SCIgen生成文献的检测方案

刘昌灵
中国人民大学 2012级本科
多媒体计算实验室

简单的发现

- SCIgen是一个使用生成式生成文本的上下文无关文法（巴科斯范式、巴科斯-诺尔范式、 Backus-Naur Form）
- SCIgen使用的生成式存在于scirules.in文件中
- 对于子生成式的展开，SCIgen的策略相对简单，不会自递归（乔姆斯基范式、Chomsky Normal Form）

通用的解决方案

- 提取特征，观察其分布
  - 词频、词距等
  - Sklearn提供了一部分文本特征
  - 使用Word2Vec准备意义空间的分析
- SVM/多层SVM
- 神经网络/循环冗余神经网络

SVM/神经网络

- 选择SVM的原因是特征空间（至少词频空间）对于正反样本较为可分（高斯核，甚至线性核都可以得到不错的效果）
- 选择RNN（Recurrent Neural Network）的原因为其每一次的运算都具有后效性，可以在优秀的上下文环境中分析语义空间。而语义空间是期望具有显著差别的。

SVM/NN总结

- 优点：
  - 能够快速适应SCIgen做出的改动（更换/添加特征向量）
  - 解决办法通用，主要的训练学习任务交给机器自己完成，较为智能
  - 易于分析意义空间
- 缺点：
  - 训练可能需要较长的时间，运行也有较高的复杂度。通常需要GPU支持。
  - 需要寻找大量的训练数据
  - 需要局限于SVM/NN的模型

里应外合

- 从内部攻破敌人
  - 学习SCIgen的组合方式（RNN-RBM?）
  - 除去可能为固定生成的词语
  - 分析可能为随机生成的部分
- 从外部强硬打击
  - 除去同义词干扰
    - （可能的）除去句式干扰
  - （可能的）联网操作
    - 查重
    - 查参考文献
    - 对于意义空间的操作

小点子

- 在能够分析词性的条件下
  - 名词与动词是主要影响意义空间的部分
    - 猜想I：名词与动词的意义可以组合成一个向量
  - 形容词与副词影响强度、与上组合影响极性
    - 猜想II：是否能将强度叠加于名词与动词
      - 词向量的叠加是一个非常复杂的过程，通常需要神经网络进行分析，这里仅作为一个方向导出

小点子

- 关于联网操作
  - 部分网站对于教育网提供查重API
    - 较高资费、且对于SCIgen不是很具有指导意义
  - 查参考文献
    - 知网、谷歌学术可以简单的实现
  - 对于意义空间的操作
    - 除去句式影响后可以根据关键字获取类似的文献
    - 比较意义向量的距离

海阔天空

- 小点子中很多点其实是神经网络的优化方向（适用于RNN）
- 比起SCIgen Cracker，成果更像是一个基于意义空间的相似文章查找器
- 比起查SCIgen更适合查论文的重复发明
- 代码量巨大（即使使用Theano、caffe等高级封装），且中间有收费项目、可能会遇到验证码的项目
- 若有兴趣，我校多媒体计算实验室拥有一定的成果可以参考

其他的办法

- 想要Crack SCIgen其实没有那么复杂
  - 回顾“简单的发现”
- Think in simple way, not lazy way.
- 顾客要一块石头，我们尝试卖给他一粒钻石

其他的办法

- 我们所拥有的信息
  - SCIgen源程序
    - SCIgen的所有生成式
    - SCIgen组合生成式的方式
  - SCIgen框架上可能的改进方向
    - 更新生成式，甚至自动生成生成式
    - 更复杂的组合方式（符合更多方式）
- 生成式?
  - 我相信SCIgen的灵感来源于编译原理

简单的想法

- 我们可以从scirules.in构建一个非递归语法
- 检测论文类似Yacc（Yet Another Compiler-Compiler）程序使用规约式规约巴科斯范式的方法，规约乔姆斯基范式。若规约成功则认为很有可能是SCIgen生成的。
  - 编译原理的小知识

举个例子

- 对于生成式
  - 1）SCI_TITLE_X deconstructing SCI_THING SCI_TITLE_POSTFIX
  - 2）SCI_THING　　SCI_THING_P
  - 3）SCI_THING_P　　suffix trees
  - 4）SCI_TITLE_POSTFIX　　using SYSNAME
  - 5）SYSNAME　　singular value decomposition
- 尝试规约
  - deconstructing suffix trees using singular value decomposition
  - 1）→规约掉deconstructing
    - SCI_THING → suffix trees; 完成
    - SCI_TITLE_POSTFIX → using singular value decomposition
  - 4）　→规约掉using
    - SYSNAME singular value decomposition; 完成

关于实现

- 简单的使用正则表达式，规约掉最先看到的模式
- 为了兼容一定的改进余地，规约到完整句子便不再进行规约（变量中所有子变量已经规约，并且再次规约会导致句点变多）
- 如果只规约一次呢?

实验效果

- 对于SCIgen生成的文章，总是有60%以上的句子是可疑的
- 对于一般的人工文章，总是有15%以下的句子是可疑的

- 阈值定义为40%

- 100%的识别准确率
  （在正反样本大小皆为100的测试集合上）
- 可以输出哪些句子是可疑的

题外话：神经网络经验分享

- 曾经做pyRNN, 基于Theano的GPU运算（Python 2.7）
- 对于特定问题，比起改进网络来说，如果能开发具有针对性的feature，效果将提高更快
  - 如果只是研究层的相对顺序和数量，很快可以得到结果
  - 根本性的改进来源于在网络中引入全新的层或特征
- 网络一般只和功能有关
  - CNN　　更好的处理图像，能输出conv map
  - RNN　　为了使输出有前向性，后效性
  - SPP-CNN @ Microsoft　　可以得到bounding box
- 研究网络本身是一个巨大的工程，需要较多的专业知识才能得到灵感
- 一般来说结构组成：
- CNN→RNN→全链接层（用于图像关于时间的序列）
- 过于深层的网络将由于显存限制变的瘦小而降低效果，另本身不需要这么抽象的概念。32位浮点误差较大（少用乘法、乘方）

RNN for pooling

- CNN对Pooling的需求出现于需要对稀疏编码出的矩阵进行向量化
- 举例，Mean Pooling：
  - Forward的时候对窗取平均
  - Backward的时候将平均值作为原值处理
- 实际上作用和模糊类似
- 为什么RNN可以用来pooling
  - 可以达到矩阵向量化的目的
  - 减小量化误差的基础上增加前文环境

进一步思考

- 如何把上述“抖机灵”提出的规约向量化，成为feature
  - 分l和w → 2-D tensor（matrix）
  - 匹配位置的分布？分l，w，P → 3-D tensor
  - 匹配距离？L，w，pattern，pattern → 4-D tensor
- 之后如何降维?
- 这个feature除了查SCIgen以外有什么用?
  - 检查语法
  - 检查坏的写作习惯