基于WEB的计算机领域新术语的自动检测

刘知远

指导教师：孙茂松教授

June 22, 2006

目录

目的

1. 随着计算机技术的迅速发展，汉语需要接纳大量外来术语；
2. 没有权威机构及时检测并给以规范的命名，导致：
   - 蹩脚的翻译，如menu；
   - 多种翻译并存，如Hash Table；
3. 因此亟需一种有效的自动检测新术语的算法。

目的

1. 随着计算机技术的迅速发展，汉语需要接纳大量外来术语；
2. 没有权威机构及时检测并给以规范的命名，导致：
   - 蹩脚的翻译，如menu；
   - 多种翻译并存，如Hash Table；
3. 因此亟需一种有效的自动检测新术语的算法。

目的

1. 随着计算机技术的迅速发展，汉语需要接纳大量外来术语；
2. 没有权威机构及时检测并给以规范的命名，导致：
   - 蹩脚的翻译，如menu；
   - 多种翻译并存，如Hash Table；
3. 因此亟需一种有效的自动检测新术语的算法。

目的

1. 随着计算机技术的迅速发展，汉语需要接纳大量外来术语；
2. 没有权威机构及时检测并给以规范的命名，导致：
   - 蹩脚的翻译，如menu；
   - 多种翻译并存，如Hash Table；
3. 因此亟需一种有效的自动检测新术语的算法。

实验日程

1. 1-4周
   - 调研前人的研究，设计系统架构；
   - 获取通用语料库，旧术语语料库；
   - 开始定期检测IT新闻网站。
2. 5-12周
   - 处理语料库，获取各种词语列表；
   - 实现系统各模块；
   - 组织各模块，完成新术语自动检测系统。
3. 13-14周
   - 设计调试工具如坐标图、表格等GUI，直观显示实验数据；
   - 调整参数，使用调试工具比对数据，提高系统性能。
4. 15-16周
   - 撰写论文。

实验日程

1. **1-4周**
   - 调研前人的研究，设计系统架构；
   - 获取通用语料库，旧术语语料库；
   - 开始定期检测IT新闻网站。
2. **5-12周**
   - 处理语料库，获取各种词语列表；
   - 实现系统各模块；
   - 组织各模块，完成新术语自动检测系统。
3. **13-14周**
   - 设计调试工具如坐标图、表格等GUI，直观显示实验数据；
   - 调整参数，使用调试工具比对数据，提高系统性能。
4. **15-16周**
   - 撰写论文。

实验日程

1. **1-4周**
   - 调研前人的研究，设计系统架构；
   - 获取通用语料库，旧术语语料库；
   - 开始定期检测IT新闻网站。
2. **5-12周**
   - 处理语料库，获取各种词语列表；
   - 实现系统各模块；
   - 组织各模块，完成新术语自动检测系统。
3. **13-14周**
   - 设计调试工具如坐标图、表格等GUI，直观显示实验数据；
   - 调整参数，使用调试工具比对数据，提高系统性能。
4. **15-16周**
   - 撰写论文。

实验日程

1. **1-4周**
   - 调研前人的研究，设计系统架构；
   - 获取通用语料库，旧术语语料库；
   - 开始定期检测IT新闻网站。
2. **5-12周**
   - 处理语料库，获取各种词语列表；
   - 实现系统各模块；
   - 组织各模块，完成新术语自动检测系统。
3. **13-14周**
   - 设计调试工具如坐标图、表格等GUI，直观显示实验数据；
   - 调整参数，使用调试工具比对数据，提高系统性能。
4. **15-16周**
   - 撰写论文。

实验日程

1. **1-4周**
   - 调研前人的研究，设计系统架构；
   - 获取通用语料库，旧术语语料库；
   - 开始定期检测IT新闻网站。
2. **5-12周**
   - 处理语料库，获取各种词语列表；
   - 实现系统各模块；
   - 组织各模块，完成新术语自动检测系统。
3. **13-14周**
   - 设计调试工具如坐标图、表格等GUI，直观显示实验数据；
   - 调整参数，使用调试工具比对数据，提高系统性能。
4. **15-16周**
   - 撰写论文。

实验日程

1. **1-4周**
   - 调研前人的研究，设计系统架构；
   - 获取通用语料库，旧术语语料库；
   - 开始定期检测IT新闻网站。
2. **5-12周**
   - 处理语料库，获取各种词语列表；
   - 实现系统各模块；
   - 组织各模块，完成新术语自动检测系统。
3. **13-14周**
   - 设计调试工具如坐标图、表格等GUI，直观显示实验数据；
   - 调整参数，使用调试工具比对数据，提高系统性能。
4. **15-16周**
   - 撰写论文。

实验日程

1. **1-4周**
   - 调研前人的研究，设计系统架构；
   - 获取通用语料库，旧术语语料库；
   - 开始定期检测IT新闻网站。
2. **5-12周**
   - 处理语料库，获取各种词语列表；
   - 实现系统各模块；
   - 组织各模块，完成新术语自动检测系统。
3. **13-14周**
   - 设计调试工具如坐标图、表格等GUI，直观显示实验数据；
   - 调整参数，使用调试工具比对数据，提高系统性能。
4. **15-16周**
   - 撰写论文。

实验日程

1. **1-4周**
   - 调研前人的研究，设计系统架构；
   - 获取通用语料库，旧术语语料库；
   - 开始定期检测IT新闻网站。
2. **5-12周**
   - 处理语料库，获取各种词语列表；
   - 实现系统各模块；
   - 组织各模块，完成新术语自动检测系统。
3. **13-14周**
   - 设计调试工具如坐标图、表格等GUI，直观显示实验数据；
   - 调整参数，使用调试工具比对数据，提高系统性能。
4. **15-16周**
   - 撰写论文。

实验日程

1. **1-4周**
   - 调研前人的研究，设计系统架构；
   - 获取通用语料库，旧术语语料库；
   - 开始定期检测IT新闻网站。
2. **5-12周**
   - 处理语料库，获取各种词语列表；
   - 实现系统各模块；
   - 组织各模块，完成新术语自动检测系统。
3. **13-14周**
   - 设计调试工具如坐标图、表格等GUI，直观显示实验数据；
   - 调整参数，使用调试工具比对数据，提高系统性能。
4. **15-16周**
   - 撰写论文。

实验日程

1. **1-4周**
   - 调研前人的研究，设计系统架构；
   - 获取通用语料库，旧术语语料库；
   - 开始定期检测IT新闻网站。
2. **5-12周**
   - 处理语料库，获取各种词语列表；
   - 实现系统各模块；
   - 组织各模块，完成新术语自动检测系统。
3. **13-14周**
   - 设计调试工具如坐标图、表格等GUI，直观显示实验数据；
   - 调整参数，使用调试工具比对数据，提高系统性能。
4. **15-16周**
   - 撰写论文。

实验日程

1. **1-4周**
   - 调研前人的研究，设计系统架构；
   - 获取通用语料库，旧术语语料库；
   - 开始定期检测IT新闻网站。
2. **5-12周**
   - 处理语料库，获取各种词语列表；
   - 实现系统各模块；
   - 组织各模块，完成新术语自动检测系统。
3. **13-14周**
   - 设计调试工具如坐标图、表格等GUI，直观显示实验数据；
   - 调整参数，使用调试工具比对数据，提高系统性能。
4. **15-16周**
   - 撰写论文。

实验日程

1. **1-4周**
   - 调研前人的研究，设计系统架构；
   - 获取通用语料库，旧术语语料库；
   - 开始定期检测IT新闻网站。
2. **5-12周**
   - 处理语料库，获取各种词语列表；
   - 实现系统各模块；
   - 组织各模块，完成新术语自动检测系统。
3. **13-14周**
   - 设计调试工具如坐标图、表格等GUI，直观显示实验数据；
   - 调整参数，使用调试工具比对数据，提高系统性能。
4. **15-16周**
   - 撰写论文。

实验日程

1. **1-4周**
   - 调研前人的研究，设计系统架构；
   - 获取通用语料库，旧术语语料库；
   - 开始定期检测IT新闻网站。
2. **5-12周**
   - 处理语料库，获取各种词语列表；
   - 实现系统各模块；
   - 组织各模块，完成新术语自动检测系统。
3. **13-14周**
   - 设计调试工具如坐标图、表格等GUI，直观显示实验数据；
   - 调整参数，使用调试工具比对数据，提高系统性能。
4. **15-16周**
   - 撰写论文。

实验日程

1. **1-4周**
   - 调研前人的研究，设计系统架构；
   - 获取通用语料库，旧术语语料库；
   - 开始定期检测IT新闻网站。
2. **5-12周**
   - 处理语料库，获取各种词语列表；
   - 实现系统各模块；
   - 组织各模块，完成新术语自动检测系统。
3. **13-14周**
   - 设计调试工具如坐标图、表格等GUI，直观显示实验数据；
   - 调整参数，使用调试工具比对数据，提高系统性能。
4. **15-16周**
   - 撰写论文。

前人的研究

- Information Extraction
- Key-phrase Extraction
- 基本方法
  - Dictionary approach
  - Linguistic approach
  - Statistical approach

前人的研究

- Information Extraction
- Key-phrase Extraction
- 基本方法
  - Dictionary approach
    - 人工生成
  - Linguistic approach
    - 根据句法、语义和规则实现
  - Statistical approach
    - 通过计算文档中的N元语法(N-gram)实现
    - TF*IDF

前人的研究

- Information Extraction
- Key-phrase Extraction
- 基本方法
  - Dictionary approach
    - 人工生成
  - Linguistic approach
    - 根据句法，语义和规则实现
  - Statistical approach
    - 通过计算互信息或N元语法(N-gram)实现
    - TF*IDF

前人的研究

- Information Extraction
- Key-phrase Extraction
- 基本方法
  - Dictionary approach
    - 人工生成
  - Linguistic approach
    - 根据句法，语义和规则实现
  - Statistical approach
    - 通过计算互信息或N元语法(N-gram)实现
    - TF*IDF

前人的研究

- Information Extraction
- Key-phrase Extraction
- 基本方法
  - Dictionary approach
    - 人工生成
  - Linguistic approach
    - 根据句法，语义和规则实现
  - Statistical approach
    - 通过计算互信息或N元语法(N-gram)实现
    - TF*IDF

前人的研究

- Information Extraction
- Key-phrase Extraction
- 基本方法
  - Dictionary approach
    - 人工生成
  - Linguistic approach
    - 根据句法，语义和规则实现
  - Statistical approach
    - 通过计算互信息或N元语法(N-gram)实现
    - TF*IDF

前人的研究

- Information Extraction
- Key-phrase Extraction
- 基本方法
  - Dictionary approach
    - 人工生成
  - Linguistic approach
    - 根据句法，语义和规则实现
  - Statistical approach
    - 通过计算互信息或N元语法(N-gram)实现
    - TF*IDF

前人的研究

- Information Extraction
- Key-phrase Extraction
- 基本方法
  - Dictionary approach
    - 人工生成
  - Linguistic approach
    - 根据句法，语义和规则实现
  - Statistical approach
    - 通过计算互信息或N元语法(N-gram)实现
    - TF*IDF

前人的研究

- Information Extraction
- Key-phrase Extraction
- 基本方法
  - Dictionary approach
    - 人工生成
  - Linguistic approach
    - 根据句法，语义和规则实现
  - Statistical approach
    - 通过计算互信息或N元语法(N-gram)实现
    - TF*IDF

前人的研究

- Information Extraction
- Key-phrase Extraction
- 基本方法
  - Dictionary approach
    - 人工生成
  - Linguistic approach
    - 根据句法，语义和规则实现
  - Statistical approach
    - 通过计算互信息或N元语法(N-gram)实现
    - TF*IDF

若干概念

若干概念

- 新术语
  - 某学科中出现的新专门词语；
  - 该算法将主要检测在计算机领域中新出现的，并在大众生活中流行的新术语。
- 通用语料库
  - 具有一般意义、与学科领域无关的语料库；
  - 用于剔除新闻文本中与IT领域无关的部分。
- 计算机领域背景语料库
  - 某时间点之前已经出现的IT术语语料库；
  - 用于剔除新闻文本中与IT领域相关，但在过去已经出现的术语。

若干概念

- 新术语
  - 某学科中出现的新专门词语；
  - 该算法将主要检测在计算机领域中新出现的，并在大众生活中流行的新术语。
- 通用语料库
  - 具有一般意义、与学科领域无关的语料库；
  - 用于剔除新闻文本中与IT领域无关的部分。
- 计算机领域背景语料库
  - 某时间点之前已经出现的IT术语语料库；
  - 用于剔除新闻文本中与IT领域相关，但在过去已经出现的术语。

若干概念

- 新术语
  - 某学科中出现的新专门词语；
  - 该算法将主要检测在计算机领域中新出现的，并在大众生活中流行的新术语。
- 通用语料库
  - 具有一般意义、与学科领域无关的语料库；
  - 用于剔除新闻文本中与IT领域无关的部分。
- 计算机领域背景语料库
  - 某时间点之前已经出现的IT术语语料库；
  - 用于剔除新闻文本中与IT领域相关，但在过去已经出现的术语。

若干概念

- 新术语
  - 某学科中出现的新专门词语；
  - 该算法将主要检测在计算机领域中新出现的，并在大众生活中流行的新术语。
- 通用语料库
  - 具有一般意义、与学科领域无关的语料库；
  - 用于剔除新闻文本中与IT领域无关的部分。
- 计算机领域背景语料库
  - 某时间点之前已经出现的IT术语语料库；
  - 用于剔除新闻文本中与IT领域相关，但在过去已经出现的术语。

若干概念

- 新术语
  - 某学科中出现的新专门词语；
  - 该算法将主要检测在计算机领域中新出现的，并在大众生活中流行的新术语。
- 通用语料库
  - 具有一般意义、与学科领域无关的语料库；
  - 用于剔除新闻文本中与IT领域无关的部分。
- 计算机领域背景语料库
  - 某时间点之前已经出现的IT术语语料库；
  - 用于剔除新闻文本中与IT领域相关，但在过去已经出现的术语。

若干概念

- 新术语
  - 某学科中出现的新专门词语；
  - 该算法将主要检测在计算机领域中新出现的，并在大众生活中流行的新术语。
- 通用语料库
  - 具有一般意义、与学科领域无关的语料库；
  - 用于剔除新闻文本中与IT领域无关的部分。
- 计算机领域背景语料库
  - 某时间点之前已经出现的IT术语语料库；
  - 用于剔除新闻文本中与IT领域相关，但在过去已经出现的术语。

若干概念

- 新术语
  - 某学科中出现的新专门词语；
  - 该算法将主要检测在计算机领域中新出现的，并在大众生活中流行的新术语。
- 通用语料库
  - 具有一般意义、与学科领域无关的语料库；
  - 用于剔除新闻文本中与IT领域无关的部分。
- 计算机领域背景语料库
  - 某时间点之前已经出现的IT术语语料库；
  - 用于剔除新闻文本中与IT领域相关，但在过去已经出现的术语。

若干概念

- 新术语
  - 某学科中出现的新专门词语；
  - 该算法将主要检测在计算机领域中新出现的，并在大众生活中流行的新术语。
- 通用语料库
  - 具有一般意义、与学科领域无关的语料库；
  - 用于剔除新闻文本中与IT领域无关的部分。
- 计算机领域背景语料库
  - 某时间点之前已经出现的IT术语语料库；
  - 用于剔除新闻文本中与IT领域相关，但在过去已经出现的术语。

若干概念

- 新术语
  - 某学科中出现的新专门词语；
  - 该算法将主要检测在计算机领域中新出现的，并在大众生活中流行的新术语。
- 通用语料库
  - 具有一般意义、与学科领域无关的语料库；
  - 用于剔除新闻文本中与IT领域无关的部分。
- 计算机领域背景语料库
  - 某时间点之前已经出现的IT术语语料库；
  - 用于剔除新闻文本中与IT领域相关，但在过去已经出现的术语。

算法思想

新术语特征

1. 自某一时间点以来首次出现；
2. 被广泛地应用，而非昙花一现。

核心思想

1. 通过语料库的比对，找到“自某一时间点”以来在计算机类语料库中新出现的词语，即候选新术语；
2. 通过考查候选新术语在时间上的频度曲线，找到其中“被广泛地应用，而非昙花一现”的词语，确定为新术语。

算法思想

新术语特征

1. 自某一时间点以来首次出现；
2. 被广泛地应用，而非昙花一现。

核心思想

1. 通过语料库的比对，找到“自某一时间点”以来在计算机类语料库中新出现的词语，即候选新术语；
2. 通过考查候选新术语在时间上的频度曲线，找到其中“被广泛地应用，而非昙花一现”的词语，确定为新术语。

算法思想

新术语特征

1. 自某一时间点以来首次出现；
2. 被广泛地应用，而非昙花一现。

核心思想

1. 通过语料库的比对，找到“自某一时间点”以来在计算机类语料库中新出现的词语，即候选新术语；
2. 通过考查候选新术语在时间上的频度曲线，找到其中“被广泛地应用，而非昙花一现”的词语，确定为新术语。

算法思想

新术语特征

1. 自某一时间点以来首次出现；
2. 被广泛地应用，而非昙花一现。

核心思想

1. 通过语料库的比对，找到“自某一时间点”以来在计算机类语料库中新出现的词语，即候选新术语；
2. 通过考查候选新术语在时间上的频度曲线，找到其中“被广泛地应用，而非昙花一现”的词语，确定为新术语。

算法思想

新术语特征

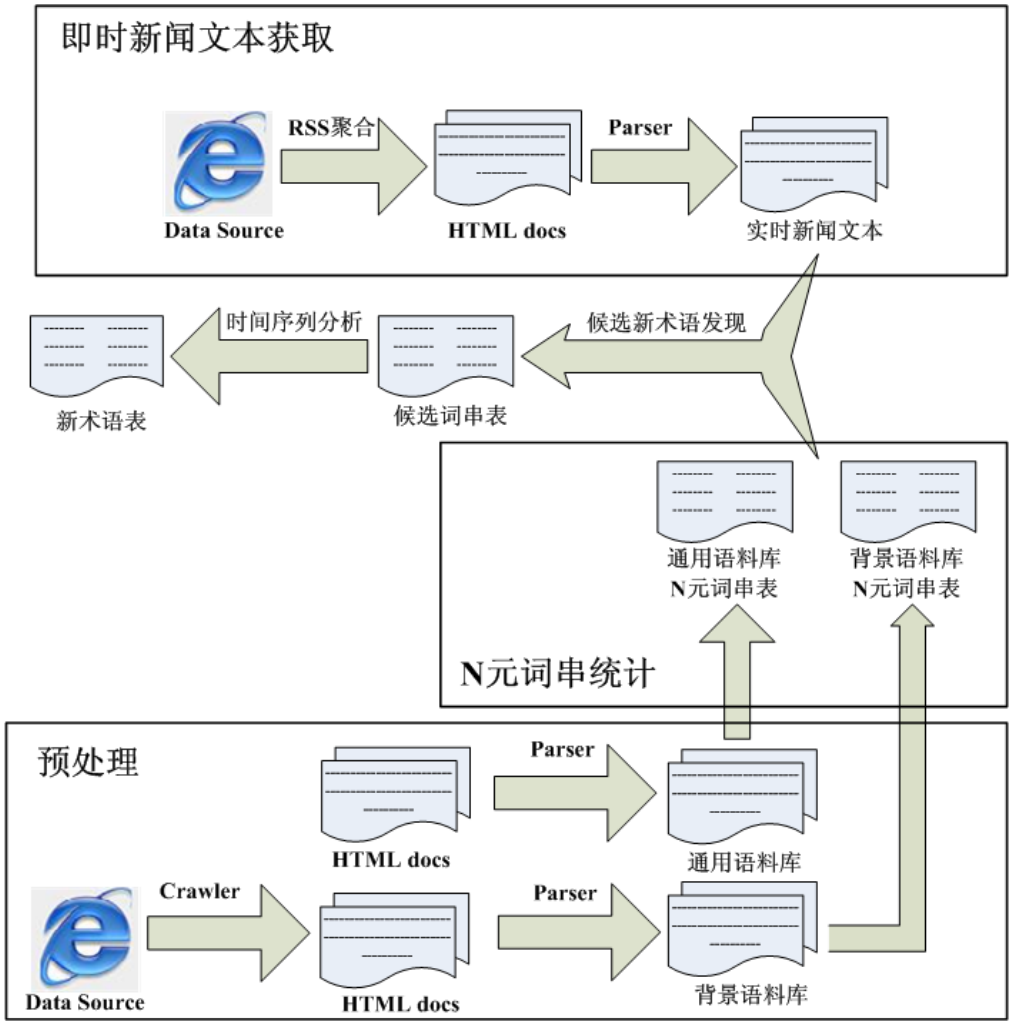
1. 自某一时间点以来首次出现；
2. 被广泛地应用，而非昙花一现。

核心思想

1. 通过语料库的比对，找到“自某一时间点”以来在计算机类语料库中新出现的词语，即候选新术语；
2. 通过考查候选新术语在时间上的频度曲线，找到其中“被广泛地应用，而非昙花一现”的词语，确定为新术语。

算法设计流程图

算法设计流程图


即时新闻文本获取
RSS聚合
Parser
Data Source
HTML docs
实时新闻文本
时间序列分析
候选新术语发现
新术语表
候选词串表
通用语料库
N元词串表
背景语料库
N元词串表
N元词串统计
预处理
Parser
HTML docs
通用语料库
Crawler
Parser
Data Source
HTML docs
背景语料库

算法说明

1. 该算法主要是检测在大众生活中流行的计算机领域新术语，而非在学术论文中出现的，一般为专业人士使用的术语。
2. 该算法只是对新术语的预警，它只是为用户提供一个可能是新术语的词语列表，至于甄别哪些是真正的新术语，仍然需要人工来确定。
3. 只需使用相应的语料库，该算法适用于任何领域的新术语的检测。

算法说明

1. 该算法主要是检测在大众生活中流行的计算机领域新术语，而非在学术论文中出现的，一般为专业人士使用的术语。
2. 该算法只是对新术语的预警，它只是为用户提供一个可能是新术语的词语列表，至于甄别哪些是真正的新术语，仍然需要人工来确定。
3. 只需使用相应的语料库，该算法适用于任何领域的新术语的检测。

算法说明

1. 该算法主要是检测在大众生活中流行的计算机领域新术语，而非在学术论文中出现的，一般为专业人士使用的术语。
2. 该算法只是对新术语的预警，它只是为用户提供一个可能是新术语的词语列表，至于甄别哪些是真正的新术语，仍然需要人工来确定。
3. 只需使用相应的语料库，该算法适用于任何领域的新术语的检测。

算法说明

1. 该算法主要是检测在大众生活中流行的计算机领域新术语，而非在学术论文中出现的，一般为专业人士使用的术语。
2. 该算法只是对新术语的预警，它只是为用户提供一个可能是新术语的词语列表，至于甄别哪些是真正的新术语，仍然需要人工来确定。
3. 只需使用相应的语料库，该算法适用于任何领域的新术语的检测。

预处理-简介

- 主要功能
  - 取《英文世界名著1000部》作为通用语料库；
  - 抓取五大英文门户网站网页数据，作为语料库。
- 实现方法
  - 使用Offline Explorer抓取网站数据；
  - 使用HTML Parser提取网页文本内容。

预处理-简介

- 主要功能
  - 取《英文世界名著1000部》作为通用语料库；
  - 抓取五大英文IT网站网页数据，作为背景语料库。
- 实现方法
  - 使用Offline Explorer抓取网站数据；
  - 使用HTML Parser提取网页文本内容。

预处理-简介

- 主要功能
  - 取《英文世界名著1000部》作为通用语料库；
  - 抓取五大英文IT网站网页数据，作为背景语料库。
- 实现方法
  - 使用Offline Explorer抓取网站数据；
  - 使用HTML Parser提取网页文本内容。

预处理-简介

- 主要功能
  - 取《英文世界名著1000部》作为通用语料库；
  - 抓取五大英文IT网站网页数据，作为背景语料库。
- 实现方法
  - 使用Offline Explorer抓取网站数据；
  - 使用HTML Parser提取网页文本内容。

预处理-简介

- 主要功能
  - 取《英文世界名著1000部》作为通用语料库；
  - 抓取五大英文IT网站网页数据，作为背景语料库。
- 实现方法
  - 使用Offline Explorer抓取网站数据；
  - 使用HTML Parser提取网页文本内容。

预处理-简介

- 主要功能
  - 取《英文世界名著1000部》作为通用语料库；
  - 抓取五大英文IT网站网页数据，作为背景语料库。
- 实现方法
  - 使用Offline Explorer抓取网站数据；
  - 使用HTML Parser提取网页文本内容。

预处理-简介

- 主要功能
  - 取《英文世界名著1000部》作为通用语料库；
  - 抓取五大英文IT网站网页数据，作为背景语料库。
- 实现方法
  - 使用Offline Explorer抓取网站数据；
  - 使用HTML Parser提取网页文本内容。

预处理-相关数据

- 五大英文IT网站

Table: 五大英文IT新闻网站列表

网站名	URL	Alexa排名
CNet News	www.news.com	54/1
ZDNet	news.zdnet.com	658/2
Slashdot	slashdot.org	287/3
PC World	www.pcworld.com	571/1
PC Magazine	www.pcmag.com	571/2

- 总共获取13.5GB网页文件，经解析得到1.37GB纯文本。

预处理-相关数据

- 五大英文IT网站

Table: 五大英文IT新闻网站列表

网站名	URL	Alexa排名
CNet News	www.news.com	54/1
ZDNet	news.zdnet.com	658/2
Slashdot	slashdot.org	287/3
PC World	www.pcworld.com	571/1
PC Magazine	www.pcmag.com	571/2

- 总共获取13.5GB网页文件，经解析得到1.37GB纯文本。

预处理-相关数据

- 五大英文IT网站

Table: 五大英文IT新闻网站列表

网站名	URL	Alexa排名
CNet News	www.news.com	54/1
ZDNet	news.zdnet.com	658/2
Slashdot	slashdot.org	287/3
PC World	www.pcworld.com	571/1
PC Magazine	www.pcmag.com	571/2

- 总共获取13.5GB网页文件，经解析得到1.37GB纯文本。

即时新闻文本获取

- 主要功能
  - 通过RSS新闻聚合（一种XML应用）获取最近发布的新闻列表。
  - 使用HTML Parser提取新闻网页文本内容。
- 主要技术
  - XML文件解析。
  - HTML Parser提取网页文本内容。
- 每日可以获得150~400篇新闻，大小约1~4MB。

即时新闻文本获取

- 主要功能
  - 通过RSS新闻聚合（一种XML应用）获取每日发布的新闻网页；
  - 使用HTML Parser提取新闻网页文本内容。
- 主要技术
  - XML文件解析；
  - HTML Parser提取网页文本内容。
- 每日可以获得150~400篇新闻，大小约1~4MB。

即时新闻文本获取

- 主要功能
  - 通过RSS新闻聚合（一种XML应用）获取每日发布的新闻网页；
  - 使用HTML Parser提取新闻网页文本内容。
- 主要技术
  - XML文件解析；
  - HTML Parser提取网页文本内容。
- 每日可以获得150~400篇新闻，大小约1~4MB。

即时新闻文本获取

- 主要功能
  - 通过RSS新闻聚合（一种XML应用）获取每日发布的新闻网页；
  - 使用HTML Parser提取新闻网页文本内容。
- 主要技术
  - XML文件解析；
  - HTML Parser提取网页文本内容。
- 每日可以获得150~400篇新闻，大小约1~4MB。

即时新闻文本获取

- 主要功能
  - 通过RSS新闻聚合（一种XML应用）获取每日发布的新闻网页；
  - 使用HTML Parser提取新闻网页文本内容。
- 主要技术
  - XML文件解析；
  - HTML Parser提取网页文本内容。
- 每日可以获得150~400篇新闻，大小约1~4MB。

即时新闻文本获取

- 主要功能
  - 通过RSS新闻聚合（一种XML应用）获取每日发布的新闻网页；
  - 使用HTML Parser提取新闻网页文本内容。
- 主要技术
  - XML文件解析；
  - HTML Parser提取网页文本内容。
- 每日可以获得150~400篇新闻，大小约1~4MB。

即时新闻文本获取

- 主要功能
  - 通过RSS新闻聚合（一种XML应用）获取每日发布的新闻网页；
  - 使用HTML Parser提取新闻网页文本内容。
- 主要技术
  - XML文件解析；
  - HTML Parser提取网页文本内容。
- 每日可以获得150~400篇新闻，大小约1~4MB。

即时新闻文本获取

- 主要功能
  - 通过RSS新闻聚合（一种XML应用）获取每日发布的新闻网页；
  - 使用HTML Parser提取新闻网页文本内容。
- 主要技术
  - XML文件解析；
  - HTML Parser提取网页文本内容。
- 每日可以获得150~400篇新闻，大小约1~4MB。

N元词串统计-简介

N元词串统计-简介

- 主要功能
  - 统计语料库中相邻两单词的TF和DF并排序，形成二元词串表；
  - 统计语料库中单词的TF和DF并排序，形成单词表。
- 相关技术
  - 引入Frequent Word List以减小统计规模；
  - 词串或单词及其对应频度的存储结构采用Hash Map；
  - 需要对文本进行分句，采用正则表达式完成。

N元词串统计-简介

- 主要功能
  - 统计语料库中相邻两单词的TF和DF并排序，形成二元词串表；
  - 统计语料库中单词的TF和DF并排序，形成单词表。
- 相关技术
  - 引入Frequent Word List以减小统计规模；
  - 词串或单词及其对应频度的存储结构采用Hash Map；
  - 需要对文本进行分句，采用正则表达式完成。

N元词串统计-简介

- 主要功能
  - 统计语料库中相邻两单词的TF和DF并排序，形成二元词串表；
  - 统计语料库中单词的TF和DF并排序，形成单词表。
- 相关技术
  - 引入Frequent Word List以减小统计规模；
  - 词串或单词及其对应频度的存储结构采用Hash Map；
  - 需要对文本进行分句，采用正则表达式完成。

N元词串统计-简介

- 主要功能
  - 统计语料库中相邻两单词的TF和DF并排序，形成二元词串表；
  - 统计语料库中单词的TF和DF并排序，形成单词表。
- 相关技术
  - 引入Frequent Word List以减小统计规模；
  - 词串或单词及其对应频度的存储结构采用Hash Map；
  - 需要对文本进行分句，采用正则表达式完成。

N元词串统计-简介

- 主要功能
  - 统计语料库中相邻两单词的TF和DF并排序，形成二元词串表；
  - 统计语料库中单词的TF和DF并排序，形成单词表。
- 相关技术
  - 引入Frequent Word List以减小统计规模；
  - 词串或单词及其对应频度的存储结构采用Hash Map；
  - 需要对文本进行分句，采用正则表达式完成。

N元词串统计-简介

- 主要功能
  - 统计语料库中相邻两单词的TF和DF并排序，形成二元词串表；
  - 统计语料库中单词的TF和DF并排序，形成单词表。
- 相关技术
  - 引入Frequent Word List以减小统计规模；
  - 词串或单词及其对应频度的存储结构采用Hash Map；
  - 需要对文本进行分句，采用正则表达式完成。

N元词串统计-简介

- 主要功能
  - 统计语料库中相邻两单词的TF和DF并排序，形成二元词串表；
  - 统计语料库中单词的TF和DF并排序，形成单词表。
- 相关技术
  - 引入Frequent Word List以减小统计规模；
  - 词串或单词及其对应频度的存储结构采用Hash Map；
  - 需要对文本进行分句，采用正则表达式完成。

词语统计模块-相关数据

Table: 通用语料库和旧术语语料库统计表

文件名	词条数目（万条）
通用语料库单词表	29
背景语料库单词表	53
通用语料库二元词串表	384
背景语料库二元词串表	291

词语统计模块-相关数据

Table: 通用语料库和旧术语语料库统计表

文件名	词条数目（万条）
通用语料库单词表	29
背景语料库单词表	53
通用语料库二元词串表	384
背景语料库二元词串表	291

候选新术语发现

候选新术语发现

- 主要功能：从新闻文本中提取、统计和存储候选新术语；
- 主要包括：
  - 孤岛词串发现算法—在新闻文本中发现候选新术语；
  - 对发现的候选新术语进行统计和存储。

候选新术语发现

- 主要功能：从新闻文本中提取、统计和存储候选新术语；
- 主要包括：
  - 孤岛词串发现算法—在新闻文本中发现候选新术语；
  - 对发现的候选新术语进行统计和存储。

候选新术语发现

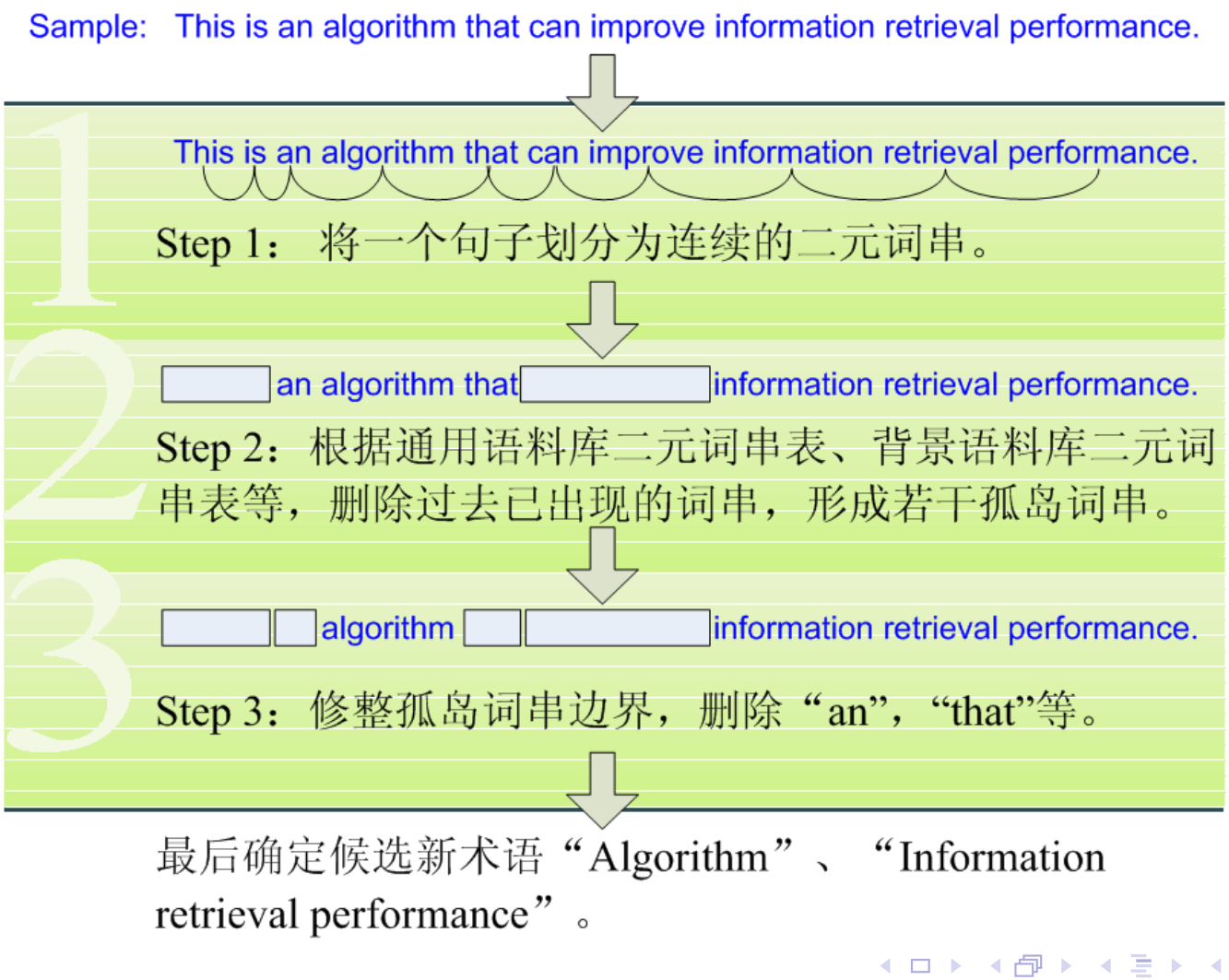
- 主要功能：从新闻文本中提取、统计和存储候选新术语；
- 主要包括：
  - 孤岛词串发现算法—在新闻文本中发现候选新术语；
  - 对发现的候选新术语进行统计和存储。

候选新术语发现

- 主要功能：从新闻文本中提取、统计和存储候选新术语；
- 主要包括：
  - 孤岛词串发现算法—在新闻文本中发现候选新术语；
  - 对发现的候选新术语进行统计和存储。

孤岛词串发现算法-流程图

孤岛词串发现算法-流程图



孤岛词串发现算法-算法介绍

第一步：划分二元词串

- 将句子划分为二元词串。对一个句子的单词序列，表示如下：

$$w_1 \quad w_2 \quad w_3 \quad \ldots \quad w_{n-1} \quad w_n.$$

可以得到二元词串：

$$B_1, B_2, B_3, \ldots, B_{n-1}.$$

- 任何相邻的两个二元词串都在一个单词上重叠。用数学公式表示就是：

$$w_i, w_{i+1} \in B_i \quad \forall i \in [1, n-1].$$

孤岛词串发现算法-算法介绍

第一步：划分二元词串

- 将句子划分为二元词串。对一个句子的单词序列，表示如下：

$$w_1 \quad w_2 \quad w_3 \quad \ldots \quad w_{n-1} \quad w_n.$$

可以得到二元词串：

$$B_1, B_2, B_3, \ldots, B_{n-1}.$$

- 任何相邻的两个二元词串都在一个单词上重叠。用数学公式表示就是：

$$w_i, w_{i+1} \in B_i \qquad \forall i \in [1, n-1].$$

孤岛词串发现算法-算法介绍

第一步：划分二元词串

- 将句子划分为二元词串。对一个句子的单词序列，表示如下：

$$w_1 \quad w_2 \quad w_3 \quad \ldots \quad w_{n-1} \quad w_n.$$

可以得到二元词串：

$$B_1, B_2, B_3, \ldots, B_{n-1}.$$

- 任何相邻的两个二元词串都在一个单词上重叠。用数学公式表示就是：

$$w_i, w_{i+1} \in B_i \qquad \forall i \in [1, n-1].$$

孤岛词串发现算法-算法介绍

第二步：初步提取孤岛词串

1. 为句子的每个单词打标签：Appeared/Unappered；
2. 将单词序列中标签为 Unappeared 的连续的子序列提取出来，这就是孤岛词串。

孤岛词串发现算法-算法介绍

第二步：初步提取孤岛词串

1. 为句子的每个单词打标签：Appeared/Unappered；
   - 如果一个二元词串的两个单词都是stop word，两个单词应该被标记为Appeared；
   - 一个单词所在的两个二元词串都已经在语料库中以很高的频率出现过，该单词被标记为Appeared；
   - 比较特殊的情况，如果B_1或B_{n-1}已经在语料库中以很高的频率出现过，简单地将w_1或w_n标记为Appeared。
2. 将单词序列中标签为 Unappeared 的连续的子序列提取出来，这就是孤岛词串。

孤岛词串发现算法-算法介绍

第二步：初步提取孤岛词串

1. 为句子的每个单词打标签：Appeared/Unappered；
   - 如果一个二元词串的两个单词都是stop word，两个单词应该被标记为Appeared；
   - 一个单词所在的两个二元词串都已经在语料库中以很高的频率出现过，该单词被标记为Appeared；
   - 比较特殊的情况，如果B_1或B_{n-1}已经在语料库中以很高的频率出现过，简单地将w_1或w_n标记为Appeared。
2. 将单词序列中标签为 Unappeared 的连续的子序列提取出来，这就是孤岛词串。

孤岛词串发现算法-算法介绍

第二步：初步提取孤岛词串

1. 为句子的每个单词打标签：Appeared/Unappered；
   - 如果一个二元词串的两个单词都是stop word，两个单词应该被标记为Appeared；
   - 一个单词所在的两个二元词串都已经在语料库中以很高的频率出现过，该单词被标记为Appeared；
   - 比较特殊的情况，如果B_1或B_{n-1}已经在语料库中以很高的频率出现过，简单地将w_1或w_n标记为Appeared。
2. 将单词序列中标签为 Unappeared 的连续的子序列提取出来，这就是孤岛词串。

孤岛词串发现算法-算法介绍

第二步：初步提取孤岛词串

1. 为句子的每个单词打标签：Appeared/Unappered；
   - 如果一个二元词串的两个单词都是stop word，两个单词应该被标记为Appeared；
   - 一个单词所在的两个二元词串都已经在语料库中以很高的频率出现过，该单词被标记为Appeared；
   - 比较特殊的情况，如果B_1或B_{n-1}已经在语料库中以很高的频率出现过，简单地将w_1或w_n标记为Appeared。
2. 将单词序列中标签为 Unappeared 的连续的子序列提取出来，这就是孤岛词串。

孤岛词串发现算法-算法介绍

第二步：初步提取孤岛词串

1. 为句子的每个单词打标签：Appeared/Unappered；
   - 如果一个二元词串的两个单词都是stop word，两个单词应该被标记为Appeared；
   - 一个单词所在的两个二元词串都已经在语料库中以很高的频率出现过，该单词被标记为Appeared；
   - 比较特殊的情况，如果B_1或B_{n-1}已经在语料库中以很高的频率出现过，简单地将w_1或w_n标记为Appeared。
2. 将单词序列中标签为 Unappeared 的连续的子序列提取出来，这就是孤岛词串。

孤岛词串发现算法-算法介绍

第三步：修整孤岛词串边界

1. 对孤岛词串，如果它是一个单词，查看是否在之前很高频度出现，如果是，则丢弃，否则进入3；如果它是一个二元词串，查看其是否在通用语料库二元词串表和背景语料库二元词串表的高频部分或Frequent Word List的二元词串组合中出现，如果是，则丢弃，否则进入步骤2；如果它是一个含有超过三个单词的词串，进入步骤2。
2. 查看词串的首尾单词是否出现在Empty Word List中，如果是则将该单词从这个词串中删除，进入3；
3. 如果该词串较1时有修改，则递归重新进入步骤1，对修整后的新词串进行新的一轮修整；否则进入步骤4。
4. 查看词串是否出现在Current IT Term List中，如果是，则丢弃；否则，输出为新术语候选词串。

孤岛词串发现算法-算法介绍

第三步：修整孤岛词串边界

1. 对孤岛词串，如果它是一个单词，查看是否在之前很高频度出现，如果是，则丢弃，否则进入3；如果它是一个二元词串，查看其是否在通用语料库二元词串表和背景语料库二元词串表的高频部分或Frequent Word List的二元词串组合中出现，如果是，则丢弃，否则进入步骤2；如果它是一个含有超过三个单词的词串，进入步骤2。
2. 查看词串的首尾单词是否出现在Empty Word List中，如果是则将该单词从这个词串中删除，进入3；
3. 如果该词串较1时有修改，则递归重新进入步骤1，对修整后的新词串进行新的一轮修整；否则进入步骤4。
4. 查看词串是否出现在Current IT Term List中，如果是，则丢弃；否则，输出为新术语候选词串。

孤岛词串发现算法-算法介绍

第三步：修整孤岛词串边界

1. 对孤岛词串，如果它是一个单词，查看是否在之前很高频度出现，如果是，则丢弃，否则进入3；如果它是一个二元词串，查看其是否在通用语料库二元词串表和背景语料库二元词串表的高频部分或Frequent Word List的二元词串组合中出现，如果是，则丢弃，否则进入步骤2；如果它是一个含有超过三个单词的词串，进入步骤2。
2. 查看词串的首尾单词是否出现在Empty Word List中，如果是则将该单词从这个词串中删除，进入3；
3. 如果该词串较1时有修改，则递归重新进入步骤1，对修整后的新词串进行新的一轮修整；否则进入步骤4。
4. 查看词串是否出现在Current IT Term List中，如果是，则丢弃；否则，输出为新术语候选词串。

孤岛词串发现算法-算法介绍

第三步：修整孤岛词串边界

1. 对孤岛词串，如果它是一个单词，查看是否在之前很高频度出现，如果是，则丢弃，否则进入3；如果它是一个二元词串，查看其是否在通用语料库二元词串表和背景语料库二元词串表的高频部分或Frequent Word List的二元词串组合中出现，如果是，则丢弃，否则进入步骤2；如果它是一个含有超过三个单词的词串，进入步骤2。
2. 查看词串的首尾单词是否出现在Empty Word List中，如果是则将该单词从这个词串中删除，进入3；
3. 如果该词串较1时有修改，则递归重新进入步骤1，对修整后的新词串进行新的一轮修整；否则进入步骤4。
4. 查看词串是否出现在Current IT Term List中，如果是，则丢弃；否则，输出为新术语候选词串。

孤岛词串发现算法-算法介绍

第三步：修整孤岛词串边界

1. 对孤岛词串，如果它是一个单词，查看是否在之前很高频度出现，如果是，则丢弃，否则进入3；如果它是一个二元词串，查看其是否在通用语料库二元词串表和背景语料库二元词串表的高频部分或Frequent Word List的二元词串组合中出现，如果是，则丢弃，否则进入步骤2；如果它是一个含有超过三个单词的词串，进入步骤2。
2. 查看词串的首尾单词是否出现在Empty Word List中，如果是则将该单词从这个词串中删除，进入3；
3. 如果该词串较1时有修改，则递归重新进入步骤1，对修整后的新词串进行新的一轮修整；否则进入步骤4。
4. 查看词串是否出现在Current IT Term List中，如果是，则丢弃；否则，输出为新术语候选词串。

时间序列分析-简介

时间序列分析-简介

- 主要功能：
  - 对每一个候选新术语，根据其在一段时间内的频度曲线，通过评价函数为其打分；
  - 当分数超过某一个阈值的时候，就最终确定该词语为新术语。

时间序列分析-简介

- 主要功能：
  - 对每一个候选新术语，根据其在一段时间内的频度曲线，通过评价函数为其打分；
  - 当分数超过某一个阈值的时候，就最终确定该词语为新术语。

时间序列分析-简介

- 主要功能：
  - 对每一个候选新术语，根据其在一段时间内的频度曲线，通过评价函数为其打分；
  - 当分数超过某一个阈值的时候，就最终确定该词语为新术语。

时间序列分析-算法设计

1. 将某候选新术语的频度数据按照设置的粒度进行聚合；
2. 由评价函数判断候选新术语是否符合新术语要求：
   - 设候选新术语对应的频度聚合数据为 $a_1, a_2, \ldots, a_n$，评价函数 $f(a_{i+1}, a_i)$ 如下：

$$f(a_{i+1}, a_i) = \begin{cases} 1 & \text{if } a_{i+1} > a_i \\ -0.5 & \text{if } a_{i+1} = a_i \\ -1 & \text{if } a_{i+1} < a_i \end{cases}$$

   - 通过以下求和函数得到评价分数：

$$S = \sum_{i=1}^{N} f(a_{i+1}, a_i) \qquad (1)$$

在 $N = 1 \sim n-1$ 过程中，比较 S 和阈值 δ，一旦 $S > \delta$ 则判定该词语为新术语。

时间序列分析-算法设计

1. 将某候选新术语的频度数据按照设置的粒度进行聚合；
2. 由评价函数判断候选新术语是否符合新术语要求：
   1. 设候选新术语对应的频度聚合数据为 $a_1, a_2, \ldots, a_n$，评价函数 $f(a_{i+1}, a_i)$ 如下：

$$f(a_{i+1}, a_i) = \begin{cases} 1 & \text{if } a_{i+1} > a_i \\ -0.5 & \text{if } a_{i+1} = a_i \\ -1 & \text{if } a_{i+1} < a_i \end{cases}$$

   2. 通过以下求和函数得到评价分数：

$$S = \sum_{i=1}^{N} f(a_{i+1}, a_i) \tag{1}$$

在 $N = 1 \rightarrow n-1$ 过程中，比较 S 和阈值 δ，一旦 $S > \delta$ 则判定该词语为新术语。

时间序列分析-算法设计

1. 将某候选新术语的频度数据按照设置的粒度进行聚合；
2. 由评价函数判断候选新术语是否符合新术语要求：
   1. 设候选新术语对应的频度聚合数据为 $a_1, a_2, \ldots, a_n$，评价函数$f(a_{i+1}, a_i)$如下：

$$f(a_{i+1}, a_i) = \begin{cases} 1 & \text{if } a_{i+1} > a_i \\ -0.5 & \text{if } a_{i+1} = a_i \\ -1 & \text{if } a_{i+1} < a_i \end{cases}$$

   2. 通过以下求和函数得到评价分数：

$$S = \sum_{i=1}^{N} f(a_{i+1}, a_i) \tag{1}$$

在 $N = 1 \rightarrow n-1$ 过程中，比较 S 和阈值 δ，一旦 $S > \delta$ 则判定该词语为新术语。

时间序列分析-算法设计

1. 将某候选新术语的频度数据按照设置的粒度进行聚合；
2. 由评价函数判断候选新术语是否符合新术语要求：
   1. 设候选新术语对应的频度聚合数据为 $a_1, a_2, \ldots, a_n$，评价函数$f(a_{i+1}, a_i)$如下：

$$f(a_{i+1}, a_i) = \begin{cases} 1 & \text{if } a_{i+1} > a_i \\ -0.5 & \text{if } a_{i+1} = a_i \\ -1 & \text{if } a_{i+1} < a_i \end{cases}$$

   2. 通过以下求和函数得到评价分数：

$$S = \sum_{i=1}^{N} f(a_{i+1}, a_i) \tag{1}$$

在 $N = 1 \rightarrow n-1$ 过程中，比较 S 和阈值 δ，一旦 $S > \delta$ 则判定该词语为新术语。

时间序列分析-算法设计

1. 将某候选新术语的频度数据按照设置的粒度进行聚合；
2. 由评价函数判断候选新术语是否符合新术语要求：
   1. 设候选新术语对应的频度聚合数据为 $a_1, a_2, \ldots, a_n$，评价函数$f(a_{i+1}, a_i)$如下：

$$f(a_{i+1}, a_i) = \begin{cases} 1 & \text{if } a_{i+1} > a_i \\ -0.5 & \text{if } a_{i+1} = a_i \\ -1 & \text{if } a_{i+1} < a_i \end{cases}$$

   2. 通过以下求和函数得到评价分数：

$$S = \sum_{i=1}^{N} f(a_{i+1}, a_i) \tag{1}$$

在 $N = 1 \rightarrow n-1$ 过程中，比较 S 和阈值 δ，一旦 $S > \delta$ 则判定该词语为新术语。

实验结果-准确率

Table: 算法返回新术语的准确率

日期	人工确定新术语数目	算法返回新术语数目	准确率(%)
2006.06.17	17	126	13.5
2006.06.18	12	71	16.9
2006.06.19	6	35	17.1
2006.06.20	21	177	11.9

说明

人工确定新术语，是指由人工确定算法返回新术语中真正为新术语的数目。其中存在一定主观因素的影响。下同。

实验结果-算法返回新术语的来源

Table: 算法返回新术语来源分类（个数/%）

日期	广告	ID	新闻正文	评论	其它	总计
2006.06.17	58/46.0	8/6.3	26/20.6	7/5.6	27/21.4	126
2006.06.18	48/67.6	4/5.6	9/12.7	4/5.6	6/8.5	71
2006.06.19	14/40.0	2/5.7	10/28.6	1/2.9	7/20.0	35
2006.06.20	89/50.3	2/1.1	40/22.6	0/0.0	47/26.6	177

实验结果-人工确定新术语的来源

Table: 人工确定新术语来源分类（个数/%）

日期	广告	ID	新闻正文	评论	其它	总计
2006.06.17	6/35.3	0/0.0	8/47.1	0/0.0	3/17.6	17
2006.06.18	9/75.0	0/0.0	1/8.3	1/8.3	1/8.3	12
2006.06.19	1/16.7	0/0.0	4/66.7	0/0.0	1/16.7	6
2006.06.20	5/23.8	0/0.0	6/28.6	0/0.0	9/42.9	21

实验结果-各来源新术语的准确率

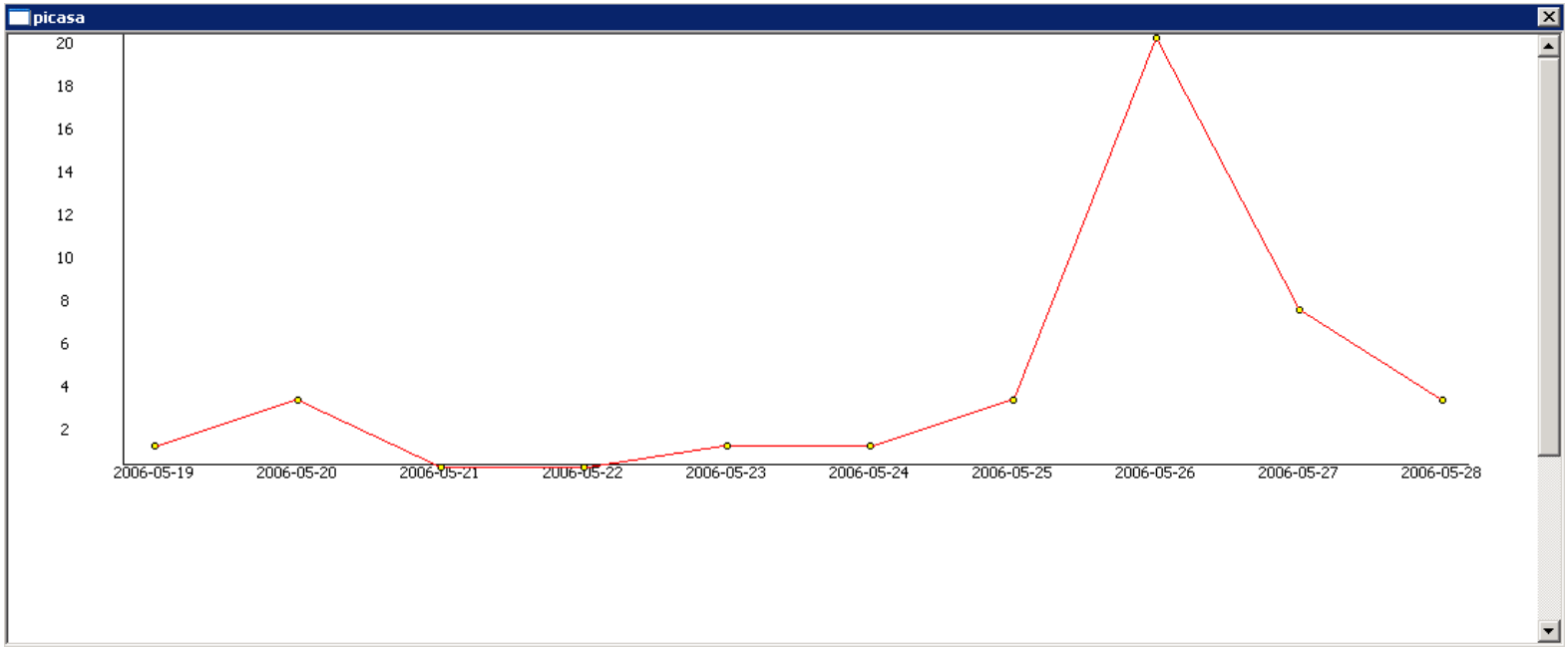
Table: 各来源新术语的准确率（%）

日期	广告	ID	新闻正文	评论	其它
2006.06.17	10.3	0.0	30.8	0.0	11.1
2006.06.18	18.8	0.0	11.1	25.0	16.7
2006.06.19	7.1	0.0	40.0	0.0	14.3
2006.06.20	5.6	0.0	15.0	0.0	19.1

实验结果-新术语示例

- Critical Strategy: ∼ for Managing Information in the 21st Century...
- overkill: Beware of broadband speed ∼.
- Wearable Tech
- googlepages
- Code Name Tracker
- Seamless Mobility: ∼: A Continuity of Experiences Across Domains, Devices and Networks Motorola.
- Vista-capable
- picasa: Google product name.
- two-way Web: ...he is only now realizing his early vision of a ∼ where people can easily work together on the same page...
- IP Migration: The Return on Voice over ∼.

实验结果-新术语“Picasa”频度曲线



说明

Picasa是Google开发的一款图片管理工具，在2006年5月26日其Linux版本发布。

实验结果-新术语“Picasa”出现位置

picasa

Date/Name	TF/Line	DF/Details
2006-05-19	1	1
Read your Reader from anywhere	71	File...
2006-05-20	3	3
Google Music gets closer	73	File...
Google (service name here) isn't that p...	74	File...
Turning the map Green	73	File...
2006-05-23	1	1
Measure Map getting closer#	71	File...
2006-05-24	1	1
Google video ads good# I think so	77	File...
2006-05-25	3	3
Calling International Rescue!	72	File...
Google Calendar RSS problems#	73	File...
Improving Software Usability#	82	File...
2006-05-26	19	5
Dell Installs Google Software at Factory	149	File...
Web 2.0 trademark tailspin	54	File...
Picasa for Linux	0 18 21 22 28 44 75	File...
Google Releases Picasa for Linux	65 68 136 207 208 248 259 269 319	File...
Dude, you got Google	74	File...
2006-05-27	7	4
Gmail to get prices	49 59 79	File...
Google's Picasa for Linux Arrives	8 10	File...
Dell wields its leverage and disrupts Wi...	59	File...
PowerPage wins on appeal	56	File...
2006-05-28	3	1
Google Video mix up	42 53 71	File...

实验分析

实验分析

- 大量的广告词在一段时间内反复出现在新闻网页中，会对数据的可靠性造成一定的影响。但不能简单去除广告了事。
  1. 广告也是新术语的重要来源之一；
  2. 目前提取正文、去除广告的准确率较低；
  3. 广告一般在一个网页中只会出现一次，广告中的词串的统计特点是其TF和DF相等。可以优化评价函数提高对广告新术语的提取。
- 活跃用户的 ID 等也是造成系统准确率较低的重要原因。
  1. 系统具有一定的自适应能力，对于这些活跃ID只会错判一次；
  2. 目前提取正文、去除广告的准确率较低；
  3. 可以根据网页评论的格式，识别哪些词语是用户ID。

实验分析

- 大量的广告词在一段时间内反复出现在新闻网页中，会对数据的可靠性造成一定的影响。但不能简单去除广告了事。
  1. 广告也是新术语的重要来源之一；
  2. 目前提取正文、去除广告的准确率较低；
  3. 广告一般在一个网页中只会出现一次，广告中的词串的统计特点是其TF和DF相等。可以优化评价函数提高对广告新术语的提取。
- 活跃用户的 ID 等也是造成系统准确率较低的重要原因。
  1. 系统具有一定的自适应能力，对于这些活跃ID只会错判一次；
  2. 目前提取正文、去除广告的准确率较低；
  3. 可以根据网页评论的格式，识别哪些词语是用户ID。

实验分析

- 大量的广告词在一段时间内反复出现在新闻网页中，会对数据的可靠性造成一定的影响。但不能简单去除广告了事。
  1. 广告也是新术语的重要来源之一；
  2. 目前提取正文、去除广告的准确率较低；
  3. 广告一般在一个网页中只会出现一次，广告中的词串的统计特点是其TF和DF相等。可以优化评价函数提高对广告新术语的提取。
- 活跃用户的 ID 等也是造成系统准确率较低的重要原因。
  1. 系统具有一定的自适应能力，对于这些活跃ID只会错判一次；
  2. 目前提取正文、去除广告的准确率较低；
  3. 可以根据网页评论的格式，识别哪些词语是用户ID。

实验分析

- 大量的广告词在一段时间内反复出现在新闻网页中，会对数据的可靠性造成一定的影响。但不能简单去除广告了事。
  1. 广告也是新术语的重要来源之一；
  2. 目前提取正文、去除广告的准确率较低；
  3. 广告一般在一个网页中只会出现一次，广告中的词串的统计特点是其TF和DF相等。可以优化评价函数提高对广告新术语的提取。
- 活跃用户的 ID 等也是造成系统准确率较低的重要原因。
  1. 系统具有一定的自适应能力，对于这些活跃ID只会错判一次；
  2. 目前提取正文、去除广告的准确率较低；
  3. 可以根据网页评论的格式，识别哪些词语是用户ID。

实验分析

- 大量的广告词在一段时间内反复出现在新闻网页中，会对数据的可靠性造成一定的影响。但不能简单去除广告了事。
  1. 广告也是新术语的重要来源之一；
  2. 目前提取正文、去除广告的准确率较低；
  3. 广告一般在一个网页中只会出现一次，广告中的词串的统计特点是其TF和DF相等。可以优化评价函数提高对广告新术语的提取。
- 活跃用户的 ID 等也是造成系统准确率较低的重要原因。
  1. 系统具有一定的自适应能力，对于这些活跃ID只会错判一次；
  2. 目前提取正文、去除广告的准确率较低；
  3. 可以根据网页评论的格式，识别哪些词语是用户ID。

实验分析

- 大量的广告词在一段时间内反复出现在新闻网页中，会对数据的可靠性造成一定的影响。但不能简单去除广告了事。
  1. 广告也是新术语的重要来源之一；
  2. 目前提取正文、去除广告的准确率较低；
  3. 广告一般在一个网页中只会出现一次，广告中的词串的统计特点是其TF和DF相等。可以优化评价函数提高对广告新术语的提取。
- 活跃用户的 ID 等也是造成系统准确率较低的重要原因。
  1. 系统具有一定的自适应能力，对于这些活跃ID只会错判一次；
  2. 目前提取正文、去除广告的准确率较低；
  3. 可以根据网页评论的格式，识别哪些词语是用户ID。

实验分析

- 大量的广告词在一段时间内反复出现在新闻网页中，会对数据的可靠性造成一定的影响。但不能简单去除广告了事。
  1. 广告也是新术语的重要来源之一；
  2. 目前提取正文、去除广告的准确率较低；
  3. 广告一般在一个网页中只会出现一次，广告中的词串的统计特点是其TF和DF相等。可以优化评价函数提高对广告新术语的提取。
- 活跃用户的 ID 等也是造成系统准确率较低的重要原因。
  1. 系统具有一定的自适应能力，对于这些活跃ID只会错判一次；
  2. 目前提取正文、去除广告的准确率较低；
  3. 可以根据网页评论的格式，识别哪些词语是用户ID。

实验分析

- 大量的广告词在一段时间内反复出现在新闻网页中，会对数据的可靠性造成一定的影响。但不能简单去除广告了事。
  1. 广告也是新术语的重要来源之一；
  2. 目前提取正文、去除广告的准确率较低；
  3. 广告一般在一个网页中只会出现一次，广告中的词串的统计特点是其TF和DF相等。可以优化评价函数提高对广告新术语的提取。
- 活跃用户的 ID 等也是造成系统准确率较低的重要原因。
  1. 系统具有一定的自适应能力，对于这些活跃ID只会错判一次；
  2. 目前提取正文、去除广告的准确率较低；
  3. 可以根据网页评论的格式，识别哪些词语是用户ID。

结论

1. 基本完成了任务书中所提的日程和任务；
2. 实验表明，该算法已经能够较好地自动检测到相当数量的新术语，达到可用程度。
3. 该算法只是对新术语的预警，它只能减小人类查找新术语的范围，不能代替人类确定哪些是新术语。

结论

1. 基本完成了任务书中所提的日程和任务；
2. 实验表明，该算法已经能够较好地自动检测到相当数量的新术语，达到可用程度。
3. 该算法只是对新术语的预警，它只能减小人类查找新术语的范围，不能代替人类确定哪些是新术语。

结论

1. 基本完成了任务书中所提的日程和任务；
2. 实验表明，该算法已经能够较好地自动检测到相当数量的新术语，达到可用程度。
3. 该算法只是对新术语的预警，它只能减小人类查找新术语的范围，不能代替人类确定哪些是新术语。

结论

1. 基本完成了任务书中所提的日程和任务；
2. 实验表明，该算法已经能够较好地自动检测到相当数量的新术语，达到可用程度。
3. 该算法只是对新术语的预警，它只能减小人类查找新术语的范围，不能代替人类确定哪些是新术语。

未来工作

1. 优化评价函数，解决广告问题带来的性能下降；
2. 提供GUI界面，方便用户使用；
3. 增加词性标注等自然语言处理手段，提高算法的性能。

未来工作

1. 优化评价函数，解决广告问题带来的性能下降；
2. 提供GUI界面，方便用户使用；
3. 增加词性标注等自然语言处理手段，提高算法的性能。

未来工作

1. 优化评价函数，解决广告问题带来的性能下降；
2. 提供GUI界面，方便用户使用；
3. 增加词性标注等自然语言处理手段，提高算法的性能。

未来工作

1. 优化评价函数，解决广告问题带来的性能下降；
2. 提供GUI界面，方便用户使用；
3. 增加词性标注等自然语言处理手段，提高算法的性能。

致谢

谢谢各位老师、同学!

敬请指正!