文本数据挖掘 课程作业

汉语命名实体自动识别

# 1.项目目标

近年来，命名实体识别任务范式逐渐由基于序列的标注任务转向基于实体对的分类任务，但对于以上两种主流方式的优劣以及适用条件，没有进行系统的实验对比分析，在理论上，CRF作为基于序列标注的典型方法之一，直观的解释，是完成了次分类任务，而基于tokens pair的方式，理论是存在种候选实体，可以解释为需要完成次的多分类任务。本文主要从实验结果分析角度，针对命名实体识别任务中的不同方法进行比较，分析方法优劣。

# 2.国内外相关工作

命名实体识别旨在将文本中的命名实体定位并分类为预先定义的类别，如人员、组织、位置、时间表达式、数量、货币值、百分比等，但由于实体数量巨大、类型灵活多变，并存在缩写、简写、错写等情况，导致命名实体识别存在一定的难度。在深度学习之前，主要采用基于特征工程的统计学习方法，如HMM算法，但特征选取耗时耗力，逐渐被神经网络取代。Guillaume Lample[1]等人提出了基于Bi-LSTM+CRF的命名实体识别系统，每个句子按照词序逐个输入双向LSTM中，结合正反向隐层输出得到每个词类别特征的表示，输入CRF中，优化目标函数，从而得到每个词所属的实体类别，但模型学习到的信息是有限的，实体相关的背景信息无法准确获取，因此，注入外部知识成为了新的突破口，如Zhang等人[2]提出了融入实体外部词典的方式、Xin等[3]将知识图谱作为新的知识库辅助模型学习等。另外，在社交媒体和短视频平台上，不但存在文本信息，还存在着大量的图片信息、视频信息、音频信息，仅利用单一的文本模态去推断命名实体的边界和标签，在社交媒体和短视屏这样的场景非常困难，需要借助其他模态的信息来共同推断，因此，多模态下的命名实体识别也是新的研究方向，Zhang 等人[4]与 Lu等人[5]在2018年分别在社交媒体上收集数据构建了Twitter2015 与 Twitter2017文本+图像双模态实体识别英文数据集，Sui等人[6]在2021年构建了第一个文本-语音双模态实体识别中文数据集等。

# 3.基于single token的命名实体识别系统

基于single token的命名实体识别模块，是针对句子中的每个单独的token判断标签类型，典型算法有：CRF、RNN、LSTM等，本文选用较经典的PLM+CRF和PLM+softmax的方式作为实体识别系统，其中，PLM（pretrain language model）在中文语料上，选用bert-base-Chinese，在英文语料中为bert-large-cased。

# 4.基于单个token的实体识别

## 4.1 CRF模块

条件随机场（Conditional Random Field，CRF）是给定一组输入随机变量条件下另一组输出随机变量的条件概率分布模型。在线性链条件随机场中，变量表示为线性序列，而命名实体识别是转化为序列结构的标注问题来解决的，因此，CRF能够有效的应用于该任务中。其具体指在给定随机变量序列为X的条件下，随机变量序列Y的条件概率P(Y |X)构成条件随机场，满足马尔科夫性，即

## 4.2系统模块

1. **句子编码模块**

在给定句子X={x1,x2,x3,…xn}的基础上（n为句子中的token的数量），采用预训练模型得到句子的向量表示。

,EMB为预训练模型。

1. **Token 预测模块**

首先在token表示方法上，每个token对应一个标签，如在“BIO”模式下，当句子为“清 华 大 学 坐 落 于 首 都 北 京”，其对应的标签序列为“B I I I O O O O O B I”，但针对多种实体类型时，标签为‘B-entity type’的形式，句子对应的标签序列为“B-ORG I-ORG I-ORG I-ORG O O O O O B-LOC I-LOC”。每个token的对应向量表示为。

1. **解码模块**

在token的类型预测时，采用两种方式，一是基于softmax的概率预测。

token的向量表示经过线性连接层得到对应每个类别的分值：

通过softmax计算token属于类别y的概率值：

二是基于CRF的概率预测，具体实验原理如上。

## 4.3损失函数

在采用softmax判别函数时，采用多分类交叉熵损失函数，尽可能使得目标类别概率最大，其表示为：

其中，当样本属于类别时，对应的y值为1，反之为0。

在采用CRF作为损失函数时，考虑一个句子中所有的类别标签序列，根据路径计算loss，表示为：

# 5.基于token pairs的命名实体识别系统

基于序列的方式，一般将token作为单独的预测目标，用两个模块分别识别实体的首和尾，这会带来训练和预测时的不一致。而基于token pair的表示方式，将首尾视为一个整体去进行判别，更有“全局观”。其基本思路是将实体作为实体识别的目标，比如，假设一个句子中的实体的位置我们已知，则对应的span表示为（entity start，entity end），而目标是直接判断这一span的实体类型。基于span的命名实体识别，主要包括如下三个模块。

1. **句子向量表示**

在给定句子X={x1,x2,x3,…xn}的基础上（n为句子中的token的数量），采用预训练模型得到句子的向量表示。

,EMB为预训练模型。

1. **span表示**

首先，我们列举所有可能的跨度S={s1, …, si, …., sm}，用于句子X={x1, …,xn}，然后为每个跨度s重新分配一个标签y∈Y，例如，对于句子，"书好看"，可能的跨度（开始，结束）是{（1，1），（2，2），（3，3），（1，2），（2，3），（1，3）}，这些跨度的标签都是 "O"，除了（1，1）（书）是 实体外，其余均为非实体。

在span表示上采用表的形式，比如type=book的实体类型下，表示为，

|  |  |  |  |
| --- | --- | --- | --- |
|  | 书 | 好 | 看 |
| 书 | 1 | 0 | 0 |
| 好 | 0 | 0 | 0 |
| 看 | 0 | 0 | 0 |

在span表示时，采用s,e表示一个span的start和end index，其中，s<=e<=n，每个span的表示为s=。

**（3）span的预测**

Span的表示是由开始和结束标记的表征串联计算出来的，具体为,在操作时，计算每个实体的得分，添加两个位置注意矩阵，，通过Wh的矩阵变换得到实体表示，每个实体得分为。（两个变换矩阵是为了使得实体向量的点积表示中包含位置，所以，W是一个变换矩阵，类似于Roformer中的相对注意力矩阵，参考<https://kexue.fm/archives/8265>）

1. **损失函数**

采用多分类交叉熵损失函数，因为表中只有0和1，尽可能另正样本接近于1，负样本接近于0，公式如下，其中P为正实例，Q为负实例，s为实体分值。

# 6.实验

## 6.1数据集介绍

中文数据集：Cluener, CLUENER数据集是在清华大学开源的文本分类数据集THUCTC基础上，选出部分数据进行细粒度命名实体标注，原数据来源于Sina News RSS，实体有：地址（address），书名（book），公司（company），游戏（game），政府（goverment），电影（movie），姓名（name），组织机构（organization），职位（position），景点（scene），其中训练集：10748条，验证集：1343。按照不同标签类别统计，训练集和验证集标签数据分布如下：

表1 训练集标签数据分布

|  |  |
| --- | --- |
| 地址（address）:2829 | 电影（movie）:1109 |
| 书名（book）:1131 | 姓名（name）:3661 |
| 公司（company）:2897 | 组织机构（organization）:3075 |
| 游戏（game）:2325 | 职位（position）:3052 |
| 政府（government）:1797 | 景点（scene）:1462 |

表2 验证集标签数据分布

|  |  |
| --- | --- |
| 地址（address）:364 | 电影（movie）:150 |
| 书名（book）:152 | 姓名（name）:451 |
| 公司（company）:366 | 组织机构（organization）:344 |
| 游戏（game）:287 | 职位（position）:425 |
| 政府（government）:244 | 景点（scene）:199 |

6.2 实验参数

针对不同的模型的实验参数设置如下，在CRF和softmax实验中，epoch设置为20，初始的lr为1e-5，采用线性变化的方式，调用在torch下封装的get\_linear\_schedule\_with\_warmup函数调节学习率，batch\_size设置为32，运行时长分为为3h、1h，loss呈现下降趋势（具体可参考notebook下保存的实验数据）。在Span实验中，batch\_size=32,，初始学习率为5e-5,epochs设置为10，运行时长约为1.5h。

# 6.3实验结果：

表3 模型效果表现

|  |  |  |  |
| --- | --- | --- | --- |
| 实验模型 | SpanNER | | |
| precision | recall | F1 |
| Span | 0.78 | 0.802 | 0.794 |
| CRF | 0.6008 | 0.6095 | 0.6052 |
| Softmax | 0.596 | 0.6068 | 0.6017 |

在三种模型下，Span的表现最佳，我认为是因为Span在预测时，考虑了实体的边界的实体信息，并相比于CRF和softmax基于单个token的预测而言，以token pair的形式表示实体，直接与最终的目标挂钩，并且，在训练时，span pair的方式添加了旋转矩阵，作为实体相对位置信息的表示，这可能也是表现较优的原因之一。

**（一）在不同实体长度下，模型的表现效果**

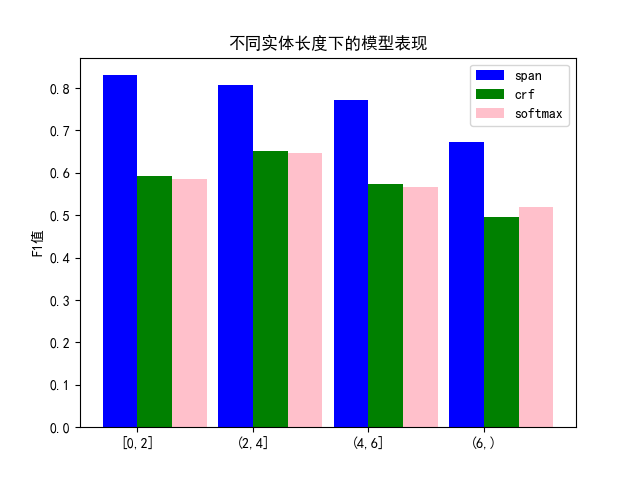


图1 不同实体长度下的模型表现结果

在中文数据集cluener中，根据实体的不同长度区间分为四类，分别为 [0,2],(2,4], (4,6], (6,)，在Span模型中，表现出实体长度越短，模型效果越佳的现象，在实体长度为2以下的区间内F1值为0.83，在实体长度高于6时，F1值为0.67，模型效果下降了0.15左右，这种情况的原因可能是因为span模型只考虑了实体头尾的embedding，在距离较长时，实体的语义信息可能在头尾中包含量递减（即头尾信息不能完整的表示实体整体的语义信息），导致模型效果下降。另外两个模型，softmax和CRF在实体长度为(2,4]时，表现效果最佳，在较短（（0，2]））和较长（（4，6]））时，F1值分别为0.59和0.57，在实体长度高于6时，F1值下降为0.50左右，这种情况可能是因为CRF和softmax以单个token作为单位时，可能能够较好的捕获较长的距离的词之间的关系，但超过一定距离时，模型无法捕获语义信息。

**（二）在不同句子长度下，模型的表现效果**

在cluener测试数据集中，句子的最大长度为69，根据句子长度分为三个区间，分别为[0,30),[30,50),[50,)，三个模型在句子长度区间<30的条件下，表现效果均最佳，句子长度增长时，效果略有下降，下降幅度大致相同，均为10%左右。

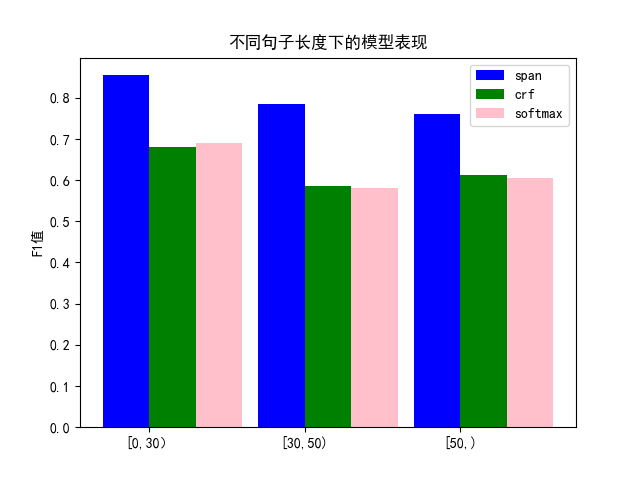


图2 不同句子长度下的模型表现

**（二）在不同实体类型下，模型的表现效果**

在cluener数据集中，数据集的类型分为以下10种，'address', 'book', 'company', 'game', 'government', 'movie', 'name', 'organization', 'position', 'scene'，在span模型中，对于类型为address的实体，模型效果最差， 这可能是因为实体长度较长导致的，在该类型下，实体的最大长度为32，平均长度为4.55。对于实体类型为game的实体，CRF和softmax的表现不佳，这可能是由于game实体长度过长导致的，该类实体的平均长度为5.59，模型可能无法捕捉长距离依赖关系，导致模型预测效果较差。

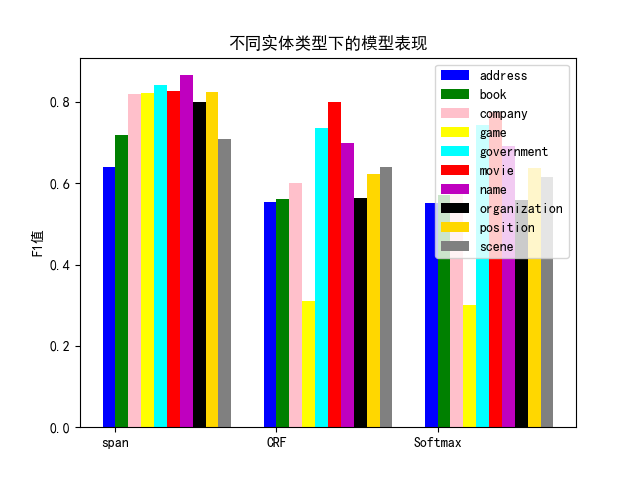


图3 不同实体类型的模型表现效果

# 7参考文献

[1] Lample G, Ballesteros M, Subramanian S. Neural Architectures for Named Entity Recognition[C]//Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, California: Association for Computational Linguistics, 2016：260-270.

[2] Zhang R, Hristovski D, Schutte D, et al. Drug repurposing for COVID-19 via knowledge graph completion[J]. J Biomed Inform. 2021, 115: 103696.

[3] Xin J, Yankai L, Zhiyuan L, et al. Improving Neural Fine-Grained Entity Typing with Knowledge Attention（NER+KB）[J]. 2018.

[4] Zhang Q, Fu J, Liu X, et al. Adaptive Co-attention Network for Named Entity Recognition in Tweets[C]//Association for the Advancement of Artificial Intelligence. 2018：5674-5681.

[5] Di Lu, Neves L, Carvalho V, et al. Visual Attention Model for Name Tagging in Multimodal Social Media[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, 2018：1990-1999.

[6] Sui D, Tian Z, Chen Y, et al. A Large-Scale Chinese Multimodal NER Dataset with Speech Clues[C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2021：2807-2818.

# 附录

在不同实体类型上，人的表现，如下官方给出的评测结果：

| **实体** | **Human Performance** | **实体** | **Human Performance** |
| --- | --- | --- | --- |
| Person Name | 74.49 | Game | 80.39 |
| Organization | 65.41 | Government | 79.27 |
| Position | 55.38 | Scene | 51.85 |
| Company | 49.32 | Book | 71.70 |
| Address | 43.04 | Movie | 63.21 |

## CRF的实验结果的具体数据如下：

1. 句子表示：

true data [{'text': '为逃避法律追究，他竟造假遗书说是自杀殉情。昨日上午，广州市中级人民法院公开审理了此案。', 'label': {'government': {'广州市中级人民法院': [[26, 34]]}}, 'id': 0}, {'text': '也同时刷新了苏富比自设名贵腕表拍卖以来的总成交额纪录。', 'label': {'company': {'苏富比': [[6, 8]]}}, 'id': 1}, {'text': 'banikostrava，港译：奥斯泰华）逼平，险些又从联盟杯中出局。平均赔率1.533.64', 'label': {'organization': {'奥斯泰华': [[16, 19]]}}, 'id': 2}]

pre\_data [{'text': '为逃避法律追究，他竟造假遗书说是自杀殉情。昨日上午，广州市中级人民法院公开审理了此案。', 'label': {'government': [[26, 34]]}}, {'text': '也同时刷新了苏富比自设名贵腕表拍卖以来的总成交额纪录。', 'label': {'company': [[6, 8]]}}, {'text': 'banikostrava，港译：奥斯泰华）逼平，险些又从联盟杯中出局。平均赔率1.533.64', 'label': {'organization': [[0, 0], [17, 19]], 'company': [[5, 8]]}}]

1. 语料中句子的最大长度为 69

预测的实体数量统计：total\_gold:1114.0, total\_pred:1130.0, total\_correct:679.0

不同实体长度下预测正确的的实体数 {1: 203, 2: 294, 3: 143, 4: 39}

不同实体长度下预测的的实体数 {1: 386, 2: 433, 3: 245, 4: 66}

不同实体长度下gold的的实体数 {1: 300, 2: 470, 3: 253, 4: 91}

不同句子长度下预测正确的实体数 {1: 123, 2: 446, 3: 110}

不同句子长度下预测正确的实体数 {1: 181, 2: 771, 3: 178}

不同句子长度下的gold实体数 {1: 180, 2: 753, 3: 181}

不同类型实体的预测正确的数量 {'address': 72, 'book': 32, 'company': 86, 'game': 31, 'government': 74, 'movie': 32, 'name': 128, 'organization': 66, 'position': 110, 'scene': 48}

不同类型实体的预测的数量 {'address': 72, 'book': 32, 'company': 86, 'game': 31, 'government': 74, 'movie': 32, 'name': 128, 'organization': 66, 'position': 110, 'scene': 48}

不同类型实体gold的数量 {'address': 72, 'book': 32, 'company': 86, 'game': 31, 'government': 74, 'movie': 32, 'name': 128, 'organization': 66, 'position': 110, 'scene': 48}

在实体长度为1的条件下的三个指标值precision, recall, f1 0.526 0.677 0.592

在实体长度为2的条件下的三个指标值precision, recall, f1 0.679 0.626 0.651

在实体长度为3的条件下的三个指标值precision, recall, f1 0.584 0.565 0.574

在实体长度为4的条件下的三个指标值precision, recall, f1 0.591 0.429 0.497

在句子长度为1的条件下的三个指标值precision, recall, f1 0.680 0.683 0.681

在句子长度为2的条件下的三个指标值precision, recall, f1 0.578 0.592 0.585

在句子长度为3的条件下的三个指标值precision, recall, f1 0.618 0.608 0.613

不同实体类型下的统计值,按如下顺序统计：

dict\_keys(['address', 'book', 'company', 'game', 'government', 'movie', 'name', 'organization', 'position', 'scene'])

precision： ['0.537', '0.593', '0.577', '0.290', '0.712', '0.865', '0.696', '0.595', '0.618', '0.667']

recall： ['0.571', '0.533', '0.623', '0.337', '0.763', '0.744', '0.703', '0.537', '0.629', '0.615']

f1: ['0.554', '0.561', '0.599', '0.312', '0.736', '0.800', '0.699', '0.564', '0.623', '0.640']

## softmax的实验结果的具体数据如下：

1. **句子表示：**

true data [{'text': '为逃避法律追究，他竟造假遗书说是自杀殉情。昨日上午，广州市中级人民法院公开审理了此案。', 'label': {'government': {'广州市中级人民法院': [[26, 34]]}}, 'id': 0}, {'text': '也同时刷新了苏富比自设名贵腕表拍卖以来的总成交额纪录。', 'label': {'company': {'苏富比': [[6, 8]]}}, 'id': 1}, {'text': 'banikostrava，港译：奥斯泰华）逼平，险些又从联盟杯中出局。平均赔率1.533.64', 'label': {'organization': {'奥斯泰华': [[16, 19]]}}, 'id': 2}]

pre\_data [{'text': '为逃避法律追究，他竟造假遗书说是自杀殉情。昨日上午，广州市中级人民法院公开审理了此案。', 'label': {'government': [[26, 34]]}}, {'text': '也同时刷新了苏富比自设名贵腕表拍卖以来的总成交额纪录。', 'label': {'company': [[6, 8]]}}, {'text': 'banikostrava，港译：奥斯泰华）逼平，险些又从联盟杯中出局。平均赔率1.533.64', 'label': {'organization': [[0, 0], [5, 9], [17, 19]]}}]

**（2）数据统计**

语料中句子的最大长度为 69

total\_gold:1114.0,total\_pred:1133.0,total\_correct:676.0

[[('government', 26, 34)], [('company', 6, 8)], [('organization', 16, 19)], [('scene', 19, 21)], [('company', 34, 36)]]

[[('government', 26, 34)], [('company', 6, 8)], [('organization', 0, 0), ('organization', 5, 9), ('organization', 17, 19)], [('scene', 19, 21)], [('name', 31, 31)]]

不同实体长度下预测正确的的实体数 {1: 204, 2: 292, 3: 140, 4: 40}

不同实体长度下预测的的实体数 {1: 396, 2: 433, 3: 241, 4: 63}

不同实体长度下gold的的实体数 {1: 300, 2: 470, 3: 253, 4: 91}

不同句子长度下预测正确的实体数 {1: 123, 2: 443, 3: 110}

不同句子长度下预测正确的实体数 {1: 178, 2: 772, 3: 183}

不同句子长度下的gold实体数 {1: 179, 2: 754, 3: 181}

不同类型实体的预测正确的数量 {'address': 71, 'book': 34, 'company': 83, 'game': 30, 'government': 74, 'movie': 31, 'name': 126, 'organization': 67, 'position': 112, 'scene': 48}

不同类型实体的预测的数量 {'address': 131, 'book': 59, 'company': 143, 'game': 108, 'government': 102, 'movie': 37, 'name': 182, 'organization': 117, 'position': 176, 'scene': 78}

不同类型实体gold的数量 {'address': 126, 'book': 60, 'company': 138, 'game': 92, 'government': 97, 'movie': 43, 'name': 182, 'organization': 123, 'position': 175, 'scene': 78}precision:0.597,recall:0.607,f1:0.602

在实体长度为1的条件下的三个指标值

precision, recall, f1 0.515 0.680 0.586

在实体长度为2的条件下的三个指标值

precision, recall, f1 0.674 0.621 0.647

在实体长度为3的条件下的三个指标值

precision, recall, f1 0.581 0.553 0.567

在实体长度为4的条件下的三个指标值

precision, recall, f1 0.635 0.440 0.519

不同实体长度下的统计值

pre ['0.515', '0.674', '0.581', '0.635']

recal ['0.680', '0.621', '0.553', '0.440']

f ['0.586', '0.647', '0.567', '0.519']

在句子长度为1的条件下的三个指标值:precision 0.691, recall 0.687, f1 0.689

在句子长度为2的条件下的三个指标值:precision 0.574, recall 0.588, f1 0.581

在句子长度为3的条件下的三个指标值:precision 0.601, recall 0.608, f1 0.604

不同句子长度下的统计值

pre ['0.691', '0.574', '0.601']

recal ['0.687', '0.588', '0.608']

f ['0.689', '0.581', '0.604']

{'address': 71, 'book': 34, 'company': 83, 'game': 30, 'government': 74, 'movie': 31, 'name': 126, 'organization': 67, 'position': 112, 'scene': 48} {'address': 131, 'book': 59, 'company': 143, 'game': 108, 'government': 102, 'movie': 37, 'name': 182, 'organization': 117, 'position': 176, 'scene': 78} {'address': 126, 'book': 60, 'company': 138, 'game': 92, 'government': 97, 'movie': 43, 'name': 182, 'organization': 123, 'position': 175, 'scene': 78}

不同实体类型下的统计值 dict\_keys(['address', 'book', 'company', 'game', 'government', 'movie', 'name', 'organization', 'position', 'scene'])

pre ['0.542', '0.576', '0.580', '0.278', '0.725', '0.838', '0.692', '0.573', '0.636', '0.615']

recal ['0.563', '0.567', '0.601', '0.326', '0.763', '0.721', '0.692', '0.545', '0.640', '0.615']

f ['0.553', '0.571', '0.591', '0.300', '0.744', '0.775', '0.692', '0.558', '0.638', '0.615']

## （三）Span 语料统计

total\_gold:1142.0,total\_pred:1175.0,total\_correct:920.0

[[('government', 26, 34)], [('company', 6, 8)], [('organization', 16, 19)], [('scene', 19, 21)], [('company', 34, 36)]]

[[('government', 26, 34)], [('company', 6, 8)], [('organization', 0, 11), ('organization', 16, 19), ('organization', 28, 30)], [('scene', 19, 21)], [('company', 34, 36)]]

不同实体长度下预测正确的的实体数 {1: 257, 2: 393, 3: 207, 4: 63}

不同实体长度下预测的的实体数 {1: 311, 2: 494, 3: 277, 4: 93}

不同实体长度下gold的的实体数 {1: 308, 2: 480, 3: 260, 4: 94}

不同句子长度下预测正确的实体数 {1: 162, 2: 709, 3: 49}

不同句子长度下预测正确的实体数 {1: 187, 2: 923, 3: 65}

不同句子长度下的gold实体数 {1: 192, 2: 886, 3: 64}

不同类型实体的预测正确的数量 {'address': 82, 'book': 46, 'company': 117, 'game': 79, 'government': 87, 'movie': 36, 'name': 161, 'organization': 100, 'position': 150, 'scene': 62}

不同类型实体的预测的数量 {'address': 129, 'book': 66, 'company': 146, 'game': 99, 'government': 106, 'movie': 40, 'name': 187, 'organization': 127, 'position': 180, 'scene': 95}

不同类型实体gold的数量 {'address': 127, 'book': 62, 'company': 140, 'game': 93, 'government': 101, 'movie': 47, 'name': 185, 'organization': 123, 'position': 184, 'scene': 80}

在实体长度为1的条件下的三个指标值

precision, recall, f1 0.826 0.834 0.830

在实体长度为2的条件下的三个指标值

precision, recall, f1 0.796 0.819 0.807

在实体长度为3的条件下的三个指标值

precision, recall, f1 0.747 0.796 0.771

在实体长度为4的条件下的三个指标值

precision, recall, f1 0.677 0.670 0.674

不同实体长度下的统计值

pre ['0.826', '0.796', '0.747', '0.677']

recal ['0.834', '0.819', '0.796', '0.670']

f ['0.830', '0.807', '0.771', '0.674']

在句子长度为1的条件下的三个指标值:precision 0.866, recall 0.844, f1 0.855

在句子长度为2的条件下的三个指标值:precision 0.768, recall 0.800, f1 0.784

在句子长度为3的条件下的三个指标值:precision 0.754, recall 0.766, f1 0.760

不同实体类型下的统计值 dict\_keys(['address', 'book', 'company', 'game', 'government', 'movie', 'name', 'organization', 'position', 'scene'])

pre ['0.636', '0.697', '0.801', '0.798', '0.821', '0.900', '0.861', '0.787', '0.833', '0.653']

recal ['0.646', '0.742', '0.836', '0.849', '0.861', '0.766', '0.870', '0.813', '0.815', '0.775']

f ['0.641', '0.719', '0.818', '0.823', '0.841', '0.828', '0.866', '0.800', '0.824', '0.709']

## 文件目录如下：

在models文件下，保存了模型在notebook中的运行记录，data中cluener为原始数据集，但test数据集无标注信息，因此，将valid中的后500条作为了测试数据，存储在new\_cluener中。Result中保存的模型输出结果和实验结果的分析。

