总体思路：

核心逻辑：

对比学习“低成本”的人为构造的无监督任务，形式上容易适配“低成本”的词典资源，因此可以为低资源情景下构造低成本的训练任务

深入的实验分析，深刻的说明问题

人类/历史知识的体现是模板、规则、词典

数据集：

CoNLL2003

论文：

1. Soft Gazetteers for Low-Resource Named Entity Recognition 2020 ACL有代码dynet

背景：gazetteer代表这个词在不在词典中0-1二分类。

难点：在小语种里gazetteers没有或覆盖不全。

方法：从英语(wikipedia entity linking)里来，翻译成英语，看为不同entity的置信度（已知实体链接模型），有之前工作的AE损失

实验：语言研究所做的，做的是付费的几个小语种实体识别数据集

2. Improving Low Resource Named Entity Recognition using Cross-lingual Knowledge Transfer IJCAI 2018 有pytorch代码刘挺组

背景：双语词典单独构造

①中文“本”很少为实体，对应到英文就懂了

②学习一个语言间映射MSE，out-of-lexicon也有表示

③单独学习一个词语-实体类别映射cos，也是invariant信息

3. A Robust and Domain-Adaptive Approach for Low-Resource Named EntityRecognition 2021 arxiv有pytorch代码

对低资源有个很好的总结+related work，外部资源、多任务、远程监督，论文倒没什么，领域内pretrain、bootstrap标伪标签迭代直到收敛

4. BERT memorisation and pitfalls in low-resource scenarios ruder组 感觉是ACL2021 暂未公开代码

noisy情况下不错，但few-shot和rare entity等low-resource场景不行。

学习噪声一般是出现准确率几个回合不变后开始，和ResNet完全相反。BERT出现100次以上才记得比较好，25次以下不行。

有预训练的更难忘记，忘记一般和训练数据内部特性有关系（和模型无关）

引入辅助任务做类似聚类(原型网络，我也可以用这种实现方式)，WNUT和03出现少次的表现好，03完整提升一点点。

小样本情况下泛化能力变弱了，记忆能力不变；如果有很少句子，其实前几个epoch就快速学的差不多了。

可以借鉴这篇对memorize和forget的定义，研究模型的提升来自于哪里。比如这篇文章说忘记的都是非典型的样本

很精彩的一篇文章！

5. Dual Adversarial Neural Transfer for Low-Resource Named EntityRecognition ACL 2019算是比较正统的，代码tf

对抗，分成共享路和私有路

6. A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios NAACL

优秀的low resource综述。包含noisy下的filtering method等；解决distant supervision中常见的False Negative漏了；

7. Towards Improving Neural Named Entity Recognitionwith Gazetteers 2019 ACL有pytorch代码

词典的使用方法看这篇，估计要看他代码自己写了

用词典的实体当成句子，随机在conll03摘取O的n-gram，攻击的地方：和原conll分布不一样，一个是短语，一个是句子；需要两次训练

8. Gazetteer-Enhanced Attentive Neural Networks forNamed Entity Recognition 2019 EMNLP

相比于他是端到端了，attention也是创新点。缺点和上面差不多，都是单独针对gazetter训练一个辅助分类器，缺失contextual信息

写代码 coach改 或者dynet改pytorch

实验思路：

考虑小语种？无监督（entity linking？）

borrow Danqi Chen的dropout self-contrastive

new思路，可以adapt到其他baseline或其他方式中

实验分析：是不是OOTV提升大，提高泛化能力；分析过拟合的现象，一词多义现象有没有缓解

如果有提升，多少字典数据就够了？

暂时先做通用的，再考虑适配到多语种里，另外distant supervision本身就用了knowledge base信息，结合数据增强（当然这一类一般是替换同义词或LM这种）

你需要做到极致的：分析what knowledge is transferred？

感觉low-resource和multilingual结合的很紧密呀

对高资源有效的增强方法不一定对低资源有效

结合数据增强的思路做，减少shift分布损失。self-training是hard，我们提出了soft的方法如结合mixup

弱监督的思路?

额外引入了mention的噪声，需要考虑合理的mention构造

可能难点：

词典信息的实体可能很偏

conll03 pytorch lstm\_crf 90.01

数据构造：

People：7个people文件(除politics)各随机取500个

Org：4个org文件各随机取500个

Loc：12个loc文件各随机取500个，实际只有3400个

Misc：自己和自己的named entity list比较吧

模型实现细节

LSTM共享不? trick是前几个epoch先优化template(detach 原来表示)，再计算两者都有梯度的MSE

暂时先粗浅地用实体第一个词和最后一个词的表示平均来作为实体表示，mask\_select

① 类似Coach的MSE

3500(25%)数据集 ours 89.61 vs baseline 89.63 vs ours refresh对比词典 89.73

500 数据集 ours 82.44 vs baseline 82.77 vs ours refresh对比词典 82.26

100 数据集 ours 68.40 vs baseline 67.24 vs ours refresh对比词典 69.24

50 数据集 ours 55.22 vs baseline 52.53 vs ours refresh对比词典54.75

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| method | baseline | ours | ours refresh对比词典 | 和自己比用dropout | ACL19-tagger |
| 50 数据集 | 52.02 | 55.22 | 54.75 | 54.13 | 50.08 |
| 100 数据集 | 67.24 | 68.40 | 69.51 | 69.11 | 68.45 |
| 500 数据集 | 82.77 | 82.44 | 82.26 | 83.59 | 75.50 |
| 3500(25%)数据集 | 89.63 | 89.61 | 89.73 | 89.50 | 80.80 |
| 全量数据集 | 90.10 | 90.34 | 90.21 | - | 91.14 |

② 点乘对比学习Danqi Chen

dropout 0.1

实体跟实体自己像呢？还是和同类实体像?

每个实体和他的negative拼在一起，只验证自己和自己像

还是要研究下对比学习怎么写

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| method | ours refresh对比词典 | As Pretrain | 加2层MLP  Multiply CSE | +Pretrain  50epoch | 加两层MLP MSE | +Pretrain  50 epoch  忘记改epoch没跑完 |
| 50 数据集 | 54.75 | 51.82 | 54.09 | 51.34 | 50.30 |  |
| 100 数据集 | 69.24 | - | 68.35 | 69.15 |  |  |
| 500 数据集 | 82.26 | 82.27 | 82.94 | 82.36 | 82.83 |  |
| 3500(25%)数据集 | 89.73 | - | 89.24 |  |  |  |

Pretrain 200个epoch，看下预训练的准确率最高到80%+

|  |  |  |  |
| --- | --- | --- | --- |
| method | ours refresh对比词典 | 加2层MLP  Multiply CSE | +Pretrain  200epoch |
| 50 数据集 | 52.09 | 54.09 | 52.18 |
| 100 数据集 | 69.24 | 68.35 | 67.39 |
| 500 数据集 | 82.26 | 82.94 | 82.32 |
| 3500(25%)数据集 | 89.73 | 89.24 | 89.61 |

换Dictionary为CoNLL03的数据集

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| method | baseline | ours refresh对比词典MSE | 加2层MLP  Multiply CSE | MSE，没2层MLP |
| 50 数据集 | 52.02 | 52.09 | 52.11 | 52.11 |
| 100 数据集 | 68.38 | 69.51 | 68.21 | 68.19 |
| 500 数据集 | 82.77 | 82.26 | 83.02 | 82.91 |
| 3500(25%)数据集 | 89.63 | 89.73 | 89.68 | 89.62 |

multiply cse换成乘法和减法，字典还是CoNLL03

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| method | baseline | ours refresh对比词典MSE | multiplyCSE  乘法+减法 | 先pretrain200个epoch | pretrain 200个epoch后就不要这个loss |
| 50 数据集 | 52.02 | 52.09 | 52.06 | 52.68 | 52.65 |
| 100 数据集 | 68.38 | 69.51 | 68.10 | 68.84 | 68.80 |
| 500 数据集 | 82.77 | 82.26 | 83.30 | 82.83 | 82.91 |
| 3500(25%)数据集 | 89.63 | 89.73 | 89.62 | 89.55 | 89.74 |

都用我的有MLM的BERT Embedding？比较的时候别人的工作也加上BERT embedding

引对比学习任务导致原embedding遗忘的问题，BERT。

替换破坏分布，用self-training等

TREC

5组

|  |  |
| --- | --- |
| method | Acc |
| BERT baseline | 52.28 |
| +GPT data | 61.24 |
| +CBERT data | 68.6 |

先试下直接生成

-> 设计监督训练

CoNLL 2003 100个句子->生成100个句子

实体标签根据前后重合的单词来判断

|  |  |
| --- | --- |
| method | F1 |
| BERT baseline | 69.34 |
| +GPT data | 69.01 |

其他考虑的trick:

1.teacher forcing

代码整理：

baseline代码：Transformer Augmentation包括BERT GPT BART

DAGA: pytorch LSTM语言模型 放在BERT没什么效果

BOND: self-training的训练方法和reweight方法

ACL19-tagger: 为词典训练一个伪标签分类器

LADA: 句子级别插值+半监督的consistency loss

low-resource数据集设定

LADA: CoNLL的5%, 10%和30%

DAGA: CoNLL 1k 2k .. 8k

NEEDLE(small sup large semi): bio all

A Robust and Domain-Adaptive Approach for Low-Resource Named Entity Recognition: bio 10% 30% .. all & dataset disease

也有个别用50/100/500

我的设置：500 1k 2k

结论：看自己文章跟哪些具体数据增强方法的文章比较接近