

# 构建最小非意向偏差的有毒评论识别模型

姓名：吴凡璐

联系电话：13671295788

邮箱：wflcufe@126.com

**摘要：**为了更好地解决有毒评论分类问题，本文提出了一种 LSTM-CNN 模型：使用 LSTM 层捕捉句子的依赖关系，将之输入到 CNN 层中提取字节片段的局部特征后进行分类。为了解决样本不平衡和非意向偏差的问题，本文提出将子样本与总样本在假阳性率和假阴性率两项指标方面的差值作为衡量指标，结合样本的身份标签信息设计加权的损失函数作为训练目标。实验结果表明，使用加权损失函数的 LSTM-CNN 模型的 AUC 高达 0.98，比基准模型 NBSVM 高 0.01，召回率为 86%，比基准模型高约 6%，FNR 和 FPR 差值也远远小于基准模型，该模型不仅在识别有毒样本的准确程度上优于基准模型，还具有更好的公平性。

**关键词：**有毒评论；文本分类；非意向偏差；LSTM-CNN

## 一、研究背景

互联网发展二十多年以来，随着技术的不断演进，商业竞争日益激烈，内容运营成为影响线上业务发展竞争力的重要因素。而对于社交网站来说，评论管理是内容运营的核心，是影响用户粘性的重要工具。如今互联网上充斥着违法、侮辱、骚扰、色情、暴力等垃圾内容，这些内容被认为是“有毒”的，它们在网站中的扩散不利于营造友善的社交氛围，破坏用户体验。

目前很多知名企业都有自己的“有毒评论”过滤网，利用人工智能技术清理运营平台。社交应用 Instagram 公司在其产品中添加了“隐藏不当的评论”的按钮，可以帮助用户屏蔽一些 Instagram 挑选的敏感词；Facebook 公司开发了一套名为 DeepText 的自然语言处理系统，利用词嵌入的机器学习理论识别垃圾信息；谷歌反滥用技术团队推出名为“Perspective”的新工具，自动检测网络上的侮辱、骚扰和虐待言论，对输入的所有词句给出“毒性”评分。

尽管这些项目已经投入应用并取得一定成效，但是在使用它们的过程中也暴露了一些问题：由于算法无法准确理解文字背后的语境，当评论中出现某些经常

被攻击的词时，模型会误判其为“有毒”。如当评论中出现“gay”时，其被判为有毒评论的可能性急剧上升，因为含有“gay”的评论文本大多数包含歧视意味，但是实际上某句话只是在客观陈述而不是发表性向歧视，如“I am a gay woman”虽然无毒却常被误判；某些算法对含有“fag”一词的句子较为敏感，以为文本中出现“fag”就是男同性恋，虽然“fag”有同性恋的含义，但是在英国俚语里也指香烟。如何在找出有毒评论的同时减小过拟合带来的非意向偏差，是一个亟待解决的行业问题。

本文旨在构建一种具有较小误分偏差的有毒文本分类模型，为社交平台进行有效的内容管控提供新的工具。

## 二、 研究意义

阻击“有毒”内容，有两方面现实意义：一方面，互联网公司需要维持商业利益。谷歌 90% 的收入来自广告，但是没有广告主愿意自己投放广告的页面包含令人不适的内容，沃尔玛和百事可乐就因为不良内容而放弃在 YouTube 上推广，给谷歌带来了上亿美元的损失。充满戾气的社区环境也会驱散被攻击过的名人用户，让优质评论生产者不愿发声，影响产品的质量；另一方面，互联网是社会媒体的重要组成部分，在信息化时代，互联网实际上就是自由言论的喉舌，深刻影响着人们的思想甚至引领变革。如何引导人们正确地使用自由言论的权利，尽可能减少偏见的伤害，同时又不会因为过于严格的规则造成误判限制言论多样性，营造友善而又活跃的交流氛围，是互联网公司的使命。

利用 AI 技术进行审核的意义有两点：一是降低人工审核的成本，提升工作效率。不断升级的恶意评论需要企业花费更多金钱雇佣审核人员，在 Google，工资是 15 美元每小时，比大多数城市的最低工资高；而 YouTube 用户每天会上传近 60 万小时的视频，需要大量审核人员不眠不休地工作，这种工作量会迫使他们把数量和速度放在准确度之上，有时候检查几小时的视频只用了不到 2 分钟；第二，机器不会代入审核员固有的情感和偏见，人类饱受偏见和矛盾之苦，而电脑没有感情，在规则无误的情况下，可以更客观地做出判断。

## 三、 文献综述

今天，互联网已经成为文本的主要来源，社会机构 80% 的数据都会以电子文档的形式存储，越来越易得的大量文本数据的重要性也在不断凸显。从这些非结构化的文本“大数据”中进行自动化的文本挖掘和知识抽取是辅助商业决策的重要工具<sup>[1]</sup>。其中，文本分类是文本挖掘的重要一环。Vandana Korde（2012）等指出，文本分类的标准流程由以下 6 部分组成：文本收集、文本预处理（分词和词干提取）、文本索引、特征提取、分类和表现评估<sup>[2]</sup>；在有毒文本分类领域，Pooja Parekh 和 Hetal Patel（2015）介绍了在学术界识别有毒评论常用的机器学习方法：逻辑回归、支持向量机、树方法、神经网络（CBOW）等，并分析了每种方法的局限性，同时介绍了业界巨头：谷歌公司和雅虎公司构造的有毒评论识别工具的基本信息和其应用瓶颈，指出了识别有毒评论的挑战：语义模糊和精确度不足的问题<sup>[3]</sup>；Isuru Gunasekara 和 Isar Nejadgholi（2018）则强调了在文本领域，选择合适的文本表征的重要性，他们认为字符级别的文本表征比词级别的文本表征更胜一筹，即使是 SVM 这样的传统机器学习模型也可以因此而获得较好的表现，RNN 以及 LSTM 等神经网络模型可以略微提升 AUC；而 Fahim Mohammad 则对文本预处理的作用提出了质疑，不同于以往研究，作者进行了文本预处理对模型性能提升作用影响的实验，文中尝试了 35 种文本预处理方法，在 4 种表现较好的经典模型上进行测试，结果显示，大部分文本转换收效甚微，甚至对于某些模型来说，文本转换还会带来精确率的下降，因此不鼓励在文本预处理上花费过多精力<sup>[4]</sup>；在选择线性分类器还是神经网络的问题上，Yoav Goldberg（2015）认为神经网络通常情况下都会取得比线性分类器更好的效果，尤其是在使用了预训练的词向量的基础上<sup>[5]</sup>。

本文将在以上文献的基础上进行研究。

## 四、 研究目标和研究内容

本文的研究立足于 kaggle 正在举办的比赛：Jigsaw Unintended Bias in Toxicity Classification。该比赛由 Conversation AI 团队发起，旨在寻求用机器学习方法识别有毒评论并降低由于身份标签带来的误判，即当文本中出现有关种族、性别、地域等特征时，一般的机器学习算法会倾向于判定其有毒，因为这些特征容易出现在有毒文本中，但是这种过拟合也会造成对原本无毒的文本的误判。因此，本

文的目标不仅是建立能有效识别有毒文本的分类模型，也要利用每条评论的身份标签信息减小非意向偏差。

本文从两个角度尝试构建文本分类模型：传统机器学习方法和神经网络方法。使用抽取文本词频信息的 NBSVM 分类模型作为基准，然后构建使用预训练 Glove 词向量的 BLSTM-CNN 模型对评论进行分类，在深度学习模型的训练过程中，利用身份信息对损失函数的权重进行重新构造，使模型在对有身份信息的评论上出现较少误判，最后比较其与基准模型在测试集在各项评估指标上的差异。

## 五、数据来源和数据描述

本文的数据来源于 Conversation AI 团队提供的网络评论英文文本数据<sup>1</sup>。训练集中共有 1804874 条评论样本，45 个变量。这些变量中不仅包含每个评论样本的文本和毒性评分，还有一些身份属性变量，用于表示该文本中出现某个身份特征的可能性。具体描述见下表：

表 1：数据变量描述

变量名	数据类型和范围	变量描述
comment_text	字符串：2-999 字符	评论文本
target	浮点型：0.0-1.0	评论的有毒程度
female	浮点型：0.0-1.0	评论中提及女性的置信度
homosexual_gay_or_lesbian	浮点型：0.0-1.0	评论中提及同性恋的置信度
jewish	浮点型：0.0-1.0	评论中提及犹太人的置信度
muslim	浮点型：0.0-1.0	评论中提及穆斯林的置信度
black	浮点型：0.0-1.0	评论中提及黑人的置信度
psychiatric_or_mental_illness	浮点型：0.0-1.0	评论中提及心理疾病的置信度

注：由于篇幅所限只列举 5 个身份属性变量，其中的“置信度”是指为文本标注的所有人员中，认为文本中包含该属性的比例。

例如：评论 “i'm a white woman in my late 60's and believe me, they are not too crazy about me either!!” 的 target 为 0.0，female 标签为 1.0，white 标签为 1.0，表示其是一条包含女性白人身份信息的无毒评论。

<sup>1</sup> <https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/data>

## 六、探索性分析

### （一）毒性分布

首先对训练集中样本的毒性分布进行统计，将浮点型毒性变量处理成二分变量：毒性超过 0.5 的认为该评论有毒，毒性小于 0.5 的评论认为其无毒。处理后发现，训练集中 92.18% 的样本是无毒评论，只有 7.82% 的样本是有毒评论，存在严重的样本不平衡问题。

### （二）评论文本的长度分布

首先我们对评论文本的长度进行探索性分析，了解长文本和短文本在评论语料中的分布。

#### 1. 字符长度分布

如图是展示每条评论字符长度分布情况的直方图，最短评论有 2 个字符，最长评论为 999 个字符。从图中可以看出，绝大多数文本的长度在 255 字符之内，是短文本。在字符长度为 1000 左右出现了一个小高峰，说明有也一定数量的文本是较长的文本。

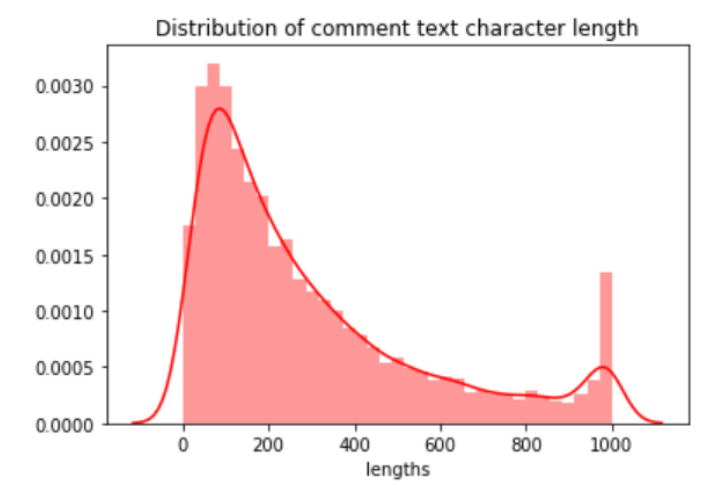


图 1：文本长度分布直方图

#### 2. 单词个数分布

如图是展示每条评论单词个数分布情况的直方图，最短评论只有 1 个单词，最长评论有 195 个单词。从图中可以看出，评论的单词个数呈现出明显的右偏分布，绝大多数评论为 60 词以下的短评论，少部分评论是 140 词以上的长评论。

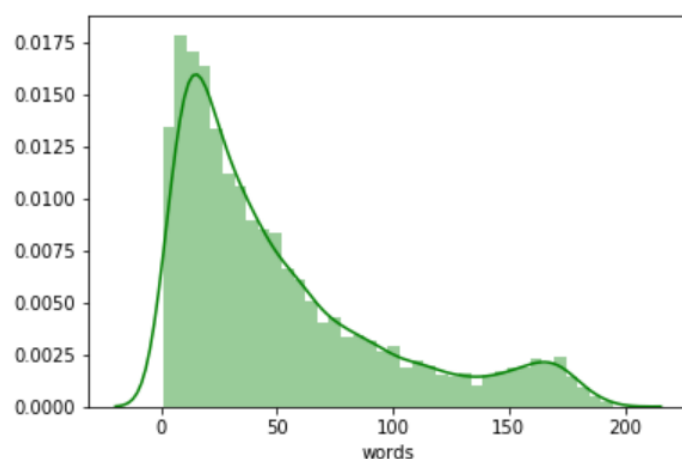


图 2: 单词个数直方图

### (三) 评论长度和毒性分布的关系

某些文本分类算法中会将评论的长度作为特征之一添加到模型中,为了探究评论长度是否和评论有毒无毒有关,使用统计图和表格进行描述。处理时,认为毒性大于等于 0.5 即认为是有毒文本,标记为 1; 小于 0.5 为无毒文本,标记为 0。将评论从字符长度和单词个数两个方面进行长短文本的划分,如表格所示:

表 2: 长短文本划分标准

文本类别	字符长度	单词个数
短文本	0-255	0-50
中文本	256-700	51-100
长文本	701-1000	101-200

#### 1. 评论字符长度和毒性的关系

下图是不同类型的文本中毒性的分布,由柱形图可见,不同类型的文本中有毒文本和无毒文本的比例基本保持一致,均接近 0.92: 0.08。

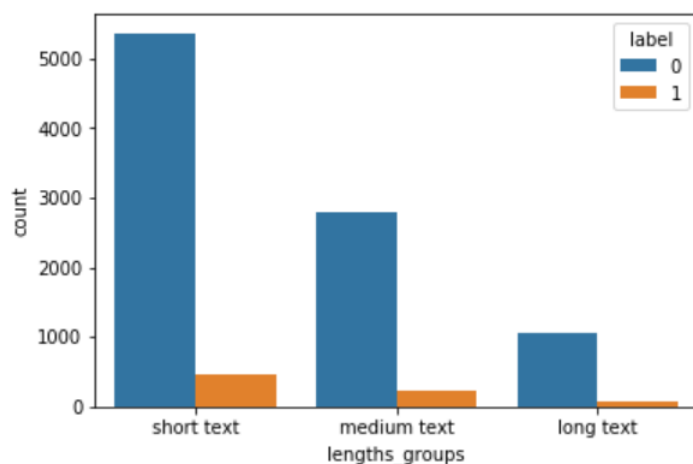


图 3：不同字符长度毒性分布柱形图

## 2. 评论单词个数和毒性的关系

下图是不同长度类型的文本中毒性分布的柱形图，由图可见，不同类型的文本中有毒文本和无毒文本的比例也基本保持一致，均接近 0.92：0.08。

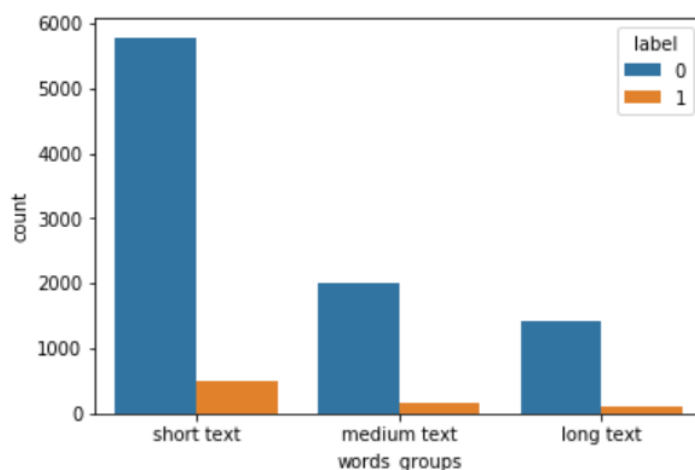


图 4：不同单词长度毒性分布柱形图

### （四）评论内容词云图

为了了解有毒评论和无毒评论的大致内容，我们绘制了反映词频信息的词云图。如图可见，有毒评论和无毒评论中出现最多的都是“Trump”和“People”，说明网络评论对时事政治关心。相比较无毒评论中出现的都是没有情感倾向的常用词，有毒评论中除了出现“stupid”和“idiot”这样的攻击性词汇外，还出现了反映种族信息的“black”和反映性别信息的“woman”。

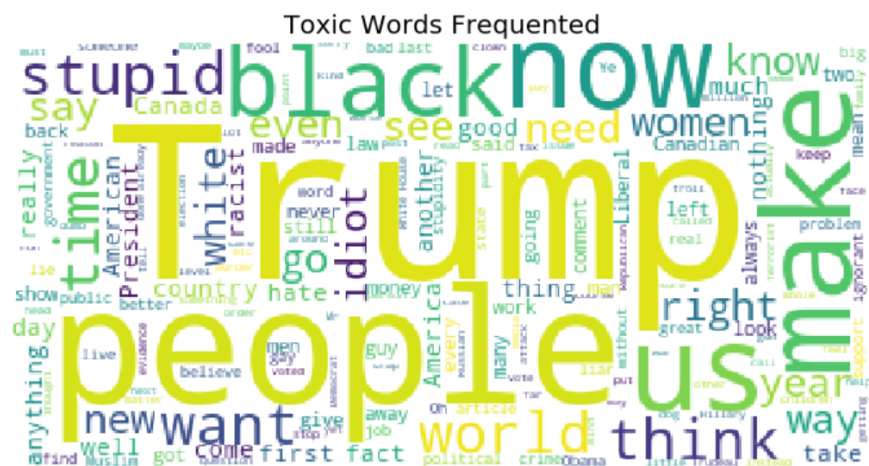


图 5: 有毒评论词云图



图 6: 无毒评论词云图

### （五）身份标签和毒性的关系

从词云图可以看出，有毒评论会包含一些反映身份标签的信息。为了探究当评论中出现身份信息时，评论是否更容易被认为是有毒的，我们绘制了不同身份变量下的毒性分布的柱状图。观察身份变量的含义后发现，可以从种族、宗教、性别等六个维度对 24 个变量进行分组介绍，分组具体如下：

表 3: 变量分组说明表

组名	种类含义	变量名
ethnics	种族	'asian' , 'latino' , 'black' , 'white', 'other_race_or_ethnicity'
religions	宗教	'atheist', 'buddhist', 'hindu', 'jewish', 'muslim', 'christian', 'other_religion'
sexual	性别	'female', 'male', 'other_gender'
sexual_orientation	性取向	'heterosexual', 'bisexual', 'transgender',



disabilities	疾病	'homosexual_gay_or_lesbian', 'other_sexual_orientation' 'intellectual_or_learning_disability', 'physical_disability', 'psychiatric_or_mental_illness', 'other_disability'
--------------	----	---

下图是包含不同种族标签的毒性分布，由图可见，包含黑人标签的评论有毒的比例较大。

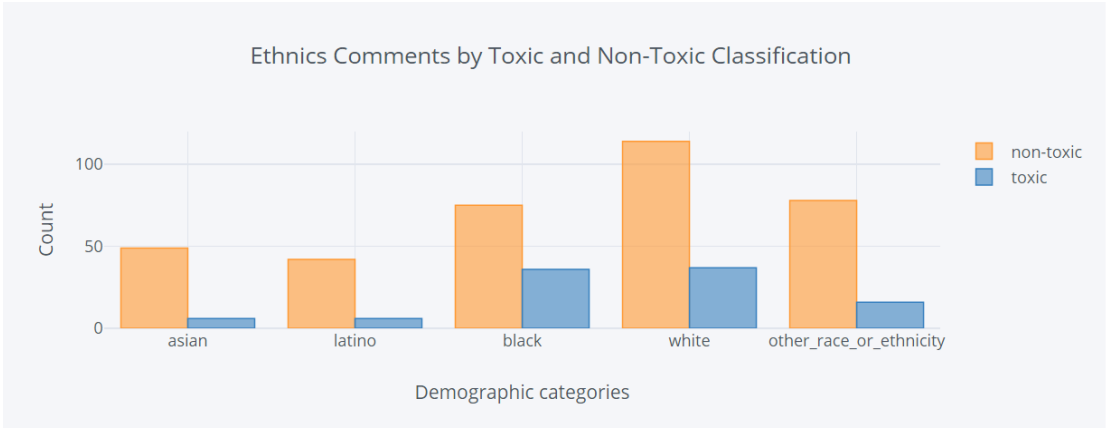


图 7：种族毒性分布柱形图

下图是反映疾病与毒性的柱状图，由图可见，在所有反映疾病的标签中，只有当评论中出现了智力或学习障碍的身份标签时，样本有毒的可能性会超过无毒的可能性，智力缺陷的人更容易在网络评论中受到攻击。

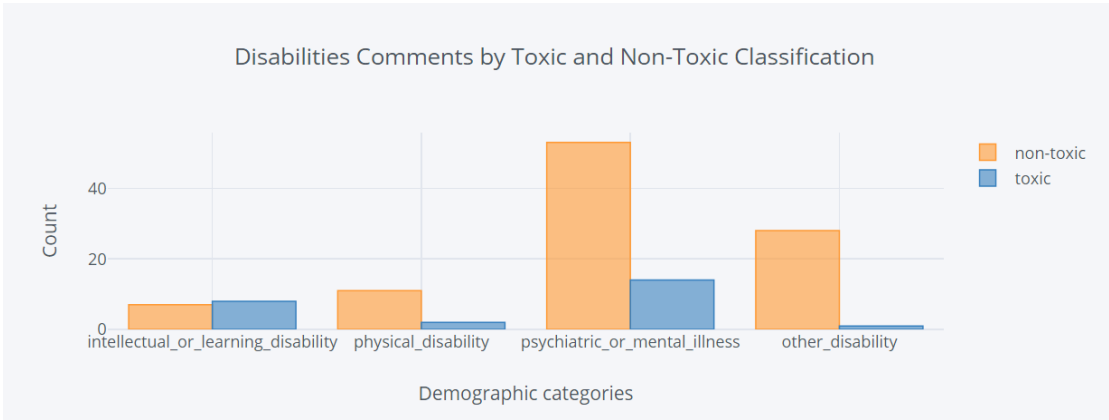


图 8：疾病毒性分布柱形图

下图是不同性取向与毒性的分布状况。当评论中出现了反映异性恋的信息时，评论更可能是有毒的，其次有毒程度较高的标签是同性恋。

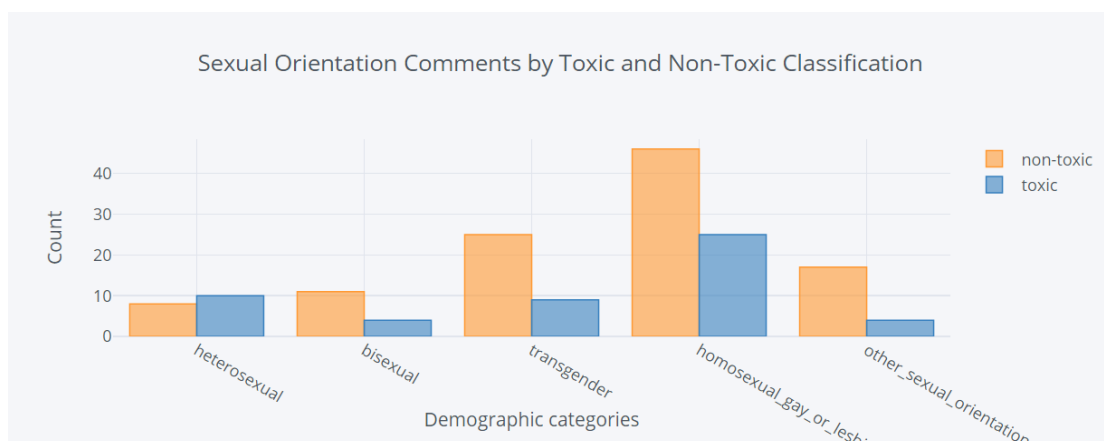


图 9：性取向毒性分布柱形图

当评论中出现了性别信息时，可以发现不同性别受到攻击的可能性相近。

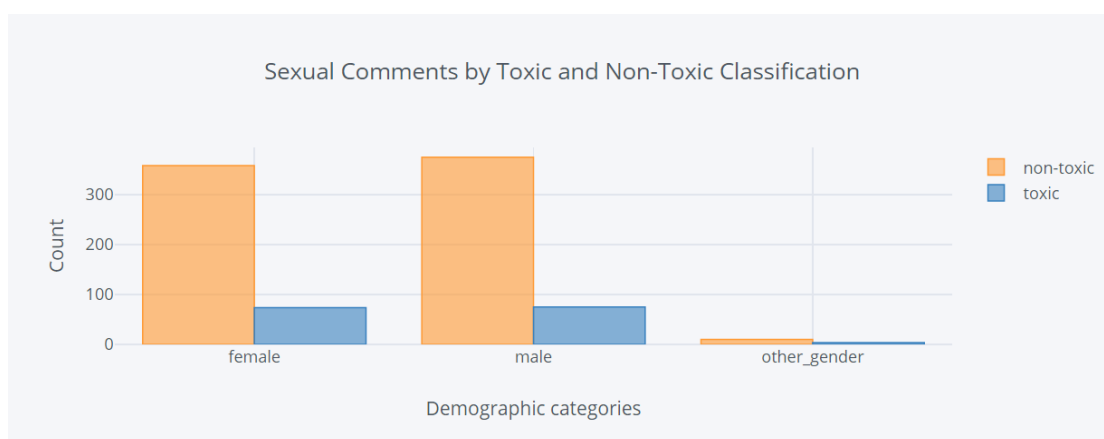


图 10：性别评论毒性分布柱形图

如下图所示，提及基督教，穆斯林教和犹太教的样本较多，其中，包含穆斯林教标签的评论有毒评论的比例是最多的。

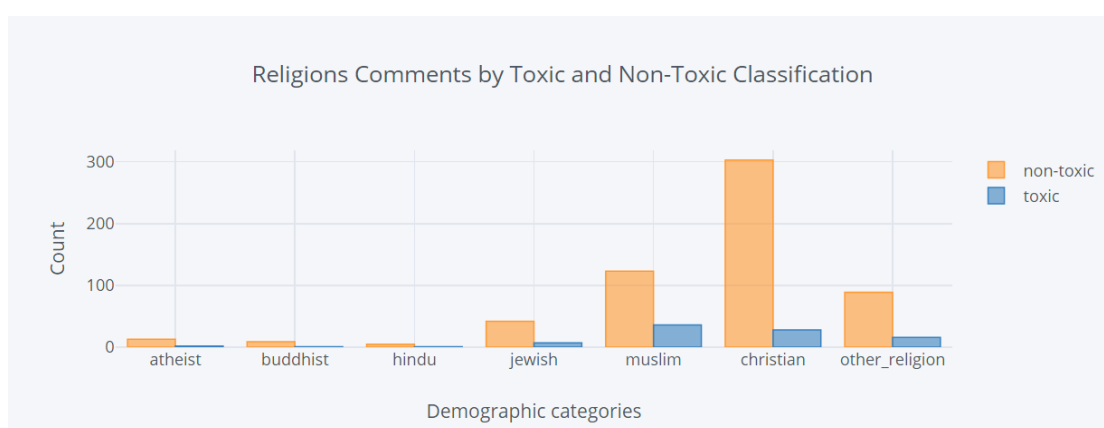


图 11：宗教毒性分布柱形图

为了从数字上更清晰地看出不同标签具体的有毒评论比例，我们计算了不同身份标签的评论中有毒的比例，展示了毒性样本比例高的前 10 个身份标签。如

表所示，反映性取向的评论毒性比例最大，智力缺陷标签的有毒比例也超过了一半，其次是种族和宗教。

表 4：不同身份标签有毒比例

身份标签	有毒比例
heterosexual	0.555556
intellectual_or_learning_disability	0.533333
sexual_explicit	0.408978
homosexual_gay_or_lesbian	0.352113
black	0.324324
other_gender	0.285714
bisexual	0.266667
transgender	0.264706
white	0.245033
muslim	0.226415

### 七、 基准模型——NBSVM

本文使用较为流行的 NBSVM 模型作为基准模型，该模型属于传统的机器学习模型，通过 TF-IDF 提取特征并进行分类。实验步骤如下：

#### （一）数据预处理

由于评论中的文本包含大量噪音，除了字符信息外，还有一部分特殊符号和拼写错误的词语，这些都会对特征提取形成干扰，因此在建模之前，需要对文本进行预处理：

1. 小写转化：将文本中所有的单词进行小写处理归一化，使得“Trump”和“trump”这样的词汇都会指向同一个人，但是也会存在信息丢失的问题，如“Bush”和“bush”指的可能并不是同一个事物。
2. 缩写处理：英文文本中存在大量的缩写短语，如“We are”会缩写成“We’re”。本文制作了缩写词和完整词的映射词典，将文本中所有的缩写词汇都还原成完整词。
3. 拼写改错：由于网络本文的不规范性，文本中会出现一些错误拼写的词汇。使用 textblob 模块的 correct 函数自动对文本进行拼写改错。
4. 清理特殊符号：文本中会出现一些如“🍔🇺🇸🍁🌂🍆🍑🥒🍌🐼👤\u200dEzKLWj”这样的表情和特殊字符，本文将之全部清理。

5. 删除停用词：英文中有一些如“a”，“the”这样在所有文档中都会出现的高频词，是文本中的冗余信息，本文使用 nltk 库中停用词词典对所有文本进行了去停用词处理。

## （二）数据集分割

由于数据集较大，本文将数据集以 95:5 的比例分割成训练集和验证集，抽取尽可能多的数据集进行训练的同时也能保证验证集数量的充足。

## （三）特征提取

在进行本文分类之前，我们需要用词袋模型把待分类的文本进行向量化表示，每篇文本的特征即是文本中的关键词，每个词的权重就是特征的数值。我们通过 TF-IDF 指标找出文本的关键词，提取每条评论的特征。步骤大致为：

### （1）计算词频 TF

$TF = \text{某个词在文章中的出现次数} / \text{文章的总词数}$

### （2）计算逆文档频率 IDF

$IDF = \log(\text{语料库的文档总数} / \text{包含该词的文档数} + 1)$

### （3）计算 TF-IDF

$TF-IDF = \text{词频 (TF)} * \text{逆文档频率 (IDF)}$

本文使用 python 中的 sklearn.feature\_extraction 包进行特征提取。

## （四）数据增强

由于样本存在严重的不平衡问题，负样本与正样本数量比接近 10:1，需要进行数据增强处理<sup>[6]</sup>。数据增强方法参考 Jason W. Wei,和 Kai Zou（2019）提出的方法<sup>[7]</sup>，具体由以下四个部分组成：

1. 同义词替换（SR: Synonyms Replace）：在句子非停用词的词语中随机抽取 n 个词，然后从同义词词典中随机抽取同义词，并进行替换。

2. 随机插入(RI: Randomly Insert)：在句子非停用词的词语中，随机抽取一个词，然后在该词的同义词集合中随机选择一个，插入原句子中的随机位置。该过程可以重复 n 次。

3. 随机交换(RS: Randomly Swap)：在句子中随机选择两个词交换位置。该过程可以重复 n 次。

4. 随机删除(RD: Randomly Delete): 将句子中的每个词以概率  $p$  随机删除。

用以上方法，我们生成了更多的正样本来纠正不平衡倾向，最后得到的负样本数：正样本数为 3:1。

### (五) 建立模型

朴素贝叶斯和支持向量机模型广泛应用于文本分类任务中，被用作基准模型。然而它们的表现很大程度上依赖于提取的特征和数据集。Sida Wang 和 Christopher D. Manning (2012) 将这两种分类器结合起来，提出更稳健的模型 NBSVM: 用朴素贝叶斯对数计数比率作为特征值的 SVM<sup>[8]</sup>。本文使用这种方法作为 baseline。模型的基本思想如下：

对于测试样本

$$y^k = \text{sign}(\mathbf{w}^T \mathbf{x}^{(k)} + b)$$

对于训练样本：

每个样本  $i$  输入的特征  $\mathbf{f}^{(i)} \in \mathbf{R}^{|V|}$ ， $V$  是特征集合， $y^{(i)} \in \{-1, 1\}$ ， $\mathbf{f}_j^{(i)}$  是样本  $i$  的第  $j$  个特征。定义计数向量  $\mathbf{p}$ ， $\mathbf{q}$  为：

$$\begin{aligned}\mathbf{p} &= \alpha + \sum_{i: y^{(i)}=1} \mathbf{f}^{(i)} \\ \mathbf{q} &= \alpha + \sum_{i: y^{(i)}=-1} \mathbf{f}^{(i)}\end{aligned}$$

其中  $\alpha$  是平滑系数，由此定义对数计数比例为：

$$\mathbf{r} = \log\left(\frac{\mathbf{p} / \|\mathbf{p}\|_1}{\mathbf{q} / \|\mathbf{q}\|_1}\right)$$

将样本特征向量与对数计数比例向量相乘得到  $\tilde{\mathbf{f}}^{(i)} = \mathbf{r} \circ \mathbf{f}^{(i)}$ ，将之作为输入向量  $\mathbf{x}^{(i)} = \tilde{\mathbf{f}}^{(i)}$ 。用 SVM 模型进行如下优化得到  $\mathbf{W}, b$ ：

$$\mathbf{W}^T \mathbf{W} + C \sum_i \max(0, 1 - y^{(i)} (\mathbf{W}^T \tilde{\mathbf{f}}^{(i)} + b))^2$$

### (六) 模型评估

#### 1. 分类性能评估

用 NBSVM 模型对验证集进行分类后得到，该模型判断评论是否有毒的总体准确率为 92.29%，召回率为 79.43%，F1 值为 0.85，标准化混淆矩阵如下所示：

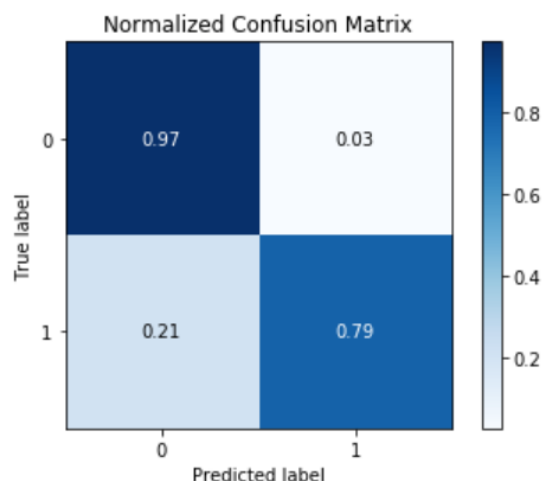


图 12: NBSVM 混淆矩阵

如图所示，该模型误判无毒样本的可能性是 3%，对于有毒样本，该模型能从中识别出 79% 的样本。

下图是模型的 ROC 曲线图，由图可见，ROC 曲线高于对角线且接近左上角，说明预测结果远远好于随机猜想，AUC 值为 0.97，模型总体分类性能较好。

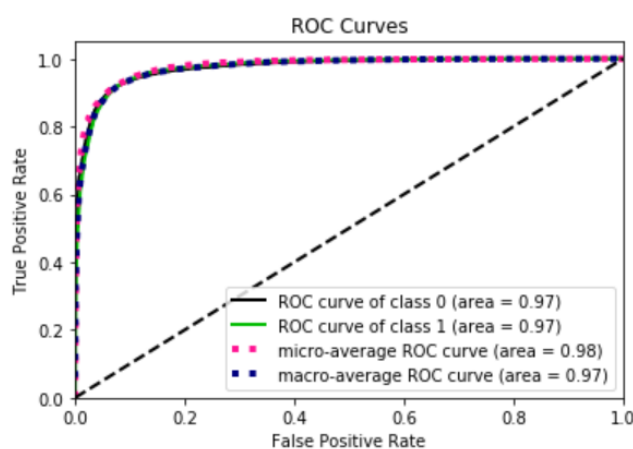


图 13: NBSVM ROC 曲线

## 2. 公平性评估

本文不仅要解决有效识别有毒评论的问题，还要将因为无毒评论文本中出现了身份信息带来的有毒误判几率降到最低，以保证模型的公平性。因此，我们参考了 Lucas Dixon 等（2017）探讨消除非意向偏差的论文<sup>[9]</sup>，使用其中设计的两项指标来对模型的公平性进行评估，其总体思想是：如果身份信息对毒性判断没有干扰，那么有身份信息样本的 FPR 和 FNR 应该和总体的 FPR 和 FNR 相差不大，因此可以通过衡量总样本和子样本的 FPR 和 FNR 的差值来判断是

否出现了非意向偏差，差距越大，非意向偏差越大，文中设定的差距阈值为 50%。衡量指标的具体描述如下：

$$\begin{aligned} \text{False Positive Equality Difference} &= \sum_{t \in T} |FPR - FPR_t| \\ \text{False Negative Equality Difference} &= \sum_{t \in T} |FNR - FNR_t| \end{aligned}$$

$t$  是不同的身份标签， $FPR_t$  表示该身份标签下的假阳性率， $FPR$  表示总的假阳性率， $FNR_t$  表示该身份标签下的假阴性率， $FNR$  表示总的假阴性率。

在 NBSVM 模型中，有标签的样本 FPR 值比总体高了 5.823%，FNR 值比总体低 3.864%。

## 八、深度学习模型——BLSTM-CNN

深度学习模型在自然语言处理领域总是能取得比普通线性分类器更好的表现，尤其是在使用预训练的词向量时，这一点尤为突出<sup>[5]</sup>。在文本分类问题上常用的模型结构有 RecNN、RNN、CNN 和其他神经网络四种<sup>[10]</sup>，本文将 RNN 结构与 CNN 结构相结合，提出一种名为 BLSTM-CNN 的深度学习模型。

### （一）预处理

深度学习模型的输入是反映词汇语义的词向量，词向量的获取方法有两种：自己训练生成或直接使用公开预训练词向量。本文使用斯坦福大学自然语言处理实验室提供的 glove.840B.300d 词向量作为预训练词向量，该词向量包含了常用的 2196008 个英文词汇和特殊符号。

对于使用预训练词向量深度学习来说，文本预处理有特点是：不需要进行复杂的预处理工作，适用于线性分类器的小写转化、去停用词和词干提取等处理反而有可能丢失重要的语义信息<sup>[7]</sup>，如“Go to work”是一个无毒评论，但是“GO TO WORK!!!!”却包含一定毒性。此时，充分利用预训练词向量中包含的大小写、表情和拉丁文等文本的向量信息尤为重要，因此需要将文本中的词和预训练中词向量的词进行对比，通过针对性的转化使预训练词向量中的词尽可能覆盖文本，降低袋外词（OOV）的比例，即文本的预处理要尽可能接近所选择的预训练词向量在训练时的预处理。因此，文本进行了如下操作：

1. 检测词向量中的词对原始文本的覆盖率：通过计算发现，词向量中的词只能

覆盖 15.82% 的原始本文词汇，能覆盖 89.63% 的原始文本。且不能被覆盖的高频袋外词为 “isn’t”, “That’s”, “won’t” 这样的常用缩写词汇，说明需要进一步处理。

2. 对部分高频缩写词进行还原，并且将词向量不能覆盖的特殊符号进行了清理，保留了 Glove 中包含的 ‘.,?!-;\*’:()%#\$&\_/@+=[]^>\\<~{}|’ 等特殊符号，按照 Glove 预处理的方法将两位数以上的数字用 “#” 代替。
3. 经过清理，词向量中词汇对文本词汇覆盖率提升到 54.41%，对文本覆盖率提升至 99.66%，袋外词按照频数从高到低前 6 名为：“tRump”，“gov’t”，“the globe and mail”，“Drumpf”，“deplorables”，“SB91”，皆为生僻词，出现的频数都在 2500 次以下，不需要再进一步处理。
4. 序列归一化。由于分词后的每条评论文本长度不一，需要统一成相同长度的文本序列。我们将对长文本进行截取处理，对短文本长度不足的地方以 0 填充，最终使得每条评论文本分词后的词序列长度为 220。

## （二）模型框架

本文提出了一种深度神经网络用于文本分类——BLSTM-CNN 模型。模型的结构如图 14 所示，分为 5 部分：Spatial dropout 层，BLSTM 层，Conv1D 层，全局池化层和输出层。

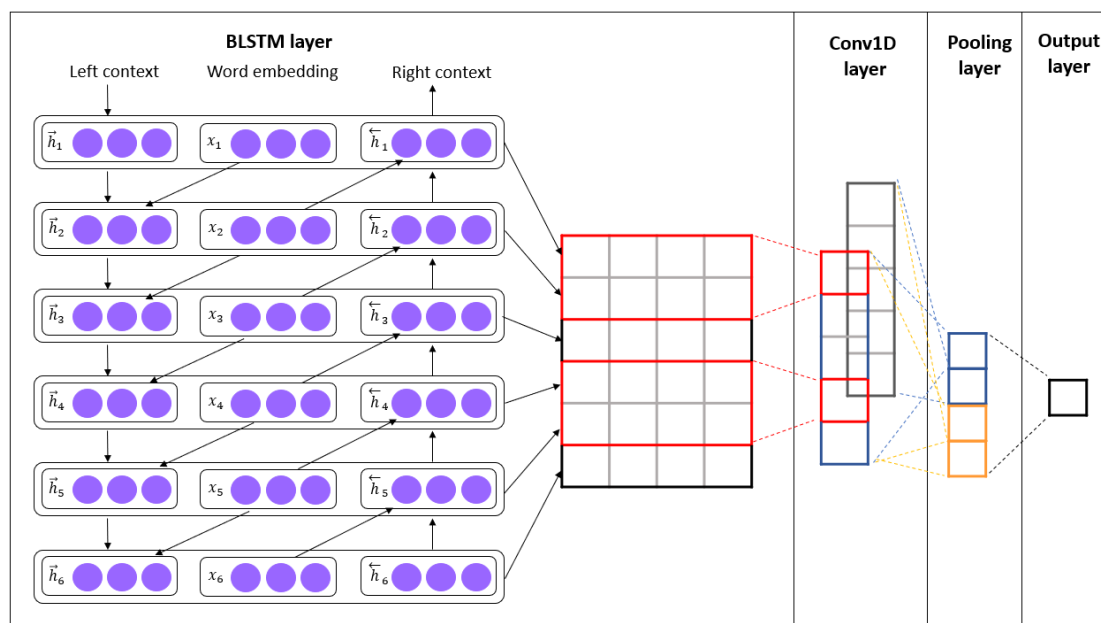


图 14: BLSTM-CNN 结构示意图 (输入序列长度为 6, word embedding 长度为 3, BLSTM 层有 4 个隐藏单元, 每个卷积核大小为 2\*4, 最大池化和平均池化算子大小为 5\*1)



## 1. Spatial dropout 层

对于每一个文本序列，都会被表示成固定大小的 Embedding 矩阵，在本文中，每条文本的 embedding 矩阵  $\mathbf{W}_E$  的大小为  $220 \times 300$ 。为了防止过拟合，文本对 embedding 矩阵进行一维 dropout 操作：通过随机地把  $\mathbf{W}_E$  中的某些行变成零向量引入噪音，使模型不会过多依赖于单个的词（A Theoretically Grounded Application of Dropout in Recurrent Neural Networks），最后输出的 embedding 矩阵和原 embedding 矩阵尺寸相同。

## 2. BLSTM 层

LSTM 最初是由 Hochreiter and Schmidhuber (1997) 提出的，它在 RNN 的基础上引入遗忘门学习句子中的长期依赖关系<sup>[11]</sup>。

给定序列  $S = \{x_1, x_2, \dots, x_l\}$ ， $l$  是给定文本的长度，LSTM 会从前往后学习每个词的信息。在时间  $t$ ，记忆门  $c_t$  和遗忘门  $h_t$  更新如下：

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ \hat{c}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} W \cdot [h_{t-1}, x_t]$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t$$
$$h_t = o_t \odot \tanh(c_t)$$

其中， $x_t$  是时间  $t$  的输入， $i, f, o$  分别是输入的输入门、遗忘门和输出门的激活函数， $c_t$  是当前单元的状态， $\sigma$  是逻辑斯蒂函数， $\odot$  代表元素对应相乘。

然而，在训练时，LSTM 只能学习之前的信息却不能学习之后的信息，为了克服这一弊病，Schuster and Paliwal (1997) 提出了 BLSTM——双向的 LSTM，通过增加反方向的序列隐藏层构建两个子网络，来同时学习序列过去和未来的信息。第  $i$  个单词的输出如下所示：

$$h_i = [\vec{h}_i \oplus \bar{h}_i]$$

$\oplus$  表示连接运算符，用来将前向输出和后向输出结合起来。

本文构建了一层 BLSTM 层，该层分为两个子层：前向层和后向层。每个子层有 64 个单元，将两个子层的输出拼接起来可得到 128 维的致密向量。通过

BLSTM 层，我们可以提取到在学习到序列的依赖关系后对文本序列进行重新编码，将新的序列表示输入到卷积层提取局部特征。

### 3. Conv1D 卷积层

卷积神经网络原本是用于计算机视觉领域的工具，但是近些年来也用于自然语言处理，在文本分类方面也发挥了独特功能：可用来提取 n-gram 级别的局部特征，而不考虑词语的位置信息<sup>[12]</sup>。

卷积层的输入是 BLSTM 层输出的重编码矩阵：

$H = \{h_1, h_2, \dots, h_l\}, H \in \mathbb{R}^{l \times d}$ ， $l$  是序列长度， $d$  是新的词嵌入向量的维度，本文中是 128。对序列进行一维卷积提取局部特征过程如下：有滤波器  $\mathbf{m} \in \mathbb{R}^{k \times d}$ ， $k$  是滤波器窗口大小，也是对序列提取字节片段的长度。特征  $o_i$  从向量窗口

$H_{i:i+k-1}$  中产生：

$$o_i = f(\mathbf{m} \cdot H_{i:i+k-1} + b)$$

$i$  的取值范围是 1 到  $(l-k+1)$ ， $\cdot$  表示点积， $b \in \mathbb{R}$ ，是一个常数项， $f$  是类似于双曲正切的非线性函数。当滤波器  $\mathbf{m}$  作用于矩阵  $H$  中所有的窗口时，就可以产生特征图  $O$ ：

$$O = [o_1, o_2, \dots, o_{l-k+1}]$$

$O \in \mathbb{R}^{(l-k+1)}$ ，以上是一个滤波器产生一个特征图的过程，如果有多个滤波器，可以学习多种特征图。

本文的卷积层使用 64 个滤波器，每个滤波器大小为 2，表示可以提取 Bi-gram 级别的局部特征。对于每个输入的序列来说，会产生 64 个特征图，每个特征图的长度为 255。

### 4. 池化层

池化层是跟在卷积层之后的，其目的是对卷积层输出进一步降维并提取显著特征。本文使用两种池化层来对特征图进行处理：

$$\begin{aligned} O_{\max} &= \max\{O\} \\ O_{\text{avg}} &= \text{avg}\{O\} \\ \hat{O} &= [O_{\max} \oplus O_{\text{avg}}] \end{aligned}$$

全局最大池化层可以提取特征图中最显著的特征作为整个特征图的的代表，全局平均池化层则衡量了某个主题在特征图上的总体分布情况。两种池化层都会输出一个长度为滤波器个数的向量，将最大池化输出向量和平均池化输出向量连接到一起的长向量作为全连接层的输出。

（全局池化层可以简化计算，往往加在神经网络最后。传统的全连接层相当于若干个和特征图一样大小的三维卷积得到的神经元，参数众多且容易过拟合，用全局 pooling 代替这种复杂的计算可以降低计算量的同时降低过拟合）

本文中，两个池化层的输出都是 64 维的向量，池化层总输出为 128 维的向量，代表从特征图中提取的特征。

## 5. 全局输出层

输出层是一个 sigmoid 函数，用来进行分类，对于样本  $s$  来说，若从池化层接受到的输入是  $\hat{O}$ ，对于二分类问题，最终输出是分类为正样本的概率为：

$$\hat{p}(y|s) = \text{sigmoid}(W^{(s)}\hat{O} + b^{(s)})$$

其中  $W^{(s)}$  是连接矩阵， $b^{(s)}$  是表示偏差的常数项。

### （三）加权损失函数

对于二分类问题，一般以最小化交叉熵损失作为训练目标，损失函数表示如下：

$$L = \sum_{i=1}^m -(y_i \log(p_i) + (1 - y_i) \log(1 - p_i))$$

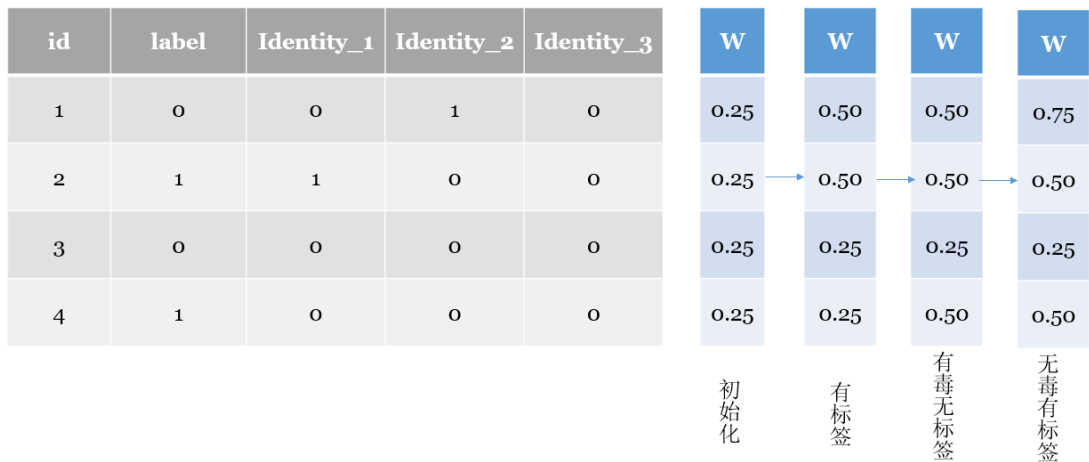
其中， $y_i$  是第  $i$  个样本真实的标签，正类为 1，负类为 0； $p_i$  是第  $i$  个样本被预测为正类的概率。

在上式中，每个样本对于损失函数的贡献是相等的，然而为了解决正负样本不平衡和非意向偏差的问题，我们需要对不同样本的损失在总损失中的权重做出调整，以得到分类性能更好且更公平的模型。因此，本文在一般的损失函数中引入样本权重，构造加权的损失函数作为优化目标。

$$\text{Weighted\_}L = \sum_{i=1}^m -w_i (y_i \log(p_i) + (1 - y_i) \log(1 - p_i))$$

$w_i$  为样本  $i$  的权重，其构造步骤如下：

1. 初始化：对于所有样本  $i=1,2,\cdots,m$ ，  $w_i=0.25$
2. 增加权重：对于有身份标签的样本，  $w_i+=0.25$
3. 增加权重：对于有毒而无身份标签的样本，  $w_i+=0.25$
4. 增加权重：对于无毒也无身份标签的样本，  $w_i+=0.25$
5. 归一化：计算所有样本的权重均值  $\bar{w}=\frac{1}{m}\sum_{i=1}^m w_i$ ，  $w_i/\bar{w}$

经过以上调整，无毒有身份标签的样本对于损失函数的贡献最大，其次是有毒样本，无毒无标签的样本对损失函数贡献最小，从而使得模型对容易产生误分偏差的样本（实际无毒而有身份标签）敏感度更高，将这类样本判为有毒的风险增大，从而朝着不容易产生偏差的方向优化；而在无毒样本中，有标签的比例仅为 5%，绝大多数无毒样本的权重比有毒样本权重低一半，增加了有毒样本损失对总损失的贡献，从而缓解了不平衡样本的问题。 以部分样本为例，演示了权重的确定过程。

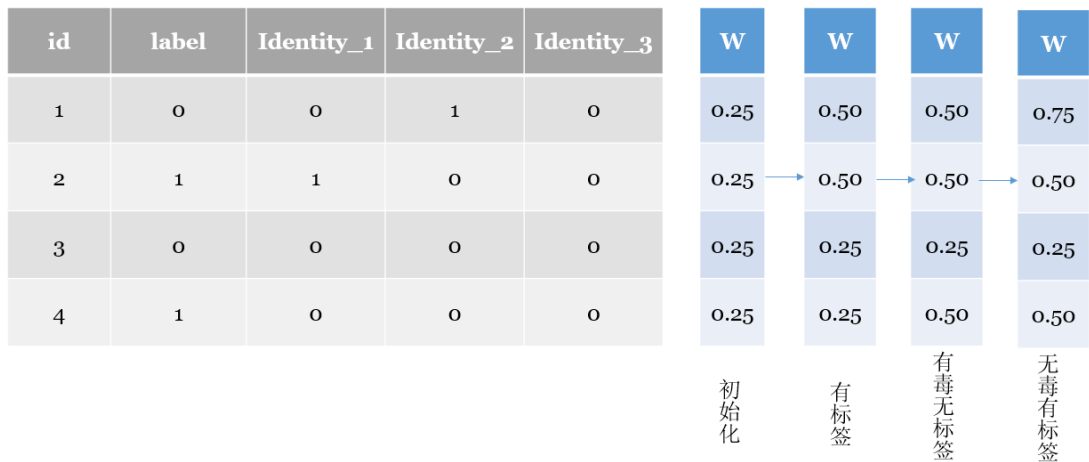


图 15：权重确定示意图

#### （四）模型结果与评估

##### 1. 分类性能评估

用深度学习模型对验证集进行分类后得到，该模型判断评论是否有毒的总体准确率为 97.46%，比基准模型高了 5.17%；召回率为 85.67%，比基准模型高 5.73%；F1 值为 0.82，比基准模型低 0.03。标准化混淆矩阵如下所示

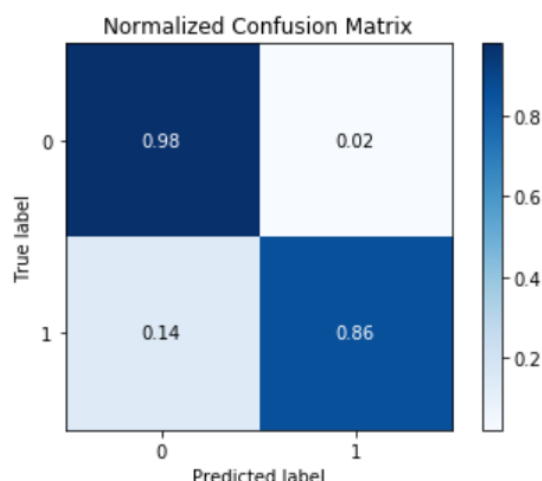


图 16: BLSTM-CNN 混淆矩阵

模型 AUC 为 0.98，比基准模型高 0.01，ROC 曲线如下图所示，可以看到各个子类的 ROC 曲线都靠近左上角，模型分类性能较好。

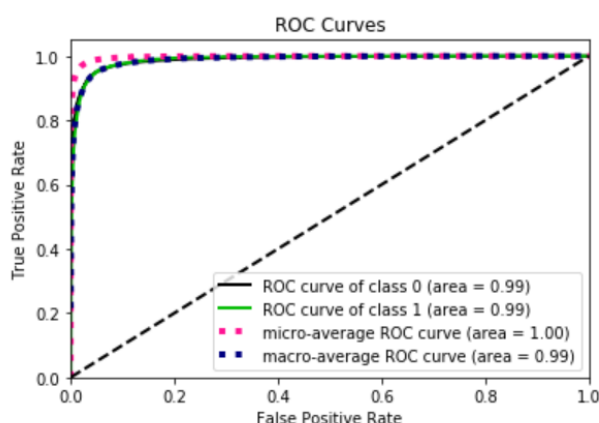


图 17: BLSTM-CNN ROC 曲线

## 2. 公平性评估

在深度学习模型中，有标签的样本 FPR 值比总体 FPR 值低了 0.006，说明相比较总体来说，有标签的样本中把无毒样本判为有毒的可能性更低；FNR 值比总体 FNR 值低了 0.006，说明相比较总体而言，有标签的样本把有毒样本判为无毒的可能性也较低。有标签样本与总体的 FPR 和 FNR 差值都远小于基准模型的 FPR 和 FNR 差值，说明深度学习模型在分类时较少受到样本中标签信息的影响，能在减少非意向偏差的同时，避免产生更多的“漏检”，较 NBSVM 模型而言，其更具公平性。

## 九、 总结和展望

为了更好地对有毒评论进行分类，在提高识别度的同时降低非意向偏差，本文提出一种将 BLSTM 和 CNN 结合的神经网络模型，并通过设计独特的加权损失函数解决数据倾斜和非意向偏差的问题。经过实验发现，BLSTM-CNN 模型比基准模型 NBSVM 有更好的分类表现：AUC 高达 0.98，比基准模型 AUC 高 0.01，整体分类表现更优；召回率达到 85.67%，比基准模型高 5.73%，能更好识别有毒样本。在解决非意向偏差的能力方面，本文提出了衡量指标：模型在有身份标签的子样本与总体的 FPR 和 FNR 指标方面的差距，差距越小，非意向偏差越小。BLSTM-CNN 模型在这两项指标上都取得了远远小于基准模型的值，说明其产生非意向偏差的概率更小，更具有公平性。

在超参数的选择上，由于时间和计算资源的限制，本文尝试有限，未来可以进一步在过拟合和正则项等参数上进一步调整；可以尝试 char 层面的 embedding；在权重方面，可以结合样本数量进行权数调整得到更优的权重；设计更多的模型进行模型融合。这些工作有可能获得更好的模型表现。

## 参考文献

- [1] McCallum A, Nigam K. A comparison of event models for naive bayes text classification[C]//AAAI-98 workshop on learning for text categorization. 1998, 752(1): 41-48.
- [2] Korde V, Mahender C N. Text classification and classifiers: A survey[J]. International Journal of Artificial Intelligence & Applications, 2012, 3(2): 85.
- [3] Parekh P, Patel H. Toxic Comment Tools: A Case Study[J]. International Journal of Advanced Research in Computer Science, 2017, 8(5).
- [4] Mohammad F. Is preprocessing of text really worth your time for toxic comment classification?[C]//Proceedings on the International Conference on Artificial Intelligence (ICAI). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2018: 447-453.
- [5] Goldberg Y. A primer on neural network models for natural language processing[J]. Journal of Artificial Intelligence Research, 2016, 57: 345-420.
- [6] He, H. and Garcia, E.A., 2008. Learning from imbalanced data. IEEE Transactions on Knowledge & Data Engineering, (9), pp.1263-1284.
- [7] Wei J W, Zou K. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks[J]. arXiv preprint arXiv:1901.11196, 2019.
- [8] Wang S, Manning C D. Baselines and bigrams: Simple, good sentiment and topic classification[C]//Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2. Association for Computational Linguistics, 2012: 90-94.
- [9] Dixon L, Li J, Sorensen J, et al. Measuring and mitigating unintended bias in text classification[C]//Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society. ACM, 2018: 67-73.
- [10] Zhou P, Qi Z, Zheng S, et al. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling[J]. arXiv preprint arXiv:1611.06639, 2016.
- [11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural computation, 9(8):1735–1780.
- [12] Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014.