专业综合设计

石锴文
樊世源
林卓凡

目录

1 项目背景

1.1 领域问题

1.1.1 文本分类

随着现代科技的快速发展，需要识别、分类的数据呈几何数量上升，形成了现在的大数据时代。因此，对大数据的分类识别成为企业的核心竞争力所在。

1.1.2 实体抽取

命名实体识别（Named Entity Recognition，NER）作为信息抽取与信息检索中的一项重要任务，在文本实体识别中起着重要作用，NER 是研究如何提高文本中命名实体识别精度并执行分类的自然语言处理技术（Natural Language Processing，NLP），该名词有时也会被称为命名实体识别和分类（Named Entity Recognition And Classification，NERC）。

互联网金融实体是指在网络上被广泛使用的金融词汇，包括金融公司名称、金融产品名称、金融项目名称、金融专业术语等。中国互联网蓬勃发展，随之而来的也是互联网金融实体急剧增长。网络上的金融项目、金融概念、金融产品呈井喷式的增长，由此引发了大量问题。互联网金融实体在文本中内容分散，数据稀疏，无结构化等特点也逐渐凸显。互联网金融文本中金融实体，往往具有以下几个特点：

（1）名称长度差别大。互联网金融实体经常以全称的形式出现，这些全称短则几个字，而长则十几个字，其中往往包含了地名、人名以及一些不常见金融词汇，比如“资易贷金融信息服务有限公司”、“e 租宝”等。

（2）同一互联网金融实体往往存在着不同的表达方式，且不同表达之间相关度有高有低。以中国农商银行为例，存在农商行、中国农商银行、农商银行、信用社等多个称谓，其中信用社和其他实体名相关度接近于无。

（3）相当一部分互联网金融实体命名无规则。以互联网金融实体“e 租宝”为例，它是某公司名，命名方式不是已出现实体的组合，因此通过常用的 NER 方法对该实体进行识别难度极大。命名实体识别一般是基于 3 类信息，包括人名、地名、机构名，而经过多年发展，命名实体识别工具中往往包含着实体库，实体库会对互联网金融实体识别进行干扰。比如将小段未知实体包含在多个实体的语段中，命名实体识别只会识别实体库中的实体，而未知实体极大可能被漏掉。面对浩瀚的互联网信息时，如何高效地找到自身所需互联网金融实体信息成为一道难题。

以金融监管部门为例，互联网金融实体激增，很多实体的命名方式是不规则的、无规律的。近些年来，资本市场违约事件频发，财务造假、董事长被抓、股权质押爆仓、城投非标违约等负面事件屡屡出现，导致互联网金融实体的可信度和权威性受到广泛质疑。

基于上述考虑，对现有模型进行改进从而提高面向互联网金融文本识别效果至关重要。传统的命名实体识别是基于现存知识库，而现存库中缺乏互联网金融实体信息。没有专门面向该领域命名实体识别方案，只能对现有的命名实体识别方案的学习与归纳。从给定的互联网信息中提取、识别出企业主体名称，以及标记风险标签。本文提出一种改进的命名实体识别模型结构，可以更好的适应互联网金融实体特点。金融实体识别模型的建立将有效提高互联网金融实体识别效果，从而更好的为有需求的相关机构和个体提供信息支撑。

1.2 国内外研究现状

1.2.1 文本分类现状

传统的分类方法中，每个样本示例只属于一个类别标记，即单标记学习。最近，TextCNN 的模型提出，将卷积神经网络（Convolutional neural network, CNN）加入到自然语言处理任务中，引发了将深度学习运用在单标签文本分类任务中，取得了较好的效果。但多标签文本分类问题的研究，尚处于发展阶段。本项目使用了 BiLSTM，GRU,FastText，Transformers，CNN，一共五个模型，对此数据集进行对比试验。

1.2.2 实体抽取现状

识别各类文本中具有独特而且明确含义的实体被称为 NER。按照传统定义 NER 任务是为了识别出各种文本中的 3 大类（命名实体类、时间类和数据类）、7 小类（人名、机构名、地名、时间、日期、货币和百分比）命名实体（Chinchor，1997）。

NER 的方法一般来说分为 4 种类别：一种是基于词典和规则的方法，这种方法出现较早，现在是作为其他方法的补充；一种是机器学习的方法，这种方法对后续的命名实体识别研究影响巨大；一种是基于深度学习的方法，这种方法目前是 NER 研究的主流，该方法由于用到的数据集相对较少而训练效果较好，因此被广泛的使用。

基于模式匹配的方法

早期的 NER 大多是基于词典和规则的方法进行的，一般会由相关领域的专家参与 NER 任务。专家们需要编写适用于该领域的规则，对该领域内的专业词汇进行总结并编写规则，为研究者提供专业名词解释，研究者的重点是根据专家们总结归纳的内容，利用词典和规则保证编写的系统在该领域有着很好的适用性。规则模板的编写一般来说对专业知识掌握程度要求极高，因此想通过非专业人士总结归纳基本不可能。例如构建模板词库，如表 1 所示：

表 1: 模板词库示例

元素	类型
< 姓氏 > 和 < 名字 >	Name
< 机构部分 > 和 < 机构类型 >	Company
< 地名部分 > 与 < 地名指示词 >	Location

命名实体的构成规则确立以后，便能利用规则与字符串序列之间拟合度进行匹配，某一实体与规则的匹配程度高于某设定值时可以确定为目标命名实体。基于该方式，Farmkiotou D 等人面向希腊金融文本提出了一种基于规则的 NER 算法（Farmkiotou D 等，2000）。传统的 NER 研究者认为典型的 NER 系统应是由词典和语法组成的。其中，词典是指研究领域中的特殊词汇，语法是一种语言的构成规则。以汉语为例，它有自己的主语、谓语以及修饰主谓的形容词、副词。面向一个新的未知领域时，首先想到的应该是该领域的规则，然后通过字符串匹配的方式实现 NER，该方法在随后的测试中取得比较优秀的成绩。基于该方法构建的系统包括 NTU 系统（Black W.J 等，1998）、OKI 系统（Ding Y.W 等，1998）、FACILE 系统（Chen H.H 等，1998）等。[1]

一般来说，规则的构建是基于专业领域的知识，知识量的增加可以帮助专家更好的发现规律，提高规则完善程度。基于该考虑，王宁等在文本识别中创建了 6 个知识库为公司名识别提供匹配信息，利用公司名该实体本身的结构特征以及识别文本实体的规则提高了模型识别效果。该方法面向开放测试集中 F1 为

62.80%（王宁等，2002）。[2] 该模型的大概思路是借助规则进行文本信息匹配，再基于贝叶斯模型进行决策，通过获取实体的左右边界获得命名实体。该方案在开放环境中的 F1 值为 84.10%。沈嘉懿等基于该模型实现中文关系抽取系统的构建（沈嘉懿等，2007）。该方案考虑从实体上下文入手，通过构建上下文所遵循的规则，利用相似度定义的方式，对机构名全称中出现的实体进行组合，如果某一机构名符合其中一到两个名字，再进行相似度匹配，如果 3 种相似度匹配都比较高，则判定该机构名是目标实体，该方式使得机构名缩写形式的识别方法成为可能。

模式匹配事件抽取方法在领域事件抽取任务中性能优异，但模板的制作需要耗费大量人力和时间，且模板局限于领域背景，很难在通用领域事件抽取任务中应用。并且基于模式匹配的方法不具有良好的泛化性能。总的来说，这种方法是早期进行 NER 所使用的方法，面向互联网金融实体该体量巨大的新兴事物，单纯依靠这种方法较难获得有效的识别结果，但该方法可以作为其他方法的一种补充，用于查缺补漏。

基于机器学习的方法

由于机器学习概念的兴起，NER 方法研究也逐渐使用机器学习方法。与人工智能方法类似，基于规则的命名实体识别方法依赖于知识的数量和质量，导致了该方法难以大规模推广。统计方法在原有的训练或加工语料库的基础上进行数学意义上的规律统计，而加工语料库的标注不需要太广泛的语言知识即可完成。另外一方面如果你使用了一个新领域的语料库进行训练，那么训练出来的系统在新的领域有着一样的训练效果，这种方式不需要对规则进行重写，因此相比于基于规则的命名实体识别方案，该方法的适用性更高，可移植性更好。基于统计机器学习的方法可以说全面超越基于规则的 NER 方案，基于统计的方法比较多，很多都是智能信息处理领域的经典模型，目前仍在使用，包括决策树（Decision Tree，DT）、最大熵模型（Maximum Entropy Model，ME）、n 元模型、隐马尔可夫模型（Hidden Markov Model，HMM）、条件随机场 (conditional random field, CRF) 、支持向量机（Support Vector Machine, SVM）、条件马尔科夫模型等。其中 ME 模型因其自身的特点仍然是当前的主要研究方向。而 HMM 模型因为具有比较好的模型结构特点，因此在各项测试中相比于其他模型的识别精准度更高，文本识别的概率更大，因此评价最高。Yang 等人利用隐马尔科夫模型与概率估值公式相结合的方法对文本中的构成组织机构名的能力进行评价，以实现中文组织机构名的自动识别，达到了 89.00% 的准确率（Yang 等，2011）。

传统的机器学习方法将事件抽取任务建模为多分类问题，提取文本的语义特征，然后输入分类器进行事件抽取，如图 1 所示。Ahn 等 [3] 用 Timbl 和 MegaM 模型进行分类，利用词汇特征、字典特征、句法特征、实体特征完成触发词分类子任务，利用事件类型、触发词特征、实体特征、句法特征完成事件元素分类子任务。

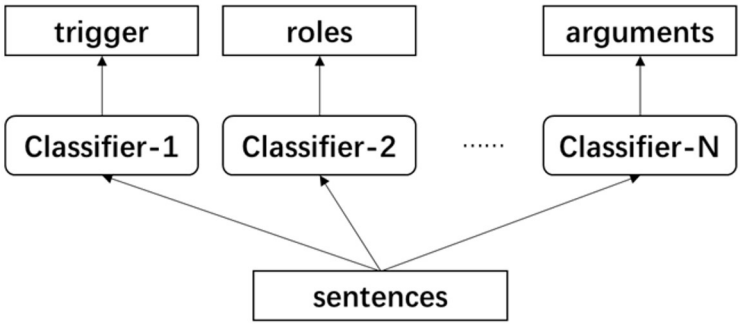

图 1: 事件抽取子任务分解

基于深度学习的方法

近年来基于深度神经网络的因果关系抽取方法比较多，主要分为两大类：一类是基于流水线方式；另一类是基于联合抽取的方式。前者将抽取任务看作是实体识别和关系分类两个子任务，后者则是利用联合模型将因果关系三元组直接抽取出来。

事件检测的主要工作是识别语料中的事件触发词和相关论元。CNN 通过捕捉句子中重要特征从而获得句子表示，传统的 CNN 模型在池化操作后获得的向量表示会错过有价值线索。为了解决这一问题，Chen 等 [4] 提出了动态多池卷积神经网络（Dynamic Multi-pooling Convolutional NeuralNetwork，DMCNN）来提取句子级特征，如图 2 所示。DMCNN 使用一个动态的多池层来获取句子各部分的最大值，句子表示的各部分依据事件的触发词和论元进行分割。与传统 CNN 模型相比，DMCNN 无需借助 NLP 工具的帮助，能够捕获更多有价值的特征。但 DMCNN 模型中的语料需要预先标记好触发词和相关论元，可以将其简单的理解成对触发词和论元之间的角色进行分类。

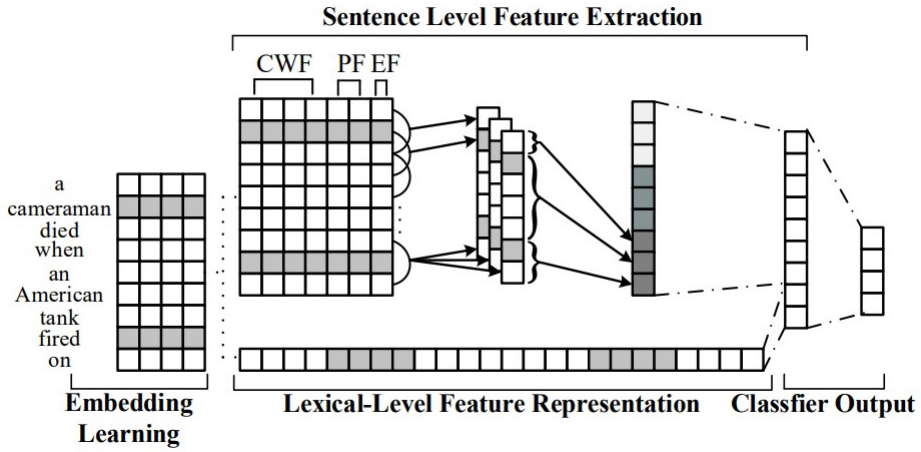

图 2: 动态多池卷积神经网络

Zheng 等 [5] 采用联合学习框架进行命名实体识别和关系分类，提出混合神经网络模型来提取实体和它们之间的关系，而不需要任何手工制作的特征。Zheng 的模型如图 3 所示，包含一个用于实体提取的双向编码器-解码器 LSTM 模块和一个用于关系分类的 CNN 模块。在 BiLSTM 中获得的实体上下文信息进一步传递给 CNN 模块，以改进关系分类。与传统的流水线方法相比，该模型不仅考虑了命名实体识别（Named entity recognition，NER）模块和关系分类模块的相关性，而且考虑了实体标签之间的长距离关系，不需要复杂的特征工程。

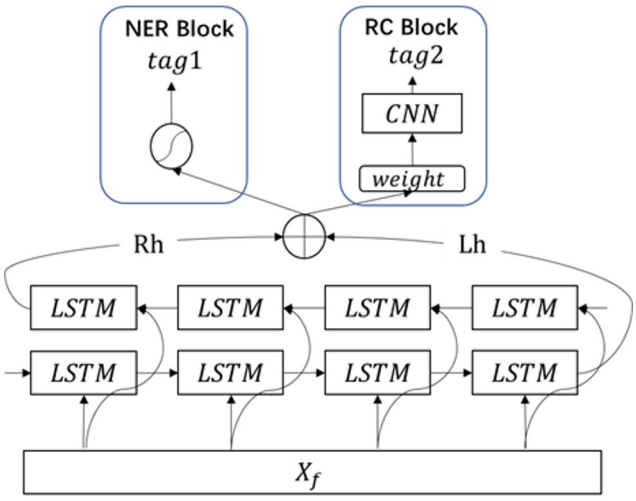

图 3: BiLSTM-CNN 混合神经网络模型

杨飘等人提出 BERT-BiGRU-CRF 该模型结构用于表征语句特征。先将预处理的文本通过词向量、句向量、位置向量输入 BERT 层，从 BERT 层输出含有整个文本信息的向量表示，再经过双向 GRU 模型训练，通过 CRF 层输出命名实体。训练方式包括训练整个模型和只训练 BiGRU-CRF。在 MSRA 语料库上的实验结果表明，两种训练方法的 F1 值分别为 95.43% 和 94.18%。Wang 等人提出一种结合 IDCNN 和 BiLSTM 的结构特点的可扩展的序列标记模型 STM（Zhili Wang 等，2020）。大量实验验证了 STM 对互联网金融实体的识别优于 IDCNN-CRF、BiLSTM-CRF 等模型结构，F1 值为 93.23%。[1]

2 探索性数据分析

2.1 文本内容分析

本项目使用预训练的字向量，故对文本内容进行字符级的分析。对文本进行词频统计后，发现分类任务和标注任务上的高频词均集中在“公司”“股份”等字词上。

图 4: 文本词云图（左为分类文本，右为标注文本）

随后对句子长度分布进行分析，分类文本的句子长度主要集中在 15-102，标注文本的长度主要集中在 21-105，随后在进行句子填充时，可以选择填充长度为 128。

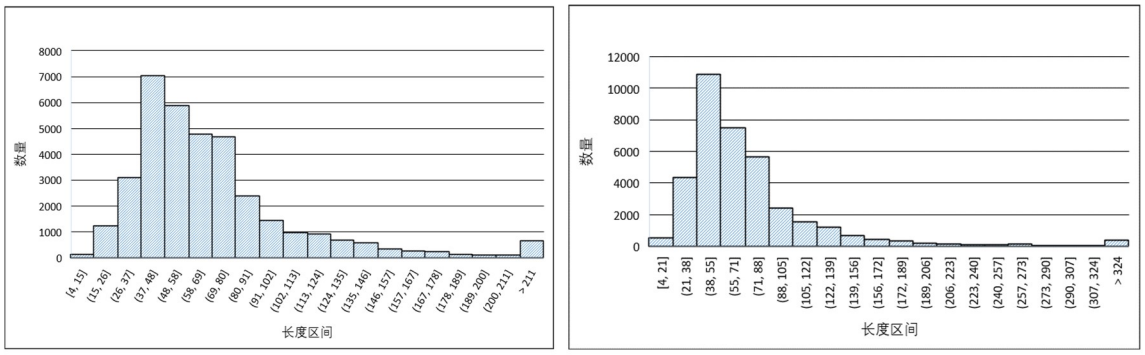

图 5: 句子长度分布图（左为分类文本，右为标注文本）

2.2 分类标签分析

对分类标签分布进行统计，发现样本存在数据标签不均衡问题，数量最多的标签为“业务资产重组”，最少的为“履行连带担保责任”。本项目数据来自海通证券和工商银行，标注数据质量高，故不便于通过重采样方法来解决不均衡样本问题。后续可以尝试使用代价敏感学习，调整少数类判别错误带来的损失。

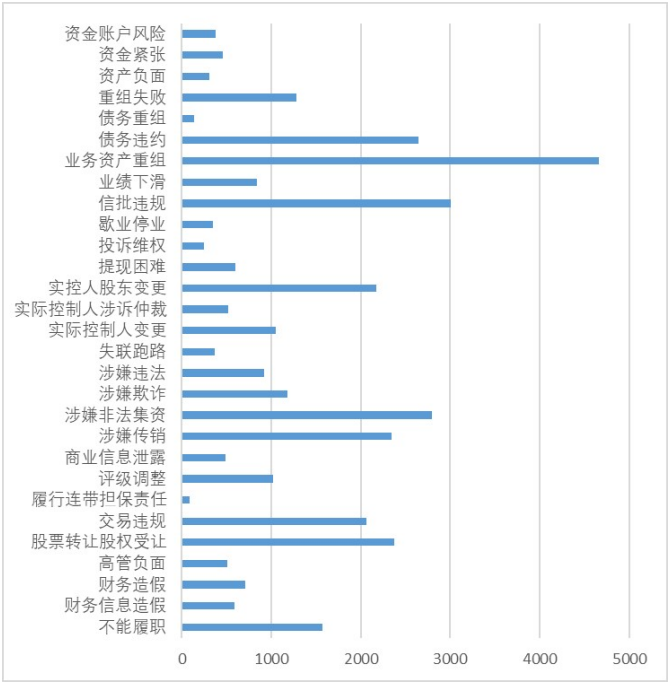

图 6: 分类标签分布情况

之后，为了排除文本长度与标签之间存在关联性的可能，对文本平均长度与标签做皮尔逊相关性检验。经过 Pearson 相关性检验，文本平均长度与标签类别在 95% 的置信水平下，p 值为 0.717，大于 0.05，所以在该显著水平下，两者不存在显著的相关性。故后续的分类模型能够排除文本长度对标签的影响，分类性能全部来源于模型本身所提取的特征。检验结果如表 2 所示。

表 2: Pearson 相关性检验

		文本平均长度	标签编码
文本平均长度	皮尔逊相关性	1	-0.070
	显著性（双尾）	-	0.717
	个案数	29	29
标签编码	皮尔逊相关性	-0.070	1
	显著性（双尾）	0.717	-
	个案数	29	29

2.3 抽取标签分析

对抽取任务的标签进行内容分析，绘制抽取文本的词云图。抽取的内容均为公司实体名称，故高频字词主要集中在“股份”“有限公司”等。

图 7: 抽取标签文本的词云图

标签不均衡问题天然地存在于各大文本抽取任务当中，故需要统计三种标签的数量分布，以便调整代价敏感的权重。标注比例为 BI/BIO = 0.0725。

3 关键技术分析

3.1 词嵌入

本模型使用了预训练好的字向量，将输入的汉字通过字向量矩阵转换为词向量，从而输入到模型里面。通过 embedding 层，汉字不仅转化为了便于计算机进行学习的向量，而且语意相近的汉字还拥有余弦值相近的向量，这对于后续的训练起到了很大程度的辅助作用。

3.2 条件层随机场

CRF 是一种获取最优输出标签序列的模型，基本思路在预训练模型训练以后会形成多种标签序列，CRF 是基于这些组合方案选取序列最优的结果进行输出，给定序列 $x = (x_1\ x_2\ \ldots\ x_n)$ 和标签序列 $y = (y_1\ y_2\ \ldots\ y_n)$，评估分数可由如下公式计算。

$$s(x,y) = \sum_{i=1}^{n} (W \cdot y_i - 1, y_i + P_j, y_i)$$

P_i 定义可由如下公式计算得到。

$$P_i = W \cdot s \cdot h^{(t)} + b_s$$

训练集合的似然函数，P 计算表示原序列到预测序列对应的概率。可由如下公式计算得到。

$$f(z) = \sum_{n=0}^{\infty} \log\left(P\left(y_i|x_i\right)\right) + \frac{\lambda}{2}\|\theta\|^2$$

$$P(y|x) = \frac{e^{s(x,y)}}{\sum_{n=0}^{Yx} e^{s(x,y)}}$$

CRF 目前广泛使用于命名实体识别领域，作为取最优标签序列的手段。这跟 CRF 作用的不可替代性有着极大关系，模型训练后会句子会产生不同的排列顺序，CRF 会选取其中最有可能的顺序输出。BiLSTM、IDCNN、Bi-GRU 模型的主要功能是进行训练，为了提高文本中命名实体识别的精确度，可以与 CRF 结合形成 BiLSTM-CRF、IDCNN-CRF、BiGRU-CRF 的模型结构。

3.3 深度网络特征抽取

3.3.1 循环神经网络

循环神经网络（Recurrent Nueral Network, RNN）

前馈神经网络以单向方式进行信息传播，从而使得网络结构更加容易计算，但也消弱了神经网络的表达能力。当处理文本数据时，比如词语、句子和文档，存在着联系上下文之间的时序特征关系。因此在 NER 任务中，需要一种性能效果更强的神经网络。循环神经网络 (Recurrent Neural Network，RNN) 是一种具有短期记忆能力的神经网络模型。循环神经网络通过对神经元接收自身的信息，而形成一个环路的网络结构，从而可以实现对文本序列中下文位置信息的记忆与处理。

由于 RNN 中存在梯度爆炸或消失现象，从而导致神经元的短期记忆。因此，如果在某时刻的输出依赖于间隔时长时刻的输入，那么当间隔时间比较长时，简单循环神经网络很难建模这种长距离的依赖关系，被称为长期依赖问题。长短期记忆 (Long Short-Term Memory，LSTM) 网络由 Hochreiter 和 Scmidhuber 在 1997 年首次提出，用于解决 RNN 的梯度爆炸或消失问题。在 LSTM 神经网络中，记忆单元通过捕捉到关键信息并将其在一定的时间内存储保留。由于记忆单元保存信息的时间要大于短期记忆，但又要小于长期记忆，从而被称为长短期记忆。LSTM 只能保留“过去”的信息，即只能正向提取句子中的词汇、语义信息假设在时刻 t 的输入向量为 x_t，前一时刻的输出为 a_{t-1}，前一时刻的隐藏状态为 c_{t-1}，则当前时刻的状态 c_t 和输出 a_t，如下所示：

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g\left(w^c x + b^c\right)$$

$$a_t = o_t \cdot s\left(c_t\right)$$

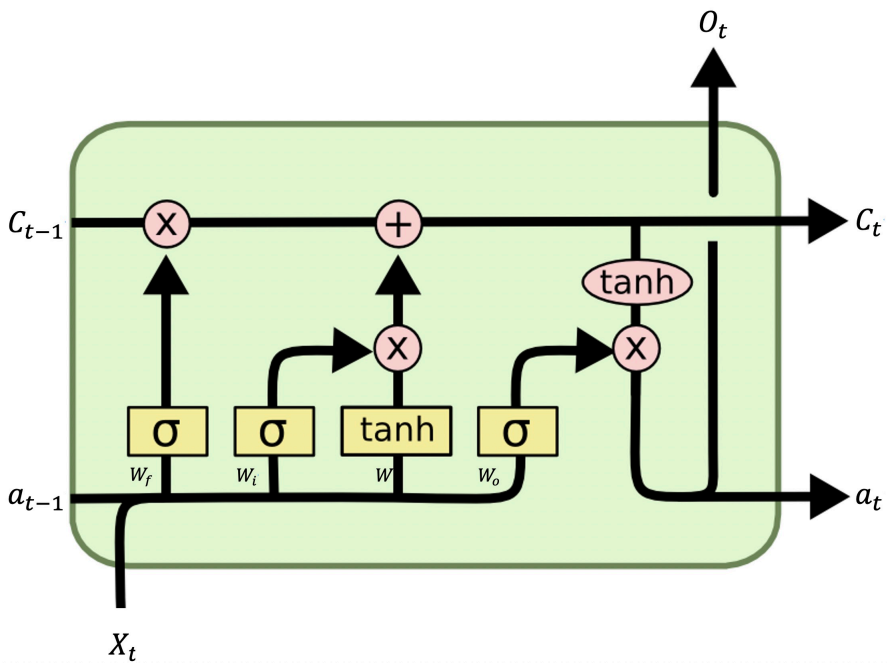

图 8: LSTM 神经元

其中，当前时刻的输入 X 由输入向量 x_t 与前一时刻的输出 h_{t-1} 组成，w 为权重，b 为偏置，g、s 分别表示状态的输入和输出的激活函数，x_t 、f_t、o_t 分别表示输入门 i、遗忘门 f、输出门 o 在 t 时刻的激活值。

$$X = \left[\begin{pmatrix} x_t \\ a_{t-1} \end{pmatrix} \right]$$

$$i_t = \sigma\left(w^i X + b^i\right)$$

$$f_t = \sigma\left(w^f X + b^f\right)$$

$$o_t = \sigma\left(w^o X + b^o\right)$$

$\sigma(*)$ 是区间为（0，1）的 logistic 函数，W 为状态-输入权重矩阵，U 为状态-状态权重矩阵，x_t 为当前时刻的输入，$h_(t-1)$ 为上一时刻的外部状态。当 f_t=0,i_t=1 时，遗忘门的信息被删除，输入门的信息全部保留，记忆单元将历史信息清空，并将信息写入候选状态向量中。当 f_t=1,i_t=0 时，记忆单元将历史信息写入，不再更新新的信息。

双向长短期记忆网络（Bi-drectional Long and Short Memory, Bi-LSTM）

由于在命名实体识别的任务中，信息的输出将受到历史信息和后续信息两个方面影响。比如在一段文本语句中，某个多义词的含义受到该文本的上下文影响。因此，在 NER 任务中，LSTM 网络需要新增一个以先后时序来表达上下文信息的网络层，从而提高神经网络的性能。LSTM 只能保留“过去”的信息，即只能正向提取句子中的词汇、语义信息，而 BiLSTM 能在访问“过去”的信息的同时，访问“未来”的信息，即能从正向、反向两个方向提取句子中的词汇、语义信息，得到更丰富、更深入的信息，对于相似句子识别任务是非常有益的。

双向长短期记忆神经网络由前向和后向的两层 LSTM 神经网络组成，对前后 LSTM 以相同的信息输入但以相反的方向进行信息传递。前后 LSTM 分别按时间顺序和时间逆序，在时刻 t 时的隐状态分别定义为 $\overrightarrow{h}$ 和 $\overleftarrow{h}$，其中 $\overrightarrow{h}$ 被称为前向 LSTM，$\overleftarrow{h}$ 被称为后向 LSTM，公式如下所示。

$$\overrightarrow{h}_t = f\left(U^{(1)} \cdot h_{t-1}^1 + W^{(1)} \cdot x_t + b^{(1)}\right)$$

$$\overleftarrow{h}_t = f\left(U^{(1)} \cdot h_{t-1}^2 + W^{(2)} \cdot x_t + b^{(2)}\right)$$

$$h_t = \overrightarrow{h}_t \oplus \overleftarrow{h}_t$$

其中 $\oplus$ 为向量的拼接操作，BiLSTM 的图像如图 9 所示：

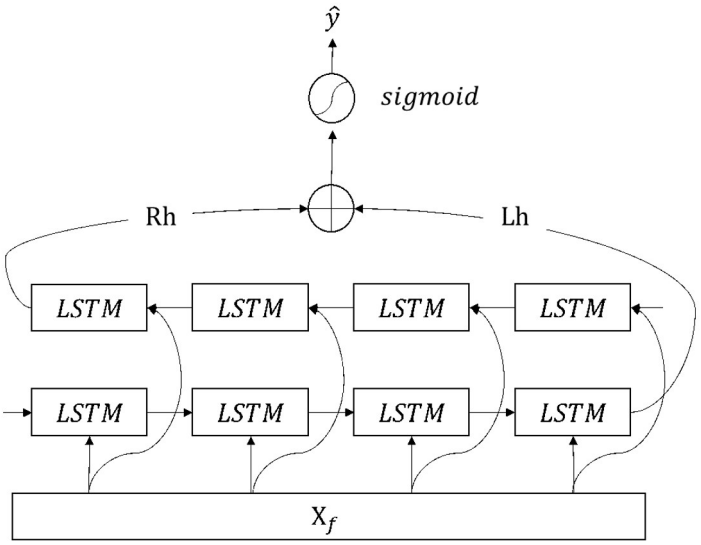

图 9: Bi-LSTM 结构图

门控循环单元神经网络（Gated Rcurrent Unit, GRU）

GRU 是 LSTM 的一种较为优秀的变体，它简化了 LSTM 的结构，加快了运算速度。前文中，我们介绍了 LSTM 的门结构，分别是输入门、遗忘门和输出门。而在 GRU 中，门结构进行了优化，只设置了两个门结构，分别是更新门和重置门。GRU 结构如图 10 所示。

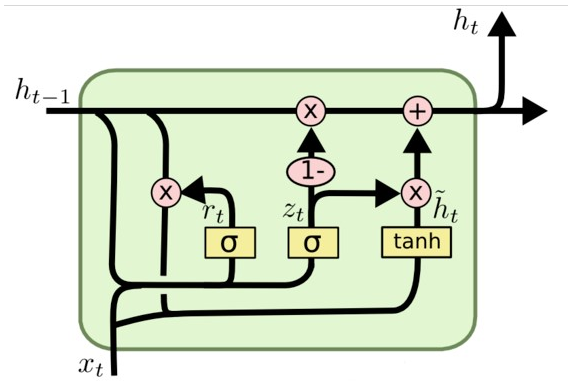

图 10: GRU 结构图

重置门

每个 GRU 单元的输入和输出结构与普通 RNN 是一样的。我们先获得上个单元的输出隐状态 h 和输出 x。令传入的隐状态 h 输入到重置门中，重置上一个单元传过来的隐状态，即选择性记忆，然后与传入的数据 x 进行拼接，得到 h'。在这一步中，重置门主要是对隐状态 h 进行清洗，这与输入门的功能相似。

更新门

更新门的作用是进行记忆和遗忘。将重置门传过来的数据 h' 和上个单元的隐状态 h 传入到更新门中，使用近似与 $(1-z)h' + zh$ 的公式进行计算。可以较为明显的看出，z 越接近 1，代表记忆下的数据越少，遗忘的很多；越接近 0 则表示相反。经过更新门后，我们就得到了最后的输出，和每一层的隐状态 h，后续继续对这两个值进行进一步处理。从整体上来看，更新门相当于 LSTM 中遗忘门和输出门的结合，起到记忆和遗忘的作用，是 GRU 中的核心。

相对于 LSTM 来说，GRU 少了一个门，参数没有 LSTM 多，但是也能达到与 LSTM 相近的功能。但是，考虑到计算能力和时间成本，很多时候 GRU 会更加实用。

3.3.2 卷积神经网络

CNN 是一种深度学习的代表算法，它通常由卷积层、池化层、全连接层三部分构成。其中，这三层可以根据实际需求重复进行。下文将简单的介绍这三层的具体结构，以及起到的作用。CNN 结构图如图 11 所示。

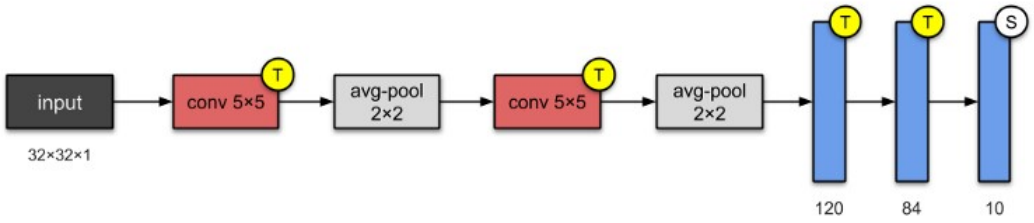

图 11: CNN 结构图

卷积层

卷积层的作用是提取输入向量中的特征信息，主要的方式是通过一个 filter（过滤器）在向量矩阵中不断以一个固定的步长进行移动，每次移动都与当前的矩阵做一个点积，最终遍历完整个矩阵，得到一个新的结果。

可以注意到，如果卷积前是 6*4 的矩阵，经过 3*3 的卷积核后，会变成了 4*2，与输入前的模型形状不同。因此，为了防止这种情况的发生，我们需要在最后的结果周围补零，让它的大小保持不变。

这种方法叫做相同填充，也就是让矩阵在卷积前后的 size 保持一致，如果不进行填充，则叫做“有效”填充。

在做完卷积操作后，为了让数据具有更强的拟合能力，我们需要对矩阵进行一个 ReLu 变换，让所有取值都大于 0。然后将矩阵输入到池化层中。

池化层

相比于卷积层来说，池化层的工作相对简单一些，池化的目的是降低不必要的冗余信息。

首先要说明的是，池化层最常见的有两种池化操作，一种是求平均，一种是求最大。在这里，我们采用求最大的方法。我们选用 2*2 大小，步长为 2 的 filter 进行池化操作。

经过池化后，矩阵丢掉了冗余信息，缩小了数据规模。这样的操作并不会丢失重要信息，在第一步的卷积操作中，一些不重要的值，对应的权重本身就比较小，因此在池化过程中去掉影响不大，反而可以在一定程度上增强鲁棒性。经过池化后，我们通过只保留影响比较大的项，不仅让矩阵的冗余性降低了，还减小了矩阵的大小，让我们的程序运行效率提高了很多。

3.3.3 Transformer

Transformer 是一种主要应用注意力机制进行训练的模型，它抛弃了 CNN，RNN 的网络结构，使用注意力机制进行训练，且起到了较好的效果，其结构图如图 11 所示。下面将介绍它的多头注意力机制是如何运行的。

首先对注意力机制做一个简单的介绍。当获得了一句话时，模型会建立这句话中每一个单词对其他单词的关系，这与 rnn 的思路有一定的相似之处。具体来说，模型首先获取 embedding 层传入的一句话所有单词的词向量，然后依次遍历，让每一个词向量分别与 W_q、W_k、W_v 三个矩阵相乘，得到这个单词的三个向量，分别是查询向量，键向量和值向量。计算公式如下。

$$attention = soft\max\left(\frac{dot\left(W_q \cdot Q, W_k \cdot K\right)}{\sqrt{d}} W_v \cdot V\right)$$

第二步就是对这些单词进行打分。假设现在对第一个单词进行打分，那么就需要将第一个单词的查询向量与所有单词的键向量分别相乘，得到它对其他单词的重视程度。然后将这个乘积进行 softmax，转化为和为 1 且都为正数的值，然后分别乘以它们的值向量并求和，得到第一个单词的输出。

需要主要的是，transformer 使用的是多头自注意力机制，也就是对上述过程并行重复 N 次，因为权值矩阵的初始化有一定随机性，所以得到多个不同的输出，然后将这不同的几个矩阵拼接在一起，就构成了多头自注意力机制的输出。

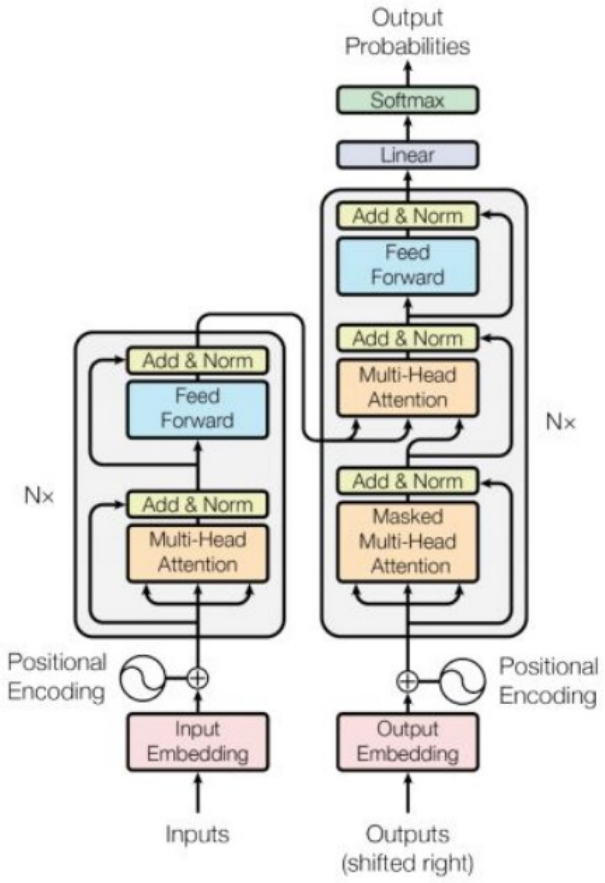

图 12: Transformer 结构图

3.3.4 BERT

BERT 模型的基本原理是利用模型各层中的上下文来进行深度双向预训练。BERT 模型结构主要基于 Transformer 结构中的 Encoder，利用 12 层和 24 层 Transformer 中的 Encoder 组装生成 2 套 BERT 模型。BERT 模型输入内容包括标记嵌入、段嵌入与位置嵌入。标记嵌入是文本的词向量表示。第一个字是 CLS 标志，是表示句子的开头标志，用于后续的分类任务。段嵌入对单词属于哪个句子作标记，以方便后续的模型训练。位置嵌入代表每个字的顺序，从而保证训练过程中字段的顺序不会出现错误。BERT 模型的训练一般是基于少量的文本数据，这也是目前预处理领域的一个趋势，即用更少的训练集获得更多的特征，更好的训练效果。如图 13 所示（Jacob 等，2019）。

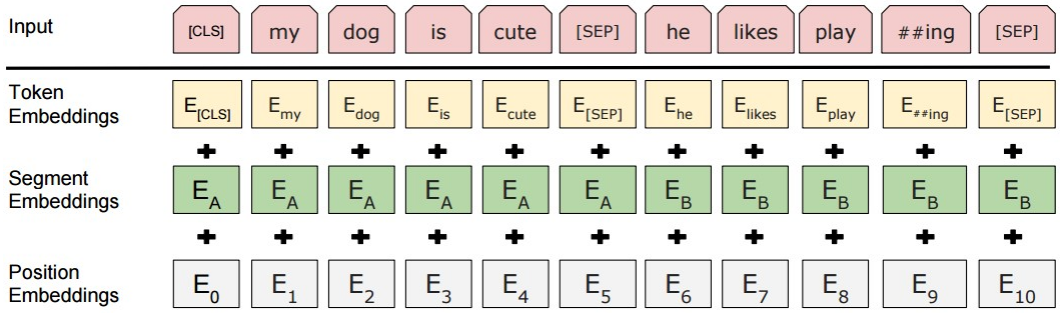

图 13: BERT 模型输入表示

BERT 模型处理核心有 2 个，分别是“词句遮罩”和“下一句预测”。“词句遮罩”的处理过程是在一句话中随机选择 15% 的词汇提取出来用于预测，随机抹除一部分数据，80% 概率使用一个特殊符号 [MASK] 替换，10% 概率使用一个任意词替换，剩余 10% 概率保持原词汇不变，整个过程是动态的。“下一句预测”的处理过程是给定目标文本的两句话，判断第二句与第一句之间的关系，再通过参数微调过程，对序列化标签进行分类，BERT 直接取第一个 [CLS]token 的最终隐藏状态 $C \in \mathrm{Re}^{H}$，加一层权重 $W \in \mathrm{Re}^{K \times H}$ 后预测标签:$P = softmax(CW^{T})$。BERT 模型结构如图 14 所示（Jacob 等，2019）。

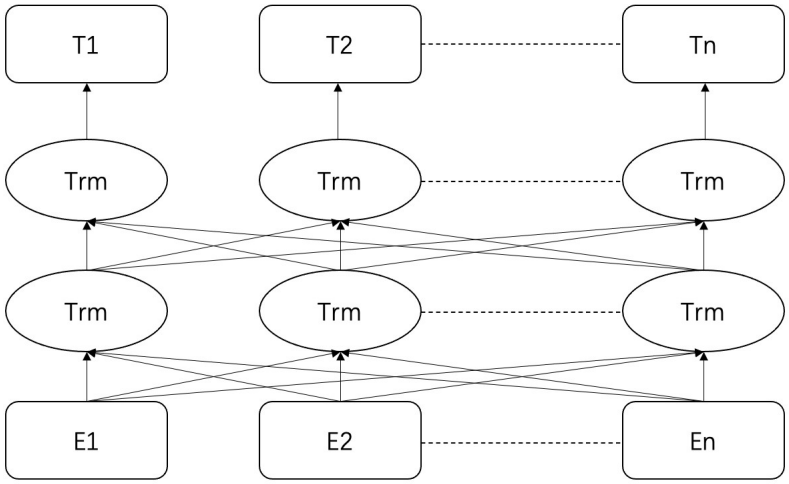

图 14: BERT 模型结构

BERT 模型经过测试，训练效果相比其他预训练模型有着极大提升，在 11 种不同 NLP 任务测试中得出最佳成绩，将极大似然不确定性估计方法基准提高至 80.4%，多源自然语言处理模块准确度提高至 86.7%（绝对改进率达到 5.6%）。在模型的改进中，该模型经常被用来当做预处理模型，获得语义的向量表示。比较常见的用法是在模型结构前增加一层 BERT 结构用于预训练，比如常用于命名实体识别的 BiLSTM-CRF、CNN-CRF、IDCNN-CRF、BiGRU-CRF 等模型都可以通过增加 BERT 层提高模型训练效果。BERT 的结构是相对固定的，针对特定任务为了提升效果，会对 BERT 结构进行一定程度的改写，从而更好适用于一些文本任务，提高模型的训练效果。

4 模型调优方案

4.1 训练层

4.1.1 训练策略

在训练模型的过程中，主要采用分层学习以及动态学习率的方法，对模型进行训练。对于本项目的抽取模型，可以将其划分为词嵌入层、特征提取层和标注预测层。由于不同层级之间存在差异性，故对不同的层使用不同的学习率，以满足更精细化的训练需求。

另外，为了避免模型在训练过程中陷入局部最优或者局部平坦区域，使用带热重启的余弦退火算法对学习率进行动态调整。在进行大轮数模型训练时，该方法可以有效地提升模型泛化性能，尤其在后期训练当中可以通过震荡学习率使模型脱离局部最优。带热重启的余弦退火调整学习率效果如图所示。

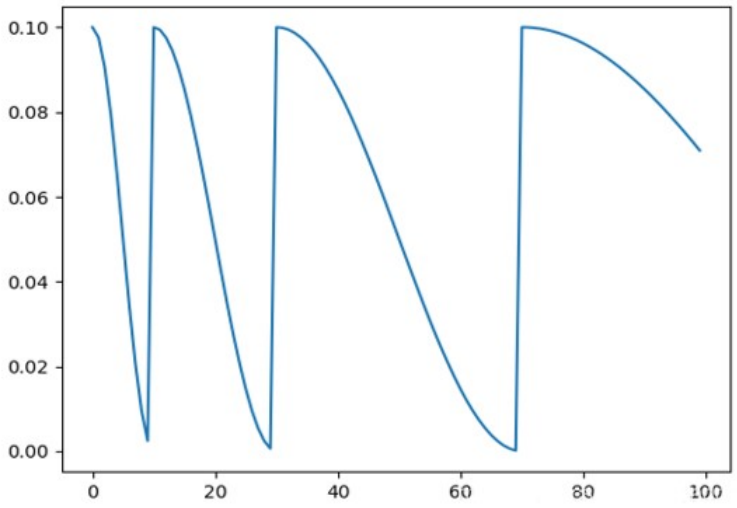

图 15: 带热重启的余弦退火调整学习率效果

4.1.2 参数调优

本项目在模型构建完成后，首先进行了试运行，寻找结果存在的问题及优化方向。在第一次的试运行中，结果的特征相当明显，几乎所有的字都被打上了 O 标签。也就是说，它没有学习到任何一个原因或者结果。针对这种现象，较容易发现是 O 标签所占比例过大带来的影响。作为应对，于损失值处使用代价敏感，增大将其他标签判断为 O 标签的损失值。在多次调整后，因果的预测结果有了较好的改善。

在使用代价敏感后，试运行后的结果依旧不够理想。初步推测是因为迭代次数不够多，因此通过租用的主机运行了 200 轮 epoch 后，结果的 loss 有了较高幅度的减小，并且收敛的效果较好。

在模型初步训练好后，使用测试集进行结果预测，发现效果并不好。对比训练集的低 loss，推测 dropout 的参数设置过低，导致过拟合现象的发生。将 dropout 从 0.2 提高到 0.5 后，测试集效果明显提高。

4.2 模型层

4.2.1 预训练词向量

在模型的词嵌入层当中，项目使用预训练的金融领域语料词向量。首先对现有数据建立词袋，并在金融新闻网站上抓取大量公告文本，扩充原始的词袋。之后将词袋中所有词对应的词向量从原始模型抽取出来，建立新的词向量嵌入词典。词向量抽取使得模型在前向传播时，向量检索速度提升，并减小内存和显存的占用率。

4.2.2 基准模型及提升

对于模型提升，本项目采用增量法对模型进行调优。首先确定基准模型为 Bi-LSTM+Softmax，之后发现 Softmax 层的标注出现逻辑错误，故修正为 Bi-LSTM+CRF。之后依据参数调优策略，选择合适的训练轮数以及各层的学习率。对输出结果进行检查发现模型泛化性能还需提升，之后为了增强模型非线性输出能力，对语义向量以进行前向编码，增强其表达能力。对比结果显示模型的损失达到最小，泛化能力有所提升。

5 对比实验

5.1 评价指标

F1-measure 是分类任务中常用的评价指标，本文的分类任务为二分类任务，计算公式如下：

$$F1 = \frac{2 \times P \times R}{P + R}$$

其中 P 为准确率，R 为召回率。准确率侧重模型的精度，召回率侧重模型的广度。通常情况，精度高的模型往往召回能力差，召回能力好的模型往往精度不够。$F1 - measure$ 是两者的调和平均，能够综合地反映模型的分类性能。本文所提出的模型结构分别从模型的精度和广度落脚，召回层提升召回率，预测层提高准确率。

对于多分类任务，我们可以在 n 个二分类混淆矩阵上综合考量准确率和召回率。本项目使用宏平均的方法来评估模型，计算公式如下。

$$macro - P = \frac{1}{n} \sum_{i=1}^{n} P_i$$

$$macro - R = \frac{1}{n} \sum_{i=1}^{n} R_i$$

$$macro - F1 = \frac{2 \times macro - P \times macro - R}{macro - P + macro - R}$$

5.2 实验结果

5.2.1 实验一（文本分类）

本次实验总共使用了 5 个模型进行对比分析。从测试集结果的准确率来看，LSTM 的发挥最为出色，CNN 次之，其他模型的发挥比较差。下面将从调参过程和测试结果对此次试验进行分析。

表 3: 文本分类对比实验

Method	$macro-Precision$	$macro-Recall$	$macro-F1$	$Accuracy$
LSTM	0.42	0.41	0.39	0.66
GRU	0.38	0.37	0.37	0.59
CNN	0.50	0.49	0.49	0.66
FastText	0.50	0.44	0.45	0.65
Transformer	0.40	0.41	0.40	0.56

Loss 不下降

在实验初期，此实验遇到的最大问题就是 loss 居高不下，并且最后只能预测一种或两种类别的数据。因此除了最开始选择的 CNN 模型之外，又尝试了 LSTM、Transformer、GRU、FastText 四个模型，并对模型的超参数和全连接层、embedding 层做了一些调整，但结果依旧不是非常理想。最后，终于从数据集中发现了端倪。因为通常的数据集都是呈随机分布的，因此只需要设置一个 dropout，然后按序读取数据即可。但是此数据集的数据都是按类别聚集在一起，因此在输入到模型中时，会导致某一类别同时大量输入，那么，模型就只学习到这一种类别。此时如果输入其他类别的数据，自然就会导致 loss 非常高。因此可以设想的是，最后模型只会学习到最后一个类别，因为学习到的其他类别都会因为高昂的 loss 而被舍弃掉。在给数据集加入了随机读取之后，loss 可以正常下降了。

准确率低

实验过程中遇到的第二个问题是准确率不够高，这个问题主要集中在 GRU 和 FastText 中。这两个模型的特点都是舍弃了一部分模型的准确度，追求处理速度。GRU 中将 LSTM 的三个门结构优化到两个，FastText 更是只对输入做了优化，真正的训练层也只有全连接层。因此，需要在保留其特点的情况下对这两个模型提升结构的复杂度。针对 GRU，在接下来的实验中提高了 GRU 的隐藏层数量，让它可以更好的对数据集进行拟合。而针对 FastText 则是在增大隐藏层数量的同时增加隐藏层大小，让它能够学习到更多的参数。在单独调参的基础上，后续实验调整了一些超参数，通过减小 batch size，增大 learning rate 来让拟合效果变强。

测试结果分析

从训练的情况来看，可以很明显的发现 Transformer 模型在 25Epoch 的时候就已经收敛，而其他模型则在 50Epoch 附近才收敛。从这方面分析可以得出，Transformer 模型的拟合性是最为优秀的，而其他模型稍次之。

但用测试集进行验证的时候，Transformer 的准确率却不够理想，反而是 Loss 最高的 LSTM 后来居上。这是过拟合的明显表现。在超参数没有调整到最优时，Transformer 的模型较好的性能反而拖了后腿。当然，这也表明 Transformer 有着更高的上限，在后续优化中拥有更好的效果。

目前模型的问题主要集中在过拟合上，针对这个问题，目前有以下几个优化方向。

调整 batch size: 增大 batch size 可以一次输入更多的数据，增大数据的差异性，一定程度上增大 loss，防止模型过于拟合。

模型复杂度: 减小模型的复杂度，如降低隐藏层层数，缩小隐藏层大小。这样做可以有效减少模型的参数，可以让模型不会过于贴合训练集。

调整 Epoch: 减小 Epoch 的数量，例如 Transformer 可以在 24Epoch 的时候 EarlyStop，终止训练。继续训练反而会容易学习数据集中的噪声数据和特征不明显的数据，降低泛化程度。

5.2.2 实验二（实体抽取模型实验）

本实验主要探究 BERT 不同参数在该命名实体识别任务上的综合抽取能力，实验总共分为 3 个部分：参数量对模型性能的影响、学习率调整对于优化过程的影响和一次训练的样本数对于模型的结果的影响。BERT 评价指标总括如下表所示：

表 4: 以 BERT+CRF 为 baseline 的系列实验

Experiments		$macro-Precision$	$macro-Recall$	$macro-F1$
Paramerter	$BERT+FC\omega=0.8$	0.51	0.38	0.40
	$BERT+FC\omega=0.9$	0.54	**0.42**	**0.45**
	$BERT+FC*5\omega=0.9$	**0.60**	0.41	0.43
Learning Rate	5e-5	**0.51**	0.51	0.51
	3e-5	0.50	**0.58**	**0.52**
Optmizer	16	**0.61**	0.55	**0.53**
	32	0.53	**0.56**	0.48

第一组实验表明，加上全链接层使得模型的 F1 等指标均有一定程度的提升，但是影响并不大。加上 5 层全链接层并不能提高该项目的总体能力。关于非 o 标签提高权重能够稍微增大一点点准确值和 f1 值。

第二、三组实验表明，学习率调小一次训练的样本数越小训练效果越好，准确率有一定的提升。但是效果仍然不明显，证明样本很容易产生过拟合现象。由于 BERT 的微调问题，很容易产生遗忘灾难等问题，需要将学习率调整较小的值防止已经学习的内容遗忘掉，有些比较好的方法是对不同层的网络设置不同的 learning rate。越靠近任务层的网络可以设置较大的 learning rate，而越是上层的通用的网络层 learning rate 偏向于较小值。

由此总结的最终参数设计如下：

表 5: 最终参数

参数	参数值
Batch size	768
Learning rate	3e-5
Epoch	40
Padding size	128

5.2.3 实验三（实体抽取参数实验）

本实验分为三个部分，通过对比实验分别探究模型参数量对模型性能的影响、学习率调整对于优化过程的影响以及不同优化算法对优化过程的影响。研究参数量时，设置两个参数量不同的模型，不进行学习率动态调整，使用 Adam 算法进行优化；研究学习率调整时则采用之前较小的模型，对比有无学习率调整下的模型性能及优化过程，使用 Adam 算法进行模型优化；研究优化算法的影响时同样使用参数量较小的模型，分别使用 3 个不同的优化算法对模型进行训练，不进行学习率动态调整。实验结果如下表所示。

表 6: 以 Bi-LSTM+CRF 为 baseline 的系列实验

Experiments		$macro-Precision$	$macro-Recall$	$macro-F1$
Paramerter	$BiLSTM*5+FC*5$	0.76	0.73	0.74
	$BiLSTM*15+FC*15$	**0.78**	**0.74**	**0.76**
Learning Rate	None	0.76	0.75	0.75
	CosineAnnealingWarmRestarts	**0.77**	**0.76**	**0.76**
Optmizer	Adam	0.76	0.75	0.75
	SGD	0.64	0.75	0.69
	RMSprop	**0.78**	**0.76**	**0.77**

模型泛化性能分析

从第一组实验结果可以发现，参数量的增加使得模型的 F1 等指标均有一定程度的提升，但是影响并不大。说明在本项目的数据集上，可以选择使用较轻量的模型，在后续的模型部署当中可以使得线上推理速度在保证精度的前提下有所提升。

由第二组的实验结果，学习率动态调整对模型最终的性能有一定的提升，效果同样不够明显。说明在使用本项目数据集进行训练时，优化过程没有遇到局部最优或者鞍点。

由第三组实验结果，RMSprop 优化算法在本数据集上取得了最优性能，而 SGD 算法的优化结果不太理想。

优化过程分析

由第二组实验，通过绘制学习率调整过程中 loss 曲线的变化，可以发现学习率重启时 loss 会上升，但是很快恢复到原有水平。虽然最终 loss 收敛情况一致，但是模型的泛化性能却更好。这表明在本数据集上，使用带热重启的余弦退火算法对学习率进行动态调整能够使模型泛化性能提升。loss 曲线图见附录图 16。

由第三组实验，绘制不同优化算法在训练集上的 loss 曲线以及验证集上的 F1 值曲线。可以发现 Adam 算法的收敛速度很快，在训练到第 10 轮时算法基本收敛。而随着训练轮数的增加，RMSprop 算法的 F1 值略微超过 Adam 算法的 F1 值。而 SGD 算法则没有找到最优解，最终在验证集上的 F1 只达到 0.7。loss 曲线图和 F1 曲线图见附录 17、18。

6 项目总结

- **石锴文个人总结:**

  本次项目的目标是完成一个文本分类任务和一个实体抽取任务。由于前期的积累，我对于这两个问题都有过一些了解，所以在项目当中我会更关注以往被忽略的细节问题。

  一方面是模型参数的设置，我在以往的项目当中积累了很多模板式的代码，模型的参数及配置通常会沿用之前的项目数据，更多的注意力则放在了模型的结构上。这一次的项目中，我无意中减小了 LSTM 和全连接层的隐藏层数量，结果发现模型收敛得很快，最终在测试上的泛化性能也不错。很显然，模型参数少，训练也更加容易，收敛速度自然就快。但是我总认为模型的深度越大越好，所以通常会忽略这个参数。而不同以往的是，标注数据量非常大，RTX 2080Ti 单卡的训练速度也非常缓慢，那么大参数模型就更不容易训练。那么在之后的项目当中，任何参数都需要花时间去调整，需要设计实验去选择最优的参数方案，否则只是完成了一次代码运行，毫无意义。

  另一方面是 GPU 的利用率，之前做项目的时候也会关注 GPU 以及 CPU 利用率的情况，在利用率不高的情况下，常用的提升方法（如设置 pin_memory 和 num_workers、增大 batch size）都有效果，而这次的项目却没有明显作用。初期验证方案的时候，只打算使用 5 千条训练数据，结果发现 256 的 batch size 都无法提升 GPU 的利用率。而最终的解决方案是同时训练两个模型，同时验证两个方案，从而将 GPU 利用率提高。其实想到多训练一个模型并不困难，只是一直在考虑如何通过参数调整来做提升，结果导致思路闭塞，没有做到整体考虑问题。

  除了以上两个项目中遇到的问题，最大的收获应该是模型并行训练的经验积累。在使用 Pytorch 的 DDP 时碰到很多问题，例如最初的无法运行、IO 输出控制还有对于并行训练的理解。DDP 的基本思路是将模型副本分发到各个 GPU 上, 再通过 DDP 封装的 DataLoader 将 batch 分发到各个 GPU 上完成反向传播，最后将所有反向传播的梯度规约到一个 GPU 上，并对模型进行优化和参数更新。由于 python 自带 GIL，python 解释器只能同时运行一个线程，故多进程管理是多卡并行训练的重要问题。而 DDP 最常用的 GPU 并行训练后端是 NCCL，主要完成张量数据的规约和进程通信。本次项目尝试使用 2 张 RTX 3090 显卡来完成所有数据的并行训练，但遗憾的是，增加数据量之后的模型泛化性能并没有特别明显的提升。也有可能是因为使用的测试集不同，导致测试性能存在差异。

  这次的项目管理任务主要由我负责，小组在假期的时候经历过前期项目的失利，复盘之后发现三个人的工作没有很大的交集，无法互相支撑，使得团队的工作效率低下。所以这次项目，我尝试重新分工，并在前期任务中负责对数据内容的分析，为后续的建模工作提供依据，增强成员对数据的理解。之前做项目的时候，往往是一个人完成所有工作，组建团队之后却发现分工极其困难，很难将一个完整的任务拆解，也发挥不出成员们的优势。这一次的项目较之前有所好转，但是我仍然觉得不够满意。经过反思，以前的思路是所有人完成同一个任务，其实应该让每个成员独立思考项目。按照以往的思路，每个人都完成流水线上的某一个任务，最后进行串联。但是深度学习任务的流水线通常很难拆开，这一次也仅仅是将数据接口部分分离出来。像模型的训练和验证，这些代码不可能分给不同的人来写。如果每个成员都负责完整的流水线，从项目的整体出发，那么至少个人效率会提高。之后就是不同流水线之间的通信交流，成员之间的工作合并起来才会有交互，最终能够做到相互支撑，使得团队的效率提高。至于这个想法是否有效，还需要在后续的工作任务中进行验证。

- **林卓凡个人总结:**

  互联网技术和互联网金融的快速发展，网络信息也越来越被重视。而由于互联网金融的发展迅速，互联网金融实体成指数倍增长，如何有效识别互联网金融实体成为一道难题。本文提出一种改进模型结构 BERT-CRF 并通过互联网金融实体可视化系统进行效果展示。具体来讲，主要的研究工作如下。

  基于 BERT 模型和 BERT-softmax 模型提出了改进模型 BERT-CRF，提高了互联网金融实体识别的效果。本文通过数据预处理和 BIO 标注将不同文本数据转化为统一格式，对 BERT 模型结构进行改写提高预训练模型效果。比较子模型结构的 P、R、F1 值，将 BERT-CRF 模型添加到子模型结构中，对模型训练性能进行优化，并通过增加迭代次数提高了训练效果，最后根据文字结果融合的形式获得互联网金融实体。通过测试集实验证明了本文所提出的模型相比于其他模型具有更好的性能。

  此次实验最大的收获就是对 BERT 模型有了更加深入的理解，之前仅会用 LSTM，GRU 等轻量的训练模型，基于更换语料的提升，第一次使用 Transformer 的大训练模型，从预训练到模型整合最后到参数调整的实现很多都和以前做过的 word-embedding 不一样，其中 BERT 中的 B 代表双向是指每个字都代表前后两个字的语义信息，命名实体识别很多都需要双向信息。在 encoder 的选择上，BERT 并没有用烂大街的 Bi-LSTM，而是使用了可以做的更深、具有更好并行性的 Transformer encoder 来做。这样每个词位的词都可以无视方向和距离的直接把句子中的每个词都有机会 encoding 进来。另一方面我主观的感觉 Transformer 相比 LSTM 更容易免受 mask 标记的影响，毕竟 self-attention 的过程完全可以把 mask 标记针对性的削弱匹配权重，但是 LSTM 中的输入门是如何看待 mask 标记的那就不得而知了。BERT 这里并没有像下游监督任务中的普遍做法一样，在 encoding 的基础上再搞个全局池化之类的，它首先在每个 sequence（对于句子对任务来说是两个拼起来的句子，对于其他任务来说是一个句子）前面加了一个特殊的 token，记为 [CLS]。这样就可以直接表明前后的位置信息

  构建互联网金融实体可视化系统对训练效果进行展示，系统功能包括互联网金融文本录入，舆情分类、实体抽取展示；通过实体抽取和词频统计的结果比较，直观展示了普通文本识别和面向互联网金融实体识别的区别。

  不足与展望

  现有的预训练模型都是在通用语料中训练得到的，并没有融入特定领域的信息对比 Bi-LSTM 的使用的是金融方向的预训练的词向量，准确度和 f1 值都有差距。

  其次训练轮数太少，模型训练效果受制于设备的限制，BERT 模型较大，训练时间和轮数都很少导致训练的结果不太理想模型文件太大，训练时间太长，一方面，这是因为 self-attention 的训练复杂度时 n 的 2 次方（set Transformer 解决这个问题）；另一方面，每轮只有 15% 的词汇预测大大降低了效率

  没有子模型的接入，只用到了 BERT-CRF，也就是 BERT 之后直接直接接全连接层。后期可以加上 BERT-BiLSTM-CRF 和 BERT-BiGRU-CRF 测试因此设计更多训练效果较好的子模型会对模型效果的提升有一定帮助。

- **樊世源个人总结:**

  本次实验中，在实验初期因为 loss 一直无法降低，所以尝试了很多种模型。但在各个模型中使用了各种调参方法后依然无法降低 loss，并且出现了只能预测 1-3 个类别的问题。查找各种论文后依然无果，在偶然间重新观察数据集后，发现所有的类别都是按照种类顺序排列，并没有进行打乱，这才恍然大悟。模型在连续学习同一个类别的数据后，就会很容易偏向这个类别，这时输入其他类别的数据就会

导致 loss 非常高。同理，这种模型也只会预测最后输入的那种类比。因此，在给数据进行 shuffle 后，loss 就能正常下降了。不仅如此，因为运用了很多深度学习的模型，我也对这些模型有了更深的理解，了解了更多的调参调优手段。

7 附录

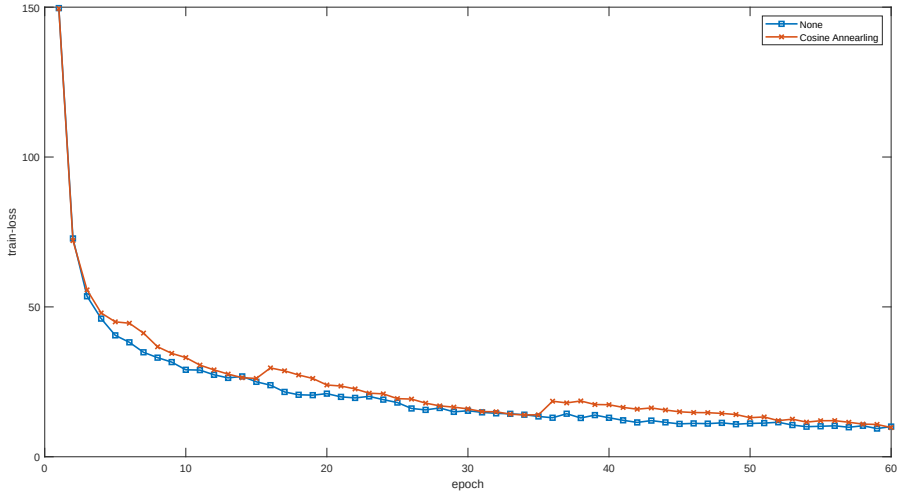

图 16: 实验三第二组实验的 loss 曲线

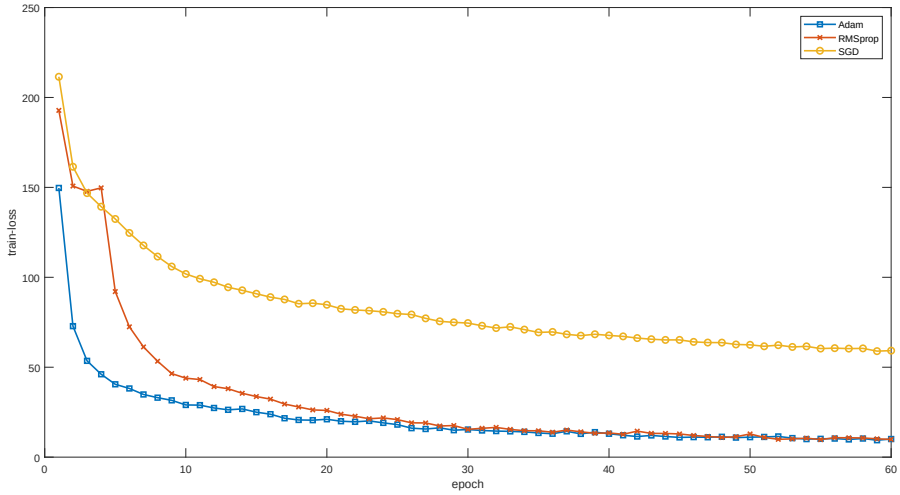

图 17: 实验三第三组实验的 loss 曲线

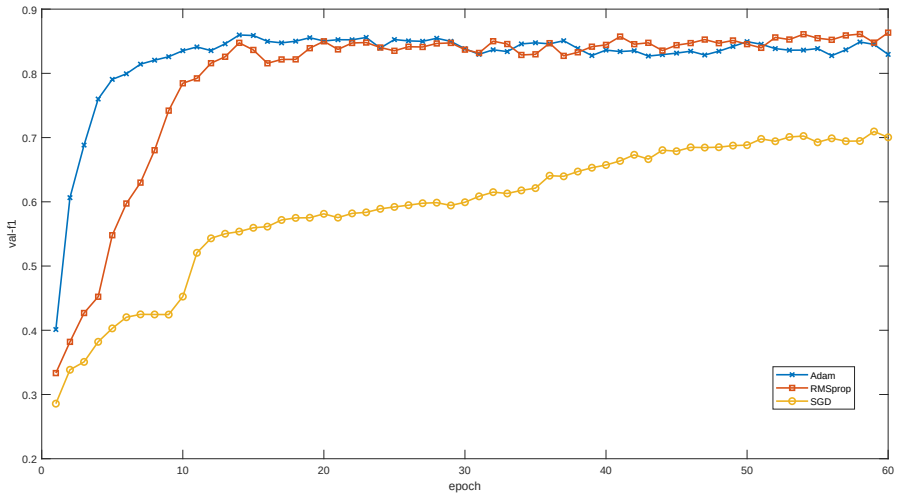

图 18: 实验三第三组实验的 val-F1 曲线

参考文献

[1] 吕学强，彭郴，张乐，董志安，游新冬. 融合 BERT 与标签语义注意力的文本多标签分类方法 [J/OL]. 计算机应用:1-8[2021-05-30]. http://kns.cnki.net/kcms/detail/51.1307.TP.20210519.1524.006.html.

[2] 陈步灿. 基于 BERT 模型的互联网金融实体识别研究 [D]. 北京林业大学,2020.

[3] AHN D. The stages of event extraction [C] //Proceedings of the workshop on annotations and reasoning about time and events. Sydney, Australia: Association for Computational Linguistics, 2006: 1 — 8.

[4] CHEN Y, XU L, LIU K, et al. Event extraction via dynamic multi-pooling convolutional neural networks[C]// Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing. Stroudsburg, PA: Association for Computational Linguistics, 2015: 167-176.

[5] ZHENG S, HAO Y, LU D, et al. Joint entity and relation extraction based on a hybrid neural network[J]. Neurocomputing, 2017, 257: 1-8.