

BERT:深度双向变压器的预训练 语言理解

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova Google
AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

抽象的

我们引入了一种称为BERT的新语言表示模型,它代表来自Transformers的双向编码器表示。与最近的语言表示模型(Peters等人,2018a;Radford等人,2018)不同,BERT旨在通过联合调节所有层中的左右上下文来预训练未标记文本的深度双向表示。因此,预训练的BERT模型可以仅通过一个额外的输出层进行微调,从而为广泛的任务创建最先进的模型,例如问答和语言推理,而无需大量特定任务架构修改。

将预训练语言表示应用于下游任务有两种现有策略:基于特征和微调。基于特征的方法,例如ELMo(Peters et al., 2018a),使用特定于任务的架构,其中包括预训练表示作为附加特征。微调方法,例如生成式预训练转换器(OpenAI GPT)(Radford等人,2018年),引入了最少的任务特定参数,并通过简单地微调所有预训练参数来对下游任务进行训练。这两种方法在预训练期间共享相同的目标函数,它们使用单向语言模型来学习通用语言表示。

BERT在概念上很简单,在经验上很强大。它在11个自然语言处理任务上获得了新的最先进的结果,包括将GLUE分数推至80.5%(7.7%点绝对提升),MultiNLI准确率提升至86.7%(4.6%绝对提升),SQuAD v1.1个问题回答测试F1至93.2(1.5分绝对改进)和SQuAD v2.0测试F1至83.1(5.1分绝对改进)。

我们认为当前的技术限制了预训练表示的能力,特别是对于微调方法。主要限制是标准语言模型是单向的,这限制了可以在预训练期间使用的架构的选择。例如,在OpenAI GPT中,作者使用从左到右的架构,其中每个标记只能倾向于Transformer的自我注意层中的先前标记(Vaswani等人,2017)。这种限制对于句子级别的任务来说是次优的,并且在将基于微调的方法应用于标记级别的任务(例如问答)时可能非常有害,在这些任务中,从两个方向结合上下文是至关重要的。

1 简介

语言模型预训练已被证明可有效改善许多自然语言处理任务(Dai和Le,2015年;Peters等人,2018a;Radford等人,2018年;Howard和Ruder,2018年)。这些包括自然语言推理(Bowman等人,2015年;Williams等人,2018年)和释义(Dolan和Brockett,2005年)等句子级任务,旨在通过整体分析来预测句子之间的关系,以及命名实体识别和问答等令牌级任务,其中模型需要在令牌级别生成细粒度输出(Tjong Kim Sang和De Meulder,2003年;Rajpurkar等人,2016年)。

在本文中,我们通过提出BERT: Bidirectional Encoder Representations from Transformers改进了基于微调的方法。

BERT受Cloze任务(Taylor,1953)的启发,通过使用“掩蔽语言模型”(MLM)预训练目标来减轻前面提到的unidirectionality约束。masked language model从输入中随机屏蔽一些token,目标是预测masked的原始词汇id

仅基于其上下文的词。与从左到右的语言模型预训练不同,MLM 目标使表示能够融合左右上下文,这使我们能够预训练深度双向 Transformer。除了掩码语言模型,我们还使用了“下一句预测”任务,联合预训练文本对表示。我们论文的贡献如下:

- 我们证明了双向预训练对语言表征的重要性。不像Radford 等人。(2018),它使用单向语言模型进行预训练,BERT 使用掩码语言模型来启用预训练的深度双向表示。这也与Peters 等人形成对比。(2018a),它使用独立训练的从左到右和从右到左的 LM 的浅层级联。

- 我们表明,预训练表示减少了对许多精心设计的任务特定架构的需求。BERT 是第一个基于微调的表示模型,它在大量句子级和标记级任务上实现了最先进的性能,优于许多特定于任务的架构。

- BERT 将最先进的技术提升了 11 年自然语言处理任务。代码和预训练模型可在<https://github.com/google-research/bert>获得。

2 相关工作

预训练通用语言表征有着悠久的历史,我们在本节中简要回顾了最广泛使用的方法。

2.1 无监督的基于特征的方法学习广泛适用的单词表示一直是一个活跃的研究领域

几十年,包括非神经方法(Brown 等人, 1992 年; Ando 和 Zhang, 2005 年; Blitzer 等人, 2006 年)和神经方法(Mikolov 等人, 2013 年; Pennington 等人, 2014 年)方法。预训练词嵌入是现代 NLP 系统不可或缺的一部分,与从头开始学习的嵌入相比有显着改进(Turian 等人, 2010 年)。为了预训练词嵌入向量,使用了从左到右的语言建模目标(Mnih 和 Hinton, 2009),以及区分左和右单词正确与错误的目标。

正确的上下文(Mikolov et al., 2013)。

这些方法已被推广到更粗粒度,例如句子嵌入(Kiros 等人, 2015 年; Logeswaran 和 Lee, 2018 年)或段落嵌入(Le 和 Mikolov, 2014 年)。为了训练句子表示,之前的工作使用目标对候选的下一个句子进行排名(Jernite 等人, 2017 年; Logeswaran 和 Lee, 2018 年),在给定前一个句子的表示的情况下从左到右生成下一个句子单词(Kiros et al., 2015),或去噪自动编码器衍生目标(Hill et al., 2016)。

ELMo 及其前身(Peters et al., 2017, 2018a)从不同的维度概括了传统的词嵌入研究。他们从从左到右和从右到左的语言模型中提取上下文相关的特征。每个标记的上下文表示是从左到右和从右到左表示的串联。

当将上下文词嵌入与现有的特定于任务的架构相结合时,ELMo 提高了几个主要 NLP 基准(Peters 等人, 2018a)的技术水平,包括问答(Rajpurkar 等人, 2016)、情感分析(Socher 等人 al., 2013)和命名实体识别(Tjong Kim Sang and De Meulder, 2003)。Melamud 等人。(2016)提出通过任务学习上下文表示,使用 LSTM 从左右上下文中预测单个单词。与 ELMo 类似,他们的模型是基于特征的,而不是深度双向的。费杜斯等人。(2018)表明完形填空任务可用于提高文本生成模型的稳健性。

2.2 无监督微调方法

与基于特征的方法一样,第一个在这个方向上的工作只是来自未标记文本的预训练词嵌入参数(Collobert 和 Weston, 2008)。

最近,生成上下文标记表示的句子或文档编码器已经从未标记的文本中进行了预训练,并针对受监督的下游任务进行了微调(Dai 和 Le, 2015 年; Howard 和 Ruder, 2018 年; Radford 等人, 2018 年)。这些方法的优点是需从头开始学习的参数很少。至少部分由于这一优势,OpenAI GPT (Radford 等人, 2018 年)在 GLUE 基准测试(Wang 等人, 2018a)的许多句子级任务上取得了先前最先进的结果。从左到右的语言模型-

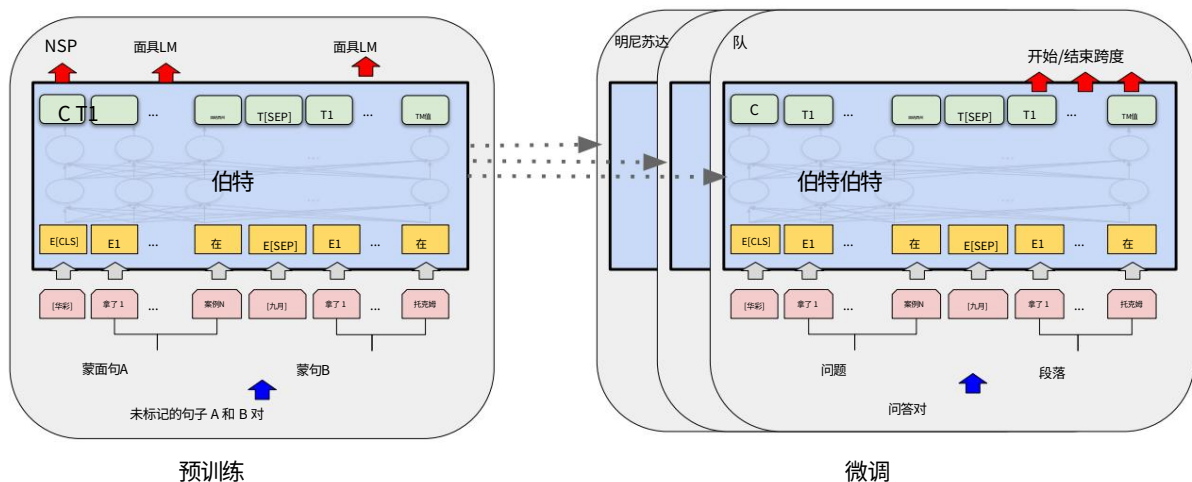


图 1:BERT 的整体预训练和微调程序。除了输出层之外,相同的架构还用于预训练和微调。相同的预训练模型参数用于为不同的下游任务初始化模型。在微调期间,对所有参数进行微调。[CLS] 是在每个输入示例前面添加的特殊符号,[SEP] 是特殊的分隔符(例如分隔问题/答案)。

ing 和自动编码器目标已用于预训练此类模型 (Howard 和 Ruder, 2018 年; Radford 等人, 2018 年; Dai 和 Le, 2015 年)。

2.3 从监督数据迁移学习

还有一些工作显示了从具有大型数据集的监督任务中进行有效迁移,例如自然语言推理(Conneau et al., 2017)和机器翻译(McCann et al., 2017)。计算机视觉研究也证明了从大型预训练模型进行迁移学习的重要性,其中一个有效的方法是微调使用 ImageNet 预训练的模型 (Deng 等人, 2009 年; Yosinski 等人, 2014 年)。

3 伯特

我们在本节中介绍了 BERT 及其详细实现。我们的框架有两个步骤:预训练和微调。在预训练期间,模型在不同的预训练任务上使用未标记数据进行训练。对于微调,BERT 模型首先使用预训练参数进行初始化,然后使用来自下游任务的标记数据对所有参数进行微调。每个下游任务都有单独的微调模型,即使它们是使用相同的预训练参数初始化的。图1中的问答示例将作为本节的运行示例。

BERT 的一个显著特点是其统一的 ar 跨不同任务的架构。有迷你

预训练架构与最终下游架构之间的差异。

Model Architecture BERT的model architec

ture 是一个多层双向 Transformer 编码器,基于Vaswani 等人描述的原始实现。(2017)并在 tensor2tensor 库中发布。1因为使用

变形金刚已经变得普遍,我们的即时通讯

实施与原始版本几乎相同,我们将省略对模型架构的详尽背景描述,并将读者推荐给Vaswani 等人。(2017)以及出色的指南,例如“The Annotated Transformer”。2在这项工作中,我们将层数 (即 Transformer 块)表示为 L,隐藏大小表示为

H,self-attention heads 的数量为 A_0 。

我们主要报告两种模型大小的结果: BERTBASE (L=12,H=768, A=12,总参数=110M)和BERTLARGE (L=24,H=1024,A=16,总参数=340M)。

出于比较目的,选择BERTBASE使其具有与 OpenAI GPT 相同的模型大小。

然而,至关重要是,BERT Transformer 使用双向自注意力,而 GPT Transformer 使用受限自注意力,其中每个标记只能关注其左侧的上下文。4

¹ <https://github.com/tensorflow/tensor2tensor>

² <http://nlp.seas.harvard.edu/2018/04/03/attention.html>

³ 在所有情况下,我们将前馈/滤波器大小设置为 4H,即,H = 768 时为 3072,H = 1024 时为 4096。

⁴我们注意到,在文献中,双向传输

输入/输出表示是为了让 BERT 处理各种下游任务,我们的输入表示能够在标记序列中明确表示单个句子和一对句子(例如,问题、答案)。

在这项工作中,“句子”可以是连续文本的任意跨度,而不是实际的语言句子。“序列”是指输入给 BERT 的 token 序列,它可以是单个句子,也可以是打包在一起的两个句子。

我们使用具有 30,000 个标记词汇表的 WordPiece 嵌入 (Wu 等人, 2016 年)。每个序列的第一个标记总是一个特殊的分类标记 ([CLS])。该标记对应的最终隐藏状态用作分类任务的聚合序列表示。句子对被打包成一个序列。我们以两种方式区分句子。首先,我们用特殊标记 ([SEP]) 将它们分开。其次,我们向每个标记添加一个学习嵌入 d_{seg} ,指示它是属于句子 A 还是句子 B。如图 1 所示,我们将输入嵌入表示为 E ,特殊 [CLS] 标记的最终隐藏向量为 $C \in \mathbb{R}^H$,

和 i 的最终隐藏向量 h_i 输入令牌 x_i 因为 $T_i \in \mathbb{R}^H$ 。

对于给定的标记,其输入表示是通过对相应的标记、段和位置嵌入求和来构建的。这种结构的可视化可以在图2中看到。

3.1 预训练BERT

与彼得斯等人不同。(2018a)和Radford 等人。(2018),我们不使用传统的从左到右或从右到左的语言模型来预训练 BERT。

相反,我们使用本节中描述的两个无监督任务对 BERT 进行预训练。此步骤显示在图1 的左侧部分。

任务 #1:Masked LM凭直觉,我们有理由相信深度双向模型比从左到右的模型或从左到右的浅层级联模型更强大

右和从右到左的模型。不幸的是,标准的条件语言模型只能从左到右或从右到左进行训练,因为双向条件允许每个词直接“看到自己”,并且模型可以简单地预测多个词中的目标词-分层上下文。

为了训练深度双向表示,我们简单地随机屏蔽一定比例的输入标记,然后预测这些屏蔽的标记。我们将此过程称为“掩蔽 LM”(MLM),尽管它在文献中通常被称为完形填空任务(Taylor, 1953)。在这种情况下,与掩码标记对应的最终隐藏向量被馈送到词汇表上的输出 softmax,就像在标准 LM 中一样。在我们所有的实验中,我们将所有 WordPiece 的 15% 随机屏蔽为每个序列中的 [UNK] 。与去噪自动编码器(Vincent et al., 2008) 相比,我们只预测屏蔽词而不是重建整个输入。

虽然这使我们能够获得双向预训练模型,但缺点是在预训练和微调之间造成了不匹配,因为 [MASK] 标记在微调期间不会出现。为了缓解这种情况,我们并不总是用实际的 [MASK] 标记替换“掩码”词。训练数据生成器随机选择 15% 的标记位置进行预测。如果选择了第 i 个标记,我们将第 i 个标记替换为 (1) [MASK] 标记 80% 的时间 (2) 随机标记 10% 的时间 (3) 未更改的第 i 个标记 10% 的时间。然后, T_i 将用于预测具有交叉熵损失的原始标记。我们在附录 C.2 中比较了此过程的变体。

任务 #2:下一句预测 (NSP)

许多重要的下游任务,如问答 (QA)和自然语言推理 (NLI),都是基于理解两个句子之间的关系,而语言建模并不能直接捕捉到这种关系。为了训练一个理解句子关系的模型,我们预训练了一个二值化的下一个句子预测任务,它可以从任何单语语料库中简单地生成。具体来说,当为每个预训练示例选择句子 A 和 B 时,50% 的时间 B 是 A 之后的实际下一个句子 (标记为 IsNext),而50% 的时间是来自 A 的随机句子

语料库 (标记为 NotNext)。正如我们在图 1 中所示,C 用于下一句预测 (NSP)。5尽管它很简单,但我们在第 5.1 节中证明了针对此任务的预训练对 QA 和 NLI 都非常有益。

前者通常被称为“变压器编码器”,而

left-context-only 版本被称为“Transformer decoder”,因为它可以用于文本生成。

5最终模型在NSP上达到了97%-98%的准确率。

6向量C不是有意义的句子表示没有微调,因为它用 NSP 训练的。

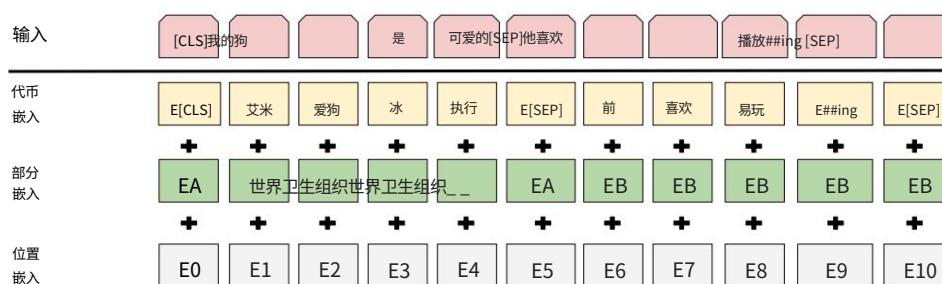


图 2:BERT 输入表示。输入嵌入是标记嵌入、分割嵌入和位置嵌入的总和。

NSP 任务与 Jernite 等人使用的表示学习目标密切相关。(2017) 以及 Logeswaran 和 Lee (2018)。然而,在之前的工作中,只有句子嵌入被转移到下游任务,其中 BERT 转移所有参数以初始化结束任务模型参数。

预训练数据预训练过程在很大程度上遵循了关于语言模型预训练的现有文献。对于预训练语料库,我们使用 BooksCorpus (800M 词) (Zhu et al., 2015) 和英语维基百科 (2,500M 词)。

对于维基百科,我们只提取文本段落并忽略列表、表格和标题。为了提取长的连续序列,使用文档级语料库而不是诸如 Billion Word Benchmark (Chelba et al., 2013) 之类的打乱句子级语料库至关重要。

3.2 微调 BERT 微调非常简单,因

为 Transformer 中的自注意力机制允许 BERT 通过交换适当的输入和输出来模拟许多下游任务 无论它们涉及单个文本还是文本对。

对于涉及文本对的应用程序,一种常见的模式是在应用双向交叉注意力之前独立编码文本对,例如 Parikh 等人。(2016);徐等。(2017)。BERT 使用自注意力机制来统一这两个阶段,因为使用自注意力编码串联文本对有效地包括两个句子之间的双向交叉注意力。

对于每项任务,我们只需将任务特定的输入和输出插入 BERT,然后端到端地微调所有参数。在输入端,预训练的句子 A 和句子 B 类似于 (1) 释义中的句子对, (2) 蕴含中的假设-前提对, (3) 问答中的问题-段落对,以及

(4) 文本分类或序列标注中的退化文本-Ø 对。在输出端,令牌表示被馈送到令牌级任务的输出层,例如序列标记或问题回答,[CLS] 表示被馈送到输出层用于分类,例如蕴含或情感分析。

与预训练相比,微调相对便宜。从完全相同的预训练模型开始,在单个 Cloud TPU 上最多可在 1 小时内或在 GPU 上数小时内复制本文中的所有结果。⁷ 我们描述了特定于任务的细节在第 4 节的相应小节中。可以在附录 A.5 中找到更多信息。

4 实验

在本节中,我们将展示 BERT 在 11 个 NLP 任务上的微调结果。

4.1 胶水

通用语言理解评估 (GLUE) 基准 (Wang et al., 2018a) 是多种自然语言理解任务的集合。GLUE 数据集的详细描述包含在附录 B.1 中。

为了在 GLUE 上进行微调,我们表示第 3 节中描述的输入序列 (对于单个句子或句子对),并使用对应于第一个输入标记 ([CLS]) 的最终隐藏向量 $C \in \mathbb{R}^H$ 作为聚合代表。微调期间引入的唯一新参数是分类权重 $W \in \mathbb{R}^{H \times V}$ 和偏置 $b \in \mathbb{R}^V$,其中 V 是词汇表的大小。我们使用交叉熵损失,即 $\log(\text{softmax}(CWT))$ 。

⁷ 例如,可以在单个 Cloud TPU 上训练 BERT SQuAD 模型大约 30 分钟,以达到 91.0% 的 Dev F1 分数。8 个

系统	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	平均值	8.5k	3.5k
	392k	363k	108k	67k					5.7k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0		
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0		
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1		
贝特贝斯	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6		
贝特大	86.7/85.9	72.1	92.7	94.9	60.5	86.5	89.3	70.1	82.1		

表 1:GLUE 测试结果,由评估服务器(<https://gluebenchmark.com/leaderboard>) 评分。每个任务下方的数字表示训练示例的数量。“平均”列与官方 GLUE 分数略有不同,因为我们排除了有问题的 WNLI 集。8 BERT 和 OpenAI GPT 是单一模型、单一任务。QQP 和 MRPC 报告了 F1 分数,STS-B 报告了 Spearman 相关性,其他任务报告了准确性分数。我们排除了使用 BERT 作为其组件之一的条目。

我们使用 32 的批量大小并微调 3 遍历所有 GLUE 任务的数据。对于每个任务,我们在开发集上选择了最佳微调学习率（在 5e-5、4e-5、3e-5 和 2e-5 之间）。

此外,对于BERTLARGE,我们发现微调有时在小型数据集上不稳定,因此我们运行了几次随机重启并选择了开发集上的最佳模型。通过随机重新启动,我们使用相同的预训练检查点,但执行不同的微调数据混洗和分类器层初始化。9结果如表1所示。BERTBASE和BERTLARGE在所有任务上的表现都大大优于所有系统,与现有技术水平相比,平均准确度分别提高了 4.5% 和 7.0%。请注意,除了注意掩蔽之外, BERTBASE和 OpenAI GPT 在模型架构方面几乎相同。对于最大和最广泛报道的 GLUE 任务 MNLI,BERT 获得了 4.6% 的绝对精度提升。在官方 GLUE 排行榜10 上, BERTLARGE获得 80.5 分,而 OpenAI GPT 在撰写本文时获得 72.8 分。

我们发现BERTLARGE在所有任务上都明显优于BERTBASE , 尤其是那些训练数据很少的任务。模型大小的影响在5.2 节中进行了更彻底的探讨。

4.2 小队 v1.1

斯坦福问答数据集 (SQuAD v1.1) 是 10 万个众包问答对的集合 (Rajpurkar 等人, 2016 年) 。给定一个问题和一段来自

包含答案的维基百科,任务是预测段落中答案文本的跨度。

如图1 所示,在问答任务中,我们将输入问题和文章表示为单个打包序列,问题使用 A 嵌入,文章使用 B 嵌入。我们只在微调期间引入一个起始 vec。单词 i 作为答案跨度开始的概率计算为 T_i 和 S 之间的点积,然后是对段落中所有单词的 $S \cdot T_i$ 的 softmax : $P_i = \frac{S \cdot T_i}{\sum_j S \cdot T_j}$ 。器 $S \in \mathbb{R}^H$ 和一个结束向量 $E \in \mathbb{R}^H$

类似的公式用于答案跨度的结尾。从位置 i 到位置 j 的候选跨度的得分定义为 $S \cdot T_i + E \cdot T_j$,并将 $j \geq i$ 的最大得分跨度用作预测。训练目标是正确开始和结束位置的对数似然之和。我们以 5e-5 的学习率和 32 的批量大小对 3 个 epoch 进行微调。

表2显示了顶级排行榜条目以及顶级已发布系统的结果 (Seo 等人, 2017 年; Clark 和 Gardner, 2018 年; Peters 等人, 2018a; Hu 等人, 2018 年) 。SQuAD 排行榜的顶级结果没有可用的最新公共系统描述,11并且在训练他们的系统时允许使用任何公共数据。

因此,我们通过首先对 TriviaQA (Joshi 等人, 2017) 进行微调,然后再对 SQuAD 进行微调,在我们的系统中使用适度的数据增强。

我们表现最好的系统在集成方面比顶级排行榜系统高出 +1.5 F1,在单个系统中高出 +1.3 F1。事实上,我们的单一 BERT 模型在 F1 分数方面优于顶级集成系统。没有 TriviaQA 没问题

9 GLUE 数据集分布不包括测试标签,我们只为BERTBASE和BERTLARGE分别进行了一次 GLUE 评估服务器提交。10<https://gluebenchmark.com/leaderboard>

Yu 等人描述了 11QANet。(2018),但该系统在发布后有了很大改进。

系统	开发 在 F1 在 F1	测试
顶级排行榜系统 (2018 年 12 月 10 日)		
人类	-	- 82.3 91.2 - 86.0
#1 一起 - nlnet	-	91.7 - 84.5 90.5
#2 一起 QANet	-	
发表		
BiDAF+ELMo (单)	- 85.6 - 85.8	81.2 87.9 82.3
RM 读者 (合奏)	88.5	
我们的		
BERTBASE (单)	80.8 88.5 - 84.1	-
BERTLARGE (单人)	90.9 - 85.8	91.8 -
BERTLARGE (合奏)		-
BERTLARGE (Sgl.+TriviaQA)	84.2 91.1 85.1	91.8 BERTLARGE (Ens. +TriviaQA)

表 2:SQuAD 1.1 结果。BERT 集成是 7x 系统,它使用不同的预训练检查点和微调种子。

系统	开发 在 F1 在 F1	测试
顶级排行榜系统 (2018 年 12 月 10 日)		
人类 86.3 89.0 86.9 89.5 #1 Single - MIR-MRC (F-Net) - - 74.8 78.0 #2 Single - nlnet - 74.2 77.1	-	
发表		
unet (一起)	-	- 71.4 74.9 71.4
SLQA+ (单身)	-	74.4
我们的		
BERTLARGE (单人)	78.7 81.9 80.0	83.1

表 3:SQuAD 2.0 结果。我们排除了使用 BERT 作为其组件之一的条目。

调整数据,我们仅损失 0.1-0.4 F1,仍然大大优于所有现有系统。
12

4.3 小队 v2.0

SQuAD 2.0 任务扩展了 SQuAD 1.1 问题定义,允许在提供的段落中不存在简短答案的可能性,使问题更加现实。

我们使用一种简单的方法来扩展 SQuAD v1.1 BERT 模型来完成这项任务。我们将没有答案的问题视为具有从 [CLS] 开始和结束的答案范围。开始和结束答案跨度位置的概率空间被扩展到包括 [CLS] 标记的位置。对于预测,我们将无答案跨度的分数: $s_{null} = S \cdot C + E \cdot C$ 与最佳非空跨度的分数进行比较

系统	开发测试
ESIM+手套	51.9 52.7
ESIM+ELMo	59.1 59.2 -
OpenAI GPT	78.0
贝特贝斯	81.6 - 86.6
贝特大	86.3
人类 (专家)+ - 85.0 人类 (5 个注释)+ - 88.0	

表 4:SWAG 开发和测试精度。†如 SWAG 论文中所报告的,人类的表现是通过 100 个样本来衡量的。

$s^{\wedge}_{i,j} = \max_j \geq iS \cdot Ti + E \cdot Tj$ 。当 $s^{\wedge}_{i,j} > s_{null} + \tau$ 时,我们预测一个非空答案,脱粒的地方在开发集上选择旧的 τ 以最大化 F1。

我们没有为这个模型使用 TriviaQA 数据。我们微调了 2 个 epoch,学习率为 5e-5,batch size 为 48。

表 3 显示了与之前的排行榜条目和顶级发表作品 (Sun et al., 2018; Wang et al., 2018b)相比的结果,不包括使用 BERT 作为其组件之一的系统。我们观察到 F1 比之前的最佳系统提高了 +5.1。

4.4 赃物

Adversarial Generations (SWAG) 数据集包含 113k 个句子对完成示例,用于评估基于常识的推理 (Zellers 等人, 2018 年)。给定一个句子,任务是在四个选项中选择最合理的延续。

在对 SWAG 数据集进行微调时,我们构建了四个输入序列,每个序列都包含给定句子 (句子 A)和可能的延续 (句子 B)的串联。引入的唯一特定于任务的参数是一个向量,其与 [CLS] 标记表示 C 的点积表示每个选择的分数,该分数用 softmax 层归一化。

我们用 2e-5 的学习率和 16 的批量大小对模型进行了 3 个周期的微调。结果如表 4 所示。BERTLARGE 比作者的基线ESIM+ELMo 系统高出 +27.1%,OpenAI GPT 8.3%。

5 消融研究

在本节中,我们对 BERT 的多个方面进行消融实验,以便更好地了解它们的相对重要性。额外的

12我们使用的 TriviaQA 数据由来自 TriviaQA-Wiki 的段落组成,这些段落由文档中的前 400 个标记组成,其中至少包含一个提供的可能答案。

任务	开发集				
	MNLI-m	QNLI	MRPC	SST-2	SQuAD
	(加速度)	(加速度)	(加速度)	(加速度)	(F1)
BERTBASE	84.4	无 NSP 83.9	88.4	86.7	84.9
LTR & 无 NSP	82.1	+ BiLSTM 82.1	86.5	84.3	77.5
			84.1	75.7	
				92.1	77.8
				91.6	84.9

表 5:使用BERTBASE架构对预训练任务进行消融。“No NSP”是在没有下一句预测任务的情况下训练的。“LTR & No NSP”被训练为一个从左到右的 LM,没有下一句预测,就像 OpenAI GPT。“+ BiLSTM”在微调期间在“LTR + No NSP”模型之上添加了一个随机初始化的 BiLSTM。

消融研究可以在附录C中找到。

5.1 预训练任务的效果

我们通过使用与BERTBASE完全相同的预训练数据、微调方案和超参数评估两个预训练目标来证明 BERT 的深度双向反应的重要性:

无 NSP:经过训练的双向模型

使用“masked LM”(MLM)但没有“下一句预测”(NSP)任务。

LTR & No NSP:使用标准的从左到右 (LTR) 训练的左上下文模型

LM,而不是传销。仅左约束也适用于微调,因为移除它会引入预训练/微调不匹配,从而降低下游性能。此外,该模型是在没有 NSP 任务的情况下进行预训练的。

这可以直接与 OpenAI GPT 相媲美,但使用我们更大的训练数据集、我们的输入表示和我们的微调方案。

我们首先检查 NSP 任务带来的影响。在表5 中,我们表明移除 NSP 会显着影响 QNLI、MNLI 和 SQuAD 1.1 的性能。接下来,我们通过比较“无 NSP”和“LTR & No NSP”来评估训练双向表示的影响。LTR 模型在所有任务上的表现都比 MLM 模型差,在 MRPC 和 SQuAD 上有较大下降。

对于 SQuAD,直觉上很明显 LTR 模型在令牌预测方面表现不佳,因为令牌级隐藏状态没有右侧上下文。为了善意地尝试加强 LTR 系统,我们在顶部添加了一个随机初始化的 BiLSTM。这确实显着改善了 SQuAD 的结果,但是

结果仍然比预训练的双向模型差得多。BiLSTM 会损害 GLUE 任务的性能。

我们认识到,也可以训练单独的 LTR 和 RTL 模型,并将每个标记表示为两个模型的串联,就像 ELMo 所做的那样。但是:(a) 这比单个双向模型贵两倍;(b) 这对于 QA 之类的任务来说是不直观的,因为 RTL 模型无法将问题的答案作为条件;(c) 它严格来说不如深度双向模型强大,因为它可以在每一层同时使用左右上下文。

5.2 模型大小的影响

在本节中,我们探讨了模型大小对微调任务准确性的影响。我们训练了许多具有不同层数、隐藏单元和注意力头的 BERT 模型,同时使用与之前描述的相同的超参数和训练过程。

选定 GLUE 任务的结果显示在

表6. 在此表中,我们报告了 5 次随机重新启动微调的平均开发集准确度。

我们可以看到,较大的模型导致所有四个数据集的精确度都得到了严格的提高,即使对于只有 3,600 个标记训练示例并且与预训练任务有很大不同的 MRPC 也是如此。同样令人惊讶的是,我们能够相对于现有文献已经相当大的模型之上实现如此显着的改进。

例如, Vaswani 等人探索的最大的变形金刚。(2017)是(L=6, H=1024, A=16) encoder参数100M,我们在文献中找到的最大 Transformer是(L=64, H=512, A=2) 235M参数 (Al-Rfou 等人, 2018 年)。相比之下, BERTBASE包含 110M 个参数, BERTLARGE包含 340M 个参数。

人们早就知道,增加模型大小将导致机器翻译和语言建模等大规模任务的持续改进,表 6 中显示的保留训练数据的 LM 困惑证明了这一点。但是,我们相信这是第一项令人信服地证明扩展到极端模型大小也会导致非常小规模任务的大幅改进的工作,前提是模型已经过充分的预训练。彼得斯等人。(2018b)提出

将预训练的双 LM 大小从两层增加到四层对下游任务影响的混合结果, Melamud 等人。(2016)顺便提到将隐藏维度大小从 200 增加到 600 有所帮助,但进一步增加到 1,000 并没有带来进一步的改进。这两项先前的工作都使用了基于特征的方法。我们假设当模型直接在下游任务上进行微调并且只使用非常少量的随机初始化的附加参数时,特定于任务的模型可以受益于更大的、更具表现力的预训练表示,即使下游任务数据非常小。

5.3 BERT 基于特征的方法

到目前为止呈现的所有 BERT 结果都使用了微调方法,即在预训练模型中添加一个简单的分类层,并在下游任务上联合微调所有参数。然而,从预训练模型中提取固定特征的基于特征的方法具有一定的优势。首先,并非所有任务都可以轻松地由 Transformer 编码器架构表示,因此需要添加特定于任务的模型架构。

其次,预先计算训练数据的昂贵表示一次,然后在该表示之上使用更便宜的模型运行许多实验,有很大的计算优势。

在本节中,我们通过将 BERT 应用于 CoNLL-2003 命名实体识别 (NER) 任务 (Tjong Kim Sang 和 De Meulder, 2003)来比较这两种方法。在 BERT 的输入中,我们使用了一个保留大小写的 WordPiece 模型,并且我们包含了数据提供的最大文档上下文。按照标准做法,我们将其模拟为标记任务但不使用 CRF

超参数	开发集准确性		
#L #H #A LM (ppl) MNLI-m MRPC SST-2			
3 768 12 5.84 6 768 3 5.24	77.9	79.8	88.4
6 768 12 4.68 12 768 12.9	80.6	82.2	90.7
12 1024 16 3.54 24 1024	81.9	84.8	91.3
16 3.23	84.4	92.9	86.9
	85.7	87.8	93.7
	86.6		

表 6:BERT 模型大小的消融。#L = 层数; #H = 隐藏尺寸; #A = 注意头的数量。“LM (ppl)”是保留训练数据的掩蔽 LM 困惑度。

系统	开发 F1 测试 F1	
ELMo (Peters 等人, 2018a)	95.7	92.2
CVT (Clark 等人, 2018 年)	-	92.6
CSE (Akbik 等人, 2018 年)	-	93.1
微调方法		
贝特大	96.6	92.8
贝特贝斯	96.4	92.4
基于特征的方法(BERTBASE)		
嵌入	91.0	-
倒数第二个隐藏	95.6	-
最后隐藏	94.9	-
加权和后四隐藏	95.9	-
Concat 最后四个隐藏	96.1	-
所有 12 层的加权总和	95.5	-

表 7:CoNLL-2003 命名实体识别结果。使用开发集选择超参数。报告的开发和测试分数是使用这些超参数对 5 次随机重启进行平均的。

输出层。我们使用第一个子标记的表示作为 NER 标签集上标记级分类器的输入。

为了消除微调方法,我们通过从一个或多个层中提取激活来应用基于特征的方法,而无需微调 BERT 的任何参数。这些上下文嵌入被用作分类层之前随机初始化的两层 768 维 BiLSTM 的输入。

结果如表 7 所示。BERTLARGE 的表现与最先进的方法相比具有竞争力。性能最好的方法连接来自预训练 Transformer 前四个隐藏层的标记表示,这仅比微调整个模型落后 0.3 F1。这表明 BERT 对于微调和基于特征的方法都是有效的。

六,结论

最近由于使用语言模型进行迁移学习而带来的实证改进表明,丰富的、无监督的预训练是许多语言理解系统不可或缺的一部分。特别是,这些结果甚至使低资源任务也能从深度单向架构中受益。我们的主要贡献是进一步将这些发现推广到深度双向架构,使相同的预训练模型能够成功处理广泛的 NLP 任务。

参考

艾伦·阿克比克、邓肯·布莱斯和罗兰·沃尔格拉夫。

2018. 用于序列标记的上下文字符串嵌入。在第 27 届国际计算语言学会议论文集中,第 1638-1649 页。

Rami Al-Rfou,Dokook Choe,Noah Constant,Mandy Guo 和 Llion Jones。 2018. 具有更深层次自我关注的字符级语言建模。arXiv 预印本 arXiv:1808.04444。

久保田理惠和张桐。 2005. 从多个任务和未标记数据中学习预测结构的框架。机器学习研究杂志,6 (十一月) :1817-1853。

Luisa Bentivogli,Bernardo Magnini,Ido Dagan,Hoa Trang Dang 和 Danilo Giampiccolo。 2009. 第五次 PASCAL 识别文本蕴涵挑战。在 TAC 中。美国国家标准技术研究院。

约翰·布利策、瑞安·麦克唐纳和费尔南多·佩雷拉。

2006. 域适应与结构对应学习。在 2006 年自然语言处理经验方法会议记录中,第 120-128 页。计算语言学协会。

Samuel R. Bowman,Gabor Angeli,Christopher Potts 和 Christopher D. Manning。 2015. 用于学习自然语言推理的大型带注释语料库。在 EMNLP 中。计算语言学协会。

Peter F Brown,Peter V Desouza,Robert L Mercer,Vincent J Della Pietra 和 Jenifer C Lai。 1992. 基于类的自然语言 n-gram 模型。计算语言学,18(4):467-479。

Daniel Cer,Mona Diab,Eneko Agirre,Inigo Lopez Gazpio 和 Lucia Specia。 2017. [Semeval-2017任务 1:语义文本相似性多语言和跨语言重点评估](#)。在第 11 届国际语义评估研讨会 (SemEval-2017) 的会议记录中,第 1-14 页,加拿大温哥华。计算语言学协会。

Ciprian Chelba,Tomas Mikolov,Mike Schuster,Qi Ge,Thorsten Brants,Phillipp Koehn 和 Tony Robin 儿子。 2013. 用于衡量统计语言建模进展的十亿字基准。arXiv 预印本 arXiv:1312.3005。

Z. Chen, H. Zhang, X. Zhang, and L. Zhao. 2018. [Quora 问题对](#)。

克里斯托弗·克拉克和卡特·加德纳。 2018. 简单有效的多段阅读理解。在 ACL 中。

Kevin Clark,Minh-Thang Luong,Christopher D Manning 和 Quoc Le。 2018. 具有交叉视图训练的半监督序列建模。在 2018 年自然语言处理经验方法会议论文集中,第 1914-1925 页。

罗南科洛伯特和杰森韦斯顿。 2008. 自然语言处理的统一架构:具有多任务学习的深度神经网络。在第 25 届机器学习国际会议论文集中,第 160-167 页。美国计算机协会。

Alexis Conneau,Douwe Kiela,Holger Schwenk,Loïc Barrault 和 Antoine Bordes。 2017.从自然语言推理数据中[监督学习通用句子表示](#)。在 2017 年自然语言处理经验方法会议记录中,第 670-680 页,丹麦哥本哈根。计算语言学协会。

Andrew M Dai 和 Quoc V Le。 2015. 半监督序列学习。在神经信息处理系统的进展中,第 3079-3087 页。

J. Deng,W. Dong,R. Socher,L.-J. Li,K. Li 和 L. Fei Fei。 2009. ImageNet:大规模分层图像数据库。在 CVPR09 中。

威廉·B·多兰和克里斯·布罗克特。 2005. 自动构建句子释义语料库。在第三届国际释义研讨会 (IWP2005) 的记录中。

William Fedus,Ian Goodfellow 和 Andrew M Dai。 2018. Maskgan:通过填写更好的文本生成。arXiv 预印本 arXiv:1801.07736。

丹·亨德里克斯和凯文·金佩尔。 2016.[用高斯误差线性单元桥接非线性和随机正则化器](#)。CoRR,abs/1606.08415。

Felix Hill,Kyunghyun Cho 和 Anna Korhonen。 2016. 从未标记的数据中学习句子的分布式表示。在计算语言学协会北美分会 2016 年会议记录中:人类语言技术。计算语言学协会。

杰里米霍华德和塞巴斯蒂安鲁德。 2018.用于文本分类的[通用语言模型微调](#)。在 ACL 中。计算语言学协会。

Minghao Hu, Yuxing Peng, Zhen Huang, Xipeng Qiu, Furu Wei, and Ming Zhou. 2018. Reinforced mnemonic reader for machine reading comprehension. In IJCAI.

Yacine Jernite,Samuel R. Bowman 和 David Son 标签。 2017.[快速无监督句子表示学习的基于话语的目标](#)。CoRR,abs/1705.00557。

Mandar Joshi,Eunsol Choi,Daniel S Weld 和 Luke Zettlemoyer。 2017. Triviaqa:用于阅读理解的大规模远程监督挑战数据集。在 ACL 中。

Ryan Kiros,Yukun Zhu,Ruslan R Salakhutdinov,Richard Zemel,Rachel Urtasun,Antonio Torralba 和 Sanja Fidler。 2015. 跳过思想向量。在神经信息处理系统的进展中,第 3294-3302 页。

国乐和托马斯·米科洛夫。 2014. 句子和文档的分布式表示。在机器学习国际会议上,第 1188-1196 页。

Hector J Levesque,Ernest Davis 和 Leora Morgen 斯特恩。 2011. winograd 模式挑战。在 Aai 春季研讨会:常识推理的逻辑形式化,第 46 卷,第 47 页。

Lajanugen Logeswaran 和 Honglak Lee。 2018.学习句子表示的有效框架。在国际学习代表大会上。

Bryan McCann,James Bradbury,Caiming Xiong 和 Richard Socher。 2017. 在翻译中学习:语境化词向量。在 NIPS 中。

Oren Melamud,Jacob Goldberger 和 Ido Dagan。 2016. context2vec:使用双向 LSTM 学习通用上下文嵌入。在 CoNLL 中。

Tomas Mikolov,Ilya Sutskever,Kai Chen,Greg S Corrado 和 Jeff Dean。 2013. 单词和短语的分布式表示及其组合性。神经信息处理系统进展 26,第 3111-3119 页。柯伦联合公司

Andrew Mnih 和 Geoffrey E Hinton。 2009.可扩展的分层分布式语言模型。由 D. Koller,D. Schuurmans,Y. Bengio 和 L. Bottou 编辑,神经信息处理系统进展 21,第 1081-1088 页。Curran As Partners, Inc.

Ankur P Parikh,Oscar Tackstrom,Dipanjan Das 和 Jakob Uszkoreit。 2016. 一种用于自然语言推理的可分解注意力模型。在 EMNLP 中。

Jeffrey Pennington,Richard Socher 和 Christopher D. Manning。 2014.手套:用于单词表示的全局向量。在自然语言处理中的经验方法 (EMNLP),第 1532-1543 页。

Matthew Peters,Waleed Ammar,Chandra Bhagavathula 和 Russell Power。 2017. 带有双向语言模型的半监督序列标注。在 ACL 中。

马修·彼得斯、马克·纽曼、莫希特·艾耶、马特·加德纳、克里斯托弗·克拉克、肯顿·李和卢克·泽特莫耶。 2018a.深度语境化的词表示。在 NAACL 中。

马修·彼得斯、马克·纽曼、卢克·泽特莫耶和叶文头。 2018b.剖析上下文词嵌入:架构和表示。

在 2018 年自然语言处理经验方法会议论文集中,第 1499-1509 页。

亚历克·拉德福德、卡尔西克·纳拉西姆汉、蒂姆·萨利曼斯和伊利亚·苏茨克维尔。 2018. 通过无监督学习提高语言理解力。技术报告,OpenAI。

Pranav Rajpurkar,Jian Zhang,Konstantin Lopyrev 和 Percy Liang。 2016. Squad:机器理解文本的 100,000 多个问题。在 2016 年自然语言处理经验方法会议记录中,第 2383-2392 页。

Minjoon Seo,Aniruddha Kembhavi,Ali Farhadi 和 Hannaneh Hajishirzi。 2017. 机器理解的双向注意力流。在 ICLR 中。

Richard Socher,Alex Perelygin,Jean Wu,Jason Chuang,Christopher D Manning,Andrew Ng 和 Christopher Potts。 2013. 情感树库语义组合的递归深度模型。在 2013 年自然语言处理经验方法会议论文集中,第 1631-1642 页。

Fu Sun, Linyang Li, Xipeng Qiu, and Yang Liu. 2018. U-net:带有无法回答问题的机器阅读理解。 arXiv 预印本 arXiv:1810.06638。

威尔逊·L·泰勒。 1953. 完形填空程序:衡量可读性的新工具。新闻公报,30(4):415-433。

Erik F Tjong Kim Sang 和 Fien De Meulder。 2003. conll-2003 共享任务简介:独立于语言的命名实体识别。在 CoNLL 中。

Joseph Turian,Lev Ratinov 和 Yoshua Bengio。 2010. 词表示:一种简单通用的半监督学习方法。在计算语言学协会第 48 届年会会议记录中,ACL 10,第 384-394 页。

Ashish Vaswani,Noam Shazeer,Niki Parmar,Jakob Uszkoreit,Llion Jones,Aidan N Gomez,Lukasz Kaiser 和 Illia Polosukhin。 2017. 注意力就是你所需要的。在神经信息处理系统的进展中,第 6000-6010 页。

Pascal Vincent,Hugo Larochelle,Yoshua Bengio 和 Pierre-Antoine Manzagol。 2008. 使用去噪自动编码器提取和组合稳健的特征。第 25 届国际机器学习会议论文集,第 1096-1103 页。

美国计算机协会。

Alex Wang,Amanpreet Singh,Julian Michael,Felix Hill,Omer Levy 和 Samuel Bowman。 2018a. Glue:多任务基准和分析平台

用于自然语言理解。在 2018 年 EMNLP 研讨会论文集 BlackboxNLP:NLP 的分析和解释神经网络,第 353-355 页。

魏王、明衍、陈武。2018b。用于阅读理解和问答的多粒度层次注意力融合网络。

在计算语言学协会第 56 届年会论文集 (第 1 卷:长篇论文) 中。计算语言学协会。

Alex Warstadt、Amanpreet Singh 和 Samuel R Bowman。2018。神经网络可接受性判断。arXiv 预印本 arXiv:1805.12471。

Adina Williams、Nikita Nangia 和 Samuel R Bowman。2018。通过推理理解句子的广泛覆盖挑战语料库。在 NAACL 中。

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016。Google 的神经机器翻译系统:弥合人类与机器翻译之间的差距。arXiv 预印本 arXiv:1609.08144。

Jason Yosinski、Jeff Clune、Yoshua Bengio 和 Hod Lipson。2014。深度神经网络中的特征如何可转移?在神经信息处理系统的进展中,第 3320-3328 页。

Adams Wei Yu、David Dohan、Minh-Thang Luong、Rui Zhao、Kai Chen、Mohammad Norouzi 和 Quoc V Le。2018。QANet:将局部卷积与全局自注意力相结合以进行阅读理解。在 ICLR 中。

Rowan Zellers、Yonatan Bisk、Roy Schwartz 和 Yejin Choi。2018。Swag:用于基础常识推理的大规模对抗数据集。在 2018 年自然语言处理经验方法会议 (EMNLP) 会议记录中。

Yukun Zhu、Ryan Kiros、Rich Zemel、Ruslan Salakhutdinov、Raquel Urtasun、Antonio Torralba 和 Sanja Fidler。2015。对齐书籍和电影:通过看电影和阅读书籍来实现故事般的视觉解释。在 IEEE 计算机视觉国际会议论文集中,第 19-27 页。

“BERT:预训练深度双向变压器语言理解”

我们将附录分为三个部分:

- BERT 的其他实施细节在附录 A 中介绍;

- 附录B 中提供了我们实验的更多详细信息;和

- 附录C中介绍了其他消融研究。

我们提出了额外的消融研究

BERT包括:

- 训练步骤数的影响;和
- 不同掩蔽过程的消融残酷的。

BERT 的其他详细信息

A.1 预训练任务的说明我们在下面提供了预训练任务的例子。

Masked LM 和 Masking Procedure由于假设未标记的句子是 my dog is hairy,并且在随机掩蔽过程中我们选择了第 4 个标记 (对应于毛茸茸),我们的掩蔽过程可以进一步说明

- 80% 的时间:用 [MASK] 标记替换单词,例如, my dog is hairy → my dog is [MASK]

- 10% 的时间:用随机词替换单词,例如, my dog is hairy → my 狗是苹果

- 10% 的时间:保持单词不变,例如, my dog is hairy → my dog is hairy。这样做的目的是使表示偏向实际观察到的词。

这个过程的优点是
变压器编码器不知道哪些字

它将被要求预测或被随机词替换,因此它被迫保留每个输入标记的分布式上下文表示。此外,由于随机替换仅发生在所有标记的 1.5% (即 15% 的 10%)中,这似乎不会损害模型的语言理解能力。在 C.2 节中,我们评估了此过程的影响。

与标准语言模型训练相比,masked LM 仅对每批中 15% 的标记进行预测,这表明模型可能需要更多的预训练步骤

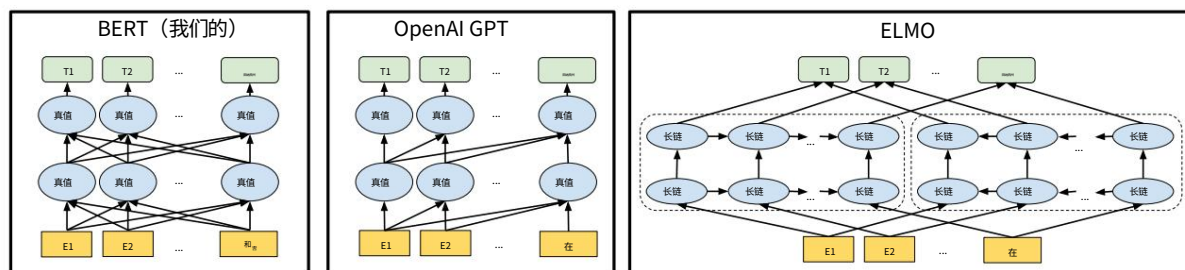


图 3: 预训练模型架构的差异。BERT 使用双向 Transformer。OpenAI GPT 使用从左到右的 Transformer。ELMo 使用独立训练的从左到右和从右到左的 LSTM 的串联来为下游任务生成特征。在这三者中,只有 BERT 表示在所有层中都以左右上下文为联合条件。除了架构差异之外,BERT 和 OpenAI GPT 是微调方法,而 ELMo 是基于特征的方法。

收敛。在 C.1 节中,我们证明了 MLM 的收敛速度确实比从左到右的模型 (预测每个标记) 慢一些,但 MLM 模型的经验改进远远超过增加的训练成本。

Next Sentence Prediction 下一句预测

预测任务可以在以下示例中说明。

输入 = [CLS] 这个人去了 [MASK] 商店 [SEP]

他买了一加仑 [MASK] 牛奶 [SEP]

标签 = IsNext

输入 = [CLS] 人 [MASK] 到商店 [SEP]

企鹅 [MASK] 是飞行 ##less 鸟 [SEP]

标签 = NotNext

A.2 预训练程序

为了生成每个训练输入序列,我们从语料库中抽取两个文本片段,我们将其称为“句子”,尽管它们通常比单个句子长得多 (但也可以更短)。第一句话接收 A embedding,第二句接收 B embedding。50% 的时间 B 是 A 之后的实际下一句,50% 的时间是随机的

句子,这是为“下一句预测”任务完成的。它们被采样,使得组合长度 ≤ 512 个令牌。LM masking 在 WordPiece tokenization 后应用,统一形式 masking rate 为 15%,不对部分 word pieces 给予特殊考虑。

我们使用 256 个序列 (256 * 512 个标记 = 128,000 个标记/批次) 的批大小训练 1,000,000 步,大约是 40

超过 33 亿个词的语料库。我们使用学习率为 $1e-4$ 、 $\beta_1 = 0.9$ 、 $\beta_2 = 0.999$ 、L2 权重衰减为 0.01、前 10,000 步的学习率预热以及学习率线性衰减的 Adam。我们在所有层上使用 0.1 的丢失概率。我们使用 gelu 激活 (Hendrycks 和 Gimpel, 2016) 而不是标准的 relu,遵循 OpenAI GPT。训练损失是平均掩蔽 LM 似然与平均下一句预测似然之和。

BERTBASE 的训练是在 Pod 配置的 4 个 Cloud TPU 上进行的 (总共 16 个 TPU 芯片)。13 BERTLARGE 的训练是在 16 个 Cloud TPU (总共 64 个 TPU 芯片) 上进行的。每个预训练需要 4 天才能完成。

较长的序列代价不成比例,因为注意力是序列长度的二次方。为了加快我们实验中的预训练,我们对 90% 的步骤使用序列长度为 128 的模型进行预训练。然后,我们训练 512 序列剩余的 10% 的步骤来学习位置嵌入。

A.3 Fine-tuning Procedure 对于微

调,除了 batch size、learning rate 和 training epochs 之外,大多数模型超参数与预训练相同。辍学概率始终保持在 0.1。最佳超参数值是特定于任务的,但我们发现以下可能值范围适用于所有任务:

· 批量大小: 16、32

13 <https://cloudplatform.googleblog.com/2018/06/云TPU-now-offers-preemptible-pricing-and-global-availability.html>

·学习率（亚当）： 5e-5、3e-5、2e-5 ·时期数： 2、3、4

我们还观察到,与小数据集相比,大数据集 (例如,100k+ 标记的训练示例)对超参数选择的敏感度要低得多。微调通常非常快,因此简单地对上述参数进行详尽搜索并选择在开发集上表现最佳的模型是合理的。

A.4 BERT、ELMo 和 OpenAI GPT 的比较

在这里,我们研究了最近流行的表示学习模型的差异,包括 ELMo、OpenAI GPT 和 BERT。图 3 直观地显示了模型架构之间的比较。请注意,除了架构差异之外,BERT 和 OpenAI GPT 是微调方法,而 ELMo 是基于特征的方法。

与 BERT 最具可比性的现有预训练方法是 OpenAI GPT,它在大型文本语料库上训练从左到右的 Transformer LM。事实上,BERT 中的许多设计决策都是有意做出的,以使其尽可能接近 GPT,以便可以将这两种方法进行最低限度的比较。这项工作的核心论点是,双向性和第3.1节中介绍的两个预训练任务占大部分实证改进,但我们确实注意到还有其他几个差异

BERT 和 GPT 的训练方式:

- GPT 在 BooksCorpus (8 亿字)上训练; BERT 在 BooksCorpus (8 亿字)和维基百科 (25 亿字)上接受训练。
- GPT 使用仅在微调时引入的句子分隔符 ([SEP])和分类器标记 ([CLS]); BERT 在预训练期间学习 [SEP]、[CLS] 和句子 A/B 嵌入。
- GPT 训练了 1M 步,批量大小为 32,000 个单词; BERT 接受了 1M 步的训练,批量大小为 128,000 个单词。
- GPT 对所有微调实验使用相同的学习率5e-5; BERT 选择在开发集上表现最好的特定于任务的微调学习率。

为了隔离这些差异的影响,我们在第5.1节中进行了消融实验,证明大部分改进实际上来自两个预训练任务及其启用的双向性。

A.5 Fine-tuning 对不同任务的说明

图 4 显示了针对不同任务微调 BERT 的图示。我们的任务特定模型是通过将 BERT 与一个额外的输出层结合在一起形成的,因此需要从头开始学习最少数量的参数。

在任务中,(a)和 (b)是序列级任务,而 (c)和 (d)是令牌级任务。图中,E表示输入嵌入, T_i 表示token i 的上下文表示,[CLS]是分类输出的特殊符号,[SEP]是分隔非连续token序列的特殊符号。

B 详细的实验设置

B.1 GLUE 基准实验的详细说明。

我们在表 1 中的 GLUE 结果来自<https://gluebenchmark.com/>从 [leaderboard](https://gluebenchmark.com/leaderboard)和<https://openai.com/language-unsupervised>。

GLUE 基准包括以下数据集,其描述最初在Wang 等人中总结。(2018a):

MNLI多流派自然语言推理是一项大规模的众包蕴含分类任务 (Williams 等人, 2018 年)。给定一对句子,目标是预测第二个句子相对于第一个句子是蕴含、矛盾还是中性。

QQP Quora 问题对是一项二元分类任务,其目标是确定在 Quora 上提出的两个问题在语义上是否等价 (Chen 等人, 2018 年)。

QNLI问题自然语言推理是斯坦福问答数据集(Rajpurkar et al., 2016)的一个版本,已转换为二元分类任务(Wang et al., 2018a)。正例是包含正确答案的 (问题,句子)对,反例是来自同一段落但不包含答案的 (问题,句子)对。

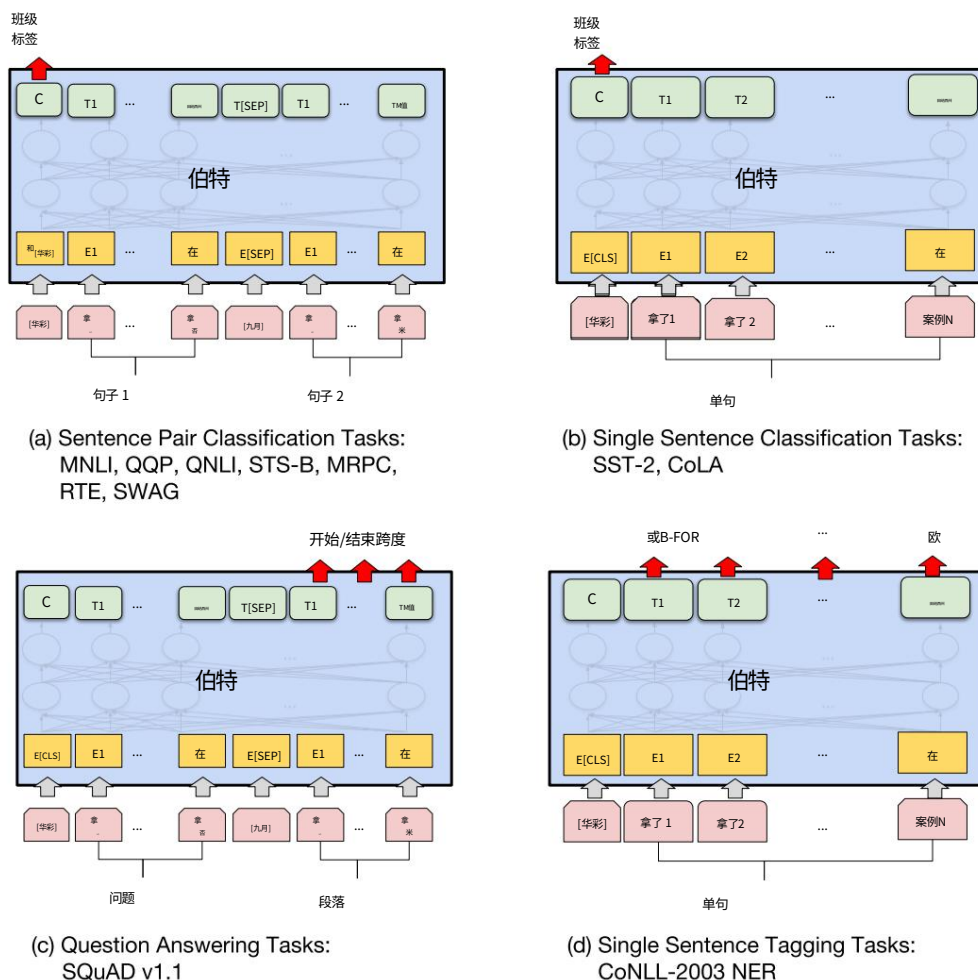


图 4:在不同任务上微调 BERT 的图示。

SST-2斯坦福情绪树库是一个二元单句分类任务,由从电影评论中提取的句子组成,并带有人类对其情绪的注释 (Socher 等人, 2013 年)。

CoLA The Corpus of Linguistic Acceptability 是一项二元单句分类任务,其目标是预测英语句子在语言上是否“可接受” (Warstadt 等人, 2018 年)。

STS-B语义文本相似性基准是从新闻标题和其他来源中提取的句子对的集合 (Cer 等人, 2017 年)。他们用 1 到 5 的分数进行注释,表示这两个句子在语义上的相似程度。

MRPC Microsoft Research Paraphrase Corpus 由自动从在线新闻源中提取的句子对组成,并带有人工注释

这对句子中的句子是否在语义上等价 (Dolan 和 Brockett, 2005)。

RTE识别文本蕴含是一种类似于 MNLI 的二元蕴含任务,但训练数据要少得多 (Bentivogli 等人, 2009 年)。¹⁴

WNLI Winograd NLI 是一个小型自然语言推理数据集 (Levesque et al., 2011)。

GLUE 网页指出,此数据集的构建存在问题,并且提交给 GLUE 的每个经过训练的系统的性能都低于预测多数类的 65 因基线精度。OpenAI GPT 公平,我们排除了这个集合。对于我们的 GLUE 提交,我们总是预测 ma

¹⁴请注意,我们在本文中仅报告单任务微调结果。多任务微调方法可能会进一步提高性能。例如,我们确实观察到使用 MNLI 进行的多任务训练对 RTE 有了实质性的改进。 [15https://gluebenchmark.com/faq](https://gluebenchmark.com/faq)

多数阶级。

C 额外的消融研究

C.1 训练步数的影响

图5显示了从经过 k 步预训练的检查点微调后的 MNLI Dev 准确性。这使我们能够回答以下问题：

1. 问题:BERT真的需要这么大的预训练量 (128,000 words/ batch * 1,000,000 steps)才能达到很高的微调精度吗？

回答:是的,与 500k 步相比,在 1M 步上训练时, BERTBASE 在 MNLI 上的准确率提高了近 1.0%。

2. 问题:MLM 预训练收敛速度是否比 LTR 预训练慢,因为每批中只有 15% 的词被预测,而不是每个词？

答:MLM 模型确实比 LTR 模型收敛稍慢。然而,就绝对准确性而言,MLM 模型几乎立即开始优于 LTR 模型。

C.2 不同掩蔽的消融程序

在3.1节中,我们提到 BERT 在使用掩码语言模型 (MLM) 目标进行预训练时使用混合策略来掩码目标标记。以下是评估不同掩蔽策略效果的消融研究。

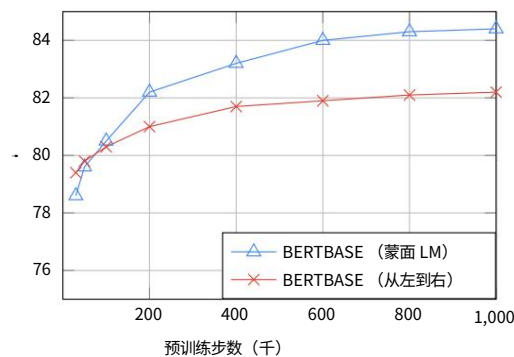


图 5:消融训练步骤数。这显示了微调后的 MNLI 精度,从已预训练 k 步的模型参数开始。x 轴是 k 的值。

请注意,掩蔽策略的目的是减少预训练和微调之间的不匹配,因为 [MASK] 符号在微调阶段永远不会出现。我们报告了 MNLI 和 NER 的开发结果。对于 NER,我们报告了微调 and 基于特征的方法,因为我们预计基于特征的方法的不匹配会被放大,因为模型将没有机会调整表示。

掩蔽率		开发集结果		
屏蔽相同的RND MNLI		向下		
		Fine-tune	Fine-tune 基于特征	
80%	10% 10% 84.2	100% 0% 0%	95.4	94.9
84.3	80% 0% 20% 84.1	80% 20% 0%	94.9	94.0
84.4	0% 20% 80% 83.7	0% 0%	95.2	94.6
100%	83.6		95.2	94.7
			94.8	94.6
			94.9	94.6

表 8:不同掩蔽策略的消融。

结果如表8 所示。表中, MASK表示我们将目标标记替换为 MLM 的 [MASK] 符号; SAME表示我们保持目标令牌不变; RND意味着我们用另一个随机令牌替换目标令牌。

表格左侧的数字代表了 MLM 预训练期间使用的特定策略的概率 (BERT 使用 80%、10%、10%)。论文的右侧部分代表开发集结果。对于基于特征的方法,我们将 BERT 的最后 4 层连接起来作为特征,这在 5.3 节中被证明是最好的方法。

从表中可以看出,微调对于不同的掩蔽策略具有惊人的鲁棒性。

然而,正如预期的那样,在将基于特征的方法应用于 NER 时,仅使用 MASK策略是有问题的。有趣的是,仅使用 RND策略的性能也比我们的策略差得多。