基于网页分割的语义信息检索研究

沈达峰
(淮阴工学院 现代教育中心,江苏 淮安 223003)

【摘 要】如何准确表达用户意图,判断网页与用户需求的相关性是信息检索技术研究的重要方向。本文提出了一种基于网页内容分割的语义信息检索算法。该算法根据网页半结构化的特点,按照HTML标记和网页的内容将网页进行区域分割。在建立HTML标记树的基础上,利用内容相似性和视觉相似性进行节点的整合。根据用户的查询,充分利用区域信息来对相关的检索结果进行排序。实验表明,本文提出的方法可以显著地提高搜索引擎的查询效果。

【关键词】网页分割;语义;信息检索;HTML标记;相似性
【中图分类号】TP391.3 【文献标识码】A 【文章编号】1673-1891(2009)04-0057-05

1 引言

在信息技术高速发展的今天,网络成了一个巨大的数据库,为我们提供了大量的信息,人们可以利用搜索引擎在网络中搜索需要的信息。许多搜索引擎,如Google,Yahoo等主要采用了关键词匹配的技术,该技术允许用户指定一些检索关键词序列,系统对包含关键词的网页进行检索,在对结果集排序后返回[1]。目前的网络信息搜索技术都有很明显的缺陷,查询时往往返回大量与用户查询无关的结果,原因在于其不能真正理解网页内容和用户的查询意图,典型的,一个网页的内容包括一些相关和不相关的主题,每个主题常常占据页面中的一部分。例如,一个教授的个人网页包括了作者的各方面的详细情况:第一节介绍他的研究工作,下一节列出他所讲授的课程,然后描述他的爱好,最后是他所提供的链接资源。如果用户的关键词序列出现在一个主题区域中,那么搜索引擎返回的结果往往是相关的。相反如果查询词分散在不同的主题区域中,那么查询结果的准确率会较低,即不相关的网页会位于检索结果的前列。尽管现有的搜索引擎在对搜索结果排序时考虑了一些关键词的距离因素(词频、权威度和枢纽度),但是它们不考虑这些词是否出现在不同的主题区域中。本文的研究工作主要是将网页进行区域分割引入信息检索,对用户需求和网页内容的语义化问题进行了研究,提出了一种基于网页分割的语义信息检索方法。

2 网页与用户查询的相关性

语义关联描述实体(包括类、属性、实例)之间语义关系的紧密程度,在本体知识库中,如果两个实体之间存在一条或多条属性序列,称这两个实体存在语义关联[2]。考虑语义关联,可以更有效地判断网页与实例的相关性。

综合实例间的语义关联,可以更准确地分析网页和实例的相关性。设与实例a语义关联大于γ的实体集合T(a)={$a_1,a_2,\cdots\cdots,a_n$},a∈T(a)。网页p与实例a的相关度[2]:

$$relationDegree(a,p)=\begin{cases}1, \sum_{i=1}^{n} relation(p,a_i)*\frac{R(a,a_i)}{R(a,a)}>1\\ \sum_{i=1}^{n} relation(p,a_i)*\frac{R(a,a_i)}{R(a,a)}, else\end{cases} \quad (1)$$

relation(p,ai)为实体ai和网页p的关键字向量相似度(当ai为实例时,以其文字说明为基本关键字,抽取3个类别和属性说明为关键字扩展;当ai为类时,直接以其文字说明为关键字)。

$$relation(p,a_i)=\frac{\sum_{j=1}^{m} td_j\times \mu tc_{ij}}{\sqrt{\sum_{j=1}^{m} td_j^2 * \sum_{j=1}^{m} \mu tc_{ij}^2}},1\leqslant i\leqslant n \quad (2)$$

m为网页文本和实体ai的关键字总个数,tdj和tcij分别为网页p和ai关键字值,μ为基本关键字和关键字扩展的权重,当tcij为基本关键字时,μ=2;当tcij为关键字扩展时,μ=1。网页p与实例a的相关度等于p与集合T(a)中的实例的关键字相似度之和,最大为1,ai与a的语义关联越强,relation(p,ai)对网页p与实例a相关度的影响越大。

3 网页分割技术

3.1 网页的预处理

网页是用HTML写的超文本文档,它包括纯文本、标记等。纯文本是不包含在标记中的字符串(不包含在括号<>中),它根据标记的定义体现出不同的颜色、字体和大小。标记定义了网页的显示属性。在一个Web文档中,大多数HTML标记是有开始和结束的HTML标记组成(<>表示开始标记,</>表示结束标记)。在一对HTML标记的内部可以嵌套HTML的标记。

收稿日期:2009-09-10
作者简介:沈达峰(1964-),男,实验师,主要从事网络管理与维护工作。

在网页的HTML代码嵌套结构的基础上，建立了HTML标记树[3]。树中的每个节点由标记名（如<head>与<table >）、内容-出现在一对HTML标记之间的纯文本、视觉属性-颜色（前景及背景）、字体、大小和三个指针-parent，child与sibling（分别表示双亲、孩子与兄弟指针）构成。

在标记树的设计中，对于任何一对HTML标记，它的子节点是它内部嵌套的第一个标记（child）。该子节点的sibling指针指向它的兄弟节点（与该节点在同一层的并列标记），注意只有叶节点包括“内容”部分（纯文本）。叶节点的祖先规定了叶节点的视觉表现形式，由于HTML文档的不规范，在非叶节点中出现的纯文本将被移到叶节点上。

3.2 合并标题和与其相邻的内容段

尽管标记树已经给了网页的初始分割（如各个叶节点），但是这个分割过于细小，需要合并树中的一些节点以形成一致性的网页分割。基于网页内容和HTML标记的节点的合并可以分为两步：一是合并标题和与其相邻的内容段，二是合并相邻的内容段。

（1）通过扫描所有叶节点的兄弟节点来发现可能的标题和内容段

令A，B分别是子树中的两个叶节点。Len表示节点A，B中内容部分的字符数；Neig（A，B）表示A，B之间的邻居关系：这里忽略了A与B之间的不含内容的节点。下列条件用于决定A，B为候选的标题和相邻的内容段。

$$(Len(A)<60)\wedge(Len(B)>Len(A)+30)\wedge Outstanding(A,B)\wedge Neig(A,B) \quad (3)$$

这里使用了长度、相邻关系、标题突出度等信息来判别是否A是B的可能的标题。其中Outstanding（A，B）用于判断A是否醒目。一般而言主要从五个因素上判断：A的字是否比B的大，A，B的颜色是否不同，A，B的字体是否不同，A是否是黑题以及A的周围是否存在空行等因素判断。一般在二个或二个以上的因素为真时，Outstanding（A，B）为真。

（2）布局评估

在（i）后，就已知道了可能的标题和内容段。下面的步骤运用布局特性进行进一步评估。对于（A，B），通过试图发现另一对可能的标题和内容段（C，D）并查看它们的视觉属性与（A，B）的相似性DisplaySim。DisplaySim主要判断两个节点视觉属性的相同个数。即，DisplaySim（x，y）=|x.features∩y.features|，（x，y是树中的两个节点）。这个属性集包括了字体、大小、前景色、背景色、标记名等，其中前四个属性的值可以为“default”。如果满足下列条件，合并A与B，同时合并C与D。

$$\exists(C,D),Neig(C,D)\wedge(DisplaySim(A,C)\geqslant\sigma)\wedge(DisplaySim(B,D)\geqslant\sigma) \quad (4)$$

该条件试图寻找与A，B并列的可能的标题内容对C，D。其中在视觉上A与C较为相象同时B与D较为相象，取σ=3。

如果满足条件（3）和（4），就合并节点A，B：将A的内容放入B中，同时将C的内容放入D。

3.3 合并具有一致性内容的相邻段落

令X和Y是两个相邻的内容段，可定义它们的一致性Sim（X，Y）为：

$$Sim(X,Y)=\lambda_1*Display\ Sim(X,Y)+\lambda_2*Content\ Sim(X,Y)+\lambda_3*Cap\ Rep(X,Y) \quad (5)$$

其中，$\lambda_1+\lambda_2+\lambda_3=1$，从三个方面来计算，即视觉一致性（DisplaySim）、内容一致性（ContentSim）和大写词重复度（CapRep）。

ContentSim和CapRep的定义如下：

首先采用数据采掘技术来发现词与词之间的语义关联性：即用支持度来过滤噪声，用可信度CONi，j来衡量关联的紧密度[2]。

对于给定的两个内容段X，Y，先进行停用词的消除和词干表示后，然后进行向量表示：$X=\{x_1,x_2,\cdots\cdots x_n\}$，$Y=\{y_1,y_2,\cdots\cdots,y_n\}$[4]。对于向量X中的任何一个分量xi（xi>0，对应特征关fi），在向量Y中查找具有语义关联的特征fj使得|xi－CONij*yjl最小。这样做的目的是：即使特征yi在段落Y中不出现，也可以以具有语义相关性的其它特征来代替，从而使得内容相似度的计算更为准确。所以ContentSim可以被定义为[4]：

$$ContentSim(X,X')=MAX\{d(X,X'),d(X',X)\} \quad (6)$$

其中，$d(X',X)=\sum_1^n Min|x_i-CON_{i,j}*x_j|$

CapRep用于衡量两个内容段的大写词重复。这些大写词包括人名、地名、组织名以及一些缩略词等成份，它们的重复在很大程度上说明两个内容段在叙述同一个主题。同时没有必要具体细分它们究竟属于哪个成份。

$$CapRep(X,Y)=\{word_i\mid First\_letter(word_i)=capital,\ word_i\in X.content,word_i\in Y.content\} \quad (7)$$

其中First_letter表示单词的一个字母；capital为大写字母。

在公式（5）中，权重λ_1，λ_2，λ_3被设定为$\lambda_1=0.2$，$\lambda_2=0.3$，$\lambda_3=0.5$。

如果Sim（X，Y）≥ω，则认为节点X和Y有较高

的视觉相似性和内容的一致性。此时节点X的内容放入Y中并将X删除。

以上分别定义了标题与内容段的合并、内容段之间的合并,注意如果所有的兄弟节点合并成一个节点,则该节点将其内容加入双亲节点并继续参与合并[5]。全面的合并进行到树中各层没有任何节点可以合并时为止(最多合并到<body>节点),算法如下:

for(treeDepth=maxDepth-1;treeDepth<1;treeDepth--)

Stree={subtreei| subtree$_i$ 为 treeDepth 层的叶节点};

while Stree 仍存在可以合并的节点时 do

for each subtree$_i$ ∈ Stree do

寻找可能的标题和内容段,合并 subtree$_i$ 中的标题和内容段

合并 subtree$_i$ 中相邻的内容段

endfor

如果 subtree$_i$ 仅包含一个节点,进行内容提升,即将其内容放入它的父节点,并将其删除。

endwhile

endfor

其中 maxDepth 是初始标记树的最大深度;treeDepth 表示合并过程中树的深度;Stree 表示深度为 treeDepth 的所有子树的集合;每一个子树 subtree 为仅包含叶节点的链表。

4 排序算法

对于任何一个输入网页 p 计算两个分值:一个是基准分;另一个是调节分。基准分是在该网页的区域中所包含的检索词的最大数目。令 seg$_i$ 是网页 p 的区域 i;queryTerms 是用户的检索词的集合。则网页 p 的基准分的 prScore(p)被定义为[6]:

$$prScore(p) = \arg\max_i(|seg_i \cap queryTerms|) \quad (8)$$

网页 p 的调节分考虑两个因素:一个是相邻互补性,相邻两个分割的是否可以包含更多的检索词;另一个因素是综合醒目度,即从标记权重、字的大小、颜色的醒目等方面进行衡量。令 seg$_i$ 是网页的第 i 个分割,Wi 是 seg$_i$ 的综合性目度。则调节分 seScore(p)可被定义为[7]:

$$seScore(p) = \arg\max_i(|seg_i \cup seg_{(i+1)}) \cap queryTerms| * W_i) \quad (9)$$

排序算法如下所示:

(1)创建一组集合 PageSet$_i$,i=1,2,…,n;

(2)pageSet$_i$=ϕ;

(3)For each page p ∈ AllPages do

(4)计算网页 p 的 prScore(p);

(5)计算网页 P 的 seScore(p);

(6)if prScore(p)=n then

(7)pageSet$_n$,=pageSet$_n$∪{p}

(8)if prScore(p)=i then

(9)pageSet$_i$,=pageSet$_i$∪{p}

(10)endfor

(11)首先按 PageSet$_n$,PageSet$_{n-1}$,…的顺序排列结果页。在每个 PageSet$_i$ 的内部,按 seScore 进行排序。如果 seScore 相同,则默认 Google 的顺序。

其中 n 是检索词的数目,Allpages 是基搜索引擎的待排序网页的集合。算法的前两行建立并初始化了一组集合,用于按基准分存放待排序的结果网页。在第六行中,如果 prScore(p)=n,表示网页 p 的某个分割包括了全部的检索词,所以该网页应该位于排序结果的前列(存储在 PageSetn)。如果 prScore(p)小于 n,按其大小存入相应的 PageSet,最后在 11 步对所有结果进行排序。本文选取的搜索引擎为 Google,排序算法每次选取的基搜索引擎搜索结果的数目作为用户指定的参数。

5 实验结果

在信息检索中,许多检索系统运用准确率和召回率来度量检索系统的性能。但是在 Web 检索中,搜索引擎返回的在检索结果前列的准确率是最为重要的因素。这是由于人们常常只看前 20~30 个结果。也就是说,即使搜索引擎有较高的召回率,但如果大多数搜索结果在前 20~30 个结果之后,那么它只有较小的机会被用户看到。所以一些研究人员认为高准确率是非常重要的即使以召回率为代价。在实验中,仅用位于检索结果前列网页的准确率来评价系统的性能。

在 Web 搜索中,由于检索词的选取问题导致排序有效性的判断成为一件困难的任务。在实验中,检索式有两个来源:一个是从站点 http://www.metacrawler.com 中挑选的二千个用户的检索(该站点允许用户察看别人提交给系统的检索词)。另一个是 TREC 的测试集合中随机选取的两千个标准测试标题。检索词挑选的准则是检索词应该是无歧义的和客观的,同时来自各个领域。许多检索词(特别是从 Metacrawler 上下载的,如“candy samples”)不被选择,因为不能确定用户的真正查询意图。

对于 Metacrawler 的 200 个检索,Google 及其系统检索结果的正确性判断依据是该结果页是否真正对用户有用。举例而言,检索词“free download music”将被认为是正确的仅仅当该网页确实存在歌

曲或音乐是真正免费的而不是该网页中存在这三个词(即使是连续出现)。每个结果的正确性至少要同时被三个人认可。

对于TREC测试集中的检索,严格按照每个检索规定的描述对所得的结果和Google的结果进行判别。对于每个检索,对Google的前200个结果进行重新排序。首先爬回这些网页,然后对它们进行预处理,最后利用排序算法对结果进行排序。

图1 Google的检索结果

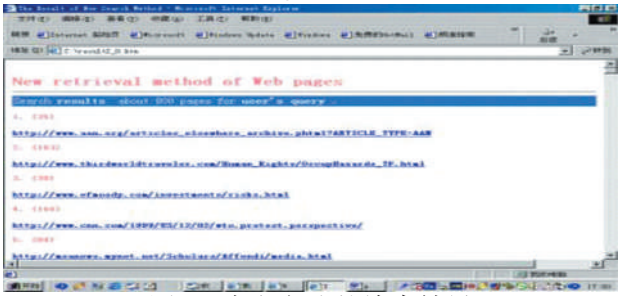

图2 本文方法的检索结果

图1、图2分别给出检索"Jounalist Risks"的Google的结果和本文的检索结果。经过判别Google的前二十个结果只有1个正确;而本文的结果有8个正确。

在表1、表2中分别列出了Google与本文的方法的前20个结果的比较。结果的第一列为检索词;二、三列分别为两种方法的准确率。通过对二、三列的比较,发现重新排序后前二十个结果的准确率有较大程度的提高。两个测试集合的十个查询结果平均准确率较Google分别提高26%与30%。

表1 Metacrawler的Goole与本文方法的准确率比较

Metacrawler的检索	准确率(本文的方法)	准确率(Google)
alternative music origins	0.70	0.40
Christmas island tour	0.60	0.40
decorative candlestick (for) sale	0.70	0.60
free download music	0.60	0.20
html tag tree	0.55	0.50
information history tomatoes	0.70	0.40
literary films list	0.60	0.20
red ladies t-shirt	0.50	0.40
Singapore programming jobs	0.70	0.20
supermodel success stories	0.70	0.50
平均准确率	0.64	0.38

表2 TREC的Google与本文方法的准确率比较

TREC的检索	准确率(本文的方法)	准确率(Google)
airbus subsidies	0.70	0.55
British Chunnel impact	0.50	0.25
computer aided crime	0.50	0.20
Dismantling Europe's arsenal	0.6	0.20
encryption equipment export	1.00	0.70
journalist risks	0.60	0.05
leveraged buyouts	0.45	0.10
most dangerous vehicles	0.55	0.35
new hydroelectric projects	0.60	0.20
transportation tunnel disasters	0.40	0.30
平均准确率	0.59	0.29

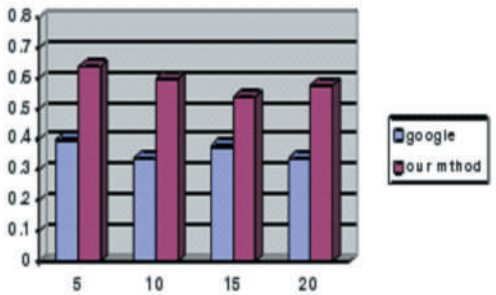

图3 Google与本文方法的平均准确率比较(Metacrawler)

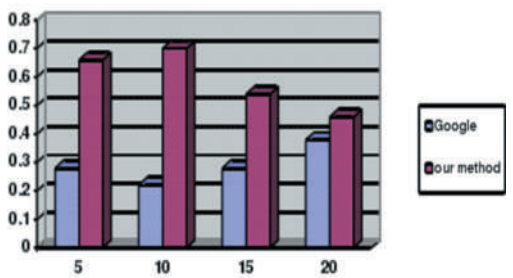

图4 Google与本文方法的平均正确率比较(TREC)

图3、图4分别给出了Google和我们的方法的所有查询式的二十个结果的平均准确率比较(每五个一组)。在两个测试集的测试中,结果显示我们的方法显著提高了Google的结果。特别是前五个结果平均从Google的0.34提高到我们的0.65,在实践中具有重要的意义。

6 结论

本文对用户需求和网页内容的语义化问题进行了研究,提出了一种基于网页分割的语义信息检索方法。该方法首先按照HTML标记和网页的内容对网页进行主题分割,准确理解网页内容和用户查询目的,在检索中,位于同一个区域或者相邻区域的网页优先用于匹配用户的检索需求,能有效提高检索的效率和准确率。此外,利用内容段的综合醒目度对网页进行进一步排序。这与当前搜索引擎中采用的技术不同。实验表明,本方法可以显著地提高传统搜索引擎的查询效果。

注释及参考文献:

[1]俞扬信.基于OWL-S服务匹配的信息查询模型[J].计算机与应用化学,2007,24(9):1277-1280.

[2]俞扬信.基于知识推理的语义信息检索研究[J].情报杂志,2008,27(11):78-80.

[3]宋玲玲,李村合.基于链接结构分析的Web信息检索方法研究[J].现代情报,2007,27(02):133-135,

[4]朱征宇,苑昆峰,陈杏环.一种基于最大权匹配计算的信息检索方法[J].计算机工程与应用,2007,43(33):176-179.

[5]Park,J.S.,Chen,M.-S.,and Yu,P.S.1995. An effective hashbased algorithm for mining association rules. In Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 175-186, San Jose, CA.

[6]刘亚军,徐易.一种基于加权语义相似度模型的自动问答系统[J].东南大学学报(自然科学版), 2004,34(5):609-612.

Semantic Information Retrieval Study Based on Page Segmentation

SHEN Da-feng

(Modern Education Technology Center, Huaiyin Institute of Technology, Huai'an, Jiangsu 223003)

Abstract: There is an important research direction of information retrieval technology for accurately judging the relations between the web pages and the user's requirement. In this paper, a semantic information retrieval algorithm based on web page segment is proposed. The key idea is to segment each web page into different topic areas or segments according to its HTML tags and contents since web pages are semi-structure. First the algorithm builds a HTML tag tree. Then it combines nodes in the tree by using both the content similarity and visual similarity. The retrieval and ranking algorithm makes use of this segmentation information to search and order the relevant pages. Experiment results show that this method is able to improve the search precision significantly.

Key words: Page segment; Semantic; Information retrieval; HTML tag; Similarity