附件 1

哈尔滨工业大学
大学生创新创业训练计划项目
中期检查报告

项 目 名 称：基于文本分类的诈骗识别

项 目 编 号：2022F0316

执 行 时 间：2022 年 9 月至 2023 年 9 月

负 责 人：徐浩铭

学院及专业：未来技术学院人工智能领域方向

指 导 教 师：孙承杰

日 期： 2023 年 4 月 6 日

哈尔滨工业大学本科生院

一、课题组成员：（包括项目负责人、按顺序）

姓名	性别	所在学院	年级	学号	身份证号	本人签名
徐浩铭	男	未来技术学院	2021	2021112905	511923200304134018	徐浩铭
杜佳兴	男	未来技术学院	2021	2021110962	140121200308027539	杜佳兴
樊宇宇	女	未来技术学院	2021	2021111004	142701200404091222	樊宇宇
王雅斌	男	计算学部	2021	2021110963	140110200301302535	王雅斌

二、指导教师意见：

项目完成了预定的中期目标,实现了基于文本分类的诈骗识别的演示系统并具有较高的准确率。中期报告格式规范，内容完整，同意参加中期答辩。

签　　名：

2023 年 4 月 9 日

三、项目研究中期报告

1、项目简介

本项目拟基于文本分类相关算法实现一个操作简便，结果准确的诈骗案情分析交互式机器人。近年来，国内网络诈骗犯罪活动猖獗，案件数量不断上升，给社会带来了极大威胁。同时，警务人员也面临着沉重的工作压力。在 NLP 迅猛发展的今天，文本分类成为了一个经典问题，在情感分析、垃圾过滤等领域得到广泛应用。因此，本项目受此启发，将运用文本分类任务来识别诈骗案件，并辨别其所属的诈骗类型。

项目的研究内容主要分为五部分。首先是构建语料库，相关数据由公安局提供。接下来是数据预处理阶段，包括分词处理、去噪和向量化。第三个阶段是选取多种模型作为文本分类器，包括经典的机器学习算法（如 SVM、贝叶斯和 KNN），以及深度学习中的 FastText 等合适的模型，以达到理想的精确率，从而实现项目的有效性。第四部分是结合 k 折交叉验证和网格搜索调参优化。最后一部分是实现项目的可视化，使用 Flask 来实现网站的开发，并将分类机器人嵌入到网站中。

本项目的难点主要在于如何实现理想的精确率和召回率，以达到项目的有效性，并真正减轻警务人员的工作负担。

2、立项背景

近年来，随着科技和互联网的快速发展，电信诈骗迅速蔓延，成为人民群众深恶痛绝的新型网络犯罪。其诈骗形式和手段多而变化莫测，给公安机关的侦查取证工作带来了很大挑战。据统计，利用电信网络技术犯罪的案件已经接近所有刑事案件的一半，并且这种情况还在不断加剧。习近平总书记指出，“坚决遏制此类犯罪多发高发态势，为建设更高水平的平安中国、法治中国作出新的更大的贡献”。

然而，目前对各种电信诈骗手段和方式的个性化分析研究甚少，尤其缺乏以数据为基础的精准分类研究。因此，将各种诈骗案件进行分类，有利于公安机关研究电信诈骗之间的交叉、区别和现状[1]。这也有助于在处理案件时，公安人员和个人能够更快速地定位案件案情，实施相应帮助和自救。

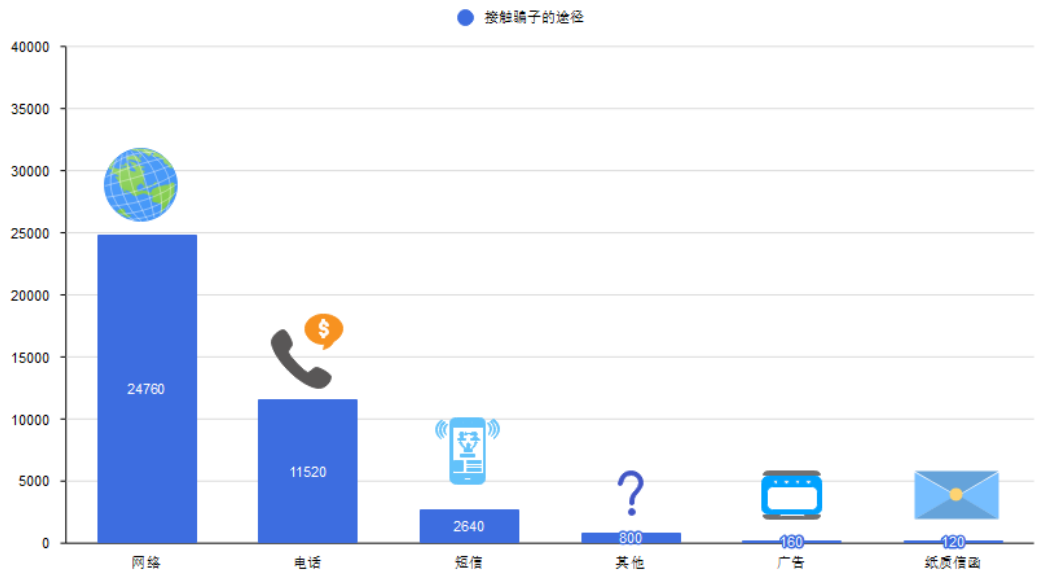

图 1 接触骗子的途径

人工智能技术的发展为诈骗案情的分析带来了新的机遇。通过自然语言处理中的文本分类技术，我们可以将文本集进行自动分类标记，从而更好地处理海量数据，提高数据的利用率。本项目旨在构建一个以文本分类技术为基础的诈骗案情分析交互式机器人，用于自动分类诈骗案情文本，从而实现对诈骗案情的精准分类。这一机器人可以在一定程度上减轻公安机关的工作压力，同时也可以让人们第一时间知晓是否处于骗局之中，减少人们的财产损失。到目前为止，尚缺乏针对电信诈骗的个性化研究。因此，本项目将为公安机关提供一个更高效、更精准的技术工具，以应对电信诈骗犯罪的挑战。

3、项目方案

按照项目实施的时间先后顺序，本项目可以大致划分为五个步骤：完成前序知识的学习和准备—数据预处理—模型—可视化和交互。

3.1 前序知识的学习和准备

学习的内容有 python 语言的学习，自然语言处理，相关软件的使用。我们用 markdown 文档记录学习过程，并把学习收获和项目代码文档部署到 github 上进行管理和记录。

3.2 数据预处理

数据预处理又可分为分词处理，去噪，向量化。这一部分小组成员可以通过自习自然语言处理入门，相关文献，小组讨论以及老师的指导来学习。我们经过学习与指导后确定使用 Jieba 完成分词处理。去噪可以使用停用词词典文件： stopwords.txt，该词典收录了常见的中

英文无意义词汇（不含敏感词），每行一个词。向量化可以用词袋模型 CountVectorizer 或者分布式表示如 Word2Vec。

3.2.1 Jieba

Jieba 分词的算法采用基于字符串匹配和统计分词的结合。在基于字符串匹配的算法中，如果一个单词存在于字典中，就直接划分出该单词；如果该单词不在字典中，就不对该单词进行分割。而在基于统计分词的算法中，核心思想是在不同的文章中，几个相连的字出现的频率越高，就越可能是一个词语。Jieba 通过将这两种方法结合使用，形成了一个层级结构。在这种结构下，每种分词的方法都可以找到一条从首字到末字的路径。在每个路径中，词语和词语间的分隔符就是边，而词语就是节点。在进行文本分词时，Jieba 首先根据前缀词典将所有分词结果都切分出来，然后使用分词结果构建一个有向无环图。接着进行文本标注，得到最大概率路径，从而得到最终的分词[2]。

3.2.2 Word2Vec

文本表示是自然语言处理的基础工作，它的处理结果会对整个自然语言处理网络造成极大的影响。用通俗的话说就是将文本转化成一系列能够表达文本语义的向量。在本项目中将使用分布式文本表示法的代表模型 Word2Vec。

Word2Vec 是一种词嵌入模型，是由谷歌团队在 2013 年提出的一个基于神经网络的语言模型，其支持对百万级的语料库进行高效训练；同时其训练的词向量语义表征性强，并且能够度量单词之间的词义相似度。其包含两种训练模型分别是 CBOW 和 Skip-gram。这两种是由 Mikolov 在 2013 年提出的基于词向量的无监督文本表示方法[3]。

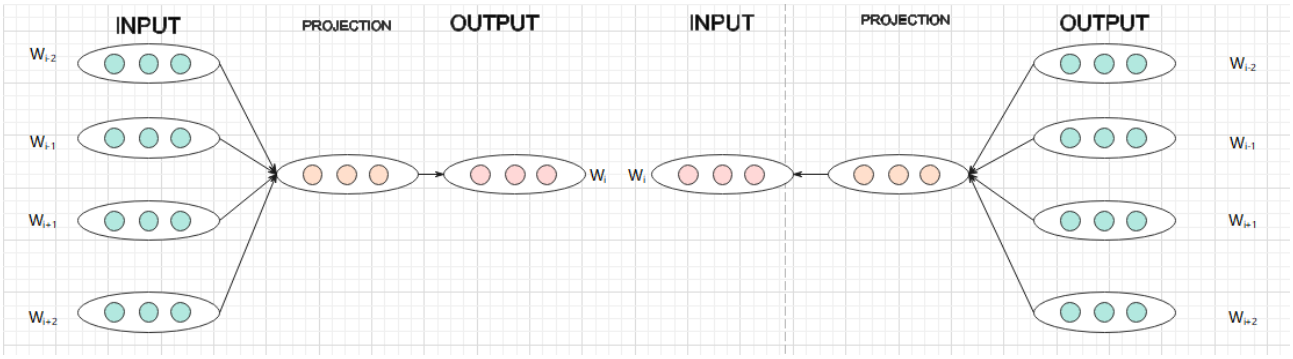

图 2 CBOW（左侧）模型与 Skip-gram（右侧）模型

CBOW 和 Skip-gram 模型都可以用于预测一个词语周围的上下文。当以一个词语作为输入，来预测它周围的上下文时，这就是 Skip-gram 模型。而当以一个词语的上下文作为输入，来预测这个词语本身时，则是 CBOW 模型。由于 CBOW 和 Skip-Gram 模型的原理相似，因

此在这里我们主要介绍 CBOW 的映射原理。 CBOW 模型主要由输入层、隐藏层和输出层构成。当我们将当前滑动窗口范围内的 k 个词作为上下文信息，并通过滑动窗口方式来预测中心词时，中心词的概率公式如下：

$$\mathrm{P}\left(W_t \middle| W_{t-k}, W_{t-(k-1)}, \ldots, W_{t+1}, W_{t+2}, \ldots, W_{t+k}\right) = \mathrm{P}(W_t|\mathrm{context}) \quad (1)$$

该算法首先会随机生成一个包含文本中所有单词的词向量矩阵。这个矩阵的每一行都代表着一个单词的向量表示。然后从词向量矩阵中提取中心词上下文的词向量并计算这些词向量的平均值。接着，对这个平均值向量进行逻辑回归训练，使用 softmax 作为激活函数。在训练完成后，该算法会比较得到的概率向量和中心词的概率向量是否接近来判断训练效果[4]。

相比之下，Skip-gram 算法则根据已知的中心词信息来预测未知的上下文信息[5]。其操作过程与 CBOW 相反。首先，也是随机生成一个包含文本中所有单词的词向量矩阵，其中每一行都代表一个单词的向量。然后，该算法会从文本中随机选取一个单词并提取它的词向量。接着，选择 softmax 作为激活函数并对这个单词的词向量进行逻辑回归训练。训练结束后，再比较得到的概率向量和上下文的概率向量是否接近。由于 Skip-gram 属于非监督学习，在大规模数据集的情况下，其训练结果更加精确。

3.3 模型

我们综合选取合适的模型。对于机器学习，由于我们是初学，我们首先采取经典的机器学习算法 SVM 达到学习和练手的目的。对于深度学习，采取了更加复杂有效的模型，综合选取两个合适的模型 FaxtText 和 TextCNN 并且最后达到理想的精确率和召回率，实现项目的有效性。

3.3.1 TextCNN 模型简介

在 2014 年 Kim[6]等人首次将 CNN 引入自然语言处理领域，应用到文本分类任务中。卷积神经网络是一种带有卷积操作的前馈神经网络，用于文本分类的卷积神经网络 TextCNN 在 CNN 的基础，对输入层进行了一些改动，但在网络结构上与 CNN 没有任何变化。如图 4 所示为 TextCNN 的网络结构，整个网络包括三个部分，输入层，计算层和输出层。

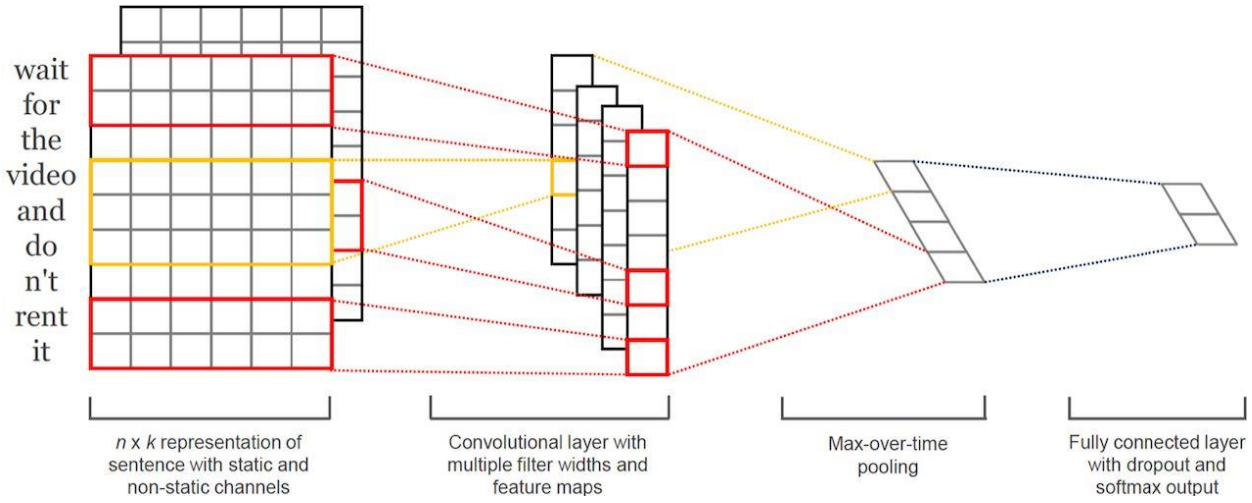

图 3 TextCNN 网络结构

输入层由 Word2vec 处理后的文本进行词嵌入，通过 K 维词嵌入后，一个词语数量为 N 的文本被编码为 N×K 的二维特征矩阵。由于句子长短不一，所以还需要进行归一化处理，长度不够 N 的用 0 进行填充，长度大于 N 的将过长的部分进行截断丢弃。

TextCNN 的计算层由一个卷积层，一个池化层以及全连接层构成。卷积计算之后拼接三个不同维度的特征图，再使用最大池化进行池化。接着再使用全连接层对特征进行降维。输出层使用 softmax 函数得到每个文本的类别[7]。TextCNN 在卷积层的操作属于一个线性操作，但是，文本数据信息复杂多变，并不是线性的，单一线性函数是无法解决非线性问题的，所以需要在模型结构中增添一层激活层，使得模型能够解决像多分类这种非线性问题。模型常用的激活函数有如下几种:

(1) Sigmoid 函数:输出范围在(0,1)之间，优化稳定,求导简单，但是容易出现梯度消失的问题。

$$Sigmoid(x) = \frac{1}{1+e^{-x}} \tag{2}$$

(2)tanh 函数:要比 Sigmoid 激活函数的收敛速度快，但是由于其需要繁琐的幂运算,计算成本较高,同时也存在梯度消失的问题。

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3}$$

(3)ReLU 函数

ReLU 激活函数应用十分广泛，因为其不需要进行幂运算，所以运算速度也较快,相对于上面两种激活函数而言,ReLU 函数也可以极大改善梯度消失问题，同时由于 ReLU 函数会失

效一部分神经元，这样的特点可以减少模型发生过拟合[8]。

TextCNN 作为用于文本分类的神经网络，网络层数不多，结构简单，计算量小，网络收敛快。作为局部寻优的模型，使用预训练效果好的词向量，TextCNN 可以在短文本分类取得很好的分类效果，但若文本序列较长的话，其特征提取能力受限，导致分类效果不佳。

3.3.2 FastText 模型简介

FastText 是 Facebook 于 2016 年开源的一个词向量计算和文本分类工具，在学术上并没有太大创新。但是它的优点也非常明显，在文本分类任务中，FastText（浅层网络）往往能取得和深度网络相媲美的精度，却在训练时间上比深度网络快许多数量级。在标准的多核 CPU 上， 能够训练 10 亿词级别语料库的词向量在 10 分钟之内，能够分类有着 30 万多类别的 50 多万句子在 1 分钟之内。如图 5，FastText 模型也只有三层：输入层、隐含层、输出层。

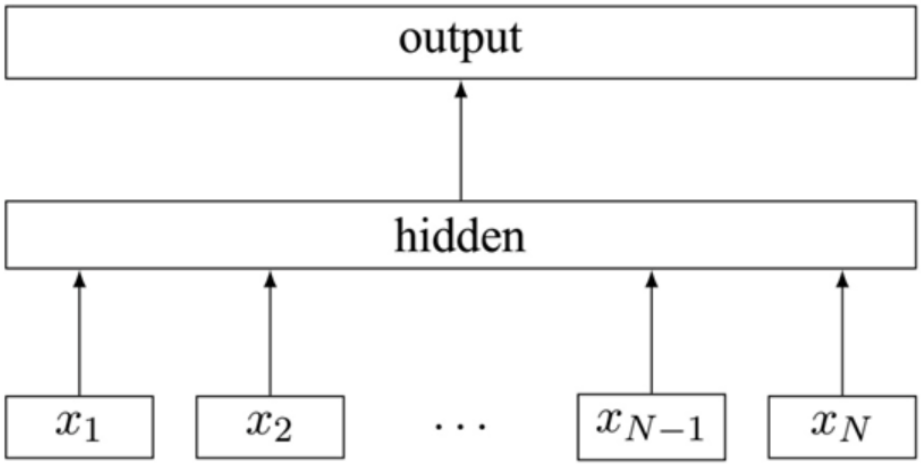

图 4　FastText 网络结构

输入都是多个经向量表示的单词，输出都是一个特定的 target，隐含层都是对多个词向量的叠加平均。值得注意的是，FastText 在输入时，将单词的字符级别的 n-gram 向量作为额外的特征；在输出时，FastText 采用了分层 Softmax，大大降低了模型训练时间。

运用模型需要了解分层 Softmax 和 n-gram：

(1)Softmax 的基本思想是使用树的层级结构替代扁平化的标准 Softmax，使得在计算 P(y=j)时，只需计算一条路径上的所有节点的概率值，无需在意其它的节点。树的结构是根据类标的频数构造的霍夫曼树。K 个不同的类标组成所有的叶子节点，K-1 个内部节点作为内部参数，从根节点到某个叶子节点经过的节点和边形成一条路径，路径长度被表示为 $L(y_j)$。

于是，P(y_j)就可以被写成：

$$p(y_j) = \prod_{l=1}^{L(y_j)-1} \sigma\left(\left[n(y_j, l+1) = LC\left(n(y_j, l)\right)\right]\right) * \theta_{n(y_j,l)^T} X \tag{4}$$

(2)n-gram 是一种基于语言模型的算法，基本思想是将文本内容按照字节顺序进行大小为 N 的滑动窗口操作，最终形成长度为 N 的字节片段序列。注意一点：n-gram 中的 gram 根据粒度不同，有不同的含义。它可以是字粒度，也可以是词粒度的。

从输入得到输出也需要经过函数的计算

隐含层 h 的输出函数为：

$$h = \frac{1}{C} W * (\sum_{i=1}^{c} x_i) \tag{5}$$

接着我们计算输出层的每个节点：

$$u_j = v_{w_j}^{\prime T} * h \tag{6}$$

最后计算得到

$$y_j = p\left(w_{y_j} \middle| w_1, \ldots, w_c\right) = \frac{\exp(u_j)}{\sum_{j'=1}^{V} \exp(u_j')} \tag{7}$$

3.4 可视化和交互

中期时搭建一个可视化网页，实现内容分析和基本功能的展示，结题时设计实现一款小程序，方便用户使用。

3.4.1 网页框架

基于上述文本分类的模型构建与优化，为了实现项目的可视化。考虑搭建一个网页，同时将分类机器人嵌入，部署上服务器。目前有很多比较知名的 web 框架，分别是 Diango、Tornado、Flask。其中 Django 是市场占有率极高的框架，适合大项目，官方文档齐全；Tornado 的异步高性能框架，包含许多底层细节，少而精； Flask 微框架，轻量级，扩展插件较多。在本项目中我们将使用 Flask 来搭建网页。

使用 Flask 是因为其是轻量级的微框架，扩展插件比较多，且其有超高的扩展性和小而精的核心本身，所以项目预计将使用 Flask 框架。

Flask 的工作流程为:在用户访问 URL 时，通过 WSGI(Python Web Server Gateway Interface)

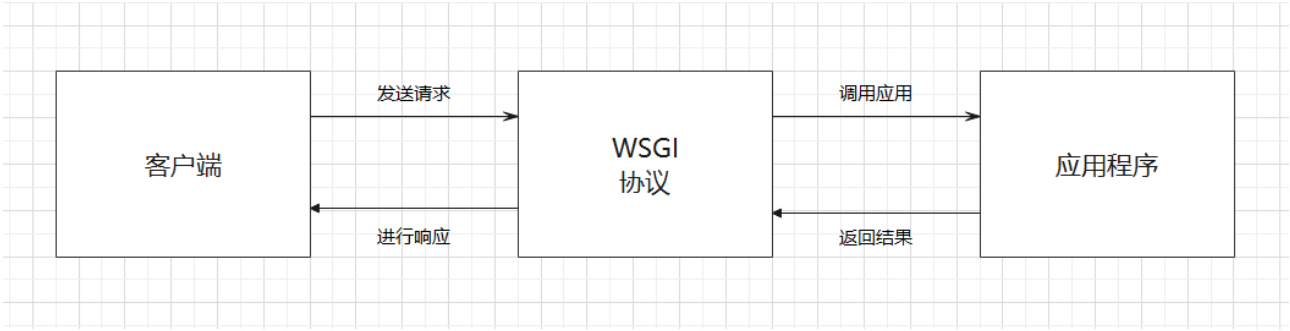

图 5 Flask 工作过程图示

协议将请求信息转换为服务器处理的相应接口格式，调用服务器的相应函数生成返回信息，经过 WSGI 协议转换格式，最后传递至前端界面展示该信息[9]。工作过程如图所示：

3.4.2 微信小程序构建

为方便个人用户使用和警方管理，本项目预计结题时基于上述文本分类的模型构建与优化，设计一款简单的针对诈骗文本分类的反诈小程序，包括最为基础的网页框架的搭建，前后端的对接，服务器的部署与进阶一步的分类机器人的嵌入等。通过这些研究，提高人机交互性，不断丰富用户的使用体验。

4、项目实施的进展情况及初步取得的成果

4.1 进展情况

本项目基本实现了立项时所要求的中期目标。基本完成了前序知识的学习与准备，掌握了 python 的使用，机器学习模型，深度学习算法与自然语言处理的基础知识。良好的完成数据预处理(包括分词处理，去噪，向量化等过程)，最大程度上减少后期计算的内存开销和计算误差，为模型运用与准确率提升打好数据基础。运用 FastText 和 TextCNN 框架初步实现了项目完整的流程，并且能得出较为准确的结果。最后用 Flask 框架搭建一个展示网页，实现较好的可视化。并且由于 ChatGPT 的爆火，加入了 ChatGPT 接口，以求实现准确性的大幅提升。

4.2 取得成果

在本项目中诈骗案件一共有 13 个分类，分别是“冒充电商物流客服类，贷款、代办信用卡类，虚假网络投资理财类，冒充领导、熟人类，冒充公检法及政府机关类，网络游戏产品虚假交易类，刷单返利类，其他类型诈骗，虚假征信类，冒充军警购物类诈骗，虚假购物、服务类网黑案件，网络婚恋、交友类（非虚假网络投资理财类）”。由于我们现阶段的数据集是已经确定的诈骗案件的描述，我们只需要对其进行分类即可，无需判断其是否为诈骗文本。

输入一段描述，即可得出其所属的类型。例如图 7，当输入一段与刷单有关的文本，即可得出该诈骗属于刷单返利类的结果；输入一段与贷款相关的文本时，即可得出该诈骗属于贷款、代办信用卡类的结果。目前我们的分类模型已实现了较高的准确率，达到了良好的效果。但由于无法判断文本是否为诈骗文本，所以当输入文本与诈骗无关时依然会得出分类的结果，例如，当输入文本为“111”时，这个文本与诈骗无关，但分类模型仍然会得出该文本是其他类型诈骗的结果，这个漏洞是我们后续需要改进的地方之一。

图 7 诈骗分类示例

此项目利用 Git 来实现 github 仓库的管理，实现了项目资源的共享，利于我们互相监督，有利于我们团队合作的展开。如图 8 所示为我们截至到中期的项目文档目录。

其中主程序文件 run.py 负责输入参数，生成模型配置参数，训练并测试神经网络分类器。其中，主程序文件中导入其他程序文件的内容，包括 train_eval.py，models、utils.py 和 utils_fasttext.py，这些程序文件中包含了生成、训练和测试深度学习模型需要用到的各个函数和类。

train_eval.py 文件是用于训练和评估模型的。训练函数包括了模型初始化、权重初始化、、定义优化器、定义学习率衰减、迭代训练、记录训练结果等，评估函数包括了加载已训练好的模型、迭代评估、计算评估指标等。

utils.py 文件包含了一些辅助函数，这些函数功能为用来建立词典、读取多个文件、计算程序运行时间。

utils_fasttext.py 文件实现了文本分类模型 FastText 中用到的一些数据处理方法和数据集迭代器的构建。

FastText.py 文件用于构建 FastText 模型进行训练和预测。

TextCNN.py 文件用 TextCNN 模型实现了文本分类任务，并提供了配置参数，与向前传播方法等函数。

图 8 Github 仓库中期文件目录展示

除了前期的准备之外，我们在本项目中较好的实现了数据的预处理，利用 Jieba 完成了分词处理，利用 Word2Vec 完成了文本的向量化，在模型选取上主要实现了 FastText 以及 TextCNN 模型的应用。如下图两种模型的预测正确率以及两种模型的损失函数，图中蓝色的曲线代表的是 FastText 模型，橘色和紫色代表的是 TextCNN 模型。左上角图为测试集的准确率可以看到 TextCNN 的准确率达到了 86.33%，且 FastText 模型的准确率达到了 78.44%，最高准确率如表 1 所示，两者的准确率都达到了较高的水平。右上角图中为训练集的准确率，可以看到两种模型的准确率都较高，达到了我们的预期。而第二行的图为两个模型的损失函数，我们可以看到在测试集中两个模型的损失函数最后都较低，同时在训练集中两者的损失函数也较低，达到了我们的预期。

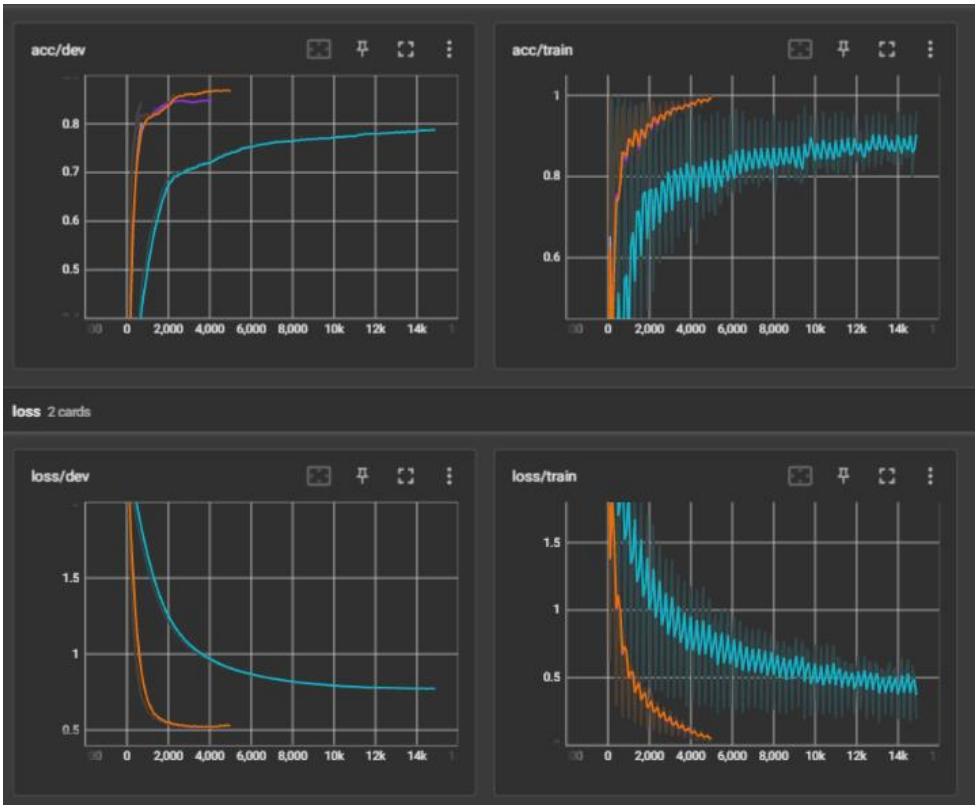

图 8 模型的准确率以及损失函数

模型	acc	备注
FastText	78.44%	Bow+bigram+trigram
TextCNN	86.33%	Kim2014 经典的 CNN 文本分类

表 1 最高准确率表格

此外我们利用 flask 框架实现了项目的可视化，目前编写了三个 html 网页，分别是本项目的核心文本分类的 index.html 主页，以及展示本项目基本教程的 start.html 网页，还有展示有关本项目的一些数据的图表的 chart.html 网页。在 index.html 主页中，可以通过下拉框来选择利用哪种模型来进行预测。由于近日 ChatGPT 的爆火，所以我们接入了 ChatGPT 接口，便于提供更加准确的分类结果，以及为人们提供一定的建议。如下图所示为 index.html 主页，利用了 TextCNN 模型进行分类，得到了分类结果以及分类的时间。预测结果下方的三个模块即为本项目的简单教程、项目数据的一些图片与表格、以及我们的参考项目。

图 9 基于 flask 框架的可视化网页

项目可视化结果分析，图 10 为案件类型分布情况的扇形图，图 11 为案件类型分布情况的热力图，图 12 为案件类型分布情况的柱状图，通过这些图可以更加直观的看到分类的结果，其中刷单返利类所占类型最多，了解了案件类别的分布情况，使得数据更加的有价值。此外用词云展示了文本中出现频率最高的 120 词，如图 13 所示读者若碰到这些敏感的词汇应该警惕。

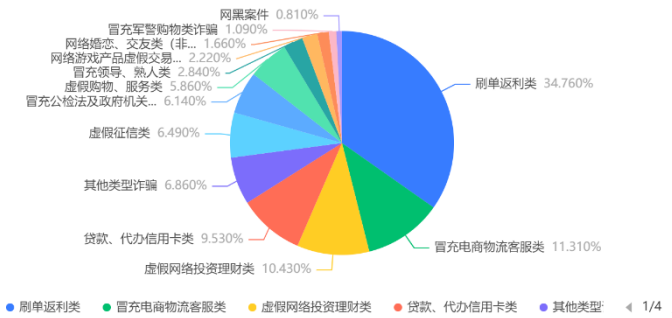

图 10 案件类型分布情况扇形图

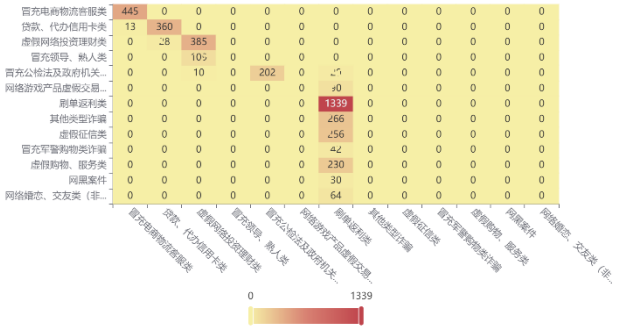

图 11 案件类型分布情况热力图

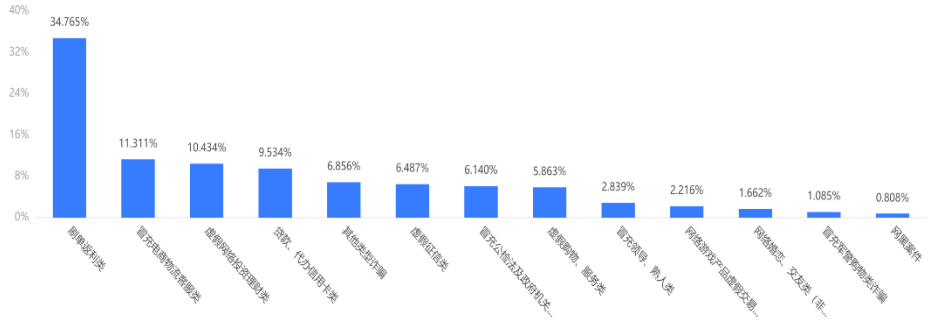

图 12 案件类型分布情况柱状图

图 13 文本中出现频率最高的 120 个词

5、特色与创新

(1)在完成模型的初步实现后，使用 k 折交叉验证和网格搜索进行调参优化。

网格搜索能够通过穷举法得到最优的超参数组合，然而，本项目可能存在部分数据量较小的情况，在网格搜索的过程中，验证集中并没有足够多的数据，不能代表总体数据的特征，因此调参过程具有极大的偶然性，会影响到最终模型的准确率。针对这个问题，本项目小组考虑采用 k 折交叉验证法，通过充分利用数据来避免这种偶然性，选择超参数的最优组合。

同时，在使用 k 折交叉验证法的过程中，由于每一个数据都有可能充当训练集，验证集或者测试集，因此所有数据都会参与到训练和测试的过程中，从而能够有效解决欠拟合和过拟合的问题。不仅如此，使用 k 折交叉验证还能评估各个模型的质量，选取给定数据集上的最优模型。因此，结合网格搜索与 k 折交叉验证的方法，能够有效提升文本分类模型的准确率，增强本项目中文本分类器的效果。

(2)项目计划构建多个基本的分类模型，基于此计划搭建多个文本分类器，后面通过一些算法例如 Bagging、Boosting 和 Stacking 等进行模型结果的综合，通过综合多个分类结果，能够在很大程度上提升结果的准确率。

(3)项目在完成了模型建立与优化调整的基础上，搭建前端网页框架，考虑嵌入分类机器人，部署服务器，更进一步完成一个小程序的初步构建，推动文本分类功能进一步完善，人机交互性进一步增强，从而能够从创新的小切入口逐渐过渡到初步的创业雏形。通过这些可视化过程，本项目不仅仅停留在理论上的分类模型建立，更能够进一步走向反诈的实践，更大的应用与实践空间也能够反向拉动模型的进一步优化，实现理论与实践的结合与良性循环。

6、项目实施过程中的收获与体会

通过一段时间的学习与知识储备，我们了解了基本的文本预处理流程以及框架，认真学习了 FastText 模型以及 TextCNN 模型，在前端部分，了解了现行主流前端框架并且选择了轻量级 flask 框架构建前端可视化界面。此外，我们还学会使用了 git 来进行 github 仓库的管理，从而能够同步大家的进度，共享项目资源。在这个过程中，我们学习到了很多实用的软件构造知识。在完成 TextCNN 模型的构建过程中，我们遇到了参数调整带来的一些问题，通过反复测试阅读文献与调参，我们总结出来一些经验，例如，卷积核的大小影响较大，一般取 1~10，如果文本中句子较长，则应选择大一些。卷积核的数量也有较大的影响，一般取 100~600 ，同时一般使用 Dropout（0~0.5）。此外，随着 feature map 数量增加，性能减少时，需要尝试大于 0.5 的 Dropout。此外，在模型构建与训练过程中，我们还碰到了模型过拟合的问题，通过反复比较测试结果，我们发现采用 dropout 方法，减少迭代次数，增大学习率等等方法可以解决过拟合问题。另外在前端的搭建过程中，我们在 flask 框架的安装以及代码编写过程中都遇到了一些问题(例如不同模型在同一个复选框内如何调用等等)。最后通过查找文献资料，阅读开源代码，请教指导老师和学长以及定期小组讨论的方式解决了这些问题，增强了我们分析问题与交流沟通的能力。

我们在项目实践中发现问题，并且针对问题展开学习和沟通，在这个过程中收获颇丰。

7、经费使用情况

预算类别	主要用途	预算金额（元）
书籍费	购买书籍等	300
资料打印	打印所需的资料	50
服务器租聘	模型训练，网站搭建	400

表 2 经费预算表

8、参考文献

[1]熊春海. 电信网络诈骗犯罪的现状及治理完善路径[D].广西师范大学,2022.DOI:10.27036/d.cnki.ggxsu.2022.000940.

[2] Ding Y, Teng F, Zhang P, et al. Research on Text Information Mining Technology of Substation Inspection Based on Improved Jieba[C]//2021 International Conference on Wireless Communications and Smart Grid (IcwCsG). IEEE,2021: 561-564.

[3]Tomas Mikolov,Kai Chen 0010,Greg Corrado,Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space[J]. CoRR,2013,abs/1301.3781.

[4]崔文艳. 基于时间卷积网络的商品标题文本分类方法研究[D].天津师范大学,2022.DOI:10.27363/d.cnki.gtsfu.2022.001116.

[5]黄聪. 基于词向量的标签语义推荐算法研究[D].广东工业大学,2015.

[6]Yoon Kim. Convolutional Neural Networks for Sentence Classification.[J]. CoRR, 2014,abs/1408.5882.

[7]曾芳. 基于混合卷积的文本分类算法研究[D].西南科技大学,2022.DOI:10.27415/d.cnki.gxngc.2022.000957.

[8]郭书武. 基于深度学习的教材德目分类方法研究[D].上海师范大学,2022.DOI:10.27312/d.cnki.gshsu.2022.001952.

[9]刘嘉伟. 基于 FLASK 的校园智能停车系统的构建[D].吉林大学,2021.DOI:10.27162/d.cnki.gjlin.2021.001834.