

****

本科毕业设计（论文）

中期检查报告

**基于深度学习的专利与标准的关联关系发现技术**

|  |  |
| --- | --- |
| 姓名： | **霍飞烨** |
| 学号： | **18231098** |
| 指导老师： | **张辉** |
| 院系 | **高等理工学院** |

2022年 4 月 8 日

目 录

[1. 课题简介 1](#_Toc98168773)

[1.1 课题背景 1](#_Toc98168774)

[1.2 研究目标和内容 2](#_Toc98168775)

[2. 论文工作进展 3](#_Toc98168776)

[2.1 总体研究进展 3](#_Toc98168777)

[2.2 总体研究方案 4](#_Toc98168778)

[2.3 主要研究成果 4](#_Toc98168779)

[2.3.1 数据集选取与构建 4](#_Toc98168780)

2.3.2 算法的选取与实现..............................................................................5

2.3.3 模型的迁移与优化..............................................................................7

[3. 尚存的问题及后期工作安排 1](#_Toc98168781)0

[3.1 尚存的问题及措施 1](#_Toc98168782)0

[3.2 后期工作安排 1](#_Toc98168783)0

[4. 参考文献 1](#_Toc98168784)1

# 课题介绍

## 1.1 课题背景

前一段时间，世界知识产权组织（WIPO）公布了 2021 年PCT国际专利申请排名榜单，申请量排名前 5 位的国家分别是中国、美国、日本、韩国和德国，其中中国企业专利数量共计 69540 件，同比增长 0.9%，连续第三年位居申请量排行榜第一位。爆炸式增长的专利背后，是繁复严谨的技术标准的支持与约束。但是随着新的技术领域的不断开拓，旧有的技术标准面临着更新与完善的问题。这就需要笔者们分析与相关领域的专利相关联的技术标准。

在此背景下，山西省“十四五”规划和山西省委《关于实施“111”创新工程支撑引领高质量转型发展的意见》（晋办发[2020]5号）、山西省政府《实施“111”创新工程支撑引领高质量转型发展工作方案》（晋政办发[2020]27号）文件精神，聚焦“111”创新工程的标准化需求，全面调研国内外标准化及专利现状，研究形成科技创新所需的标准化及专利数据集成方案，集成与产业密切相关的标准化及专利数据信息，产出1-2个优势重点技术领域的标准、专利内容揭示数据集，提出“111”创新工程的标准及专利创制服务体系建设方案，通过构建知识重组能力来加快技术标准及专利形成，以标准数据及专利数据形成的知识库服务促进山西省科技成果产业化，以标准化服务助力山西省打造一流创新生态，为实现山西省经济社会高质量转型发展提供标准化支撑。

传统的标准化分析方法，需要相关的专家人工分析与新增专利的相关联的技术标准，从而再根据提炼的结果来制定相关的决策。在过去专利数量增速未曾如此迅猛的情况下，这种方法未尝不是一种解决问题的有效方式。但是随着信息时代的到来，与专利相关的信息也正式步入大数据时代，此时仍然依赖人工分析数据并制定决策便显得低效且不可避免地导致决策准确率的下降。因此，笔者们亟需一个高效且准确的方法来对日新月异的技术专利与现有的技术标准进行数据挖掘。

## 1.2 研究目标和内容

### 1.2.1 研究目标

本毕设，旨在开发一个用于匹配技术专利以及相关技术标准的算法。该算法需要基于深度学习的相关知识来找到合适的模型及其参数，同时这个模型还必须在实际的应用中——即与专利、标准相关的数据集中，取得预期中的效果。

### 1.2.2 研究内容

A 改进已有的NER算法，并通过第三方的数据集来训练合适的NER模型

在目前流行的NER算法中，选择综合了轻量、高准确度两种优点的合适的算法，并且在此基础上进行改进。同时由于项目提供的专利标准数据集并没有分词的label属性，因此无法在该数据集上训练模型，而需要通过第三方的数据集来负责训练模型，而对于模型迁移来说，不同的训练集也会影响到迁移的效果，因此如何选择合适的训练集也是一项亟待解决的问题。

B 将训练好的NER模型迁移至专利、标准数据集中

在第三方的数据集上训练好所需的模型后，就需要将之迁移到本毕设真正需要处理的数据集——也即是专利、标准数据集上，笔者们需要对这两个数据集进行实体抽取操作，并将抽取出的实体进行匹配操作。而对于迁移学习，笔者们很难用网络上开源数据集训练出来的模型原封不动地用于预测目标数据集，因此，笔者们需要手动构建自己的数据集，并用该数据集来训练已经优化好的模型算法，得到最终的深度学习模型。最后再将这个模型应用到专利-标准数据集，即可得到最后的想要的结果。

C 将毕设成果整合进项目工程中

由于本毕设实质上是山西111创新工程平台中的一个子功能，因此，在完成了本毕设后，还需要将成果整合进整个平台系统中，这就需要和负责相关模块的开发人员进行对接，并且DEBUG使得整体工程鲁棒可用。

# 论文工作进展

## 2.1 总体研究进展

NER算法的研究方面，已经完成了对合适的NER数据集的选取与构建，同时在研究现有的流行NER算法的基础上，对基于深度学习的NER算法进行了优化改进，并且使用构建的NER数据集训练出了在测试集上表现相对优秀的模型。

NER模型迁移方面，由于网络上开源的中文NER数据集与本次毕设的目标数据集存在着较大的差异，因此，如果只是原封不动地将在开源数据集上训练出的模型迁移至目标数据集上，效果不会非常理想。因此，笔者选择以文本风格与专利-标准信息相似，且自带关键词的知网摘要-关键词数据集对模型进行训练。该模型直接迁移至专利-标准数据集上的表现也非常好。

因此目前第三部分的任务只剩下如何将专利与标准分别抽取出来的实体进行关系匹配，而对于最后的整合与DEBUG阶段，将会在所有的工作完成后进行。

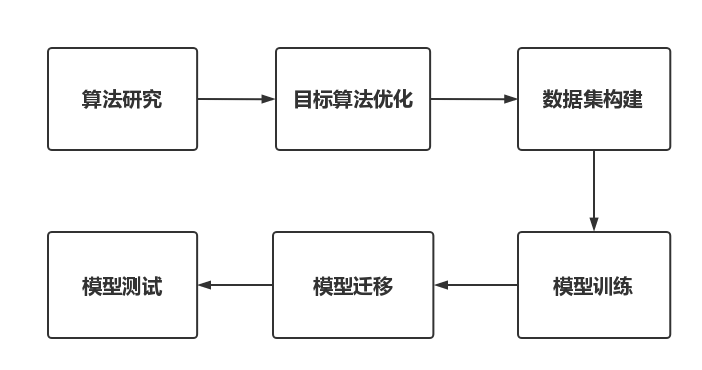
**表2.1 计划进度表**

|  |  |  |
| --- | --- | --- |
| 周 | 工作任务 | 完成情况说明 |
| 1-4 | 学习毕设所需的知识 | 完成 |
| 5-8 | 找到合适的NER算法并且在其基础上优化改进，并且在第三方数据集上训练得到相对优秀的准确率。 | 完成 |
| 9-11 | 将在开源模型上得到验证的算法运用至专利-标注数据集中。最后，则会对抽取出专利实体数据集与标准实体数据集间进行关键词匹配。 | 已经大体上完成模型迁移部分，尚未开始实体匹配工作。 |
| 12 | 将最终的成品模型整合进整体项目中。 | 尚未开始 |

## 

### 2.2 总体研究方案

本毕设的研究路线主要分为现有算法研究、目标算法优化、数据集构建、模型训练、模型迁移、模型测试六个阶段。流程图如图2.2。



**图2.2 研究路线**

## 2.3 主要研究成果

目前阶段，本毕设的研究成果主要有3个方面：NER中文数据集选取与构建、NER算法选取与优化、目标算法对应的模型训练与测试。

### 2.3.1数据集选取与构建

在中文BERT研究领域，数据一直是困扰研究者的主要问题之一，数据量与颗粒度是影响BERT模型训练效果的两大主要因素。本次毕设选取的是由CLUE(中文语言理解测评基准)项目组在2020年基于清华大学的开源文本分类数据集THUCTC构建的细粒度命名实体识别数据集CLUEFineGrainNER，源数据来自Sina News Rss。

该数据集中包含训练集与验证集两部分，其中训练集10748例，验证集1343例，数据分为10个标签类别，分别为: 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）。其中各种标签类别的具体分布如表2.3。

**表2.3 数据集标签分布表**

|  |  |  |
| --- | --- | --- |
| Label | TrainSet | TestSet |
| Address | 2829 | 364 |
| Book | 1131 | 152 |
| Company | 2897 | 366 |
| Game | 2325 | 287 |
| Government | 1797 | 244 |
| Movie | 1109 | 150 |
| Name | 3661 | 451 |
| Organization | 3075 | 344 |
| Position | 3052 | 425 |
| Scene | 1462 | 199 |
| Total | 10748 | 1343 |

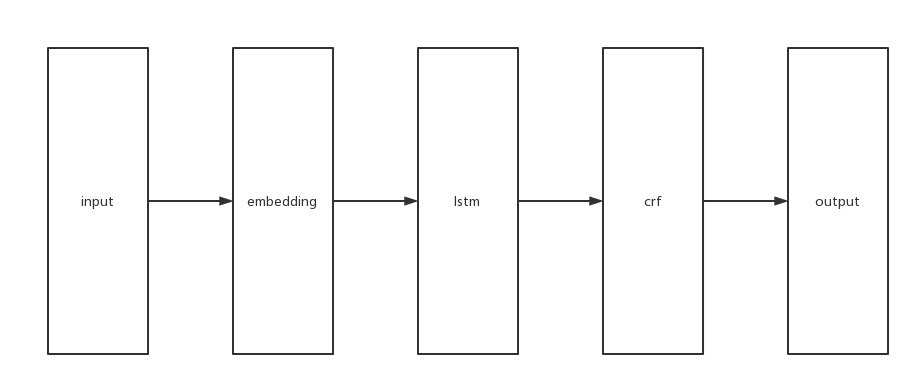
本数据集以json的格式储存，对于每条具体的数据，数据分为两个部分：text和label，其中text代表文本信息，label代表实体标签，例如：

{text: "北京勘察设计协会副会长兼秘书长周荫如",

label: {"organization": {"北京勘察设计协会": [[0, 7]]}, "name": {"周荫如": [[15, 17]]}, "position": {"副会长": [[8, 10]], "秘书长": [[12, 14]]}}}

### 2.3.2 算法的选取与实现

对现有的算法的研究中，目前的流行的用于实体抽取的算法主要分为基于LSTM的各种衍生算法以及近年来的出现的基于Transformer的BERT算法。在目标算法选取上，由于本毕设的最终成果要求并非对专利-标准数据集的精确实体抽取，而是基于两个数据集各自抽取出来的实体集间进行匹配，因此个位数维度的准确性的差异并不会显著影响最终匹配结果，而BERT算法在模型训练过程中，对于算力的要求比较苛刻，在本科毕设阶段日常所能接触的算力中，都很难满足其要求。经过实测，在I7-8750H与GTX1060的环境下，同样一批数据训练同样的epoch，在改进后的LSTM算法上进行模型训练仅需45分钟，而在BERT上训练时间多达10小时。而本毕设并不是单纯的对于算法性能的学术性研究，而是需要应用于工业用途，因此在保证一定的性能的前提下，还需要尽量选取轻量级的模型算法，所以在最后笔者选择了基于LSTM的改进模型LSTM-CRF作为baseline。该算法的模型结构如图2.4。



**图2.4 lstm-crf模型图**

本毕设研究的基于深度学习的专利与标准关系发现技术所用的算法主要基于Pytorch深度学习框架实现，训练过程中，超参数来源于选取的baseline，而初始值则为Pytorch中随机种子函数生成。

作为在CLUEFineGrainNER数据集上模型性能的对照试验，笔者分别选取了由Woodbridge等人实现的不包含CRF的Lstm模型以及最新的由CLUE项目组复现并提供的BERT-base模型作为对照组，以F1与训练时长作为实验评价指标，同时排除了Other标签实体对于结果的干扰，可以得到效果对比如表2.5。

**表2.5 效果对比表**

|  |  |
| --- | --- |
| Model | F1(%) |
| LSTM | 63.62 |
| BERT-base | 78.32 |
| LSTM-CRF | 74.57 |

由该表，笔者们可以看到比起简单的LSTM，LSTM-CRF复合模型的效果要显著优越，同时二者的训练代价几近相同，而对比最近流行的BERT算法，由于具体应用场景中，文本的长度普遍较短并且训练集的数据量很大，二者的性能差异不大，这些差异是在可接受范围内的，同时考虑到毕设可以接触到的算力的限制以及工业实现对于轻量级模型的需求，对于算力要求以及网络复杂度大大低于BERT模型的LSTM-CRF显然是更加适合本次毕设的模型。而对于BERT与LSTM-CRF模型，具体到每个标签（不包含Other）的测试实验效果如表2.6。其中P代表精确率(precision)，R代表召回率(recall)，L代表LSTM-CRF网络模型，B代表BERT网络模型。

**表2.6 实验具体效果对比表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Label | P(%) | | R(%) | | F1(%) | |
| L | B | L | B | L | B |
| Scene | 69.73 | 65.77 | 56.11 | 69.86 | 62.18 | 67.75 |
| Position | 80.26 | 74.46 | 71.38 | 79.45 | 75.56 | 76.87 |
| Government | 77.96 | 75.36 | 79.33 | 85.43 | 78.64 | 80.08 |
| Organization | 74.56 | 71.15 | 67.90 | 79.29 | 71.07 | 75.00 |
| Name | 75.08 | 84.35 | 77.60 | 89.25 | 76.32 | 86.73 |
| Address | 69.13 | 54.63 | 64.18 | 63.27 | 66.56 | 58.63 |
| Game | 79.91 | 78.27 | 86.85 | 89.15 | 83.23 | 83.36 |
| Book | 84.05 | 77.02 | 63.85 | 80.52 | 72.57 | 78.73 |
| Company | 77.16 | 72.58 | 75.79 | 81.22 | 76.47 | 76.65 |
| Movie | 78.57 | 82.19 | 77.29 | 79.47 | 77.93 | 80.81 |

### 2.3.3 模型的迁移与优化

由于CLUEFineGrainNER数据集来源自日常新闻，其文本风格与专利信息这种偏科研学术的文本比有着较大的区别；并且比起只对专利的信息提取其中的关键词一种命名实体，对于CLUEFineGrainNER的任务需要抽取多达10种类别的命名实体，因此如果直接将第一部分工作中训练出来的模型迁移至专利-标准数据集上，效果非常差。而本次项目所给的数据集又缺少NER模型训练所需要的标签体系，无法直接对专利-标准数据集进行模型训练。

针对这个问题，笔者发现知网中论文预览界面的摘要部分与专利-标准文本的风格非常相似，而且知网的每一篇论文都有其自带的关键词，这些关键词可以作为实体抽取的标签进行训练。因此，在模型迁移部分，可以用知网的摘要-关键词数据集作为训练集训练模型，然后将训练集迁移至专利-标准数据集上。

而为了获取可以被LSTM-CRF算法程序识别的知网摘要-关键词数据集，笔者爬取了与最后需要处理的专利-标准数据集相同领域的相关论文信息，其数据量分布如表2.7。

**表2.7 知网爬取数据分布**

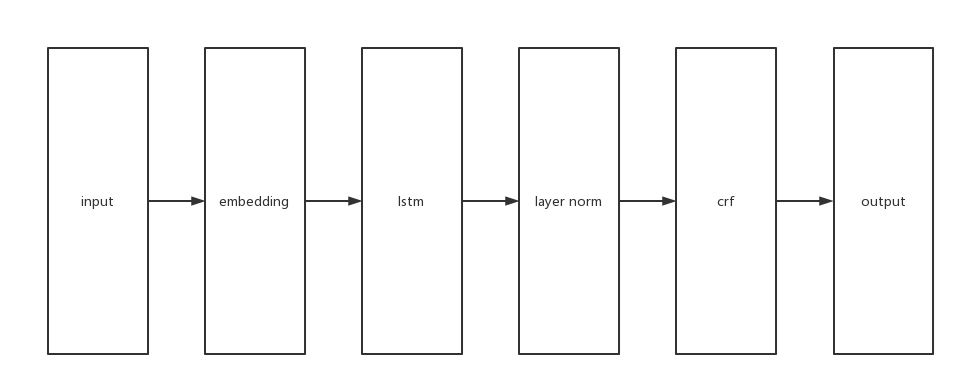
|  |  |
| --- | --- |
| 领域 | 数据量 |
| 半导体 | 1230 |
| 大数据融合创新 | 73 |
| 非常规天然气 | 1205 |
| 轨道交通装备制造 | 205 |
| 节能环保 | 1188 |
| 煤机智能制造 | 10 |
| 碳基新材料 | 17 |
| 特种金属材料 | 528 |
| 通用航空 | 1229 |
| 现代医药和大健康 | 10 |
| 新能源 | 1251 |
| 新能源汽车 | 1233 |
| 信息技术应用创新 | 1225 |
| 有机旱作农业 | 250 |

而在爬取到原始数据后，我还需要将原始数据转换成可以被识别的JSON格式，其大致结构形如：

{"text": "实施机械化旱作农业开发实现农业高产优质高效——高密县周阳乡旱作小麦开发纪实",

"label": {"keyword": {"高产优质高效": [[15, 20]], "旱作小麦": [[29, 32]], "机械化旱作农业": [[2, 8]]}}}

在算法优化环节，由于知网摘要-关键词信息知识密度低，粒度粗，在训练时train\_loss以及acc迭代缓慢。与此同时，笔者注意到该模型为非树形结构RNN模型，因此可以在LSTM层与CRF层之间添加归一化LayerNorm层来加速收敛速度，同时可以丢弃无用信息较多的摘要主体部分，选取知识密度高的标题部分与摘要中含有较多关键词的关键句作为训练集来进行训练，优化后的模型结构为图2.8。



**图2.8 优化后的lstm-crf模型图**

最后，该模型在知网摘要-关键词测试集上的表现如表2.9。

**表2.9 知网数据测试集表现**

|  |  |  |  |
| --- | --- | --- | --- |
| Label | P(%) | R(%) | F1(%) |
| Other | 73.84 | 83.39 | 78.33 |
| Keyword | 66.54 | 64.11 | 65.88 |

可以看到，比起CLUEFineGrainNER数据集而言，对于有用实体的抽取的准确度大致上下降了10%，不过仍然属于期望中的结果。而将该模型迁移至专利-标准数据集，由于没有大量的已标注的专利-标准数据作为测试集，只能对有限的小样本进行抽样测试，经过人工校验，其准确度大致在60-70%间，符合预期的期望。

# 尚存的问题及后期工作安排

## 3.1 尚存的问题及措施

### 3.1.1 存在的问题

1) 在知网摘要-关键词数据集上，LSTM-CRF的表现比起CLUEFineGrainNER数据集训练得到的模型性能上仍然存在差异。

2) 假设已经分别抽取出来专利信息的命名实体以及标准信息的命名实体，如何在两个数据集间挖掘实体的相关性。

### 3.1.2 解决的方法

1) 通过对不同领域的论文摘要分别训练，或者对数据进行进一步的蒸馏，可以提升数据问题带来的性能下降。或者可以尝试其他的模型。

2) 由于标准数据集属于一个领域的不同item抽取出的命名实体间可能存在很强的耦合性，因此为了防止一个专利关联到冗余的标准，不能对两个实体数据集进行复杂的文本相似度算法的计算，恰恰相反，只需要对两个数据集进行简单的match操作，如何对于多个命名实体相匹配的专利-标准对进行定量加权排序输出。同时为了确保匹配结果的准确性，还需要对标准数据集进行一定的清洗。

## 3.2 后期工作安排

本毕设的后期工作安排如表3.1。

**表3.1 后期任务计划表**

|  |  |  |
| --- | --- | --- |
| 周 | 工作任务 | 完成计划 |
| 9-10 | 继续优化模型算法在知专利-标准数据集上的表现 | 通过对不同领域的论文摘要分别训练，或者对数据进行进一步的蒸馏，可以提升数据问题带来的性能下降 |
| 11 | 清洗并匹配专利数据集与标准数据集分别提取出的命名实体。 | 运用python对两个数据集进行match以及设计加权算法来得到正确的结果 |
| 12 | 将毕设成品整合进系统。 | 通过和相关开发人员合作完成。 |

# 参考文献

1. Lample G , Ballesteros M , Subramanian S , et al. Neural Architectures for Named Entity Recognition[J]. 2016.
2. Collobert R , Weston J , Bottou L , et al. Natural Language Processing (almost) from Scratch[J]. Journal of Machine Learning Research, 2011, 12(1):2493-2537.
3. Ando R K , Zhang T . A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data[J]. Journal of Machine Learning Research, 2005, 6:1817-1853.
4. Xu L , Tong Y , Dong Q , et al. CLUENER2020: Fine-grained Named Entity Recognition Dataset and Benchmark for Chinese[J]. 2020.
5. Beryozkin G , Drori Y , Gilon O , et al. A Joint Named-Entity Recognizer for Heterogeneous Tag-sets Using a Tag Hierarchy[J]. 2019.
6. Devlin J , Chang M W , Lee K , et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.
7. Wang, Xuan, Xu, et al. Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN[J]. Expert Systems with Application, 2017.
8. Greff K , Srivastava R K , J Koutník, et al. LSTM: A Search Space Odyssey[J]. 2015.
9. 韩雪冬. 基于CRFs的中文分词算法研究与实现[D]. 北京邮电大学, 2010.
10. 洪铭材, 张阔, 李涓子. 基于条件随机场(CRFs)的中文词性标注方法[J]. 计算机科学, 2006, 33(10):148-151.