应用多准则学习实现快速、准确的神经汉语分词

黄伟鹏* **程兴义*†** **陈坤龙** **王泰峰** **魏楚**
蚂蚁集团
{weipeng.hwp，fanyin.cxy，kunlong.ckl，taifeng.wang，weichu.cw}@antgroup.com

摘要

模糊的注释标准导致汉语分词数据集在各种粒度上的差异。 多准则中文分词旨在捕捉数据集之间的各种注释标准，并利用它们共同的底层知识。 在本文中，我们提出了一种域自适应分割器来利用各种数据集的不同标准。 我们的模型是基于双向编码器表示从变压器(BERT)，负责引入开放域知识。 提出了私有和共享投影层，分别捕获特定领域的知识和公共知识。 我们还通过蒸馏、量化和编译器优化来优化计算效率。 实验表明，我们的分割器在10个CWS数据集上的性能优于以前的ART(SOTA)模型，效率更高。

1 导言

中文分词(CWS)通常被认为是一个低级的NLP任务。 与使用空间标记分隔单词的英语和法语不同，汉语是一种多合成语言，其中化合物是从本地语素发展而来的(Jernudd和Shapiro，2011年； 龚等人 艾尔，2017年)。 语素和复合词之间的模糊区分导致了词概念的认知差异。 因此，由于注释不一致，标记数据集严重发散，导致多粒度化合物。 如表所示1，给一个句子“刘国梁赢得世界冠军”（刘国良赢得世界冠军），两个常用的语料库，即PKU的人民日报(PKU)和Penn中国树银行(CTB)，使用不同的分割标准。

在实践中，分段器通常提供具有不同粒度的多个配置，以更好地服务于各种下游任务。 细粒度准则能够减少词汇量，从而缓解稀疏性问题。 另一方面，粗粒度单词提供了更具体的含义，这可能有利于特定领域的任务。

近年来，为了探索异构数据集的公共知识，提出了几种CWS的多准则学习方法。 通过利用所有语料库的信息，多准则学习方法可以提高词汇外(OOV)回忆和实际性能(邱等人，2013年；Chao等人，2015年；陈等人，2017年)。 尽管其有效性，但仍有三个未决问题。 （1）即使有多个数据集，数据仍仅限于提供足够的语言知识。 （2）从数据集学习可能会伤害其他数据集，因为分割遵循不一致的标准。 （3）先进模型，例如Bi-LSTM-CRF(Ma等人，2018年)，在计算上是昂贵的。 它们是基于递归神经网络(RNN)。 由于RNN是自回归的，计算不能并行完成，由于计算效率差，它们的应用通常受到限制。 事实上，CWS系统作为NLP管道的基本模块，对推理速度有很大的要求。 例如，搜索引擎通常只能在CWS中花费几十毫秒甚至毫秒。

* 同等贡献。
† 相应的作者。

标准	刘	国良	赢了	世界锦标赛	
CTB	刘国梁		赢得	世界冠军	
pku	刘	国梁	赢得	世界	冠军

表1：不同的分割标准。

为了缓解现有方法的局限性，我们提出了一种CWS的多准则方法。 最近的研究(杨等人，2017年； Ma等人，2018年； 王等人，2019年) 指出利用外部知识可以提高CWS的准确性。 在此基础上，我们采用了BERT(Vaswani等人。， 2017; Devlin等人，2018年)作为提取开放域知识的骨干。 在BERT的顶部，使用私有投影层和共享投影层分别捕获特定领域的知识和常见的底层知识。

为了使它更实际，采用了三种技术，即知识蒸馏、数值量化和编译器优化，以加速我们的分割。 Bertology分析(Clark等人。， 2019; Jawahar等人，2019年； 徐等人，2020年) 表明来自不同层次的BERT表示捕获特定的含义。 对于CWS任务，使用中间层的表示就足够了(参见部分4.5.2 供详细分析)。 为了更好地利用BERT，提出了知识蒸馏法(Hinton等人，2015年)被利用了。

同时，我们还采用量化和编译器优化技术来提高可伸缩性。 实验表明，该方法不仅在10个CWS数据集上显著优于已知的结果，而且具有更好的效率。

捐款可概述如下。

- 利用顶部有域投影层的BERT来捕获异构分割准则和共同的底层知识。 据我们所知，这是第一次利用预先训练的模型在CWS。
- 我们可视化BERT层和注意分数，以提供一个洞察语言信息在CWS。
- 为了提高分割速度，采用了蒸馏、量化和编译器优化等模型加速技术。
- 实验结果表明，该模型在10个CWS语料库中具有不同的分割准则，优于以往的结果。

2 模型描述

目前的神经CWS模型通常由三个部分组成：字符嵌入层、特征提取层和CRF标记推理层。 为了使我们的模型具有多准则学习的能力，我们在推理层之前插入了一个额外的域投影层，如图所示1. 在这一部分中，我们详细描述了所提出的模型体系结构和目标函数。

2.1 特征提取层

我们雇用BERT(Vaswani等人，2017年；Devlin等人，2018年)提取输入序列的特征。 误码率对分词任务具有重要意义。 如图所示1，首先将字符映射到嵌入向量中，然后输入到几个变压器块中。 与逐步处理序列的BiLSTM相比，变压器在所有时间步长上并行学习特征，从而加快解码速度。 然而，原始的12个变压器层的BERT仍然太重，无法应用于实际的分词应用。 为了加快微调和推理过程，我们进行了进一步的优化，如本节所讨论的3.

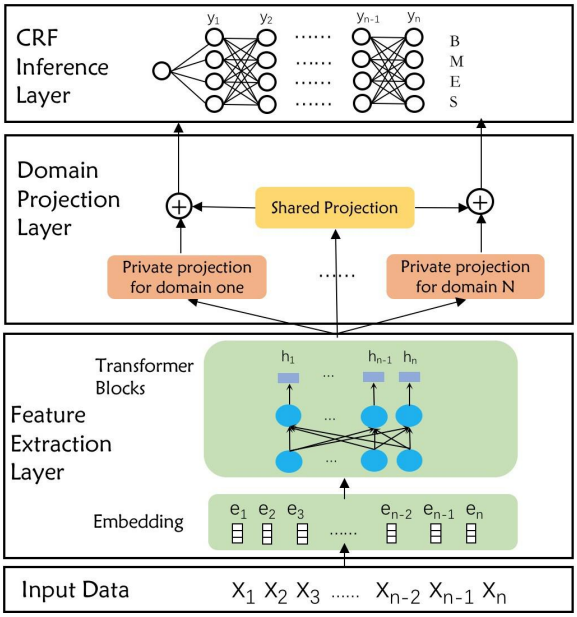


图1：所提出的模型的体系结构，层叠有特征提取层、域投影层和CRF标记推理层。

给定一个句子X=x_1，x_2，x_n 我们首先映射每个字符x_i 嵌入向量e_i。 然后将嵌入向量输入BERT，得到特征表示：

$$h_i = \text{BERT}(E_1 e_2, e_n; \theta), \tag{1}$$

其中 θ 表示BERT模型中的所有参数。

2.2 域投影层

如表所示1，根据不同的数据集标准，可以将同一个句子分割成不同的单词。 如果我们简单地组合数据集来训练单个模型，模型将被不同的标准所混淆，从而损害性能。 传统方法为每个数据集训练单个模型，这在实践中会产生巨大的部署成本。

受以前作品的启发(Chen等人，2017年；彭德泽，2017)，我们提出了一个域投影层，使我们的模型能够适应不同标准的数据集。 域投影层有助于捕获每个数据集的异构分割标准。 投影层有许多变化；在本文中，我们使用线性变换层，这对于这项任务是简单但有效的。 如图所示1，我们引入了一个额外的共享投影层来从数据集学习公共知识。 计算图不包括特定领域的投影，并保留共享投影，以分割标准。

形式上，用域投影层，我们可以得到特定于域的表示和共享表示：

$$h_{域} = W^{t}_{域} \quad h + b_{域}, \tag{2}$$

$$h_{共享} = W^{t}_{共享} \quad h + b_{共享}, \tag{3}$$

在哪里 $t_{域} \in r^{\times d_h}$, d_W $_{分享}$ $\in r^{d_h \times d_h}$ $b_{域} \in r^{d_h}$ 和 $b_{共享} \in r^{d_h}$ 是可以训练的参数。 d_h 是h的维数。

2.3 参照层

我们将CWS任务视为基于字符的序列标记问题。 句子X中的每个字符=x_1，x_2，x_n 标记为=B、M、E、S中的一个，表示单词的开始、中间、结尾和一个具有单一字符的单词。 如图所示1，特定于域的输出 投影

并将共享投影串联起来，然后馈送到一阶线性链条件随机场(CRF)层中(Lafferty等人，2001年)来推断这些标签。

形式上，标签序列形式化的概率为：

$$p(Y|X)=\frac{\psi\ /x)}{\sigma_{y^{J}\in^{n}\ \psi^{J}/x)'}} \quad (4)$$

其中Ψ(Y|X)是潜在函数：

$$\psi(y|x)=\overset{y^{\beta}}{\underset{我=2}{}}\psi(X，i，y_{我-1}，是的_{i})\ ， \quad (5)$$

$$\psi(X，i，y_{我-1}，是的_{i})=支出(X，i)_{yi}+b_{y我-1y_{I}})\ ， \quad (6)$$

其中y∈{B，M，E，S}是标记标签，b∈$R^{/我/\times/我/}$是可训练的参数和$b_{y我-1y_{i}}$意味着从标记y过渡$_{我-1}$敬你$_{i}$。 得分函数s(X，i)是投影层在i处的输出$_{th}$字符，它为标记i的每个标签分配分数$_{th}$人物：

$$S(X，i)=W^{t}\ _{s}[h_{域}h_{共享}]+b_{s}，$$

$$_{s}($$

7）在哪里$_{域}h_{共享}]\in$是域特定投影和共享投影的级联$^{t}\in$
$r^{2d_{h}\times/我/}$和b_{s} $r^{/我/}$是可训练的参数。

推断可以通过最大后验概率来实现：

$$y^{*}=\text{argmaxp}(Y|X)。 \quad (8)$$

2.4 目标功能

对网络的参数进行训练，使数据集上真实标签的条件对数似然最大化。 目标函数$J_{塞格}$计算为：

$$j_{塞格}(\theta)=\overset{\sigma}{j}\text{日志}p(Y^{(j)}/x^{(j)};\ \theta)，\quad (9)$$

其中Θ表示模型中的所有参数，$(X^{(j)}，是的^{(j)})$表示j_{th}数据集中的样本。 多准则学习的总损失是每个数据集中损失的组合。

3 模型加速

神经CWS模型通过增加模型复杂度来提高性能，但这损害了解码速度，限制了它们的实际应用。 为了弥补这一差距，我们应用了模型加速技术如下。

3.1 蒸馏

为了平衡计算成本和分割精度，我们提取了知识(Ba和Caruana，2014年； Hinton等人，2015年)从BERT到一个较小的变压器网络。 并对数据集进行监督微调。 最近的分析(Clark等人，2019年；贾瓦哈尔等人，2019年)表明BERT层提供短语级信息，中间层提取句法特征，顶层能够处理语义特征。 CWS本质上是一项句法集群任务，严重依赖于词汇和句法特征。 因此，我们转向使用底层到中间层作为骨干来学习注释，并共同提取BERT的顶层。 具体而言，具有12层的原始中文BERT作为教师，截断（3或6层）BERT作为学生向教师学习。 教师网络和学生网络在我们提出的模型的特征提取层中存在差异，如图所示1.

数据集	培训组	发展	准备好了	测试装置	平均字长
数控	5920公里	657k		727K	1.52
作为	4903k	546k		122k	1.51
Msr	2132k	235k		106k	1.68
城市	1309k	146k		40k	1.62
pku	994k	115k		104k	1.61
ctb6	641k	59k		81k	1.63
苏	476k	53k		113k	1.57
ud	100公里	11k		12k	1.49
ZX	79k	9k		34k	1.42
wtb	15k	1k		2k	1.53

表2：十个数据集的详细信息：训练集、开发集和测试集中的单词数、每个数据集的平均单词长度(char/word。

	pku	Msr	作为	城市	ctb6	苏	ud	数控	wtb	ZX
(Yang等人，2017年)	96.3	97.5	95.7	96.9	96.2	-	-	-	-	-
(Chen等人，2017年)	94.3	96.0	94.6	95.6	96.2	96.0	-	-	-	-
（徐和孙，2017年）	96.1	96.3	-	-	95.8	-	-	-	-	-
(Ma等人，2018年)	96.1	98.1	96.2	97.2	96.7	-	96.9	-	-	-
（龚等人，2018年）	96.2	97.8	95.2	96.2	97.3	97.2	-	-	-	-
（周等人，2019年）	96.2	97.0	96.9	97.1	95.2	-	-	-	-	-
（他，2019年）	96.0	97.2	95.4	96.1	96.7	96.4	94.4	97.0	90.4	95.7
我们的（学生-3层）	96.7	97.9	96.8	97.6	97.5	97.3	97.4	97.1	93.1	97.0
我们的(学生-3层+FP16)	96.6	98.0	96.6	97.5	97.4	97.3	97.3	97.1	92.7	96.8
我们的（学生-6层）	97.2	98.3	97.0	97.7	97.7	**97.5**	**97.8**	97.2	93.0	**97.1**
我们的(学生-6层+FP16)	97.0	98.2	96.8	**97.8**	97.7	97.4	97.7	97.1	93.1	96.8
我们的（教师-12层）	**97.3**	**98.5**	**97.0**	**97.8**	**97.8**	**97.5**	**97.8**	**97.3**	**93.2**	**97.1**
我们的(教师-12层+FP16)	97.2	98.3	96.9	**97.8**	97.7	97.3	97.7	97.2	93.1	96.9

表3：不同数据集的最新性能比较(F1-score，%)。

为了提取原始的BERT，我们通过最小化来自教师模型的归一化logits和来自学生模型的logits之间的平方损失添加一个logits回归目标。 精馏损失制定为：

$$\sigma_{迪斯j_s}(_t\theta,\theta)=\frac{1}{2t}\sum_{j=0}^{m}\sum_{我=0}^{n}\left(\frac{h_s^{(j,i)}}{//h_s^{(j,i)}//^2}-\frac{h_t^{(j,i)}}{//h_t^{(j,i)}//^2}\right)^{\sigma_2}, \tag{10}$$

哪里Θ_s， θ_t 表示学生网络和教师网络中的参数，M表示数据集中的样本数，N表示序列长度，h_s h_t 表示从学生网络中提取的逻辑 还有 老师 网络 分别。 期间 的 蒸馏 进程， 的 参数 θ_t 都是 冰冻的。

将分割损失和蒸馏损失相结合，总体损失为：

$$j(\theta_s, \theta_t)=j_{塞格}(\theta_s)+\alpha J_{迪斯}(\theta_s, \theta_t), \tag{11}$$

1）如果 α 是一个超参数来抵消这两个损失函数。

3.2 量化

还研究了网络加速的量化方法。 这些方法主要分为两类：标量量化和矢量量化（龚等人，2014年），定点量化(Gupta等人，2015年)。 传统的神经网络实现使用32位单精度浮点进行权值和激活，导致计算和模型存储资源的大量增加。 因此，我们进行定点量化，以利用NVIDIA的Volta体系结构特征。 提出了定点量化来缓解这些复杂性。

具体来说，在我们的模型中，半精度(FP16)应用于多头注意层和前馈层的内核，而诸如嵌入和归一化参数等其余参数使用全精度(FP32)。 微调过程中的梯度也使用完全精确。 量化方法不仅加快了计算速度，而且减小了模型尺寸。

教师-贝尔特（12层）	97.2	97.0	97.1
学生-转化器（6层）	97.1	97.0	97.0
学生-转化者（3层)）	96.8	96.9	96.8
学生-转化者（1层）	95.9	96.1	96.0

精确 召回 F1-Score

表4：教师网络和学生网络10个数据集的平均精度、召回率、F1分数。

	作为	城市	数控	ctb6	Msr	pku	苏	ud	wtb	ZX	所有数据集
作为	0.0	13.9	14.8	8.4	7.1	6.4	4.5	2.3	0.4	0.9	30.3
城市	33.5	0.0	31.4	14.0	20.9	17.5	9.8	4.0	0.8	2.4	50.5
数控	14.8	6.2	0.0	4.2	7.7	6.0	2.7	1.7	0.2	0.6	25.1
ctb6	47.3	31.2	40.5	0.0	28.0	24.6	15.4	7.1	1.2	3.1	63.9
Msr	18.4	10.5	27.3	8.0	0.0	10.7	5.9	2.3	0.1	1.3	36.0
pku	39.6	31.7	50.1	25.3	35.2	0.0	15.7	8.6	0.7	3.9	67.3
苏	49.8	39.7	56.2	29.3	41.8	36.6	0.0	10.9	1.7	4.9	73.2
ud	55.3	43.2	57.5	37.6	45.1	37.2	30.2	0.0	3.8	6.2	70.2
wtb	76.5	71.0	79.2	64.3	72.5	69.8	65.1	41.6	0.0	22.0	85.1
ZX	71.4	49.8	71.8	43.0	53.1	44.2	35.3	20.1	6.9	0.0	81.1

表5：每一行表示数据集中出现在其他数据集中的OOV词率。

3.3 编译器优化

加速线性代数(XLA)是一种特定领域的线性代数编译器，通过优化计算来加速传感器流模型。它提供了一种运行TensorFlow模型的替代模式：它将TensorFlow图编译成一系列专门为给定模型生成的计算内核。 由于这些内核是模型特有的，它们可以利用特定于模型的信息进行优化。 例如，加法、乘法和约简等操作可以融合到单个GPU内核中。

通过将XLA引入到我们的模型中，将图形编译成机器指令，并将低级操作融合起来，以提高执行速度。 例如，在变压器计算图中，批处理matmul总是后面跟着转置操作。 通过融合这两个操作，中间产品不需要回写入内存，从而减少了冗余内存访问时间和内核启动开销。

4 实验

4.1 实验设置

所有实验都是用Intel(R)Xeon(R)CPUE5-2682v4@2.50GHz和NVIDIA特斯拉V100在硬件上实现的。

数据集。 我们在10个标准的中文分词数据集上评估了我们的模型：MSR、PKU、AS、CITYU，来自SIGHAN2005烘焙任务（艾默生，2005年）。 来自SXU的2008年烘焙任务（教育部，2008年）。 中国宾夕法尼亚州立银行6.0(CTB6)从（薛等人，2005年）。 中国环球树行(UD)从Conll2017共享任务(Zeman和Popel，2017年)。 wtb(王和杨， 2014)，ZX（张和梅山，2014年）和数控语料库。 2005年和2008年的每一次

		pku	Msr	作为	城市	ctb6	苏	ud	数控	wtb	ZX	Avg
f1	单一标准	94.7	95.3	95.2	95.7	95.4	94.4	94.6	96.3	89.9	94.4	94.5
得分	多准则	**96.7**	**97.9**	**96.7**	**97.6**	**97.5**	**97.3**	**97.4**	**97.1**	**93.1**	**97.0**	**96.8**
奥天	单一标准	74.8	78.0	**78.3**	83.7	62.8	80.1	73.6	64.2	73.9	74.8	74.2
召回	多准则	**81.6**	**84.0**	77.3	**90.1**	**89.4**	**85.7**	**91.6**	**65.0**	**82.9**	**89.1**	**83.6**

表6：OOV回忆（%），F1评分（%）与多标准学习和单标准学习。 对于单准则和多准则学习，变压器层的数目都设置为3。

数据集，我们随机选择10%的训练数据作为开发集。 对于其他数据集，我们使用官方数据拆分。 表2 显示了十个数据集的详细信息。 我们可以注意到，这些数据集的平均字长（每字字符）从1.42到1.68不等，这反映了这些数据集的不同分割粒度和数据分布。

预处理。 分割前将AS和CITYU从繁体中文映射到简体中文。 数据集中的连续英文字符和数字分别替换为唯一的令牌。 全宽令牌被转换为半宽度，以处理训练集和测试集之间的不匹配。

超参数。 域投影层数为1，最大序列长度设置为128. 在微调过程中，我们使用Adam，学习速率为2e-5，L2重量衰减为0.01，辍学率为0.1。对于权衡的超参数α，我们尝试了几个值，并在接下来的实验中将其经验固定为0.15。 教师网络和学生网络特征提取层中的参数用预先训练的BERT进行初始化[1]，还有 全部 其他 参数 都是 已初始化 与更重的制服 初始化器。

评价指标。 汉语分词的目标是精确地切割输入句子变成单独的词。 因此，要达到精度的平衡($P=\frac{\#字_{黄金}\cap\#字_{赛斯}}{\#字_{赛斯}}$)和回忆($r=\frac{\#字_{黄金}\cap\#字_{赛斯}}{\#字_{黄金}}$我们使用F1评分($R=\frac{2pr}{p+r}$).

4.2 主要成果

我们用截断的BERT提取教师网络，与使用原始的12层BERT相比。 使用3层的10个数据集的平均F1得分从97.1%略微下降到96.8%，如表所示4. 我们认为，一个具有3个变压器层的学生网络是一个很好的选择，以平衡计算成本和分割精度。

我们的模型和最近的神经CWS模型的性能如表所示3. 我们的模型在10个数据集上的性能优于以往的工作，分别在PKU、MSR、AS、CITYU、CTB6、SXU、UD、CNC、WTB、ZX数据集上分别有13.5%、10.5%、15.8%、17.9%、14.8%、10.7%、32.2%、10.0%、29.1%、32.5%的误差减少。通过进一步应用半精度(FP16)，精度降低较小，模型在10个数据集上仍然优于以前的SOTA结果。 在应用编译器优化时，F1评分没有变化，因为它对预测结果没有影响。

4.3 域投影层的影响

以前的工作(黄和赵，2007年； 马等人，2018年)指出OOV是一个主要的错误，探索进一步的知识来源对于解决这个问题是必不可少的。 从某种角度来看，数据集是相互补充的，因为数据集中的OOV可能出现在其他数据集中。 我们做了一些分析，统计结果如表所示5. 以数据集为例，13.9%的OOV词出现在DataSetCITYU中，30.3%的OOV词出现在所有其他数据集中。

为了利用彼此的知识来提高OOV的召回率，我们的模型使用域投影层进行多准则学习。 为了评估这一点，我们分别在每个数据集上训练所提出的模型，即单准则学习。 在单标准学习设置中，共享投影层被排除在外，每个数据集只保留私有投影层。 对于单标准和多标准学习，学生变压器层的数量都设置为3。 表6 结果表明，与单标准学习相比，多标准学习显著提高了所有数据集的F1分数，平均提高了2.3。 它还提高了9/10数据集上的OOV召回率，平均提高了9.4。

4.4 卑鄙

解码速度在实践中是必不可少的，因为分词是许多下游NLP任务的基础。 以前的神经CWS模型(Ma等人，2018年；陈等人，2017年；龚等人，2018年；周等人 艾尔，2019年)使用Bi-LSTM，级联嵌入大小分别为100，100，128，100。 但是，

[1] https://github.com/google-research/bert

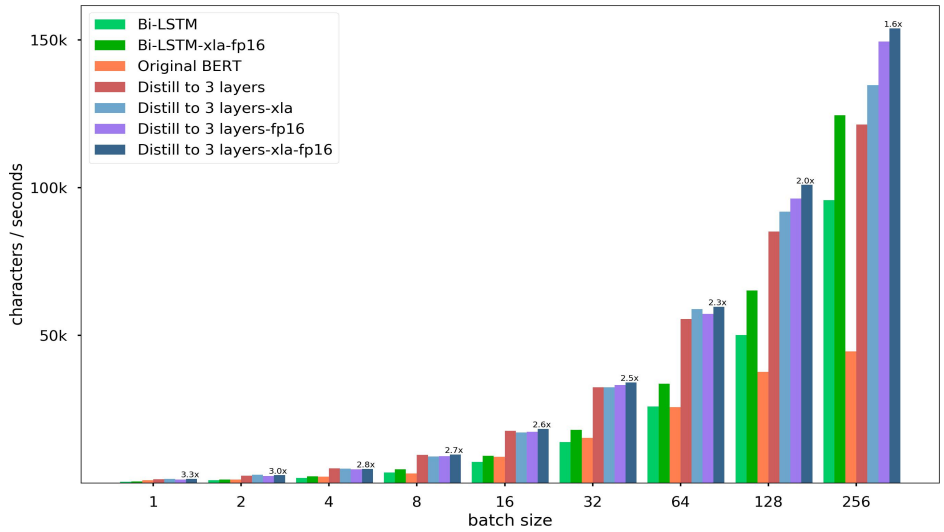


图2：解码速度w.r.t批处理大小。 序列长度为64。

他们没有报告解码速度。 为了进行公平的比较，我们将Bi-LSTM嵌入大小和隐藏大小设置为100，一个隐藏层，顶部有CRF。

图2 显示与批处理大小有关的解码速度。 我们的模型采用了12个变压器层的原始中文误码率，比Bi-LSTM慢。 然而，通过优化可以提高速度，包括蒸馏、权重量化和编译器优化。 结合所有这三种技术，我们的模型优于Bi-LSTM，具有1.6-3.3加速度，具有不同的批处理大小。我们试图用与BERT相同的计算优化来适应Bi-LSTM。 结果表明，Bi-LSTM通过优化实现了30%左右的加速度。 我们的方法仍然优于优化的Bi-LSTM，即。 Bi-LSTM-xla-fp16，加速度1.3-2.5。 此外，与Bi-LSTM相比，我们的模型更具可伸缩性，因为它们在处理涉及非常长序列的任务的能力上受到限制。 通过观察序列长度分布，我们可以搜索一个合适的层数来平衡F1分数和解码速度。

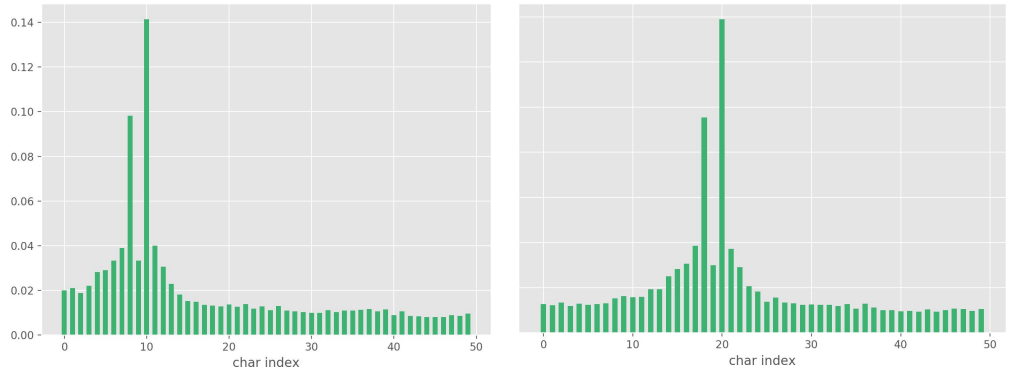
4.5 形象化

4.5.1 自我关注评分可视化

在变压器的自我注意层中，用其余字符计算每个字符的注意分数。 通过可视化的注意评分，我们可以直观的看到每个人物都注意到了什么。 具体来说，我们在所有十个数据集中选择长度大于50的句子，将这些句子输入模型，并在每个索引处平均注意分数。 注意分数如图所示3(a)(b)(c)。 我们可以注意到，当前字符周围的字符比遥远的字符获得更大的权重。 结果表明，分词更多地依赖于短语级信息，长期依赖相对不重要。 它直观地证明了不需要为CWS保留序列的长期记忆。

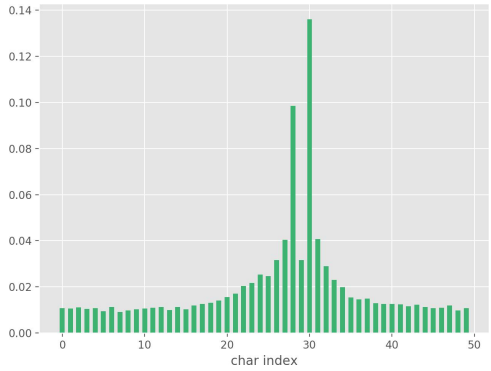
4.5.2 层关注评分可视化

在许多NLU任务中，BERT通过预先训练一堆12个变压器层来学习丰富的知识，取得了巨大的成功。 直觉上，顶层捕获高级语义特征，而底层学习低级特征，如语法。 对于中文分词任务，高级语义特征可能会产生较小的影响，因此我们进一步研究变压器层数的最小值。 我们在预先训练的BERT中冻结每个层的权重，并对分词数据集进行层注意微调。 如图所示3(d)，注意分数在顶层从7层逐渐下降到12层，第三层获得最高的注意分数。 结果表明，三层模型包含了大部分的分词信息。 最近的Bertol-

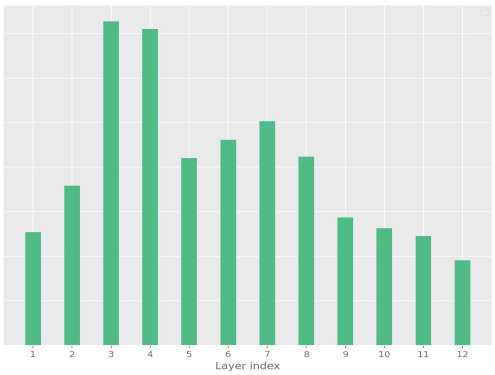


(a) 当前char指数10

(b)当前指标20



(c)当前指标30



(d)图层注意得分

图3：在某些char指数（10,20,30）和层注意分数下的自我关注分数分布。

奥吉(Jawahar等人，2019年；Liu等人，2019年；Tenney等人，2019年；休伊特和曼宁，2019年；戈德伯格， 2019）分析旨在从语言学的角度理解BERT的内在工作。 这些作品得出了类似的结论，即重庆的基本句法信息出现在下层或中间层，这与我们的分析是一致的。 BERT的较高层次是最具语义性或任务特异性的。 这些分析也反映了汉语分词在语言学中是一项句法较多但语义知识较少的任务。

5 相关工作

近年来，神经预警系统的多准则学习引起了学者们的广泛关注。 （邱等人，2013年） 采用基于堆栈的模型来利用来自多个源的注释数据。 (Chao等人。， 2015）利用多个语料库，利用耦合序列标记模型直接学习和推断两个异构注释。 （龚等人，2018年）提出了Switch-LSTM，通过利用跨多个异构标准的底层共享子标准来提高每个单一标准的性能。 (Chen等人，2017年)为CWS提出了一个多准则学习框架。 他们提出了三个共享-私有模型来集成多个分割标准。 一种对抗性策略被用来迫使共享层学习标准不变特征。 所有这些工作都利用了异构注释数据，并表明它们确实可以相互帮助。

在实际应用中，深度网络中的模型压缩和加速是至关重要的，这使得在移动、嵌入式和物联网设备上部署深度模型成为可能。 参数修剪、低秩分解、量化和知识蒸馏等技术在视觉任务中得到了广泛的应用(辛顿 等人，2015年；Ba和Caruana，2014年；Gupta等人，2015年；龚等人，2014年)。 然而，模型压缩和加速在NLP任务中很少被研究，特别是神经CWS任务。 据我们所知，我们首先压缩神经CWS模型以加快分割速度，通过三种模型加速技术，知识蒸馏，量化和编译器优化。 我们强调分割速度在工业应用中是非常重要的，例如搜索引擎。

6 结论

在本文中，我们提出了一种有效的中文分词方法，它使用BERT，并在顶部添加一个多准则学习的域投影层。 它们都有助于捕获异构分割标准和共同的底层知识。 我们将注意力评分可视化，以说明CWS中的语言。 为了提高分词的实用性，应用加速技术来提高分词速度。 由知识蒸馏，量化和编译器优化组成。 实验表明，与最先进的方法相比，该模型在分词精度和预测速度方面具有更高的性能。

参考资料

吉米·巴和里奇·卡鲁阿纳。 2014. 深网真的需要深网吗？ 神经信息处理系统的进展，第2654-2662页。

赵佳元，李正华，陈文良，张敏。 2015. 利用异构注释进行微博分词和pos标记。 在自然语言处理和中文计算，第495-506页。 斯普林格。

陈X.、石中、邱X.、黄X。 2017. 汉语分词的对抗性多准则学习。 计算语言学协会第五十五届年会论文集，第1卷，第1193-1203页。

凯文·克拉克，UrvashiKhandelwal，OmerLevy和克里斯托弗·D·曼宁。 2019. 伯特看什么？ 伯特注意力的分析。 *AR XIV预印Ar XIV：1906.04341。*

雅各布·德夫林，张明伟，肯特·李，克里斯蒂娜·图塔诺娃。 2018. 伯特：深度双向变压器的预训练，用于语言理解。 *AR XIV预印AR XIV：1810.04805。*

托马斯·爱默生。 2005. 第二届国际汉语分词bakoff。 《第四届新加坡汉语加工研讨会论文集。

Yoav Goldberg。 2019. 评估伯特的句法能力。 *AR XIV预印Ar XIV：1901.05287。*

龚云超，刘柳，明阳，卢博米尔·布尔德夫。 2014. 利用矢量量化压缩深卷积网络。 *阿希夫预印阿希夫：1412.6115。*

陈工，李正华，张敏，蒋新洲。 2017.多粒度中文分词。 在
2017年自然语言处理经验方法会议记录，第692-703页。

龚晶晶，陈新池，陶贵，邱锡鹏。 2018. 用于多准则中文分词的switch-lstms。 *AR XIV预印AR XIV：1812.08033。*

Suyog Gupta、Ankur Agrawal、Kailash Gopalakrishnan和Pritish Narayanan。 2015. 数值精度有限的深度学习。 在国际机器学习会议上，第1737-1746页。

汉和。 2019. 多准则分词的有效神经解。 第133-142页。 智能计算与应用。 斯普林格。

约翰·休伊特和克里斯托弗·D·曼宁。 2019. 在单词表示中查找语法的结构探针。 计算语言学协会北美分会2019年会议记录：人类语言技术，第1卷（长和短论文），第4129-4138页。

杰弗里·辛顿，奥利奥·维尼亚斯和杰夫·迪安。 2015. 在神经网络中蒸馏知识。 *阿西夫预印阿西夫：1503.02531。*

长宁黄和海昭。 2007. 中文分词：十年回顾。 *中文信息处理杂志，21（3）：8-20。*

Ganesh Jawahar，Benoˆıt Sagot，Djame´Seddah，Samuel Unicomb，Gerardo In˜iguez，Ma´rton Karsai，Yannick Le´o，Ma´rton Karsai，Carlos Sarraute，E´ 里奇·弗莱里等人。 2019. 关于语言的结构，伯特学到了什么？ *计算语言学协会第五十七届年会，意大利佛罗伦萨。*

Bjo¨Rn H Jernudd和Michael J Shapiro。 2011. *语言纯粹主义的政治，第54卷。* 沃尔特·德·格鲁伊特。

拉弗蒂，麦克卡勒姆，和FCN佩雷拉。 2001. 条件随机字段：分割和标记序列数据的概率模型。

纳尔逊·F·刘，马特·加德纳，约纳坦·贝林科夫，马修·彼得斯和诺亚·A·史密斯。 2019. 语境表征的语言知识和可转移性。 *阿希夫预印阿希夫：1903.08855。*

季马，库兹曼·甘切夫和大卫·魏斯。 2018. 最先进的中文分词双lstms。 在 *2018年自然语言处理经验方法会议记录。*

中华人民共和国。 2008. 第四个国际汉语处理bakoff：中文分词，命名实体识别和中文pos标记。 第六届中国语言加工研讨会论文集。

N Peng和M Dredze。 2017. 用于序列标注的多任务域适配。 第2期NLP表征学习研讨会论文集，第91-100页。

邱锡鹏，赵嘉义，黄宣靖。 2013. 多任务学习的异构注释语料库上的中文分词和pos标记。 《2013年自然语言处理经验方法会议记录》，第658-668页。

伊恩·特尼，迪潘扬·达斯，还有艾莉·帕弗利克。 2019. 伯特重新发现了经典的nlp管道。 *AR XIV预印Ar XIV：1905.05950。*

Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser和Illia Polosukhin。 2017. 你只需要注意。 神经信息处理系统的进展，第5998-6008页。

王和杨威廉。 2014. 微博的依赖解析：一种高效的概率逻辑编程方法。 *在2014年自然语言处理经验方法会议记录(EMNLP)中，第1152-1158页。*

王晓斌，邓彩，李林林，徐光伟，赵海，罗思。 2019. 无监督学习有助于有监督的神经分词。

徐杰和孙旭。 2017. 基于依赖门控递归神经网络的中文分词。 计算语言学协会第54届年会，第1193-1203页。

徐伟迪，程兴义，陈坤龙，王泰峰。 2020. 基于对称正则化的Bert用于两两语义推理。 第43届国际ACM SIGIR信息检索研究与发展会议记录，1901-1904页。

薛乃文，费霞，傅东秋，马尔塔·帕尔默。 2005. 宾州中文树库：大型语料库的短语结构注释。 *自然语言工程，11（02）：207s ˇC238。*

杨俊阳，张永章，董芳。 2017. 神经分词具有丰富的预训练。 计算语言学协会第五十五届年会论文集，第839-849页。

泽曼和马丁·波佩尔。 2017. Conll2017共享任务：从原始文本到通用依赖关系的多语言解析。 在CoNLL2017共享任务：多语言解析从原始文本到通用依赖，第1-19页。 计算语言学协会。

张和梅山。 2014. 用于联合分割和pos标记的类型监督域适配。 计算语言学协会欧洲分会第十四届会议论文集，第588-597页。

周嘉宁，王景康，刘功申。 2019. 多字符嵌入用于中文分词。 计算语言学协会第五十七届会议论文集：学生研究讲习班，第210-216页。