**MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification**

会议：ACL2020

代码：<https://github.com/GT-SALT/MixText>

**Abstract**

本文介绍了MixText，这是一种用于文本分类的半监督学习方法，它使用了我们新设计的数据增强方法TMix。TMix通过在隐藏空间中插入文本来创建大量的增强训练样本。此外，我们利用数据增强的最新进展来猜测未标记数据的低熵标记，从而使它们像标记数据一样易于使用。 通过混合标记的，未标记的和扩充的数据，MixText在多个文本分类基准上明显优于当前的预训练和微调模型以及其他最新的半监督学习方法。当监督非常有限时，改进尤其显着。

**1 Introduction**

在深度学习时代，研究在大多数受监督的学习环境中均取得了出色的成绩（LeCun等，2015； Yang等，2016）。 但是，当只有有限的标记数据时，监督式深度学习模型通常会遭受过度拟合的影响（Xie等人，2019）。 由于需要大量的时间，金钱和专业知识来获取足够的标记数据，因此对标记数据的强烈依赖很大程度上阻止了将神经网络模型应用于新的设置或现实情况。 结果，半监督学习受到了广泛的关注，以将标记和未标记的数据用于不同的学习任务，因为未标记的数据总是更容易收集且更便宜（Chawla和Karakoulas，2011）。

这项工作仔细研究了半监督文本分类，这是语言技术社区中最基本的任务之一。 对半监督文本分类的先前研究可以分为以下几类：（1）利用变分自动编码器（VAE）重构句子并使用从重构中学到的潜在变量来预测句子标签，例如（Chen等人，2018; Yang等人，2017; Gururangan等人 。，2019）; （2）鼓励模型对未标记的数据输出自信的预测以进行自我训练，例如（Lee，2013; Grandvalet and Bengio，2004; Meng et al。，2018）; （3）在添加对抗性噪声（Miyato等，2019,2017）或数据增强（Xie等，2019）之后进行一致性训练; （4）使用未标记的数据进行大规模的预训练，然后使用标记的数据进行微调（Devlin等，2019）。 尽管这些模型取得了巨大的成功，但大多数先前的工作分别使用了标记和未标记的数据，从而没有监督可以从标记到未标记的数据或从未标记到标记的数据过渡。结果，尽管没有标签的数据非常丰富，但大多数半监督的模型仍很容易地对非常有限的有标签数据进行过拟合。

为了克服局限性，在这项工作中，我们引入了一种新的数据增强方法TMix（第3节），这受Mixup最近在图像分类方面的成功（Gururangan等人，2019; Berthelot等人，2019）的启发。 如图1所示，TMix接受两个文本实例，并将它们插值到它们相应的隐藏空间中。由于组合是连续的，因此TMix有潜力创建无限数量的新扩充数据样本，从而可以极大地避免过度拟合。 然后，基于TMix，我们引入了一种新的文本分类的半监督学习方法，称为MixText（第4节），以显式地对标记和未标记样本之间的关系进行建模，从而克服了上述先前的半监督模型的局限性。

简而言之，MixText首先猜测未标签数据的低熵标签，然后使用TMix内插标签和未标签数据。 通过鼓励模型在训练示例之间线性表现，MixText可以促进挖掘句子之间的隐式关系，并在学习标记句子的同时利用来自未标记句子的信息。 同时，MixText利用几种半监督学习技术进一步利用未标记的数据，包括自我目标预测（Laine和Aila，2016），熵最小化（Grandvalet和Bengio，2004）和一致性正则化（Berthelot等人，2019; Xie等人） 等（2019）。

为了证明我们方法的有效性，我们在四个基准文本分类数据集上进行了实验（第5节），并将我们的方法与以前的最新半监督方法进行了比较，包括那些基于大量预训练模型的方法。 就测试集的准确性而言，未标记数据的数量。 我们进一步进行了消融研究，以证明每个组件对模型最终性能的影响。结果表明，我们的MixText方法明显优于基线，尤其是在给定的标记训练数据非常有限的情况下。

**2 Related Work**

**2.1 Pre-training and Fine-tuning Framework**

近年来，预训练和微调框架在NLP应用上取得了巨大成功，并已应用于各种NLP任务（Radford等人，2018; Chen等人，2019; Akbik等人， 2019）。 霍华德和鲁德（Howard and Ruder，2018）建议在大型通用语料库上预先训练语言模型，并使用判别式微调，倾斜的三角学习率和逐渐解冻等一些新颖的技术对目标任务进行微调。 以这种方式，即使使用少量标记数据，此类预训练模型也显示出出色的性能。 培训前的方法通常设计为具有不同的目标，例如语言建模（Peters等，2018; Howard和Ruder，2018; Yang等，2019b）和掩盖语言建模（Devlin等，2019; Lample和Conneau） ，2019）。 通过在更多数据上训练更大的模型，它们的性能也得到了改善（Yang等，2019b; Liu等，2019）。

**2.2文本数据的半监督学习**

在NLP社区中，半监督学习受到了很多关注（Gururangan等，2019; Clark等，2018; Yang等，2015），因为与标记数据相比，无标记数据通常很多。 例如，Gururangan等（2019）; Chen等（2018）; 杨等（2017）利用变式自动编码器（VAE），以文本分类和顺序标记的序列到序列建模形式。 Miyato等。 （2017）通过对单词嵌入应用扰动，将对抗和虚拟对抗训练用于文本域。 杨等。 （2019a）利用层次结构来利用从较高级别标签到较低级别标签的监督。 谢等UDA（2019）利用逆向翻译和TF-IDF单词替换后对未标记数据的一致性正则化。 克拉克等。 （2018）提出了针对未标记数据的交叉训练，其中他们使用了辅助预测模块，该模块可以看到输入的受限视图（例如，仅句子的一部分），并且可以匹配完整模型的预测以查看整个输入。

**2.3基于插值的正则器**

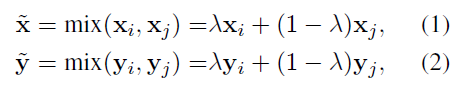
最近提出了基于插值的正则化器（例如Mixup）用于监督学习（Zhang等人，2017; Verma等人，2019a）和半监督学习（Berthelot等人，2019; Verma等人，2019b）用于 通过叠加两个输入图像并将图像标签合并为虚拟训练数据来获得图像格式的数据，并在诸如图像分类和网络体系结构之类的各种任务中实现了最先进的性能。 混合方法的不同变体被设计出来，例如在输入空间中执行插值（Zhang等人，2017），结合插值和截止值（Yun等人，2019），以及在隐藏空间表示中进行插值（Verma等人，2019a，c）。 但是，由于在文本中大多数输入空间是离散的，即，one-hot向量而不是图像中的连续RGB值，并且文本通常在结构上更复杂，因此在NLP领域中尚未研究这种内插技术。

**2.4文本的数据扩充**

当标记的数据受到限制时，数据扩充已成为增加训练数据量的有用技术。 例如，在计算机视觉中，图像会通过色调进行移位，放大/缩小，旋转，翻转，变形或着色（Perez和Wang，2017），以训练数据增强。 但是，由于其复杂的句法和语义结构，增加文本数据相对具有挑战性。最近，Wei和Zou（2019）利用同义词替换，随机插入，随机交换和随机删除来增强文本数据。 同样，Kumar等。 Xie等人（2019）根据单调亚模函数最大化提出了一种新的措辞表述，以获得高度多样化的措辞。（2019）和Chen等。（2020年）应用反向翻译（Sennrich等人，2015年）和单词替换以在未标记数据上生成复述，以进行一致性训练。其他研究噪声的方法及其将噪声纳入半监督命名实体分类的研究（Lakshmi Narayan等人，2019; Nagesh和Surdeanu，2018）。

**3 TMix**

在本节中，我们将Mixup（一种最初由（Zhang等，2017）提出的用于图像的数据增强方法）扩展到文本建模。 Mixup的主要思想非常简单：给定两个标记的数据点（xi，yi）和（xj， yj），其中x可以是图像，y是标签的one-hot表示，该算法创建虚拟训练样本 通过线性插值：



其中。 新的虚拟训练样本用于训练神经网络模型。Mixup可以用不同的方式解释。一方面，可以将Mixup视为一种数据增强方法，该方法可基于原始训练集创建新的数据样本。 另一方面，它在模型上实施正则化，以在训练数据之间线性表现。 事实证明，混合可以在连续图像数据上很好地工作（Zhang等，2017）。 但是，将其扩展为文本似乎具有挑战性，因为计算离散标记的插值是不可行的。

为此，我们提出了一种克服这一挑战的新颖方法-在文本隐藏空间中进行插值。 给定一个句子，我们经常使用诸如BERT（Devlin等人，2019）的多层模型对句子进行编码以获得语义表示，并以此为基础做出最终预测。先前的一些工作（Bowman等人，2016）表明，对两个隐藏向量进行插值解码会生成一个具有两个原始句子混合含义的新句子。 因此，我们建议在隐藏空间内应用插值作为文本的数据增强方法。 对于具有L层的编码器，我们选择将第m层的隐藏表示混合起来。

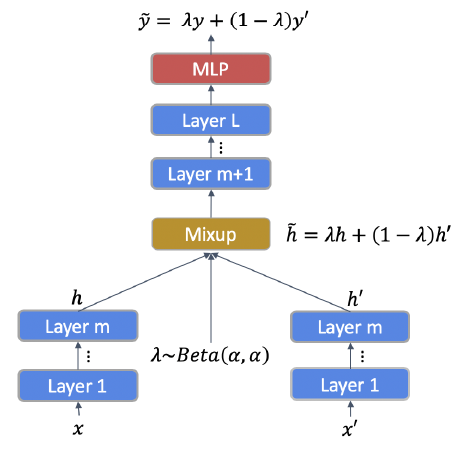
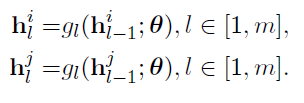
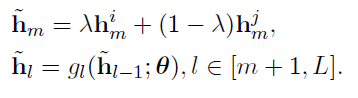


图1：TMix接收带有标签y和y’的两个文本样本x和x’，将它们在第m层的隐藏状态h和h’与权重混合为〜h，然后继续向前传递以预测混合的标签〜y。

如图1所示，我们首先在底层分别计算两个文本样本的隐藏表示。 然后，我们将第m层的隐藏表示混合在一起，并将插值后的隐藏表示提供给上层。 在数学上，将编码器网络中的第l层表示为，因此可以将第l层的隐藏表示计算为：对于两个文本样本xi 和xj，将第0层定义为嵌入层，即，则较低层的两个样本的隐藏表示为：



在第m层的混合和继续向前传递到高层的定义为：

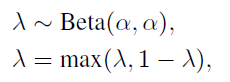


我们调用上述方法TMix并将新的混合操作定义为获得〜h L的整个过程：



通过使用编码器模型，TMix内插文本语义隐藏表示作为一种数据扩充。与在公式1的数据空间中定义的Mixup相反，TMix取决于编码器功能，因此为计算插值定义了更为广泛的范围。为了便于表示，在以下各节中，我们删除了对g的显式依赖，并将其简单表示为。

在我们的实验中，我们针对每个批次从Beta分布中采样混合参数以执行插值：



其中是用于控制的分布的超参数。 在TMix中，我们以与公式2相同的方式混合标签，然后使用对作为下游应用程序的输入。

不是像Verma（2019a）等在随机输入层执行混合，选择要混合的隐藏表示的哪一层是一个有趣的问题。 在我们的实验中，我们使用12层BERT（Devlin等人，2019）作为我们的编码器模型。 最近的工作（Jawahar et al。，2019）研究了BERT在不同层次上学到的东西。 具体来说，作者发现{3、4、5、6、7、9、12}层在BERT中具有最大的表示能力，并且每一层都捕获从文本的表面，句法到语义级别表示的不同类型的信息。例如，第9层在语义任务中具有预测能力，例如检查协同子句的随机交换，而第3层在表面任务（例如预测句子长度）方面表现最佳。

基于这些发现，我们选择既包含句法信息又包含语义信息的层作为我们的混合层，即M = {7; 9; 12}对于每一批，我们从集M计算插值的过程中随机抽取m层（用于混合表示的层）。 我们还在5.5节中进行了消融研究，以显示TMix的性能如何随着混合层设置的不同选择而变化。

文本分类 请注意，TMix提供了一种扩充文本数据的通用方法，因此可以应用于任何下游任务。 在本文中，我们将重点放在文本分类上，并将其他应用程序留作潜在的未来工作。 在文本分类中，我们将混合标签和分类器之间的KL散度作为监督损失的可能性最小化：



其中是编码器模型之上的分类器。 在我们的实验中，我们将分类器实现为两层MLP，它将混合表示作为输入并返回概率向量。 我们共同优化编码器参数和分类器参数，以训练整个模型。

**4 Semi-supervised MixText**

在本节中，我们演示如何利用TMix来帮助半监督学习。 给定有限的标记文本集，其标签为和一个大的未标记集，其中n和m是每组数据点的数量。是一个热点向量，C是类数。我们的目标是学习一种能够有效利用标记数据和未标记数据的分类器。

我们提出了一个名为MixText的新文本半监督学习框架。我们框架的核心思想是在标记和未标记的数据上同时利用TMix进行半监督学习。为了实现此目标，我们提出了一种标签猜测方法，以在训练过程中为未标记的数据生成标签。 使用猜测的标签，我们可以将未标签的数据视为其他标签数据，并执行TMix进行训练。 此外，我们将TMix与其他数据增强技术结合在一起以生成大量的增强数据，这是使我们的算法在极其有限的监督下可以很好地设置的关键组件。 最后，我们引入了一个熵最小化损失，鼓励模型在未标记的数据样本上分配尖锐的概率，当类C的数量很大时，这进一步有助于提高性能。 总体架构如图2所示。我们将详细解释每个组件。

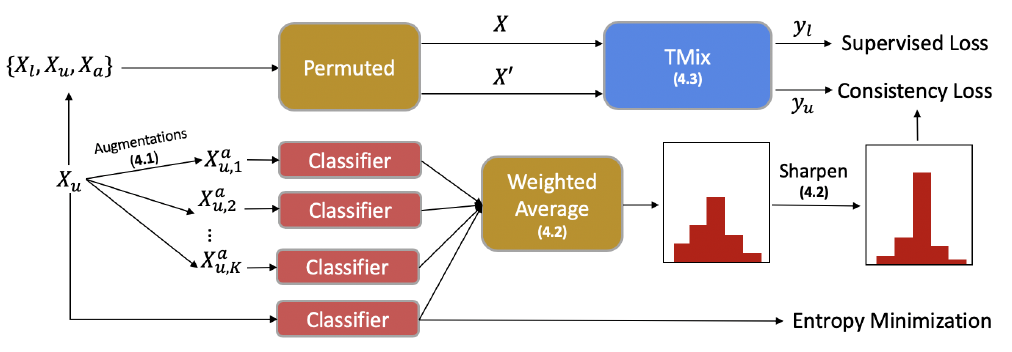


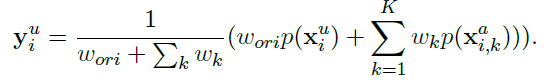
图2：MixText的总体架构。 MixText接收带标签的数据和未带标签的数据，进行扩充并预测未带标签的数据的标签，对带标签和无标签的数据执行TMix，并计算监督损失，一致性损失和熵最小化项。

**4.1 Data Augmentation**

反向翻译（Edunov et al。，2018）是一种常见的数据增强技术，可以在保留原始句子语义的同时生成多种释义。 我们利用反向翻译来解释未标记的数据。 对于每个在未标记的文本集Xu中，我们生成K个扩充; 通过使用不同中间语言的反向翻译来完成。例如，我们可以将原始句子从英语翻译为德语，然后再将其翻译回以获得释义。 在增强文本生成中，我们采用具有可调温度的随机采样来代替波束搜索以确保多样性。然后，这些扩充用于为未标记的数据生成标签，我们将在下面进行描述。

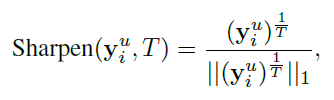
**4.2 Label Guessing**

对于未标记的数据样本及其K扩充; k，我们使用当前模型的预测结果的加权平均值为其生成标签：



注意，是概率向量。 我们希望模型能够预测不同扩增的一致标签。 因此，为了强制执行约束，我们将所有预测的加权平均值而不是任何单个数据样本的预测用作生成的标签。 此外，通过显式引入权重wori和wk，我们可以控制不同质量的扩增对生成标签的贡献。 我们的标签猜测方法改进（Tarvainen和Valpola，2017），该方法利用教师和学生模型来预测未标记数据的标签，而UDA（Xie等人，2019）仅使用作为生成的标签。

为避免加权平均值过于均匀，我们在预测的标签上使用了锐化功能。 给定温度超参数T：



其中是向量的l1范数。 当时，生成的标签成为one-hot。

**4.3 TMix on Labeled and Unlabeled Data**

在获得未标记数据的标签后，我们将标记文本，未标记文本Xu和未标记扩充文本合并在一起，形成超集。 相应的标签为，其中，我们定义，即所有增强样本共享与原始未标记样本相同的生成标记。

在训练中，我们随机采样两个数据点，然后我们计算，并使用KL散度作为损失：



由于x; x’是从X随机采样的，我们从许多不同的类别插值文本：标记数据之间的混合，标记和未标记数据之间的混合以及未标记数据之间的混合。 根据样本的类别，损失可以分为两种类型：

Supervised loss 当时，我们实际使用的多数信息来自标记的数据，因此在监督损失下训练模型。

Consistency loss 当样本来自未标记或扩充集时，即，大多数信息来自未标记数据时，KL散度是一致性损失的一种类型，将扩充样本约束为具有与原始数据样本相同的标记。

**4.4 Entropy Minimization**

为了鼓励模型对未标记的数据产生可信的标签，我们建议最小化对未标记的数据的预测概率的熵作为自训练损失：



为边距超参数。 如果概率向量大于，则我们将其最小化。即只考虑小于的数据，使其最小，使尽量大，但不大于， 对预测标签的正则化。限制标签置信度。

结合两个损失，我们得到MixText的总体目标函数：

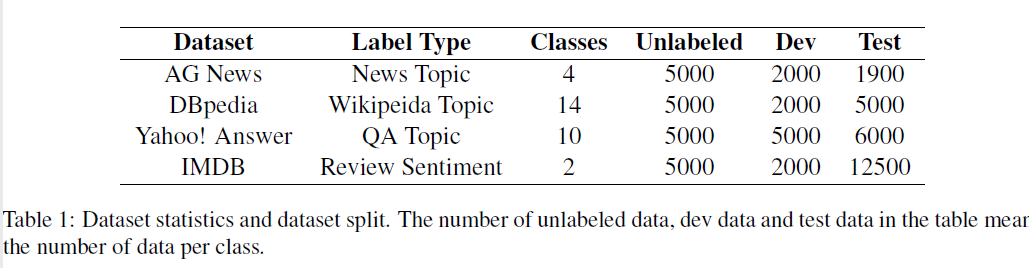


由于抑制有标签数据的权重，使模型不会偏向有标记，考虑置信度高的标签是否可用

**5实验**

**5.1数据集和预处理**

我们使用四个英语文本分类基准数据集进行了实验：AG News（Zhang等人，2015），BPpedia（Mendes等人，2012），Yahoo！ 答案（Chang等，2008）和IMDB（Maas等，2011）。 我们使用原始测试集作为测试集，并从训练集中随机采样以形成训练无标签集和开发集。 数据集统计信息和拆分信息如表1所示。



对于未标记的数据，我们选择了德语和俄语作为使用FairSeq（https://github.com/pytorch/fairseq）进行反向翻译的中间语言，并且随机采样温度为0.9。 这是一个例子，来自AG News数据集的一条新闻：“由于对冬季燃料供应紧缩的担忧和世界第二大用户中国经济的强劲增长，油价周五上涨至每桶55美元以上的纪录高位，” 通过德文和俄文的补充材料是：“由于担心冬季滑坡和世界第二大经济体增长强劲的担忧，油价周五飙升至每桶55美元以上的创纪录高位”和“油价飙升至历史高点 由于人们越来越担心冬季美国石油库存的减少以及世界第二大石油消费国中国的强劲经济增长，上周五油价升至每桶55美元。”

**5.2 Baselines**

为了测试我们方法的有效性，我们将其与几种最新模型进行了比较：

VAMPIRE（Gururangan et al。，2019）：在资源受限环境中进行预训练的各种方法（VAMPIRE）在域内未标记的数据上将unigram文档模型作为变体自动编码器进行了预训练，并将其内部状态用作下游分类器中的特征。

BERT（Devlin et al。，2019）：我们使用了预训练的基于BERT的无大小写模型3并对其进行了微调以进行分类。 详细而言，我们对BERT编码器的输出和MixText中使用的相同的两层MLP进行平均池化以预测标签。

UDA（Xie et al。，2019）：由于我们无权使用TPU，并且需要使用少量的未标记数据，因此我们自己使用pytorch实施了无监督数据增强（UDA）。 具体来说，我们使用与基于MixText相同的基于BERT的无大小写模型，未标记的增强数据和批处理大小，使用原始的未标记数据来预测具有与MixText相同的softmax锐化温度的标签，并计算增强的未标记数据之间的一致性损失。

**5.3 Model Settings**

我们使用基于BERT的无大小写标记器对文本进行标记化，使用基于BERT的无大小写模型作为文本编码器，并在编码器的输出上使用了平均池，具有128个隐藏大小和tanh的两层MLP作为激活 预测标签的功能。 句子的最大长度设置为256。对于超出限制的句子，我们保留前256个标记。 BERT编码器的学习速率为1e-5，MLP的学习速率为1e-3。 对于beta分布中的，通常，当每个类别的标记数据少于100个时，将其设置为2或16，较大的数据更有可能生成0.5左右的，从而创建“较新的”数据作为数据增强； 当每个类别的标记数据大于200时，将设置为0.2或0.4，因为较小的数据更有可能生成0.1左右，因此会在添加噪声正则化时创建“相似”数据。

对于TMix，我们仅将标记的数据集用作Bert基线中的设置，并设置批次大小为8。在MixText中，我们使用带标签的数据和未带标签的数据进行训练，使用与UDA中相同的设置。 我们将K设置为2，即，对于每个未标记的数据，我们执行两次扩充，特别是德语和俄语。 对于标记的数据，批次大小为4；对于未标记的数据，批次大小为8。 0.5用作调整温度T的起点。在我们的实验中，我们将AG News设置为0.3，将DBpedia和Yahoo!设置为0.5，IMDB为1。

**5.4 Results**

我们评估了基线和提出的方法，使用了5000种未标记的数据以及每类从10到10000（IMDB为5000）的不同数量的标记数据的准确性。

**5.4.1改变标签数据的数量**

表2和图3中显示了不同文本分类数据集的结果。与VAMPIRE相比，所有基于变压器的模型（BERT，TMix，UDA和MixText）表现出更好的性能，因为采用了更大的模型。 TMix的表现优于BERT，尤其是当标签数据限制为每类10个时。 例如，带有10个标记数据的AG News的模型准确性从69.5％提高到74.1％，证明了TMix的有效性。 当在UDA中引入未标记的数据时，它的表现优于TMix，例如Yahoo!上的数据从58.6％增至63.2％。 包含10个标记数据，因为使用了更多数据，并且增加了一致性正则化损失。 与四个基础数据集上的不同基线模型相比，我们提出的MixText始终表现出最佳性能，因为MixText不仅合并了未标记数据，并通过TMix利用了标记数据和未标记数据之间的隐式关系，而且还通过加权对未标记数据进行了更好的标签猜测 扩充句子和原始句子中的平均值。

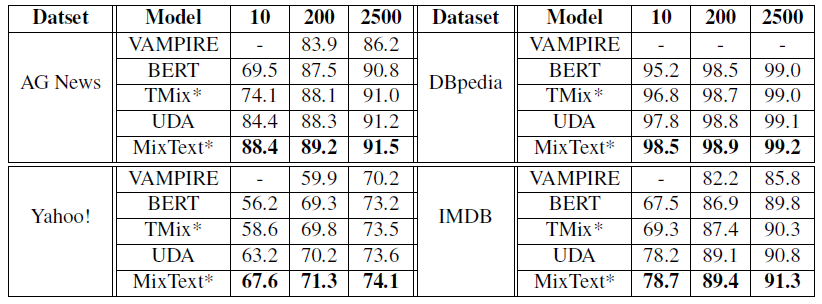


表2：性能（测试准确性（％））与基线的比较。 在三轮运行后平均结果以显示其重要性（Dror等，2018），每轮运行大约需要5个小时。 每个课程使用10、200、2500个标记数据训练模型。VAMPIRE，Bert和TMix在训练期间不使用未标记的数据，而UDA和MixText使用未标记的数据。\*表示我们的型号。

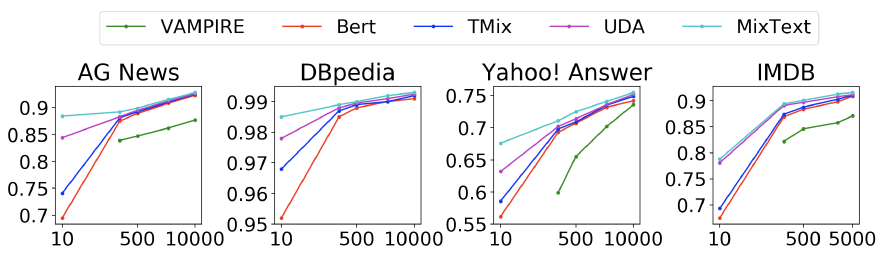


图3：AG News，DBpedia，Yahoo！的性能（测试准确度（％））。 Answer和IMDB具有5000个未标记的数据，并且每个模型的每个类具有不同数量的标记数据。

**5.4.2更改未标记数据的数量**

我们还进行了实验，以在AG News和Yahoo!上使用10个标记数据和不同数量的未标记数据（从0到10000）测试我们的模型性能，如图4所示。有了更多未标记的数据，AG News和Yahoo!的准确性都大大提高了。这进一步验证了使用未标记数据的有效性。

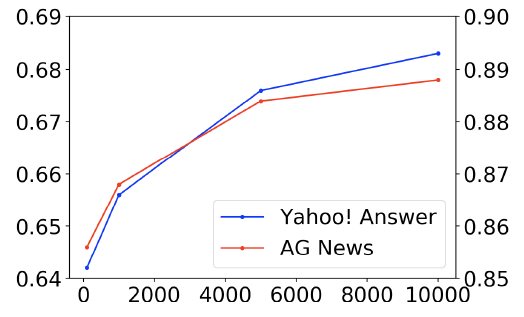


图4：AG新闻（右侧y轴）和Yahoo! （左侧的y轴）的性能（测试准确度（％）。包含10个带标签的数据，以及每个类对MixText使用不同数量的未带标签的数据。

**5.4.3开发集损失**

为了探索在给定有限的标记数据时我们的方法是否可以避免过度拟合，我们绘制了在IMDB和Yahoo!培训期间开发所遭受的损失 在图5中，每个类使用200个带标签的数据进行回答。我们发现，对于Bert，开发集的损失往往会在10个纪元内大量增加，这表明该模型过度适合训练集。 尽管UDA通过一致性正则化可以缓解过度拟合的问题，但TMix和MixText始终显示出更稳定的趋势和更低的损失。 TMix的损耗曲线还表明，即使没有额外的数据，它也可以帮助解决过度拟合的问题。

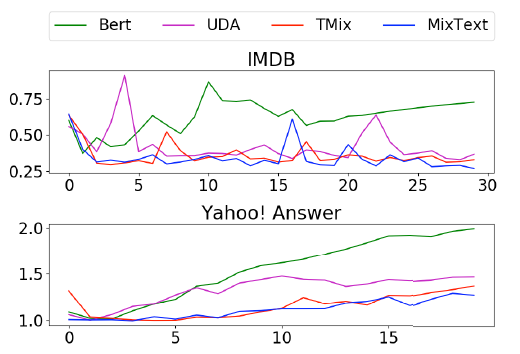


图5：IMDB和Yahoo!上的开发损失，在训练每个类时使用200个标记数据和5000个未标记数据进行训练。

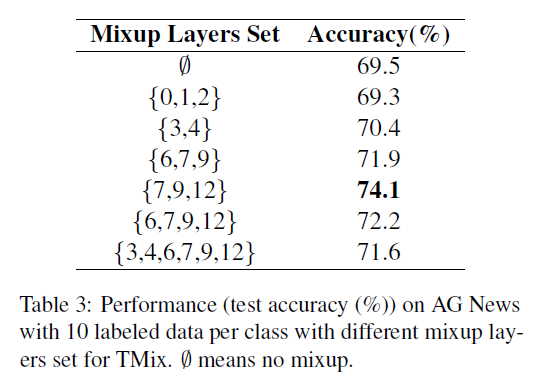
**5.5 Ablation Studies**

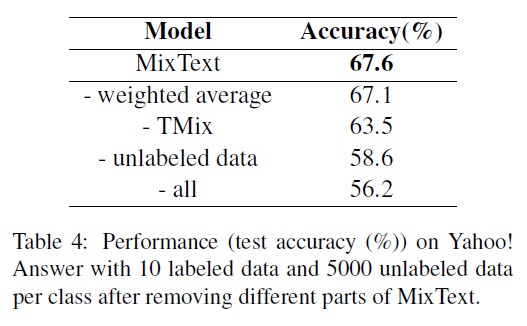
我们进行了消融研究，以显示MixText中每个组件的有效性。

**5.5.1 Different Mix Layer Set in TMix**

我们为TMix探索了不同的混合层集M，结果如表3所示。基于（Jawahar et al。，2019），{3,4,5,6,7,9,12}是BERT中信息最多的层基于模型，并且每个模型都捕获不同类型的信息（例如表面，句法或语义）。

我们选择使用这些层的不同子集进行混合，以查看哪些子集提供了最佳性能。当不进行任何混合时，我们的模型精度为69.5％。如果仅在输入层和较低层（{0、1、2}）进行混合，则似乎性能没有提高。当使用不同的图层集（例如，{3,4}或{6,7,9}）进行混合时，我们发现在模型性能方面存在很大差异：f3,4g主要包含诸如句子长度之类的表面信息对文本分类没有很大帮助 ，因此显示较弱的性能。第六层捕获语法树的深度，这在分类中也没有太大帮助。 我们的模型在{7、9、12}时达到了最佳性能；该层子集包含大多数语法和语义信息，例如语法树中顶级成分的序列，主要子句中的对象编号，对词序的敏感性以及对名词/动词的随机替换的敏感性。





**5.5.2从MixText中删除不同的部分**

我们还通过每次剥离每个组件来测量MixText的性能，并将结果显示在表4中。我们观察到删除每个部分后性能下降，这表明MixText中的所有组件都对最终性能有所贡献。在按预期方式删除未标记的数据之后，模型性能最显着下降。 与未标记数据的加权平均预测相比，删除TMix的减少幅度更大，这表明TMix除未标记数据外，影响最大，这也证明了我们提出的文本混合（基于插值的正则化和扩充技术）的有效性。

**6 Conclusion**

为了减轻监督模型对标记数据的依赖性，这项工作提出了一种简单但有效的半监督学习方法MixText，用于文本分类，其中我们还引入了TMix（一种基于插值的扩充和正则化技术）。 通过对四个基准文本分类数据集进行的实验，我们证明了与现有的预训练和微调模型以及其他状态相比，我们提出的TMix技术和Mixup模型的有效性，该模型具有更好的测试准确性和更稳定的损失趋势。 先进的半监督学习方法。 为了将来的发展方向，我们计划探索MixText在其他NLP任务（例如顺序标记任务和其他带有有限标记数据的实际场景）中的有效性。