基于内容的论文推荐系统

陈根宝

2013 年 6 月 21 日

目　录

1 引言 **3**

2 算法设计 **3**
2.1 算法输入 3
2.2 算法输出 3
2.3 算法设计 3
2.4 论文内容 3
2.4.1 论文内容特征词提取 4
2.4.2 内容相似度计算 4
2.5 论文作者 4
2.6 论文引用关系 4
2.7 读者行为 4

3 数据库设计 **4**
3.1 论文信息表 --- PaperInfo table 4
3.2 作者信息表 --- AuthorInfo table 5
3.3 论文作者关系表 --- PaperAuthorRel table 5
3.4 论文引用关系表 --- PaperRef table 5
3.5 用户推荐论文结果表 --- RecForU$uid table 5

4 模块设计 **5**

5 接口设计 **5**

6 **some troubles** **5**

1 引言

论文推荐旨在为闪记科技工作者用户提供论文推荐功能。论文推荐不同于一般的推荐，论文有很强文本特性，因此能够用基于内容的推荐（标题，关键字，摘要）；其次论文的作者一般具有特定的领域属性，因此论文能按其作者进行领域聚合；再次，论文一般都有相互之间的引用关系，这种引用关系可以看做是作者对相关文献的推荐，这好似是网页之间的相互链接，因此 pagerank 算法这里很好用。

2 算法设计

2.1 算法输入

我们的算法是以用户提交的阅读记录为推荐算法的输入。这些关键字可以从用户使用闪记产品对论文进行标记的时候获取，闪记客户端能在用户使用的时候识别论文，并能从其中摘取论文的五元组 --- 标题、作者、概要、引用、会议期刊名称，并将这些信息存入后台数据库，然后将用户 id 和论文 id 通过接口推送给推荐引擎。推荐引擎从后台数据库中读取论文五元组信息进行相关推荐。

2.2 算法输出

推荐算法的输出是与输入的阅读文献高度相关的 n（n >= 1）篇论文的信息，这个结果会以文献 id 的形式写入到用户推荐数据库中。

2.3 算法设计

正如引言中的论文推荐具有以上的一些特点，所以论文推荐的算法应该兼顾这些特点，因此我们将算法有以下几个影响因子:

- 论文内容，包含论文标题、概要、用户摘取内容以及关键字
- 论文作者
- 论文引用关系
- 读者行为

下面我们将对这些影响因子一一进行分析，并给出量化的计算方式。

2.4 论文内容

论文内容，这里指的并不是论文的正文，而是泛指区别于论文的作者、引用文献之外的一切文本实体，在此我们特指有论文的标题、概要、关键字和用户摘录内容混合之后经过一定处理得到的文本内容。

在用文本的内容进行推荐的时候，如果将论文内容和库里所有的文献进行内容相似度计算则计算量是十分大的，几乎是不能忍受的，问了降低计算复杂度，我们论文内容中抽取关键词并以此检索（因此事先我们要用 sphinx 对文献库建立倒排索引），然后将获得的文献集合与论文内容进行内容相似度匹配。

因此在该步骤中关键的有两步，1）提取论文内容特征词；2）内容相似度计算方法，下面我们分别给予介绍。

2.4.1 论文内容特征词提取

特征词提取的方法很多，有基于词性分析的、基于 TFIDF 阈值、基于 Bigram 的等多种方法。我们初步设计会选用 TD、-IDF 阈值的方法，这种方法实现起来会比较简单。但是英文中词的区别性不如词组好，比如 cloud computing 作为一个词组的时候表示的含义就很特殊，如果拆分则区分度会大大降低，所以我们应该在算法中加入词组的识别，这样会显著提高推荐效果。关于词组的识别，我认为可以采用字典的方法，将很多论文的 keyword 收集并存入字典中。

2.4.2 内容相似度计算

相似度计算比较简单，直接采用向量空间模型（vector space model）进行计算。

2.5 论文作者

根据实际经验，论文的作者一般会专注于某一个或几个领域的研究。因此可以将论文作者在知识领域的影响力作为影响论文推荐评分的一个指标。论文作者的影响力正比于其所发的论文的被引用次数。

2.6 论文引用关系

如果将 A 论文对 B 论文的引用看作是 A 对 B 的推荐，由此很自然的想到 Pagerank 算法，并由此可以计算出论文的重要性。

2.7 读者行为

在读者使用闪记产品过程中，可以将用户的阅读论文的行为看成是对论文的评分，将论文看做物品，从而可以用协同过滤进行推荐。这中推荐在闪记用户越来越大的情况下会越来越精确。

3 数据库设计

论文推荐需要用到以下数据库表。

3.1 论文信息表 --- PaperInfo table

该表描述论文的基本信息，有 id、标题、概要、关键字字段。建表语句如下。

```
CREATE TABLE PaperInfo(
    id INT NOT NULL,
    title VARCHAR(256),
    abstract TEXT,
    keywords VARCHAR(128),
    PRIMARY KEY(id)
)ENGINE=MyISAM  AUTO_INCREMENT=11 DEFAULT CHARSET=utf8;
```

3.2 作者信息表 --- AuthorInfo table

```
CREATE TABLE AuthorInfo(
    id INT NOT NULL,
    name VARCHAR(64),
    PRIMARY KEY(id)
)ENGINE=MyISAM  AUTO_INCREMENT=11 DEFAULT CHARSET=utf8;
```

3.3 论文作者关系表 --- PaperAuthorRel table

```
CREATE TABLE PaperAuthorRel(
    id INT NOT NULL,
    author_id INT NOT NULL,
    paper_id INT NOT NULL,
    PRIMARY KEY(id)
)ENGINE=MyISAM  AUTO_INCREMENT=11 DEFAULT CHARSET=utf8;
```

3.4 论文引用关系表 --- PaperRef table

```
CREATE TABLE PaperRef(
    id INT NOT NULL,
    ref_id INT NOT NULL,
    beref_id INT NOT NULL,
    PRIMARY KEY(id)
)ENGINE=MyISAM  AUTO_INCREMENT=11 DEFAULT CHARSET=utf8;
```

3.5 用户推荐论文结果表 --- RecForU$uid table

```
CREATE TABLE RecForU$uid(
    id INT NOT NULL,
    paper_id INT NOT NULL,
    recom_time DATETIME NOT NULL,
    PRIMARY KEY(id)
)ENGINE=MyISAM  AUTO_INCREMENT=11 DEFAULT CHARSET=utf8;
```

对于每一个用户都有这样一张表，其中 $uid 为用户的 id。

4 模块设计

5 接口设计

6 some troubles