Calibration of Pre-trained Transformers

葛昊

摘要

预训练的 Transformer 现在在自然语言处理中无处不在，但尽管它们在最终任务中表现非常出色，但人们对它们是否经过校准知之甚少。具体来说，这些模型的后验概率是否等于在给定示例上正确的可能性我们并不知道。在本文中关注 BERT 和 RoBERTa[1-2]，并分析它们在三个任务中的校准：自然语言推理、段落检测和常识推理。对于每项任务，我们都考虑 In-domain 和 Out-of-domain 两种设置。同时我们采用了三种方法 Temperature Scaling、Label Smoothing 以及 Mixup 进行相关实验，探究它们对模型准确度以及校准程度的影响。实验结果表明：（1）Temperature Scaling 是一种简单有效的方法，无论是 In-domain 还是 Out-of-domain 都可以有效的对模型进行校准。（2）Label Smoothing 方法对 Out-of-domain 的情况下较有效果，In-domain 时甚至还会降低模型的校准程度。（3）采用简单的随机 Mixup 样本合成时，对模型的泛化性有一定程度的增强，但是模型的校准并没有起到很好的作用。

关键词：预训练语言模型；校准；置信度；Transformer;

1 引言

随着预训练语言模型研究的不断深入，采用预训练语言模型 + 下游任务微调的方式已经在绝大部分的自然语言处理（NLP）任务上实现了 SOTA 表现。虽然模型的表现很好，但是，深度神经网络模型却由于缺乏可解释性，导致并不清楚其原理以及我们什么时候应该相信模型的预测，探究清楚这个问题很重要。例如在自动驾驶领域、医疗诊断、金融等领域，若我们一味的选择相信模型预测，那么由于模型预测错误引发的后果将不可估计。同样的，这对模型之后的发展方向也很重要，解决这个问题的方法之一就是我们让模型输出的置信度符合它预测的真实准确率，这样的话，用户在得到预测的同时也能得到预测的置信度，用户能够结合置信度来进一步考虑是否相信该预测。要达到这样的目的，必须让模型变得更可信赖，让模型输出的置信度更可信，这样我们就能依据模型的置信度进行自主决策判断是否相信模型的预测[3]。若模型输出的置信度不等于其真实准确率，则称该模型是 $miscalibration$ 的，如图 1。

让一个模型的输出置信度接近于它预测的真实准确率，我们称这个过程为模型校准，一个被完美校准的模型，它输出预测的置信度能真实的匹配该预测的真实准确率。在 NLP 领域，得益于预训练模型 Transformer[4]的优异表现，它之后衍生出了一系列的预训练语言模型，比如 BERT、RoBERTa[1-2]，它们被广大研究者使用，在 NLP 的各个领域都取得了优异的效果，然而我们却并不知道它们的校准程度，即它们输出预测的置信度，是否能反映该预测的真实准确率。

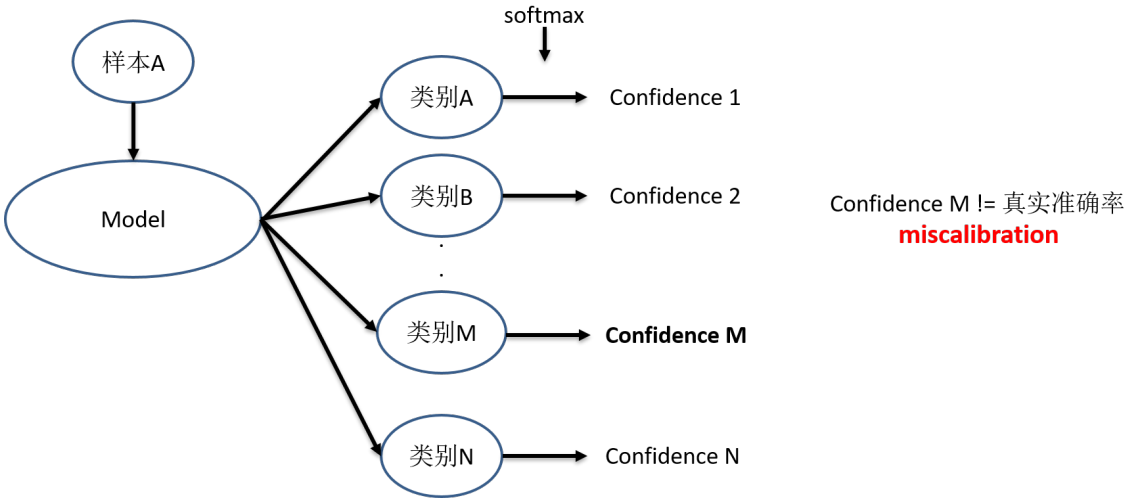

图 1: 模型 $miscalibration$ 示意图

2 相关工作

在此之前，很多研究者通过统计机器学习对决策树[5-7]、计算机视觉[8-10]进行了校准研究。此外在 NLP 领域上，Nguyen 等人[11]和 Kumar 等人[12]在若干 NLP 的任务上进行了相关校准研究。然而，之前的工作并没有对大型预训练语言模型进行分析研究，并且也没考虑到 Out-of-Domain（OOD）的情况[9,11]。

2.1 后验校准

若一个模型的预测置信度评估等于该预测的真实正确概率，那么称该模型是一个校准好了的模型。例如，若有 100 个样本，若模型的后验概率为 0.7，那么这个模型应该预测对了 70 个样本。更具体来说，给予输入 x，真实标签 y 以及模型的预测标签 $\hat{y}$，当模型被完美校准时，模型输出的置信度 $\text{Conf}(x, \hat{y})$ 将满足如下等式：$\forall p \in [0, 1], P(\hat{y} = y|\text{Conf}(x, \hat{y}) = p) = p$。

2.2 Expected Calibration Error

Guo 等人[9]他们基于数学统计的方法来评估模型的校准程度即 **Expected Calibration Error (ECE)**。在计算某个模型的 ECE 时，若该模型输出了 N 个预测置信度，则将这 N 个置信度放入到 M 个置信度桶区间中，在这里相同置信范围内的预测将被放入到同一个置信度桶区间中。

让 B_m 表示为三元组 $(x, y, \hat{y})$ 的第 m 个置信度桶区间，准确率 $\text{Acc}(B_m)$ 表示落在该置信度区间的预测样本的准确率。

$$\text{Acc}(B_m) = \frac{1}{|B_m|} \sum_{i=1}^{|B_m|} \mathbb{I}(y = \hat{y}),$$

在上面公式中，$|\text{B}_\text{m}|$ 表示落在第 m 个置信度区间的样本数量，$\text{Conf}(B_m)$ 表示该置信度桶区间的平均置信度。

$$\text{Conf}(B_m) = \frac{1}{|B_m|} \sum_{i=1}^{|B_m|} \text{Conf}(x, \hat{y}).$$

通过统计方法，算出每个置信度桶区间的 $\text{Acc}(B_m)$ 和 $\text{Conf}(B_m)$，模型的 ECE 也能依此求得。此外通过取平均的方法得出每个置信度其所对应的期望准确率，当置信度桶区间无限区分时，此时便能

无限接近我们所需要的完美校准。

$$\mathrm{ECE}=\sum_{m=1}^{M}\frac{|B_m|}{N}|\mathrm{Acc}(B_m)-\mathrm{Conf}(B_m)|.$$

现有的大量工作都是使用的等值划分置信度桶区间，即每个置信度桶区间的置信度都是一样大小的：三元组 $(x,y,\hat{y})$，$\frac{m}{M}\leqslant \mathrm{Conf}(x,\hat{y})\leqslant\frac{m+1}{M}$ 被分配在第 m 个置信度桶区间。此外也有工作是采用等数量样本来划分区间的，即每个置信度区间的样本数量都是一致的：先将模型输出的所有预测按照置信度进行排序，每个区间分配 $\frac{N}{M}$ 个预测。

3 本文方法

3.1 Temperature Scaling

在没有校准之前，模型输出的置信度相较于准确率来说通常都是过于大的（也有少部分是过于小的），因此，这些置信度需要统一的进行缩放或者扩大。Temperature Scaling[9]被证明是一种非常简单有效的模型校准方法，Temperature Scaling 让模型输出的 logit 值 (Z) 除以一个温度系数 T，用于缩放/扩大原 logit 值。如图 2。这里所使用的温度系数 T 是通过在一个保留的 dev 数据集上进行优化获得的。给予答案候选集合 $\mathcal{C}$ 以及模型输的 logit 值 $\mathbf{Z}\in\mathbb{R}^{|\mathcal{C}|}$、模型预测 $\hat{y}$，模型对 $\hat{y}$ 的置信度是在 $\mathcal{C}$ 中的第 j 个 lable：

$$\mathrm{Softmax}\left(\frac{\mathbf{Z}}{T}\right)_j.$$

对于一个分类任务，温度系数 T 通过负对数（NLL）在 dev 数据集上优化得到，Temperature Scaling 方法只会改变模型输出的置信度大小，并不会改变模型的预测准确度。

这里还能加个求温度系数 T 的算法描述。。。。

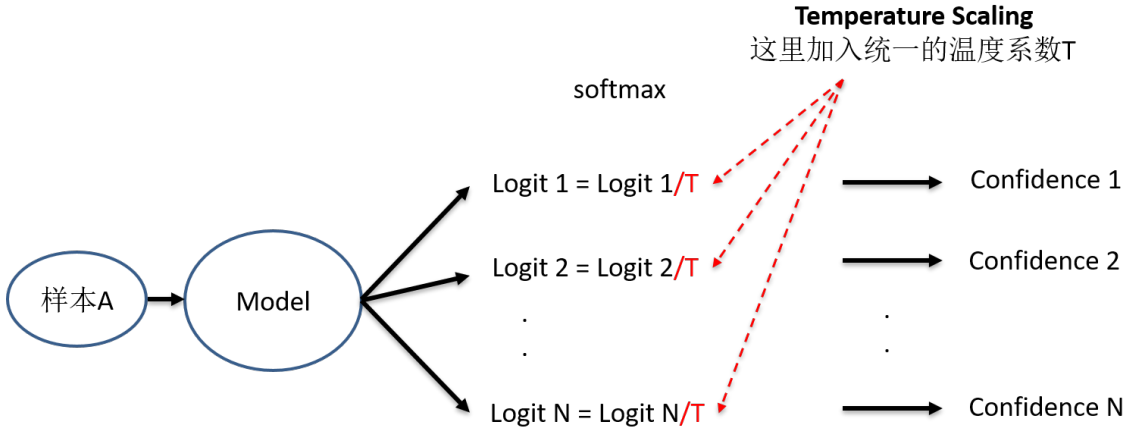

图 2: Temperature Scaling 方法

3.2 Label Smoothing

在分类任务中，深度神经网络会输出一个当前数据对应于各个类别的 logit 值，将这些值通过 softmax 进行归一化处理，最终会得到当前数据属于每个类别的概率。由于训练神经网络时，是最小化预测概率和标签真实概率之间的交叉熵，从而得到最优的预测概率分布，这会促使自身往正确标签和错误标签差值的最大方向学习，这在一定程度上会导致网络的过拟合，从而导致在 in-domain 的情况下，效果较好，但是 OOD 时，出现过于自信的情况。而 Label Smoothing 可以解决上述问题，这是一种正则化策略，主要通过软化 one-hot 来加入噪声，减少了真实样本标签的类别在计算损失函数时

的权重，最终起到抑制过拟合的效果。从而在一定程度上减低了模型输出的置信度，缓解模型过于自信的问题，让模型输出的置信度更加可信赖。传统的 one-hot 编码如下：

$$\begin{cases} 1, \text{ if } (i = y) \\ 0, \text{ if } (i \neq y) \end{cases}$$

而 Label Smoothing 引入了一个 α 系数，用于软化 one-hot 编码：

$$\begin{cases} (1 - \alpha), \text{ if } (i = y) \\ \dfrac{\alpha}{|Y| - 1}, \text{ if } (i \neq y) \end{cases}$$

如下例子，假设有 3 个类别，分别为 dog，cat，bird，我们对其进行编码如下：dog = 0; cat = 1; bird = 2;

若采用 one-hot 进行编码，则其结果如下：

dog = [1, 0, 0]

cat = [0, 1, 0]

bird = [0, 0, 1]

若采用 label smoothing 对其进行编码，这里将α设置成 0.1，则生成的标签如下：

dog = [0.9, 0.05, 0.05]

cat = [0.05, 0.9, 0.05]

bird = [0.05, 0.05, 0.9]

3.3 Mixup

Mixup 是一种简单且有效的数据增强方法，自 2018 年[13]提出之后，无论在业界还是在学术界都有了很强的地位，成为大家的一种标配。若 $\mathcal{D}_{train} = \{(x_i, y_i)\}_{i=1,\cdots,n}$ 是训练集，那么通过如下规则[13]能生成 Mixup 之后的数据增强样本。

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j$$

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j$$

在这里 x_i 和 x_j 是两个随机的输入样本，y_i 和 y_j 是他们对应的 one-hot 编码真实标签，λ 是 mixup 时的一个参数，它服从 Beta(α, α) 分布，其中 α 为超参数。在这里可以看出 Mixup 是一种线性空间中的变换。在 2019 年[14]经过大量实验发现，在 CV 领域采用 Mixup 对模型进行训练能对模型进行校准，在这里我将在原论文的基础上新增 Mixup 方法用于 NLP 的预训练语言模型训练，查看其效果。

4 复现细节

4.1 与已有开源代码对比

这篇文章的代码已经开源，本文工作主要在此基础上进行改进，新增了 Mixup 方法。关于 Mixup 方法的介绍可以看 3.3 部分。

4.2 实验环境搭建

代码运行环境：Linux-18.04-Ubuntu；

硬件信息：GPU-A100(80G 显存)；

Python 依赖包：numpy 1.20.2；Python 3.7.15；torch 1.13.0；tokenizers 0.13.2；transformers 4.25.1；

4.3 创新点

除了 3.1 和 3.2 中所使用方法，我还采用 Mixup 方法，用于模型的训练，并查看其校准效果以及对模型的准确度影响情况。

5 实验结果分析

本部分主要分为两个方面，包括模型的准确度以及模型的校准程度两个方面的实验结果与分析。实验主要在 3 个方向任务的数据集上进行测试，每个方向包括 2 个数据集，进行 In-domain 和 Out-of-domain 两种设置的测试。

5.1 自然语言推理任务

在自然语言推理方面包括 SNLI 和 MNLI 两个数据集。Stanford Natural Language Inference（SNLI）数据集是一个自然语言推理任务，在给定两个文本序列的情况下推测其关系，一共包涵 3 种关系：蕴含、矛盾、中立[15]。Multi-Genre Nautral Language Inference（MNLI）它包涵的领域比 SNLI 要更多[16]。

5.2 段落检测任务

Quora Question Pairs（QQP）是一个段落检测任务数据集，给定两个问题文本序列，判断两个文本序列是否语义同等[17]。TwitterPPDB（TPPDB）里面的文本序列对都来自 Twitter，它的任务是判断两个文本序列在共享时是否能达到相同的语义[18]。

5.3 常识推理任务

Situations With Adversarial Generations（SWAG）是一个常识推断任务，从给定的 4 个候选项中选择最符合问题的候选项[19]。HellaSWAG 则是使用对抗过滤去生成 Out-of-domain 的数据。

5.4 模型准确度

这里对模型准确度（%）的影响进行了探究，使用了两种方法的组合 Label Smoothing（LS）以及 Mixup 在六个数据集上进行实验，结果如下：

表 1: 原论文以及复现结果之间在准确率上的对比，并附加上了在各种方法上的效果对比

模型/方法	In-domain			Out-of-domain		
	SNLI	QQP	SWAG	MNLI	TwitterPPDB	HellaSWAG
BERT	89.82	**90.36**	**79.55**	72.32	87.78	34.36
BERT-原	**90.04**	90.27	79.4	**73.52**	87.63	34.48
+LS	87.23	87.62	74.85	72.23	**87.92**	**36.64**
+Mixup	88.82	89.12	74.98	69.54	87.21	34.12
+Mixup+LS	88.67	89.35	75.82	69.41	87.39	35.42
RoBERTa	91.12	**91.24**	**82.63**	77.83	86.54	40.57
RoBERTa-原	**91.23**	91.11	82.45	**78.79**	86.72	**41.68**
+LS	89.65	87.37	79.67	77.45	**87.89**	40.21
+Mixup	90.78	89.43	79.88	75.65	84.92	40.77
+Mixup+LS	90.21	87.45	79.23	76.53	87.62	39.78

通过表格 1 可以看到，我们复现的结果和原论文相比相差不大，并且，我们还新增加了采用 Lable Smoothing 方法、Mixup 方法以及 Mixup+Label Smoothing 方法在准确度上的实验结果。通过观察表格

可以知道：1）在单独使用 Label Smoothing 方法时，对 In-domain 的设置下，其准确率还会有所下降，但是在 Out-of-domain 时，它能在保持原有准确率的基础上在某些数据集上有所提升，说明这在一定程度上提升了模型的泛化性。2）单独采用 Mixup 方法时，使用随机方法进行样本合成，则可以发现，不管是 In-domain 还是 Out-of-domain，对模型的准确率都会有所下降，这里我们推测是因为采用的随机选样本的方法所以造成这样的结果，若采用一定的规则，则说不定会有所上升，这将作为未来工作进行。3）若采用 Mixup+Label Smoothing 方法则可以看到其结果在 Mixup 方法原有的基础上又有了一定的提升，并且在 In-domain 上要优于单独使用 Label Smoothing 方法，在 Out-of-domain 上和其差不多，这说明 Mixup 方法在一定程度上和 Label Smoothing 方法形成了相互增强的趋势，增强了模型的泛化性。

5.5 模型校准程度

这里进行了 In-domain 和 Out-of-domain 两种设置，采用 BERT 和 RoBERTa 在六个数据集上进行实验，这里使用了三种方法的组合 Temperature Scaling（TS）、Label Smoothing（LS）以及 Mixup，采用 ECE 作为模型校准程度的衡量指标，实验结果如表 2、表 3。

表 2: In-domain 设置下，原论文以及复现结果在各种方法上 ECE 指标的结果，并附上了 Mixup 方法的结果

In-domain						
模型/方法	SNLI		QQP		SWAG	
	无 TS	加 TS	无 TS	加 TS	无 TS	加 TS
BERT	**2.47**	**1.02**	2.78	**0.89**	3.12	**0.72**
BERT-原	2.54	1.14	**2.71**	0.97	**2.49**	0.85
+LS	7.25	8.56	6.43	7.87	9.88	11.46
+LS-原	7.12	8.37	6.33	8.16	10.01	10.89
+Mixup	7.63	3.46	9.67	4.52	6.75	2.75
+Mixup+LS	8.14	2.75	10.11	3.25	7.18	2.23
RoBERTa	**1.95**	**0.78**	**2.24**	0.97	**1.67**	**0.66**
RoBERTa-原	1.93	0.84	2.33	**0.88**	1.76	0.76
+LS	6.75	8.56	6.45	8.45	8.78	11.53
+LS-原	6.38	8.7	6.11	8.69	8.81	11.4
+Mixup	8.12	5.62	4.21	2.16	3.56	1.15
+Mixup+LS	6.17	1.78	7.12	4.21	2.63	0.87

通过表格 2、表格 3 我们可以得到一下结论：1）Label Smoothing 方法在 In-domain 的设置下，并不能对模型校准起到作用，相反它甚至提高了 ECE 值，但是在 Out-of-domain 的设置下，它能起到一定的作用，能对模型进行一定的校准。2）Mixup 这种简单的数据增强方法并不能对模型的校准起到很好的作用，但是当结合 Label Smoothing 的时候，它在某些任务上能起到良好的模型校准。3）采用 Temperature Scaling 时，不管它是和 Label Smoothing 还是 Mixup 进行结合都能起到很好的效果，对模型进行校准。4）在 Out-of-domain 时，模型的 ECE 都会比 In-domain 要高，说明模型在 Out-of-domain 时有很多过于自信的情况。

表 3: Out-of-domain 设置下，原论文以及复现结果在各种方法上 ECE 指标的结果，并附上了 Mixup 方法的结果

Out-of-domain						
模型/方法	MNLI		TwitterPPDB		HellaSWAG	
	无 TS	加 TS	无 TS	加 TS	无 TS	加 TS
BERT	6.51	3.98	8.95	6.54	13.23	11.45
BERT-原	7.03	3.61	8.51	7.15	12.62	12.83
+LS	4.28	3.85	6.86	5.23	**5.45**	6.12
+LS-原	**3.74**	4.05	**6.3**	5.78	5.73	5.34
+Mixup	18.72	3.58	12.35	5.13	11.23	4.35
+Mixup+LS	17.26	**2.61**	11.73	**4.98**	8.77	**4.13**
RoBERTa	3.89	2.06	9.78	8.12	12.09	12.12
RoBERTa-原	**3.62**	**1.46**	9.55	7.86	11.93	11.22
+LS	3.95	6.13	**8.85**	5.67	**3.21**	2.59
+LS-原	4.5	5.93	8.91	5.31	2.14	**2.23**
+Mixup	17.85	5.79	11.23	5.41	7.15	4.12
+Mixup+LS	13.12	1.95	9.47	**3.67**	7.56	3.12

6 总结与展望

总的来说，采用简单的数据增强（Label Smoothing、Mixup）方法对预训练语言模型进行训练在一定程度上能增强模型的泛化性，但是对模型的校准并不能起到很好的作用，所以关于数据增强方面的方法，或许可以寻找更加复杂的方法，例如在 Mixup 的时候使用一定的规则来进行样本的选择合成，而不是简单的使用随机方法，使得既能增强模型的表现也能提升模型的校准程度。

参考文献

[1] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[2] LIU Y, OTT M, GOYAL N, et al. Roberta: A robustly optimized bert pretraining approach[J]. arXiv preprint arXiv:1907.11692, 2019.

[3] JIANG X, OSL M, KIM J, et al. Calibrating predictive model estimates to support personalized medicine [J]. Journal of the American Medical Informatics Association, 2012, 19(2): 263-274.

[4] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[5] BRIER G W, et al. Verification of forecasts expressed in terms of probability[J]. Monthly weather review, 1950, 78(1): 1-3.

[6] RAFTERY A E, GNEITING T, BALABDAOUI F, et al. Using Bayesian model averaging to calibrate forecast ensembles[J]. Monthly weather review, 2005, 133(5): 1155-1174.

[7] PALMER T, DOBLAS-REYES F, WEISHEIMER A, et al. Toward seamless prediction: Calibration of climate change projections using seasonal forecasts[J]. Bulletin of the American Meteorological Society,

2008, 89(4): 459-470.

[8] KENDALL A, GAL Y. What uncertainties do we need in bayesian deep learning for computer vision? [J]. Advances in neural information processing systems, 2017, 30.

[9] GUO C, PLEISS G, SUN Y, et al. On calibration of modern neural networks[C]//International conference on machine learning. 2017: 1321-1330.

[10] LEE K, LEE H, LEE K, et al. Training confidence-calibrated classifiers for detecting out-of-distribution samples[J]. arXiv preprint arXiv:1711.09325, 2017.

[11] NGUYEN K, O'CONNOR B. Posterior calibration and exploratory analysis for natural language processing models[J]. arXiv preprint arXiv:1508.05154, 2015.

[12] KUMAR A, SARAWAGI S. Calibration of encoder decoder models for neural machine translation[J]. arXiv preprint arXiv:1903.00802, 2019.

[13] ZHANG H, CISSE M, DAUPHIN Y N, et al. mixup: Beyond empirical risk minimization[J]. arXiv preprint arXiv:1710.09412, 2017.

[14] THULASIDASAN S, CHENNUPATI G, BILMES J A, et al. On mixup training: Improved calibration and predictive uncertainty for deep neural networks[J]. Advances in Neural Information Processing Systems, 2019, 32.

[15] BOWMAN S R, ANGELI G, POTTS C, et al. A large annotated corpus for learning natural language inference[J]. arXiv preprint arXiv:1508.05326, 2015.

[16] WILLIAMS A, NANGIA N, BOWMAN S R. A broad-coverage challenge corpus for sentence understanding through inference[J]. arXiv preprint arXiv:1704.05426, 2017.

[17] IYER S, DANDEKAR N, CSERNAI K. Quora question pairs[J]. First Quora Dataset Release: Question Pairs, 2017.

[18] LAN W, QIU S, HE H, et al. A continuously growing dataset of sentential paraphrases[J]. arXiv preprint arXiv:1708.00391, 2017.

[19] ZELLERS R, BISK Y, SCHWARTZ R, et al. Swag: A large-scale adversarial dataset for grounded commonsense inference[J]. arXiv preprint arXiv:1808.05326, 2018.