山东大学

# 语义计算实验室

## 面向少量标记样本的文本分类

学生姓名 杨涛

指导教师 孙宇清

2021 年 9 月 11 日

# 面向少量标记样本的文本分类综述


## 摘要

文本分类是对于给定文本推断其所属的预定义类别的任务，如情感分析、主题分类、新闻分类等。本文首先介绍文本分类问题的基本概念，讨论少量标记样本情况下文本分类问题的意义和挑战；然后梳理了常用的文本分类数据集和开源工具，以及基于深度神经网络的文本分类方法。最后，综述梳理了面向少量标记数据情况的文本分类方法。



**关键词**：文本分类，深度学习，少量标注样本

# 第一章 简介

## 1.1. 背景意义

文本分类任务是对于给定文本$x$指定类别$y \in C$的任务，其中$C = \{c_1, c_2, \dots, c_t\}$为预定义的类别集合。文本分类在用户情感分析、文章主题分类、新闻分类、问答系统、用户意图识别等方面有着广泛的应用场景。对用户的网上评论进行情感分析可为企业机构等提供决策支持；将文章根据主题、新闻根据类别进行归类可方便信息的检索等等。

借助大量的标记数据进行监督学习完成文本分类任务，已经取得了巨大的成功。但大量的训练数据需要领域专家花费大量时间和精力进行标注，这使得依赖于大量标记数据的监督学习方法在标记数据稀缺的场景下不适用。研究面向少量标记样本的文本分类方法，能够扩大文本分类的应用场景，减少标注成本，具有现实意义。

## 1.2. 问题描述

文本分类是指给定预定义类别集合$C$和训练样本集合$(X, Y)$，其中$C = \{c_1, c_2, \dots, c_t\}$，$X = \{x_1, x_2, \dots, x_m\}$为文本集合，$Y = \{y_1, y_2, \dots, y_m\}$为标签集合，训练模型$M$使其对于给定文本$x$推断其类别$y \in C$。面向少量标记样本的文本分类是指，在标记数据稀缺的领域，借助大量易于获取的无标记文本集合$X^u = \{x_1^u, x_2^u, \dots, x_n^u\}$，辅以少量标记样本$(X^l, Y^l)$一同训练分类模型$M$使其对于给定文本$x$推断其类别$y \in C$。

## 1.3. 挑战

目前有效的文本分类方法，建立在使用大量标记数据的基础上，借助深度神经网络对文本进行向量化，Softmax 等分类器对向量化的文本进行分类。但在一些场景下的文本分类，每个类别可使用的标记数据量从数个到数百不等，标记数据十分稀缺，如法律领域、生物医学领域的文本数据标注需要有专业背景知识的人进行，高昂的标注成本使得标记数据十分稀缺。在这些场景下，若还使用监督

学习方式进行分类，极容易导致模型过拟合，需要研究少量标记样本的分类模型。

# 第二章 评价指标、数据集及开源模型

## 2.1. 评价指标

### 2.1.1. 二分类评价指标

情感分析等文本分类任务只有正类和负类两个类别，常用评价指标为准确率(Accuracy)，精度（Precision），召回率（Recall），F1 score 等。定义:

TP 是将正类预测为正类的样本个数；

FN 是将正类预测为负类的样本个数；

FP 是将负类预测为正类的样本个数；

TN 是将负类预测为负类的样本个数。

准确率计算模型正确分类的样本数占总样本数的比例：

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

精度代表预测为正例的数据中，真正的正例所占比例：

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

召回率计算预测正确的正例数据占实际为正例数据的比例：

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

F1 score 计算$Precision$和$Recall$的调和平均：

$$\mathrm{F1} = \frac{2TP}{2TP + FP + FN} \tag{4}$$

### 2.1.2. 多分类评价指标

对于多分类来说，设总样本数是$N$。最直接的评价指标即考虑所有类别的准确率：

$$Accuracy = \frac{\sum_{i=1}^{N} \boldsymbol{I}(y_i = \hat{y}_i)}{N} \tag{5}$$

## 2.2. 数据集

根据目标任务的不同，将常用文本数据集划分为单文本分类任务的情感分析（Sentiment Analysis，SA）、新闻分类（News Categorization，NC)、主题分类（Topic Analysis，TA）数据集以及成对文本分类任务的释义识别（Paraphrase Detection，PD）和自然语言推断（Natural Language Inference，NLI）等数据集，其中释义识别任务判断给定的两个句子含义是否相同，属于二分类任务；自然语言推理任务，给定一对句子（前提和假设），判断二者的关系属于蕴含、中立、矛盾中的哪一类，属于多分类任务。常用的文本分类数据集及其相关信息如表 1 所示。

表 1.常用文本分类数据集

| 数据集 | 类别数 | 任务 | 描述 | 使用论文 | 链接 |
|---|---|---|---|---|---|
| Yelp[3] | 2 或 5 | SA | 包含两种情感分类任务的数据。一种是检测细粒度的标签，称为 Yelp-5。另一个预测负面和正面情绪的 Yelp-2。Yelp-5 每个类别有 650,000 个训练样本和 50,000 个测试样本，Yelp-2 包含 560,000 个训练样本和 38,000 个针对积极和消极类的测试样本。 | [52][46] | 链接 |
| IMDB[4] | 2 | SA | 为电影评论的二分类情感分类任务而开发的。IMDB 由相等数量的正面和负面评论组成，均为 25,000 条评论。 | [16][37][41] | 链接 |
| MR[5] | 2 | SA | 电影评论数据集，其目的是检测与特定评论相关的情绪并确定其是负面还是正面的。它包括 10662 个句子样本。 | [25][29] | 链接 |
| SST[6] | 2 或 5 | SA | 斯坦福情感树库数据集。有两个版本可用，SST-1 带有细粒度标签（五分类），另一个带有二分类的标签(SST-2)。SST-1 包含 11855 条电影评论，分为 8544 个训练样本，1101 个验证样本和 2210 个测试样本。SST-2 分为三组，分别为训练集，开发集和测试集，大小分别为 6,920、872 和 1,821。 | [16][25][19] | 链接 |

表 1.常用文本分类数据集（续）

| | | | | | |
|---|---|---|---|---|---|
| AG News[7] | 4 | NC | 包括 120,000 个训练样本和 7,600 个测试样本。每个样本都是带有四类标签的短新闻文本。 | [41][55][47] | 链接 |
| DBpedia[8] | 14 | TA | 大规模的多语言知识库，根据 Wikipedia 中最常用的信息框创建的。最受欢迎的版本包含 560,000 个训练样本和 70,000 个测试样本，包含 14 类标签。 | [41][55][47] | 链接 |
| Yahoo Answers[9] | 10 | TA | Yahoo Answers 是一个包含 10 个类的主题标记任务。包括 140000 个训练数据和 5000 个测试数据。文本包含问题标题、问题上下文和最佳答案三个元素。 | [47][55] | 链接 |
| Quora[10] | 2 | PD | 用于释义识别（检测重复的问题）。包含超过 40 万个问题对，每个问题对标有一个二进制值，指示两个问题是否相同。 | [37] | 链接 |
| SNLI[11] | 3 | NLI | 斯坦福自然语言推理数据集，被广泛用于 NLI。该数据集由 550,152、10,000 和 10,000 个句子对组成，分别用于训练，开发和测试。 | [23][38] | 链接 |
| Multi-NLI[12] | 3 | NLI | 是 SNLI 的扩展，是一个 433k 句子对的集合，涵盖了口语和书面语体裁。 | [23][38] | 链接 |
| SICK[13] | 3 | NLI | 包含大约 10,000 个英语句子对，用三个标签进行注释：蕴涵，矛盾和中立。 | [23][38] | 链接 |

## 2.3. 开源模型性能对比

### 2.3.1. 开源深度学习文本分类模型性能对比

表 2 列举了经典的基于深度神经网络的文本分类模型在常见的文本分类数据集上的性能比较。基于“预训练-微调”两阶段的分类模型如 BERT[37]、RoBERTa[39]、XLNet[41]等的效果要远远好于在特定任务上进行监督训练的分类模型。

表 2. 经典文本分类模型性能对比

| | SA | | | NC | TA | NLI |
|---|---|---|---|---|---|---|
| model | MR | SST-2 | IMDB | AG News | DBpedia | SNLI |
| TextCNN [25] | 81.5% | 88.1% | | | | |
| DAN [16] | | 86.3% | 89.4% | | | |
| LSTM [19] | | 84.9% | | | | |
| Bi-LSTM [19] | | 87.5% | | | | |
| Tree-LSTM [19] | | 88% | | | | |
| TopicRNN[21] | | | 93.72% | | | |
| BLSTM-2DCNN [29] | **82.3%** | 89.5% | | | | |
| DPCNN[28] | | | | 93.13% | 99.12% | |
| RNN-Capsule[22] | 83.8% | | | | | |
| Bert-base [37] | | 93.5% | 95.63% | | | 91.0% |
| Bert-large [37] | | 94.9% | 95.79% | | | 91.7% |
| XLNet-large [41] | | 94.4% | **96.8%** | **95.55%** | **99.4%** | |
| RoBERTa [39] | | **96.4%** | | | | **92.6%** |

### 2.3.2. 面向少量标记样本的文本分类模型性能对比

面向少量标记样本的文本分类模型，其性能对比如表 3 所示。使用来自情感分析、主题分类、新闻分类领域的数据集以及相同数量的少量训练样本对模型 VAMPIRE[44]、BERT[37]、TMix[55]、UDA[52]、MixText[55]性能进行比较，其中 BERT 和 TMix 在少量标记样本上进行监督学习，为表格中的基线模型。10，200，2500 为各个类别的标记样本数量。

表 3. 面向少量标记样本的文本分类模型性能对比（数据来源[55]）

| Dataset | Model | 10 | 200 | 2500 | Dataset | Model | 10 | 200 | 2500 |
|---|---|---|---|---|---|---|---|---|---|
| AG-News | VAMPIRE[44] | - | 83.9 | 86.2 | DBpedia | VAMPIRE | - | - | - |
| | BERT[37] | 69.5 | 87.5 | 90.8 | | BERT | 95.2 | 98.5 | 99.0 |
| | TMix[55] | 74.1 | 88.1 | 91.0 | | TMix | 96.8 | 98.7 | 99.0 |
| | UDA[52] | 84.4 | 88.3 | 91.2 | | UDA | 97.8 | 98.8 | 99.1 |
| | MixText[55] | **88.4** | **89.2** | **91.5** | | MixText | **98.5** | **98.9** | **99.2** |
| Yahoo answers | VAMPIRE | - | 59.9 | 70.2 | IMDB | VAMPIRE | - | 82.2 | 85.8 |
| | BERT | 56.2 | 69.3 | 73.2 | | BERT | 67.5 | 86.9 | 89.8 |
| | TMix | 58.6 | 69.8 | 73.5 | | TMix | 69.3 | 87.4 | 90.3 |
| | UDA | 63.2 | 70.2 | 73.6 | | UDA | 78.2 | 89.1 | 90.8 |
| | MixText | **67.6** | **71.3** | **74.1** | | MixText | **78.7** | **89.4** | **91.3** |

从表中可以看出，标记样本数量的增多，会带来各个模型性能的提升。

针对小样本情况，预训练方式的使用上，BERT 基于 Transformer 在海量无标记数据上预训练掩码语言模型和下一句预测任务，其效果远好于使用 VAE 进行自编码预训练的 VAMPIRE 模型。

针对小样本情况，使用 Mixup[53,54]插值技术获取融合样本进行训练（TMix 模型），效果好于 BERT 基线，二者有着相同的模型深度和参数量。

针对小样本情况，使用数据增强和一致性训练的方式，UDA 以及 MixText 效果远远好于使用 BERT 在少量样本上微调的方法。对比 UDA 及 MixText，UDA 获取一致性标记仅使用原文本的锐化后的预测结果，MixText 使用了增强文本和原文本的预测结果加权求和再锐化的方式，蕴含集成的思想；另外 MixText 创新地将 Mixup 技术应用于 NLP 领域，使得无标记样本和有标记样本在训练过程中产生信息流动，对比 UDA 有标记样本和无标记样本的“割裂式”训练，可获得更好的性能。

# 第三章 基于深度神经网络的文本分类方法

文本分类任务，首先需要将文本向量化，提取分类相关的特征。传统文本分类方法中基于人工设计的特征进行文本分类。深度学习中基于深度神经网络对文本进行特征向量编码，获得文本的分类特征接分类器进行分类，图 1 展示了经典的文本分类技术发展脉络。本章主要介绍基于深度神经网络的文本分类方法。

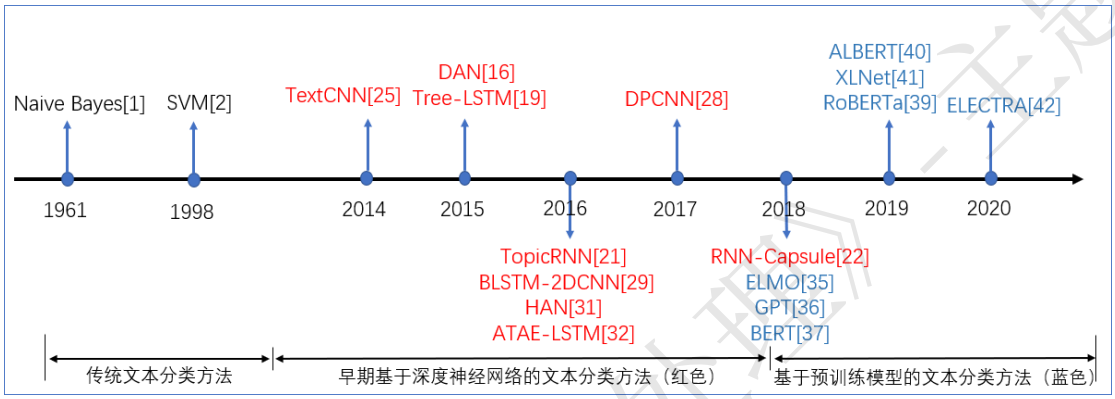


图 1. 经典文本分类方法发展脉络

## 3.1. 基于前馈神经网络的文本分类方法

在文本的词汇级别，基于分布式假设进行预训练的词向量如 word2vec[14]、GloVe[15]等包含着单词的语义信息。基于此，Iyyer[16]等提出如图 2 所示的深度平均网络(Deep averaging network, DAN)，借助词向量和前馈神经网络（Feed-forward neural network，FFNN）进行文本分类。首先将词序列的每个词的词向量进行求和平均，并输入到多层前向传播网络中，每一层的隐藏向量通过公式 6 计算得到，最终输出的向量作为句子的特征表示，使用 softmax 函数获得分类结果，如公式 7 所示。

$$\boldsymbol{h}_i = f(\boldsymbol{W}_i \cdot \boldsymbol{h}_{i-1} + b_i) \tag{6}$$

$$\hat{y} = softmax(\boldsymbol{W}_s \cdot \boldsymbol{h} + \boldsymbol{b}) \tag{7}$$

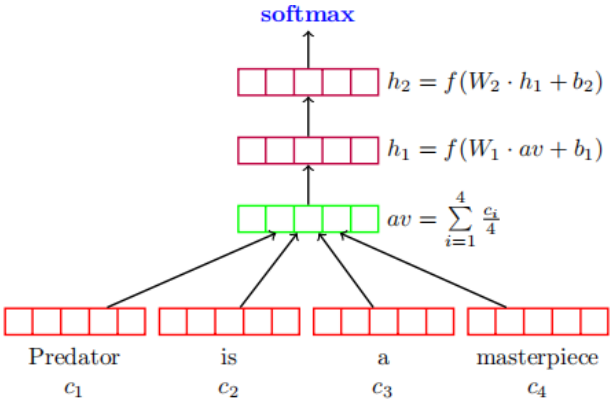


图 2. DAN 文本分类模型[16]

### 3.2. 基于循环神经网络的文本分类

文本通常包含语序信息，这对文本分类任务十分重要。基于前向传播神经网络的模型忽略了语序信息。深度学习中，循环神经网络(Recurrent neural networks，RNN)是一种专门用来处理序列化数据的神经网络，能够捕获句子中各个词的依赖信息，并建模语序信息。传统的 RNN 每个时间步输入当前位置的词向量$\boldsymbol{x}_t$以及前一时间步输出的隐藏状态$\boldsymbol{h}_{t-1}$，计算出当前的隐藏状态$\boldsymbol{h}_t$，如公式 8 所示。$\boldsymbol{W}_h$、$\boldsymbol{W}_x$和$\boldsymbol{b}$是各时间步共享的权重。参数的共享使得 RNN 能够处理任意序列长的文本数据，但也因此存在梯度消失问题。

$$\boldsymbol{h}_t = \sigma(\boldsymbol{W}_{\mathrm{h}}\boldsymbol{h}_{t-1} + \boldsymbol{W}_x\boldsymbol{x}_t + \boldsymbol{b}) \tag{8}$$

Schmidhuber[17]等提出长短期记忆模型(Long short-term memory，LSTM)，增加记忆向量和门控单元缓解梯度消失问题。图 3 展示了一个 LSTM 的单元结构。

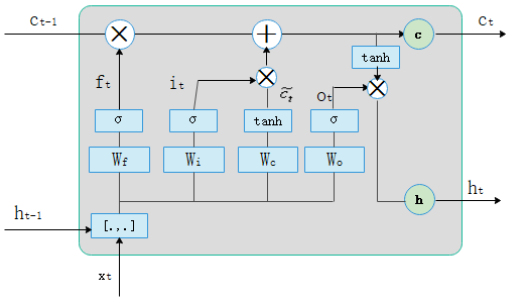


图 3. LSTM 单元结构

LSTM 的单元内部计算如公式 9-14 所示，$\boldsymbol{f}_t$、$\boldsymbol{i}_t$、$\boldsymbol{o}_t$分别为遗忘门、输入门、输出门的输出向量，各元素均在 0 到 1 之间。以此控制信息的去留。最终获得$\boldsymbol{t}$时间步的记忆向量$\boldsymbol{c}_t$和隐藏向量$\boldsymbol{h}_t$。

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_f\right) \tag{9}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_i) \tag{10}$$

$$\tilde{\boldsymbol{c}}_t = tanh(\boldsymbol{W}_c \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_c) \tag{11}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \tilde{\boldsymbol{c}}_t \tag{12}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_o) \tag{13}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ tanh(\boldsymbol{c}_t) \tag{14}$$

Cho[18]等提出门控循环网络(gated recurrent unit，GRU)，将 LSTM 中三个控制门简化为两个。Tai[19]等人提出 Tree-LSTM 模型，将 LSTM 推广到树形结构，适用于语言中的语法结构。

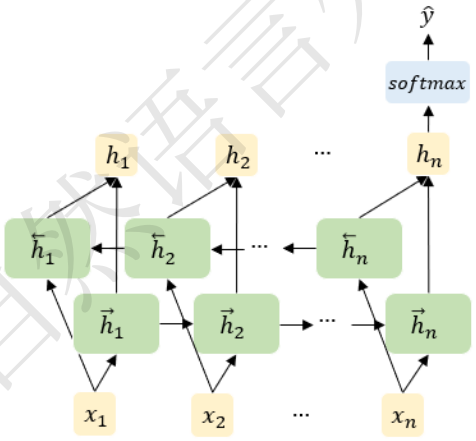


图 4. Bi-LSTM 文本分类模型

基于循环神经网络对文本进行分类。Tai[19]等人使用 LSTM，双向 LSTM（Bi-LSTM），Tree-LSTM 进行文本分类任务，对比得出在情感分类等任务上 Tree-LSTM 效果好于序列 LSTM 模型，图 4 展示了一个 Bi-LSTM 分类模型，使用最后时间步的隐藏向量$\boldsymbol{h}_n$作为文本特征进行分类。Liu[20]等人提出基于多任务学习和循环神经网络的文本分类模型。Dieng[21]等和 Wang[22]等也都将循环神经网络用于文本分类任务。Conneau[23]等提出 InferSent 模型，使用共享参数的两个 Bi-LSTM

编码句子，进行自然语言推理任务，同时获取句嵌入向量。Miyato[24]等人使用LSTM 编码文本，提出虚拟对抗训练（Virtual Advertise Training）方法，对无标签样本加入噪声后进行一致性训练，进行半监督的文本分类。

### 3.3. 基于卷积神经网络的文本分类

卷积神经网络(Convolutional neural network，CNN)也可用来处理文本。相比于 RNN，CNN 更容易捕捉片段信息比如一个短语，可提取 n-gram 的特征信息，整合这些信息来表示整个句子。

Kim[25]提出 TextCNN 模型，使用单层的 CNN 和池化操作获取句子的嵌入表示用于文本分类。设词向量$\boldsymbol{x}_i \in R^k$，使用词向量进行拼接得到$\boldsymbol{x}_{1:n} = \boldsymbol{x}_1 \oplus \boldsymbol{x}_2 \oplus \ldots \oplus \boldsymbol{x}_n$，设卷积核大小为$\mathrm{h}$，以$\mathrm{h}$为句子划分成子序列$\{\boldsymbol{x}_{1:\mathrm{h}}, \boldsymbol{x}_{2:\mathrm{h}+1}, \ldots, \boldsymbol{x}_{n-\mathrm{h}+1:n}\}$，用卷积核$\boldsymbol{w} \in R^{\mathrm{h}*k}$在其上滑动进行卷积，卷积公式为公式 15 所示。

$$c_i = f(\boldsymbol{w}^T \boldsymbol{x}_{i:i+\mathrm{h}-1} + b) \tag{15}$$

对卷积后的向量$\boldsymbol{c} = [c_1, c_2, \ldots, c_{n-\mathrm{h}+1}] \in R^{n-\mathrm{h}+1}$进行池化，得到$\hat{c} = maxpooling\{\boldsymbol{c}\}$。使用$m$个卷积核进行卷积可获得最终句向量$\boldsymbol{z} = [\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_m]$。图 5 为 TextCNN 文本分类模型。

Johnson 和 Zhang 提出多篇[26,27,28]基于卷积神经网络的文本分类模型。Zhou[29]等人将 LSTM 与 CNN 的特点进行结合提出 BLSTM-2DCNN 的文本分类改进方法。

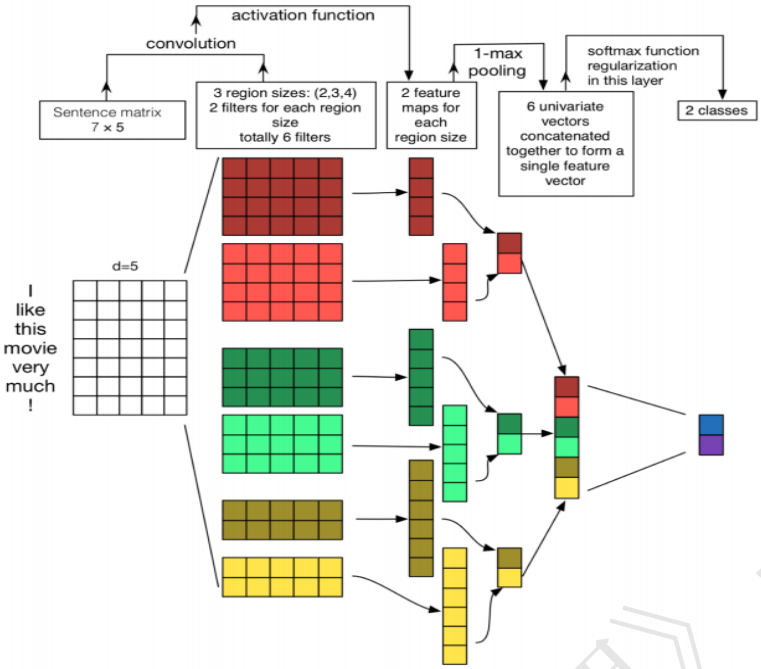


图 5. TextCNN 模型[25]

### 3.4. 基于注意力机制的文本分类

Bahdanau[30]等人使用基于 RNN 的“编码-解码”架构做机器翻译任务，首次提出了注意力机制，解码器第$i$时间步的隐藏状态$s_i = f(s_{i-1}, y_{i-1}, c_i)$，$y_{i-1}$为上一时间步的输出词，$c_i = \sum_{j=1}^{T} \alpha_{ij} h_j$为通过注意力机制获得的第$i$时间步特定的上下文向量，解码每个单词时对源输入的不同位置单词具有不同的关注程度，其中$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} e_{ik}}$，$e_{ij} = a(s_{i-1}, h_j)$，原文中$a$为前馈神经网络。

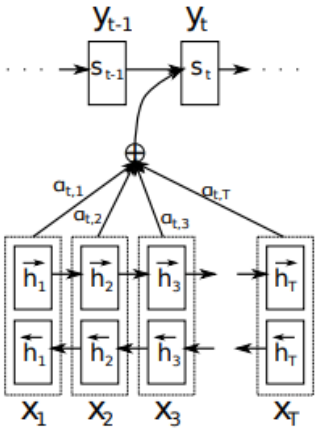


图 6. 基于注意力机制的解码过程[30]

Yang[31]等人提出分层注意力网络（Hierarchical Attention Networks, HAN），先后从单词级别和句子级别进行文本编码和注意力计算，进行文档分类任务。Wang[32]等人提出 ATAE-LSTM 模型，在 LSTM 编码器的基础上结合 attention 机制对句子进行语义建模，解决 aspect level 情感分析问题。

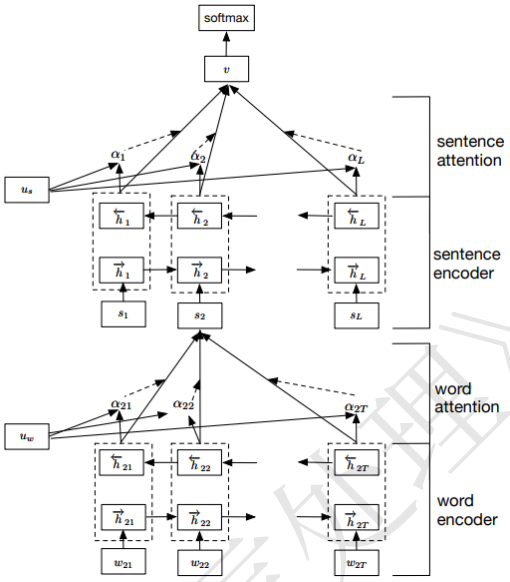


图 7. Hierarchical Attention Network（HAN）[31]

在注意力机制中，自注意力机制（self-attention）通过构建 K（key）、Q(query)、V(value)矩阵计算句子中各个单词之间的权重分布，不借助$s_{i-1}$这样的外部信息，只借助$h_j$这样的隐藏状态进行注意力权重计算，能够捕获文本分类任务当中的长距离依赖信息。Lin[33]等基于 Bi-LSTM 模型的隐藏向量进行自注意力的计算，进行文本分类任务。图 8 为自注意力计算。对于位置$i$输入的词向量$a_i$，派生出三个$n$维的参数向量$q_i$、$k_i$、$v_i$，经过自注意力计算，获得位置$i$输出的词向量$b_i = \sum_j softmax(\frac{q_i \cdot k_j}{\sqrt{n}})v_j$，$b_i$相比于$a_i$包含了更多的上下文信息。

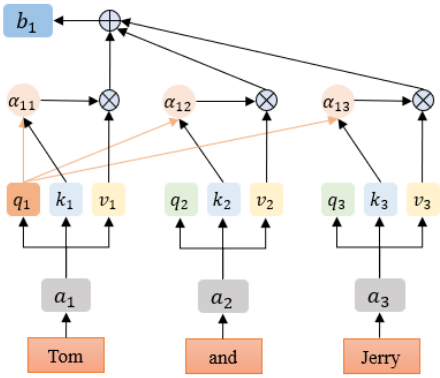


图 8. 自注意力计算

注意力机制解决了 RNN 及其变体模型不能并行计算的问题，并解决了 RNN 中存在的长期依赖问题。相比于 CNN 在捕获长距离特征方面的天然缺陷，注意力机制能做到更好。

### 3.5. 基于预训练语言模型的文本分类

Transformer[34]是 Google 在 2017 年提出的基于深层自注意力的机器翻译模型。此后基于 Transformer 的预训练语言模型陆续涌现。预训练的语言模型通过构建无监督的预训练目标自动挖掘语义知识，可以有效地学习上下文语义，动态地调整词汇在不同语境下的表示。基于“预训练-微调”两阶段的学习方式，显著提高包括文本分类在内的 NLP 任务。

最早的两阶段语言模型是 Peters[35]等在 2018 年提出的 ELMO 模型，使用双层双向的 LSTM 作为编码器，第一个阶段是利用语言模型进行预训练，第二个阶段是在做下游任务时，将句子输入预训练网络中提取对应单词的网络各层（词嵌入层、第一层 LSTM、第二层 LSTM）的词嵌入经过加权求和获得新特征补充到下游任务中，可以解决静态词向量如 word2vec 中存在的无法表示多义词的问题，但是 ELMO 所使用的编码器 LSTM 的特征提取性能远不如 Transformer。相比于 ELMO 的基于特征融合的两阶段训练方式，Radford[36]等提出的 GPT 是第一个

基于“预训练-微调”的两阶段训练模型，第一个阶段是利用语言模型进行预训练，第二阶段通过微调解决下游任务。GPT 的预训练的语言模型是单向的语言模型，这使得其不适用于阅读理解等需要上下文信息的任务。Devlin[37]等提出的 Bert 模型成为集大成者，同样是“预训练-微调”两阶段训练方式，不同在于在采用了 Mask language model(MLM)这一完型填空式任务进行预训练，能够利用双向的上下文信息。图 9 为上述三种预训练语言模型架构。使用预训练的 Bert 既可以对单文本也可对成对文本进行分类。如图 10 所示，使用预训练后的 Bert 将文本编码后，使用[CLS]位置输出进行文本分类即可。

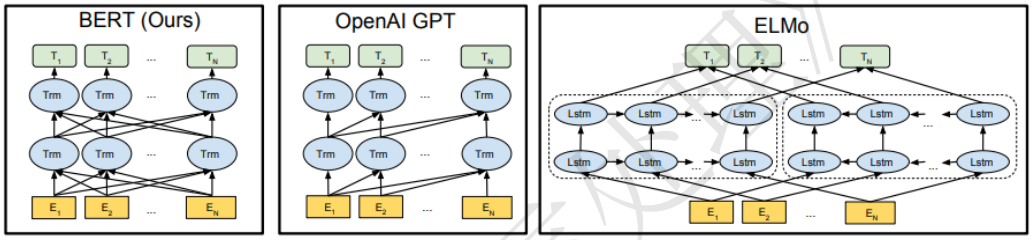


图 9. 预训练语言模型结构对比[37]

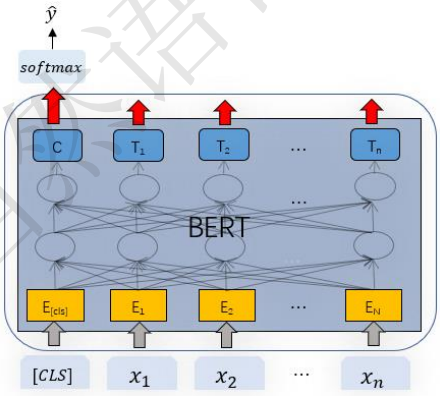


图 10. 使用预训练 Bert 进行单文本分类[37]

Reimers[38]等提出 Sentence-Bert 模型，使用孪生的 Bert 编码器进行 NLI 自然语言推理任务。预训练语言模型如 Bert 有着良好的迁移性，在 11 个包括文本分类在内的 NLP 任务中达到当时最好的效果，某些任务性能有极大的提升。Bert 的极大成功刺激了预训练语言模型的研究，在此之后涌现出大量基于 transformer 的语言模型，如 RoBERTa[39]，ALBERT[40]，XLNet[41]，ELECTRA[42]

等。

### 3.6. 小结

基于深度神经网络的文本分类模型，使用“深度神经网络+分类器”的方式先编码文本的分类特征再分类，早期使用前向传播神经网络对文本进行编码，忽略了文本的语序特征。利用循环神经网络对文本进行编码，建模了文本的语序信息，卷积神经网络对文本进行编码，有利于建模 n-gram 信息。但循环神经网络和卷积神经网络分别存在无法并行计算和长距离建模问题，注意力机制则不存在这些问题，并且注意力机制可以与其他深度神经网络结合起来使用，提升分类性能。但这些深度神经网络分类模型，不具有普适性，模型编码的文本特征往往面向特定的分类任务，不能复用。基于预训练语言模型的文本分类模型使用“预训练-微调”的方法，利用下游分类任务数据进行微调，不仅可复用预训练后的模型参数，还可获得比以往的深度神经网络分类模型更出色的性能。

但少量样本情况下，仅使用“深度神经网络+分类器”的方式进行监督学习，会容易发生过拟合，因为少量样本提供的监督信息在分类空间中不足以产生鲁棒的分类边界，更好的文本分类模型应当被设计以面对少量标记样本的情况。

# 第四章 面向少量标记样本的文本分类方法

文本分类任务，依赖于标记数据的监督信息进行训练。若只有少量的标记数据提供监督信号，在分类空间中不足以产生鲁棒的分类边界，利用大量无标记数据或者增强数据一同进行模型的训练，可以优化分类边界，以获得较强分类性能。本文根据模型的训练方式，可将面向少量标记样本的文本分类方法做出表 4 中的归类：

表 4. 面向少量标记样本的文本分类方法归纳

| 训练方式及模型 | | 外部数据 | 领域内标记数据 | 领域内无标记数据 |
|---|---|---|---|---|
| 预训练 | BERT[37]等 | 进行语言模型预训练 | 可进一步利用领域内数据进行预训练 | 可进一步利用领域内数据进行预训练 |
| | VAMPIRE[44] | -- | 监督训练 | 自编码任务预训练 |
| 自训练 | Pseudo-label[45] | -- | 训练初始分类器 | 预测伪标记加入训练集、损失影响由弱到强 |
| | Delta-training[46] | -- | 训练初始分类器 | 预测伪标记、选取信息量大的数据加入训练集 |
| | SALNet[47] | -- | 训练初始分类器 | 用分类器及词表预测伪标记，加入训练集 |
| 一致性训练 | VAT[24,49] | -- | 加噪声后监督训练 | 加噪声后一致性训练 |
| | Π-model、Temporal ensembling[51] | -- | 监督训练、原样本与增强样本进行一致性训练 | 原样本与增强样本进行一致性训练 |
| | UDA[53] | -- | 监督训练 | 回译、TF-IDF 词替换获得增强数据，在一致性训练 |
| | MixText[55] | -- | Mixup 混合样本后监督训练 | 回译获取增强样本、mixup 混合样本后进行一致性训练 |

## 4.1. 预训练

少量标记样本提供的监督信号较少，但是有大量可供利用的无标记文本。采用一些预训练任务，从大量无标记文本中学习语言学知识，再将学得的模型利用少量标记样本进行监督学习，有助于提升分类效果。预训练任务可以是掩码语言模型（Masked Language Model，MLM）等语言模型预训练任务，也可使用自编

码任务进行预训练。

### 4.1.1. 语言模型预训练

BERT 等语言模型的两阶段学习方式如图 11 所示，利用深度网络和无标记数据进行预训练，微调下游少量标记样本，可增强下游任务上的性能。Sun[43]等人研究利用领域内的无标记数据继续进行语言模型的预训练，可提升领域内的文本分类效果。

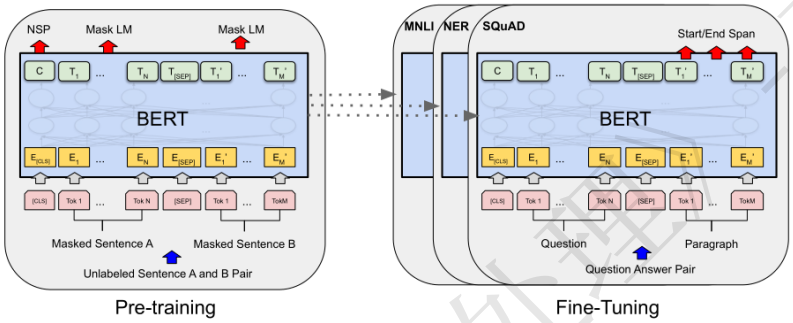


图 11. BERT 的“预训练-微调”两阶段学习方式[37]

### 4.1.2. 自编码预训练

基于“编码-解码”结构，对输入数据进行重构，中间编码获得的向量包含原始文本的全部语义特征。Gururangan[44]等人提出 VAMPIRE 模型，利用变分自编码器(variational autoencoder ,VAE)预先对无标记数据进行自编码任务，之后利用预训练的 VAE 的编码标记文本进行文本分类的监督训练。

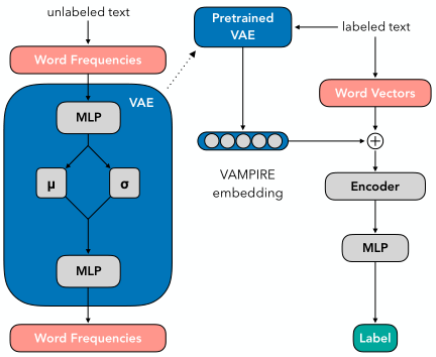


图 12. VAMPIRE 模型[44]

## 4.2. 自训练

自训练方法使用迭代训练的思想，首先使用有限的标记数据训练初始分类器，然后对无标记数据进行预测，选择置信度高的样本标注伪标记，并加入到训练集中，继续训练，直到没有可以加入的无标记样本为止。

Lee[45]等人提出 Pseudo-label 模型，用训练中的模型对无标记数据进行预测，以概率最高的类别作为无标记数据的伪标记，将拥有伪标记的无标记数据视为有标记的数据。运用熵最小化思想，将对无标记数据的预测加入目标函数作为正则项，使用交叉熵来评估误差大小，即损失$\mathrm{L}=\frac{1}{n}\sum_{m=1}^{n}\sum_{i=1}^{C}L(y_i^m,f_i^m)+\alpha(t)\frac{1}{n'}\sum_{m=1}^{n'}\sum_{i=1}^{C}L(y'^m_i,f'^m_i)$，第一项为交叉熵，第二项为无标记样本的正则化项。此外，模型为了平衡有标签数据和无标签数据的信号强度，在目标函数中引入了时变参数$\alpha(\mathrm{t})=\begin{cases}0 & t<T_1\\ \frac{t-T_1}{T_2-T_1}\alpha_f & T_1\le t<T_2\\ \alpha_f & T_2<t\end{cases}$，逐步释放无标记数据的信号。

Jo 和 Cinarel[46]提出 Delta-training 模型，基于 n 个随机初始化词向量的模型($M_{rand}$)以及 n 个使用预训练词向量的模型($M_{emb}$)分别集成，在训练集上进行训练，在验证集上进行验证和早停，之后冻结参数对无标记样本进行预测，对于$M_{emb}$和$M_{rand}$预测结果不同的样本，模型认为其具有更高的信息量，将其作为下一轮的训练样本加入训练集中，伪标签为$M_{emb}$的预测标签，因为$M_{emb}$比$M_{rand}$具有更好的性能。

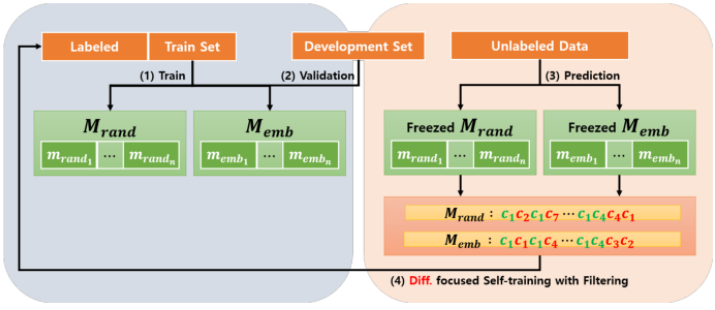


图 13. Delta-training 训练流程图[46]

Lee[47]等人提出模型 SALNet，基于注意力机制，利用标记样本训练分类器后，在对无标记样本进行预测时获取对每个单词的注意力，从中挑选出每个类的

代表词汇构建类别词库。在获取无标记样本的伪标记时，利用分类器的预测结果和词库的匹配结果二者共同作用。提高伪标记的准确性。Du[48]等人认为之前的半监督文本分类方法仍依赖于领域内的无标记数据。提出了 SentAugment 方法用于从海量的网络文本数据中寻找任务相似的数据作为无标记数据解决这一依赖。使用任务的训练文本生成任务编码，从海量文本中检索相似文本。之后使用自训练方法进行学习。用训练集微调预训练语言模型 RoBERTa[39]作为教师网络，并使用教师网络对无标记数据进行预测，选取每个类的 top k 文本作为伪标记数据加入训练集，训练学生网络。

自训练式的半监督文本分类方法存在一个主要缺点，即错误累积问题。模型很难纠正自己的错误，使得偏差在训练中被放大。上述的逐步释放无标记信号、多分类器集成、建立辅助词库等方法，能够提升为无标记样本标注伪标记的准确率，缓解错误累积问题。

### 4.3. 一致性训练

半监督学习的一个新的研究方向是一致性训练，也叫一致性正则化。对于分类问题，若学习得到的决策边界位于低密度区域，则模型具有较强的泛化能力。基于此，一致性正则在不改变原文本语义(标签相同)的情况下进行数据增强获得增强文本，或者对输入数据进行噪声扰动获得噪声样本，以之平滑分类边界。由于增强（噪声）文本的语义和原文本语义相同，分类器应当对二者输出一致的概率分布。即对于无标记文本$x$，其增强样本$\mathrm{Augment}(x)$应当被分为同一个类。借助无标记样本及其增强样本，度量二者概率分布的差异来获得一致性损失，加入目标函数中一同优化，模型可获得强有力的泛化效果。常用的度量概率分布差异的方式有均方误差（MSE）、KL 散度、JS 散度等。

#### 4.3.1. 噪声对抗

对抗训练(Adversarial Training)是在目标函数中加入扰动样本的正则化项，使

得模型对输入样本的扰动版本保持鲁棒性，由 Goodfellow[50]等人首先提出。Miyato 等人在此基础上提出虚拟对抗训练(Virtual Adversarial Training, VAT)[24,49]并将之运用于半监督文本分类。输入文本后，VAT 对于有标记的输入文本的词嵌入做梯度方向上的扰动$\boldsymbol{r}_{adv} = -\epsilon \boldsymbol{g}/||\boldsymbol{g}||_2$，其中$\boldsymbol{g} = \nabla_{\boldsymbol{s}} log\ p(y|\boldsymbol{s};\widehat{\boldsymbol{\theta}})$，对无标记的输入文本做 KL 散度变化最快方向上的扰动$\boldsymbol{r}_{v-adv} = -\epsilon \boldsymbol{g}/||\boldsymbol{g}||_2$，其中$\boldsymbol{g} = \nabla_{\boldsymbol{s}+\boldsymbol{d}} KL[p(\cdot\,|\boldsymbol{s};\widehat{\boldsymbol{\theta}})||p(\cdot\,|\boldsymbol{s};\widehat{\boldsymbol{\theta}})]$，对扰动后的标记文本用其标记计算交叉熵损失：$L_{adv}(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N} log\ p(y_n|\boldsymbol{s}_n + \boldsymbol{r}_{adv,n};\boldsymbol{\theta})$，对扰动后的无标记文本和原文本的输出概率分布计算 KL 散度损失来维持一致性：$L_{v-adv}(\boldsymbol{\theta}) = -\frac{1}{N'}\sum_{n'=1}^{N'} KL[\,p(\cdot\,|\boldsymbol{s}_{n'};\widehat{\boldsymbol{\theta}})||p(\cdot\,|\boldsymbol{s}_{n'} + \boldsymbol{r}_{adv,n'};\boldsymbol{\theta})\,]$。这种加入噪声的训练方式使得模型对噪声不敏感，使得决策边界偏向于低密度区域。

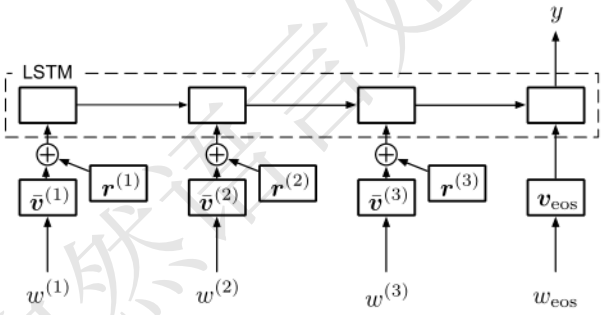


图 14. VAT 噪声注入[24]

### 4.3.2. 数据增强

数据增强在图像领域被广泛使用，通过旋转、缩放、剪裁图片等方式就可以获得类别不变的增强样本。由于语言的离散性，自然语言的数据增强稍复杂，常用的增强方式有同义词替换、回译、随机插入、随机交换、随机删除等。

#### (1) π-model 和 Temporal ensembling 模型

Laine[51]等在图像分类领域，于 2017 年提出使用数据增强和 Dropout 获得语义不变的增强样本，并设计如图 15 所示 π-model 和 Temporal ensembling 两个半监督学习框架，对 NLP 领域的半监督文本分类模型设计具有重要借鉴意义。

π-model 对于任何给定的输入$x$，经过数据增强和 Dropout 计算两次，目标是减小两次预测结果之间的距离，提升模型在不同扰动下的一致性，π-model 使用 MSE 作为两个概率分布之间的损失函数。若文本具有标记$y_i$则额外计算交叉熵损失。

Temporal ensembling 模型的整体框架与 π-model 类似，不同在于目标函数中， Pi-Model 是两次前向计算结果的均方差，而在 Temporal ensembling 中使用时序组合模型，采用的是当前模型预测结果与历史预测结果的平均值计算均方差。相对于 π-model，Temporal ensembling 减少前向推理次数因而减少了训练时间；通过历史预测做平均，有利于平滑单次预测中的噪声。

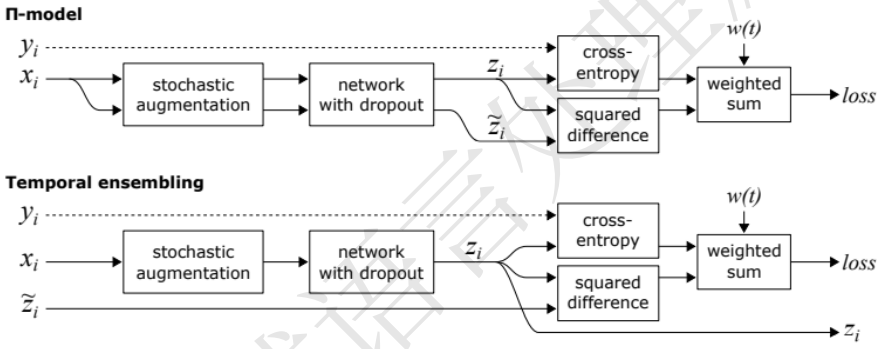


图 15. 半监督学习框架——π-model 和 temporal ensembling[51]

**(2) Unsupervised data augmentation（UDA）**

之前的工作中对无标记数据加入噪声进行增强的方式，采用的主要是简单随机噪声。Xie[52]等人发现选择何种噪声增强方式对模型的性能提升有着十分重要的影响，并提出 UDA 模型，对无标记数据采取更多样化、更真实的数据增强方式。在文本领域，采用回译(back-translation)技术以及 TF-IDF 词替换技术对无标记文本进行数据增强，获得语义一致的样本。UDA 对有标记的文本计算交叉熵损失进行监督学习，对无标记文本及其增强文本采用交叉熵计算一致性损失,其损失函数有两部分组成：$L(\boldsymbol{\theta}) = E_{x_1 \sim p_L(x)}[-log p_{\boldsymbol{\theta}}(f^*(x_1)|x_1)] + \lambda E_{x_2 \sim p_U(x)} E_{\hat{x} \sim q(\hat{x}|x_2)}[D_{KL}(p_{\bar{\theta}}(y|x_2) \,||\, p_{\boldsymbol{\theta}}(y|\hat{x}))]$。其中$p_L(x)$和$p_U(x)$分别代表有标

记和无标记数据分布，$q(\hat{x}|x_2)$代表无标记样本的增强样本分布。

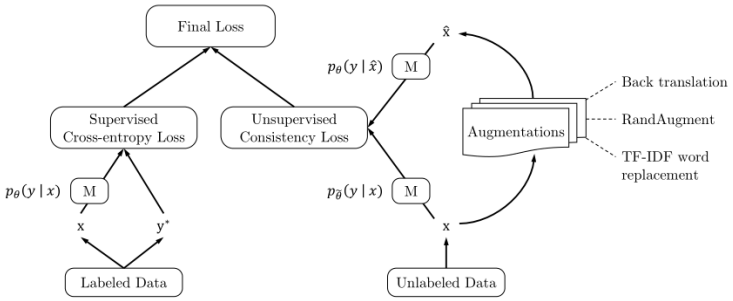


图 16. UDA 架构图[52]

此外，在计算一致性损失时，设定阈值$\beta$剔除置信度不高的样本；并在计算无标记样本的概率分布时使用锐化技术，获得熵值更低的概率分布，损失函数第二项被改写为$\frac{1}{|B|}\sum_{x\in B} I\left(max_{y'} p_{\tilde{\theta}}(y'|x) > \beta\right) D_{KL}(p_{\tilde{\theta}}^{(sharp)}(y|x) \,||\, p_\theta(y|\hat{x}))$，锐化公式为$p_{\tilde{\theta}}^{(sharp)}(y|x) = \frac{\exp(z_y/\tau)}{\sum_{y'}\exp(z_{y'}/\tau)}$，被证明可以提升模型性能。

### (3) MixText

Mixup [53,54]技术被应用于图像分类领域并取得不错效果。接受两个标记样本$(x_i, y_i)$和$(x_j, y_j)$，对其隐藏表示和标签进行线性插值获得融合样本$(\tilde{x}, \tilde{y})$用于训练，即$\tilde{x} = mix(x_i, x_j) = \lambda x_i + (1-\lambda)x_j$，$\tilde{y} = mix(y_i, y_j) = \lambda x_i + (1-\lambda)x_j$，其中$\lambda \sim \mathrm{Beta}(\alpha, \alpha)$且$\lambda = \max(\lambda, 1-\lambda)$，使得融合样本中$x_i$占主导地位。mixup 可视为对模型进行正则化，使得输入数据在嵌入空间中表现出线性关系，同时该方法也可产生额外的训练样本$(\tilde{x}, \tilde{y})$。

受图像分类领域的 mixup 技术的启发，Chen[55]等人提出 TMix 模型，并进一步提出半监督文本分类模型 MixText。和图像领域直接对输入进行插值不同，TMix 使用 Transformer 的前几层对文本进行向量编码，对编码向量和标签进行相同权重的插值，获得融合样本，将 mixup 技术应用于自然语言处理领域。

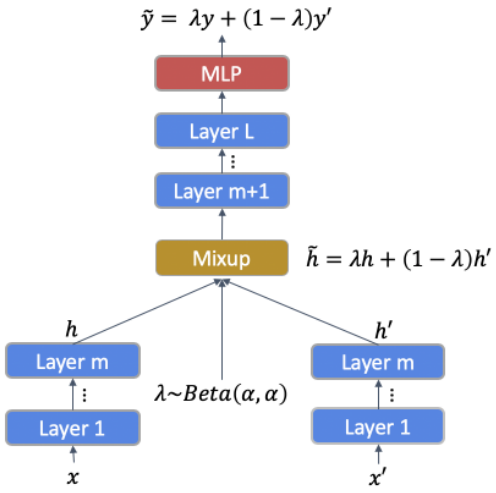


图 17. Mixup 技术的 NLP 领域应用——TMix[55]

MixText 对无标记样本进行一致性训练，和 UDA 一样通过回译获取无标记文本$x_i^u$的增强文本$x_{i,1}^a, x_{i,2}^a, \ldots, x_{i,K}^a$，对增强文本$x_{i,1}^a, x_{i,2}^a, \ldots, x_{i,K}^a$连同原文本$x_i^u$一起输入分类器中获得多个预测概率分布，加权求和得到$y_i^u = \frac{1}{w_{ori}+\sum_k w_k}(w_{ori}p(x_i^u) + \sum_{k=1}^K w_k p(x_{i,k}^a))$，对之进行锐化获得低熵的一致性标记$y_i^u = \text{Sharpen}(y_i^u, \text{T}) = \frac{(y_i^u)^{\frac{1}{T}}}{\|(y_i^u)^{\frac{1}{T}}\|_1}$，作为$x_i^u$及其增强文本$x_{i,1}^a, x_{i,2}^a, \ldots, x_{i,K}^a$的伪标记，将标记文本集合$X_l$、无标记文本集合$X_u = \{x_i^u\}$、增强文本集合$X_a = \{x_{i,k}^a\}$组成训练文本集$X = X_l \cup X_u \cup X_a$，一同使用 TMix 进行训练。对于$x, x' \in X$，若$x \in X_l$模型只计算$x$的交叉熵损失$L_{TMix} = CE(y, x)$，若$x \in X_u \cup X_a$则使用 KL 散度计算一致性损失$L_{TMix} = E_{x,x' \in X} KL(mix(y, y') \,||\, p(TMix(x, x'))$。此外，为鼓励模型在无标记样本上产生更可信的伪标记，加入熵最小化损失$L_{margin} = E_{x \in X_u} \max(0, \gamma - ||y^u||_2^2)$，模型总损失为$L_{MixText} = L_{TMix} + \gamma_m L_{margin}$。

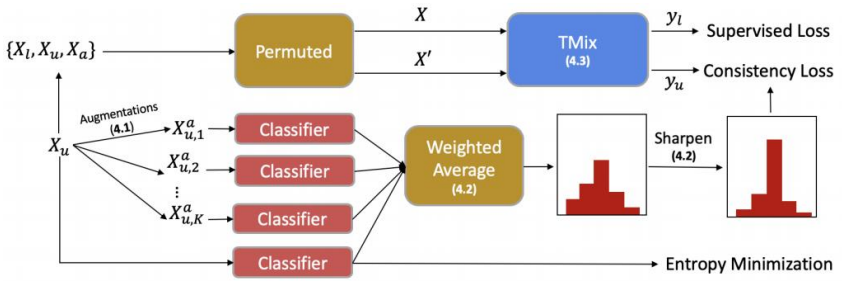


图 18. MixText 模型[55]

相比于 UDA，MixText 中的一致性损失一次考虑进了更多的增强样本，并且应用了 mixup 混合有标记和无标记样本，使得二者之间产生了信息流动，在半监督文本分类任务上获得了比 UDA 更强的性能。

### 4.4. 小结

面向少量标记样本的半监督文本分类，使用无标记样本或增强样本提供额外的语言学知识，根据训练方式的不同，面向少量标记样本的文本分类方法可分为（1）对无标记样本进行预训练获得语言学知识，再在少量标记数据上微调的两阶段方法，预训练后的模型可作为文本编码器进行复用。若有足量的领域内数据，可继续语言模型的预训练，能提升相关领域的模型性能。（2）使用少量标记数据训练初始分类器，获取无标记数据的伪标记，加入训练集中以迭代的方式自训练，训练数据的增加可以带来性能上的提升，但也会受到错误累积问题的影响。（3）以数据增强或噪声注入等方式获得增强样本，训练分类器输出一致分类的一致性训练方法，能够平滑分类界面增加模型泛化性能。

# 第五章 前沿问题讨论和研究方向

在深度学习模型（如循环神经网络、卷积神经网络、注意力、预训练语言模型等）的帮助下，文本分类任务获得了巨大的进步，但仍然存在着众多的问题和挑战。一个主要挑战是面向少量标记样本的文本分类问题，获取标记样本的成本和分类模型的性能之间存在着矛盾，现今主要有预训练、自训练和一致性训练三类研究方法。未来的少量标记样本情况下的文本分类，挖掘困难样本、数据增强方式是值得研究的方向。

## 5.1. 困难样本挖掘

少量标记样本情况下分类的困难在于，少量标记样本不足以提供泛化性好的分类边界，究其原因是样本量过少且样本不具有困难性（离分类边界较远），引入无标记样本和生成增强样本都是为了平滑分类边界以增强其泛化性。基于此，挖掘困难样本供分类模型学习，将有利于分类边界的确定。在少量标记样本的自训练方法中，往往选择高置信样本作为伪标记样本加入训练集进行迭代训练，但高置信的样本往往不具有困难性，给模型提供的界定分类边界的信息较少，研究在扩充训练集时加入困难样本，如 Delta-training[46]的方式，利于提升分类效果。另外，相比于随机选取初始训练集，让初始训练集包含更多困难样本也有利于后续界定分类边界。

## 5.2. 数据增强方式

对文本数据进行增强，加以一致性训练使得增强数据和源数据同属一类，可以增强分类边界的泛化性，是少量标记样本文本分类值得努力的研究方向。由于文本的离散性，对文本的简单增强，如删除词、替换词等收益不高甚至有损模型性能。反向翻译的方式是较好的文本增强方式，但依赖于大量平行数据训练的翻译模型，一定程度上受到翻译质量的影响。噪声注入和 Mixup 方法在嵌入空间中进行文本增强，带来不错的效果提升。文本的数据增强方式，是值得今后面向少量标记样本文本分类问题的研究的方向。

# 参考文献


[1] M. E. Maron, “Automatic indexing: An experimental inquiry,” J. ACM, vol. 8, no. 3, pp. 404–417, 1961.

[2] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” in Proc. ECML, 1998, pp. 137–142, 1998.

[3] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015.Character-level convolutional networks for text classification. In Advances in neural information processing systems, pages 649–657.

[4] Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In Proceedings of the International Conference on Association for Computational Linguistics, ACL 2011, 142–150.

[5] Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of ACL’05.

[6] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of EMNLP.

[7] Yaslan, Y.; and Cataltepe, Z. 2010. Co-training with relevant random subspaces. Neurocomputing 73: 1652–1661.Zhang, X.; Zhao, J. J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. In Proceedings of the International Conference on Neural Information Processing Systems, NIPS 2015, 649–657.

[8] Mendes, P. N.; Jakob, M.; and Bizer, C. 2012. DBpedia: A Multilingual Cross-domain Knowledge Base. In Proceedings of the International Conference on Language Resources and Evaluation, LREC 2012, 1813–1817.

[9] X. Zhang, J. J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in Proc. NeurIPS, 2015, pp. 649–657, 2015.

[10] Z. Chen, H. Zhang, X. Zhang, and L. Zhao. 2018. Quora question pairs.

[11] Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning Distributed

Representations of Sentences from Unlabelled Data. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1367– 1377, San Diego, California. Association for Computational Linguistics.

[12] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1112–1122. Association for Computational Linguistics.

[13] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

[14] Mikolov, Tomas, et al. "Efficient Estimation of Word Representations in Vector Space." ICLR (Workshop Poster), 2013.

[15] Pennington, Jeffrey, et al. "Glove: Global Vectors for Word Representation." Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[16] Iyyer, Mohit, et al. "Deep Unordered Composition Rivals Syntactic Methods for Text Classification." Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, 2015, pp. 1681–1691.

[17] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long Short-Term Memory." Neural Computation, vol. 9, no. 8, 1997, pp. 1735–1780.

[18] Cho, Kyunghyun, et al. "Learning Phrase Representations Using RNN Encoder--Decoder for Statistical Machine Translation." Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1724–

1734.

[19] Tai, Kai Sheng, et al. “Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks.” Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, 2015, pp. 1556–1566.

[20] Liu, Pengfei, et al. “Recurrent Neural Network for Text Classification with Multi-Task Learning.” IJCAI’16 Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, pp. 2873–2879.

[21] Dieng, Adji B., et al. “TopicRNN: A Recurrent Neural Network with Long-Range Semantic Dependency.” ICLR (Poster), 2016.

[22] Wang, Yequan, et al. “Sentiment Analysis by Capsules.” Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1165–1174.

[23] Conneau, Alexis, et al. “Supervised Learning of Universal Sentence Representations from Natural Language Inference Data.” Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 670–680.

[24] Miyato, Takeru, et al. “Adversarial Training Methods for Semi-Supervised Text Classification.” ICLR (Poster), 2017.

[25] Kim, Yoon. “Convolutional Neural Networks for Sentence Classification.” Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1746–1751.

[26] Johnson, R., & Zhang, T. (2015). Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. To Appear: NAACL-2015, (2011).

[27] Johnson, R., & Zhang, T. (2015). Semi-supervised Convolutional Neural Networks for Text Categorization via Region Embedding.

[28] Johnson, Rie, and Tong Zhang. “Deep Pyramid Convolutional Neural Networks for Text Categorization.” Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) , vol. 1, 2017,

pp. 562–570.

[29] Zhou, Peng, et al. “Text Classification Improved by Integrating Bidirectional LSTM with Two-Dimensional Max Pooling.” Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 3485–3495.

[30] Bahdanau, Dzmitry, et al. “Neural Machine Translation by Jointly Learning to Align and Translate.” ICLR 2015 : International Conference on Learning Representations 2015, 2015.

[31] Yang, Zichao, et al. “Hierarchical Attention Networks for Document Classification.” 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference, 2016, pp. 1480–1489.

[32] Wang, Yequan, et al. “Attention-Based LSTM for Aspect-Level Sentiment Classification.” Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 606–615.

[33] Lin, Zhouhan, et al. “A Structured Self-Attentive Sentence Embedding.” ICLR (Poster), 2017.

[34] Vaswani, Ashish, et al. “Attention Is All You Need.” Proceedings of the 31st International Conference on Neural Information Processing Systems, vol. 30, 2017, pp. 5998–6008.

[35] Peters, Matthew E., et al. “Deep Contextualized Word Representations.” Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), vol. 1, 2018, pp. 2227–2237.

[36] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, OpenAI.

[37] Devlin, Jacob, et al. “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.” Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human

Language Technologies, Volume 1 (Long and Short Papers), 2018, pp. 4171–4186.

[38] Reimers, Nils, and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks.” Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3980–3990.

[39] Liu, Yinhan, et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach.” ArXiv Preprint ArXiv:1907.11692, 2019.

[40] Lan, Zhenzhong, et al. “ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations.” ICLR 2020 : Eighth International Conference on Learning Representations, 2020.

[41] Yang, Zhilin, et al. “XLNet: Generalized Autoregressive Pretraining for Language Understanding.” Advances in Neural Information Processing Systems, vol. 32, 2019, pp. 5753–5763.

[42] Clark, Kevin, et al. “ELECTRA: Pre-Training Text Encoders as Discriminators Rather Than Generators.” ICLR 2020 : Eighth International Conference on Learning Representations, 2020.

[43] Sun, Zijun, et al. “Neural Semi-Supervised Learning for Text Classification Under Large-Scale Pretraining.” ArXiv Preprint ArXiv:2011.08626, 2020.

[44] Gururangan, Suchin, et al. “Variational Pretraining for Semi-Supervised Text Classification.” Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 5880–5894.

[45] Dong-Hyun Lee. 2013. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. ICML 2013 Workshop : Challenges in Representation Learning (WREPL).

[46] Jo, Hwiyeol, and Ceyda Cinarel. “Delta-Training: Simple Semi-Supervised Text Classification Using Pretrained Word Embeddings.” Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3456–3461.

[47] Lee, Ju Hyoung, et al. “SALNet: Semi-Supervised Few-Shot Text Classification with Attention-Based Lexicon Construction.” AAAI, 2021, pp. 13189–13197.

[48] Du, Jingfei, et al. “Self-Training Improves Pre-Training for Natural Language Understanding.” Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 5408–5418.

[49] Miyato, Takeru, et al. “Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning.” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 8, 2019, pp. 1979–1993.

[50] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In ICLR, 2015.

[51] Laine, Samuli Matias, and Timo Oskari Aila. “Temporal Ensembling for Semi-Supervised Learning.” ICLR (Poster), 2017.

[52] Xie, Qizhe, et al. “Unsupervised Data Augmentation for Consistency Training.” ArXiv Preprint ArXiv:1904.12848, 2019.

[53] Berthelot, David, et al. “MixMatch: A Holistic Approach to Semi-Supervised Learning.” Advances in Neural Information Processing Systems, vol. 32, 2019, pp. 5049–5059.

[54] Zhang, Hongyi, et al. “Mixup: Beyond Empirical Risk Minimization.” International Conference on Learning Representations, 2017.

[55] Chen, Jiaao, et al. “MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification.” Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 2147–2157.