北京理工大學
BEIJING INSTITUTE OF TECHNOLOGY

调研报告

题　　目	：	Multi-QA与KPCA融合模型
姓　　名	：	贾昌国
学　　号	：	1120200664
专　　业	：	计算机科学拔尖班
授课教师	：	辛欣

2022 年 6 月 5 日

Multi-QA与KPCA融合模型

1. 项目简介

1.1 项目目标

若有两个存在着关系的实体，我们可将两个实体分别成为主体和客体，那么关系抽取就是在非结构或半结构化数据中找出主体与客体之间存在的关系，并将其表示为实体关系三元组，即（主体，关系，客体）。

本次任务主要以2019年发表在ACL中的论文《Entity-Relation Extraction as Multi-turn QuestionAnswering》提出的Multi-QA模型为基础，在复现模型的基础上加入了一些机器学习中的有关算法，最后使用ACE2005数据集进行训练测试。

1.2 项目内容

本次项目主要包含以下内容：

- 数据集预处理：使用$ACE2005$数据集，预处理方法使用论文中的$Multi-QA$方法，结合论文公开源码中的$template$产生针对$ACE2005$数据集的$Question$，以此对数据集进行处理。
- 模型组成：首先以$BERT$模型为基础，使用MLM、双向$Transformers$对数据进行编码后，然后使用线性模型结合$KPCA$算法对数据进行处理。
- 预测与评估：针对预测任务编写$Precision$、$Recall$和$F1-measure$用于模型效果的检验与评价。

1.3 项目难点

当下流行的关系抽取方法主要有两种：

- pipeline

  先从文本中抽取全部实体$(e_1, ..., e_n)$，然后针对全部可能的实体对(e_i, e_j)，$i \neq j$，判定实体对之间的关系类别。

- joint extraction

  通过修改标注方法和模型结构直接输出文本中包含的(e_i, r_k, e_j)三元组。

当前存在的方法主要存在的不足有：

- Formalization Level

  关系三元组不能充分地表达结构化数据，不同的tags之间有一定的层次依赖。

- Algorithm Level

  现有的方法很难从文本中捕捉到词法、语义和句法特征，特别是碰到以下情况：

  a. 实体距离主体语句很远

  b. 一个实体在多个语句中出现

  c. 语义关系存在重叠问题（overlap），主要包含以下三种类型：

  - $Single\ Entity\ Pverlap(SEO)$——两个三元组之间有一个实体重叠
  - $Entity\ Pair\ Overlap(EPO)$——即一个实体对之间存在多种关系
  - $Normal$——两个实体之间仅有一种联系且不与其他实体产生联系

2. 模型设计

2.1 预处理模型设计

目前主流MRC模型都是从给定的文本中寻求答案。Text span extraction可以简化为两个多分类任务，如预测答案的开始和答案的末尾。一般的MRC策略可以扩展到多篇章的MRC任务，多篇章MRC又可以通过拼接段变成单篇章的MRC任务：首先对篇章进行排序，然后在选中的篇章上做MRC。现在有一个大的趋势是很多非QA任务当作QA任务来处理。论文中把关系抽取任务看作一个多轮QA任务来处理。本次项目借用其思想，采用Multi-QA算法对数据集进行预处理。

下图是预处理模型的基本算法流程：

Input: sentence s, EntityQuesTemplates, ChainOfRelTemplates
Output: a list of list (table) M = []

$M \leftarrow \emptyset$
HeadEntList$\leftarrow \emptyset$
for entity_question in EntityQuesTemplates **do**
    e_1 = Extract_Answer(entity_question, s)
    if $e_1 \neq$ NONE **do**
        HeadEntList = HeadEntList + $\{e_1\}$
    endif
end for
for head_entity in HeadEntList **do**
    ent_list = [head_entity]
    for [rel, rel_temp] in ChainOfRelTemplates **do**
        for (rel, rel_temp) in List of [rel, rel_temp] **do**
            q = GenQues(rel_temp, rel, ent_list)
            e = Extract_Answer(rel_question, s)
            if $e \neq$ NONE
                ent_list = ent_list + e
            endif
        end for
    end for
    if len(ent_list)=len([rel, rel_temp])
        M = M + ent_list
    endif
end for
return M

2.1.1算法流程简述

算法大概分为两步：

1. 头实体抽取（4-9行）
   多轮QA的每个阶段是通过一个实体进行触发。为了抽取开始的实体，文章将每个实体类型，使用EntityQuesTemplates和实体e转换为一个问题，然后这个实体e是从问题的答案中抽取的。如果系统输出None标记，表明文本中无实体。
2. 尾实体和关系抽取（10-24行）
   ChainOfRelTemplates定义了一系列的关系，我们需要遵循关系的顺序来运行多轮QA。因为一些实体的抽取依赖其他类型实体的抽取。因为不同实体之间还是存在依赖的，比如不同时间实体，同时抽取出来的岗位也是对应的。而抽取顺序则是手工预定义的。
   ChainOfRelTemplates为每个关系定义模板，每个模板包含许多待填槽值。为了能生成问题，我们需要插入之前抽取出来的实体到这些槽值中。而关系和尾实体会通过回答这些问题，联合地抽取出来。

值得注意的是，从头部实体提取阶段提取的实体可能并非都是头部实体。在随后的关系和尾部实体提取阶段中，首先假设来自第一阶段的提取的实体是头部实体，并且将其提供给模板以生成问题。如果从第一阶段提取的实体e确实是关系的头部实体，则QA模型将通过回答相应的问题来提取尾部实体。否则，答案将为NONE，从而被忽略。

2.1.2 用模板生成问题

借助于论文提供的源码，结合数据集ACE2005，我们有两种基于template生成问题的方法：

1. natural language question

   例如：对于Facility 类型，生成的问题可能是：*Which facility is mentioned in the text*?

2. pseudo-question

例如：对于Facility类型，生成的可能只是：*entity: facility*

RESUME			
Model	Overall P	Overall R	Overall F
Pseudo Q	90.2	92.3	91.2
Natural Q	91.0	93.2	92.1

ACE04						
Model	EP	ER	EF	RP	RR	RF
Pseudo Q	83.7	81.3	82.5	49.4	47.2	48.3
Natural Q	**84.4**	**82.9**	**83.6**	**50.1**	**48.7**	**49.9**

ACE05						
Model	EP	ER	EF	RP	RR	RF
Pseudo Q	83.6	84.7	84.2	60.4	55.9	58.1
Natural Q	**84.7**	**84.9**	**84.8**	**64.8**	**56.2**	**60.2**

CoNLL04						
Model	EP	ER	EF	RP	RR	RF
Pseudo Q	87.4	86.4	86.9	68.2	67.4	67.8
Natural Q	**89.0**	**86.6**	**87.8**	**69.6**	**68.2**	**68.9**

由论文中的对比结果可以得出natural questions明显得分更高，效果更好。

2.1.3 获取关系范围

对于QA框架，文章使用BERT作为主干。为了与BERT框架保持一致，通过连接列表[CLS，Q，SEP，C，SEP]来组合问题Q和上下文C，其中CLS和SEP是BERT模型中使用的特殊标记，Q是标记化问题，C是上下文。使用多层变换器获得每个上下文标记的表示。

2.1.4 Multi-QA模型的优点

这种多转QA形式化具有几个关键优势：

1. 多轮QA能够更好地捕获标签的层次依赖性。
2. 问题查询为我们想要识别的实体/关系类编码重要的先验信息。
3. QA框架提供了一种同时提取实体和关系的自然方式：支持输出特殊的NONE标记，表明该问题没有答案。

2.1.5 预训练模型的一些特殊优化

1. 计算实体间的距离后，通过将距离与规定的阈值比较来过滤一些频率小的不可能存在的关系。
2. 使用最大距离来进一步过滤一些尾部实体类型不在头部实体范围内的不可能存在的关系。
3. 使用两个特殊标记来标记上下文中的头实体，以避免窗口中的coreference。

2.2 训练模型设计

2.2.1 原论文模型的缺点

论文中《Entity-Relation Extraction as Multi-turn QuestionAnswering》中介绍的训练模型较为简单，模型输入后经过BERT改编处理后，通过一层线性模型和dropout层后即输出结果。但是原文模型的后半部分为了迎合Multi-QA模型的预处理结果，仅仅对非线性数据强行进行线性处理后便得出结果，并不能很好地保留非线性数据原本的特性。

2.2.2 项目模型简介

本次项目中的训练模型主要包含以下几个部分：

1. BERT模型的编码部分
2. 线性模型
3. KPCA算法

4. dropout
5. BERT模型的解码部分

2.2.3 BERT模型编码解码部分与dropout部分

BERT模型的编码部分和解码部分：生成深度的双向语言表征。输入为每一个token对应的表征，单词字典是采用WordPiece算法进行构建的。为了完成具体的关系抽取任务，除了单词的token之外，我们还在输入的每一个序列开头和结尾都插入特定的分类标记，该分类标记对应的最后一个Transformer层输出起到聚集整个序列表征信息的作用。

dropout层主要用于防止过拟合。

2.2.4 KPCA算法

核主成分分析（KPCA）主要用来实现数据的非线性降维，在降维的同时保持数据的特性。其核心思想是通过一个非线性映射把原始空间的数据投影到高维特征空间，然后在高维特征空间中进行基于PCA的数据处理。

算法的流程如下：

1. 首先用一个映射$\phi : R^m -> R^n, m < n$，将样本X映射到高维空间，得到高温空间的数据矩阵$\phi(X)$，
   $\phi(X) = [\phi(x^{(1)})\ \phi(x^{(2)}) \ldots\ldots \phi(x^{(n)})]$
2. 计算高维空间中数据矩阵的协方差矩阵$\phi(X)\phi(X)^T$，进一步计算特征值与特征向量。
3. 将特征向量按照对应的特征值大小从上到下按行排列成矩阵，取前k行组成矩阵P，P即为降维后的数据。

在对数据进行非线性映射的时候，用到的映射一般称作核函数。当前常用的核函数主要有以下几种：

1. 线性核函数

$$K(x,y)=x^Ty+c$$

2. 多项式核函数

$$K(x,y)=(\alpha x^Ty+c)^d$$

3. 高斯核函数

$$K(x^{(i)},x^{(j)})=exp(-\gamma*||x^{(i)}-x^{(j)}||^2)$$

4. 指数核函数

$$K(x,y)=exp(-\frac{||x-y||}{2\sigma^2})$$

5. 拉普拉斯核函数

$$K(x,y)=exp(-\frac{||x-y||}{\sigma})$$

这里我们采用常用的高斯核函数。

得到$K(x,y)$之后还需要对数据进行中心化处理：

$$K'=K-I_n*K-K*I_n-I_n*K*I_n$$

得到$K'(x,y)$之后求出其特征向量和特征值，将特征向量和特征值从大到小排序按行拼接之后按照主成分参数要求进行裁切，得到数据降维后的结果。最后，将降维后的结果通过线性模型的处理与BERT的格式相对应。

2.3 损失函数与评价准则

本项目使用交叉熵损失函数来计算关系抽取的预测误差，评价指标采用$F1\ measure$。

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

2.4 对于模型的一些特殊优化

2.4.1 联合训练

在测试时，模型基于关系抽取目标分别提取头部实体和尾部实体，通过一个目标函数联合地训练了算法的两个步骤：

$$L = (1 - \lambda)L(headentity) + \lambda L(tailentity, rel)$$

2.4.2 强化学习

因为从一轮问答中提取答案，不仅仅会影响它自己的精度，也会影响到后面几轮的精度，所以文章中使用了强化学习的方法来解决。

模型的QA依赖于标记输出，所以选择确定的范围$\{\omega_1, \omega_2, ... \omega_n\}$是$\omega_1$属于B的联合概率，$\omega_2, \omega_3, ..., \omega_{n-1}$属于M的联合概率，$\omega_n$属于E的联合概率分布：

$$p(y(\omega_1, ..., \omega_n) = answer|question, s) =$$
$$p(\omega_1 = B) \times p(\omega_n = E) \prod_{i \in [2, n-1]} p(\omega_i = M)$$

而Reward对于句子s，文中使用正确检索三元组的数量作为奖励，用REINFORCE算法来计算策略梯度找到最大化奖励函数的策略。

$$\nabla E(\theta) \approx [R(\omega) - b] \nabla log \pi(y(\omega)|questions)$$

其中b表示baseline value，每轮QA，正确回答则reward+1，最后的奖励是多轮累积的结果。baseline value是所有前面的奖励的平均。并且模型不初始化策略网络，而是使用预训练的头尾实体抽取模型来描述之前的部分。

3. 项目实现

3.1 文件说明

下面主要介绍本次项目的代码实现过程中主要生成的文件。

1. 数据文件：
    a. ACE2005文件夹中存储了train、dev和test文件夹，分别表示本次项目中的训练集、测试集和验证集。
    b. query-templates文件夹中生成的ace2005.json文件主要是用论文源码生成的针对ACE2005数据集的一系列questions。
    c. cleaned_data文件夹中的json文件主要是数据预处理后的结果。
    d. checkpoints文件夹中的cpt文件主要用来保存训练过程中的各种参数。
2. 代码文件：
    a. preprocess.py文件主要用来进行数据预处理；
    b. model.py主要用来保存训练模型；
    c. train.py主要结合预处理结果以及调用model来进行训练；
    d. dataloader.py主要用来进行训练数据集加载等工作；
    e. evaluation.py主要用来保存模型预测的相关函数；
    f. ckpt_eval.py主要用来调用模型预测的函数并且保存相关结果；
    g. constants.py主要用来保存项目中的一些参数；

h. generate_query.py主要用来针对ACE2005数据集进行处理，生成Multi-QA所需要的questions。

3.2 模型对比试验

实验项目环境：

nvidia driver 470.103.01

3.2.1 Multi-QA与Multi-QA+KPCA对比

Model	Entity P	Entity R	Entity F	Relation P	Relation R	Relation F
Multi-QA	73.6	73.3	73.9	57.9	53.4	54.4
Multi-QA+KPCA	73.5	73.6	73.7	59.3	56.8	58.7

限于实验项目环境与时间等因素，对于Multi-QA模型的最终评分无法达到论文中的高度，故对比实验时两模型都采用现有环境（nvidia driver 470.103.01，cuda11）。最终预测评分时对两类（$Entity$和$Relation$）预测分别评分，得到了$Precision$、$Relation$和$F1\ measure$结果如上图所示。从图中可以看出，在$Entity$的预测上，两种模型几乎没有差别，评分大抵相当；但是在$Relation$的预测上，Multi-QA+KPCA，也即改进后的模型，有显著的优势。由此可以看出Multi-QA+KPCA模型对于结果的预测要优于原论文中的模型。

3.2.2 λ参数的对比实验

本次项目在抽取头部实体和尾部实体的时候，通过参数λ将两者结合起来进行联合训练，参数λ控制着两个子任务的tradeoff。

$$L = (1-\lambda)L(headentity) + \lambda L(tailentity, rel)$$

对λ做对比实验后结果如图所示。

Multi-QA+KPCA

λ	Entity F	Relation F
λ=0	72.9	58.4
λ=0.2	72.6	58.2
λ=0.4	73.7	57.9
λ=0.6	73.6	58.5
λ=0.8	73.1	58.7
λ=1.0	73.4	57.2

可以看到，λ参数在$\lambda = 0.4$时$Entity$的$F1\ measure$最大，在$\lambda = 0.8$时$Relation$的$F1\ measure$最大，这与论文中所展示的Multi-QA模型评分最高时的λ值几乎相同（从下图中可以看出，λ参数在$\lambda = 0.2$时$Entity$的$F1\ measure$最大，在$\lambda = 0.7$时$Relation$的$F1\ measure$最大）。

Results regarding different values of λ on the ACE05 dataset are given as follows:

λ	**Entity F1**	**Relation F1**
$\lambda = 0$	85.0	55.1
$\lambda = 0.1$	84.8	55.4
$\lambda = 0.2$	**85.2**	56.2
$\lambda = 0.3$	84.8	56.4
$\lambda = 0.4$	84.6	57.9
$\lambda = 0.5$	84.8	58.3
$\lambda = 0.6$	84.6	58.9
$\lambda = 0.7$	84.8	**60.2**
$\lambda = 0.8$	83.9	58.7
$\lambda = 0.9$	82.7	58.3
$\lambda = 1.0$	81.9	57.8

4. 总结与反思

本文提出了一种全新的视角来审视关系抽取任务。Multi-QA+KPCA融合模型的建立是以Multi-QA模型为基础，结合了KPCA算法，最终生成的模型性能相较于原模型有一定提升。

优点：

1. 本文很好地利用了KPCA方法可以更好地保存数据现有特征的特点，对数据进行降维处理。
2. 通过阈值的调节很好地减弱了overlap对实验结果的影响。

缺点：

1. 由于项目环境的配置问题，导致在与原模型做对比实验时，原模型的效果与论文中的效果差距较大，无法得到模型的最佳效果。
2. 做的对比实验个数较少，对于一些参数（例如 *learning rate*、*batch size*、*epoch*等），均采用原论文模型给出的最佳参数，甚至在原论文模型的基础上对参数进行削弱以减少运行时间成本，并没有确定适合于改编后模型的最佳参数。