基于搜索引擎的词汇语义相似度计算方法

陈海燕
(华东政法大学计算机科学与技术系 上海 201620)

摘 要 词汇语义相似度的计算在网页浏览和查询推荐等网络相关工作中起着重要的作用。传统的基于分类的方法不能处理持续出现的新词。由于网络数据中隐藏着大量的噪音和冗余,鲁棒性和准确性仍然是一个挑战,因此提出了一种基于搜索引擎的词汇语义相似度计算方法。语义片段和检索结果的页数被用来去除词汇语义相似度计算过程中的噪音和冗余。此外,还提出了一种方法来整合查询结果页数、语义片段和显示的搜索结果的数量,该方法不需要任何先验知识与本体。实验结果显示,所提出的方法在 Rubenstein-Goodenough 测试集的相关系数为 0.851,优于现有的基于网络的词汇语义相似度计算方法,同时在搜索引擎的查询扩展任务中具有较为良好的应用效果。

关键词 语义相似度,信息检索,查询建议,网络检索
中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.1.058

Measuring Semantic Similarity between Words Using Web Search Engines

CHEN Hai-yan
(Department of Computer Science and Technology, East China University of Political Science and Law, Shanghai 201620, China)

Abstract Semantic similarity measures play important roles in many Web-related tasks such as Web browsing and query suggestion. Because taxonomy-based methods cannot deal with continually emerging words, recently Web-based methods have been proposed to solve this problem. Because of the noise and redundancy hidden in the Web data, robustness and accuracy are still challenges. We proposed a method integrating page counts and snippets returned by Web search engines. Then, the semantic snippets and the number of search results were used to remove noise and redundancy in the Web snippets. After that, a method integrating page counts, semantics snippets and the number of already displayed search results was proposed. The proposed method does not need any human annotated knowledge, and can be applied Web-related tasks easily. A correlation coefficient of 0.851 against Rubenstein-Goodenough benchmark dataset shows that the proposed method outperforms the existing Web-based methods by a wide margin. Moreover, the proposed semantic similarity measure significantly improves the quality of query suggestion against some page counts based methods.

Keywords Semantic similarity, Information retrieval, Query suggestion, Web search

1 引言

词汇语义相似度的研究在学术研究和应用中都是一个重要的问题。例如:词义消歧[1]、知识流构造[2-5]、图像检索[6]、自然语言处理[7]、主题检测[8]、查询推荐等[9]。

近年来,随着网络的快速发展,在许多 Web 相关任务中词汇语义相似度的计算也越来越重要。在查询推荐方面[10],例如 Google 的"Search related to"和雅虎的"Also Try"都提供一系列相关词来帮助用户找到最想要的结果,从而改善用户的搜索体验和检索效率。

除了 Web 相关业务外,词汇语义相似度的计算也在语义网中得到应用,包括 Web 页面的自动标注[11]和半结构化数据搜索[12]等。例如:"apple"是一种水果,也是一个公司。一个用户在网络上搜索"apple",可能是对这两种不同解释中的一种感兴趣。在这个例子中,"apple"附近的词语可以通过语义相似度来辅助查询。

即使是在电子领域,词汇语义相似度的计算也起着重要的作用。在付费搜索模型中,商业机构通过支付搜索引擎费用使自己网站的广告排在搜索结果的前面。

目前,现有的基于 Web 的词汇语义相似度的计算方法研究可以分为 3 类:

1)基于查询结果数的方法。通过搜索引擎获取的查询结果数是一个重要的信息资源。例如:查询"Microsoft CEO AND Bill Gates"在 Google 上的结果是 191000 条,而"Microsoft CEO AND Steve Ballmer"的结果是 844000 条,事实上"Steve Ballmer"比"Bill Gates"在相似度程度上更加接近于"Microsoft CEO",因为后者的查询结果数大于前者。

2)基于查询结果文本内容的方法。下载大量搜索排名在前面的文档,并将其进行文本处理。Resnik[13]采用网络作为语料库,因为新的词汇会在网络中不断出现。这些方法通过

到稿日期:2014-03-21 返修日期:2014-05-18 本文受国家社会科学基金项目(06BFX051),上海高校选拔培养优秀青年教师科研专项基金(hzf05046)资助。

陈海燕(1978—),男,博士生,讲师,主要研究方向为计算机网络、数据挖掘、人工智能、信息安全,E-mail:tom_chy@163.com。

使用搜索引擎，将其作为一种处理海量网络信息的一种有效手段。这些方法认为出现在相似语境中的词汇具有很高的语义相似度。

3)以上两种方法的整合。

通过相关分析，现有的方法没有考虑到搜索引擎反馈的结果中存在噪音和冗余。噪音的来源主要是词汇随机地出现在一些页面中，这将会降低页面搜索数目的准确度。除此之外，冗余也是影响词汇语义相似度准确的一个重要因素。相关研究[14]表明爬虫访问过的1.7%～7%的网络页面是重复出现的。很多重复出现的网页使得搜索结果数目不可信。本文的主要贡献如下：

1)文中通过整合检索页面数目和检索结果片段来计算词汇之间的语义相似度。本文提出的方法是自动的，不需要人工干预，而且易于被应用到如查询建议等与网络相关的工作中。

2)本文第一次提出噪音与冗余在基于Web的词汇语义相似度的计算方法中的影响。本文提出利用词汇共同出现在一句话中来去除噪音，利用搜索引擎的重复记录数来去除冗余。

3)本文提出的方法在Rubenstein-Goodenough测试集上进行了测试。测试相关系数为0.851，说明了本方法能够有效地计算词语间的相似度。同时，所提出的方法可以提高查询扩展任务的准确度。

本文第2节是相关工作；第3节是本文算法的具体描述；第4节是实验结果和讨论，并给出一个在查询建议中的使用实例；最后总结全文。

2 相关工作

词汇语义相似度计算主要可以分为两个方面：基于语料库和基于网络的方法。基于语料库的方法使用信息论和开源语料库(例如WordNet0)。基于网络的方法使用网络作为一个不断更新的开放语料库。

Resnik[1]提出了使用信息内容来衡量词汇语义相似度。基于语料库的方法主要利用信息，来计算词汇语义相似度。Richardson[15]不仅考虑距离，还考虑密度、深度和边缘强度等统计特征来计算词汇语义相似度。在文献[16]中，作者还考虑了连接的类型、深度和密度。Jiang[17]提出了综合信息内容和距离的组合模型来计算词汇语义相似度。Erk[18]利用向量空间模型来计算词汇语义相似度。Li[19]通过多个信息源来计算词汇语义相似度，信息源包括词汇分类中的结构化词语和语料库中的信息内容。因为新的词汇不断产生，已经存在的词汇也会不断有新的语义。向语义库中人工地不断加入新词和新的含义是不切实际的。

与基于语料库的方法不同，Turney[20]利用互信息来计算词汇语义相似度。Chen[21]提出了另一种名为co-occurrence double check的方法，他将网络作为一个不断更新的语料库，这种方法在很大程度上依赖于搜索引擎的排序算法。Sahami[22]通过基于搜索引擎的相似性函数来计算词汇语义相似度。文献[23]提出了一种新的基于搜索引擎的词汇语义相似度计算方法，作者将这种方法命名为second order co-occurrence PMI，并用互信息来对这两个目标词汇的重要相邻词汇进行排序。Bollegala[24]利用搜索引擎查询的网页数和检索片断来计算词汇语义相似度。作者从4562471个不同的样式中提取出了200种模式。因为模式是随着时间不断变化的，大量数目的样式的更迭(大约5百万次)使得这种方法非常耗时。因此，提取模式对于这种方法有很大的影响。

总之，现有的基于网络的词汇语义相似度计算方法都缺乏相关的机制来处理数据中的噪音和冗余。

3 基于去噪和去冗余的词汇语义相似度计算

3.1 基于查询页数的词汇语义相似度计算

查询页数是指包含查询词语q的网页数目。例如：用Google查询"Obama"有297000000条结果。在本文的剩余部分，将使用符号$N(q)$表示用Google查询q的返回查询页数。然而，词语p和q的单独的查询页数不足以计算其语义相似度，还应该加入查询"p AND q"的查询页数。例如：当用Google查询"Obama"和"United States"时，可以查到110000000个网页，也就是$N(\text{Obama} \cap \text{United States}) = 110000000$。

基于查询页数的词汇语义相似度计算核心是"you shall know a word by the company it keeps"[25]。在本文中，Jaccard、Overlap、Dice和PMI等4种方法被用来计算词汇语义相似度，具体公式如下。

$$Jaccard(p,q)=\frac{N(p\cap q)}{N(p)+N(q)-N(p\cap q)} \tag{1}$$

$$Overlap(p,q)=\frac{N(p,q)}{\min(N(p),N(q))} \tag{2}$$

$$Dice(p,q)=\frac{2*N(p\cap q)}{N(p)+N(q)} \tag{3}$$

$$PMI(p,q)=\log(\frac{N*N(P\cap q)}{N(p)*N(q)})/\log N \tag{4}$$

其中，N是搜索引擎中的网页数目，设为$N=10^{11}$。

使用查询页数计算词汇语义相似度有以下缺点：

1)忽略了网络数据中存在的噪音。由于网络的规模巨大，词汇可能会随机地出现在一些网页中，因此需要减少两个词汇随机出现的情况，以提高语义相似度计算的准确度。

2)忽略了网络数据中存在的冗余。除了词汇会随机地出现在网页中外，网页也存在大量的重复，因此，搜索引擎返回的查询页数是不准确的，大量重复的页面应该去除，以提高语义相似度计算的准确度。

3.2 利用语义片段去除噪音

正如之前提到的网络噪音将影响词汇语义相似度计算的准确度，因此基于查询页数的词汇语义相似性计算方法中的$N(p\cap q)$部分需要进行修正。由于网络规模巨大，通过扫描所有网页来找到包含词汇p和q是不切实际的。

搜索引擎返回搜索结果时也会返回检索结果片段，这些片段通常是不超过30个词的短小的文本，这些文本为我们提供了非常重要的语义信息。例如：图1是查询"computer AND hardware"的结果。在图1中，"computer"和"hardware"出现在一句话中，我们可以看出它们具有语义相似性，因为"components of"表示"hardware"和"computer"之间是部分与整体的关系。

In addition, hardware devices can include external components of a computer system. The following are either standard or vey common.
…

图1 用Google查询"computer AND hardware"的一个结果片段

不是所有的片段都表示了词语 p 和 q 之间具有有效的语义关系。图 2 是查询"food AND fruit" 的结果，与图 1 不同的是，虽然这个片段在 Google 返回结果的前列，但是并没有提供有效的语义信息，因为两个词语没有在同一句话里。

Freeze no more food. at one time than will freeze within 24 hours-usually two. to three pounds of fruit per cubic foot of freezer space. …

图 2 用 Google 查询"food AND fruit" 的一个无意义的结果片段

根据图 1 和图 2 的对比，提出以下定义：

定义(语义片段) 语义片段是指词语 p 和 q 在同一个句子中共同出现的检索结果片段。片段中以句号、问号、省略号或感叹号为结尾的称之为一个句子。

根据定义，图 1 中的片段是一个语义片段，因为它提供了词语 p 和 q 之间的有用的语义关系。因此语义片段可以用来判断两个词汇是否是随机地出现在网络页面中。

3.3 根据重复记录数目去除冗余

除了噪音以外，冗余也是影响词汇语义相似度计算准确度的重要因素。Google 提供了每个结果的链接，直观上看可以通过这个途径来消除冗余，但是，由于网页数目巨大，而且增长速度快，因此对每个搜索结果进行直接的分析是非常困难的。此外，不同的网址有不同的网页格式，在巨大的网络范围内逐一分析网站也是不切实际的。

Google 提供了一个去除重复结果的功能。当用 Google 搜索时，将有 1000 个以上的搜索结果，为了使结果的相关度高，Google 省略了一些非常相似的搜索结果。例如："Obama" 的搜索结果中，Google 省略了 194 条重复记录。Google 的重复记录数目可以用来去除冗余。

3.4 基于查询页数、去噪和去冗余的词汇语义相似度计算

本节提出一个通过整合查询页数、语义片段和重复记录数目来计算词汇语义相似度的方法。

策略 1：词汇间的语义相似程度是由查询页数和语义片段决定的。主要步骤如下：

1)在搜索引擎中分别搜索"p"，"q"，"p AND q"；

2)得到 $N(p)$，$N(q)$和 $N(p\cap q)$；

3)在"p AND q"的结果中，计算语义片段在前 n 个片段中的比例，记为：$SS(p\cap q)$；例如在搜索结果前 100 个片段中，p、q 同时出现在一句话的语义片段有 40 个，则 $SS(p\cap q)$ 为 $40/100=40\%$ 。

4)用 $N(p\cap q)*SS(p\cap q)$代替 $N(p\cap q)$。

策略 1 通过使用语义片段来修正基于查询页数的方法中的 $N(p\cap q)$，这样可以除掉噪音。例如：根据这个策略，用基于语义片段来修正 PMI，定义如式(5)所示：

$$SPMI(p,q)=\log_2\left(\frac{N(p\cap q)*N*SS(p\cap q)}{N(p)*N(q)}\right)\frac{1}{\log_2 N} \tag{5}$$

策略 2：词汇间的语义相似程度是查询页数和重复记录数目共同决定的。主要步骤如下：

1)在搜索引擎中分别搜索"p"，"q"，"p AND q"；

2)得到 $N(p)$，$N(q)$和 $N(p\cap q)$；

3)得到搜索"p"，"q"，"p AND q"时的重复记录数目，记为：$R(p)$，$R(q)$和 $R(p\cap q)$；

4)用 $N(p)*R(p)$，$N(q)*R(q)$和 $N(p\cap q)*R(p\cap q)$代替 $N(p)$，$N(q)$和 $N(p\cap q)$。

该策略用重复记录数目修正了基于查询页数的方法中的 $N(p)$，$N(q)$和 $N(p\cap q)$，这样可以减少网络数据中的冗余。例如：根据该策略，用基于重复记录数目来修正 PMI，如式(6)所示：

$$RPMI(p,q)=\log\left(\frac{N(p\cap q)*N*R(p\cap q)}{N(p)*R(p)*N(q)*R(q)}\right)\frac{1}{\log N} \tag{6}$$

策略 3：两个词汇间的语义相似度是通过策略 1 和策略 2 共同决定的，即不仅考虑语义片段，还考虑重复记录数目。例如：根据该策略，用基于语义片段和重复记录数目来修正 PMI，如式(7)所示：

$$SRPMI=\begin{cases}0, & \text{if } SS(p\cap q)<\alpha\\ \left(\dfrac{N(p\cap q)*N*SS(p\cap q)*R(p\cap q)}{N(p)*R(p)*N(q)*R(q)}\right)\dfrac{1}{\log N}, & \text{else}\end{cases} \tag{7}$$

与策略 1 类似，在这个策略中，参数 α 对结果影响很大。第 4 节中将会详细分析参数的影响。

4 实验及结果分析

本文通过返回结果分别计算整合查询页数、语义片段和重复记录数，并根据式(5)－式(7)分别计算出 3 种策略的词汇语义相似度。

本文实验使用的是 R-G 标准数据集。Rubenstein 和 Goodenough[27] 提出了一个包含 28 个词汇对的语义相似度评分的数据集。词汇语义相似度计算结果和 R-G 数据集中的相关系数越高，表示该方法越准确。

4.1 选择查询页数计算公式

利用 R-G 数据集和 80 个 TOEFL 问题，对 Jaccard、Overlap、Dice、PMI 4 种算法进行评估，图 3 和图 4 分别是 R-G 数据集和 80 个 TOEFL 问题的结果。图 3 和图 4 显示，PMI 相对于 R-G 数据集来说相关系数最高，在 80 个 TOEFL 问题中准确度也最高。因此，本文选择 PMI 作为查询页数的计算公式。

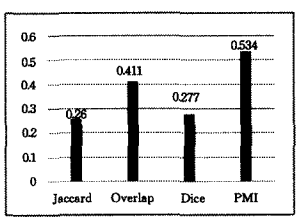

图 3 4 种基于页面数的语义相似度测量方法对 Rubenstein-Goodenough 数据集的相关系数

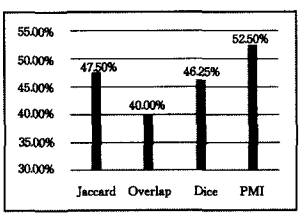

图 4 4 种基于页面数的语义相似度测量方法对 80 个 TOEFL 问题集的准确度

4.2 整合 PMI 和语义片段

本节测量策略 1 的性能，策略 1 整合了 PMI 公式和语义片段。在给出策略 1 的结果之前，本文首先引入了基于上下文窗口的方法，因为语义片段与这种方法很相似。基于上下文的窗口的方法使用了一个固定大小的上下文窗口(K 个词)来提取语义片段。不同于本文提出的方法，基于上下文窗口的方法认为词汇组出现在一个长度为 K 的窗口中便可以认

为是语义片段。实验比较了策略 1 和基于上下文窗口的方法，结果如表 1 所列。此外，因为 Google 返回的片段数目达到了 1000，实验计算了对于不同数目的片段和 R-G 数据集的相关系数。

表 1　窗口的长度与精度

有效片段数目	K=25	K=12	K=6	K=5	K=4	K=3
100	0.679	0.687	0.711	0.723	0.732	0.722
200	0.702	0.709	0.726	0.736	0.744	0.741
300	0.713	0.718	0.736	0.744	0.752	0.751
400	0.714	0.718	0.736	0.744	0.752	0.751
500	0.715	0.719	0.738	0.748	0.759	0.765
600	0.714	0.718	0.735	0.744	0.754	0.760
700	0.722	0.725	0.736	0.745	0.754	0.760
800	0.730	0.732	0.738	0.746	0.755	0.761
900	0.738	0.740	0.744	0.751	0.759	0.764

实验结果见图 5，可以得出以下结论：

1）SPMI 对于 R-G 数据集的相关系数是随着搜索结果数量单调递增的。因此，在本文中计算语义片段的搜索结果数量被设定为 900。

2）相对于 PMI，SPMI 显著提高了相对于 R-G 数据集的相关系数。SPMI 最好的准确度是 0.7695，比 PMI 提高了 44%。不同于 PMI，SPMI 使用语义片段去除网络数据中的噪音，这使得词汇 p 和 q 的同时出现更加准确。

3）SPMI 比基于上下文窗口的方法在相对于 R-G 数据集的相关系数中表现得更加出色。基于上下文窗口的方法的最好的相关系数是 0.765。这个结果证明了基于句子的词汇共现比基于窗口的词汇共现更加准确。

因此，SPMI 的较高相关系数证明了使用语义片段将提高词汇语义相似度计算的准确度。

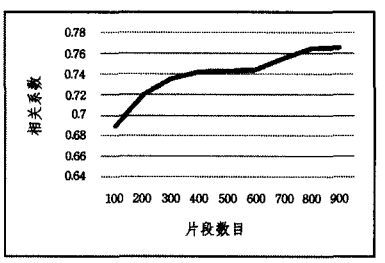

图 5　不同片段数时 SPMI 算法的相关系数

4.3　整合 PMI 和重复记录数目

本节测量策略 2 的性能，策略 2 整合了 PMI 公式和重复记录数目。实验结果如下：

1）相对于 PMI，RPMI 提高了相对于 R-G 数据集的准确度。RPMI 的相关系数是 0.542，比 PMI 提高了 1.5%。因为 RPMI 使用了 Google 提供的重复记录数目，这将去除掉网络数据中的冗余。

2）RPMI 没有 SPMI 效率提升得多，说明网络数据中噪音比冗余对于计算词汇语义相似度的准确性影响更大。

4.4　整合 PMI、语义片段和重复记录数目

本节测量策略 3 的性能，策略 3 整合了 PMI 公式、语义片段和重复记录数目。因为参数 α 对词汇语义相似度计算具有影响，本文测量了不同的 α 相对于 R-G 数据集的相关系数。实验结果见图 6，SRPMI 最好的相关度是 0.851，此时 α 为 0.38，其相对于 SPMI 和 RPMI 分别提升了 10.6% 和 57%。因此，SRPMI 整合 PMI 公式、语义片段和重复记录数目是最好的策略。

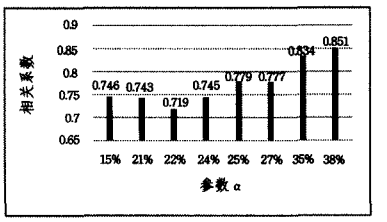

图 6　不同参数 α 时 SRPMI 算法的相关系数

4.5　基于网络的方法和 SRPMI 的比较

前人已经提出的一些基于网络的语义相似度检测方法，如：Sahami[22]、CODC[21] 和 SemSim[24]，在本式子将被用来与 SRPMI 进行比较。选择这 3 个方法的原因是：(1) SemSim[24] 声称在当前的基于网络的语义相似度检测中有着最高的相关系数；(2) Sahami[22] 只考虑了片段，而 CODC[21] 只考虑了页面数目。比较结果见表 2 和表 3，在我们的实验中 SRPMI 为 0.851 的相关系数表现最好。

表 2　不同指标的属性(1 表示含有该属性)

检测方法	网络搜索引擎	页面数	片段	Lexico 语法模式	WordNet 本体	下载文档	外部知识	去除噪音
PMI	1	1	0	0	0	0	0	0
Sahami	1	0	1	0	0	0	0	0
CODC	1	0	1	0	0	0	0	0
SemSim	1	1	1	1	1	1	1	0
SPMI	1	1	1	0	0	0	0	0
RPMI	1	1	1	0	0	0	0	1
SRPMI	1	1	1	0	0	0	0	1

SemSim 的方法是次优的，相关系数为 0.797。SemSim 采用了从文本片段中自动提取词汇模式来测量词语间的语义相似度。SemSim 的 200 个模式是从 4562471 个片段得到的。因为最接近的片段是随着时间的变化而改变的，大量的(通常是 500 万)不同模式的更改使得 SemSim 算法很费时。因此，SemSim 算法是受提取出的模式影响的。

Chen[21] 提出的 CODC 算法表现出的相关度是 0.723。CODC 算法用式(8)来定义：

$$CODC(p,q)=\begin{cases}0, & \text{if } f(p@q)=0\\ e^{-\log[\frac{f(p@q)}{N(p)}*\frac{f(q@p)}{N(q)}]^{\alpha}}, & \text{otherwise}\end{cases} \quad (8)$$

其中，$f(p@q)$ 表示用 Google 查询 q 时，最接近的结果片段中含有 p 的数目。α 是 CODC 算法中的一个定值，将它设为 0.15。即使词语 p 和 q 是一对语义相似的词语，搜索其中一个时也不一定每次都能够得到另一个词。因此，对于很多 R-G 数据集中的词对 CODC 算法给出了结果为 0。

Sahami 提出的相似度检测方法的相关系数为 0.621。这个方法在检测相似度时使用了片段来代替，它仅仅使用片段来计算词语间的相似度，而忽略了 Google 返回的页面数。而且，Sahami 没有将其算法与其它基于网络的算法进行比较。

与其它基于网络的算法不同，SRPMI 有以下优点：

1）SRPMI 整合了页面数目和片段来检测词语间的语义相似度。而 PMI 只使用了页面数，其相关系数为 0.534。Sahami 只使用了片段，其相关系数为 0.621。

2）SRPMI 使用语义片段和重复结果数目来去除噪音和

冗余，这将提高语义相似度检测的准确度。CODC 和 SemSim 将重点放在所有的相关结果，而忽略了不是所有的结果都提供了有效的语义信息。其相关系数分别是 0.723 和0.797，都不及 SRPMI 准确。

综上所述，表 3 的结果表明本文提出的 SRPMI 算法比其它现有的基于网络的算法更加准确。

表 3　人工语义相似度评分和 R-G 数据集的基线

Word Pair	R-G Ratings	PMI	Sahami	CODC	SemSim	SRPMI (α=0.38)
chord-smile	0.02	0.146	0.090	0	0	0
rooster-voyage	0.04	0.152	0.197	0	0.017	0
noon-string	0.04	0.152	0.082	0	0.018	0
glass-magician	0.44	0.214	0.143	0	0.18	0
monk-slave	0.57	0.200	0.095	0	0.375	0
coast-forest	0.85	0.186	0.248	0	0.405	0.171
monk-oracle	0.91	0.151	0.045	0	0.328	0
lad-wizard	0.99	0.164	0.149	0	0.22	0
forest-graveyard	1	0.184	0	0	0.547	0
food-rooster	1.09	0.150	0.075	0	0.06	0.123
coast-hill	1.26	0.173	0.293	0	0.874	0.153
car-journey	1.55	0.142	0.189	0.290	0.286	0.127
crane-implement	2.37	0.186	0.152	0	0.133	0.156
brother-lad	2.41	0.182	0.236	0.379	0.344	0.161
bird-crane	2.63	0.186	0.223	0	0.879	0.165
bird-cook	2.63	0.177	0.058	0.502	0.593	0.177
food-fruit	2.69	0.175	0.181	0.338	0.998	0.167
brother-monk	2.74	0.187	0.267	0.547	0.377	0.192
asylum-madhouse	3.04	0.245	0.212	0	0.773	0.230
furnace-stove	3.11	0.234	0.310	0.928	0.889	0.234
magician-wizard	3.21	0.220	0.233	0.671	1	0.211
journey-voyage	3.58	0.202	0.524	0.417	0.996	0.188
coast-shore	3.6	0.203	0.381	0.518	0.945	0.187
implement-tool	3.66	0.185	0.419	0.419	0.684	0.169
boy-lad	3.82	0.181	0.471	0	0.974	0.185
automobile-car	3.92	0.169	1	0.686	0.98	0.158
midday-noon	3.94	0.228	0.289	0.856	0.819	0.190
gem-jewel	3.94	0.198	0.211	1	0.686	0.237
Correlation Coefficient	1	0.534	0.621	0.723	0.797	0.851

4.6　将 SRPMI 算法用于查询推荐

在这节中，我们将提出的词汇语义相似度计算方法应用到查询推荐任务中。

搜索引擎的查询词通常都是短而且目的明确的，通常它提供的信息不足以使用户有效地检索到需要的页面。为了解决这个问题，大多数搜索引擎都使用了查询推荐，例如：Google 的“Search related to”和雅虎的“Also Try”功能。查询推荐功能提供了一系列与用户原始查询词语义相似的词，从而提高了用户的搜索体验和搜索效果。本文提出的词汇语义相似度计算方法可以用来选择合适的推荐词语。基于 SRPMI 的查询推荐步骤如下：

1)在 Google 上查询词汇 p。

2)提取查询词汇 p 的相关词汇作为关键词来作为查询推荐。相关词汇提取方法采用数据挖掘中频繁模式挖掘算法[28]。根据认知科学理论[29]，如果一个词汇频繁地出现在查询词汇 p 的结果片段中，那么它就是一个与查询词汇 p 相近的概念，因为它和查询词汇 p 共同存在于搜索结果中。

3)用 SRPMI 算法计算提取出的词汇和查询词汇 p 的语义相似度。

4)选择词汇中的语义相似度高的词汇作为查询推荐词。

为了评估基于 SRPMI 的查询推荐方法，实验邀请 50 个评估人来对 Google、雅虎和本文提出的方法的搜索推荐词汇进行评价。每个评估人采用 5 分制[22]对查询推荐词汇进行评价，评分越高，该方法就越好。

评估的查询词是从 Google Zeitgeist 得到的。Google Zeitgeist 每个月都在更新热门搜索词。使用 Google Zeitgeist 的原因如下：

1)Google Zeitgeist 中的搜索词是全世界用户最热门的搜索词。这些搜索词对于评估者来说更加熟悉。评估者如果能明确知道查询词的意思，就能够更加正确地评价查询推荐词。

2)因为 Google Zeitgeist 中的搜索词非常热门，如果能找到有效的查询推荐词，它们将会适用于查询相同信息的大量用户。

表 4 列出了本文评估中使用的查询词。这些查询词都是 Google 2004 年到 2008 年的前 10 名热门搜索词。在表 5 中，本文以“MSN”作为一个示例搜索推荐。图 7 中显示的是基于 SRPMI 的查询推荐方法的结果评估。根据图 7 可以看出，我们的方法比 Google 和雅虎的方法有效，因为在每一年中，我们的方法的评分均比 Google 和雅虎的高；还可以看出我们方法的人工评估结果高于其它 3 个，推荐的概念词和查询词非常接近。

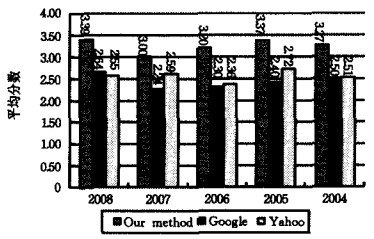

图 7　关于 2004 年到 2008 年的这些查询使用本文方法、Google、Yahoo 的结果

表 4　用于查询推荐的查询词，即 2004－2008 年 Google 的查询词汇的前 10 名

Year	2008	2007	2006	2005	2004
Ranking1	Sarah Palin	iPhone	Bebo	Myspace	Britney Spears
Ranking2	Beijing 2008	Badoo	Myspace	Ares	Paris Hilton
Ranking3	Facebook login	Facebook	World cup	Baidu	Christina Aguilera
Ranking4	Tuenti	Dailymothion	Metacafe	Wikipedia	Pamela Anderson
Ranking5	Health ledger	Webkinz	Radioblog	Orkut	Chat
Ranking6	Obama	Youtube	Wikipedia	Itunes	Games
Ranking7	Nasza Klasa	Ebuddy	Video	Sky News	Carmen Electra
Ranking8	Wer kennt wen	Second life	Rebeled	World of Warcraft	Orlando Bloom
Ranking9	Euro 2008	Hi5	Mininova	Green day	Harry Potter
Ranking10	Jonas brothers	Club penguin	Wiki	Leonardo daVinci	Mp3

表 5　不同推荐方法对于 MSN 的推荐词

Query	SRPMI	Google search related	Yahoo also try
MSN	Microsoft	Japan	Hotmail
	Shopping	Names	Messenger
	Weather	Games	Games
	Movies	Groups	Plus
	Sports	Web	7.5
	Network	Dollies	Groups
	Hotmail	Plus	Music
	Games	Blocker	Names

结束语　本文主要是基于网络的词语间语义相似度计算。本文认为，由于网络数据中的噪音和冗余，词语随机地出现在一些页面中。检测语义相似度的算法用于去除噪音和冗余的影响，以提高准确度和鲁棒性。

4 个基于页面数的方法被用来作为检测语义相似度的候选方案，包括：Jaccard、Overlap、Dice 和 PMI 等。用语义片段而不是所有的搜索结果片段来解决噪音问题。语义片段是含有查询词 p 和 q 在一个句子中的片段，这样可以去除搜索结果中的噪音。此外，Google 提供的重复出现的结果数目被用来去除冗余。本文提出了 3 种整合了页面数目、语义片段和重复出现页面数目的策略来测量词语间的语义相似度。

除此之外，80 个 TOEFL 问题和 R-G 数据集被用来挑选最好的基于页面数的算法，结果显示 PMI 优于其他方法的结果。R-G 数据集是用来选择最好的策略的，结果显示本文提出的结果优于当前使用的方法，这说明在测量语义相似度时使用语义片段和重复页面数目可以提高准确度。将本文提出的算法与目前已知的基于网络的方法比较，如：CODC、Sem-Sim 等。本文提出的方法以与 R-G 标准数据集 0.851 的相关系数表现大幅优于现有的基于网络的方法。

最后，本文将提出的语义相似度的算法应用于一个实际问题：查询推荐。本文提出的算法表现非常卓著，相比一些非常受欢迎的搜索引擎，如：Google 和 Yahoo，它提高了查询建议的质量，从而验证了该方法通过去除噪音和冗余提高了捕捉语义信息的能力。

参 考 文 献

[1] Resnik P. Semantic similarity in a taxonomy: an information based measure and its application to problems of ambiguity in natural language[J]. Journal of Artificial Intelligence Research 1999,11:95-130

[2] Luo X, Hu Q, Xu W, et al. Discovery of textual knowledge flow based on the management of knowledge maps[J]. Concurrency and Computation: Practice and Experience, 2008, 20: 1791-1806

[3] Luo X, Xu Z, Li Q, et al. Generation of similarity knowledge flow for intelligent browsing based on semantic link networks [J]. Concurrency and Computation: Practice and Experience 2009, 21: 2018-2032

[4] Luo X, Yu J, Li Q, et al. Building web knowledge flows based on interactive computing with semantics[J]. New Generation Computing, 2010, 28: 113-120

[5] Zhang S, Luo X, Chen J, et al. Measuring knowledge delivery quantity of associated knowledge flow[C]// Proceedings of the Fourth International Conference on Semantics, Knowledge and Grid. IEEE Computer Society: Washington, DC, 2008: 117-124

[6] Smeulders A, Worring M, Santini S, et al. Content-based image retrieval at the end of the early years[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(12): 1349-1380

[7] Srihari R, Zhang Z, Rao A. Intelligent indexing and semantic retrieval of multimodal documents [J]. Information Retrieval, 2000, 2: 245-275

[8] Makkonen J, Ahonen-Myka H, Salmenkivi M. Simple semantics in topic detection and tracking[J]. Information Retrieval, 2004, 7: 347-368

[9] Green S J. Building hypertext links by computing semantic similarity[J]. IEEE Transactions on Knowledge and Data Engineering, 1999, 11(5): 713-730

[10] Vojnovic M, Cruise J, Gunawardena D, et al. Ranking and suggesting popular items[J]. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(8): 1133-1146

[11] Cimano P, Handschuh S. Towards the self-annotating web[C]// Proceedings of the 13th International World Wide Web Conference. ACM Press: New York, 2004: 462-471

[12] Schenkel R, Theobald A, Weikum G. Semantic similarity search on semistructured data with the XXL search engine[J]. Information Retrieval, 2005, 8: 521-545

[13] Resnik P, Smithm A. The Web as a parallel corpus[J]. Computational Linguistics 2003, 29(3): 349-380

[14] Xiao C, Wang W, Lin X, et al. Efficient similarity joins for near duplicate detection [C] // Proceedings of 17 th International World Wide Web Conference. ACM Press: New York, NY, 2008: 131-140

[15] Richardson R, Smeaton F. Using WordNet in a knowledge-based approach to information retrieval[D]. Working Paper, CA-0395, School of Computer Applications, Dublin City University, Ireland, 1999

[16] Sussna M. Word sense disambiguation for free-text indexing using a massive semantic network[C]// Proceedings of the Second International Conference on Information and Knowledge Management. ACM Press: New York, NY, 1993: 67-74

[17] Jiang J J, Conrath D W. Semantic similarity based on corpus statistics and lexical taxonomy[C]// Proceedings of International Conference Research on Computational Linguistics. 1997

[18] Herdagdelen A, Erk K. Measuring semantic relatedness with vector space models and random walks[C]// Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing. 2009: 50-53

[19] Li Y, Bandar A, McLean D. An approach for measuring semantic

similarity between words using multiple information sources [J]. IEEE Transaction on Knowledge and Data Engineering, 2003,15(4):871-882

[20] Turney P D. Features of similarity[J]. Psychological Review, 1997,84(4):327-352

[21] Chen H, Lin M, Wei Y. Novel association measures using web search with double checking[C]// Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics. 2006:1009-1016

[22] Sahami M, Heilman D. A Web-based kernel function for measuring the similarity of short text snippets[C]// Proceedings of the 15th International World Wide Web Conference. ACM Press: New York, NY, 2006:377-386

[23] Islam A, Inkpen D. Second order co-occurrence PMI for determining the semantic similarity of words[C]// Proceedings of the International Conference on Language Resources and Evaluation. 2006:1033-1038

[24] Bollegala D, Matsuo Y, Ishizuka M. Measuring semantic similarity between words using web search engines[C]// Proceedings of 16th International World Wide Web Conference. ACM Press: New York, NY, 2007:757-766

[25] Firth R. A synopsis of linguistic theory 1930-1955[D]. Studies in Linguistic Analysis, Philological Society: Oxford, 1957

[26] Bayardo R J, Ma Y, Srikant R. Scaling up all pairs similarity search[C]// Proceedings of 16th International World Wide Web Conference. ACM Press: New York, NY, 2007:131-140

[27] Rubenstein H, Goodenough B. Contextual correlates of synonymy[J]. Communications of the ACM, 1965, 8(10):627-633

[28] Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases[C]// Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. Vol. 22, ACM Press: New York, NY, 1993:207-216

[29] Church W, Hanks P. Word association norms, mutual information and lexicography[C]// Proceedings of the 27th Annual Conference of the Association of Computational Linguistics. 1989:76-83

(上接第 235 页)

实验中，AP 算法将 Soybean 数据集分为 5 类，M-AP 算法将其准确地分为 4 类，而 K-means 算法和 FCM 算法也需要预先设置好类别。另外，从表 4 可以看出，DBSCAN 算法在此数据集中表现很差，而 M-AP 算法相对于 AP 算法在指标 RI 和 NMI 上略有下降，在指标 AC 和 Purity 上均有很大的提升。而 M-AP 算法的所有指标均明显优于 K-means 算法和 FCM 算法。

由上述 4 个实验可以看出，M-AP 算法在整体上相对于 AP 算法、K-means 算法、FCM 算法以及 DBSCAN 算法都有一定的优势，有效地解决了某些算法（AP 等）并不适用于非团状数据集的问题，并可以得到每个聚类的聚类中心代表点（DBSCAN 等无法得到），具有相当高的实用价值。

结束语 针对 AP 聚类算法无法正确地处理非团状数据集，提出了一个基于 AP 聚类算法的新的聚类 M-AP 算法。该方法将 merge 过程拓展至 AP 聚类算法中，解决了 AP 算法对非团状数据聚类效果不好的问题，而对团状数据仍有着较好的支持。对大规模数据的聚类，M-AP 算法可以先采用压缩算法对数据集进行压缩，然后再进行聚类，来取得很好的效果，从而为数据的聚类提供了一个可靠且有效的解决方案。

参 考 文 献

[1] Frey B J, Dueck D. Clustering by passing messages between data points[J]. Science, 2007, 315(5814):972-976

[2] Pollard D. Strong consistency of Kmeans clustering[J]. Ailnals of Statistics, 1981, 9(1):135-140

[3] Zhang T, Ramakrishnan R, Livny M. BIRCH. An efficient data clustering method for very large databases[J]. Montreal, 1996, 6(96):103-114

[4] Pal N R, Bezdek J C. On cluster validity for the fuzzy c-means model[J]. IEEE Transactions on Fuzzy Systems, 1995, 3(3):370-379

[5] Tsang I W, Kwok J T, Cheung P M. Core vector machines: fast SVM training on very large data sets[J]. Journal of Machine Learning Research, 2005, 8(6):363-392

[6] Deng Zhao-hong, Choi K S, Chung F L, et al. Enhanced soft subspace clustering integrating within cluster and between bluster-Information[J]. Pattern Recognition, 2010, 43(3):767-781

[7] Liu Jun, Mohammed J, Carter J, et al. Distance based clustering of CGH data[J]. Bioinformatics, 2006, 22(16):1971 -1978

[8] Chen W Y, Song Y, Bai H, et al. Parallel spectral clustering in distributed systems[J]. IEEE Transactions onPattern Analysis and Machine Intelligence, 2011, 33(3):568-586

[9] Wu M, Schölkopf B. A local learning approach for clustering[J]. Proc. Conf. Neural Information Processing Systems, 2007(1):1529-1536

[10] Papadimitriou C H, Steiglitz K. Combinatorial Optimization: Algorithms and Complexity[M]. Dover Publications, 1998

[11] Optdigits数据集[OL]. https://archive.ics.uci.edu/ml/machine-learning-databases/optdigits/

[12] Iris 数据集[OL]. http://archive.ics.uci.edu/ml/datasets/Iris

[13] Clean 数据集[OL]. http://archive.ics.uci.edu/ml/machine-learning-databases/musk/

[14] Soybean 数据集[OL]. http://archive.ics.uci.edu/ml/datasets/Soybean+(Small)

[15] 孙吉贵，刘杰，赵连宇. 聚类算法研究[J]. Journal of Software, 2008, 19(1):48-61

[16] 肖宇，于剑. 基于近邻传播算法的半监督聚类[J]. Journal of Software, 2008, 19(11):2803-2813

[17] 牟廉明，詹德川，黎铭，等. 基于结构相似性和压缩变换的聚类方法[J]. Pattern Recognition and Artificial Intelligence, 2011, 24(5):637-644

[18] 于剑，程乾生. 模糊聚类方法中的最佳聚类数的搜索范围[J]. 中国科学 E 辑，2002, 32(2):274-280

[19] 杨善林，李永森，胡笑旋，等. K-means 算法中的 k 值优化问题研究[J]. 系统工程理论与实践，2006, 26(2):97-101

[20] 周水庚，周傲英，曹晶. 基于数据分区的 DBSCAN 算法[J]. 计算机研究与发展，2000(10):1153-1159

[21] Ester M. A density-based algorithm for discovering clusters in large spatial databases with noise[C]// Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, Portland: AAAI Press. 1996:226-239

[22] 计华，张化祥，孙晓燕. 基于最近邻原则的半监督聚类算法 [J]. 计算机工程与设计，2011, 32(7):2455-2459

[23] 李昆仑，曹铮，曹丽苹. 半监督聚类的若干新进展[J]. 模式识别与人工智能，2009(5):735-742

[24] DBSCAN 算法代码[OL]. http://wenku.baidu.com/view/47a26ebbba0d4a7302763a9c.html