Alibaba Group 阿里巴巴集团 | 阿里技术

+ 对抗学习
+ 聊天机器人
+ 无监督学习框架
+ 低比特神经网络
+ 机器翻译

阿里巴巴机器智能技术精选合集

顶级学术会议AAAI-2018收录论文

阿里技术

扫一扫二维码图案，关注我吧

「阿里技术」微信公众号

「阿里巴巴机器智能」公众号

序

2018 年伊始，万众期待的人工智能学术会议 AAAI 2018 在华人春节前一周正式召开，这也标志着全球学术会议新一年的开启。作为一个已举办 32 届的成熟会议，AAAI 不仅因其理论性与应用性交织的特点被中国计算机学会（CCF）推荐为人工智能 A 类会议，更凭借高质量的论文录用水准成为国内高校及研究机构乃至全球学者们密切关注的学术会议。

AAAI 2018 共收到 3808 篇投递论文，相较往年提升了 47%；而今年的录用论文数共有 938 篇，录用率与上年持平，约为 24.6%。

来自中国的论文投递数在今年有了巨大提升，在 AAAI 2018 上共收到 1242 篇论文投稿，并有 785 篇论文被录用。

录用论文现场报告

阿里巴巴在 AAAI 2018 上也收获了 11 篇录用论文，分别来自 iDST、业务平台事业部、阿里妈妈事业部、人工智能实验室、云零售事业部，其中有 5 位作者受邀在主会做 Oral 形式报告，另有 1 位作者携两篇论文在主会以 Poster 形式做报告。论文内容涉及对抗学习、神经网络、提高轻量网络性能的训练框架、聊天机器人、无监督学习框架、极限低比特神经网络等技术方向。

目录

τ-FPL：线性时间的约束容忍分类学习算法

τ-FPL: Tolerance-Constrained Learning in Linear Time

主要作者（中英文）：张翱 Ao Zhang　李楠 Nan Li　王骏 Jun Wang　严峻驰 Junchi Yan
浦健 Jian Pu　查宏远 Hongyuan Zha

论文下载地址：http://arxiv.org/abs/1801.04701

摘要

许多实际应用需要在满足假阳性率上限约束的前提下学习一个二分类器。对于该问题，现存方法往往通过调整标准分类器的参数，或者引入基于领域知识的不平衡分类损失来达到目的。由于没有显式地将假阳性率上限融合到模型训练中，这类方法的精度往往受到制约。本文提出了一个新的排序－阈值方法 τ-FPL 解决这个问题。首先，我们设计了一个新的排序学习方法，其显式地将假阳性率上限值纳入考虑，并且展示了如何高效地在线性时间内求得该排序问题的全局最优解；而后将学到的排序函数转化为一个低假阳性率的分类器。通过理论误差分析以及实验，我们验证了 τ-FPL 对比传统方法在性能及精度上的优越性。

研究背景

在疾病监测，风险决策控制，自动驾驶等高风险的分类任务中，误报正样本与负样本所造成的损失往往是不同的。例如，在高死亡率疾病检测的场景下，遗漏一名潜在病人的风险，要远高于误诊一名正常人。另一方面，两类错误的损失比也很难量化估计。在这种情况下，一个更加合理的学习目标是：我们希望可以在保证分类器假阳性率（即错误地将负样本分类为正样本的概率）低于某个阈值 τ 的前提下，最小化其误分正样本的概率。可以看到，由于问题的转换，传统的基于精度（Accuracy），曲线下面积（AUC）等目标的学习算法将不再适用。

假阳性率约束下的分类学习，在文献中被称为 Neyman-Pearson 分类问题。现存的代表性方法主要有代价敏感学习 (Cost-sensitive learning)，拉格朗日交替优化 (Lagragian Method), 排序 - 阈值法 (Ranking-Thresholding) 等。然而，这些方法通常面临一些问题，限制了其在实际中的使用：

1. 需要额外的超参数选择过程，难以较好地匹配指定的假阳性率；
2. 排序学习或者交替优化的训练复杂度较高，难以大规模扩展；
3. 通过代理函数或者罚函数来近似约束条件，可能导致其无法被满足。

因此，如何针对现有方法存在的问题，给出新的解决方案，是本文的研究目标。

动机：从约束分类到排序学习

考虑经验版本的 Neyman-Pearson 分类问题，其寻找最优的打分函数 f 与阈值 b，使得在满足假阳性率约束的前提下，最小化正样本的误分概率：

$$\min_{f,b} \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(\boldsymbol{x_i^+}) < b) \quad \text{s.t.} \quad \frac{1}{n}\sum_{j=1}^{n}\mathbb{I}(f(\boldsymbol{x_j^-}) > b) \leq \tau \quad (1)$$

我们尝试消除该问题中的约束。首先，我们阐述一个关键的结论：经验 Neyman-Pearson 分类与如下的排序学习问题是等价的，即它们有相同的最优解 f 以及最优目标函数值：

$$\min_{f} \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}\left(f(\boldsymbol{x}_i^+) - f(\boldsymbol{x}^-_{[\lfloor \tau n \rfloor]}) < 0\right)$$

这里，f(x[j]^-) 表示取负样本中第 j 大的元素。直观上讲，该问题本身是一个 pairwise ranking 问题，其将所有的正样本与负样本中第 τn 大的元素相比较。从优化 AUC 的角度，该问题也可看作一个部分 AUC 优化问题，如图 1 所示，其尝试最大化假阳性率 τ 附近的曲线下面积。

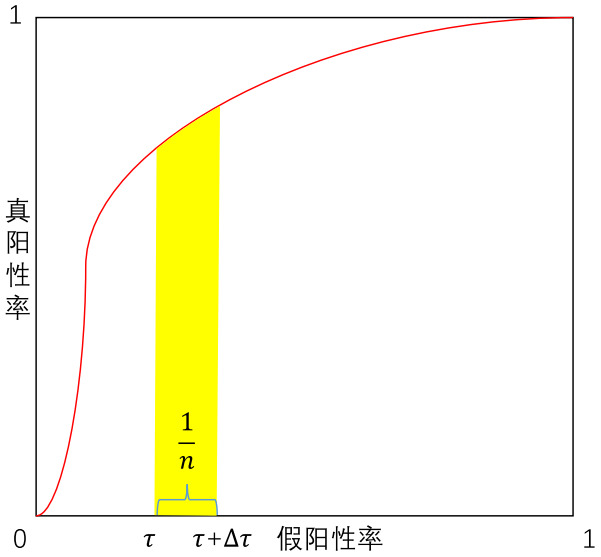


图 1　Neyman-Pearson 分类等价于一个部分 AUC 优化问题

然而，由于引入了取序操作符 [.]，可以证明，即使将 0-1 损失用连续函数替换，该优化问题本身也是 NP-hard 的。因此，我们考虑优化该问题的一个凸上界：

$$\min_f \frac{1}{m}\sum_{i=1}^{m} l\left(f(\boldsymbol{x}_i^+) - \frac{1}{\lfloor \tau n \rfloor}\sum_{j=1}^{\lfloor \tau n \rfloor} f(x_{[j]}^-)\right) \quad (2)$$

这里 l 是任意 0-1 损失的凸代理函数 (convex surrogate function)。(2) 仍然是一个排序问题，其尝试最大化负样本中得分最高的那部分的“质心”与正样本之间的距离。这个新问题有一些良好的性质：

1. 通过设计高效的学习算法，我们可以在线性时间内求得该问题的全局最优解，这使其非常适合于大规模数据下的场景；
2. 形式上显式地包含 τ，无需引入额外的损失超参数 (cost-free)；
3. 最优解 f 有可理论保证的泛化误差界。

我们也可以从对抗学习 (Adversarial learning) 的角度，给出排序问题 (2) 的一

个直观解释。读者可以验证，(2) 与如下的对抗学习问题是等价的：

$$\min_{f} \max_{\boldsymbol{p}\in\Delta} \frac{1}{m}\sum_{i=1}^{m} l\left(f(\boldsymbol{x}_i^+) - \sum_{j=1}^{n} p_j f(x_j^-)\right) \quad (3)$$

其中 k = τn，且

$$\Delta = \{\boldsymbol{p} = (p_1, ..., p_n)^T \mid 0 \le p_j \le \tfrac{1}{k}, \mathbf{1}^T\boldsymbol{p} = 1\}$$

换句话说，排序学习问题 (2) 可以看作是在两个玩家——打分函数 A 与样本分布 B 间进行的一个 min-max 游戏。对于 A 给出的每个 f，B 尝试从负样本分布的集合△中给出一个最坏的分布 p，以最小化 A 的期望收益。该游戏达到纳什均衡 (Nash equilibrium) 时的稳点，也就是我们要求的最优解。

τ-FPL 算法总览

如上所述，τ-FPL 的训练分为两个部分，排序 (scoring) 与阈值 (thresholding)。在排序阶段，算法学习一个排序函数，其尝试将正样本排在负样本中得分最高的那部分的“质心”之前。阈值阶段则选取合适的阈值，将学到的排序函数转化为二分类器。

排序学习优化算法

考虑与 (2) 等价的对抗学习问题 (3)，其对偶问题如下：

$$\min_{(\boldsymbol{\alpha},\boldsymbol{\beta}\in\Gamma_k)} g(\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{2m\lambda}||\boldsymbol{\alpha}^T\boldsymbol{X}^+ - \boldsymbol{\beta}^T\boldsymbol{X}^-||^2 + \sum_{i=1}^{m} l_*(-\alpha_i)$$

$$\Gamma_k = \left\{\boldsymbol{\alpha}\in\mathbb{R}_+^m, \boldsymbol{\beta}\in\mathbb{R}_+^n \mid \sum_{i=1}^{m}\alpha_i = \sum_{j=1}^{n}\beta_j; \beta_j \le \frac{1}{k}\sum_{i=1}^{n}\beta_i, \forall j\right\}$$

这个新问题不含任何不可导项，并且目标函数 g 是光滑的 (Smooth)。因此，我们可以使用投影梯度下降算法求解该问题，并利用加速梯度方法 (Nesterov) 获得最优的收敛率。

Algorithm 1 τ-FPL 排序函数学习

输入： 样本矩阵 $X^+ \in \mathbb{R}^{m\times d}, X^- \in \mathbb{R}^{n\times d}$,

输入： 假阳性率上限 τ, 正则化参数 λ, 精度 ϵ

随机初始化 $\boldsymbol{\alpha_0}$ 与 $\boldsymbol{\beta_0}$

初始化计数器: $t \leftarrow 0$

while $t = 0$ **or** $|g(\boldsymbol{\alpha_t}, \boldsymbol{\beta_t}) - g(\boldsymbol{\alpha_{t-1}}, \boldsymbol{\beta_{t-1}})| > \epsilon$ **do**

  计算 $g(\cdot)$ 在 $(\boldsymbol{\alpha_t}, \boldsymbol{\beta_t})$ 处的梯度

  梯度下降，得到 $\boldsymbol{\alpha'_{t+1}}, \boldsymbol{\beta'_{t+1}}$

  投影 $\boldsymbol{\alpha'_{t+1}}, \boldsymbol{\beta'_{t+1}}$ 到可行域 Γ_k 上:
  $(\boldsymbol{\alpha_{t+1}}, \boldsymbol{\beta_{t+1}}) \leftarrow \Pi_{\Gamma_t}(\boldsymbol{\alpha'_{t+1}}, \boldsymbol{\beta'_{t+1}})$

  更新计数器: $t \leftarrow t + 1$;

end while

Return $\boldsymbol{w} \leftarrow (m\lambda)^{-1}(\boldsymbol{\alpha_t}^T X^+ - \boldsymbol{\beta_t}^T X^-)^T$

线性时间的变上界欧式投影

排序学习算法的一个关键步骤，是将梯度下降的解投影到可行集 Gamma_k 上。我们注意到，这个投影问题是一大类被广泛研究的欧式投影问题的推广。然而传统方法仅对一些特例可以高效求解，即便对于该问题的一个简化版本，也仅能达到 O(nlogn + τ n^2) 的超线性复杂度。

本文中，我们提出了一个算法，能够在 O(n) 的线性时间内高效地求解该投影问题，且其性能不受 τ 增长所带来的影响。该算法的核心是二分求根法与分治法的有效结合。根据 KKT 最优条件，我们将投影问题转换为一个求解分段线性方程组的问题，该方程组仅包含三个未知的对偶变量，且可以通过二分求根法获得指定精度的解。进一步地，利用方程组分段线性的特殊结构，以及对偶变量间“同变”的单调性质，我们可以在二分过程中逐步减少每次迭代的计算消耗，最终显著减少总的算法运行时间。实验中，我们观察到随着 n 与 τ 的增长，我们的算法较现有的求解该类问题的方法有一至三个数量级的性能提升，见图 2。

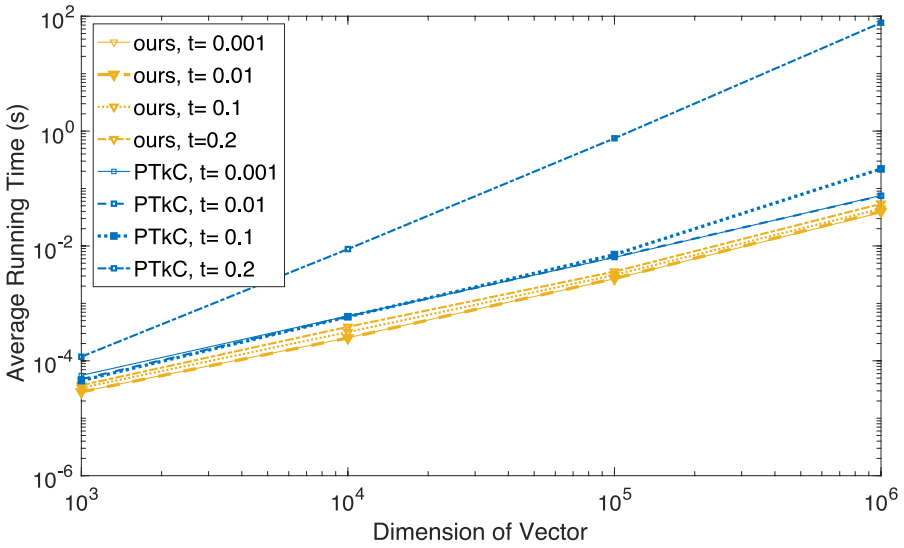


图 2　我们的方法与现存算法 (PTkC) 在求解简化版问题时的性能对比 (log−log 曲线)

阈值选择

阈值选择阶段，算法每次将训练集分为两份，一份训练排序函数，另一份用来选取阈值。该过程可以进行多次，以充分利用所有样本，最终的阈值则是多轮阈值的平均。该方法结合了 out−of−bootstrap 与软阈值技术分别控制偏差及方差的优点，也适于并行。

理论结果

收敛率与时间复杂度　通过结合加速梯度方法与线性时间投影算法，τ −FPL 可以确保每次迭代的线性时间消耗以及最优的收敛率。图 3 将 τ −FPL 与一些经典方法进行了对比，可以看到其同时具备最优的训练及验证复杂度。

泛化性能保证　我们也从理论上给出了 τ −FPL 学得模型的泛化误差界，证明了泛化误差以很高的概率被经验误差所上界约束。这给予了我们设法求解排序问题 (2) 的理论支持。

算法	训练 复杂度	交叉验证 复杂度
τ-FPL	$O((m+n)d/T^2)$	线性
TopPush	$O((m+n)d/T^2)$	线性
CS-SVM	$O((m+n)d/T)$	二次
$\text{SVM}_{\text{tight}}^{\text{pAUC}}$	$O((m\log m + n\log n + (m+n)d)/T)$	线性
Bipartite Ranking	$O(((m+n)d + (m+n)\log(m+n))/T)$ $\sim O(mnd + mn\log(mn)/\sqrt{T})$	线性

图 3　不同算法的训练复杂度比较

实验结果

Dataset	heart 120/150,d:13		spambase 1813/2788,d:57					real-sim 22238/50071,d:20958					w8a 2933/62767,d:300					
τ(%)	5	10	0.1	0.5	1	5	10	0.01	0.1	1	5	10	0.05	0.1	0.5	1	5	10
CS-SVM	.526	.691	.109	.302	**.487**	.811	.920	.376	**.748**	.921	.972	.990	.501	.520	.649	.695	.828	.885
TopPush	**.541**	.711	.112	.303	.484	.774	.845	**.391**	.747	.920	.968	.983	**.508**	**.551**	.627	.656	.761	.842
$\text{SVM}_{\text{tight}}^{\text{pAUC}}$	.509	.728	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
τ-Rank	**.541**	**.740**	**.112**	**.305**	.460	**.842**	**.929**	**.391**	.747	.920	**.975**	**.991**	**.508**	**.551**	.645	**.710**	**.832**	**.894**
2τ-Rank	**.547**	**.734**	**.112**	**.311**	.477	**.862**	**.936**	**.391**	.747	**.922**	**.978**	**.992**	**.508**	.549	**.675**	**.739**	**.841**	**.902**

图 4 报告了不同算法优化部分 AUC 的效果，'N/A' 代表该模型的训练无法在一周内完成。可以看到，τ −FPL 对于不同 τ 值，在大部分实验中都具有较好的表现。另外，其相比二分排序算法有明显的性能优势。

Dataset(+/-)	τ(%)	BS-SVM	CS-LR	CS-SVM	CS-SVM-OOB	τ-FPL	2τ-FPL
heart	5	(.069, .675), .713	(.035, .394), .606	(.027, .327), .673	(.058, .553), .609	(.053, .582), **.468**	(.055, .584), **.514**
120/150,d:13	10	(.121, .774), .435	(.058, .615), .385	(.078, .666), .334	(.088, .682), .318	(.086, .686), **.314**	(.080, .679), **.317**
breast-cancer	1	(.015, .964), .559	(.007, .884), **.116**	(.006, .870), .130	(.014, .955), .451	(.013, .955), .324	(.011, .949), .192
239/444	5	(.063, .978), .276	(.013, .965), .035	(.017, .965), .034	(.046, .974), **.026**	(.041, .976), **.025**	(.045, .974), **.026**
d:10	10	(.113, .985), .142	(.035, .970), .030	(.044, .973), .027	(.095, .981), .020	(.098, .982), **.018**	(.094, .982), **.018**
spambase	0.5	(.008, .426), 1.220	(.007, .011), 1.362	(.002, .109), .891	(.005, .275), .790	(.005, .278), **.722**	(.004, .268), **.732**
1813/2788	1	(.013, .583), .748	(.007, .011), .989	(.004, .256), .744	(.009, .418), .582	(.008, .416), .584	(.008, .440), **.560**
d:57	5	(.054, .895), .192	(.007, .011), .989	(.020, .667), .333	(.047, .793), .207	(.041, .822), **.178**	(.046, .845), **.155**
	10	(.103, .941), .087	(.007, .011), .989	(.051, .716), .284	(.090, .902), .099	(.087, .925), **.075**	(.090, .928), **.072**
real-sim	0.01	(.002, .813), 22.376	(.001, .207), 7.939	(.000, .209), .791	(.000, .268), .833	(.000, .270), **.730**	(.000, .270), **.730**
22238/50071	0.1	(.008, .919), 7.09	(.001, .207), .826	(.001, .700), .428	(.001, .584), .416	(.001, .585), **.415**	(.001, .585), **.415**
d:20958	0.5	(.023, .966), 3.680	(.001, .207), .794	(.001, .755), .245	(.003, .810), .190	(.003, .829), **.174**	(.003, .827), **.181**
	1	(.036, .978), 2.570	(.001, .207), .794	(.007, .880), .121	(.007, .875), .125	(.007, .894), **.115**	(.006, .891), **.109**
	5	(.094, .994), .878	(.078, .994), .575	(.029, .931), .139	(.039, .965), .035	(.041, .972), **.028**	(.044, .974), **.028**
	10	(.133, 0.997), .336	(.078, .994), **.007**	(.069, .993), .007	(.099, .986), .019	(.092, .991), .009	(.094, .991), .009
w8a	0.05	(.001, .525), .966	(.000, .101), .900	(.000, .420), .580	(.000, .438), **.562**	(.000, .428), .572	(.000, .428), .572
1933/62767	0.1	(.001, .585), .710	(.000, .119), .881	(.000, .447), .553	(.001, .493), .507	(.001, .495), **.505**	(.001, .499), **.501**
d:123	0.5	(.006, .710), .437	(.000, .119), .881	(.002, .595), .405	(.003, .634), .366	(.003, .654), **.347**	(.003, .667), **.333**
	1	(.011, .749), .341	(.014, .696), .715	(.006, .642), .358	(.006, .695), .305	(.006, .702), **.298**	(.007, .726), **.274**
	5	(.048, .823), .177	(.014, .696), .305	(.013, .701), .299	(.046, .805), .195	(.033, .818), .182	(.036, .827), **.173**
	10	(.049, .823), .177	(.014, .696), .305	(.013, .701), .299	(.053, .814), .186	(.042, .833), **.167**	(.038, .826), **.174**

图 5 比较了不同算法输出的分类器的分类性能。这里选取 NP-score 作为评价标准，其综合考虑了分类器间的精度差异与违背假阳性率约束的惩罚。可以看到，采用 OOB 阈值的算法在大部分情况下均可有效地抑制假阳性率在允许范围内。另外，即使采用同样的阈值选择方法，τ-FPL 也可以获得较代价敏感学习 (CS-SVM-OOB) 更好的精度。

总结

在高风险分类任务中控制假阳性率是重要的。本文中，我们主要研究在指定的假阳性率容忍度 τ 下学习二分类器。为此，我们提出了一个新的排序学习问题，其显式地最大化将正样本排在 前 τ % 负样本的质心之上的概率。通过结合加速梯度方法与线性时间投影，该排序问题可以在线性时间内被高效地解决。我们通过选取合适的阈值将学到的排序函数转换为低假阳性率的分类器，并从理论和实验两个角度验证了所提出方法的有效性。

基于注意力机制的用户行为建模框架及其在推荐领域的应用

ATRank: An Attention-Based User Behavior Modeling Framework for Recommendation

主要作者（中英文）：周畅 Chang Zhou　白金泽 Jinze Bai　宋军帅 Junshuai Song

论文下载地址：https://arxiv.org/abs/1711.06632

一、摘要

本文提出一种基于注意力机制的用户异构行为序列的建模框架，并将其应用到推荐场景中。我们将不同种类的用户行为序列进行分组编码，并映射到不同子空间中。我们利用 self-attention 对行为间的互相影响进行建模。最终我们得到用户的行为表征，下游任务就可以使用基本的注意力模型进行有更具指向性的决策。我们尝试用同一种模型同时预测多种类型的用户行为，使其达到多个单独模型预测单类型行为的效果。另外，由于我们的方法中没有使用 RNN,CNN 等方法，因此在提高效果的同时，该方法能够有更快的训练速度。

二、研究背景

一个人是由其所表现出的行为所定义。而对用户精准、深入的研究也往往是很多商业问题的核心。从长期来看，随着人们可被记录的行为种类越来越多，平台方需要有能力通过融合各类不同的用户行为，更好的去理解用户，从而提供更好的个性化服务。

对于阿里巴巴来说，以消费者运营为核心理念的全域营销正是一个结合用户全生态行为数据来帮助品牌实现新营销的数据 & 技术驱动的解决方案。因此，对用户行为的研究就成为了一个非常核心的问题。其中，很大的挑战来自于能否对用户的异构行

为数据进行更精细的处理。

在这样的背景下，本文提出一个通用的用户表征框架，试图融合不同类型的用户行为序列，并以此框架在推荐任务中进行了效果验证。另外，我们还通过多任务学习的方式，期望能够利用该用户表征实现不同的下游任务。

三、相关工作

异构行为建模：通常通过手动特征工程来表示用户特征。这些手工特征以聚合类特征或无时序的 id 特征集合为主。

单行为序列建模：用户序列的建模通常会用 RNN（LSTM/GRU）或者 CNN + Pooling 的方式。RNN 难以并行，训练和预测时间较长，且 LSTM 中的 Internal Memory 无法记住特定的行为记录。CNN 也无法保留特定行为特征，且需要较深的层次来建立任意行为间的影响。

异构数据表征学习：参考知识图谱和 Multi-modal 的表征研究工作，但通常都有非常明显的映射监督。而在我们的任务中，异构的行为之间并没有像 image caption 这种任务那样明显的映射关系。

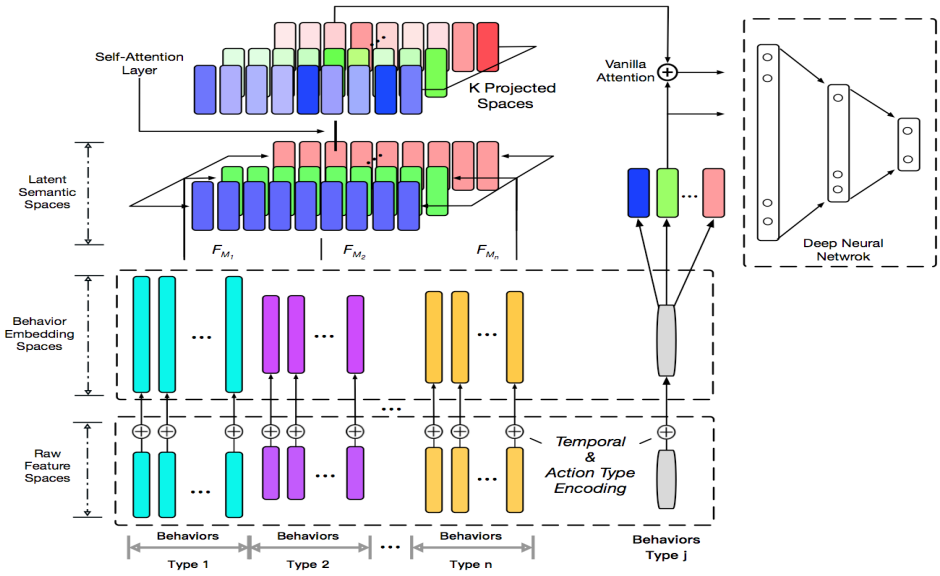
本文的主要贡献如下：

1. 尝试设计和实现了一种能够融合用户多种时序行为数据的方法，较为创新的想法在于提出了一种同时考虑异构行为和时序的解决方案，并给出较为简洁的实现方式。

2. 使用类似 Google 的 self-attention 机制去除 CNN、LSTM 的限制，让网络训练和预测速度变快的同时，效果还可以略有提升。

3. 此框架便于扩展。可以允许更多不同类型的行为数据接入，同时提供多任务学习的机会，来弥补行为稀疏性。

四、ATRank 方案介绍



整个用户表征的框架包括原始特征层，语义映射层，Self-Attention 层和目标网络。语义映射层能让不同的行为可以在不同的语义空间下进行比较和相互作用。Self-Attention 层让单个的行为本身变成考虑到其他行为影响的记录。目标网络则通过 Vanilla Attention 可以准确的找到相关的用户行为进行预测任务。通过 Time Encoding + Self Attention 的思路，我们的实验表明其的确可以替代 CNN/RNN 来描述序列信息，能使模型的训练和预测速度更快。

1. 行为分组

某个用户的行为序列可以用一个三元组来描述（动作类型，目标，时间）。我们先将用户不同的行为按照目标实体进行分组，如图中最下方不同颜色 group。例如商品行为，优惠券行为，关键字行为等等。动作类型可以是点击 / 收藏 / 加购、领取 / 使用等等。

每个实体都有自己不同的属性，包括实值特征和离散 id 类特征。动作类型是 id 类，我们也将时间离散化。三部分相加得到下一层的向量组。

即，某行为的编码 = 自定义目标编码 + lookup（离散化时间）+ lookup（动作类型）。

由于实体的信息量不同，因此每一组行为编码的向量长度不一，其实也代表行为所含的信息量有所不同。另外，不同行为之间可能会共享一些参数，例如店铺 id，类目 id 这类特征的 lookup table，这样做能减少一定的稀疏性，同时降低参数总量。

分组的主要目的除了说明起来比较方便，还与实现有关。因为变长、异构的处理很难高效的在不分组的情况下实现。并且在后面还可以看到我们的方法实际上并不强制依赖于行为按时间排序。

2. 语义空间映射

这一层通过将异构行为线性映射到多个语义空间，来实现异构行为之间的同语义交流。例如框架图中想表达的空间是红绿蓝（RGB）构成的原子语义空间，下面的复合色彩（不同类型的用户行为）会投影到各个原子语义空间。在相同语义空间下，这些异构行为的相同语义成分才有了可比性。

类似的思路其实也在 knowledge graph representation 里也有出现。而在 NLP 领域，今年也有一些研究表明多语义空间的 attention 机制可以提升效果。个人认为的一点解释是说，如果不分多语义空间，会发生所谓语义中和的问题。简单的理解是，两个不同种类的行为 a,b 可能只在某种领域上有相关性，然而当 attention score 是一个全局的标量时，a,b 在不那么相关的领域上会增大互相影响，而在高度相关的领域上这种影响则会减弱。

尽管从实现的角度上来说，这一层就是所有行为编码向一个统一的空间进行映射，映射方法线性非线性都可以，但实际上，对于后面的网络层来说，我们可以看作是将一个大的空间划分为多语义空间，并在每个子空间里进行 self-attention 操作。因此从解释上来说，我们简单的把这个映射直接描述成对多个子语义空间进行投影。

3. Self Attention 层

Self Attention 层的目的实际上是想将用户的每一个行为从一个客观的表征，做成一个用户记忆中的表征。客观的表征是指，比如 A,B 做了同样一件事，这个行为本身的表征可能是相同的。但这个行为在 A,B 的记忆中，可能强度、清晰度是完全不一样的，这是因为 A,B 的其他行为不同。实际上，观察 softmax 函数可知，某种相似行为做的越多，他们的表征就越会被平均。而带来不一样体验的行为则会更容易保留

自己的信息。因此 self attention 实际上模拟了一个行为被其他行为影响后的表征。

另外，Self Attention 可以有多层。可以看到，一层 Self-Attention 对应着一阶的行为影响。多层则会考虑多阶的行为影响。这个网络结构借鉴的是 google 的 self-attention 框架。

具体计算方式如下：

记 S 是整个语义层拼接后的输出，Sk 是第 k 个语义空间上的投影，则经过 self-attention 后第 k 个语义空间的表征计算公式为：

$$A_k = softmax(S_k W_k S^T)$$
$$C_k = A_k Q_k S$$

这里的 attention function 可以看做是一种 bilinear 的 attention 函数。最后的输出则是这些空间向量拼接后再加入一个前馈网络。

$$C = \mathscr{F}_{self}(concat^{(1)}(C_1, C_2, ..., C_K))$$

4. 目标网络

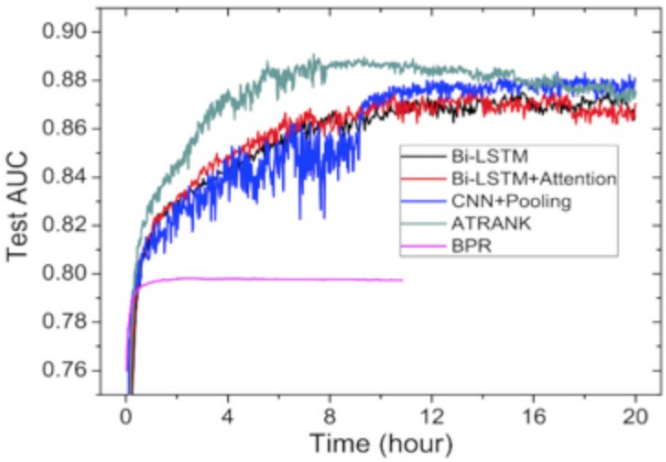
目标网络会随着下游任务的不同而定制。本文所涉及的任务是用户行为预测及推荐场景的点击预测的任务，采用的是 point-wise 的方式进行训练和预测。

框架图中灰色的 bar 代表待预测的任意种类的行为。我们将该行为也通过 embedding、projection 等转换，然后和用户表征产出的行为向量做 vanilla attention。最后 Attention 向量和目标向量将被送入一个 Ranking Network。其他场景强相关的特征可以放在这里。这个网络可以是任意的，可以是 wide & deep，deep FM，pnn 都行。我们在论文的实验中就是简单的 dnn。

五、离线实验

为了比较框架在单行为预测时的效果，我们在 amazon 购买行为的公开数据集上的实验。

训练收敛结果如下图：



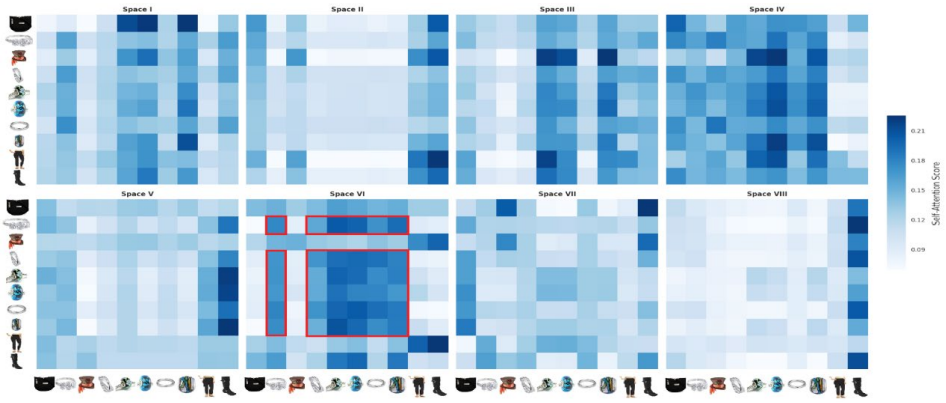
用户平均 AUC 如下图：

Dataset	Electro.	Clothe.
BPR	0.7982	0.7061
Bi-LSTM	0.8757	0.7869
Bi-LSTM + Attention	0.8769	0.7835
CNN + Max Pooling	0.8804	0.7786
ATRank	**0.8921**	**0.7905**

实验结论：在行为预测或推荐任务中，self-attention + time encoding 也能较好的替代 cnn+pooling 或 lstm 的编码方式。训练时间上能较 cnn/lstm 快 4 倍。效果上也能比其他方法略好一些。

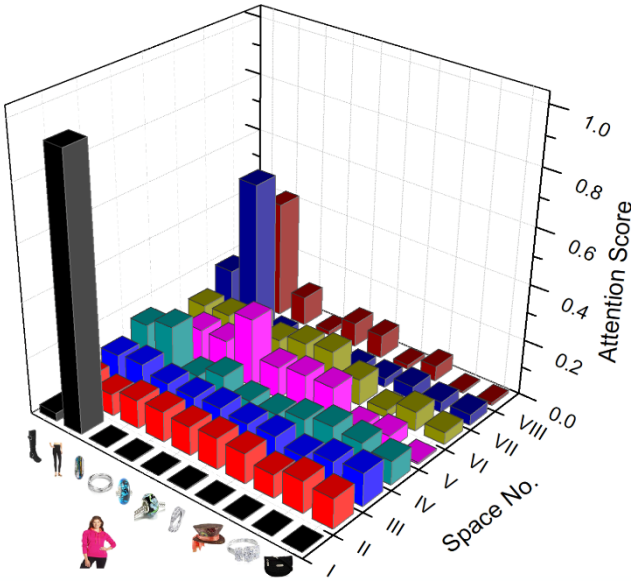
Case Study

为了深究 Self-Attention 在多空间内的意义，我们在 amazon dataset 上做了一个简单的 case study。如下图：



从图中我们可以看到，不同的空间所关注的重点很不一样。例如空间 I, II, III, VIII 中每一行的 attention 分的趋势类似。这可能是主要体现不同行为总体的影响。另一些空间，例如 VII，高分 attention 趋向于形成稠密的正方形，我们可以看到这其实是因为这些商品属于同样的类目。

下图则是 vanilla attention 在不同语义空间下的得分情况。



多任务学习

论文中，我们离线收集了阿里电商用户对商品的购买点击收藏加购、优惠券领

取、关键字搜索三种行为进行训练，同样的也对这三种不同的行为同时进行预测。其中，用户商品行为记录是全网的，但最终要预测的商品点击行为是店铺内某推荐场景的真实曝光、点击记录。优惠券、关键字的训练和预测都是全网行为。

我们分别构造了 7 种训练模式进行对比。分别是单行为样本预测同类行为（3 种），全行为多模型预测单行为（3 种），全行为单模型预测全行为（1 种）。在最后一种实验设置下，我们将三种预测任务各自切成 mini-batch，然后统一进行 shuffle 并训练。

实验结果如下表：

Predict Target	Item	Query	Coupon
Bi-LSTM	0.6779	0.6019	0.8500
Bi-LSTM + Attention	0.6754	0.5999	0.8413
CNN + Max Pooling	0.6762	0.6100	0.8611
ATRank-one2one	0.6785	0.6132	0.8601
ATRank-all2one	**0.6825**	**0.6297**	**0.8725**
ATRank-all2all	0.6759	0.6199	0.8587

all2one 是三个模型分别预测三个任务，all2all 是单模型预测三个任务，即三个任务共享所有参数，而没有各自独占的部分。因此 all2all 与 all2one 相比稍低可以理解。我们训练多任务 all2all 时，将三种不同的预测任务各自 batch 后进行充分随机的 shuffle。文中的多任务训练方式还是有很多可以提升的地方，前沿也出现了一些很好的可借鉴的方法，是我们目前正在尝试的方向之一。

实验表明，我们的框架可以通过融入更多的行为数据来达到更好的推荐 / 行为预测的效果。

六、总结

本文提出一个通用的用户表征框架，来融合不同类型的用户行为序列，并在推荐任务中得到验证。

未来，我们希望能结合更多实际的商业场景和更丰富的数据沉淀出灵活、可扩展的用户表征体系，从而更好的理解用户，提供更优质的个性化服务，输出更全面的数据能力。

极限低比特神经网络：通过 ADMM 算法进行极限压缩

Extremely Low Bit Neural Network: Squeeze the Last Bit Out with ADMM

主要作者（中英文）：冷聪 Cong Leng　窦则胜 Zesheng Dou　李昊 Hao Li
朱胜火 Shenghuo Zhu　金榕 Rong Jin

论文下载地址：https://arxiv.org/abs/1707.09870

研究背景

近年来，深度学习在人工智能领域取得了重大的突破。在计算机视觉、语音识别等诸多领域，深度神经网络 (DNN, Deep Neural Network) 均被证明是一种极具成效的问题解决方式。如卷积神经网络 (CNN, Convolutional neural network) 在计算机视觉诸多传统问题（分类、检测、分割）都超越了传统方法，循环神经网络 (RNN, Recurrent Neural Networks) 则在时序信号处理，如机器翻译，语音识别等超过传统方法。

在利用深度网络解决问题的时候人们常常倾向于设计更为复杂的网络收集更多的数据以期获得更高的性能。但是，随之而来的是模型的复杂度急剧提升，直观的表现是模型的层数越来越深，参数越来越多。这会给深度学习带来两个严重的问题：

(1) 随着模型参数的增多，模型的大小越来越大，给嵌入式端模型的存储带来了很大的挑战。

(2) 随着模型的增大，模型 inference 的时间越来越长，latency 越来越大。

以上两个问题给深度学习在终端智能设备上的推广带来了很大的挑战。比如，经典的深度卷积网络 VGG-16 的模型大小达到 528M，用户很难接受下载一个如此大的模型到手机或者其他终端设备上。同时，在一般的智能手机上，VGG-16 识别一

张图像的时间高达 3000+ms，这个 latency 对于大多数用户来说也是难以接受的。此外，由于深度网络的计算量很大，运行深度网络的能耗很高，这对于手机等终端设备也是一个巨大的挑战。

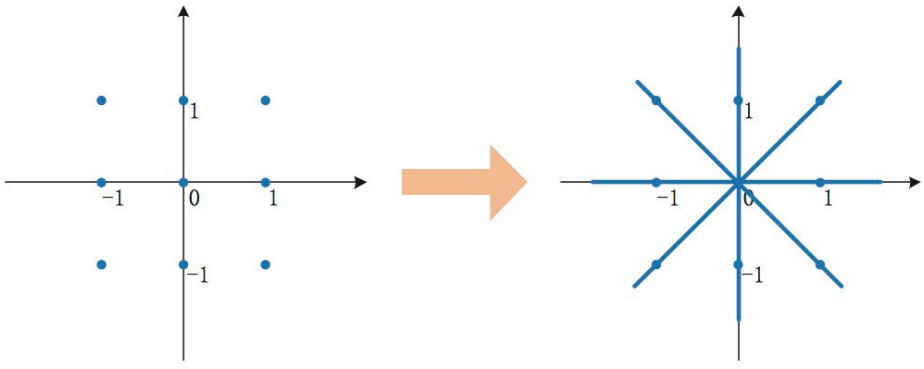
所提出的算法

在这个工作中，我们提出一种基于低比特表示技术的神经网络压缩和加速算法。我们将神经网络的权重表示成离散值，并且离散值的形式为 2 的幂次方的形式，比如{−4，−2，−1，0，1，2，4}。这样原始 32 比特的浮点型权重可以被压缩成 1−3 比特的整形权重，同时，原始的浮点数乘法操作可以被定点数的移位操作所替代。在现代处理器中，定点移位操作的速度和能耗是远远优于浮点数乘法操作的。

首先，我们将离散值权重的神经网络训练定义成一个离散约束优化问题。以三值网络为例，其目标函数可以表示为：

$$\min_{W} \quad f(W) \qquad \text{s.t. } W \in \mathcal{C} = \{-1, 0, +1\}^d$$

更进一步，我们在约束条件中引入一个 scale 参数。对于三值网络，我们将约束条件写成 {−a, 0, a}, a>0. 这样做并不会增加计算代价，因为在卷积或者全连接层的计算过程中可以先和三值权重 {−1, 0, 1} 进行矩阵操作，然后对结果进行一个标量 scale。从优化的角度看，增加这个 scale 参数可以大大增加约束空间的大小，这有利于算法的收敛。如下图所示，



(a) (b)

对于三值网络而言，scale 参数可以将约束空间从离散的 9 个点扩增到 4 条直线。

为了求解上述约束优化问题，我们引入 ADMM 算法。在此之前，我们需要对目标函数的形式做一个等价变换。

$$\begin{aligned} \min_{W,G} \quad & f(W) + I_{\mathcal{C}}(G) \\ \text{s.t.} \quad & W = G \end{aligned}$$

其中 Ic 为指示函数，如果 G 符合约束条件，则 Ic(G)=0，否则 Ic(G) 为无穷大。该目标函数的增广拉格朗日形式为：

$$L_\rho(W, G, \lambda) = f(W) + I_{\mathcal{C}}(G) + \frac{\rho}{2}\|W - G + \lambda\|^2 - \frac{\rho}{2}\|\lambda\|^2$$

ADMM 算法将上述问题分成三个子问题进行求解，即

$$\begin{aligned} W^{k+1} &:= \arg\min_{W} L_\rho(W, G^k, \lambda^k) \\ G^{k+1} &:= \arg\min_{G} L_\rho(W^{k+1}, G, \lambda^k) \\ \lambda^{k+1} &:= \lambda^k + W^{k+1} - G^{k+1} \end{aligned}$$

与其它算法不同的是，我们在实数空间和离散空间分别求解，然后通过拉格朗日乘子的更新将两组解联系起来。

第一个子问题需要找到一个网络权重最小化

$$L_\rho(W, G^k, \lambda^k) = f(W) + \frac{\rho}{2}\|W - G^k + \lambda^k\|^2$$

在实验中我们发现使用常规的梯度下降算法求解这个问题收敛速度很慢。在这里我们使用 Extra-gradient 算法来对这个问题进行求解。Extra-gradient 算法包含两个基本步骤，分别是：

$$W^{(p)} := W - \beta_p \partial_W L(W),$$
$$W^{(c)} := W - \beta_c \partial_W L(W^{(p)})$$

第二个子问题在离散空间中进行优化。通过简单的数学变换第二个子问题可以写成：

$$\min_{Q_i, \alpha_i} \quad \|V_i - \alpha_i \cdot Q_i\|^2$$
$$\text{s.t.} \quad Q_i \in \{0, \pm 1, \pm 2, \cdots, \pm 2^N\}^{d_i}$$

该问题可以通过迭代优化的方法进行求解。当 a 或 Q 固定时，很容易就可以获得 Q 和 a 的解析解。

实验结果

ImageNet 图像识别：我们分别在 Alexnet、VGG16、Resnet18、Resnet50、GoogleNet 等五个主流的 CNN 框架上验证了所提出的算法。实验中我们分别尝试了 Binary 网络、Ternary 网络、{−2, −1, 0, 1, 2}、{−4, −2, −1, 0, 1, 2, 4} 四种形式。在 Imagenet 上 Top−1 和 Top−5 准确度结果如下：

Alexnet 和 VGG16：

	Accuracy	Binary	BWN	Ternary	TWN	{-2, +2}	{-4, +4}	Full Precision
AlexNet	Top-1	**0.570**	0.568	**0.582**	0.575	0.592	0.600	0.600
	Top-5	**0.797**	0.794	**0.806**	0.798	0.818	0.822	0.824
VGG-16	Top-1	**0.689**	0.678	**0.700**	0.691	0.717	0.722	0.711
	Top-5	**0.887**	0.881	**0.896**	0.890	0.907	0.909	0.899

Resnet：

	Accuracy	Binary	BWN	Ternary	TWN	{-2, +2}	{-4, +4}	Full Precision
Resnet-18	Top-1	**0.648**	0.608	**0.670**	0.618	0.675	0.680	0.691
	Top-5	**0.862**	0.830	**0.875**	0.842	0.879	0.883	0.890
Resnet-50	Top-1	**0.687**	0.639	**0.725**	0.656	0.739	0.740	0.753
	Top-5	**0.886**	0.851	**0.907**	0.865	0.915	0.916	0.922

GoogleNet

Accuracy	Binary	BWN	Ternary	TWN	{-2, +2}	{-4, +4}	Full Precision
Top-1	**0.603**	0.590	**0.631**	0.612	0.659	0.663	0.687
Top-5	**0.832**	0.824	**0.854**	0.841	0.873	0.875	0.889

其中 BWN[1] 和 TWN[2] 为我们对比的两种 Binary 网络和 Ternary 网络量化方法。从这些结果可以看出，在各个网络框架下，我们的算法都显著超过对比算法。同时，当比特数达到 3 时，量化之后的网络精度相比于原始网络几乎可以达到无损。在 Alexnet 和 VGG16 这两个冗余度比较高的网络上，量化之后的网络甚至可以取得超过原始网络的精度，这是因为量化操作可以起到一个正则的作用，从而提高这类网络的泛化性能。

Pascal VOC 目标检测：我们在 SSD 检测框架下对算法进行验证，分别采用了 VGG16+SSD 和 Darknet+SSD 两种网络结构。对于检测任务，尝试了 Ternary 网络和 {−4, −2, −1, 0, 1, 2, 4} 两种量化形式。实验结果如下：

mAP	Darknet+SSD	VGG16+SSD
Ternary	0.609 (0.621)	0.762
{-4, +4}	0.624 (0.639)	0.776
Full Precision	0.642	0.778

对于 Darknet 我们使用了两种设置，第一种设置中所有的权重进行相同的量化；第二种设置中，1x1 的卷积核使用 INT8 量化，即括号中的结果。和识别中的结果类似，在 VGG+SSD 结构中，我们的算法几乎可以做到无损压缩。

参考文献

[1] Rastegari, M.; Ordonez, V.; Redmon, J.; and Farhadi, A. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. *European Conference on Computer Vision*.

[2] Li, F.; Zhang, B.; and Liu, B. 2016. Ternary weight networks. *arXiv preprint arXiv:1605.04711*.

一种基于词尾预测的提高英俄翻译质量的方法

Improved English to Russian Translation by Neural Suffix Prediction

作者（中英文）：宋楷 Kai Song　张岳 Yue Zhang　张民 Min Zhang　骆卫华 Weihua Luo

论文下载地址：https://arxiv.org/abs/1801.03615

摘要

神经网络翻译模型受限于其可以使用的词表大小，经常会遇到词表无法覆盖源端和目标端单词的情况，特别是当处理形态丰富的语言（例如俄语、西班牙语等）的时候，词表对全部语料的覆盖度往往不够，这就导致很多“未登录词”的产生，严重影响翻译质量。

已有的工作主要关注在如何调整翻译粒度以及扩展词表大小两个维度上，这些工作可以减少“未登录词”的产生，但是语言本身的形态问题并没有被真正研究和专门解决过。

我们的工作提出了一种创新的方法，不仅能够通过控制翻译粒度来减少数据稀疏，进而减少“未登录词”，还可以通过一个有效的词尾预测机制，大大降低目标端俄语译文的形态错误，提高英俄翻译质量。通过和多个比较有影响力的已有工作（基于 subword 和 character 的方法）对比，在 5000 万量级的超大规模的数据集上，我们的方法可以成功的在基于 RNN 和 Transformer 两种主流的神经网络翻译模型上得到稳定的提升。

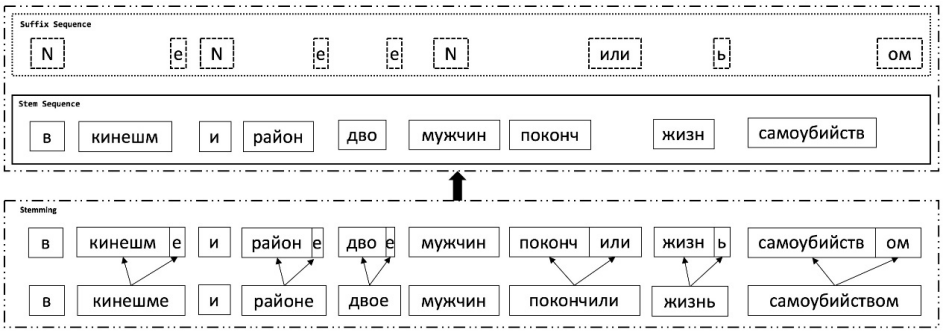
研究背景

近年来，神经网络机器翻译 (Neural Machine Translation, NMT) 在很多语种和场景上表现出了明显优于统计机器翻译 (Statistic Machine Translation, SMT) 的

效果。神经网络机器翻译将源语言句子编码 (encode) 到一个隐状态 (hidden state)，再从这个隐状态开始解码 (decode)，逐个生成目标语言的译文词。NMT 系统会在目标端设置一个固定大小的词表，解码阶段的每一步中，会从这个固定大小的词表中预测产生一个词，作为当前步骤的译文词。受限于计算机的硬件资源限制，这个词表往往不会设的很大（一般是 3 万 -5 万）。并且，随着词表的增大，预测的难度也会相应的增加。基于词 (word) 的 NMT 系统经常会遭遇“未登录词”(Out of vocabulary, OOV) 的问题，特别是目标端是一个形态丰富 (Morphologically Rich) 的语言时，这个问题会更加严重。以“英 – 俄”翻译为例，俄语是一种形态非常丰富的语言，一个 3-5 万的词表往往不能覆盖俄语端的所有词，会有很多 OOV 产生。OOV 的出现对翻译质量的影响是比较大的。

针对这个问题，有很多方法尝试解决。其中一些方法会从翻译粒度的角度出发 (translation granularity)，另外还有一些方法尝试有效的扩展目标端词表大小。这些方法虽然能有效的将少 OOV，但是这些方法并没有对目标端语言的形态 (morphology) 进行专门的建模。

对于俄语这种形态丰富的语言，词干 (stem) 的个数会比词的个数少很多，因此很自然的，我们会想到要对词干和词尾 (suffix) 分别进行建模。我们设计实现了一种方法，在解码时每一个解码步骤 (decoding step) 中，分别预测词干和词尾。训练阶段，目标语言端会使用两个序列，分别是词干序列和词尾序列。词干序列和词尾序列的生成过程如下图所示：



（词干序列和词尾序列的生成，“N”表示词干和词本身相同，即这个词没有词尾）

通过这种方式，数据稀疏问题会得到缓解，因为词干的种类会显著小于词的种类，而词尾的种类只有几百种。

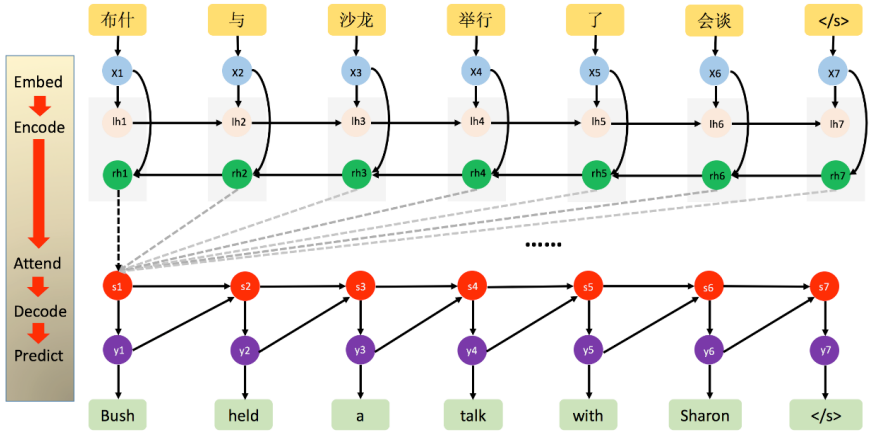
相关工作

基于子词 (subword) 的和基于字符 (character) 的这两种方法，从调整翻译粒度的角度出发来帮助缓解目标端形态丰富语言的翻译问题。一种基于子词的方法利用 BPE(Byte Pari Encoding) 算法来生成一个词汇表。语料中经常出现的词会被保留在词汇表中，其他的不太常见的词则会被拆分成一些子词。由于少数量的子词就可以拼成全部不常见的词，因此 NMT 的词表中只保留常见词和这些子词就可以了。还有一种基于字符的 NMT 系统，源端句子和目标端句子都会表示为字符的序列，这种系统对源端形态丰富的语言可以处理的比较好，并且通过在源端引入卷积神经网络 (convolutional neural network, CNN)，远距离的依赖也可以被建模。上述两种方式虽然可以缓解数据稀疏，但是并没有专门对语言的形态进行建模，子词和字符并不是一个完整的语言学单元 (unit)。

还有一些研究工作是从如何有效的扩大目标端词汇表出发的，例如在目标端设置一个很大的词汇表，但是每次训练的过程中，只在一个子表上进行预测，这个子表中包含了所有可能出现的译文词。这种方法虽然可以解决未登录词的问题，但是数据稀疏问题仍然存在，因为低频的词是未被充分训练的。

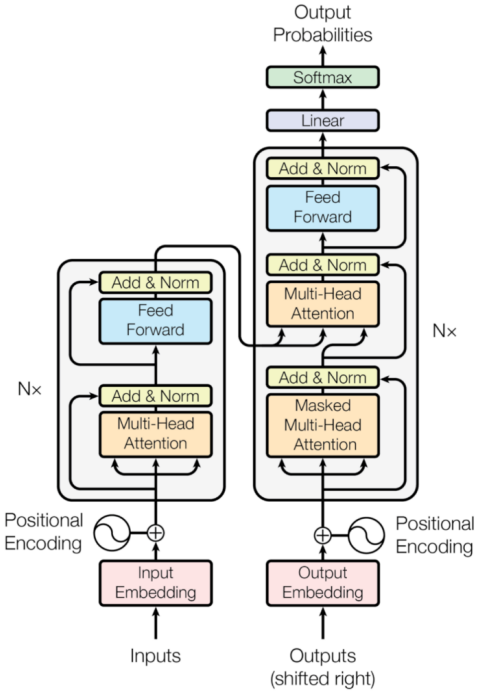
神经网络机器翻译

本文在两种主要的神经网络翻译系统上验证了“基于词尾预测”的方法的有效性，分别是基于递归神经网络的机器翻译 (Recurrent Neural Network Based, RNN-based) 和谷歌在 17 年提出的最新的神经网络翻译模型 (Transformer)，详细介绍可以查看相应论文。RNN-based 神经网络机器翻译如下图：



(“Neural Machine Translation by Jointly Learning to Align and Translate”, Bahdanau et al., 2015)

Transformer 的结构如下图：



(“Attention Is All You Need”, Ashish Vaswani et al., 2017)

俄语的词干和词尾

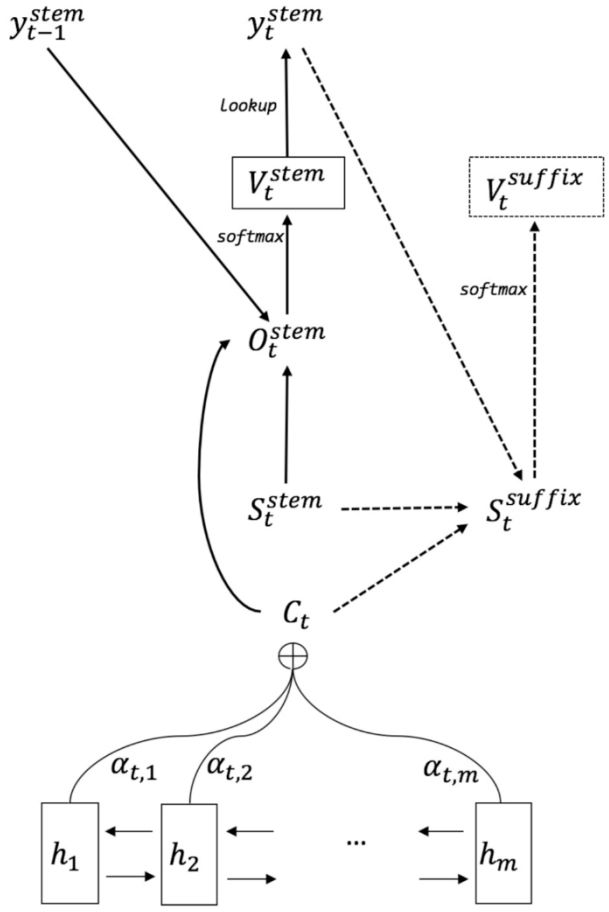
俄语是一种形态丰富的语言，单复数 (number)、格 (case)、阴阳性 (gender) 都会影响词的形态。以名词“ball”为例，“ball”是一个中性词，因此不会随阴阳性的变化而变化，但当单复数、格变化时，会产生如下多种形态：

Case	Singular	Plural
Nominative	мяч	мячи
Genitive	мяча	мячей
Dative	мячу	мячам
Accusative	мяч	мячи
Instrumental	мячом	мячами
Prepositional	мяче	мячах

一个俄语词可以分为两部分，即词干和词尾，词尾的变化是俄语形态变化的体现，词尾可以体现俄语的单复数、格、阴阳性等信息。利用一个基于规则的俄语词干获取工具，可以得到一个俄语句子中每一个词的词干和词尾。

词尾预测网络

在 NMT 的解码阶段，每一个解码步骤分别预测词干和词尾。词干的生成和 NMT 原有的网络结构一致。额外的，利用当前 step 生成的词干、当前 decoder 端的 hidden state 和源端的 source context 信息，通过一个前馈神经网络 (Feedforward neural network) 生成当前 step 的词尾。网络结构如下图：



最后，将生成的词干和词尾拼接在一起，就是当前 step 的译文单词。

实验

我们在 RNN 和 Transformer 上都进行了实验，在 WMT-2017 英俄新闻翻译任务的部分训练语料（约 530 万）上，效果如下图：

	Vocabulary		Coverage		Test set		
Systems	Source	Target	Source	Target	News2014	News2015	News2016
RNN-based + Subword	30K	30K	99.7%	97.0%	19.72(22.59)	16.11	15.41
Fully Character-based	861	853	100%	100%	20.32(25.74)	17.60	15.65
RNN-based + **Suffix Prediction**	30K	30K	99.7%	100%	**21.30(26.22)**	**18.09**	**17.09**
Transformer + Subword	30K	30K	99.7%	97.0%	23.18(26.39)	18.66	18.31
Transformer + **Suffix Prediction**	30K	30K	99.7%	100%	**24.41(29.14)**	**20.54**	**19.62**

其中，Subword 是使用基于子词方法作为 baseline，Fully Character-based 是使用基于字符的 NMT 系统作为 baseline。“Suffix Prediction”是我们的系统。

另外，我们还在电子商务领域的数据上，使用超大规模的语料（5000 万），证明了该方法的有效性，实验结果如图：

	Vocabulary		Coverage		Test set		
Systems	Source	Target	Source	Target	Title	Offer	Comments
RNN-based + Subword	45K	45K	99.8%	100%	17.52	29.78	33.29
RNN-based + **Suffix Prediction**	45K	45K	99.8%	100%	**17.85**	**30.60**	**34.18**

测试集包括商品的标题 (Title)、详情 (Description) 和用户评论 (Comment) 内容，示例如下：

Title
Amazing hot selling air scent machine Large capacity men backpack bags. Strap slash neck women pencil dress
Description
Along with tie shoulder straps, three-quarter sleeves. Compare the detail sizes with yours.
Comment
I did not expect that the backpack is so happy. Thanks for the very quick shipping. I liked the dress. the quality is good.

一些翻译结果的例子：

Source Sentence 1: if you need to return , return has to be done within 15 days after the arrival .
RNN+Subword: если вам нужно вернуться[1], возвращение должно быть сделано в течение 15 дней после прибытия .
Character-based: если вам необходимо вернуться[2], возвращение необходимо сделать в течение 15 дней после прибытия .
RNN+Suffix: если вам нужно вернуть[3], возврат должен быть сделан в течение 15 дней после прибытия .
Reference: если вам нужно вернуть , возврат должен быть сделан в течение 15 дней после получения .

Source Sentence 2: please leave positive feedback and 5 stars .
RNN+Subword: пожалуйста , оставьте положительные[1] отзывы и 5 звезды .
Character-based: пожалуйста , оставьте положительные[2] обратные сведения и 5 звезд[3].
RNN+Suffix: пожалуйста , оставьте положительный[4] отзыв и 5 звезд .
Reference: пожалуйста , оставьте положительный отзыв и 5 звезд .

Source Sentence 3: the center operated in moscow for more than 22 years .
RNN+Subword: центр работает[1] в москве более 22 лет .
Character-based: центр действует[2] в москве на протяжении более 22 лет .
RNN+Suffix: центр работал[3] в москве более 22 лет .
Reference: центр работал в москве более 22 лет .

第一个例子中，标号为 1 和 2 的俄语词的形态代表着这个词是一个反身动词 (reflexive verb)，反身动词的直接宾语和主语是同一个事物，换句话说，反身动词的施事者和受事者是同一个事物。从源端句子中可以看出，“return”的施事者是购买商品的人，受事者是某个要退还的商品，因此 1 和 2 的译文词是错误的。3 的译文词是正确的，它的词尾代表着它是一个不定式动词 (infinitive verb)，这个不定式动词是可以有宾语的。在第二个例子中，标号 1 和 2 代表复数形式，4 代表单数。第三个例子中，3 代表过去时，1 和 2 代表现在时。上面的例子中，相比于基于子词和基于字符的模型，我们的模型可以产生更正确的俄语形态。

总结

我们提出了一种简单、有效的方法来提高目标端是形态丰富语言 (例如“英 – 俄”) 的 NMT 系统的翻译质量。在解码阶段的每一个步骤中，首先生成词干，然后生成词尾。我们在两种 NMT 模型 (RNN-based NMT 和 Transformer) 上，和基于子词 (subword) 和字符 (character) 的方法进行了对比，证明了方法的有效性。我们使用了大规模 (530 万) 和超大规模 (5000 万) 的语料，在新闻和电子商务两个领域上进一步这种方法可以带来稳定的提升。在我们的工作中，词尾在 NMT 中首次被专门的建模。

火箭发射：一种有效的轻量网络训练框架

Rocket Launching: A Universal and Efficient Framework for Training Well-performing Light Net

主要作者（中英文）：周国睿 Guorui Zhou　范颖 Ying Fan　卞维杰 Weijie Bian
朱小强 Xiaoqiang Zhu　盖坤 Kun Gai

附论文下载链接：https://arxiv.org/abs/1708.04106

摘要

像点击率预估这样的在线实时响应系统对响应时间要求非常严格，结构复杂，层数很深的深度模型不能很好的满足严苛的响应时间的限制。为了获得满足响应时间限制的具有优良表现的模型，我们提出了一个新型框架：训练阶段，同时训练繁简两个复杂度有明显差异的网络，简单的网络称为轻量网络（light net），复杂的网络称为助推器网络（booster net），相比前者，有更强的学习能力。两网络共享部分参数，分别学习类别标记，此外，轻量网络通过学习助推器的 soft target 来模仿助推器的学习过程，从而得到更好的训练效果。测试阶段，仅采用轻量网络进行预测。我们的方法被称作“火箭发射”系统。在公开数据集和阿里巴巴的在线展示广告系统上，我们的方法在不提高在线响应时间的前提下，均提高了预测效果，展现了其在在线模型上应用的巨大价值。

研究背景

响应时间直接决定在线响应系统的效果和用户体验。比如在线展示广告系统中，针对一个用户，需要在几 ms 内，对上百个候选广告的点击率进行预估。因此，如何在严苛的响应时间内，提高模型的在线预测效果，是工业界面临的一个巨大问题。

已有方法介绍

目前有 2 种思路来解决模型响应时间的这个问题：一方面，可以在固定模型结构和参数的情况下，用计算数值压缩来降低 inference 时间，同时也有设计更精简的模型以及更改模型计算方式的工作，如 Mobile Net 和 ShuffleNet 等工作；另一方面，利用复杂的模型来辅助一个精简模型的训练，测试阶段，利用学习好的小模型来进行推断，如 KD, MIMIC。这两种方案并不冲突，在大多数情况下第二种方案可以通过第一种方案进一步降低 inference 时间，同时，考虑到相对于严苛的在线响应时间，我们有更自由的训练时间，有能力训练一个复杂的模型，所以我们采用第二种思路，来设计了我们的方法。

研究动机及创新性

火箭发射过程中，初始阶段，助推器和飞行器一同前行，第二阶段，助推器剥离，飞行器独自前进。在我们的框架中，训练阶段，有繁简两个网络一同训练，复杂的网络起到助推器的作用，通过参数共享和信息提供推动轻量网络更好的训练；在预测阶段，助推器网络脱离系统，轻量网络独自发挥作用，从而在不增加预测开销的情况下，提高预测效果。整个过程与火箭发射类似，所以我们命名该系统为“火箭发射”。

训练方式创新

我们框架的创新在于它新颖的训练方式：

1. 繁简两个模型协同训练，协同训练有以下好处：

a) 一方面，缩短总的训练时间：相比传统 teacer-student 范式中，teacher 网络和 student 网络先后分别训练，我们的协同训练过程减少了总的训练时间，这对在线广告系统这样，每天获得大量训练数据，不断更新模型的场景十分有用。

b) 另一方面，助推器网络全程提供 soft target 信息给轻量网络，从而达到指导轻量网络整个求解过程的目的，使得我们的方法，相比传统方法，获得了更多的指导信息，从而取得更好的效果。

2. 采用梯度固定技术：训练阶段，限制两网络 soft target 相近的 loss，只用于轻量网络的梯度更新，而不更新助推器网络，从而使得助推器网络不受轻量网络的影响，只从真实标记中学习信息。这一技术，使得助推器网络拥有更强的自由度来学习更好的模型，而助推器网络效果的提升，也会提升轻量网络的训练效果。

结构创新

助推器网络和轻量网络共享部分层的参数，共享的参数可以根据网络结构的变化而变化。一般情况下，两网络可以共享低层。在神经网络中，低层可以用来学习信息表示，低层网络的共享，可以帮助轻量网络获得更好的信息表示能力。

方法框架

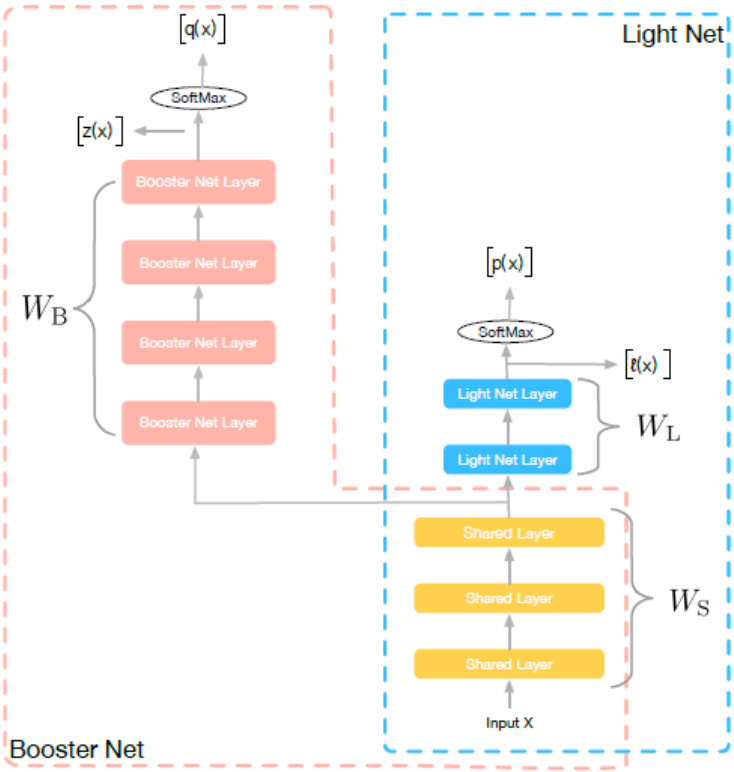


图 1　网络结构

如图 1 所示，训练阶段，我们同时学习两个网络：Light Net 和 Booster Net, 两个网络共享部分信息。我们把大部分的模型理解为表示层学习和判别层学习，表示层学习的是对输入信息做一些高阶处理，而判别层则是和当前子 task 目标相关的学习，我们认为表示层的学习是可以共享的，如 multi task learning 中的思路。所以在我们的方法里，共享的信息为底层参数（如图像领域的前几个卷积层，NLP 中的 embedding），这些底层参数能一定程度上反应了对输入信息的基本刻画。

整个训练过程，网络的 loss 如下：

$$\mathcal{L}(\boldsymbol{x}; \mathbf{W}_{\mathrm{S}}, \mathbf{W}_{\mathrm{L}}, \mathbf{W}_{\mathrm{B}}) = \mathcal{H}(\boldsymbol{y}, \boldsymbol{p}(\boldsymbol{x})) + \mathcal{H}(\boldsymbol{y}, \boldsymbol{q}(\boldsymbol{x})) + \lambda \|\boldsymbol{l}(\boldsymbol{x}) - \boldsymbol{z}(\boldsymbol{x})\|_2^2,$$

Loss 包含三部分：第一项，为 light net 对 ground truth 的学习，第二项，为 booster net 对 ground truth 的学习，第三项，为两个网络 softmax 之前的 logits 的均方误差（MSE），该项作为 hint loss, 用来使两个网络学习得到的 logits 尽量相似。

Co-Training

两个网络一起训练，从而 booster net 会全程监督轻量网络的学习，一定程度上，booster net 指导了 light net 整个求解过程，这与一般的 teacher-student 范式下，学习好大模型，仅用大模型固定的输出作为 soft target 来监督小网络的学习有着明显区别，因为 booster net 的每一次迭代输出 虽然不能保证对应一个和 label 非常接近的预测值，但是到达这个解之后有利于找到最终收敛的解 。

Hint Loss

Hint Loss 这一项在 SNN-MIMIC 中采用的是和我们一致的对 softmax 之前的 logits 做 L2 Loss:

$$\mathcal{L}_{\mathrm{MSE}}(\boldsymbol{x}) = \|\boldsymbol{p}(\boldsymbol{x}) - \boldsymbol{q}(\boldsymbol{x})\|_2^2$$

Hinton 的 KD 方法是在 softmax 之后做 KL 散度，同时加入了一个 RL 领域常

用的超参 temperature T:

$$\mathcal{L}_{\text{mimic}}(\boldsymbol{x}) = \|\boldsymbol{l}(\boldsymbol{x}) - \boldsymbol{z}(\boldsymbol{x})\|_2^2$$

也有一个半监督的工作再 softmax 之后接 L2 Loss:

$$\mathcal{L}_{\text{KD}}(\boldsymbol{x}) = \mathcal{H}(\frac{\boldsymbol{p}(\boldsymbol{x})}{T}, \frac{\boldsymbol{q}(\boldsymbol{x})}{T})$$

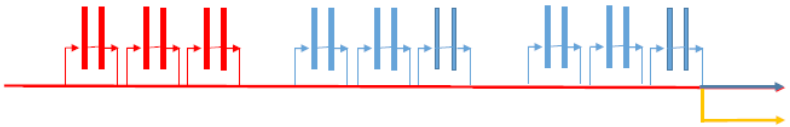
已有的文献没有给出一个合理的解释为什么要用这个 Loss，而是仅仅给出实验结果说明这个 Loss 在他们的方法中表现得好。KD 的 paper 中提出在 T 足够大的情况下，KD 的 Loss $\mathcal{L}_{KD}(x)$ 是等价于 $\mathcal{L}_{mimic}(x)$ 的。我们在论文里做了一个稍微细致的推导，发现这个假设 T 足够大使得 $\mathcal{L}_{KD}(x) = \mathcal{L}_{mimic}(x)$ 成立的情况下，梯度也是一个无穷小，没有意义了。同时我们在 paper 的 appendix 里在一些假设下我们从最大似然的角度证明了 $\mathcal{L}_{mimic}(x)$ 的合理性。

Gradient Block

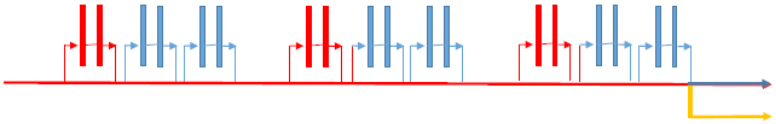
由于 booster net 有更多的参数，有更强的拟合能力，我们需要给他更大的自由度来学习，尽量减少小网络对他的拖累，我们提出了 gradient block 的技术，该技术的目的是，在第三项 hint loss 进行梯度回传时，我们固定 booster net 独有的参数 $\mathbf{W}_b$ 不更新，让该时刻，大网络前向传递得到的 $z(x)$，来监督小网络的学习，从而使得小网络向大网络靠近。

实验结果

实验方面，我们验证了方法中各个子部分的必要性。同时在公开数据集上，我们还与几个 teacher-student 方法进行对比，包括 Knowledge Distillation(KD), Attention Transfer(AT)。为了与目前效果出色的 AT 进行公平比较，我们采用了和他们一致的网络结构宽残差网络（WRN）。实验网络结构如下：



(a) bottom rocket net on wide residual net



(b) interval rocket net on wide residual net

图 2　实验所用网络结构

红色 + 黄色表示 light net, 蓝色 + 红色表示 booster net。(a) 表示两个网络共享最底层的 block，符合我们一般的共享结构的设计。(b) 表示两网络共享每个 group 最底层的 block，该种共享方式和 AT 在每个 group 之后进行 attention transfer 的概念一致。

各创新点的效果

我们通过各种对比实验，验证了参数共享和梯度固定都能带来效果的提升

light	booster	base	rocket (no GB)[1]	rocket (no sharing)[2]	rocket (no joint training)[3]	rocket
WRN-16-1(b)	WRN-40-1	8.77	8.50	8.06	8.04	**7.87**
WRN-16-1(a)	WRN-40-1	8.69	8.30	8.23	8.23	**7.85**

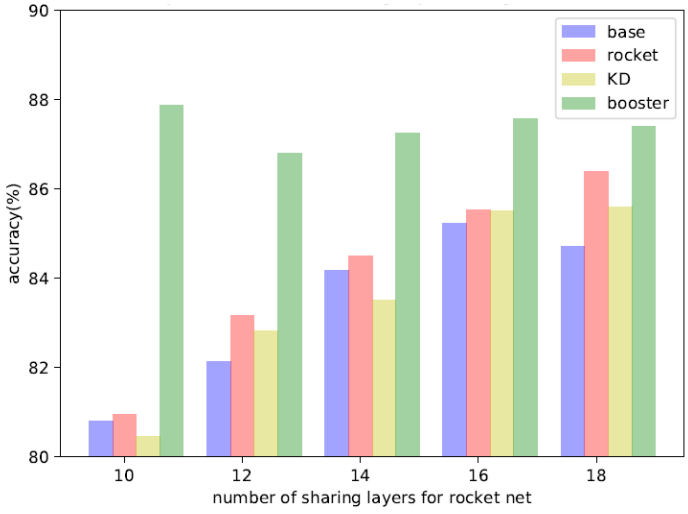
[1] rocket (no GB) means rocket launching without gradient block.
[2] rocket (no sharing) means rocket launching without parameter sharing. [3] rocket (no joint training) means booster net trains first, then light net use some layers of booster to initialize, and use hint loss to learn booster net's logits.

各种 LOSS 效果比较

light	booster	$\mathcal{L}_{\text{mimic}}$	$\mathcal{L}_{\text{MSE}}$	$\mathcal{L}_{\text{KD}}$
WRN-16-1 (b)	WRN-40-1	7.87	8.32	**7.98**
WRN-16-1 (a)	WER-40-1	7.85	8.36	**8.26**

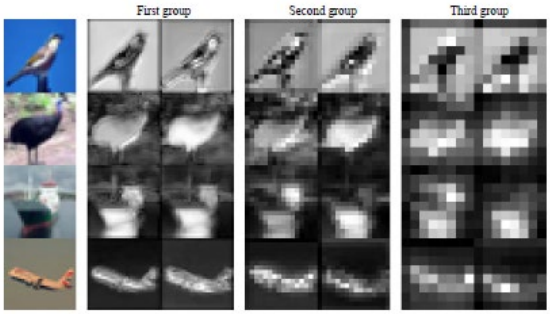
轻量网络层数变化效果图

固定 booster net, 改变 light net 的层数，rocket launching 始终取得比 KD 要好的表现，这表明，light net 始终能从 booster net 中获取有价值的信息。

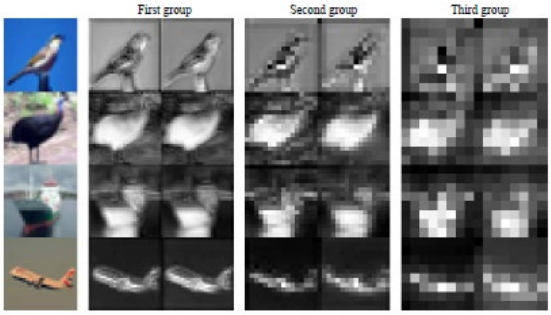


可视化效果

通过可视化实验，我们观察到，通过我们的方法，light net 能学到 booster net 的底层 group 的特征表示。



(a) different group's visualization result on attention transfer



(b) different group's visualization result on rocket launching

公开数据集效果比较

除了自身方法效果的验证，在公开数据集上，我们也进行了几组实验。

在 CIFAR-10 上，我们尝试不同的网络结构和参数共享方式，我们的方法均显著优于已有的 teacher-student 的方法。在多数实验设置下，我们的方法叠加 KD，效果会进一步提升

light	booster	base	rocket (no GB)[1]	rocket (no sharing)[2]	rocket (no joint training)[3]	rocket
WRN-16-1(b)	WRN-40-1	8.77	8.50	8.06	8.04	**7.87**
WRN-16-1(a)	WRN-40-1	8.69	8.30	8.23	8.23	**7.85**

[1] rocket (no GB) means rocket launching without gradient block.
[2] rocket (no sharing) means rocket launching without parameter sharing. [3] rocket (no joint training) means booster net trains first, then light net use some layers of booster to initialize, and use hint loss to learn booster net's logits.

这里 WRN-16-1,0.2M 表示 wide residual net, 深度为 16，宽度为 1，参数量为 0.2M.

同时在 CIFAR-100 和 SVHN 上，取得了同样优异的表现

dataset	light	booster	base	AT	KD	rocket	rocket+KD
SVHN	WRN-16-1, 0.2M(b)	WRN-40-1, 0.6M	3.58	2.99	2.31	2.29	**2.20**
CIFAR-100	WRN-16-1, 0.2M(b)	WRN-40-1, 0.6M	43.7	34.1	36.4	33.3	**33.0**

真实应用

同时，在阿里展示广告数据集上，我们的方法，相比单纯跑 light net, 可以将 GAUC 提升 0.3%.

我们的线上模型在后面的全连接层只要把参数量和深度同时调大，就能有一个提高，但是在线的时候有很大一部分的计算耗时消耗在全连接层 (embedding 只是一个取操作，耗时随参数量增加并不明显)，所以后端一个深而宽的模型直接上线压力会比较大。表格里列出了我们的模型参数对比以及离线的效果对比：

model	# parameters in FC layers	# multiplications in FC layers	inference time of FC Layers	GAUC
base	$576 \times 200 \times 80 \times 2$	131360	7.6 ms	0.632
rocket	$576 \times 200 \times 80 \times 2$	131360	7.6 ms	0.635
booster only	$576 \times 720 \times 360 \times 240 \times 180 \times 90 \times 2$	837900	23.2 ms	0.637

总结

在线响应时间对在线系统至关重要。本文提出的火箭发射式训练框架，在不提高预测时间的前提下，提高了模型的预测效果。为提高在线响应模型效果提供了新思路。目前 Rocket Launching 的框架为在线 CTR 预估系统弱化在线响应时间限制和模型结构复杂化的矛盾提供了可靠的解决方案，我们的技术可以做到在线计算被压缩 8 倍的情况下性能不变。在日常可以减少我们的在线服务机器资源消耗，双十一这种高峰流量场景更是保障算法技术不降级的可靠方案。

句法敏感的实体表示用于神经网络关系抽取

Syntax-aware Entity Embedding for Neural Relation Extraction

主要作者（中英文）：何正球 Zhengqiu He　陈文亮 Wenliang CHEN　张梅山 Meishan Zhang
李正华 Zhenghua Li　张伟 Wei Zhang　张民 Min Zhang，

论文下载地址：https://arxiv.org/abs/1801.03603

摘要

句法敏感的实体表示用于神经网络关系抽取。关系抽取任务大规模应用的一个主要瓶颈就是语料的获取。近年来基于神经网络的关系抽取模型把句子表示到一个低维空间。这篇论文的创新在于把句法信息加入到实体的表示模型里。首先，基于 Tree-GRU，把实体上下文的依存树放入句子级别的表示。其次，利用句子间和句子内部的注意力，来获得含有目标实体的句子集合的表示。

研究背景和动机

关系抽取任务大规模应用的一个主要瓶颈就是语料的获取。远程监督模型通过将知识库应用于非结构化文本对齐来自动构建大规模训练数据，从而减轻对人工构建数据的依赖程度，并使得模型跨领域适应能力得到增强。然而，在利用远程监督构建语料的过程中，仅仅利用实体名称进行对齐，而不同实体在不同关系下应该具有更加丰富的多样的语义表示，从而导致错误标注等问题。因此，一套更加丰富的实体表示显得尤为重要。

另一方，基于语法信息的方法通常作用于两个实体之间的关系上，而语法信息是可以更加丰富实体的表示的。因此，本文基于句法上下文的实体表示来丰富实体在不同关系模式下的语义，并结合神经网络模型处理关系抽取任务。

相关工作介绍

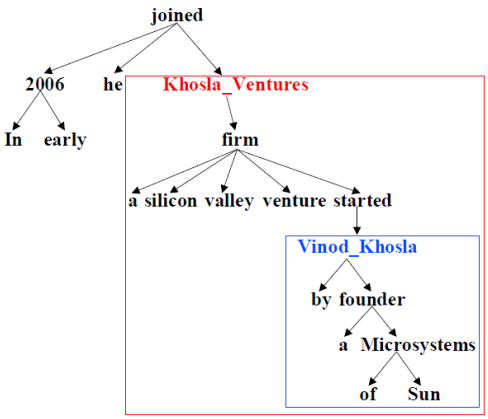
我们把相关的工作大致分成早期基于远程监督的方法和近年来基于神经网络模型两类。

为了解决关系抽取任务严重依赖于标注语料的问题，Mintz et al.(2009) 率先提出了基于远程监督的方法构建标注语料。然而，这样构建的自动标注语料含有大量的噪声。为了缓解语料中噪声带来的影响，Riedel et al.(2010) 将关系抽取看成是一个多实例单类别的问题。进一步的，Hoffmann et al.(2011) 和 Surdeanu et al.(2012) 采取了多实例多类别的策略。同时，采用最短依存路径作为关系的一个语法特征。上述方法典型的缺陷在于模型的性能依赖于特征模板的设计。

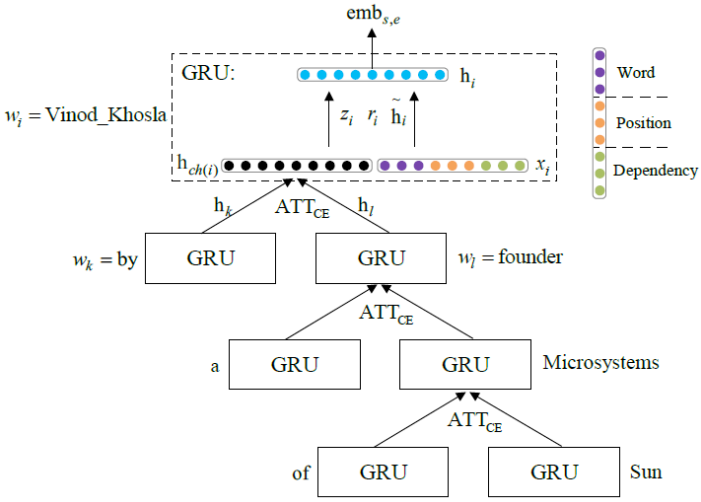
近年来，神经网络被广泛的应用于自然语言处理任务上。在关系抽取领域，Socher et al.(2012) 采用循环神经网络来处理关系抽取。Zeng et al.(2014) 则构建了端到端的卷积神经网络，进一步的，Zeng et al.(2015) 假设多实例中至少有一个实例正确地表示了相应的关系。相比于假设有一个实例表示一对实体的关系，Lin et al.(2016) 通过注意力机制挑选正面的实例更充分的使用了标注语料含有的信息。

以上这些基于神经网络的方法大多数都使用词层次的表示来生成句子的向量表示。另一方面，基于语法信息的表示也受到了众多研究者的青睐，其中最主要的即最短依存路径 (Miwa and Bansal(2016) 和 Cai et al.(2016))。

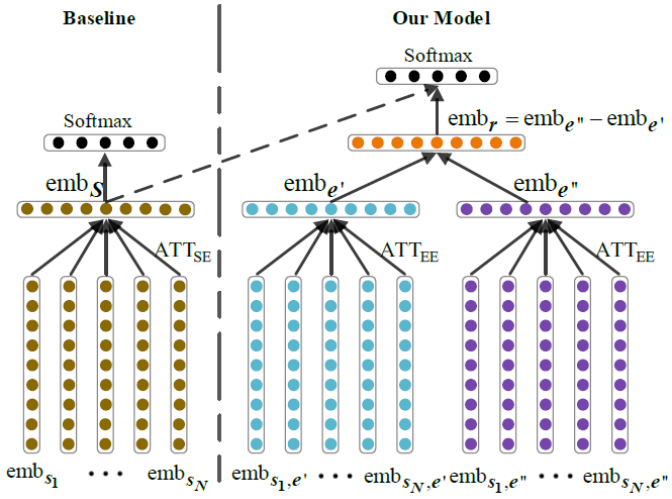
主要方法



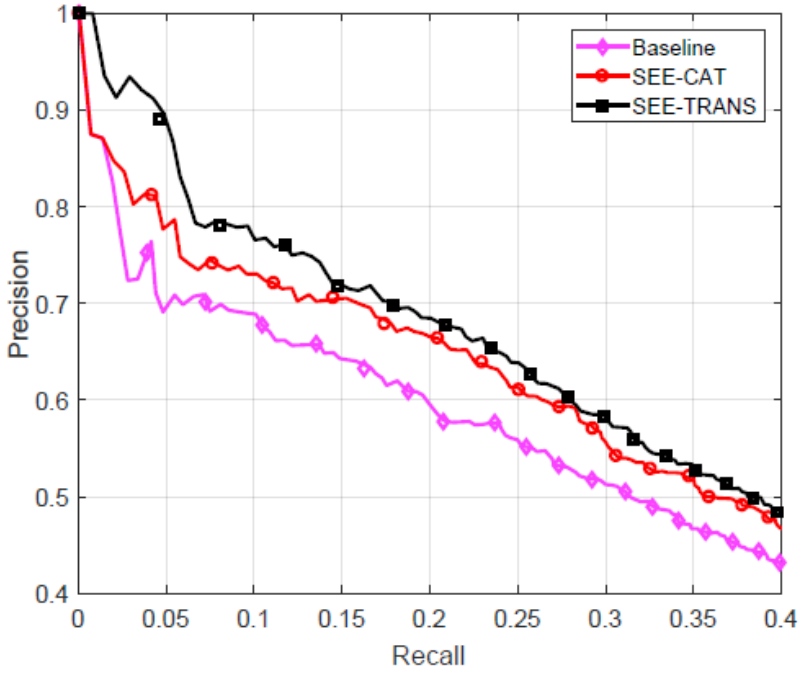
首先，基于依存句法树，利用基于树结构的循环神经网络（Tree-GRU）模型生成实体在句子级别的表示。如上图所示，有别于仅仅使用实体本身，我们能够更好地表达出长距离的信息。具体的实体语义表示如下图所示。我们使用 Tree-GRU 来获得实体的语义表示。



其次，利用基于子节点的注意力机制（ATTCE，上图）和基于句子级别的实体表示注意力机制（ATTEE，下图）来减轻句法错误和错误标注的负面影响。



实验结果



本文在 NYT 语料上进行了实验。最终结果如上图所示。其中，SEE-CAT 和 SEE-TRAINS 分别是本文使用的两种结合三种向量表示（句子的向量表示，两个实体的向量表示）的策略。从图中可以看出，本文提出的模型在相同数据集上取得了比现有远程监督关系抽取模型更好的性能。

总结

本文的实验结果表明，更丰富的命名实体语义表示能够有效地帮助到最终的关系抽取任务。

一种利用用户搜索日志进行多任务学习的商品标题压缩方法

A Multi-task Learning Approach for Improving Product Title Compression with User Search Log Data

主要作者（中英文）：王金刚 Jingang Wang　田俊峰 Junfeng Tian　裘龙 Long Qiu
李生 Sheng Li　郎君 Jun Lang　司罗 Luo Si　兰曼 Man Lan

附论文下载链接：https://arxiv.org/abs/1801.01725

摘要

在淘宝、天猫等电商平台，商家为了搜索引擎优化 (SEO)，撰写的商品标题通常比较冗余，尤其是在 APP 端等展示空间有限的场景下，过长的商品标题往往不能完全显示，只能进行截断处理，严重影响用户体验。如何将原始商品标题压缩到限定长度内，而不影响整体成交是一个极具挑战的任务。以往的标题摘要方法往往需要大量的人工预处理，成本较高，并且未考虑电商场景下对点击率、转化率等指标的特殊需求。基于此，我们提出一种利用用户搜索日志进行多任务学习的商品标题压缩方法。该方法同时进行两个 Sequence-to-Sequence 学习任务：主任务基于 Pointer Network 模型实现从原始标题到短标题的抽取式摘要生成，辅任务基于带有注意力机制的 encoder-decoder 模型实现从原始标题生成对应商品的用户搜索 query。两个任务之间共享编码网络参数，并对两者的对原始标题的注意力分布进行联合优化，使得两个任务对于原始标题中重要信息的关注尽可能一致。离线人工评测和在线实验证明通过多任务学习方法生成的商品短标题既保留了原始商品标题中的核心信息又能透出用户搜索 query 信息，保证成交转化不受影响。

研究背景

商品标题是卖家和买家在电商平台沟通的重要媒介，用户在搜索入口输入 Query 检索，在搜索结果页 (SRP) 浏览商品列表，选择目标商品，最终完成购买。在整条购物成交链路中，商品标题、商品描述、商品图片等各种信息共同影响着用户的购买决策，信息量丰富而不冗长的标题能大大提升终端用户体验。

根据第 40 次《中国互联网络发展状况统计报告》显示，截止 2017 年 6 月，我国手机网民规模已经达到 7.24 亿，网民使用手机上网的比例由 2016 年底的 95.1% 提升至 96.3%。越来越多的在线购买行为已经从 PC 端转移到无线端 (APP)，并且两者之间的差距还在进一步扩大，因此各大电商平台的资源也在往各自 APP 端倾斜。PC 和 APP 最明显的区别在于显示屏幕尺寸，通常智能手机显示屏在 4.5 到 5.5 寸之间，远小于 PC 的屏幕尺寸，对算法和产品设计都有新的要求。

当前淘系商品标题主要由商家负责撰写，为了提高搜索召回和促进成交，商家往往会在标题中堆砌大量冗余词，当用户在手机端进行浏览的时候，过长的商品标题由于屏幕尺寸限制显示不全，只能做截断处理，严重影响用户体验。如图 1 所示，在 SRP 页，商品原始标题显示不完整，只能显示 14 个字左右的短标题，用户如果想获取完整标题，还需要进一步点击进入商品详情页，商品原始标题包含近 30 个字。此外，在个性化推送和推荐场景中，商品短标题作为信息主体，对长度也有一定限制，如何使用尽可能短的文本体现商品的核心属性，引起用户的点击和浏览兴趣，提高转化率，是值得深入研究的问题。

图 1　用户搜索「碎花裙长袖女」，搜索结果页商品原始标题过长无法完整显示，只有点击进入详情页才能看到完整标题。

已有方法介绍

文本摘要（压缩）是自然语言处理中重要研究方向之一。按摘要的生成方式，可以分为抽取式和生成式两种。顾名思义，抽取式方法生成的摘要句子和词均从原文中抽取，而生成式方法更为灵活，摘要中的句子和词并不要求一定从原文中抽取。传统的抽取式摘要方法大致可以分为贪心方法、基于图的方法和基于约束的优化方法等。近年来神经网络的方法也被应用到文本摘要领域并取得显著进步，尤其是生成式摘要方法。业界已有方法都是以压缩文章长度为优化目标实现文本的摘要，电商场景下除了文本压缩率还有其他考量，如何在商品标题长度精简的同时又不影响整体的成交转化率成为一个业界难题。

方法介绍

如图 2 所示，本文提出的多任务学习方法包含两个 Sequence to Sequence 任务，主任务是商品标题压缩，由商品原始标题生成短标题，采用 Pointer Network 模型，通过 attention 机制选取原始标题的关键字输出；辅助任务是搜索 query 生成，

由商品原始标题生成搜索 query，采用带 attention 机制的 encoder-decoder 模型。两个任务共享编码网络参数，并对两者的对原始标题的注意力分布进行联合优化，使得两个任务对于原始标题中重要信息的关注尽可能一致。 辅助任务的引入可以帮助主任务更好地从原始标题中保留更有信息量、更容易吸引用户点击的词。相应地，我们为两个任务构建训练数据，主任务使用的数据为女装类目下的商品原始标题和手淘推荐频道达人改写的商品短标题，辅助任务使用的数据为女装类目下的商品原始标题和对应的引导成交的用户搜索 query。

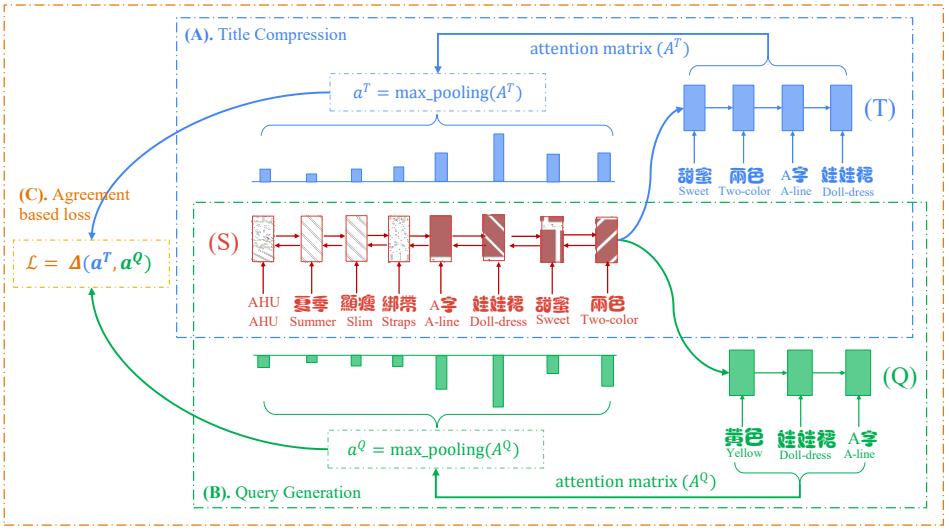


图 2　多任务学习框架，两个 Seq2Seq 任务共享同一个 encoder

主要贡献

1. 本文的多任务学习方法进行商品标题压缩，生成的商品短标题在离线自动评测、人工评测以及在线评测中均超过传统抽取式摘要方法。
2. 端到端的训练方式避免了传统方法的大量人工预处理以及特征工程。
3. 多任务学习中的 Attention 分布一致性设置使得最终生成的商品短标题中能透出原始标题中重要的词，尤其是能引导成交的核心词，对于其他电商场景也有重要意义。

实验结果

我们使用了淘宝女装类目下的商品标题数据进行了实验，对比了五种不同的文本摘要方法。第一种是 baseline 方法，根据目标长度直接截断方法（Trunc.）；第二种是经典的整数线性规划方法（ILP），需要对标题进行分词、NER、Term Weighting 等预处理；第三种是基于 Pointer Network 实验的 encoder-decoder 抽取式方法（Ptr-Net）；第四种是多任务学习方法，直接将两个子任务的损失函数相加作为整体损失函数进行优化（Vanilla-MTL）；第五种是本文提出的考虑 Attention 分布一致性的多任务学习方法（Agree-MTL）。

不同方法的自动评价对比

表1　不同文本摘要方法产生的商品短标题自动评测结果

Method	ROUGE-1	ROUGE-2	ROUGE-L
Trunc.	30.43	19.13	29.00
ILP	48.28	29.84	43.65
Ptr-Net	69.03	55.30	67.98
Vanilla-MTL	65.92	52.94	65.20
Agree-MTL	**70.89**	**56.80**	**69.61**

通过计算生成的短标题和参考短标题之间的三种 ROUGE 分作为自动评测结果，表 1 对比了不同的文本摘要方法。本文提出的多任务学习方法显著超过了其他几种方法。

不同方法的人工评价对比

表2　不同方法产生的商品短标题的人工评测结果

Method	Avg. Accu	Avg. Read	Avg. Info
Trunc.	8.33 %	1.93	1.96
ILP	93.33%	4.63	3.90
Ptr-Net	**98.33 %**	4.66	4.13
Vanilla-MTL	96.67%	4.63	3.90
Agree-MTL	**98.33 %**	**4.80**	**4.66**

表 2 展示了不同方法产生的商品短标题人工评测对比。由于电商场景下商品的核心产品词比较敏感，所以在常见的可读性 (Readability) 和信息完整性 (Informa-tiveness) 指标以外，我们还比较了不同方法产生的短标题中核心产品词是否准确 (Accuracy)。从表 2 结果看，本文提出的方法在三个指标上均超过其他方法。

除了离线的自动评测和人工评测，我们还在真实线上环境中进行了 AB 测试，相比线上原来的 ILP 压缩方法，本文提出的多任务学习方法在 CTR 和 CVR 两个指标上分别有 2.58% 和 1.32% 的提升。

图 3 给出了不同方法产生的商品短标题示例。受预处理结果影响，直接截断和 ILP 两种 baseline 方法生成的短标题流畅度和可读性较差，而 Ptr-Net 和多任务学习属于 Sequence-to-Sequence 方法，生成的短标题在可读性上优于两种 baseline。图 3 左侧例子说明，本文方法生成的短标题会透出用户高频搜索 query 中出现过的词（用户搜索 query 中多使用英文品牌名而非中文品牌名），更容易促进成交。

Original Title		D'ZZIT 地素 秋 专柜 新款 丝绒 拉链 设计 半身 短裙 (D'ZZIT DiSu Autumn Counter New Silk Zipper Designed Half-length Skirt)	MIUCO 女装 夏季 新款 金线 刺绣 高腰 A字 摆 牛仔 背带 连衣裙 (MIUCO Women Summer New Gold-thread Embroidery High-waist A-line Jeans Braces Dress)
Top User Search Queries		D'ZZIT 短裙 丝绒 短裙 D'ZZIT 丝绒 短裙 ...	牛仔裙 连衣裙 牛仔 连衣裙 牛仔裙 ...
Compressed Title	Trunc. ILP Ptr-Net Vanilla-MTL Agree-MTL	D'ZZIT 地素 秋 专柜 新款 地素 D'ZZIT 丝绒 拉链 短裙 地素 丝绒 拉链 半身 短裙 地素 丝绒 拉链 半身裙 D'ZZIT 丝绒 拉链 半身裙	MIUCO 女装 夏季 新款 金线 MIUCO [摆] 背带 连衣裙 MIUCO 背带 连衣裙 MIUCO 牛仔 背带裙 MIUCO 牛仔 背带 连衣裙

图 3　不同方法产生的短标题示例

总结

由于商家 SEO 过度，C2C 电商平台的商品标题通常长度过长且比较冗余且，无法在 APP 端完整展示。为了解决这个问题，本文使用抽取式摘要方法对过长的商品标题进行压缩。(前面这句在前面好像没有提) 传统的摘要方法仅在保持原标题语义的情况下实现标题的压缩，未考虑电商场景下对压缩后商品点击率和成交转化率的影响。电商平台累积了大量用户搜索 query 和商品成交信息，利用这部分数据我们可以更有针对性地对原始长标题进行压缩。因此，我们提出一种多任务学习的标题压缩方法，包含两个序列学习子任务：其中主任务是基于 Pointer Network 模型实现的从原始标题到短标题的抽取式摘要生成，辅任务是基于带有 Attention 机制的 encoder-decoder 模型实现的从原始标题生成对应商品的用户搜索 query，两个任务之间共享编码参数，使得两个子任务在原始标题上的 Attention 分布尽可能一致，对两者的注意力分布进行联合优化，进行联合优化，最终使得主任务生成的短标题在保留原始商品标题中的核心信息的同时，更倾向于透出能促进成交转化的关键词。

离线人工评测和在线实验证明使用本文方法在保证不影响成交转化率的前提下，生成的短标题在可读性、信息完整度、核心产品词准确率上都超过了传统摘要方法。

基于对抗学习的众包标注用于中文命名实体识别

Adversarial Learning for Chinese NER from Crowd Annotations

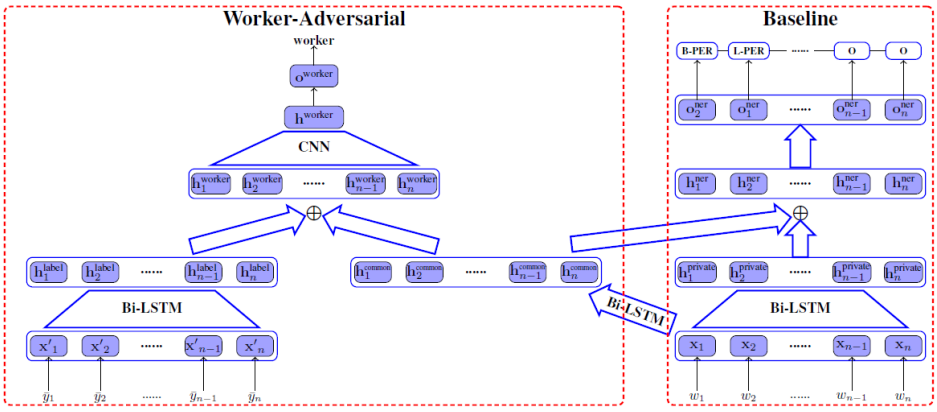
主要作者（中英文）：杨耀晟 YaoSheng Yang　张梅山 Meishan Zhang　陈文亮 Wenliang CHEN
王昊奋 Haofen Wang　张伟 Wei Zhang　张民 Min Zhang

论文下载地址：https://arxiv.org/abs/1801.05147

1. 文章目的与思想

为了能用较低的成本获取新的标注数据，我们采用众包标注的方法来完成这个任务。众包标注的数据是没有经过专家标注员审核的，所以它会包含一定的噪声。在这篇文章中，我们提出一种在中文 NER 任务上，利用众包标注结果来训练模型的方法。受到对抗学习的启发，我们在模型中使用了两个双向 LSTM 模块，来分别学习众包标注数据中的公有信息和属于不同标注员的私有信息。对抗学习的思想体现在公有块的学习过程中，以不同标注员作为分类目标进行对抗学习，从而优化公有模块的学习质量，使之收敛于真实数据（专家标注数据）。我们认为这两个模块学习到的信息对于任务学习都有积极作用，并在最终使用 CRF 层完成 ner 标注。

模型如下：



2. 数据使用

我们在对话数据和电商数据上对模型的性能进行验证。

1）对话数据是由 gowild 公司提供的，我们让 43 位标注员在两万句语料上标注“人名”和“歌名”实体。我们认为这份数据非常适合我们的任务。

（1）若让一位专家标注员标注对话数据，由于他的认知是有限的，所以当他出现标注失误时对模型的影响是比较大的。在这种情况下，多位标注员可以在一定程度上弥补单个标注员对于“歌名”和“人名”的认知不足。例如：歌手“周传雄”，但并不是所有人都知道他的另一个称呼“小刚”。多人的知识面肯定要比一个人来的广。

（2）人机对话语料中包含一定比例的语法错误：

- 你怎么子我都看的手机死机了，在弄自己开门进来干嘛都记得。
- 你说谢谢的诗意哥哥吗?

不同的标注员对于上述句子的语义理解可能是不同的，我们也希望模型能学习到这些特征，使模型能更好收敛到最真实的数据分布，提高模型的泛化能力。

最终，我们的模型在对话数据上取得了近一个点的 F1 提升。

Model	P	R	F1
CRF	89.48	70.38	78.79
CRF-VT	85.16	65.07	73.77
CRF-MA	72.83	**90.79**	80.82
LSTM-CRF	**90.50**	79.97	84.91
LSTM-CRF-VT	88.68	75.51	81.57
LSTM-Crowd	86.40	83.43	84.89
Crowd-NER	89.56	82.70	**85.99**

2）电商数据是由阿里巴巴提供。首先我们让五位标注员对标题数据和用户请求数据进行标注，目标是标注出已定义好的五类实体：品牌、产品、型号、规格、原料，每句标注任务随机分配给两位标注员。对于标注员的标注结果，我们通过样本抽样，分析得到造成标注噪声（标注不一致）的主要原因是不同标注员对于标注规范和每一句标注样例的认识是不同的。特别是在标题数据集中，产品、型号实体的边界定

义非常容易造成标注不一致。

在上述众包标注得到的数据集上训练我们论文中提出的模型，可以得到一个点左右的提升：

Model	Data: EC-MT			Data: EC-UQ		
	P	R	F1	P	R	F1
CRF	75.12	66.67	70.64	65.45	55.33	59.96
LSTM-CRF	75.02	72.84	73.91	71.96	66.55	69.15
LSTM-Crowd	73.81	**75.18**	74.49	67.51	**71.10**	69.26
ALCrowd	**76.33**	74.00	**75.15**	**74.72**	68.60	**71.53**

文章分块解析：

相关工作

（1）序列标注：早期用来处理序列标注问题的模型都十分依赖人工设计的特征模板，例如：HMM, MEMM 和 CRF 模型，模型的性能很大程度上受限于特征模板的质量。神经网络热潮来临后，一个成熟的新模型被广泛应用：它使用双向 LSTM 来提取序列特征，并用 CRF 解码，在序列任务上取得了显著成果，这也是我们文章中的 baseline 模型。

（2）对抗训练：对抗网络最早被成功的应用在计算机视觉领域。近几年，“对抗”这一概念也被引入到 NLP 任务中，分别在跨语言、跨领域和多任务学习中取得突破。在这些任务中使用“对抗学习”，目的就在于学习到训练语料中的“共有特征”。我们的工作也是以这一目的为出发点，希望通过对抗学习的方式，让模型能分辨出“众包”数据中的“标注噪声”。

（3）众包标注模式：为了能在短时间内以较低成本获取标准语料，我们采用众包标注的模式，具体得到的数据情况见上面的“数据使用”。

Baseline

在文章的所有实验中，我们使用 BIOE 的标签集合。首先，我们训练 CRF 作为传统 baseline 模型。随后，尝试将序列特征映射到更高维度，也就是用 LSTM 模块

提取特征。在中文任务中，输入单位为 char（字符），每个字符经过 lookup-table 映射成向量后，经过**双向**的 LSTM 层提取特征：

$$\mathbf{h}_t^{\text{ner}} = \mathbf{W}\mathbf{h}_t^{\text{private}} + \mathbf{b},$$

where $\mathbf{W}$ and $\mathbf{b}$ are both model parameters.

$$\mathbf{h}_t^{\text{private}} = \overrightarrow{\mathbf{h}}_t \oplus \overleftarrow{\mathbf{h}}_t.$$

最终用 CRF 层进行解码，使模型能更好得学习标签之间的依赖关系：

$$\mathbf{o}_t^{\text{ner}} = \mathbf{W}^{\text{ner}}\mathbf{h}_t^{\text{ner}}, \quad t \in [1, n]$$

$$\text{score}(\mathbf{X}, \mathbf{y}) = \sum_{t=1}^{n} (\mathbf{o}_{t,y_t} + T_{y_{t-1},y_t})$$

$$\mathbf{y}^{\text{ner}} = \arg\max_{\mathbf{y}} \big(\text{score}(\mathbf{X}, \mathbf{y})\big),$$

这一部分的 loss 为：

$$\text{loss}(\Theta, \mathbf{X}, \bar{\mathbf{y}}) = -\log p(\bar{\mathbf{y}}|\mathbf{X}),$$

优化目标为最小化这个 loss 值。

对抗学习部分：Worker Adversarial

我们使用的是众包数据作为训练语料，数据集中存在一定量的标注错误，即“噪声”。这些标注不当或标注错误都是由标注员带来的。不同标注员对于规范的理解和认识面是不同的，我们可以认为一位标注质量高的标注员的标注结果和专家标注员是非常相近的。对抗学习模块如下：

1）baseline 中的 BiLSTM 称为“private”，它的学习目标是拟合多为标注员的

独立分布；再加入一个名为“common”的 BiLSTM 模块，common 与 private 的输入相同，它的作用是学习标注结果之间的共有特征：

$$\mathbf{h}_1^{\text{common}}\mathbf{h}_2^{\text{common}}\cdots\mathbf{h}_n^{\text{common}} = \text{Bi-LSTM}(\mathbf{x}_1\mathbf{x}_2\cdots\mathbf{x}_n).$$

2）再引入一个新的 BiLSTM 模块，名为“label”，以**当前训练样例的标注结果序列**为输入。

$$\mathbf{h}_1^{\text{label}}\mathbf{h}_2^{\text{label}}\cdots\mathbf{h}_n^{\text{label}} = \text{Bi-LSTM}(\mathbf{x'}_1\mathbf{x'}_2\cdots\mathbf{x'}_n).$$

3）分别将 common 和 private 模块的输出合并，作为 ne 部分的输入：

$$\mathbf{h}_t^{\text{ner}} = \mathbf{W}(\mathbf{h}_t^{\text{common}} \oplus \mathbf{h}_t^{\text{private}}) + \mathbf{b},$$

最后用 CRF 解码，公式与 baseline 相同，不再贴出。

4）label 和 common 的输出合并，再输入 CNN 进行特征提取，最终对标注员进行分类：

$$\begin{aligned}
\mathbf{h}_t^{\text{worker}} &= \mathbf{h}_t^{\text{common}} \oplus \mathbf{h}_t^{\text{label}} \\
\tilde{\mathbf{h}}_t^{\text{worker}} &= \tanh(\mathbf{W}^{\text{cnn}}[\mathbf{h}_{t-2}^{\text{worker}}, \mathbf{h}_{t-1}^{\text{worker}}, \cdots, \mathbf{h}_{t+2}^{\text{worker}}]) \\
\mathbf{h}^{\text{worker}} &= \text{max-pooling}(\tilde{\mathbf{h}}_1^{\text{worker}}\tilde{\mathbf{h}}_2^{\text{worker}}\cdots\tilde{\mathbf{h}}_n^{\text{worker}})
\end{aligned}$$

$$p(\bar{z}|\mathbf{X}, \bar{\mathbf{y}}) = \frac{\exp(\mathbf{o}_{\bar{z}}^{\text{worker}})}{\sum_z \exp(\mathbf{o}_z^{\text{worker}})},$$

要注意的是，我们希望标注员分类器最终失去判断能力，所以它在优化时要反向更新：

$$\begin{aligned}
\mathrm{R}(\Theta, \Theta', \mathbf{X}, \bar{\mathbf{y}}, \bar{z}) &= \text{loss}(\Theta, \mathbf{X}, \bar{\mathbf{y}}) - \text{loss}(\Theta, \Theta', \mathbf{X}) \\
&= -\log p(\bar{\mathbf{y}}|\mathbf{X}) + \log p(\bar{z}|\mathbf{X}, \bar{\mathbf{y}}),
\end{aligned}$$

CoChat：聊天机器人人机协作框架

CoChat: Enabling Bot and Human Collaboration for Task Completion

Xufang Luo[†]*, Zijia Lin[‡], Yunhong Wang[†] and Zaiqing Nie[§]
[†]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China
[‡]Microsoft Research, Beijing, China
[§]Alibaba AI Labs, Beijing, China
{luoxufang,yhwang}@buaa.edu.cn; zijlin@microsoft.com; zaiqing.nzq@alibaba-inc.com

论文原文链接：

https://102.alibaba.com/downloadFile.do?file=1517882502971/CoChat%20Enabling%20Bot%20and%20Human%20Collaboration%20for%20Task%20Completion%20.pdf

摘要

聊天机器人（chatbots）近来在产业界和学术界都得到了很大的关注。对于产业界中大多数用于完成任务的聊天机器人而言，人类干预是避免其在复杂的现实世界中出错的唯一方法。但是，就我们所知，目前还不存在对任务完成型聊天机器人与人类工作者协作建模的研究工作。我们在本论文中介绍了 CoChat，这是一种能够实现聊天机器人与人类工作者之间有效协作的对话管理框架。在 CoChat 中，人类工作者可以随时引入新的动作来应对之前未曾见过的情况。我们提出了一种记忆增强型分层 RNN（MemHRNN），以在 CoChat 中处理由即时引入新动作所导致的单样本学习（one-shot learning）难题。我们在真实世界数据集上进行了大量实验，结果很好地表明 CoChat 能够大量减轻人类工作者的工作量；与其它之前最先进的框架相比，CoChat 也能得到更好的用户满意率。

引言

任务完成型聊天机器人得到了产业界和学术界的很大关注。它们的目标是通过更

自然的交互方式（即对话）来帮助用户完成特定任务（比如预订电影票）。任务完成型聊天机器人（简称 bot）的重点是成功完成任务，同时实现较高的用户满意度。对于这样的任务完成型 bot 而言，优良的对话管理器是一大关键组件。对话管理器以语言理解的结果为输入，然后决定应该采取哪种动作（比如询问电影名称）。因此，对话管理器直接控制对话流程，决定任务完成得成功或失败，并且还会影响用户满意度（Lee et al. 2010; Young et al. 2013）。

尽管用于开发自动对话管理器的方法有很多，但这些自动系统存在潜在的问题，因为它们没有与人类工作者一样的经验来帮助避免会对用户产生负面影响的严重错误，比如任务失败或让用户厌烦。目前，人类干预是在复杂真实世界环境中避免这种错误的唯一方法（Saunders et al. 2017）。此外，人类工作者已经深入参与了很多常见的任务完成型系统，比如呼叫中心。因此，建模 bot 与人类工作者之间的协作是一种合理的方法。这甚至可能是在很多真实世界应用中学习和使用对话管理器的唯一实用的方法。

在这篇论文中，我们提出了一种名叫 CoChat 的实用型对话管理框架，它可以实现 bot 与人类工作者之间的有效协作。就我们所知，我们的工作是首个建模任务完成型 bot 与人类工作者协作的研究，这种协作对很多真实世界应用而言都是必需的和合理的。在 CoChat 中，人类工作者可以随时干预学习过程；而且通过从标注过的对话日志中学习，从人类工作者的反馈中学习，和从用户的反馈中学习，对话管理器可以得到持续不断的改进。如图 1 所示，对话管理器首先通过监督学习进行初始化。之后，该 bot 可以与人类工作者协作，向他们提供动作建议以降低他们的工作量以及通过在线学习（online learning）持续不断地根据他们的反馈进行学习。当没有人类工作者时（比如下班时），如果用户愿意尝试该 bot，那么该 bot 可以直接与用户交互，其对话管理器可以通过强化学习得到进一步改进。简而言之，对话管理器可以不断得到提升以实现用户满意度的最大化以及在协作过程中减轻人类工作者的工作量。

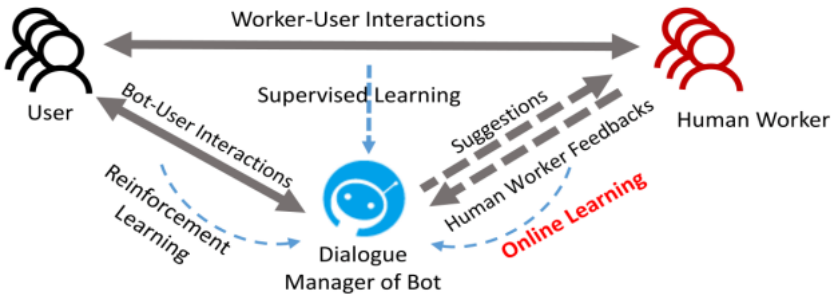


图 1　CoChat 框架示意图

人类工作者的参与使得引入新的动作来处理复杂或未曾见过的情况成为可能，而这又为对话管理器带来学习新动作的难题。在（Lee et al. 2010; Su et al. 2017）等之前的研究中，对话管理器需要固定的动作集合，然后再学习从其中选择一个动作来处理每种情况。但是，当有人类工作者参与进来处理复杂或未曾见过的情况时，这种设定是不现实的，因为为了促进任务完成，可能会对原始动作集之外的新动作有迫切的需求。比如，客户支持中心可能会收到他们最新产品的问题的相关问题，这些问题可能前所未见，所以这样的 bot 无法处理。然后“告诉用户如何处理这些新问题”就成了新的必要动作，而且这些动作将会由负责处理这些情况的人类工作者提供。对话管理器也应该学习这个新动作，以便之后更好地提供动作建议和处理类似的情况。这种学习新动作的难题之前从未得到过很好的研究。尽管之前的研究工作可以通过重新训练它们对应的模型来适应新动作，但强化学习等学习过程往往无法完全准确地重现，因此这个方法有丢失已经积累的习得知识的风险。

为了实现我们提出的 CoChat 框架和更好地处理学习新动作的难题，我们进一步提出了一种全新的对话管理器模型 MemHRNN。我们提出的 MemHRNN 由一个分层循环神经网络（HRNN）（Li, Luong, and Dan 2015; Xie et al. 2016; Yang; and Huang 2016; Sordoni et al. 2015）和一个外部记忆构成。其中 HRNN 将对话历史、语言理解结果和 API 调用结果等外部信息组合起来作为输入，然后输出所有动作的概率分布以用于动作选择。当出现新动作时，该 HRNN 的架构可以相应地改变，而不会丢失在之前的学习过程中积累的知识。最重要的是，我们还在 MemHRNN 中

进一步引入了外部记忆来处理由即时引入新动作所导致的单样本学习难题，即这些新动作出现的次数太少，让 HRNN 难以学习与它们有关的有效策略。具体而言，我们会将新动作的出现记录在记忆中，然后利用它们来推导选择这些新动作的概率，然后这些概率会与 HRNN 的输出合并起来提升动作选择的表现。

我们在两个真实任务上进行了实验：i）预订餐厅；ii）预订电影票。实验结果表明：1）在 CoChat 中，当有人类工作者时，可以实现很高的用户满意度。此外，通过为这两个任务提供动作建议，由 MemHRNN 学习得到的对话管理器可以将工作者的工作量分别减少 91.35% 和 86.32%。2）在连续学习后，所学习到的对话管理器能实现很高的用户满意率，分别能达到人类工作者所实现的用户满意率的 97.04% 和 92.62%。3）当新动作出现时，MemHRNN 可以通过平滑的架构适应和利用外部记忆来快速学习策略以应对这些新动作。

我们的工作的贡献总结如下：

· 我们提出了 CoChat，这是一种实用的对话管理学习框架，能建模 bot 与人类工作者之间的协作来完成任务。

· 我们提出了一种记忆增强型分层 RNN 模型（MemHRNN）来实现 CoChat 框架和处理由即时引入新动作所导致的单样本学习难题。

相关工作

在任务完成型 bot 中，对话管理器决定了在当前对话状态条件下的下一个动作应该是什么（Young 2006; Zhao and Eskenazi 2016）。一般而言，对话状态是根据消息历史、之前的动作、自然语言理解（NLU）结果等信息推导得出的。除了人工设计规则之外，研究者们已经为开发这样的对话管理器提出了多种不同的基于学习的方法。

受非面向任务型 bot（Vinyals and Le 2015; Shang, Lu, and Li 2015）的启发，研究者们已经提出了多种通过监督学习训练对话管理器的方法。Wen et al. (2016) 提出了一种基于神经网络的可训练对话系统和一种收集对话数据的新方法。Bordes and Weston (2016) 报告了一种基于记忆网络的端到端对话系统，可以得到出色但还不完美的表现。Eric and Manning (2017) 描述了一种复制增强型

（copy-augmented）序列到序列架构，该架构也可实现优良的对话管理表现。

对话管理器也可以通过强化学习方法学习得到（Levin, Pieraccini, and Eckert 1998; Williams and Young 2007; Young et al. 2013）。具体而言，该学习过程可以被建模为部分可观察马尔可夫决策过程（POMDP）。这种对话管理器会直接与用户进行交互，然后它能在“试错”过程中根据有延迟的奖励进行学习。但是，对于产业界 bot 而言，通过强化学习从头开始学习对话管理器可能会有风险和无法实现。这种过程可能会执行随机探索，甚至有时候无法学习到可接受的结果，Williams and Zweig (2016) 的实验就表明了这一点。

最近还有研究者提出通过结合监督学习和强化学习来学习对话管理器。Zhao and Eskenazi (2016) 提出了一种结合两者优势的算法，可以实现更快的学习速度。Williams and Zweig (2016) 报告指出使用监督学习进行预训练可以显著加速强化学习的学习速率。Su et al. (2016) 表明监督学习是有效的，而且在监督学习之后再使用强化学习可以进一步实现性能提升——尤其是在高噪声条件下。

在这篇论文中，我们提出了一种名为 CoChat 的新型对话管理学习框架。CoChat 可以为完成任务建模 bot 和人类工作者之间的协作——就我们所知这之前还从未被考虑过。CoChat 不仅像之前的工作一样结合了监督学习和强化学习，而且还结合了在线学习以在 bot- 工作者协作过程中更好地向人类工作者学习。此外，不同于之前的工作预设对话管理器有固定的动作集，CoChat 支持由人类工作者引入的新动作。因为新动作一般出现次数较少，所以学习如何选择它们本质上是一个单样本学习问题（Santoro et al. 2016; Graves, Wayne, and Danihelka 2014）。为了解决这个问题，我们提出了一种记忆增强型分层 RNN 模型（MemHRNN）来实现 CoChat。带有外部记忆的神经网络已经在机器翻译等多种 NLP 问题（Tang et al. 2016; Wang et al. 2016）上被成功用于解决类似的难题。

CoChat 框架

如图 1 所示，我们提出的学习框架 CoChat 能够实现 bot 与人类工作者之间的协作，从而可以最大限度降低失败风险和确保用户满意度。同时，它还有助于最小化协作过程中人类工作者的工作量。简而言之，CoChat 结合了基于可用有标注日志的

监督学习、基于人类工作者给出的反馈的在线学习和基于用户给出的有延迟奖励 / 反馈的强化学习，以便持续不断地改进对话管理器。

具体而言，当只有少量可用的有标注用户 – 工作者对话时，对话管理器首先会通过监督学习进行初始化，从而有一个更高的起点。

然后上线该对话管理器以向人类工作者提供动作建议，实现协作以及帮助他们更高效地工作。一般而言，人类工作者可以毫不费力地选择一个建议的动作，或在建议动作不适宜时输入一个新动作。建议动作的接受 / 拒绝和新输入动作等都是反馈，可以被用于进一步通过在线学习增强该对话管理器。特别是当人类工作者输入新动作时，CoChat 也有望学习它们以便更好地建议动作和处理未来相似的情况。实际上，引入新动作可能会带来单样本学习难题。我们提出的用于实现 CoChat 的模型 MemHRNN 利用了外部记忆来处理这样的难题，具体将在下面论述。

当没有人类工作者（比如下班时）而用户又愿意尝试该 bot 时，对话管理器可以直接与用户交互。在这些情况下，对话管理器可以根据用户给出的有延迟的奖励 / 反馈通过强化学习获得进一步的提升，这些奖励 / 反馈与任务完成的成败、用户满意度等有关。

在 CoChat 中，在通过监督学习初始化之后，又会根据是否有人类工作者与 bot 协作的情况依次执行在线学习和强化学习。每一步都可以根据之前的步骤而获得进一步的提升。因此，对话管理器可以通过监督学习、在线学习和强化学习而获得持续提升。

用于实现 CoChat 的 MemHRNN

为了实现 CoChat 框架，我们提出了一种名为 MemHRNN 的记忆增强型分层 RNN 模型。MemHRNN 由一个基于分层 RNN 的学习模型（即 HRNN）和一个外部记忆构成。其中外部记忆仅在在线学习阶段工作，而 HRNN 则在所有学习阶段都会一直工作。用 θ 表示 HRNN 中的所有参数，θ 在所有学习过程中是共享的而且会持续得到优化。我们将在以下小节中详细阐述 HRNN 和所有学习过程。

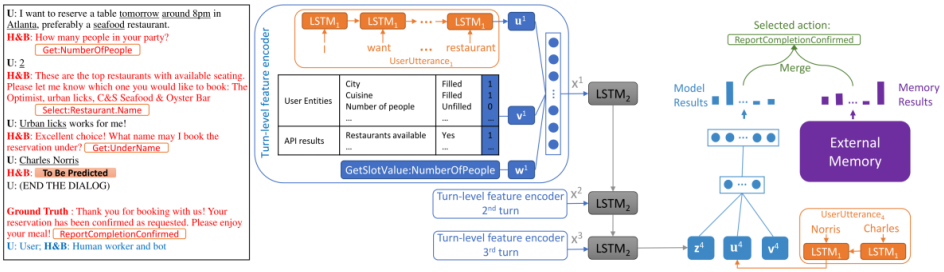


图 2　我们提出的用于实现 CoChat 框架和学习对话管理器的 MemHRNN 模型示意图，另外还给出了一个来自餐厅预定任务的真实完整示例。其中带下划线的词是标记的实体，加框的词是对应话语的动作标签。

用于所有学习过程的 HRNN

MemHRNN 中的 HRNN 由单回合对话特征编码器（turn-level feature encoder）和动作选择器（action selector）构成。任意对话都可被分割为多个回合，每个回合都包含一个用户话语和人类工作者或 bot 所采取的动作。这里会应用单回合对话特征编码器来处理一场对话中的每个回合，以得到该回合对应的特征向量。然后动作选择器会以这种单回合对话层面的特征向量为输入来决定选择哪个动作。下面我们将对这两者进行介绍，其中重要的符号标记总结在表 1 中。

表 1　与第 m 回合对话相关的特征向量

$\mathbf{u}^m$	Feature vector of the user utterance, encoded by $LSTM_1$
$\mathbf{v}^m$	Feature vector of the entity form
$\mathbf{w}^m$	1-hot vector of the taken action
$\mathbf{x}^m$	Integrated feature vector of the turn, encoded by *turn-level feature encoder*
$\mathbf{z}^m$	Feature vector of the previous $(m-1)$ turns, encoded by $LSTM_2$

单回合对话特征编码器设在第 m 回合对话，用户话语 um 被定义为 $u^m = \{w^{m,1}, w^{m,2}, \ldots, w^{m,n_m}\}$，其中 nm 是词的数量。然后再利用如下用 LSTM1 表示的长短期记忆网络（LSTM）（Hochreiter and Schmidhuber 1997）来逐个编码每个词的嵌入向量。

$$\mathbf{u}^{m,j} = LSTM_1(\omega^{m,j}, \mathbf{u}^{m,j-1}) \qquad (1)$$

其中 ωm,j 是词 wm,j 的嵌入向量，um,j−1 是为之前的 (j−1) 个词所编码的特征向量。因此，um 的特征向量 um 是 LSTM1 的最后一个输出，即 $\mathbf{u}^m = \mathbf{u}^{m,n_m}$

对于第 m 回合对话，我们也考虑以下特征。1）实体表格（entity form），收集用于完成任务的信息。如图 2 所示，它一般包含两类信息，即从用户话语中提取的实体和由 API 调用返回的信息。这里我们只简单地将实体表格编码成二元（0 或 1）特征向量 vm，表示是否提取出了实体或 API 调用是否返回了真实 / 可用的数据。2）在第 m 回合所采取的动作（taken action），表示为 na 维的 1−hot 向量 wm，其中 na 是动作的数量。仅有对应于所采取的动作的元素值为 1，其余元素皆为 0。

对于第 m 回合对话，um、vm 和 wm 被连接合并在一起成为特征向量 $\hat{\mathbf{x}}^m = [\mathbf{u}^m, \mathbf{v}^m, \mathbf{w}^m]$，然后再被送入一个全连接层 FN 以学习新的特征向量 xm。

$$\mathbf{x}^m = FN(\hat{\mathbf{x}}^m). \qquad (2)$$

使用 xm 而不用 $\hat{\mathbf{x}}^m$ 的原因是使用 FN 可以帮助更好地保留积累的知识，尤其是当涉及到新动作且需要更改 HRNN 时。这将在后面关于在线学习的小节中更详细地解释。

动作选择器 动作选择器的输入是历史对话回合和当前对话回合，然后再据此预测每个候选动作被选中的概率。

设当前对话回合为第 m 回合。再利用与分层 LSTM（Li, Luong, and Dan 2015; Xie et al. 2016; Yang; and Huang 2016; Sordoni et al. 2015）类似的另一个 LSTM（记为 LSTM2，如下所示）来将之前的 (m−1) 个对话回合逐个编码到向量 zm 中。

$$\mathbf{z}^m = LSTM_2(\mathbf{x}^{m-1}, \mathbf{z}^{m-1}) \qquad (3)$$

其中 zm−1 是编码了之前 (m−2) 回合对话的特征向量，并且可以以一种相同的方式递归式地导出。对于当前回合，因为还没有采取任何动作，所以我们只能导出用户话语的特征向量 um 和实体表格的特征向量 vm，如表 1 中的定义。然后 zm、um 和 vm 会连接合并成一个综合向量 sm=[zm, um, vm]，然后再被送入一个二层全连接神经网络 FN2。注意我们为第二层使用了一个 softmax 函数，以得到用于选择所有候选动作的概率分布 pm：

$$\mathbf{p}^m = FN_2(\mathbf{s}^m) \tag{4}$$

根据日志进行学习：监督学习

当有可用的有标注用户 – 工作者对话日志时，对话管理器通过监督学习进行初始化以模仿人类工作者。具体来说，在这些日志的每一个对话回合中，由人类选择的动作都被视为基本真值（ground-truth），表示为 one-hot 向量$\mathbf{y}^m \in \{0,1\}^{n_a}$，其中 m 是对话回合编号，na 是候选动作的数量。我们使用 HRNN 进行监督学习的目标是最小化 ym 与公式 (4) 中在所有动作上预测的概率分布 pm 之间的交叉熵，正如以下损失函数所示：

$$\mathcal{L}_{SL}(\theta) = \frac{1}{n_d}\sum_{m=1}^{n_d}\mathcal{E}(\mathbf{p}^m) \quad s.t., \mathcal{E}(\mathbf{p}^m) = -\sum_{i=1}^{n_a}\mathbf{y}_i^m \log \mathbf{p}_i^m \tag{5}$$

其中 θ 表示 HRNN 的所有参数，nd 是有标注日志中对话的回合数。

向人类工作者学习：在线学习

当该 bot 与人类工作者开始合作时，其对话管理器可以使用来自人类工作者的反馈获得提升。具体来说，在每个对话回合，该 bot 会向人类工作者建议 k 个动作。这 k 个动作是由 HRNN 与外部记忆共同决定的。人类工作者可以接受其中一个（即在前 n 个建议的动作中选择一个并直接将其用于与用户交互）或拒绝所有并输入另一个可以更好地处理当前对话状态的动作。然后 HRNN 和外部记忆会进行对应更新。

更新 HRNN 根据人类工作者是否输入了已有动作集之外的新动作，HRNN 的更新方式也不一样。

情况 1： 当人类工作者接受了一个建议的动作或输入了一个已有的候选动作时，这样的反馈可以被当作是新标注的日志并被直接用于更新对话管理器。该模型会使用这些新收集的日志持续更新，从而使对话管理器更好地适用于随时间变化的动作选择。当有一小批新收集到的日志时，我们的目标是最小化以下的用于在线学习的损失函数。

$$\mathcal{L}_{OL}(\theta) = \frac{1}{n_o}\sum_{m=1}^{n_o}\mathcal{E}(\mathbf{p}^m) + \lambda||\theta - \theta'||_2 \qquad (6)$$

其中 no 是这一小批新收集的日志中对话回合的数量，E(pm) 是用公式 (5) 中一样的方式定义的交叉熵，θ 表示 HRNN 的所有参数，θ' 表示在收集到该小批量的日志之前 θ 的值。考虑到大多数情况下对话管理策略都是渐进变化的，我们增加了这个第二项以避免 θ 在在线学习过程中急剧变化，这也可以使学习过程更加稳定。这里的 λ 是一个平衡因子，我们根据实践经验将其设为 0.05。

情况 2： 如果人类工作者输入了一个新动作，它就将被加入到候选动作集中，然后 HRNN 的网络架构就需要相应地更新。具体来说，HRNN 的单回合对话特征编码器和输出层将会改变。

对于单回合对话特征编码器的输入层，表 1 中特征向量 wm 的维度将会因新动作而增大 1；而 FN 的输出维度不会改变。然后我们在新添加的维度和 FN 的输出单元之间加入全连接。FN 原来的权重全部都会保留，而在新添加的连接上的权重会被初始化为接近 0 的值，这样在刚加入新动作时，FN 的效果仍能与之前的近乎一样。因为 FN 的输出维度不变，所以 LSTM2（这是编码历史对话回合的关键部分）的架构需要改变。因此在 HRNN 上加入新动作的影响就显著降低了，所以该学习模型可以尽可能多地保留积累的知识。这也是引入 FN 的原因。

至于 HRNN 的输出层（即 FN2 的最后一层），我们增加一个对应新动作的新维度，并且还在它与该层的输入单元之间加入全连接。类似地，为了使 HRNN 的效果

在新动作刚加入时接近之前的效果，我们也将在新加入的连接上的权重初始化为接近于 0 的随机值。

在改变架构之后，HRNN 就将以情况 1 中一样的方式更新。

利用外部记忆 引入外部记忆的原因是为了处理由即时引入新动作所导致的单样本学习难题。具体而言，新动作的出现会被记录在外部记忆 M 中，然后再与 HRNN 一起被用于提供动作建议。下面详述了外部记忆的用法，其中为了简洁省略了对话回合编号 m。

在监督学习阶段中预定义的动作集之外的动作被视为新动作。新动作 a 的出现在记忆中被记录成关键值对 <a,r>，其中 r 表示动作发生的对话状态。因为 HRNN 为某个对话状态所推导出的表征总是会随其参数在优化阶段的更新而变化，所以我们这里利用了一种简化的方法来为记忆推导 r。具体来说，r 由两部分组成，即 r=[v, c]，其中 v 是表 1 的实体表格，c 是一个背景向量（context vector），其是作为最后一个用户话语中词的嵌入向量的平均池化而得出的。

此外，当发生的新动作超出了预定义的阈值时，其对应的记录会被从记忆中删除。实际上，对于这些被删除的新动作，它们的出现应该足以让 HRNN 学习到有关它们的有效策略，使单样本学习难题得到很好的缓解，因此来自外部记忆的帮助对它们而言是微不足道的。这里我们将外部记忆中记录的动作表示为 A。

给定一个对话回合，HRNN 可以像公式 (4) 那样直接预测概率分布 p，以用于选择每个候选动作。而对于记忆，设对话回合的状态表示为$\hat{\mathbf{r}} = [\hat{\mathbf{v}}, \hat{\mathbf{c}}]$且是按记录中对话回合状态一样的方式得到的。我们可以按以下方式将外部记忆与 HRNN 一起利用。

第一步：我们按以下方式计算$\hat{\mathbf{r}}$与任意记录中的情况 r 之间的相似度。

$$Sim(\hat{\mathbf{r}}, \mathbf{r}) = \exp\left(-\frac{\alpha\|\hat{\mathbf{v}} - \mathbf{v}\|_2 + (1-\alpha)\|\hat{\mathbf{c}} - \mathbf{c}\|_2}{2\sigma^2}\right) \quad (7)$$

其中 α 是一个平衡因子，根据我们实验的经验设为 0.1，而 σ 是一个平滑参数。

第二条：我们定义了一个所有候选动作的相似度向量 h，其中 hi 的定义如下。

$$\mathbf{h}_i = \begin{cases} 0 & \text{if } a_i \notin A, \\ \max_{\mathbf{r} \in R(a_i)} Sim(\hat{\mathbf{r}}, \mathbf{r}) & \text{if } a_i \in A. \end{cases} \tag{8}$$

其中 ai 是第 i 个候选动作，R(ai) 表示记录中所有 ai 的出现情况。

因为新动作仅有少数出现情况记录在记忆中，所以应该以一种泛化更弱的有局限性的方式使用它们。也就是说，只有当对话状态 $\hat{\mathbf{r}}$ 非常近似于其对应的记录中出现的某种出现情况时，我们才有可能选择 A 中的一个新动作。否则，我们可能倾向选择其它动作。为了实现这一点，我们推导了折现因子（discounted factor）β =sigmoid(max(h))，其中 max(h) 是 h 的最大值，sigmoid 表示一个 sigmoid 函数。然后我们以如下方式推导一个折现概率分布 q，以选择 A 中的新动作。

$$\mathbf{q}_i = \beta \frac{\exp(\mathbf{h}_i)}{\sum_j \exp(\mathbf{h}_j)} \tag{9}$$

第三步：我们将 q 与来自 HRNN 的概率分布 p 融合成一个综合的概率分布$\hat{\mathbf{p}}$，如下所示。

$$\hat{\mathbf{p}}_i = \begin{cases} \mathbf{p}_i & \text{if } a_i \notin A, \\ \mathbf{q}_i & \text{if } a_i \in A. \end{cases} \tag{10}$$

然后 MemHRNN 将$\hat{\mathbf{p}}$中概率最高的 k 个动作建议给人类工作者。

向用户学习：强化学习

当 bot 有机会自己直接与用户交互和选择动作时，其对话管理器可以使用来自用户的有延迟奖励 / 反馈通过强化学习获得进一步提升，这些奖励 / 反馈一般是在整个对话结束之后收到的而且大都与任务完成的成败、用户满意度等有关。

在强化学习中，对话管理器在任意状态 sm（其中 m 表示对应的对话回合编号）的目标是选择能最大化累积的未来奖励的动作 am。因此，强化学习要按如下方式定

义一个最优的动作 – 价值函数。

$$Q^*(s,a) = \max_{\pi} \mathbb{E}[r^{(m)} + \gamma r^{(m+1)} + \cdots | s^m = s, a^m = a, \pi]$$

其中 γ ∈ [0, 1] 是一个折现因子，r(m) 和 r(m+j) (j=1,2,...) 分别是第 m 个对话回合和后续回合所收到的奖励，π 是最优的对话管理策略，因此 Q*(s,a) 是当观察到 sm 是对话状态 s 时采取动作 a 作为 am 的最大折现累积奖励。

我们使用的 HRNN 类似于深度 Q 网络（DQN）（Mnih et al. 2015），并且已经通过监督学习和在线学习预训练过，以近似最优动作 – 价值函数 Q(s,a; θ)，其中 θ 是 HRNN 的所有参数。然后我们最小化以下损失函数以便为迭代式地更新 θ 执行 Q 学习（Mnih et al. 2015）。

$$\mathcal{L}_{RL}(\theta) = \mathbb{E}_{s,a,r,s'}[(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta))^2]$$

其中 θ^- 表示该算法（Mnih et al. 2015）中目标网络的参数，s' 和 r 分别是在状态 s 采取动作 a 后的新对话状态和收到的奖励，a' 是 s' 下的一个新动作。

总结

尽管监督学习、在线学习和强化学习可以具有不同的损失函数（即 LSL、LOL 和 LRL）和不同的模型参数更新方式，但它们都使用了同样的神经网络（即 θ）来分享它们学习到知识和根据彼此持续改进对话管理器。

实验

数据集和用户模拟器

我们雇佣了对预订餐厅和预订电影票这两项现实任务很熟悉的人类工作者和用户，并收集他们的对话为我们的实验构建了两个数据集。图 2 中有一个完整的对话示例，表 2 列出两个数据集的一些基本统计数据。这样的现实数据集可以更好地评估对话管理器的学习框架和模型的实用性。一般而言，我们将一个动作定义为在某些（可

能为空）槽（slot）上所执行的一项或多项操作。操作包括“Get（取）”、“Select（选择）”和“ChitChat（闲聊）”等，如图 2 中所示。槽的值可以通过 API 调用获得，而且不同的值（比如不同的餐厅名）可以对应同一个动作。补充材料（http://irip.buaa.edu.cn/Research/luoxufang/CoChat_supp.pdf）提供了完整的动作列表。在收集这两个数据集时，我们还发现对话管理策略通常会随时间改变，动作集也会随时间增大。在我们的实验中，所有对话都会根据它们的收集时间戳按顺序输入学习过程。

表2　所收集的数据集的基本统计情况

	Restaurant	Movie
# average dialog turns	3.8	3.5
# max dialog turns	8	10
# actions	49	70
# dialogs	1490	1490

因为实验需要与用户进行大量交互，所以我们按照之前的工作（Schatzmann et al. 2007）使用收集到的对话构建了用户模拟器。然后我们使用它们生成了更多对话，以用于通过强化学习训练对话管理器和测试每个学习后的对话管理器。注意，为了更好地模拟真实用户，用户模拟器会随时间重建。与（Lipton et al. 2016）相似，我们设置强化学习奖励的方式如下：每个回合一个固定惩罚（即 −0.025）、一个用于成功完成任务的大奖励（即 1）和一个用于任务失败的小奖励（即 0.5）。

评估 CoChat 框架

我们通过在不同的框架下训练所提出的 MemHRNN 而对这些不同框架进行了比较，以了解 CoChat 能否更好地最大化用户满意度和最小化工作者的工作量。

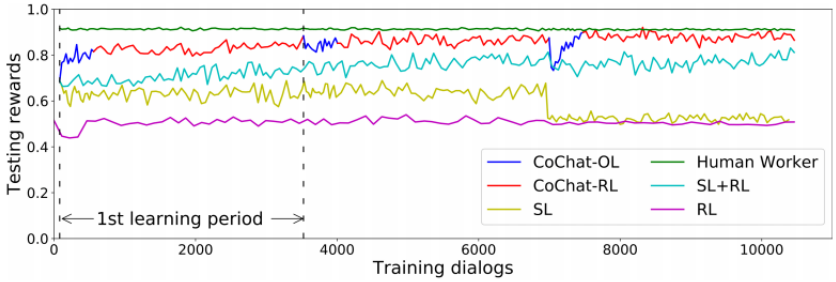
设置　为了清楚说明 CoChat 中的每个学习过程，我们执行了如下实验。首先，为了跟踪真实世界情况，对话管理器是使用少量收集到的对话（即前 50 个）通过监督学习训练得到的。然后再在后续的对话上交替执行在线学习和强化学习。一个在线学习过程及其后续的强化学习过程组成一个学习期（learning period）。在每个学习期

中，在线学习是在收集到的对话上执行的，而强化学习的执行使用的是根据在线学习中的对话所构建的用户模拟器。

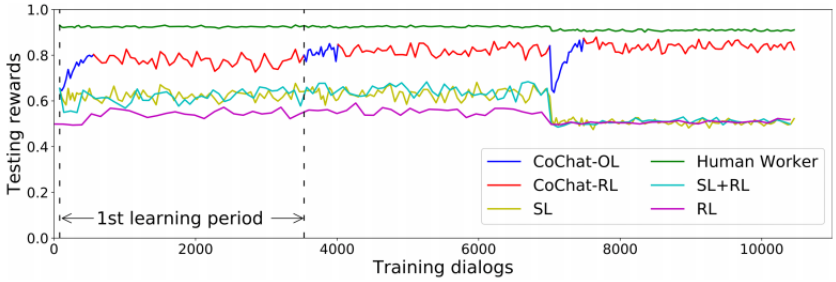
参与比较的框架包括强化学习（RL）、监督学习（SL）和两者的组合方法（SL+RL）。在 RL 中，对话管理器仅使用强化学习从头开始训练。在 SL 中，对话管理器使用前 50 个收集到的对话通过监督学习训练得到，并且不会更新。SL+RL 是在使用监督学习进行初始化后再应用强化学习。在每个学习期，RL 框架和 SL+RL 框架的对话管理器的训练都使用了与 CoChat 所用的一样的用户模拟器。

至于表现指标，用户满意度是以奖励衡量的。具体来说，我们将测试奖励定义为框架在与用户模拟器交互了 100 次对话之后获得的每次对话平均奖励。为了衡量工作者的工作量，我们重点关注了 bot 与人类工作者协作并为他们提供动作建议的情况。为了衡量动作建议的表现，我们使用了前五命中率（top 5 hit rate），即工作者接受前五个建议动作中某个动作的对话回合所占百分比。命中率更高表示工作量越少。

结果 图 3 给出了不同框架的测试奖励跟踪曲线。我们可以观察得出：1）在最大化用户满意度方面，人类工作者的表现优于所有学习到的对话管理器，因此我们提出的有人类工作者参与的 CoChat 在有人类工作者时当然能够实现用户满意度的最大化；2）当学习到的对话管理器与用户直接交互时，通过 CoChat 学习到的对话管理器的表现优于使用其它框架学习的对话管理器，这充分表明了 CoChat 的优越性；3）由于对话管理策略的改变或出现的新动作，通过 CoChat 学习到的对话管理器的表现可能会在在线学习阶段（即 CoChat 中的在线学习）下降，但它在下降之后可以快速恢复，而不会像 SL 那样一直保持下降后的表现，这说明 CoChat 能更好地适应新策略 / 动作；4）除了策略改变或出现新动作的情况，一般情况下 CoChat 中的监督学习、在线学习和强化学习会共同让对话管理器的表现获得持续的提升。



(a) Restaurant.



(b) Movie.

图 3　人类工作者和所学习到的对话管理器的测试奖励的跟踪曲线

表 3 给出了不同的对话管理器在结束阶段（即最后 400 个对话）的平均测试奖励。在经过 3 个学习期之后，通过 CoChat 学习到的对话管理器在这两个任务上所获的奖励分别达到了人类工作者所获奖励的 97.04% 和 92.62%。

表 3　人类工作者和所学习到的对话管理器在结束阶段的测试奖励

Dataset	Human Worker	*CoChat*	*SL*	*RL*	*SL+RL*
Restaurant	0.911	**0.884**	0.523	0.501	0.791
Movie	0.908	**0.841**	0.505	0.512	0.511

我们报告了 CoChat 的 3 个在线学习过程中动作建议的前五命中率，并与 SL 的结果进行了比较，结果见表 4。可以看到，当 bot 建议 5 个动作时，通过 CoChat 学习到的 bot 可以减轻工作者 80%–90% 的工作量。

表4 在 3 个在线学习过程中动作建议的前五命中率

Dataset	Frameworks	1^{st}	2^{nd}	3^{rd}
Restaurant	*CoChat*	**89.91%**	**89.87%**	**86.87%**
	SL	82.07%	80.27%	80.10%
Movie	*CoChat*	**82.49%**	**83.82%**	**78.73%**
	SL	66.81%	65.60%	51.14%

关于新动作的分析

关于新动作的统计情况 为了阐明解决新动作难题的必要性，我们进一步分析了真实世界数据集中某些关于新动作的统计数据。

图 4 给出了两个数据集中动作数量随所收集的对话数量的增多而增多的情况。可以看到，新动作随时间推移不断出现。

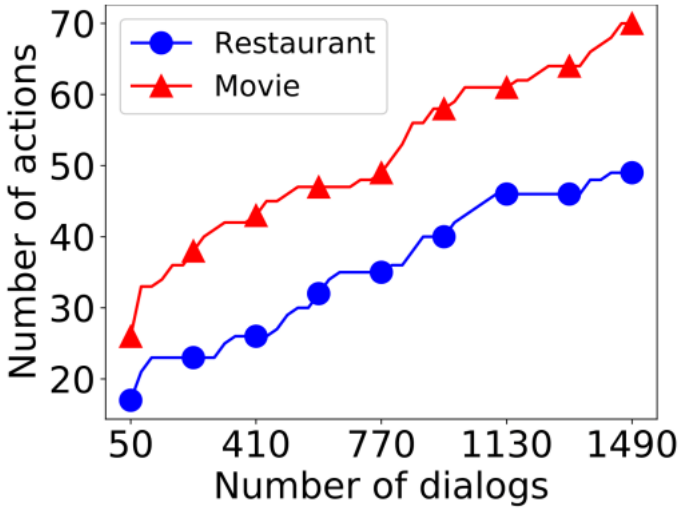


图 4 动作数量随所收集的对话数量而增多的跟踪曲线

此外，图 5 表明带有新动作的对话的比例在两个数据集中都大约为 20%，这说明新动作在出现之后在后续对话中也有频繁使用。因此，妥善处理新动作对学习得到优良的对话管理器而言是必要的。

	# dialogs with new actions	Ratio
Restaurant	305	21.18%
Movie	278	19.31%

图 5　两个数据集中带有新动作的对话的数量和相应百分比

用于处理新动作的 MemHRNN 为了验证我们提出的 MemHRNN 能否在在线学习过程中更好地处理新动作，我们在同样的学习过程中将其与其它之前最佳的模型进行了比较。

设置 在这里，所有模型首先都使用了前 50 个对话进行训练，然后再使用后续的 1440 个收集的对话来执行在线学习。这个学习过程将会涉及到新动作。对于这些用于比较的模型中不能处理新动作的模型，在训练时会略过带有新动作的对话回合。

结果 表 5 给出了所比较的模型给出的动作建议的前五命中率。可以看到，通过允许引入新动作和妥善处理对应的单样本学习难题，我们提出的 MemHRNN 可以向人类工作者提供更好的建议以减轻他们的工作量。

表 5　所比较的模型在两个数据集上的动作建议的前五命中率

Models	Restaurant	Movie
MemHRNN	**91.35%**	**86.32%**
HCN (Williams, Asadi, and Zweig 2017)	89.47%	82.15%
HLSTM (Xie et al. 2016)	87.80%	81.36%
Liu and Lane (2017)	87.97%	81.66%

利用外部记忆的优越性

设置 这里的实验设置和上面的实验一样。我们将我们提出的 MemHRNN 与没有使用记忆的 HRNN 进行了比较，以便了解外部记忆是否真的有助于解决由新动作所导致的单样本学习难题。

结果 表 6 给出了我们提出的 MemHRNN 和没有使用记忆的 HRNN 的动作建

议的前五命中率，这表明引入外部记忆确实能增强 MemHRNN 处理新动作的单样本学习难题的能力。详细的案例分析可参考补充材料。

表 6　MemHRNN 和 HRNN（无记忆）在在线学习阶段的动作建议的前五命中率

Models	Restaurant	Movie
MemHRNN	**91.35%**	**86.32%**
HRNN (NO Memory)	90.34%	84.82%

结论

在本论文中，我们提出了一种对聊天机器人与人类工作者协作共同完成任务进行建模的学习框架 CoChat。CoChat 的目标是最大化用户满意度和最小化工作者的工作量。此外，CoChat 还将监督学习、在线学习和强化学习结合到了一起，来持续地改进有待学习完成的对话管理器。我们还在这里提出了一种用于实现 CoChat 框架的记忆增强型分层 RNN 模型，即 MemHRNN。特别需要指出，在外部记忆的帮助下，MemHRNN 可以解决由即时引入新动作所导致的单样本学习难题。我们在真实世界数据集上进行了大量实验，结果很好地表明了我们提出的 CoChat 框架和 MemHRNN 模型的有效性。

阿里巴巴 AAAI 论文 CoLink：知识图谱实体链接无监督学习框架

CoLink: An Unsupervised Framework for User Identity Linkage

Zexuan Zhong[†] * , Yong Cao[‡], Mu Guo[‡] and Zaiqing Nie[§]
[‡]Microsoft Research, China
[†]University of Illinois at Urbana-Champaign, USA
[§]Alibaba AI Labs, China
zexuan2@illinois.edu, {yongc, muguo}@microsoft.com, zaiqing.nzq@alibaba-inc.com

主要作者（中英文）：钟泽轩 Zexuan Zhong　曹涌 Yong Cao　郭沐 Mu Guo
聂再清 Zaiqing Nie

论文下载地址：
https://102.alibaba.com/downloadFile.do?file=1518508273059/CoLink%20An%20Unsupervised%20Framework%20for%20User%20Identity%20Linkage.pdf

摘要

将几个子知识图谱上的同一实体信息链接在一起（也被称为用户身份链接（UIL）问题）对很多应用而言都至关重要。实体链接问题有两大主要难点。第一，收集人工链接的实体信息对（user pairs）作为训练数据的成本非常高昂。第二，不同子知识图谱的实体属性通常有非常不同的定义方式和格式，这使得属性对齐（attribute alignment）非常困难。我们在本论文中提出了 CoLink，一种用于实体信息链接问题的通用型无监督框架。CoLink 使用了一种能同时操作两个独立模型（基于属性的模型和基于关系的模型）的协同训练算法，并且能以无监督学习的方式迭代式地让两个模型彼此互相增强。我们还提出使用“序列到序列”学习作为基于属性的模型非常有效，这种方法能将属性对齐难题当作机器翻译问题处理。我们将 CoLink 应用到了将企业网络中的员工映射到他们的领英（LinkedIn）个人资料的实体信息链接任务上。实验结果表明 CoLink 在 F1 分数上的表现超过之前最佳的无监督方法的 20% 以上。

引言

将不同子知识图谱上的同一实体信息链接起来（也被称为用户身份链接（UIL）问题）通常能得到对该实体的更好和更深度的理解，这通常又能进一步得到更好的商业智能。

尽管机器学习算法已经在实体链接问题上得到了广泛的应用，但训练数据的标注工作并不简单。首先，寻找已链接实体信息配对是极其耗时的，因为这需要搜索所有子知识图谱以及仔细评估大量候选配对。另外这个工作还需要人类标注者具有广泛的领域知识。其次，由于隐私保护的原因，并非所有知识图谱的实体数据都可以提供给人类标注者，尤其是当这些资料来自个人社交网络或企业内部网络时。

在两个子知识图谱之间链接实体需要仔细比对两个子图谱中的实体属性，比如名称、职位、位置等。因此，属性值的对齐对实体链接问题而言至关重要。但是，传统的字符串相似度函数有两个不足之处：

- 没有一个通用方法可以处理相同属性在不同实体网络中的变化
- 无法找到隐式的属性对应关系

在这篇论文中，我们提出了一种用于实体链接问题的通用型无监督框架 CoLink。知识图谱中的实体数据可以自然地划分为两个独立的角度的特征：属性和关系，这完美契合协同训练（co-training）算法的要求。CoLink 使用两个独立的模型：一个基于属性的模型和一个基于关系的模型。基于属性的模型和基于关系的模型都是二元分类器，决定两个实体是否能链接起来。它们可以基于任何机器学习或启发式算法。因此，只要知识图谱资料中包含属性和关系，那就可以将 CoLink 应用于该知识图谱的实体链接问题上。

更进一步，我们在 CoLink 的基于属性的模型的实现中使用了“序列到序列”学习算法，这为不同实体网络之间的属性对齐提供了一种通用方法。我们没有将属性对齐当成字符串相似度比较而进行处理，而是试图将一种“语言”（一种特定风格的网络）的属性值“翻译”成另一种“语言”。缩略语、缩写、同义词甚至隐式对应关系都可被视为翻译的特殊情况。我们选择“序列到序列”算法的原因是其已经表现出了在机器翻译任务上的有效性。具体而言，“序列到序列”方法有两种可用于 CoLink 的

优势。首先，它几乎无需手动提取特征就能自动得到词层面的映射和序列层面的映射。其次，它只需要正例（已对齐的属性对）作为训练数据，这能减轻采样负例的工作。

我们将 CoLink 应用到链接社交网络的相同用户的任务上，其中我们试图将企业网络中的员工和他们的领英个人资料链接起来。我们进一步比较了 CoLink 和之前最佳的无监督方法。实验结果表明 CoLink 在 F1 分数上的表现总体上能超过之前最佳的无监督方法的 20%。我们的贡献总结如下：

- 我们最早将协同训练算法用在了知识图谱实体链接的问题上。由于实体网络中的实体属性和实体关系是自然分开的，这使得协同训练是一种完美且无成本的解决方案。
- 我们最早将属性对齐问题建模为机器翻译。我们使用“序列到序列”方法作为基于属性的模型的基础，这几乎无需提取特征就能实现很好的泛化。
- 我们进行了大量实验，比较了我们提出的方法和之前最佳的无监督方法，列举了不同的设置和模型，结果表明了我们提出的解决方案的有效性。

CoLink

问题定义

知识图谱上的实体链接问题定义为：其输入包括一个源知识图谱和一个目标知识图谱。其输出为一个实体链接对集合，表示从源图谱中链接到目标图谱中的实体对。

CoLink 框架

CoLink 框架基于如算法 1 所示的协同训练算法。我们在该框架中定义两个不同的模型：一个基于属性的模型 f_{att} 和一个基于关系的模型 f_{rel}。这两个模型都会进行二元分类预测，将一组给定实体对分类为正例（链接的）或负例（非链接的）。该协同训练算法以迭代的方式不断增强这两个模型。在每一次协同训练迭代过程中，两个模型都会使用已链接配对集 S 进行再训练。然后使用这两个模型生成的高质量的已链接配对会被合并到 S 中以用于下一次迭代，直到 S 收敛。在最开始时，需要一个初始的已链接配对集（简称种子集）来启动这个协同训练过程，这个集合可以通过一组种子

规则生成。根据模型所用的算法，基于属性的模型和基于关系的模型的训练可能会需要负例。算法 1 中没有给出采样负例的过程。

Input: a source social network G^s, a target social network G^t
Output: a set of user pairs S
$S \leftarrow$ the set of seed pairs generated with seed rules;
repeat
  /* generate pairs from attribute-based model */
  $D_{att} \leftarrow f_{att}(S, G^s, G^t)$;
  /* generate pairs from relationship-based model */
  $D_{rel} \leftarrow f_{rel}(S, G^s, G^t)$;
  /* join two sets and remove conflicting pairs */
  $D \leftarrow merge(D_{att}, D_{rel})$;
  $S \leftarrow S \cup D$;
until $D = \emptyset$;
return S ;

算法 1　CoLink 中的协同训练算法

这个协同训练算法不会修改之前的迭代中生成的已链接配对。因此由之前的迭代引入的误差不会在后面得到修复。这种算法的一种替代方案是在协同训练收敛之后进行一次最终修改。即使用该协同学习过程所得到的最终模型来重构 S。

种子规则

该协同训练算法的启动需要一个已链接实体对构成的小型种子集。获取种子集的一种简单直接的方法是根据人工设计的规则来生成，我们称之为种子规则。这些种子规则可以考虑来自目标知识图谱的以下事实：

- 实体名称唯一性
- 实体属性值映射
- 实体关系传播

种子规则的选取会直接影响 CoLink 的表现。

基于属性的模型

基于属性的模型通过考虑实体的属性来预测链接的实体对。它可以使用任何分类算法。在这篇论文中，我们尝试了两种不同的机器学习算法：“序列到序列”和支持向

量机（SVM）。

序列到序列 由于属性有不同的变化形式，所以传统的字符串相似度方法在处理属性对齐方面表现很差。由于属性对齐类似于机器翻译问题，所以我们采用了“序列到序列”方法。缩略语、缩写、同义词甚至隐式链接都可被视为翻译的特殊情况。

我们采用了 Sutskever, Vinyals, and Le (2014) 提出的“序列到序列”网络结构。该网络由两部分构成：序列编码器和序列解码器。编码器和解码器都使用了深度长短期记忆（LSTM）架构。编码器深度 LSTM 会读取输入序列并求出每个词位置的表示向量。然后这些向量会被送入一个注意层（attention layer），从而得到一个考虑了输出词位置的输入序列的整体表示。然后解码器深度 LSTM 的隐藏状态会进一步被送入一个全连接层（其输出包含词汇库大小的维度信息），进而预测输出词。

我们按照之前的工作，使用已链接属性值配对训练了“序列到序列”网络。但是，我们不是使用网络预测输出序列，而是在 CoLink 中使用所学习到的“序列到序列”网络来进行二元分类。首先，我们使用该网络求出对于一对属性匹配的概率。然后，我们选择一个匹配概率阈值，超过该阈值的实体对被认为是有链接的。

支持向量机 SVM 等传统的分类算法也可以用在基于属性的模型中。不同于只需要正例训练样本（已链接配对）的“序列到序列”方法，SVM 还需要负例。因为用户配对空间非常大，所以正例在整个空间中实际上非常稀疏。在每次联合训练迭代中，给定已链接配对，我们还会选择同等数量的随机实体对作为负例。

基于关系的模型

基于关系的模型仅使用实体关系来预测链接实体对。只根据关系来寻找两个网络中同等结点的问题通常被称为网络对齐问题。

基于关系的模型可以使用任何基于关系的网络对齐模型。因为本文的重点是协同训练算法和“序列到序列”的基于属性的模型，所以我们在本论文中使用了一种简单的启发式模型，该模型基于一个假设：如果两个来自不同网络的实体都具有大量互相关联的已链接实体，那么这两个实体很可能也是链接的。

实验

我们的实验比较了 CoLink 与当前最佳的无监督方法。我们还研究了种子规则和链接概率阈值的选择，以更好地理解它们对链接结果的可能影响方式。

数据集

我们选择了一个真实数据集来评估 CoLink，它包含两个社交网络。其中一个社交网络是领英，另一个网络是一个企业内部用户网络。

表 1　数据集总体情况

		Enterprise	LinkedIn
Network	# vertices	221,869	2,480,410
	# relationships	4,411,721	16,069,575
Attribute	# names	221,869	2,468,072
	# job titles	132,430	2,207,871
	# locations/offices	148,471	2,466,961

候选实体对的选择

我们构建了一个候选实体对过滤器，它能移除大量不可能链接的实体对。该候选项过滤器考虑了以下属性。

- 实体名
- 组织机构

在过滤之后，我们得到了 758 046 个候选实体对，其涵盖了测试集合中所有有链接的配对。

序列到序列

我们实验中的“序列到序列”网络由一个带注意网络的深度 LSTM 编码器和一个深度 LSTM 解码器构成。编码器深度 LSTM 和解码器深度 LSTM 都有 2 个层叠的 LSTM，因为我们发现对于实体链接任务而言，超过 2 层的编码器或解码器不能再带来更多提升。每个 LSTM 的循环单元大小为 512。每个词在被送入编码器和解码器之前都首先会被转换成一个 512 维的嵌入向量。“序列到序列”模型的训练时间

取决于训练数据的规模。平均而言，使用一个 Tesla K40 GPU，让模型在 10 万个属性配对上完成训练需要 30 分钟。

种子规则

为了测试 CoLink 的稳健性，我们尝试了下列 3 个种子规则集：

- 粗略调整的集合
- 精细调整的集合
- 有噪声集合

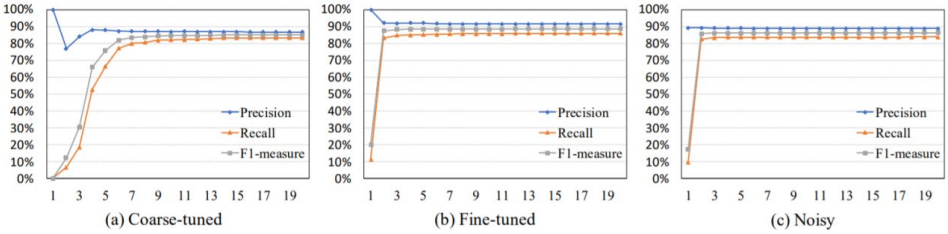


图 1：种子集比较；协同训练迭代开始后的 P/R/F1 趋势

协同训练

我们通过将关系特征和属性特征分开而使用了协同训练。基于属性的模型和基于关系的模型都能在每次迭代中找到新配对然后增强彼此。图 2 给出了每个模型所得到的已链接配对的统计情况。在这项任务中，基于属性的模型生成的配对比基于关系的模型多，这是因为我们没有完整的领英关系数据。我们爬取了公开的领英个人资料中的“人们还看了”列表，这只能为每位用户提供不到 10 个关系。

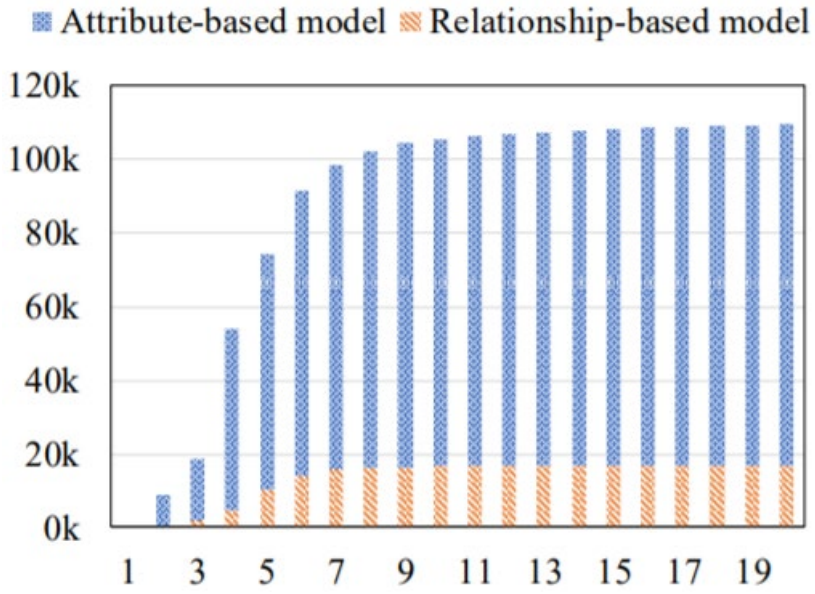


图 2　基于粗略调整的种子配对使用联合训练迭代得到的已链接配对的增长情况

概率阈值

图 3 给出了不同阈值的比较情况。使用更严格的阈值（更小的百分数）会得到更高的精度和相对更低的召回率。我们在本任务中选择的阈值是 95%。

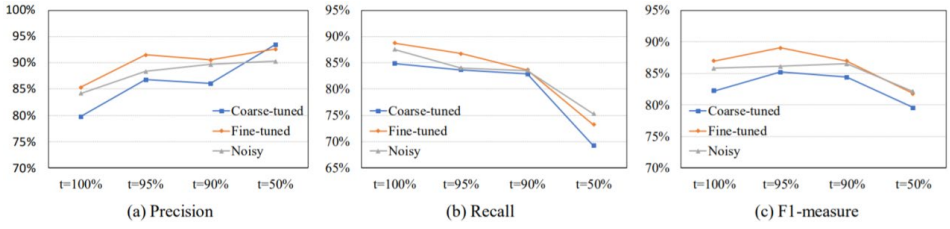


图 3　序列到序列链接概率阈值比较

比较结果

表 2　不同方法的表现的比较

Method	P	R	F1
Random-select	49.31	54.21	51.64
SiGMa	91.00	44.28	59.57
Alias-disamb	82.35	58.92	68.69
CoLink (S2S+Coarse-tuned)	86.74	83.67	85.18
CoLink (S2S+Coarse-tuned+Rev)	89.51	86.20	87.82
CoLink (S2S+Fine-tuned)	**91.47**	**86.70**	**89.02**
CoLink (S2S+Fine-tuned+Rev)	89.22	86.36	87.77
CoLink (SVM+Fine-tuned)	84.16	62.63	71.81

属性对齐

通过使用“序列到序列”方法，CoLink 可以处理使用传统字符串相似度函数难以应付的属性对齐问题。表 3 给出了一些选择出的应该是对齐的属性示例以及来自不同方法的相似度分数（全都位于 [0,1] 区间中）。在“序列到序列”的帮助下，几乎无需提取特征，就可以轻松地将这种方法应用于其它实体匹配任务。

表 3　选择出的一些属性示例以及它们的相似度分数

Enterprise	LinkedIn	Seq-to-seq likelihood	Cosine Similarity	Jaccard Similarity	Jaro Distance	LCS-based Fuzzy-match
SUNNYVALE-1020/6221	San Francisco Bay Area	0.931	0	0	0.181	0.028
PARIS-ISSY/O3E17F	Parijs en omgeving, Frankrijk	0.703	0	0	0.575	0.250
PFE	Premier Field Engineer	0.817	0	0	0.529	0.067
SR SDE	Senior Software Development Engineer	0.733	0	0	0.272	0.087
ESCAL ENG	Escalation Engineer	0.843	0	0	0.650	0.500
SR CONSULTANT PROD	Senior Consultant	0.722	0.387	0.323	0.618	0.444

层叠描述：用于图像描述的粗略到精细学习

Stack-Captioning: Coarse-to-Fine Learning for Image Captioning

Jiuxiang Gu[1], Jianfei Cai[2], Gang Wang[3], Tsuhan Chen[2]
[1] ROSE Lab, Interdisciplinary Graduate School, Nanyang Technological University, Singapore
[2] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[3] Alibaba AI Labs, Hangzhou, China
{jgu004, asjfcai, tsuhan}@ntu.edu.sg, gangwang6@gmail.com

论文下载地址：

https://102.alibaba.com/downloadFile.do?file=1518074198430/AAAI2018Stack-Captioning_Coarse-to-Fine%20Learning%20for%20Image%20Captioning_12213(1).pdf

摘要

现有的图像描述方法通常都是训练一个单级句子解码器，这难以生成丰富的细粒度的描述。另一方面，由于梯度消失问题，多级图像描述模型又难以训练。我们在本论文中提出了一种粗略到精细的多级图像描述预测框架，该框架由多个解码器构成，其中每一个都基于前一级的输出而工作，从而能得到越来越精细的图像描述。通过提供一个实施中间监督的学习目标函数，我们提出的学习方法能在训练过程中解决梯度消失的难题。尤其需要指出，我们使用了一种强化学习方法对我们的模型进行优化，该方法能够利用每个中间解码器的测试时间推理算法的输出及其前一个解码器的输出来对奖励进行归一化，这能一并解决众所周知的曝光偏差问题（exposure bias problem）和损失 - 评估不匹配问题（loss-evaluation mismatch problem）。我们在 MSCOCO 上进行了大量实验来评估我们提出的方法，结果表明我们的方法可以实现当前最佳的表现。

引言

图像描述的困难之处是让设计的模型能有效地利用图像信息和生成更接近人类的丰富的图像描述。在自然语言处理近期进展的推动下，当前的图像描述方法一般遵循编码 – 解码框架。这种框架由一个基于卷积神经网络（CNN）的图像编码器和基于循环神经网络（RNN）的句子解码器构成，有多种用于图像描述的变体。这些已有的图像描述方法的训练方式大都是根据之前的基本真值词（ground-truth words）和图像，使用反向传播，最大化每个基本真值词的可能性。

这些已有的图像描述方法存在三个主要问题。第一，它们很难生成丰富的细粒度的描述。第二，在训练和测试之间存在曝光偏差。第三，存在损失与评估的不匹配问题。

考虑到使用单级模型生成丰富的图像描述的巨大挑战性，我们在本论文中提出了一种粗略到精细的多级预测框架。我们的模型由一个图像编码器和一系列句子解码器构成，可以重复地生成细节越来越精细的图像描述。但是，直接在图像描述模型中构建这样的多级解码器面临着梯度消失问题的风险。Zhang, Lee, and Lee 2016; Fu, Zheng, and Mei 2017 等在图像识别上的研究工作表明监督非常深度的网络的中间层有助于学习，受这些研究的启发，我们也为每级解码器实施了中间监督。此外，Rennie et al. 2017 这项近期的图像描述研究使用了强化学习（RL）来解决损失 – 评估不匹配问题，并且还在训练中包含推理过程作为基准来解决曝光偏差问题；我们也设计了一种类似的基于强化学习的训练方法，但是将其从单级扩展成了我们的多级框架，其中每级都引入了作为中间监督的奖励。尤其需要指出，我们使用了一种强化学习方法对我们的模型进行优化，该方法能够利用每个中间解码器的测试时间推理算法的输出及其前一个解码器的输出来对奖励进行归一化。此外，为了应对我们的粗略到精细学习框架，我们采用了一种层叠式注意模型来为每个阶段的词预测提取更细粒度的视觉注意信息。图 1 给出了我们提出的粗略到精细框架的示意图，它由三个层叠的长短期记忆（LSTM）网络构成。第一个 LSTM 生成粗尺度的图像描述，后面的 LSTM 网络用作精细尺度的解码器。我们模型中每一级的输入都是前一级所得到的注意权重和隐藏向量，这些被用作后一级的消岐线索。由此，每一级解码器就会生成注

意权重和词越来越精细的句子。

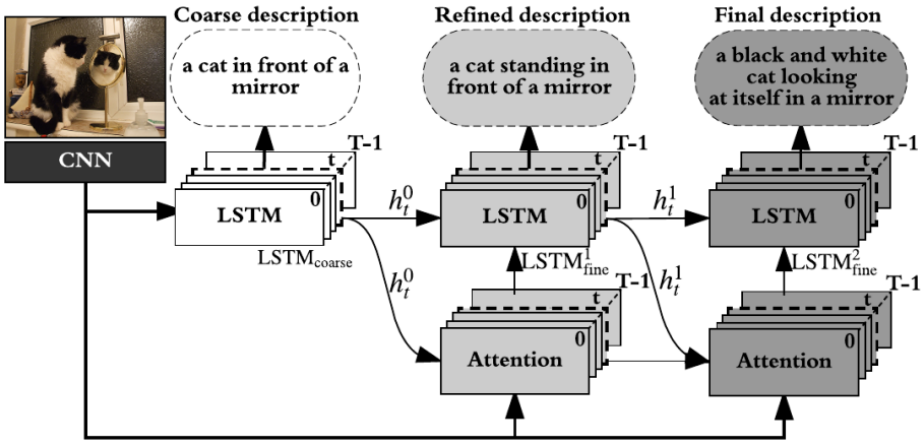


图 1　我们提出的粗略到精细框架示意图。我们的模型由一个图像编码器（CNN）和一系列句子解码器（基于注意的 LSTM 网络）构成。该模型以图像为输入，能够从粗略到精细不断细化图像描述。这里我们展示了两级式的图像描述渐进提升（灰色和深灰色）。

本工作的主要贡献包括：（a）一种用于图像描述的粗略到精细框架，可以使用越来越细化的注意权重逐渐增大模型复杂度；（b）一种使用归一化后的中间奖励直接优化模型的强化学习方法。实验表明我们的方法在 MSCOCO 上表现出色。

方法

在本论文中，我们考虑了学习生成图像描述的问题。我们的算法构建了一个粗略到精细模型，它具有与单级模型一样的目标，但在输出层和输入层之间具有额外的中间层。我们首先根据输入图像和目标词的黄金历史，通过最大化每个连续目标词的对数似然而对该模型进行训练，然后再使用句子层面的评估指标对模型进行优化。结果，每个中间句子解码器会预测得到越来越细化的图像描述，最后一个解码器的预测结果用作最终的图像描述。

图像编码

我们首先将给定图像编码成空间图像特征。具体来说，我们从 CNN 的最后卷积层提取图像特征，然后使用空间自适应平均池化将这些特征的尺寸调整成固定尺寸的

空间表示。

粗略到精细解码

整体的粗略到精细句子解码器由一个粗略解码器和一系列基于注意的精细解码器构成，这些解码器可以根据来自前一个解码器的线索来得到每个词预测的细化后的注意图（attention map）。我们模型的第一级是一个粗略解码器，能根据全局图像特征预测得到粗略的描述。在后续阶段，每一级都是一个精细解码器，可以基于图像特征和前一级的输出而预测得到更好的图像描述。尤其需要指出，我们使用了前一级的注意权重来提供后一级词预测的区域信念。也就是说，我们以多级方式解码图像特征，其中每级的预测结果都是对前一级预测结果的精细化。

图 2 给出了我们提出的粗略到精细解码架构，其中每一级之后都使用了中间监督（奖励）。上面一行（灰色）包含一个粗略解码器（左）和两个层叠的基于注意的精细解码器（处于训练模式下）；下面一行给出了处于推理模式（贪婪解码）下的精细解码器，用于计算奖励以将中间监督整合进来。

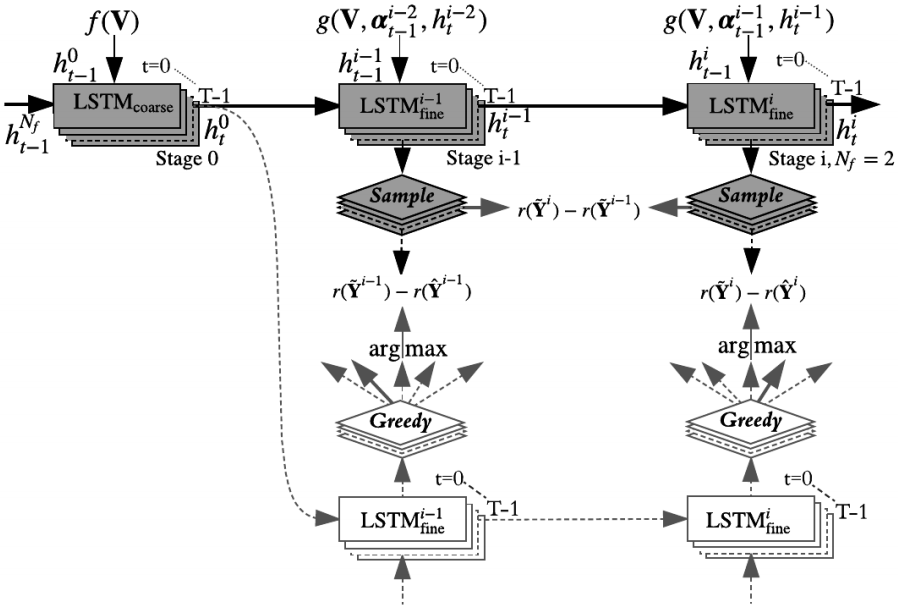


图 2

粗略解码器。我们首先在第一级的粗略搜索空间中解码，我们在这里使用一个 LSTM 网络学习一个粗略解码器，称为 LSTMcoarse。LSTMcoarse 在每个时间步骤的输入都由前一个目标词（连接着全局图像特征）和之前的隐藏状态构成。

精细解码器。在后续的多级中，每个精细解码器都会再次基于图像特征以及来自前一个 LSTM 的注意权重和隐藏状态来预测词。每个精细解码器都由一个 LSTMfine 网络和一个注意模型构成。LSTMfine 在每个时间步骤的输入都包含已出现的图像特征、前一个词嵌入及其隐藏状态、来自前一个 LSTM 的更新后的隐藏状态。

层叠式注意模型。如前所述，我们的粗略解码器基于全局图像特征生成词。但是在很多情况下，每个词都只与图像中的很小一部分有关。由于每次预测时图像中的无关区域会引入噪声，所以为词预测使用全局图像特征会得到次优的结果。因此，我们引入了注意机制，这能显著提升图像描述的表现。注意机制通常会得到一个空间图（spatial map），其中突出显示了与每个预测词相关的图像区域。为了为词预测提取更细粒度的视觉信息，我们在本研究中采用了一种层叠式注意模型来逐渐滤除噪声和定位与词预测高度相关的区域。在每个精细处理级中，我们的注意模型都会对图像特征和来自前一级的注意权重进行操作。

学习

上面描述的粗略到精细方法能得到一种深度架构。训练这样一种深度网络可能很容易出现梯度消失问题，即梯度的幅度会在反向传播通过多个中间层时强度减小。解决这种问题的一种自然方法是将监督训练目标整合到中间层中。每一级粗略到精细句子解码器的训练目标都是重复地预测词。我们首先通过为每一级定义一个最小化交叉熵损失的损失函数来训练网络。

但是，只使用这里的损失函数进行训练是不够的。

为了优化每一级的评估指标，我们将图像描述生成过程看作是一个强化学习问题，即给定一个环境（之前的状态），我们想要智能体（比如 RNN、LSTM 或 GRU）查看环境（图像特征、隐藏状态和之前的词）并做出动作（预测下一个词）。在生成了一个完整句子之后，该智能体将观察句子层面的奖励并更新自己的内部状态。

实验

数据集和设置

我们在 MSCOCO 数据集上评估了我们提出的方法。

用于比较的基准方法

为了了解我们提出的方法的有效性，我们对以下模型进行了相互比较：

LSTM 和 $LSTM_{3\ layers}$。我们根据 Vinyals et al. 2015 提出的框架实现了一个基于单层 LSTM 的图像描述模型。我们还在该单层 LSTM 模型之后增加了另外两个 LSTM 网络，我们将其称为 $LSTM_{3\ layers}$。

$LSTM+ATT_{Soft}$ 和 $LSTM+ATT_{Top-Down}$。我们实现了两种基于视觉注意的图像描述模型：Xu et al. 2015 提出的软注意模型（$LSTM+ATT_{Soft}$）和 Anderson et al. 2017 提出的自上而下注意模型（$LSTM+ATT_{Top-Down}$）

Stack-Cap 和 Stack-Cap*。Stack-Cap 是我们提出的方法，Stack-Cap* 是一种简化版本。Stack-Cap 和 Stack-Cap* 的架构相似，只是 Stack-Cap 应用了我们提出的层叠式注意模型而不是独立的注意模型。

定量分析

在实验中，我们首先使用标准的交叉熵损失对模型进行了优化。我们报告了我们的模型和基准模型在 Karpathy test split 上的表现，如表 1 所示。注意这里报告的所有结果都没有使用 ResNet-101 的微调。

表 1　在 MSCOCO 上的表现比较，其中 B@n 是指 BLEU-n，M 指 METEOR，C 指 CIDEr。这里所有的值都是百分数（加粗数字是最佳结果）

Approach	B@1	B@2	B@3	B@4	M	C
LSTM (XE)	72.1	54.8	39.6	28.5	24.3	91.4
$LSTM_{3\ layers}$ (XE)	70.5	53.1	38.9	28.3	23.2	85.7
$LSTM+Att_{Soft}$ (XE)	73.8	57.2	43.1	33.0	25.7	101.0
$LSTM+Att_{Top-Down}$ (XE)	74.9	58.6	44.5	33.3	25.8	103.4
Stack-Cap* (XE)	75.6	59.6	45.6	34.6	26.3	108.0
Stack-Cap (XE)	**76.2**	**60.4**	**46.4**	**35.2**	**26.5**	**109.1**

在使用交叉熵损失优化了模型之后，我们又使用基于强化学习的算法针对 CIDEr 指标对它们进行了优化。表 2 给出了使用 SCST（Rennie et al. 2017）为 CIDEr 指标优化的 4 种模型的表现以及使用我们提出的粗略到精细（C2F）学习方法优化的 2 种模型的表现。可以看到我们的 Stack-Cap 模型在所有指标上都有显著的优势。

表 2

Approach	B@1	B@2	B@3	B@4	M	C
LSTM (CIDEr)	76.7	58.3	42.8	30.8	25.5	100.2
$LSTM_{3\ layers}$ (CIDEr)	73.0	56.1	41.1	29.9	25.1	95.9
$LSTM+Att_{Soft}$ (CIDEr)	77.3	59.3	44.1	32.1	25.9	104.8
$LSTM+Att_{Top\text{-}Down}$ (CIDEr)	76.7	60.4	45.6	33.9	26.5	112.7
Stack-Cap* (C2F)	77.9	61.6	46.7	35.0	26.9	115.9
Stack-Cap (C2F)	**78.6**	**62.5**	**47.9**	**36.1**	**27.4**	**120.4**

表 3 比较了我们的 Stack-Cap（C2F）模型与其它已有方法在 MSCOCO Karpathy test split 上的结果。Stack-Cap 在所有指标上都表现最佳。

表 3

Approach	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
Google NIC (Vinyals et al. 2015)	—	—	—	27.7	—	23.7	85.5	—
Hard-Attention (Xu et al. 2015)	70.7	49.2	34.4	24.3	23.9	—	—	—
Soft-Attention (Xu et al. 2015)	71.8	50.4	35.7	25.0	23.0	—	—	—
VAE (Pu et al. 2016)	72.0	52.0	37.0	28.0	24.0	—	90.0	—
Google NICv2 (Vinyals et al. 2016)	—	—	—	32.1	25.7	—	99.8	—
Attributes-CNN+RNN (Wu et al. 2016)	74.0	56.0	42.0	31.0	26.0	—	94.0	—
$CNN_{\mathcal{L}}$+RHN (Gu et al. 2017a)	72.3	55.3	41.3	30.6	25.2	—	98.9	18.3
PG-SPIDEr-TAG (Liu et al. 2017c)	75.4	59.1	44.5	33.2	25.7	55.0	101.3	—
Adaptive (Lu et al. 2017)	74.2	58.0	43.9	33.2	26.6	—	108.5	—
SCST:Att2in (Rennie et al. 2017)	—	—	—	33.3	26.3	55.3	111.4	—
SCST:Att2in (Ens. 4) (Rennie et al. 2017)	—	—	—	34.8	26.9	56.3	115.2	—
Stack-Cap (C2F)	**78.6**	**62.5**	**47.9**	**36.1**	**27.4**	**56.9**	**120.4**	**20.9**

在线评估。表 4 报告了我们提出的使用粗略到精细学习训练的 Stack-Cap 模型在官方 MSCOCO 评估服务器上的表现。可以看到，与当前最佳的方法相比，我们的方法非常有竞争力。注意，SCST:Att2in (Ens. 4) 的结果是使用 4 个模型联合实现的，而我们的结果是使用单个模型生成的。

表 4

Approach	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Google NIC	71.3	89.5	54.2	80.2	40.7	69.4	30.9	58.7	25.4	34.6	53.0	68.2	94.3	94.6
Hard-Attention	70.5	88.1	52.8	77.9	38.3	65.8	27.7	53.7	24.1	32.2	51.6	65.4	86.5	89.3
PG-SPIDEr-TAG	75.1	91.6	59.1	84.2	44.5	73.8	33.1	62.4	25.5	33.9	55.1	69.4	104.2	107.1
Adaptive	74.8	92.0	58.4	84.5	44.4	74.4	33.6	63.7	26.4	35.9	55.0	70.5	104.2	105.9
SCST:Att2in (Ens. 4)	78.1	93.1	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.7
Ours: Stack-Cap (C2F)	77.8	**93.2**	61.6	**86.1**	46.8	**76.0**	34.9	**64.6**	**27.0**	**35.6**	56.2	70.6	**114.8**	**118.3**

定性分析

为了表明我们提出的粗略到精细方法可以逐级生成越来越好的图像描述，并且这些图像描述与自适应关注的区域有良好的关联，我们对生成的描述中词的空间注意权重进行了可视化。我们以 16 的采样系数对注意权重进行了上采样，并使用了一个高斯过滤器使之与输入图像一样大小，并将所有上采样后的空间注意图叠加到了原始输入图像上。

图 3 给出了一些生成的描述。通过多个注意层逐步进行推理，Stack-Cap 模型可以逐渐滤除噪声和定位与当前词预测高度相关的区域。可以发现，我们的 Stack-Cap 模型可以学习到与人类直觉高度对应的对齐方式。以第一张图像为例，对比粗略级生成的描述，由第一个精细解码器生成的首次细化后的描述中包含“dog”，第二个精细解码器不仅得到了“dog”，还识别出了“umbrella”。

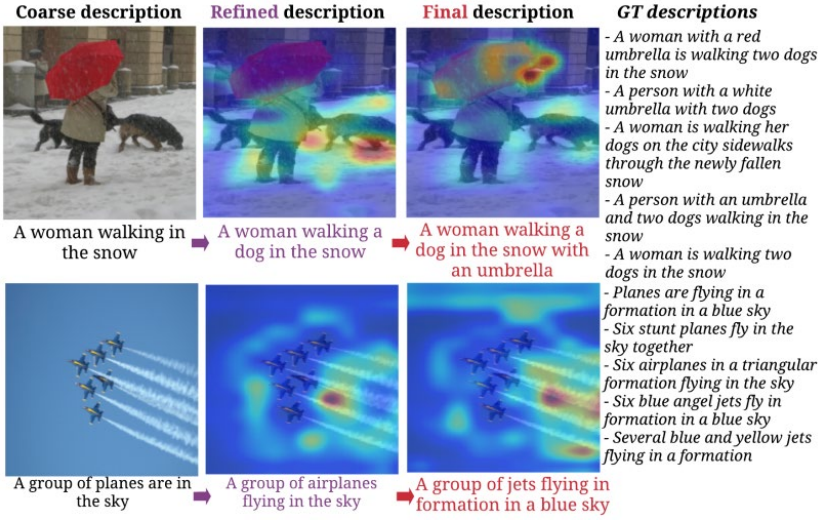


图 3

此外，我们的方法还能生成更为描述性的句子。比如，喷气机图像的注意可视化表明 Stack-Cap 模型可以查询这些喷气机的关系以及它们身后长长的烟雾尾迹，因为这些突出区域有很高的关注权重。这个例子以及其它案例说明层叠式注意可以为序列预测更有效地探索视觉信息。也就是说，我们使用层叠式注意的方法可以从粗略到精细地考虑图像中的视觉信息，这与通常通过粗略到精细流程理解图像的人类视觉系统很相似。

阿里技术

扫一扫二维码图案，关注我吧

「阿里技术」微信公众号

「阿里巴巴机器智能」公众号