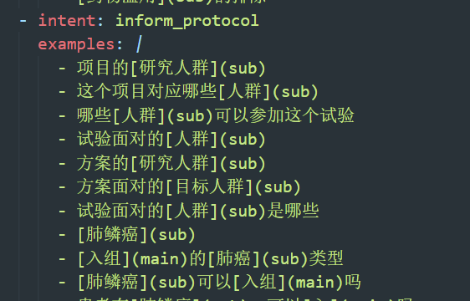
**DCTA 测试报告**

**本报告包括NLU测试和反向测试两个部分。NLU是验证RASA对意图(intent)和实体(entity)的判断。主要指标是confusion matrix, F1 score. 反向验证是指从对话结果中抽取对话的实体，和系统返回的结果，再和KG的结果再次核对。**

1. NLU 测试

1.1 系统识别的过程

系统的NLU训练集(nlu.yml）现在有466句语料，示例如左：

这个例子中，在意图(intent) inform\_protocal 下，定义好对应的训练数据。例如：

肺鳞癌可以入组吗？

其中，肺鳞癌的实体（entity）类型是sub, 入组是实体类型是 main. 通过定义训练语料，我们可以训练系统从自然语言的对话中，识别对话的意图和相关的实体。

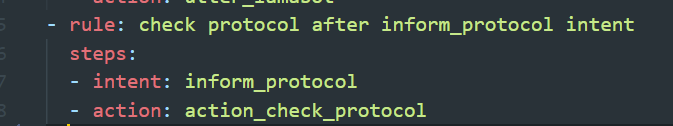
在具体的实例类型中，可以定义不同的值(value). 这个例子中，我们得到：、

Intent: inform\_protocol

Entity: sub, value: 肺鳞癌

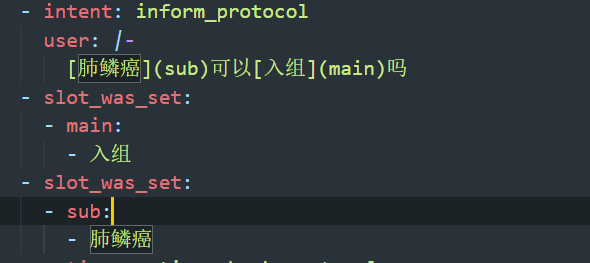
Entity: main, value: 入组

之后，我们定义rules, 就是更加对话的意图，执行下一步的动作



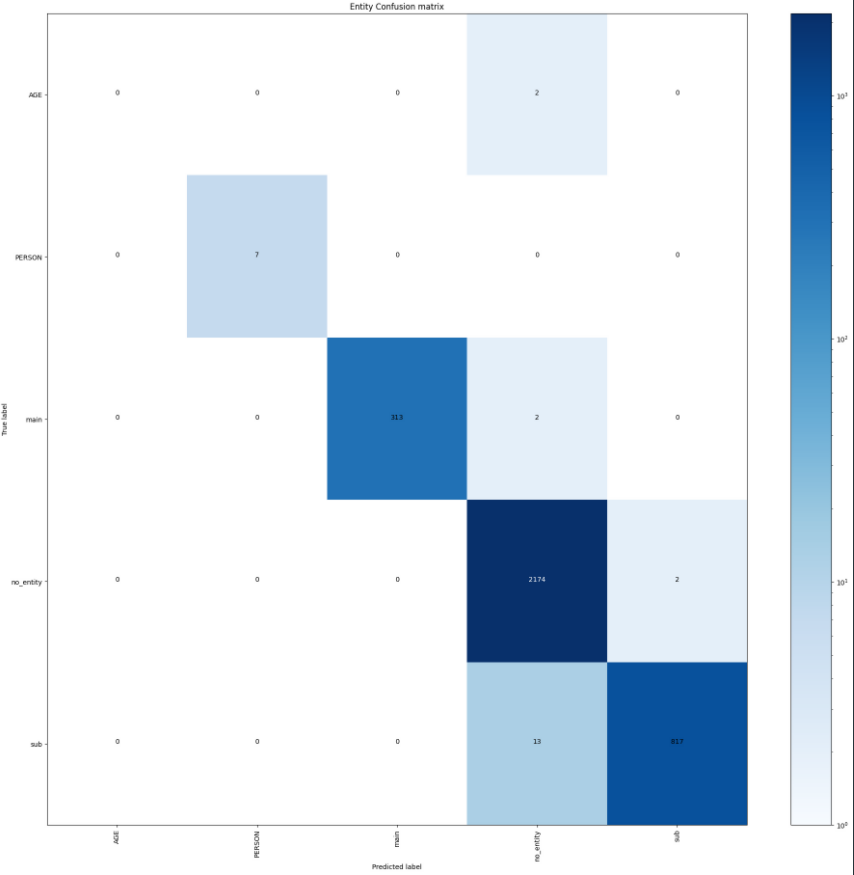
这个例子中，在识别对话的意图是inform\_protocol后，系统会执行action\_check\_protocol. 在这个过程中，相关的entity( sub: 肺鳞癌，main: 入组)都会代入 action\_check\_protocol.

1.2 测试方法

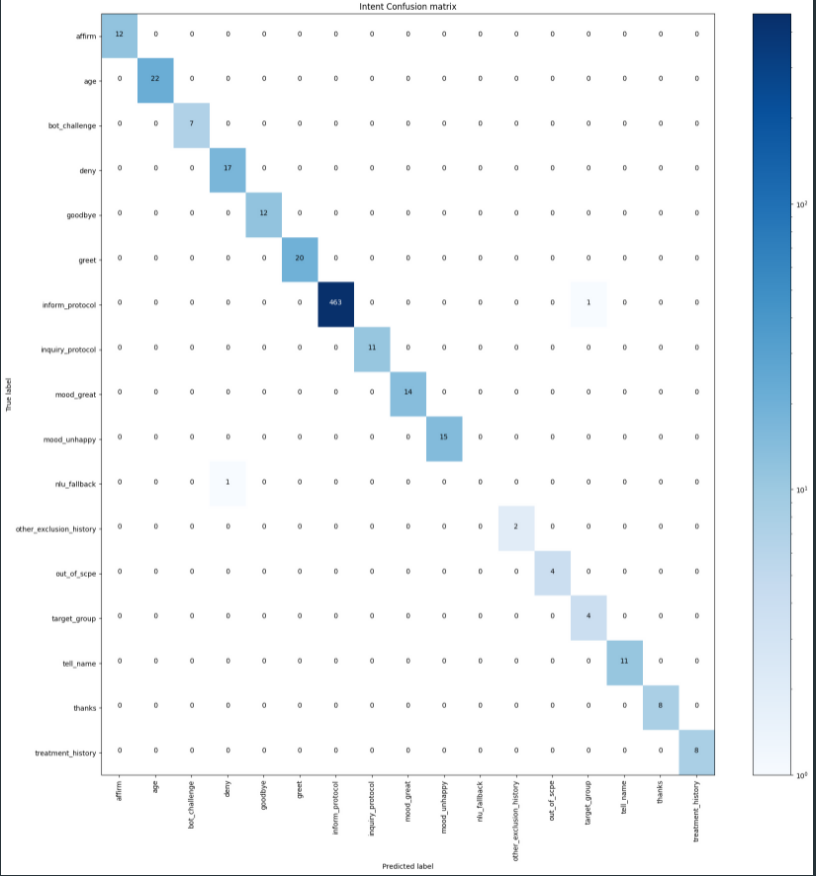
我们会按照对话的语料，生成对话文件。还是上面的例子，当用户问“肺鳞癌可以入组吗”’时，系统应该识别出意图是inform\_protocol, sub entity 是肺鳞癌， sub main 是入组

我们把写好的test stories 按照这个格式开始运行测试脚本。系统会比较模型预测的结果，和test story 设的结果，按照intent, entity 统计结果

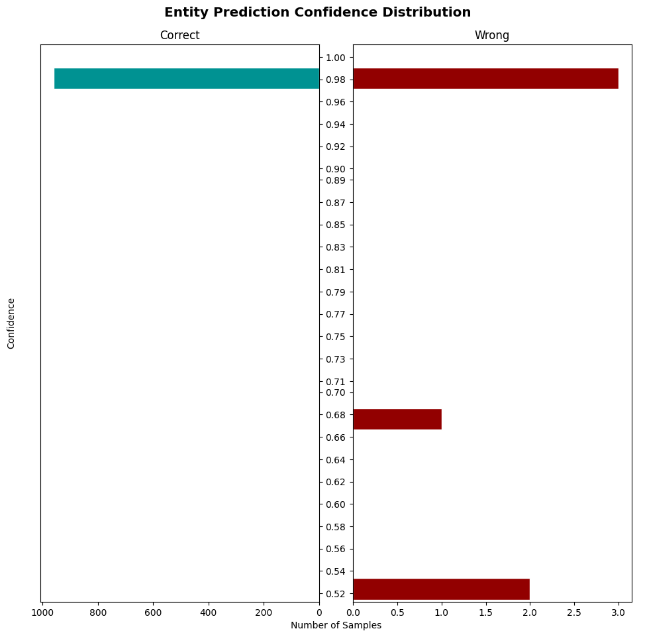
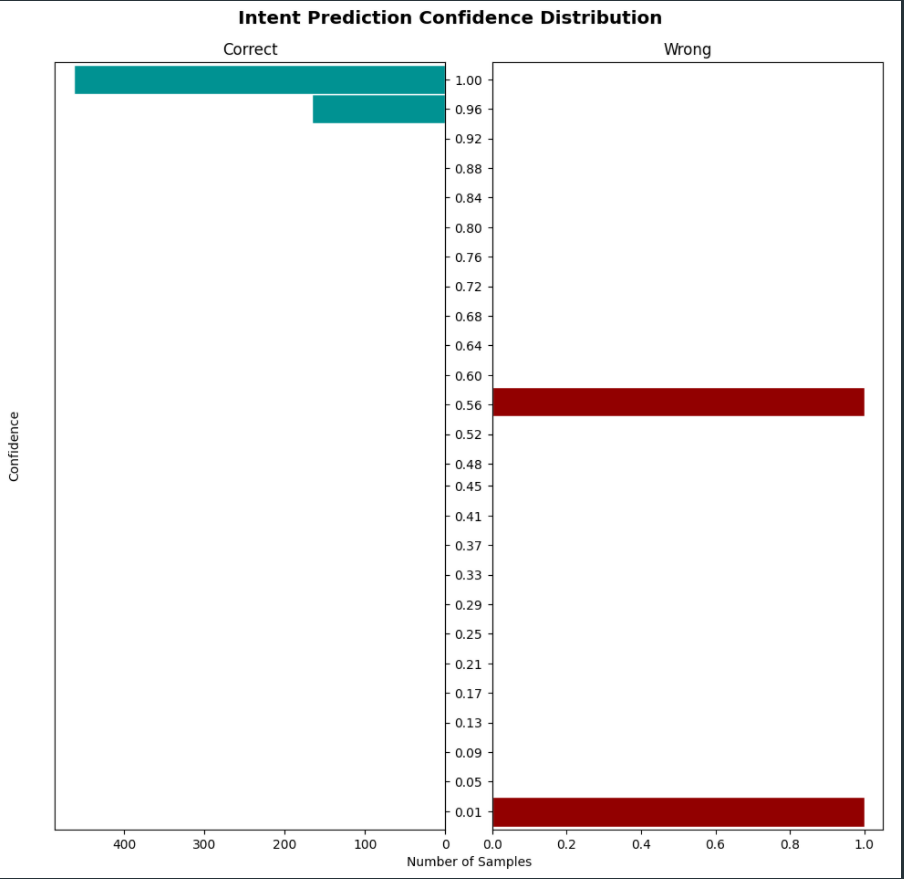
2. NLU测试结果



上面对entity例如对于sub，系统有817个预测是正确的，13个系统没识别出



上面对意图Intent 的分析，例如对于463个informa\_protocol, 系统都可以正确识别



上面分别是intent和Entity的confidence 分析。

这个是entity的F1 score

3. 反向验证

由于Rasa测试是对应NLU的测试，我们需要另外执行方向测试。就是从对话结果中，取得系统抽取的entity, 以及访问KG的结果，再运行一个测试脚本，验证是否正确。这个结果应用是100% 正确。对应出现问题的对话，系统会输出的错误的文件中。

