**总结**

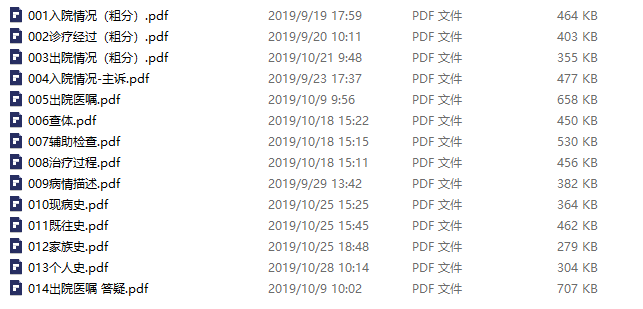
**概要：**

电子病历（住院病历）包含入院记录、首次病程记录、病程记录、出院记录、手术记录、护理记录、病案首页等多个部分。其中既有完全机构化的数据（护理记录，病案首页…），又有大量非结构化的数据（入院记录，病程记录，出院记录…）。对于住院病而言，非结构化部分包含大量的有用信息，这些信息对于后续AI辅助诊断，编码推荐（诊断编码，手术编码），质控，支付保障等一系列系统的研发起着至关重要的作用。因此电子病历非结构化数据的结构化是后续研发的基础。

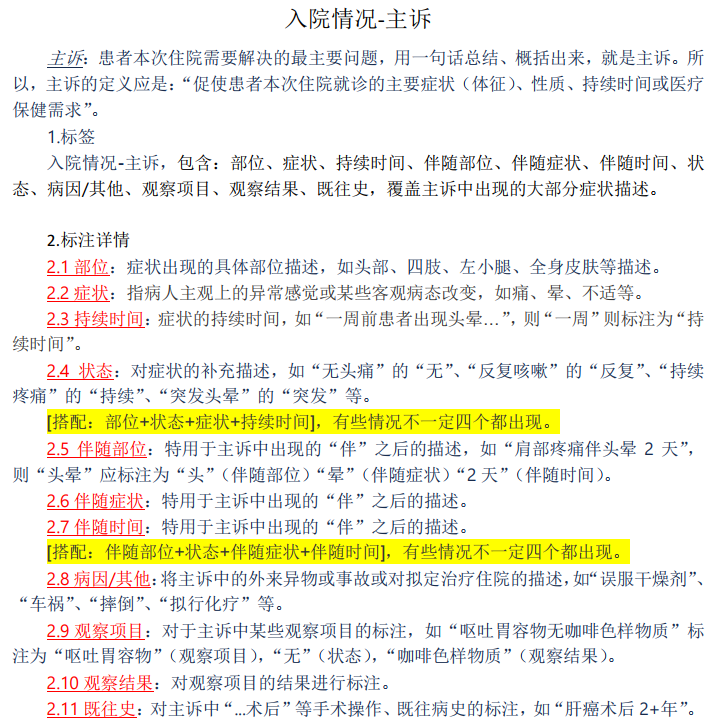
**一、数据标注**

电子病历结构化主要借助的方法有NER，RE，正则等，其中最主要的方法就是NER和正则。对于NER而言目前主流的模型有，Bi-LSTM+CRF，LatticleLSTM，BertNER（基于bert的NER方法）。然而对于任何机器学习（或深度学习）模型而言，数据标注毫无疑问都是首先需要解决的问题。我们采用的是基于字的BIO的标注规范。

数据标注应该明确两点：一、标注任务应该跟着需求走，而不是理论性的；二、标注规则应该尽可能的简洁。因此根据需求，我们分别针对电子病历入院记录以及病程记录中的部分信息进行标注。如下图，我们分别针对主诉、现病史、既往史、个人史…等进行标注，对于每一个标注任务，我们分别设计对应的标注规则。

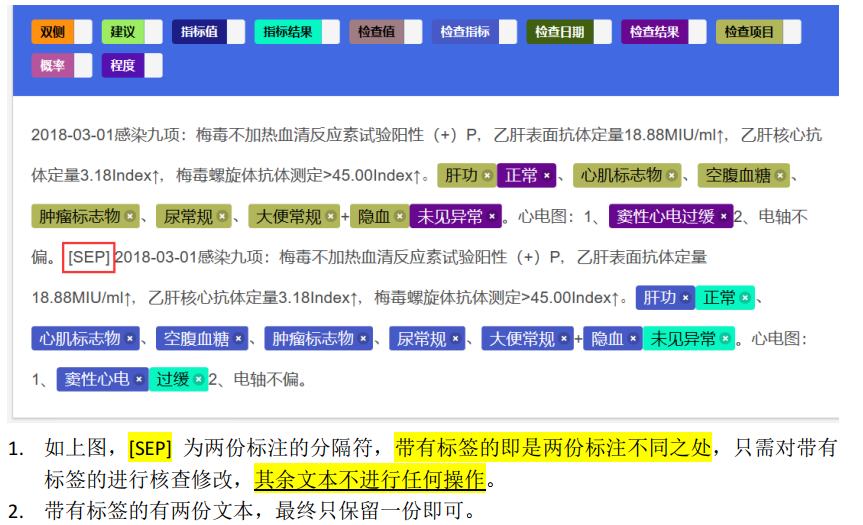


对于每一个标注任务，我们定义的规则应该尽可能的简洁。



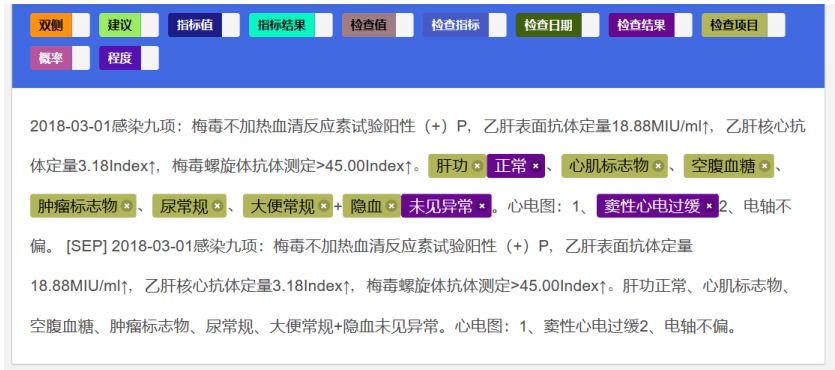
**1、数据标注实施步骤：**

对于每一份待标注数据，均分配给两个标注人员独立完成，待两人标注结束后将两人的标注结果分配给带三人进行审核。审核主要目的是核对两人标注不同的地方，对于两人标注结果一致的部分不做任何修改。如下图



整合后的每一条核对数据由之前两人的标注结果组成，中间用分隔符SEP分隔，SEP前面部分代表第一个人的标注结果，SEP后面部分代表第二人标注结果。界面只显示两人标注结果不同的词语，对于标注相同的部分没有任何显示。审核人员只对颜色显示的部分进行审核（不得修改其余文本），并且最终只保留一份带标签的数据即可。

例如上图中SEP前部分的标注时正确标注，而SEP后部分的标注的为错误标注，因此叉掉第二个人的标注结果，只保留第一个人的标注结果。



如果前后均有对的标注和错的标注，则选择保留某一个，并将所有的正确标注结果均标注在保留的那一个上。

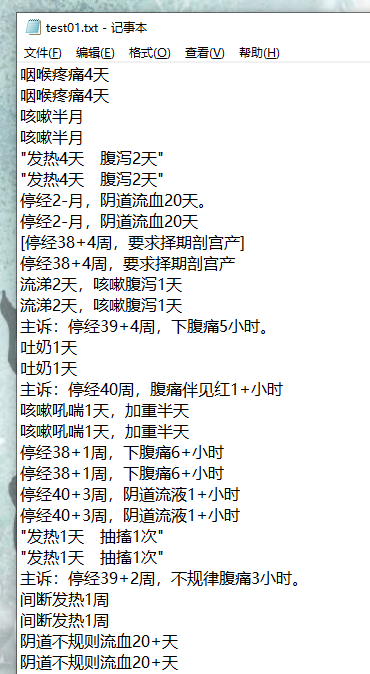
**2、数据标注的预处理和后处理：**

**1）、原始标注数据预处理（空白未标注数据）**

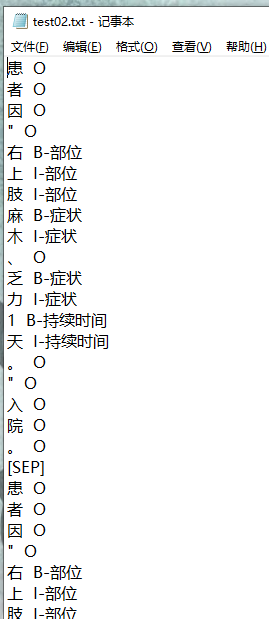
导入数据为原始病案数据（例如原始病案的主诉），每条数据一行，txt格式。

**2）、原始标注数据后处理**

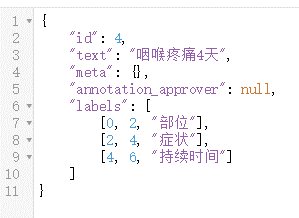
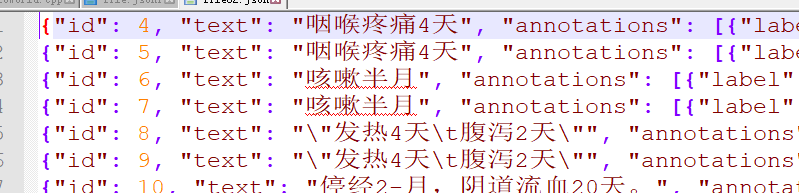
将原始未标注过（未经模型预测）的数据导入标注工具，标注后导出数据（json格式），由于模型输入的训练数据格式是BIO标注的数据格式形式，因此必须将导出的json格式数据转换成模型的训练数据格式，如下图



原始未标注数据



模型训练数据的格式



标注工具导出数据格式

**3）、模型预测数据标注前处理**

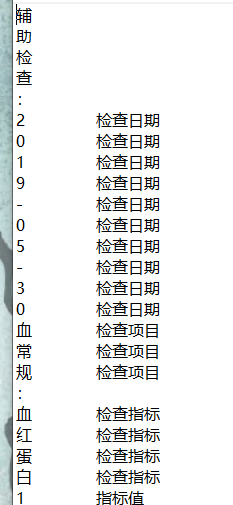
模型预测输出数据如上图右边所示，而标注工具对导入数据格式要求只有三种，第一种就是上图左边所示，第二种json格式如上图中间所示，第三种类似上图右边所示（但是略有不同需要做修改）。

由于模型输出格式与工具导入数据要求格式的第三种格式最为接近，因此我们采用第三种格式。将模型预测输出数据进行处理，再导入标注工具。字与label标签之间的空格替换为\t（制表符）或者替换为两个空格，去掉标签中的B-和I-，只保留实体类别。如下图所示



模型预测输出格式

工具导入要求的第三种格式



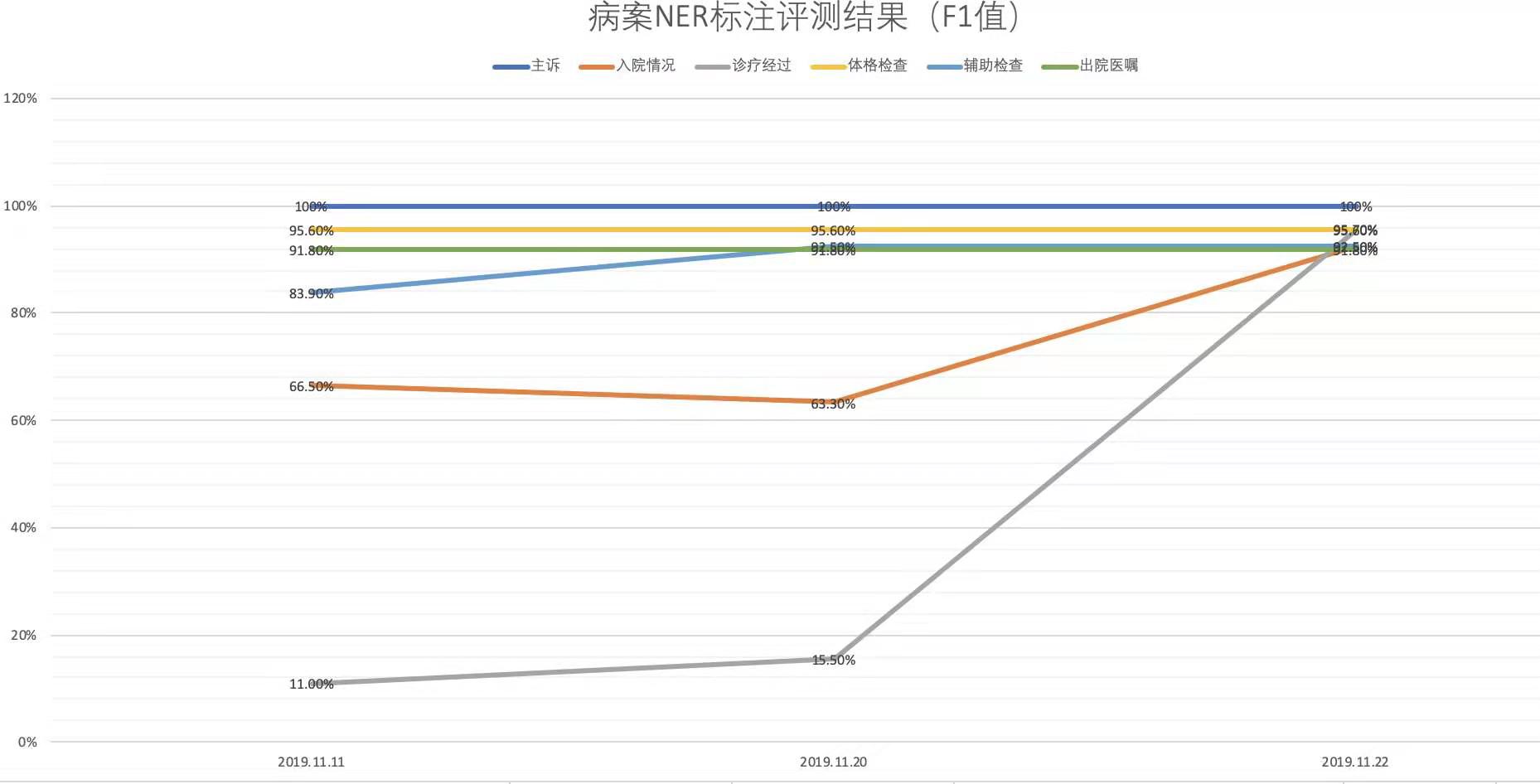
**4）、待审核数据导入前预处理**

两人独立标注的数据导出后的json格式如上图所示，要进行审核则需将两人的标注数据进行拼接，拼接后用[SEP]分隔，如前图所示。

**二、NER模型训练（数据迭代训练）**

NER任务中目前较为流行的模型主要有Bi-LSTM+CRF，LatticleLSTM，BERT以及基于BERT改进的一些模型（XLnet等）。我们通过对比实验我们最终采用BERT-BiLSTM+CRF作为最终的模型，分别在每一类任务上进行训练测试。

首先我们在每一类任务上分别标注200~250份原始病历（两人分开独立标注，第三人（医学专员）审核），然后利用该标注数据训练一个初始模型（训练时间1-2天）。待初始模型训练结束后，进行测试并记录测试结果。然后输入2k~3k份新的数据进行预测，将预测数据导入标注工具，同时分配给两个标注人员进行标注（对模型预测结果纠错），待标注结束后分配给第三人进行审核（具体审核规则可以参见核对说明）。最终将审核后的数据加入到训练数据对模型进行迭代训练，测试。每次迭代训练结束后都保存训练好的模型，并基于新的模型预测一批（4k-5k份）新的数据，然后分别标注（纠错）、审核……。



针对新数据的预测，我们采用的是写一个Server接口，接口的输入是待预测的新的病历文本（txt格式），输出为模型的预测结果（txt格式），输出结果可以直接导入到标注工具进行标注。 每次要对新的数据进行预测时我们都利用脚本开启Server接口，当然这里也可以基于flask来写一个简单的预测功能界面。

**三、结构化工程实现（利用java写的API接口）**

由于模型的预测输出结果格式是“字+标签”的形式，因此需要对其进行组合拼接处理，使其还原成原始的完整词的形式。结构化工程的目的是使使得我们可以按照需求输入病历文本时，可以得到对应的结构化的NER实体词，并且这些实体词按照json格式进行返回。

<https://www.sohu.com/a/227772374_100111898>

注册服务

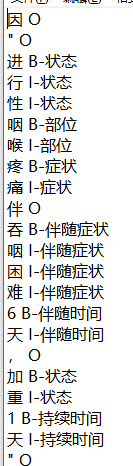
Server

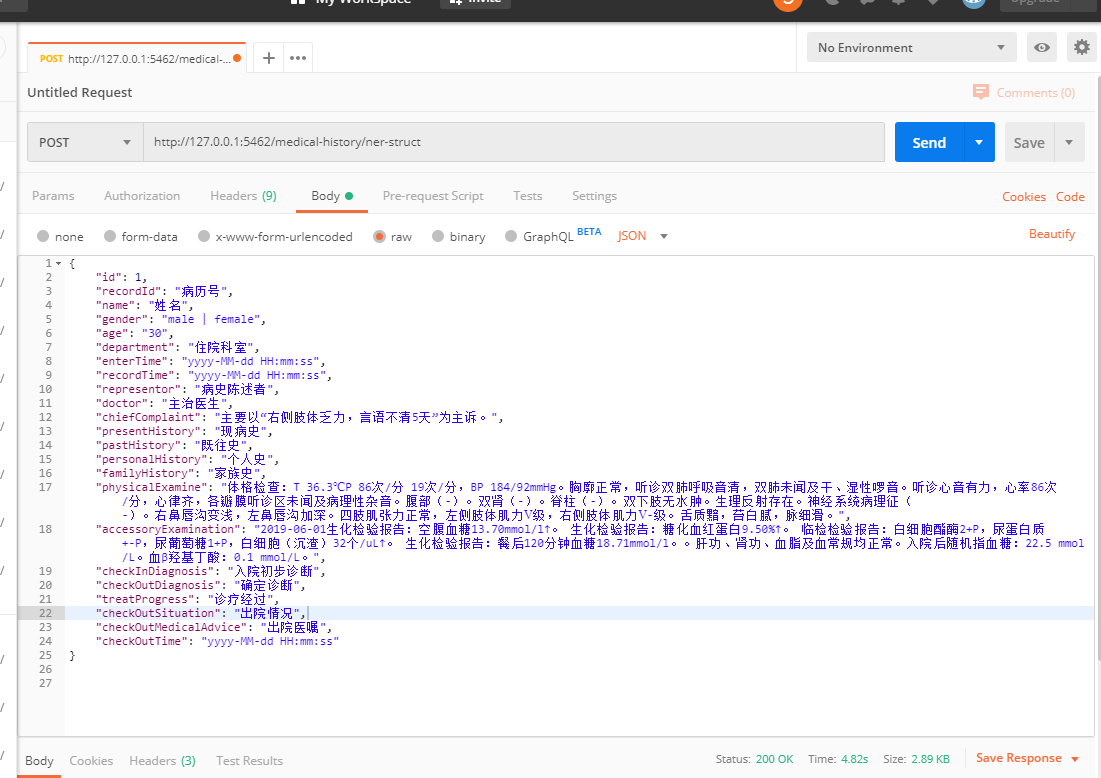
Maven工程

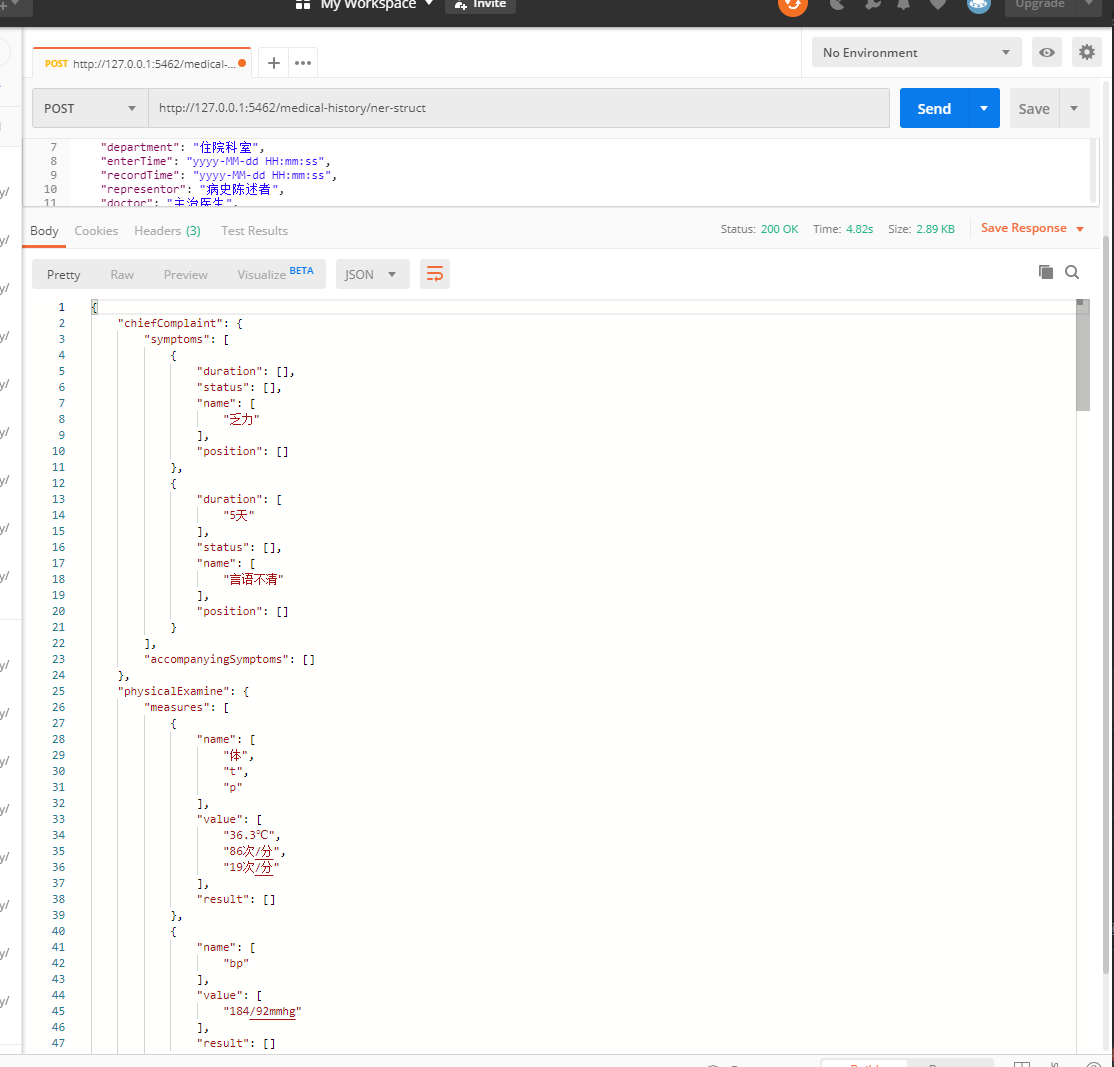
病案数据（病案号）

NER接口

解析后的字段，json格式

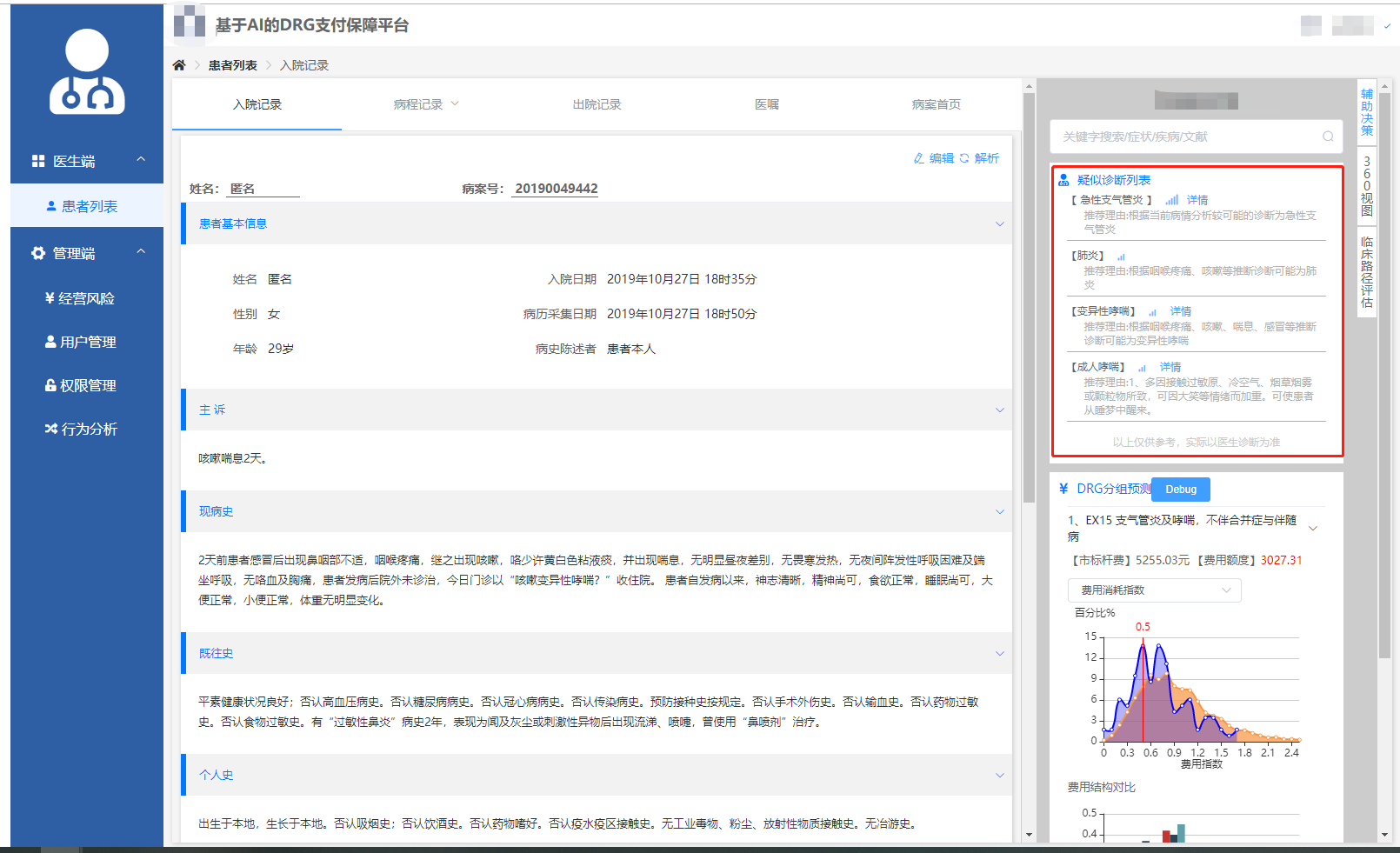


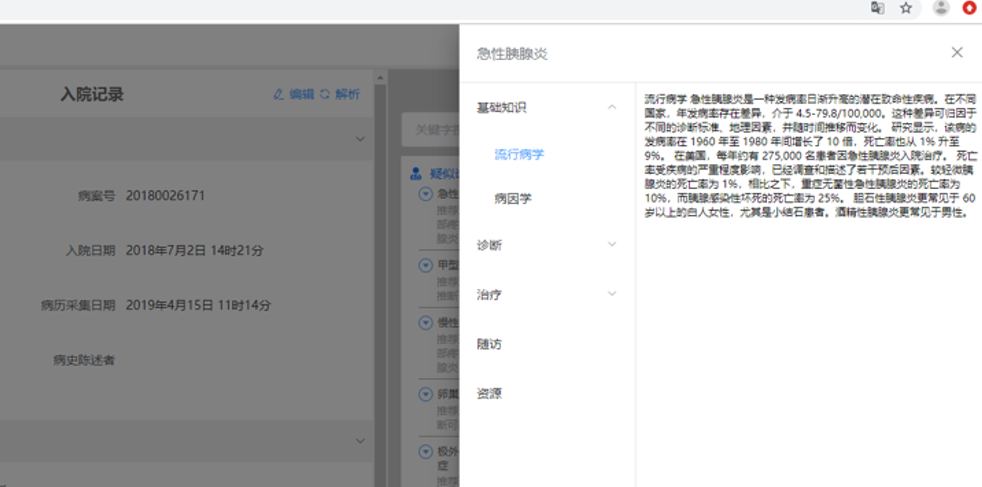




**四、疑似诊断列表推荐**

所谓的疑似诊断列表推荐，就是根据患者的入院记录数据对患者进行诊断推荐，这里的推荐结果将为医生确定“初步诊断”提供参考。该模型同样采用BERT来构建多分类模型，训练数据就是入院记录数据（包括主诉、现病史、既往史、体格检查、辅助检查等），训练数据的label由“入院记录-入院初步诊断”或者“病案首页-主要诊断+其他诊断”来构成。





说明：由于数据有限，而疾病类别（ICD-10）多达2w+，因此在构建数据集时，对数据进行了统计，凡是出现次数过少（低于20次）的疾病所对应的病历将被剔除掉。因此最终获得的标签总共800+，换句话说所有的病案数据中疾病出现次数大于20次的疾病只有大约800类。

推荐理由根据推荐的疾病查找对应的知识库（利用临床指南构建的知识库），将查找到的因素作为推荐理由，如果知识库中没有该疾病对应的知识内容（没有该疾病对应的指南数据）则统一显示“根据病情描述…推荐为…”。通过详情可以查看该疾病对应的病因学，症状等指南信息。

在此需要构建疾病知识库，该知识库从卫生部，中华医学会等官方机构获取各种疾病对应的临床指南，然后将其结构化到相应的数据库中。

**五、疾病诊断编码预测**

疾病诊断编码主要是出现在病案首页中，由于卫计委和医保局要求各医院定期，将质控后的病历数据上传到卫计委的病历数据中心。因此各医院病历科会对本院的病历进行质控，主要会看病案首页是否完善（诊断编码是否有遗漏，错误，费用信息是否完善等）。

诊断编码预测原理与疑似诊断列表推荐的原理类似，同样采用BERT来构建文本分类模型，其中针对主要诊断编码推荐构建单分类模型（一份病历一个主要诊断），针对次要诊断编码推荐构建多分类模型（一份病历对应多个次要诊断）。训练数据来源于病历的各个部分，但是由于BERT对于输入序列长度敏感（最大长度不超过512），因此在输入数据的截取上需要进行优先级划分（出院记录>病程记录>入院记录）



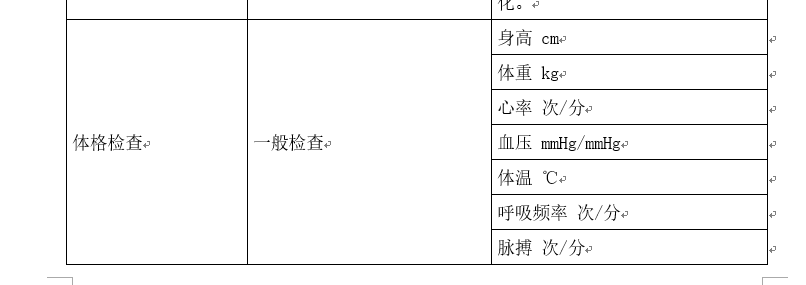
**六、手术编码预测**

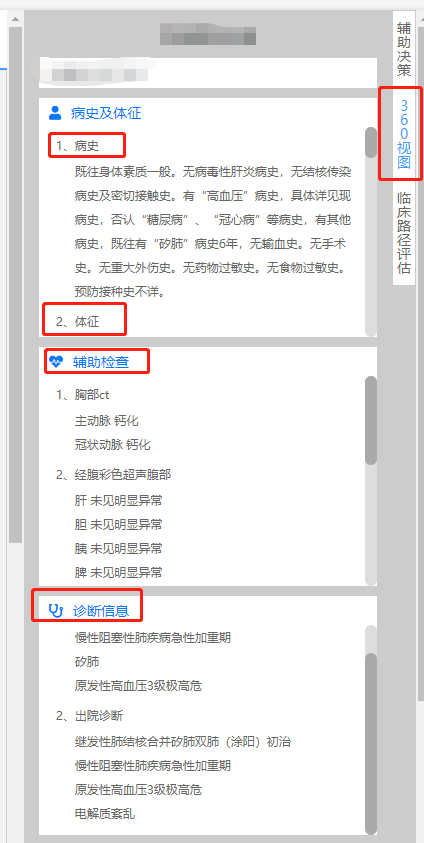
手术编码预测原理跟疾病诊断编码预测原理类似，在模型选择上同样采用BERT来构建分类模型。只是数据集的构建上相对而言更为麻烦，因为手术编码预测必须根据手术信息来进行预测，而病历中除了“手术记录”会记录手术信息以外，其他的很多地方可能会记录手术信息。必须结合实际病案数据情况进行分析后再确定构建数据集的方法。

在我们拿到的数据中有将近60%的病历的病案首页有手术编码，但是病历中却没有手术记录，通过查看case发现很多妇科手术或者清创手术、微创手术等是没有手术记录的，但是它们通常会出现在“诊疗经过”或“病程记录”中。因此前期需要把所有可能记录手术信息的字段都提取出来，然后依据优先级来构建数据集。例如，如果存在手术记录那么首先加入手术记录，手术记录的数据长度>512则不再找其他的数据，如果没有“手术记录”则查看“诊疗经过”中是否有手术信息，依此类推构建数据集。

**七、360视图**

所谓的360试图其实就是利用NER与正则化技术将病历基本新进行结构化展示，辅助动图给人以直观化的感受。360视图主要包括病史及体征、辅助检查、诊断信息、治疗信息。其中病史主要采用NER和正则化对既往史进行结构化，主要展示既往疾病、既往诊断、既往用药等信息。体征主要利用NER对“体格检查”进行结构化，辅助检查主要利用NER对“辅助检查”部分进行结构化。这里主要调用前期的结构化工程的API接口，以及正则化技术。





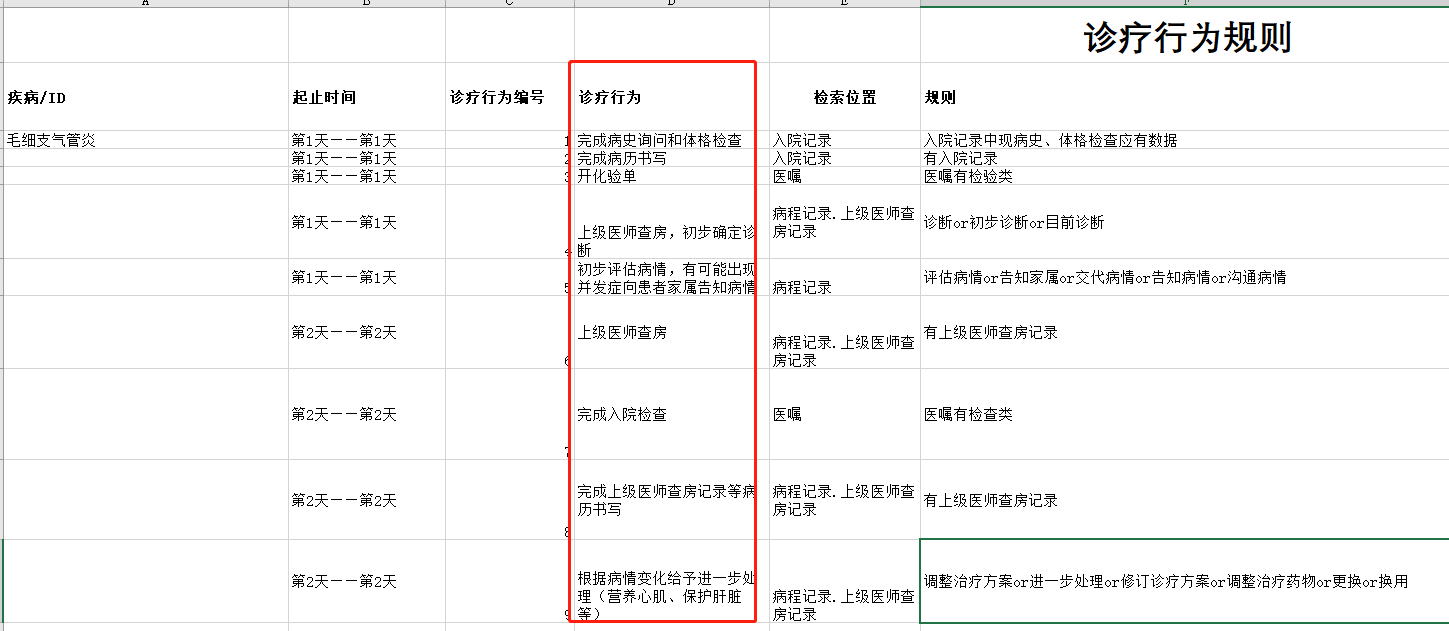
**八、数据库重构**

所谓的数据库重构其实就是根据产品的功能需求对原始的病案数据进行数据抽取，将抽取的信息重新存入到新的数据库（数据表）中，并开放相应的API接口供不同的开发团队调用。

首先是360视图组，根据需求360视图需要展示的信息多达近百个字段，而这些字段在原始病案数据中并没有固定的字段与其对应，因此必须利用NER和正则化技术从某些字段中提取出所需的信息，然后存入对应的新的数据库中，并开放相应的API结构供360视图项目调用。

临床路径，根据卫生部2016年公布的112个临床路径表，从原始病案中获取对应的信息存入新的数据库并开放API接口。目的为后期的质控和DRG分组提供服务。根据临床路径表和从病案中提取的对应信息判断某一个病人在住院期间的某项检查或者治疗是否是合理的，从而判定某项费用的支出是否合理。112个临床路径表，需求的字段多达400+个，因此需要耗费大量的时间和精力来完成。

例如下图是“毛细支气管炎”对应的临床路径所需的字段及提取规则，我们需要根据规则从原始数据的相应字段中提取所需的信息。



**九、DRG分组支付保障**

DRG（疾病诊断相关分组）分组的主要目的是医保控费，目前的医保主要是根据诊疗项目来进行报销，因此存在很严重医保滥用的现象。为了达到医保的合理利用，因此提出了基于DRG分组的新的医保执行标准。

根据病人的入院记录信息，先期预测该病人可能的分组，将为医生的后期诊断和治疗提供参考。（本部分专业性较强，参与较少，所以不是很很熟悉）



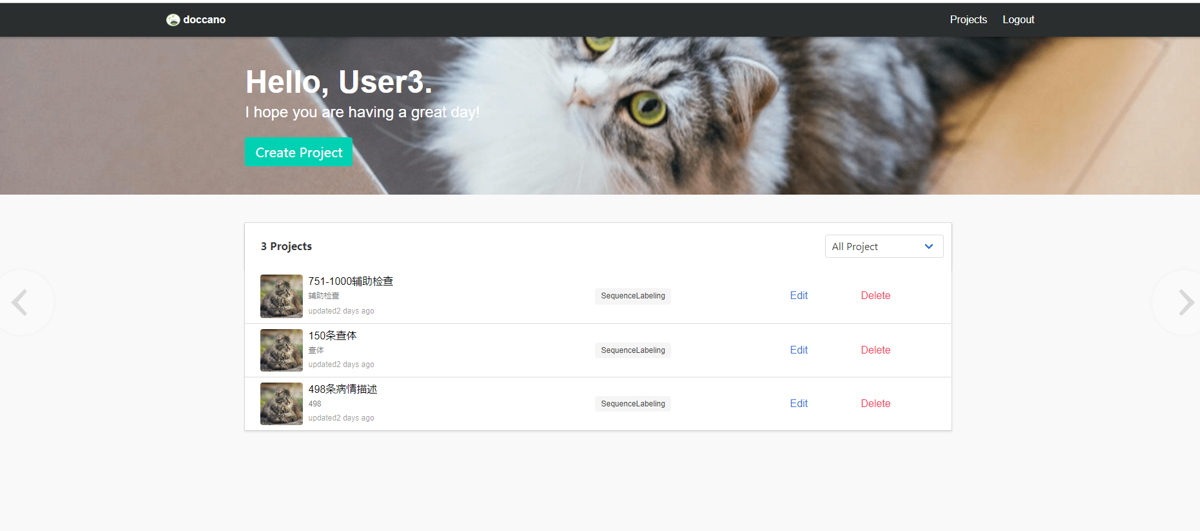
**标注工具升级**

目前的标注工具无法满足多个模型的标注需求，主要是因为目前的标注工具中类别标签是固定的，当有新的标注需求时，需要通过修改代码来改变类别。还有在审核任务中表现得不是很方便，因此需要对现有的工具进行升级。

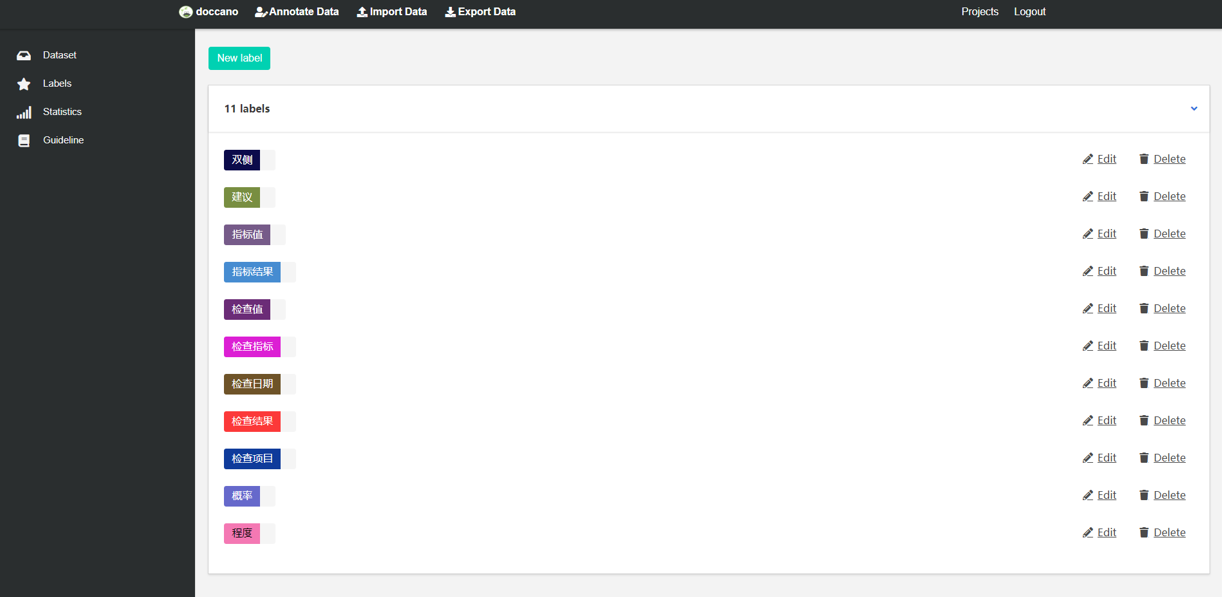
参考开源工具：<https://github.com/doccano/doccano>

开源工具demo：<http://doccano.herokuapp.com/demo/named-entity-recognition/>

1. 增加用户管理管理功能，管理员可以为标注用户添加（分配）标注任务，标注用户登录后即可查看管理员分配的标注任务。
2. 增加标签修改功能，用户可以根据具体标注任务，自行添加、修改标注类别标签。对于纠错数据在管理员为用户分配任务后，同时为用户设定初始标签类别。
3. 权限管理，管理员可创建标注任务，修改标注任务，删除标注任务，对于标注完成的任务管理员具备导出数据的权限。而普通标注用户只具备修改某一个具体任务的类别标签，删除某条标注数据的权限。普通标注用户不具备新建标注任务，导出数据的功能。
4. 标注进度跳转，用户标注页面增加翻页功能，同时增加快速翻页功能（根据输入的页面数快速跳转到对应的页面）。

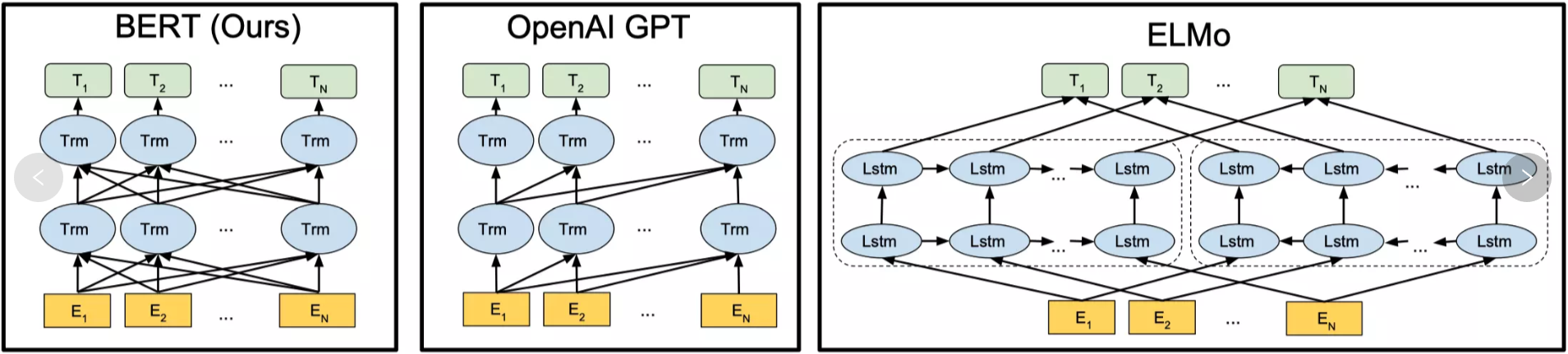


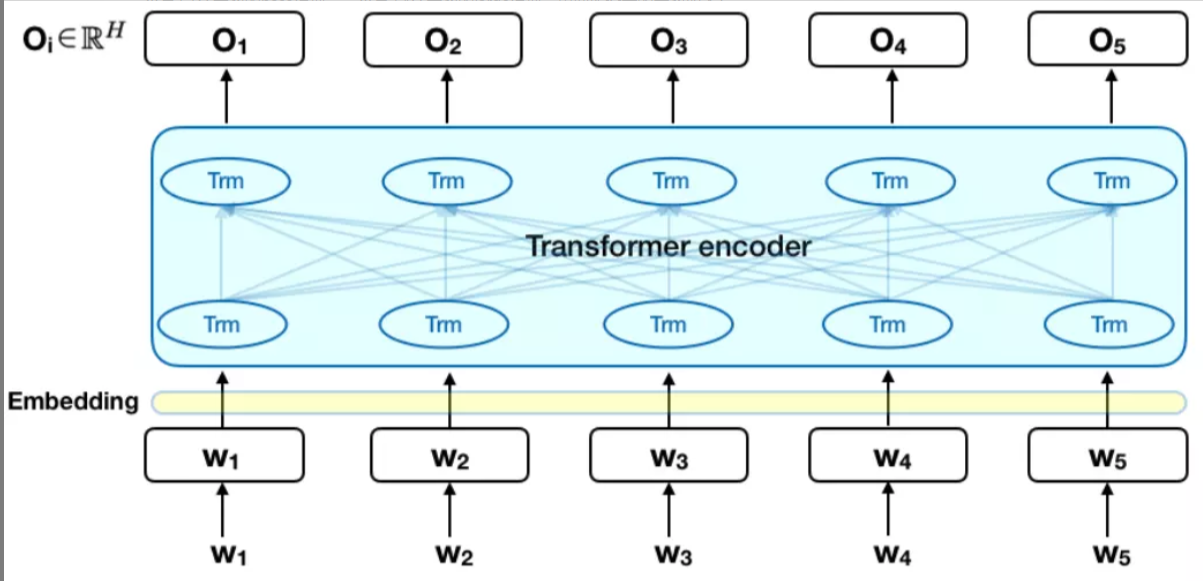




**BERT简介**

BERT(Bidirectional Encoder Representations from Transformers) 自google在2018年10月底公布BERT在11项nlp任务中的卓越表现后，BERT就彻底改变了预训练产生词向量和下游具体NLP任务的关系，提出龙骨级的训练词向量概念。





预训练的语言模型对于众多自然语言处理问题起到了重要作用，比如SQuAD问答任务、命名实体识别以及情感识别。目前将预训练的语言模型应用到NLP任务主要有两种策略，一种是基于特征的语言模型，如ELMo模型；另一种是基于微调的语言模型，如OpenAI GPT。这两类语言模型各有其优缺点，而BERT的出现，似乎融合了它们所有的优点，因此才可以在诸多后续特定任务上取得最优的效果。

参考资料：

1、<https://www.cnblogs.com/huangyc/p/9898852.html>

2、<https://www.jianshu.com/p/160c4800b9b5>

3、<https://www.jianshu.com/p/bfd0148b292e>

4、<https://baijiahao.baidu.com/s?id=1623243370308313370&wfr=spider&for=pc>

参考论文：

[1] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

<https://arxiv.org/abs/1810.04805>

[2] Attention Is All You Need <https://arxiv.org/abs/1706.03762>

参考模型：

[1] BERT-BiLSTM-CRF：<https://github.com/macanv/BERT-BiLSTM-CRF-NER>

[2] BiLSTM-CRF：<https://github.com/Determined22/zh-NER-TF>

[3]：LatticeLSTM：<https://github.com/jiesutd/LatticeLSTM>

[4]：Parallel\_LatticeLSTM：<https://github.com/LeeSureman/Batch_Parallel_LatticeLSTM>