua Li[1], Yongbin Li[1]*, Jian Sun[1], Xiaodan Zhu[2]
[1] Alibaba Group, Beijing
[2] Ingenuity Labs Research Institute & ECE, Queen's University
{ruiying.gry,binhua.lbh,shuide.lyb,jian.sun}@alibaba-inc.com
zhu2048@gmail.com

Our Model

DMIN

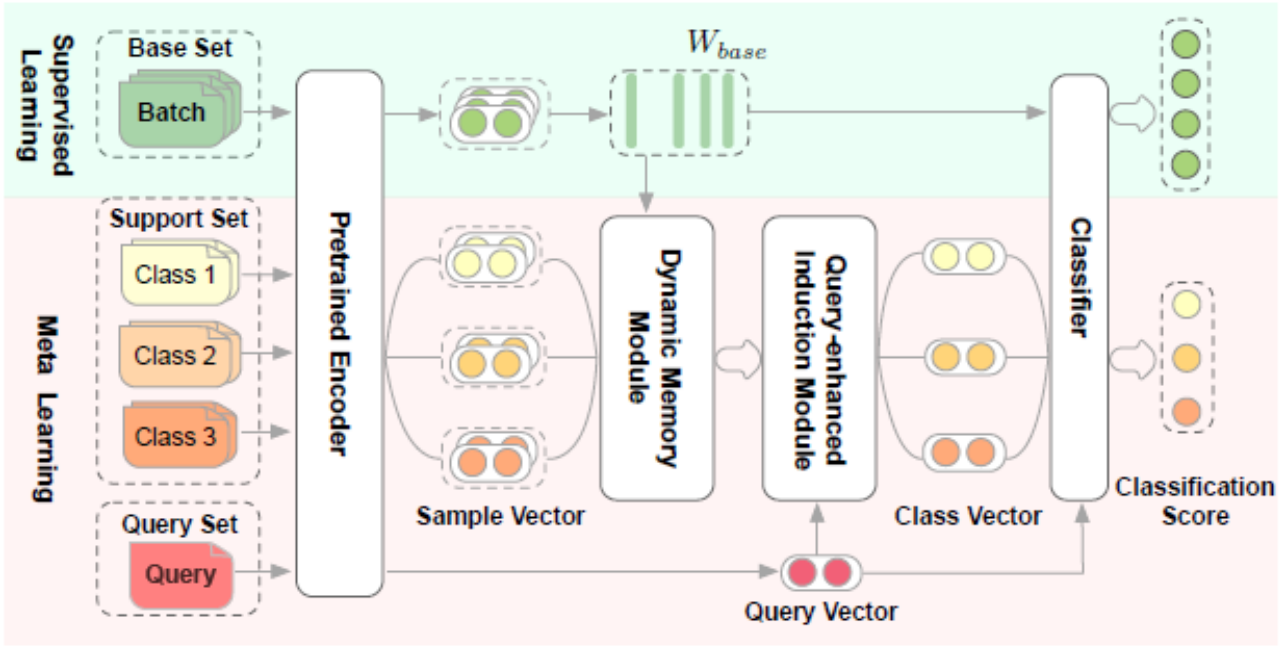


Figure 1: An overview of Dynamic Memory Induction Network with a 3-way 2-shot example.

Dynamic Memory Module

Dynamic Memory Routing Process(DMR)

Algorithm 1 Dynamic Memory Routing Process

Require: r, q and memory $M = \{m_1, m_2, ..., m_n\}$

Ensure: $\boldsymbol{v} = v_1, v_2, ..., v_l, q'$

1: **for** all m_i, v_j **do**
2: $\hat{m}_{ij} = squash(W_j m_i + b_j)$
3: $\hat{q}_j = sqush(W_j q + b_j)$
4: $\alpha_{ij} = 0$
5: $p_{ij} = tanh(PCCs(\hat{m}_{ij}, \hat{q}_j))$
6: **end for**
7: **for** r iterations **do**
8: $d_i = softmax\,(\alpha_i)$
9: $\hat{v}_j = \sum_{i=1}^{n}(d_{ij} + p_{ij})\hat{m}_{ij}$
10: $v_j = squash(\hat{v}_j)$
11: for all i, j: $\alpha_{ij} = \alpha_{i,j} + p_{ij}\hat{m}_{ij}v_j$
12: for all j: $\hat{q}_j = \frac{\hat{q}_j + v_j}{2}$
13: for all i, j: $p_{ij} = tanh(PCCs(\hat{m}_{ij}, \hat{q}_j))$
14: **end for**
15: $q' = concat[\boldsymbol{v}]$
16: **Return** q'

Target:给定一个记忆矩阵 M和样本向量 q,算法的目标是基于记忆矩阵M对样本向量进行适应

$$q' = DMR(M, q)$$

对于样本向量q和记忆矩阵M中的每个类别向量 m_i ，使用动态路径选择中的标准的矩阵变换和squash操作来处理:

$$\hat{m}_{i,j} = \mathbf{squash}(W_j m_i + b_j)$$

$$\hat{q}_j = \mathbf{squash}(W_j q + b_j)$$

$$squash(x) = \frac{||x||^2}{1 + ||x||^2}\frac{x}{||x||}$$

为了更好地适应少样本学习场景，变换权重 w_j和偏移量 b_j 在输入间是共享的。之后，计算 $\hat{m}_{i,j}$和 $\hat{q}_j$之间的皮尔逊相关系数.

$$p_{i,j} = tanh(PCCs(\hat{m}_{ij}, \hat{q}_j))$$

$$PCCs = \frac{Cov(x_1, x_2)}{\sigma_{x_1}\sigma_{x_2}}$$

Dynamic Memory Module

Dynamic Memory Routing Process(DMR)

Algorithm 1 Dynamic Memory Routing Process

Require: r, q and memory $M = \{m_1, m_2, ..., m_n\}$

Ensure: $\boldsymbol{v} = v_1, v_2, ..., v_l, q'$

1: **for** all m_i, v_j **do**
2: $\hat{m}_{ij} = squash(W_j m_i + b_j)$
3: $\hat{q}_j = sqush(W_j q + b_j)$
4: $\alpha_{ij} = 0$
5: $p_{ij} = tanh(PCCs(\hat{m}_{ij}, \hat{q}_j))$
6: **end for**
7: **for** r iterations **do**
8: $d_i = softmax\,(\alpha_i)$
9: $\hat{v}_j = \sum_{i=1}^{n}(d_{ij} + p_{ij})\hat{m}_{ij}$
10: $v_j = squash(\hat{v}_j)$
11: for all i, j: $\alpha_{ij} = \alpha_{i,j} + p_{ij}\hat{m}_{ij}v_j$
12: for all j: $\hat{q}_j = \frac{\hat{q}_j + v_j}{2}$
13: for all i, j: $p_{ij} = tanh(PCCs(\hat{m}_{ij}, \hat{q}_j))$
14: **end for**
15: $q' = concat[\boldsymbol{v}]$
16: **Return** q'

路由迭代过程能够自动调整耦合系数 d_i

$$d_i = softmax(\alpha_i)$$

$$\alpha_{ij} = \alpha_{ij} + p_{ij}\hat{m}_i v_j$$

由于模型的目标是开发基于记忆的动态路由机制用于少样本学习，因此这里添加了PCCs到路由协议中:

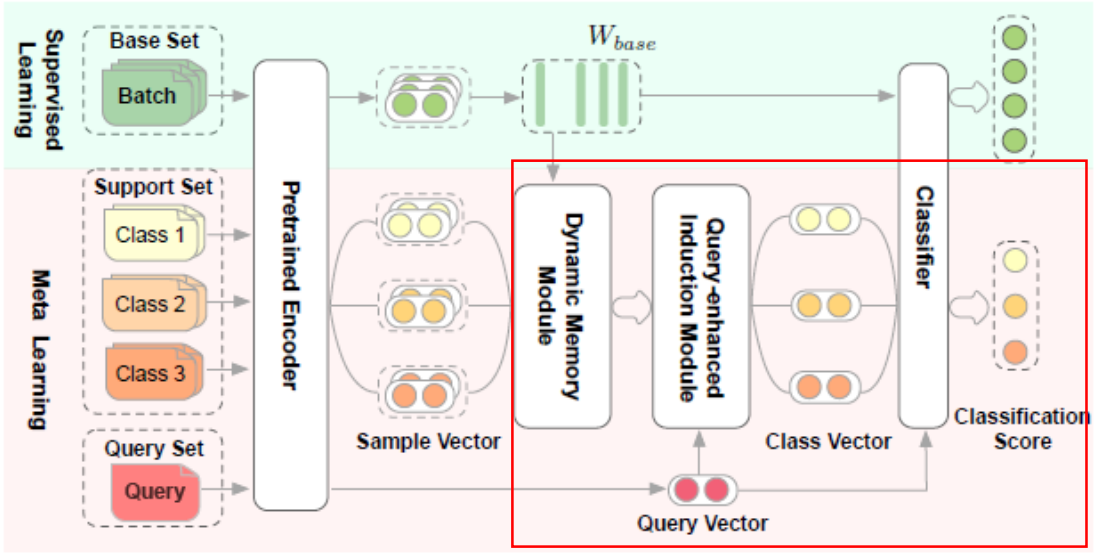
$$\hat{v}_j = \sum_{i=1}^{n}(d_{ij} + p_{ij})m_{ij}$$

$$v_j = squash(\hat{v}_j)$$

在记忆矩阵 W_{base} 的指导下对样本向量 $e_{(c,s)}$进行适应

$$e'_{c,s} = DMR(W_{base}, e_{c,s})$$

Query-enhanced Induction Module



样本向量 $e'_{c,s} = DMR(W_{base}, e_{c,s})$

查询向量 $\{e_q\}_{q=1}^{L}$

$$e_c = DMR(\{e'_{c,s}\}_{s=1,\dots,K}, e_q).$$

Similarity Classifier

$$s_{q,c} = \tau \cdot cos(e_q, e_c) = \tau \cdot \bar{e}_q^T \bar{e}_c.$$

其中τ是可学习的标量值

Objective Function

In the supervised learning stage:最小化基类上input text x和他的类标签的交叉熵

$$L_1(x, \boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_{k=1}^{C_{base}} y_k log(\hat{y}_k)$$

In the meta-training stage：在每一个episode，对于给定support set S和query set Q,最小化在C个新类上的交叉熵

$$L_2(S, Q) = -\frac{1}{C}\sum_{c=1}^{C}\frac{1}{L}\sum_{q=1}^{L} y_q log(\hat{y}_q)$$

其中：$\hat{y}_q = softmax(s_q)$ ，$s_q = \{s_{q,c}\}_{c=1}^{C}$

Experiments

Model	5-way Acc.		10-way Acc.	
	1-shot	5-shot	1-shot	5-shot
BERT	30.79±0.68	63.31±0.73	23.48±0.53	61.18±0.82
ATAML	54.05±0.14	72.79±0.27	39.48±0.23	61.74±0.36
Rel. Net	59.19±0.12	78.35±0.27	44.69±0.19	67.49±0.23
Ind. Net	60.97±0.16	80.91±0.19	46.15±0.26	69.42±0.34
HATT	60.40±0.17	79.46±0.32	47.09±0.28	68.58±0.37
LwoF	63.35±0.26	78.83±0.38	48.61±0.21	69.57±0.35
DMIN	**65.72**±0.28	**82.39**±0.24	**49.54**±0.31	**72.52**±0.25

Table 1: Comparison of accuracy (%) on miniRCV1 with standard deviations.

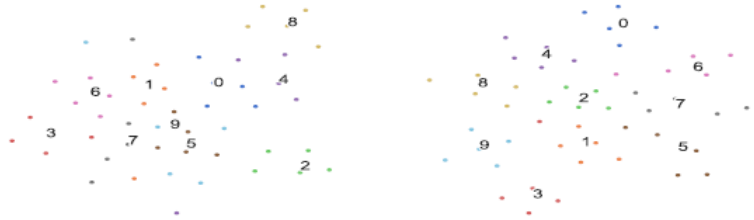
Model	5-way Acc.		10-way Acc.	
	1-shot	5-shot	1-shot	5-shot
BERT	38.06±0.27	64.24±0.36	29.24±0.19	64.53±0.35
ATAML	79.60±0.42	88.53±0.57	63.52±0.34	77.36±0.57
Rel. Net	79.41±0.42	87.93±0.31	64.36±0.58	78.62±0.54
Ind. Net	81.28±0.26	89.67±0.28	64.53±0.38	80.48±0.25
HATT	81.57±0.47	89.27±0.58	65.75±0.61	81.53±0.56
LwoF	79.52±0.29	87.34±0.34	65.04±0.43	80.69±0.37
DMIN	**83.46**±0.36	**91.75**±0.23	**67.31**±0.25	**82.84**±0.38

Table 2: Comparison of accuracy(%) on ODIC with standard deviations.

Ablation Study

Model	Iteration	1 Shot	5 Shot
w/o DMM	3	81.79	90.19
w/o QIM	3	82.37	90.57
DMIN	1	82.70	90.92
DMIN	2	82.95	91.18
DMIN	3	**83.46**	**91.75**

Table 3: Ablation study of accuracy (%) on ODIC in a 5-way setup.



(a) Before DMM (b) After DMM

Figure 2: Effect of the Dynamic Memory Module in a 10-way 5-shot setup.

conclusion

我们提出了动态记忆归纳网络(Dynamic Memory Induction network, DMIN)用于少样本文本分类，它建立在外部工作记忆的动态路由基础上，利用它来跟踪以前的学习经验，而前者则更好地适应和推广以支持集，从而更好地支持不可见类。