互联网金融新实体发现

迎难而上

李涛
计算机科学与技术 硕士研究生
哈尔滨工业大学（深圳）
中国-深圳
17816873784@163.com

陈帅
计算机科学与技术 硕士研究生
哈尔滨工业大学（深圳）
中国-深圳
768693436@qq.com

黄源航
计算机科学与技术 硕士研究生
哈尔滨工业大学（深圳）
中国-深圳
1091491753@qq.com

许雯婷
计算机科学与技术 硕士研究生
哈尔滨工业大学（深圳）
中国-深圳
1004526056@qq.com

付沪豪
计算机科学与技术 硕士研究生
哈尔滨工业大学（深圳）
中国-深圳
13667007570@163.com

团队简介

团队由 5 名成员组成，分别是队长李涛以及队员陈帅、黄源航、许雯婷和付沪豪，目前均就读于哈尔滨工业大学（深圳）计算机科学与技术学院，此次比赛均以个人名义参加。团队曾经参赛获奖经历有 BDCI2018《汽车行业用户观点主题及情感识别》竞赛、DataFountain《汽车论坛消费者用车体验内容的判别与标注》赛题第一名、2018 美团点评细粒度文本分类第七名、Kaggle 《Jigsaw Unintended Bias in Toxicity Classification》竞赛、IJCAI-19《阿里巴巴人工智能对抗算法竞赛》、2019 n2c2（track2) 比赛第一名。

摘要

团队在初赛和复赛阶段均使用了两类实体识别模型，分别是序列标注模型和机器阅读理解模型。由于各种预训练模型，如 Google 公司提出的 BERT[1]以及百度公司提出的 ERNIE[2]，在多项自然语言处理任务中被证明是有效且通用的，尤其是在数据有限的情况下，预训练模型能够更快且更好的学习新数据集中的分布，因此在本次任务中团队也采用了讯飞公司提供的中文大语料预训练模型 RoBERTa-wwm-ext[3]作为文本表示的模型。

随着深度学习的发展，BiLSTM 这种 RNN 结构被广泛应用于序列标注的实体识别任务中，然而单纯的 RNN 结构在对每个字符的标签进行预测时，没有考虑到多个字符的标签之间存在的关系，所以往往会采用 CRF 结构对多个字符及其标签进行联合概率的建模，并采用维特比算法获得最大可能的序列标注结果。本文采用了 RoBERTa+BiLSTM+CRF 的模型作为序列标注的基础模型，并结合外部特征、模型微调 ITPT-FiT[4]、数据增强以及多任务训练等方式对模型进行改进。

机器阅读理解任务是给定一段文本以及一个问句，让机器通过阅读文本后，在文本中找到对应问句的答案，其前提是问句的答案需要出现在文本中，而实体识别任务正好符合这个前提，因此本文额外采用了机器阅读理解模型来完善序列标注的抽取结果，在复赛中使用了香侬科技提出的通用阅读理解模型[5]，并根据实际效果进行了相应的改进以符合任务的实际情况。

除了模型之外，数据处理也对最终结果存在很大影响，因此本文针对不同的模型进行了不同的数据预处理，并且对模型最终的结果也进行了相应的后处理，来提高结果的准确率和召回率。

关键词

预训练模型 数据增强 多任务学习 特征工程 模型融合 错误分析

1 数据和模型预处理

数据预处理部分包括数据清洗、数据切分、文本特征提取以及数据增强。数据清洗的主要目的是去除不重要甚至是有害的字符，包括：不常用的标点、网页相关的标签、年月日时间、微信号、emoji 表情、空白字符以及网络中提及某人使用的”@xxx:”类型的短语。数据切分针对的是训练及测试文本过长地问题，目的是更好地保留每个训练文本所具有的实体以及上下文信息，会将过长文本切分成多个相同长度的段落，在切分过程中，为了让同一个原始文本的上下文信息在切分后的段落中有所保留，使用了带有重叠的切分方法，具体来说，对一串过长的训练文本S，设定最长的段落长度为L（在实际使用中$L=512$）：

1） 若S的长度小于或等于L，将S视为段落，并返回，否则执行步骤 2）；

2） 切出段落$S[:L]$，并在$S[:L]$找到最后一次出现逗号“，”的位置p（如果不存在逗号则$p=L$），并让$S=S[p:]$，返回步骤 1）。

由于官方只提供了训练数据，我们在训练集中划分出了包含 100 个样本的数据作为验证集来作为线下的模型评估依据。在序列标注识别实体任务中，数据将被处理成 BIO 标注的格式，“B”表示实体的开头，“I”表示实体除了开头之外的部分，“O”表示非实体的部分，包括 RoBERTa 所需的特殊字符“[CLS]”、“[SEP]”以及填充使用的“[PAD]”，而机器阅读理解方式的实体识别任务中，主要标注出实体的开头“B”以及实体的结尾“E”，并将构成实体的位置对(B_i, E_j)记录下来作为训练标签。

文本特征提取主要采用了一些传统方法抽取字符级别的特征，来辅助深度学习神经网络的隐藏层表示，提取的特征包括：

1） 当前字符是否在标题中出现；

2） 是否是小写英文字母；

3） 是否是大写英文字母；

4） 是否重要，利用训练集的所有实体的集合M，得到所有出现在实体中字的频率表，并使用前K个频率最高的字作为重要的字；

5） 是否是标点；

6） 出现在文本中的相对位置；

7） 是否是在文本的前 100 个字或者后 100 个字的范围内出现；

8） 是否是数字；

9） 使用 jieba 分词工具，获得字所在词的词性；

10） 使用 jieba 分词工具，判断字所在词的相对位置，采用“BIE”标签区分，同时将“[CLS]”、“[SEP]”以及“[PAD]”视为特殊的词，不采用“BIE 区分。

数据增强目的是增加训练样本数，同时提高模型泛化能力，采用的操作是对训练文本中的字符进行随机替换，替换的规则是：

1）60%的概率不对当前段落进行替换，否则执行步骤 2）；

2）对当前段落的每个字符，90%的概率不进行替换，否则执行步骤 3）；

3）30%的概率当前词被替换为“[MASK]”；

4）剩余 70%的情况中，仅对中文字符进行随机替换。

模型的预处理主要参考了邱锡鹏提出的 ITPT-FiT[4]的方法，使用清洗后的初复赛训练集文本对原本的 RoBERTa-wwm-ext 模型进行任务上的微调，使用的训练轮数为 1，学习率为 2e-5，将得到的微调模型保存以供后续使用。

2 实体识别

2.1 基于序列标注的实体识别

基于序列标注的实体识别方法采用的是微调后的 RoBERTa+BiLSTM+CRF 以及融合提取的文本特征的方式，模型框架如图 1 所示。将 RoBERTa 模型最后一层的输出T_i，作为 BiLSTM 的输入$H_i = BiLSTM(T_{i-1}, Q_i, T_{i+1})$，并和数据

预处理阶段得到的特征经过线性层后得到的稠密的分布式表示F_i拼接得到每个字符的隐状态信息 $Q_i = [H_i; F_i]$，作为 CRF 层的输入，CRF 层的作用是对“BIO”标签之间的转移概率和发射概率进行约束，最终得到每个字符的标签 Logits G_i。

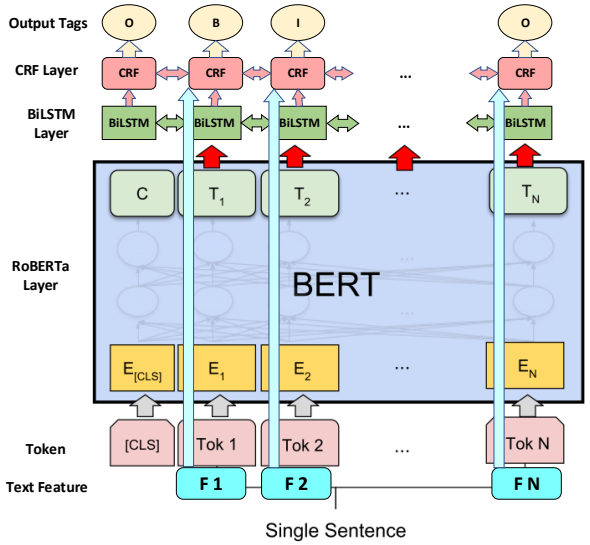

图 1：RoBERTa+BiLSTM+CRF 模型结构

在计算损失时用到了 CRF 的转移得分s_t和发射得分s_e：

$$Loss_{base} = -(s_t + s_e) \tag{1}$$

在序列标注任务中，本文同样采用了多任务联合学习的方式，来辅助和提高序列标注任务的效果。具体来说，使用数据增强得到的数据，在 RoBERTa 的输出T_i上做掩码语言模型任务（MLM），也就是预测被替换过的字符原本的字符。所以会额外增加 MLM 的损失值：

$$Loss_{mlm} = CrossEntropy(predict\ word, target\ word) \tag{2}$$

最终的损失值为：

$$Loss_{total} = Loss_{base} + Loss_{mlm} \tag{3}$$

在优化器的选择上，采用了 SGD+Momentum 的方式，将学习率调小到 8e-6，同时设置 batch size 为 8，momentum 设置为 0.5。使用以上方法可以将原本 RoBERTa+BiLSTM+CRF 方法的结果提高 6 个百分点。

2.2 基于机器阅读理解的实体识别

2.2.1 初赛模型

初赛中使用的机器阅读理解模型是 BERT 做 SQUAD 任务中所采用的方法，本文构造了问句“*有哪些金融公司、平台、中心、投资、币、银行、基金、外汇、集团、链、股份、商城、店、资本、家园、金服、交易所、理财、贷款*”，并与训练文本进行拼接，利用 RoBERTa 在位置i的输出T_i得到起始向量$S_i \in R^H$和结尾向量$E_i \in R^H$，由此计算位置i为实体起始位置的概率：

$$P_i = \frac{e^{S_i \cdot T_i}}{\sum_j e^{S_i \cdot T_j}} \tag{4}$$

同理可以得到结束位置的概率，最终起始和结束位置概率和最大的范围内的词作为实体。

2.2.1 复赛模型

初赛中机器阅读理解能成功的原因是，标注人员标注的实体偏少，所以提取一个实体能够对序列标注的结果起到补全，然而在复赛中，数据标注的实体偏多，模型无法很好的处理段落中存在多个实体的情况，因此结果会很差，所以复赛中本文参考了香侬科技提出的通用阅读理解模型[5]来做实体识别。其基本思想是：分别预测实体的开始位置、结束位置以及从开始位置到结束位置是实体的概率，如图 2 所示，所以一共存在三部分损失：

$$L_{start} = CrossEntropy(P_s, Y_s) \tag{5}$$

$$L_{end} = CrossEntropy(P_e, Y_e) \tag{6}$$

$$L_{span} = CrossEntropy(p_{ij}, y_{ij}) \tag{7}$$

最终的损失值为三者相加。

然而本文在实现时发现，模型无法识别出实体，分析后发现问题出现在L_{span}上，因为实体跨度中起始位置必定出现在结束位置之后，所以需要对跨度概率矩阵中的下三角部分进行掩盖，使其不参与损失函数的计算，也就是设置其权重为 0，同时为了保证实体能够被更好的召回，将$y_{ij} = 1$处的权重调整为 10，最终跨度矩阵的权重如图 3 所示。

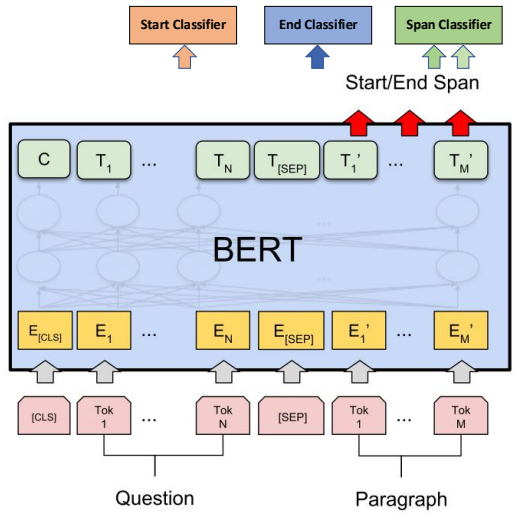

图 2：通用机器阅读理解模型结构

	来	赶	街	，
来	1	1	1	1
赶	0	1	10	1
街	0	0	1	1
，	0	0	0	1

图 3：句子“来赶街，”对应的跨度权重矩阵，不可能的跨度的权重被设为 0，真实实体“赶街”的跨度(1,2)权重被设为 10

该模型的存在的另一个问题是，计算跨度损失$Loss_{span}$没有考虑字符所在的位置信息，所以当前后存在相同字且作为边界时，会识别出多个实体，而长的往往是错的，例如，真实的实体为“*5050 社区众筹*”，模型的确识别到了“5”和“筹”为开始和结束字符，但是由于段落中实体多次重复出现，所以模型最终会识别出“*5050 社区是短期项目还是长期项目？【揭秘】5050 社区众筹*”以及“*5050 社区众筹*”。为了解决这个问题，在解码实体过程中，只考虑最近的两个边界，使得只得到“*5050 社区众筹*“。

在实际预测阶段，还发现预测出许多非实体的结果，所以最终使用时还调高了识别为实体的阈值。

3 模型融合及后处理

模型融合，分为两步，序列标注模型自身的融合，也就是训练过程中取验证集上结果最好几个模型，对参数取平均，得到融合后的结果，让模型更稳定。对于多个模型结果的融合，本文也尝试过很多方式，包括多个模型的 BIO 标签投票，多个模型实体投票，但是效果都不好，而且基本都是降分的，最终使用的是序列标注模型和阅读理解之间取并集。

本文尝试过多种后处理方式，其中扩展实体的简称或全称的方式是无效的，失败的很大原因是标注的不统一（比如有些数据标出了平台结尾，有些没有）。序列标注采用的一个后处理方式是对英文边界的扩展，因为有些包含英文的实体边界存在异常，比如丢失了前面的英文字母，所以采用的处理方式是如果识别出的实体的第一个字是英文字母，且在原文中实体的前一个字也是英文，那就把实体边界前移，直到遇到不是英文字母就停止，同理对于丢失后面的英文字母的情况也采取类似的处理措施。同时还进行人工收集了一些肯定不是实体的结果的集合，作为筛查表，并将每次预测的结果中存在于筛查表的实体进行剔除。

致谢

感谢 CCF BDCI 提供的这次比赛机会。感谢 PyTorch[6] 框架的开发者们，以及 transformers[7]库的开发者们，他们的工作极大地方便了我们的代码实现。

参考

[1] Devlin J, Chang M-W, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[C]//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). 2019: 4171–4186.

[2] Sun Y, Wang S, Li Y, et al. ERNIE: Enhanced Representation through Knowledge Integration[J]. .

[3] Cui Y, Che W, Liu T, et al. Pre-Training with Whole Word Masking for Chinese BERT[J]. arXiv preprint arXiv:1906.08101, 2019.

[4] Sun C, Qiu X, Xu Y, et al. How to Fine-Tune BERT for Text Classification?[J]. arXiv:1905.05583 [cs], 2019.

[5] Li X, Feng J, Meng Y, et al. A Unified MRC Framework for Named Entity Recognition[J]. arXiv preprint arXiv:1910.11476, 2019.

[6] Paszke A, Gross S, Chintala S, et al. Automatic differentiation in PyTorch[C]//2017.

[7] Wolf T, Debut L, Sanh V, et al. HuggingFace's Transformers: State-of-the-art Natural Language Processing[J]. ArXiv, 2019, abs/1910.03771.