

大数据环境下信息抽取模板自动发现与聚类

计 92 丘骏鹏 2009011282

指导老师：朱小燕 郝宇

Outline

Introduction

Idea & Design

Schedule

References

Topic

Introduction

Idea & Design

Schedule

References

Background

- 已经获取到海量的新闻、博客、论坛等网页原始数据，需要从中提取结构化的信息
- 从已有数据中抽取模板，利用模板去抽取相似网页中的信息
- 新的网页可能采取新的模板，需要自动检测，分类和抽取这些新的模板

Motivation

- 一个网站可能采用多个模板，如果采用人工标注模板的办法工作量很大
- 前期同学的工作主要基于一个网页检测其中的模板，没有利用大数据的优势
- 充分利用数据的冗余性来进行模板的抽取

Related Work

- 计算 HTML 文档结构相似度
 - 基于 DOM Tree 本身的方法: Tree Edit Distance[1]。特点: 直接对树结构进行操作, 算法复杂度大, 不适用于大量数据的处理。
 - 基于 Path 集合: 每个节点的路径是根节点到该节点的序列, 将 DOM Tree 用路径集合表示, 在两个网页的路径集合上计算相似度 [4][5][6]。
 - 基于 Tag 集合: 直接计算两个网页的 Tag 集合的 Jaccard 相似度, 或者利用树的遍历将 DOM Tree 转化为 Tag 序列, 然后通过最长公共子序列等方法计算相似度 [4][5][8]。
 - 基于傅里叶变换: 将文档结构转化为一个序列, 将其视为时序序列, 用傅里叶变换变换到频域计算幅度差别 [4][7]。

Related Work

- 网页聚类算法
 - 基于划分的聚类。主要用的是 k-medoids，但需要已知模板个数，对于本问题不太适用。
 - 层次聚类。这种聚类算法不需要预先知道类的个数，速度较快。
- 模板提取算法
 - 无监督方法：利用网页结构和内容的重复出现发现模板，后期需要人工指定语义 [2]。
 - 半监督方法：利用少量标注，进行学习，然后抽取出模板。学习过程可以考虑语义的信息 [3]。

Topic

Introduction

Idea & Design

Schedule

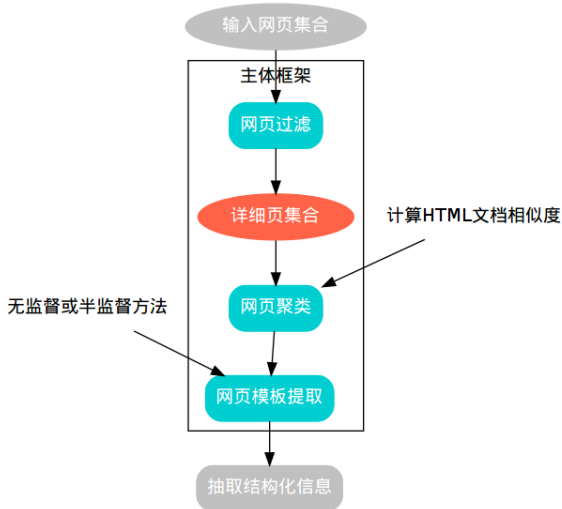
References

Idea

- 实验的数据量决定了设计的算法不能太复杂，但同时数据的冗余也可以弥补算法的粗糙性，从而也能获得较好的效果。
- 每一步可以先采用一些复杂度较低的算法，评价其运行效果和时间，然后再进行改进
- 同时需要考虑可以动态增加模板和进行并行化处理

Framework

整体框架示意图

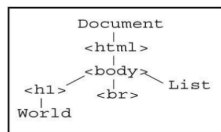


Design

- 网页过滤
 - 网页中含有目录页和详细页，需要过滤掉目录页
 - 要能快速进行计算，不需要过于复杂的机器学习算法
 - 可以采用 URL 特征结合一些文本特征，比如目录页一般是一些超链接列表，且一般都是短文本
- 聚类
 - 计算相似度可以采用基于 Path 集合或者 Tag 集合的方法，并可以利用一些已有的集合相似度的近似算法，降低计算复杂度
 - 采用层次聚类方法进行模板的自动聚类
 - 初步计划参照 [4][5][6] 的方法进行实现

模板表示

- DOM Tree



- Path 集合

Document\<html>	
Document\<html>\<body>	
Document\<html>\<body>\<h1>	
Document\<html>\<body>\<h1>\World	
Document\<html>\<body>\ 	
Document\<html>\<body>\List	

- Tag 集合

Document,<html>,<body>,<h1>,World,
,<List>

相似度计算

- 基于 Path 集合

- Common Path

$$d_{CP}(D_1, D_2) = 1 - \frac{|p(D_1) \cap p(D_2)|}{\max(|p(D_1)|, |p(D_2)|)}$$

- Common Path Shingles

$$d_{CPS}(D_1, D_2) = 1 - \frac{|S(D_1, w) \cap S(D_2, w)|}{\max(|S(D_1, w)|, |S(D_2, w)|)}$$

- MDL

利用路径 - 文档矩阵进行计算, 较为复杂, 根据实验效果决定是否使用

相似度计算

- 基于 Tag 集合
 - Tag Vector

$$d_{TV}(D_1, D_2) = \sqrt{\sum_{i=1}^N (v_i(D_1) - v_i(D_2))^2}$$

- Longest Common Tag Subsequence

$$d_{LCTS}(D_1, D_2) = 1 - \frac{|lcts(D_1, D_2)|}{\max(|D_1|, |D_2|)}$$

Design

- 模板抽取
 - 可以考虑直接利用聚类的结果（例如，一个类中的文档之间的共同路径集合），利用路径集合可以直接表示文档的模板结构
 - 或者采取一些标注，结合一些已有的模板检测算法进行模板的抽取
- 优化和改进
 - 在计算相似度时加入其他特征，和结构特征共同考虑。内容特征，比如文本的行间分布，tag 的属性值，比如某些 div 的 id，CSS 类名
 - 考虑算法的并行性，部署到 Hadoop 集群上，加快算法运行速度

Topic

Introduction

Idea & Design

Schedule

References

日程安排

- 1-4 周：任务分析，文献阅读，研究算法
- 5-8 周：网页过滤，网页聚类 and 网页模板提取模块初步实现
- 9-12 周：算法修正，系统改进，结果分析
 - 是否需要改进相似度算法（包括加入新的特征，改变计算方法）
 - 如何设计并行化计算
 - 如何有效评价系统的功能
- 13-15 周：论文撰写

Topic

Introduction

Idea & Design

Schedule

References

参考文献

1. Reis, Davi De Castro, et al. "Automatic web news extraction using tree edit distance." Proceedings of the 13th international conference on World Wide Web. ACM, 2004.
2. A. Arasu and H. Garcia-Molina. "Extracting structured data from web pages". In Proc. SIGMOD, pages 337–348, 2003.
3. A. Carlson and C. Schafer. "Bootstrapping information extraction from semi-structured webpages". In Proc. ECML, pages 195–210, 2008.
4. Buttler, David. "A short survey of document structure similarity algorithms". United States. Department of Energy, 2004.
5. Gottron, Thomas. "Clustering template based web documents." Advances in Information Retrieval (2008): 40-51.
6. Kim, Chulyun, and Kyuseok Shim. "Text: Automatic template extraction from heterogeneous web pages." Knowledge and Data Engineering, IEEE Transactions on 23.4 (2011): 612-626.
7. S. Flesca, G. Manco, E. Masciari, L. Pontieri, and A. Pugliese, "Detecting structural similarities between XML documents," Fifth International Workshop on the Web and Databases, 2002.
8. 潘有能,《XML 挖掘: 聚类、分类与信息提取》, 浙江大学出版社,2012