

一种新的复合核函数及在问句检索中的应用

王君^{*①} 李舟军^① 胡侠^② 胡必云^①

^①(北京航空航天大学计算机学院 北京 100191)

^②(新加坡国立大学计算机学院 新加坡 117590)

摘要: 问句检索在问答系统中有着重要的作用, 其核心问题在于研究查询问句与候选问句之间的相似性计算问题, 实现问句之间的高精度匹配。该文采用树核函数的方法计算问句之间的结构相似性, 并针对原有算法的不足, 做了相应的改进。为降低句法解析器性能对树核函数的影响, 该文在改进的树核函数基础上, 将其与字符串核结合, 提出了一种能同时融合问句的句法信息, 词性信息和词序信息的复合核函数, 用以计算问句之间的综合语义相似性。在社区问答系统 Yahoo!Answer 的数据上进行测试, 相对传统的基于词频的特征向量法, 问句检索平均准确率提高了 24.02%。

关键词: 信息检索; 问答系统; 问句检索; 复合核函数

中图分类号: TP391

文献标识码: A

文章编号: 1009-5896(2011)01-0129-07

DOI: 10.3724/SP.J.1146.2010.00268

A Novel Composite Kernel and Application to Question Retrieval

Wang Jun^① Li Zhou-jun^① Hu Xia^② Hu Bi-yun^①

^①(School of Computer Science and Engineering, Beihang University, Beijing 100191, China)

^②(School of Computing, National University of Singapore, Singapore 117590)

Abstract: Question retrieval plays important role in question and answering systems. The main problem is how to measure the similarity between candidate questions and query question. This paper presents a tree kernel based method, named weighted tree kernel, to calculate the similarity of sentences' structures and proposes improvements to the original tree kernel algorithm. In order to reduce the effect on tree kernel bringing by syntactic parsing, a composite kernel is proposed based on the weighted tree kernel and two other string kernels, which can capture syntax, part-of-speech and lexical level information of a sentence, to calculate the semantic similarity between question sentences. Experimental results on Yahoo!Answers dataset show that the proposed method outperforms traditional vector space model based methods by 24.02% in question retrieval accuracy.

Key words: Information retrieval; Question answering system; Question retrieval; Composite kernel

1 引言

问答系统是一种自然语言检索, 也称作问答式信息检索。它既能够让用户用自然语言句子提问, 又能够为用户返回一个简洁、准确的答案, 而不是一些相关的网页。考虑到自然语言理解技术的难度和鲁棒性, 问答系统从研究初期就提出并开展了基于问答对的技术路线的研究, 即从问答对库中检索出于用户问题最为相似的问答对并把答案部分直接反馈给用户的技术路线。(Frequent Ask Questions, FAQ) 页面是早期获取问答对的主要来源。

FAQFinder^[1]是第 1 个此技术路线下实现的较大规模的自动问答系统。近年来, 随着社区问答系统 (Community Question Answering, CQA) 的急速增长, 如百度知道、雅虎知识堂、新浪爱问等, 这些网站聚集了千万级的可直接下载的问答对, 因此更多的研究开始转向以这些问答对作为语料库。基于问答对的问答系统, 即从问答对库中搜索出与用户问题最为相似的已回答问题, 并把该相似问答对的答案部分反馈给用户, 其核心问题是研究查询问句与候选问句之间的相似性计算问题, 实现问句之间的高精度匹配。

目前针对问句的大多数相似性计算, 相关的研究有: 文献[1,2]提出向量空间模型, 计算查询问句向量和候选问句向量的夹角余弦。文献[3,4]提出将

2010-03-23 收到, 2010-07-05 改回

国家 973 规划项目 (2007CB310803) 资助课题

*通信作者: 王君 wangjun0706149@cse.buaa.edu.cn

语言模型应用到社区问答系统问句检索中;文献[5]提出了基于翻译模型的问答系统检索模型。以上这些方法以特征向量为处理对象,难以表示结构化的特征,存在数据稀疏的问题。

针对上述问题,文献[6]使用树核^[7]对结构化特征进行建模并取得了不错的效果。文献[6]使用问句的句法树(syntactic parsing tree),简称为句法树,表示问句的结构特征。在一棵句法树中,树中节点的深度越深,如叶子节点,则该节点表达的信息越具体,包含的信息越多;深度越浅,如根节点,则该节点表达的信息越抽象,包含的信息越少。此外,对于一个句子,根据语言学知识,通常有主要成分(如主语、谓语、宾语等),和次要成分(如定语、状语、补语等)构成,不同的成分对于表达一个句子的语义起着不同的作用,因此在比较两个句子的相似程度时应予以区别对待。文献[7]提出的树核,通过计算两棵句法树之间的相同子树的数量来比较句法树之间的相似程度,没有区别节点的深度特征和句法成分特征。为此,本文在文献[7]基础上,在核函数的设计中,做了进一步的改进,针对句法树节点的成分特征和深度特征,引入加权机制,提出一种加权树核,并在问句检索问题中取得了预期的效果。

基于问答对的问答系统中问句检索所面向的处理对象是相对简短的问句,问句通常包含较少的词,因此,要使检索性能得以提高,就需要从简短的问句中尽可能多地提取对检索有帮助的信息。但是,自然语言处理中一个不可避免的问题是,随着处理层次的深入,处理结果的准确率越低。以英语的分词、分块和句子解析为例,其准确率分别是99%,92%和90%^[8]。为充分利用问答系统中问句的各种特征,同时降低句法分析精度对问句检索性能的影响,本文提出一种以加权树核和字符串核为基础的复合核函数,通过融合问句的结构特征,词性特征和词序特征,进一步提高问句检索的性能。该方法不需要构造高维特征向量,直接计算离散对象之间的相似程度。理论上,可探索隐含的高维特征空间,易于实现对新的特征提取以及与新的核函数的组合,具有良好的扩展性和适应性。在社区问答系统Yahoo!Answer的问答对测试数据上的实验表明,与传统的基于词频的特征向量法相比,本文提出的复合核函数法,显著提高了问答系统中问句搜索的性能。

2 基于核函数的问句相似性度量方法

2.1 树核函数简介

树核(tree kernel)是由文献[7]提出的,通过计算两句法树之间的相同树片段的数量来比较句法树之

间的相似程度。为了获得句子中的语法结构信息,文献[7]将句法树中的所有树片段(Syntactic Tree Fragments, STF)作为特征空间。每个树片段(STF)是句法树的一部分,至少包含一条语法产生式,并且要保证每条产生式的完整性。以问句“What is an atom”为例,图1(a)表示了该问句的句法树,图1(b)表示了句法树(a)的一棵子树及其产生式,图1(c)列出了子树(b)包含的所有STFs。

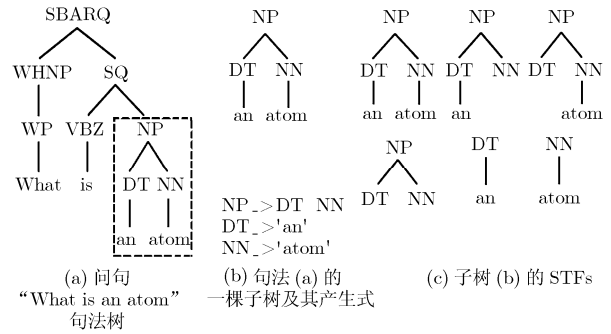


图1 句法树、子树及其树片段

两棵句法树 T_1, T_2 的树核函数定义为

$$k(T_1, T_2) = \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} C(n_1, n_2) \quad (1)$$

其中 N_1, N_2 分别是两棵树 T_1, T_2 中节点的集合, $C(n_1, n_2)$ 表示分别以 n_1, n_2 为根节点的子树中相同的树片段的个数, 计算方法如下:

$$C(n_1, n_2) = \begin{cases} 0, & n_1 \neq n_2 \\ 1, & n_1 = n_2 \text{ 并且 } n_1, n_2 \text{ 是叶子节点} \\ \lambda, & n_1 = n_2 \text{ 并且 } n_1, n_2 \text{ 是叶子节点} \\ & \text{的直接父节点} \\ \lambda \prod_{j=1}^{nc(n_1)} [1 + C(\text{ch}(n_1, j), \text{ch}(n_2, j))], & \text{其它} \end{cases} \quad (2)$$

其中 $nc(n)$ 表示节点 n 的子节点的个数, $\text{ch}(n, j)$ 表示节点 n 的第 j 个子节点, $n_1 = n_2$ 表示节点 n_1, n_2 的标签和产生式都相同, λ 是一个权值参数。

2.2 加权树核函数

文献[7]中提出的树核函数是针对语义标注这一问题提出的,没有考虑问题系统中问句的特点,如果直接使用,进行问句检索可能并不合适。此外,文献[7]在树核的定义中假定,每个树片段对句法树的贡献是相同的,没有区分问句中主要成分和次要成分的区别。本文针对这一问题提出了一种改进的加权树核函数,并将其用于比较问句的句法树相似性。

在一棵句法树中，树中节点的深度越深，如叶子节点，则该节点表达的信息越具体，包含的信息越多；深度越浅，如根节点，则该节点表达的信息越抽象，包含的信息越少。

定义 1 节点的深度：令 j 表示句法树 T 中的一个非叶子节点， dep_j 表示节点 j 在 T 中的深度，其值等于 i 在 T 中所在的层次，其中 $\text{dep}_{\text{root}}=0$, root 是 T 的根节点。

如图 1(c)所示，这 6 棵 STF 在图 1(a)所示的句法树中的深度分别是：2, 2, 2, 2, 3, 3(按从左到右，从上到下顺序)。

根据语言学知识可知，任何句子都是由关键成分(主语、谓语、宾语等)和修饰成分(定语、状语、补语等)构成的。关键成分对句子起了主要作用，修饰成分对句子起了次要作用。在一棵句法树中，不同的节点代表不同的句子成分。在通常情况下，一个句子中作为主语和宾语的多数为名词或代词，作为谓语的多为动词。疑问词在问题检索中也有着重要的作用。本文用节点的权重来表示节点在一棵句法树中的重要性。

定义 2 节点的权重：令 j 表示句法树 T 中的一个非叶子节点， δ_j 表示节点 j 在 T 中的权重， $\text{label}(j)$ 表示节点 j 的标签，如 WP, VP, NN 等，节点 j 的权重等于：

- (1) $\delta_j = \nu_Q$, $\text{label}(j) = W^*$ (表示任意字母);
- (2) $\delta_j = \nu_{NV}$, $\text{label}(j) = N^*$ 或 $\text{lable}(j) = V^*$;
- (3) $\delta_j = 0.1$, 其它。

其中 ν_Q 表示和疑问词相关的节点的权重， ν_{NV} 表示和名词或代词相关的节点的权重。

根据节点的深度和节点的权重的定义，定义 3 为 STF 的权重定义。

定义 3 STF 的权重：令 i 表示句法树 T 中一个 STF， ϖ_i 表示 i 的权重，则

$$\varpi_i = \prod_{j=1}^{s(i)} \sqrt{\delta_j} \sqrt{\mu}^{d(i)} \quad (3)$$

其中 $d(i)$ 表示 i 的根节点的深度， μ 是一个常量表示 i 的影响因子， $s(i)$ 表示 i 中包含的非叶子节点的个数， δ_j 表示每个非叶子节点的权重。

将要处理的数据映射到一个 m 维空间中，令每棵句法树 T 用一个 m 维向量表示， $\mathbf{V}(T)=(v_1(T), v_2(T), \dots, v_m(T))$ ，其中第 i 个分量表示在 m 维空间第 i 个 STF 在 T 中权重。加权树核函数定义为

$$\text{WTK}(T_1, T_2) = V(T_1) \cdot V(T_2) \quad (4)$$

由于 m 是一个很大的值，并且不容易求得具体值，用下面的方法计算 $\text{WTK}(T_1, T_2)$ 的值。

令 N 表示句法树 T 中所有非叶子节点的集合， n 是 N 中的一个节点， i 是 m 维空间中第 i 个 STF，定义 $I_i(n)$ 为这样的一个指示函数，

$$I_i(n) = \begin{cases} 1, & \text{以节点 } n \text{ 为根节点} \\ 0, & \text{其它} \end{cases} \quad (5)$$

可推导出如下等式：

$$v_i(T) = \sum_{n \in N} \prod_{j=1}^{s(i)} \sqrt{\delta_j} \sqrt{\mu}^{d(i)} I_i(n) = \sum_{n \in N} \prod_{j=1}^{s(i)} \sqrt{\delta_j} \sqrt{\mu}^{d(n)} I_i(n) \quad (6)$$

则加权树核函数 $\text{WTK}(T_1, T_2)$ 等于

$$\begin{aligned} \text{WTK}(T_1, T_2) &= V(T_1) \cdot V(T_2) = \sum_i v_i(T_1) v_i(T_2) \\ &= \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \mu^{\frac{d(n_1)+d(n_2)}{2}} \Delta(n_1, n_2) \end{aligned} \quad (7)$$

其中

$$\Delta(n_1, n_2) = \sum_i \prod_{j=1}^{s(i)} \delta_j I_i(n_1) I_i(n_2) \quad (8)$$

计算方法为

$$\Delta(n_1, n_2) = \begin{cases} 0, & n_1 \neq n_2 \\ \delta_{n_1}, & n_1 = n_2 \text{ 并且 } n_1, n_2 \text{ 是叶子节点} \\ & \text{的直接父节点} \\ \delta_{n_1} \prod_{j=1}^{\text{nc}(n_1)} [1 + \Delta(\text{ch}(n_1, j), \text{ch}(n_2, j))], & n_1 = n_2 \text{ 且非叶子节点或叶子直接父节点} \end{cases} \quad (9)$$

其中 $\text{nc}(n)$ 表示节点 n 的子节点个数。如果 n_1 和 n_2 节点处有相同产生式则 $\text{nc}(n_1)=\text{nc}(n_2)$, $\text{ch}(n, j)$ 表示节点 n 的第 j 个子节点。

2.3 字符串核和复合核

树核能有效地挖掘句子中的结构化信息，但是树核只捕获了句子的语法信息，而对于计算句子相似度有用的词序、词性等信息需要另外获取。本文用字符串核来挖掘句子中的词序列和词性序列信息。

字符串核函数的思想是通过比较两个字符串共同包含的子串个数和连续程度来衡量两个字符串的相似程度。共同的子串越多，两个字符串就越相似。这里的子串不一定是连续的，但是它的连续程度被用来作为衡量相似度的一个指标。

字符串核函数的形式化定义如文献[9]。

定义 4 设 Σ 是一个有限字符集合， $S=S_1, S_2, \dots, S_{|S|}$ 是 Σ 上的一个字母序列，其中 $S_i \in \Sigma$, $1 \leq i \leq |S|$ 。设 $i = [i_1, i_2, \dots, i_n]$ ，且 $1 \leq i_1 \leq i_2 < \dots <$

$i_n \leq |s|$ 是 s 的下标的子集。则用 $s[i] \in \Sigma^n$ 来表示字母序列 $S_{i_1}, S_{i_2}, \dots, S_{i_n}$ 。这里的 $S[i]$ 并不一定 S 的连续子序列。例如: $s = \text{CART}$, $i = [2, 4]$, 那么, $s[i] = \text{AT}$ 。令 $l(i)$ 表示 $s[i]$ 在 s 中的跨度, 那么: $l(i) = i_n - i_1 + 1$ 。则在 Σ 上的两个字符串 s 和 t 的字符串核表示为

$$K_n(s, t) = \sum_{u \in \Sigma^n} \sum_{i: s[i]=u} \sum_{j: t[j]=u} \lambda^{l(i)+l(j)} \quad (10)$$

其中 $\lambda \in [0, 1]$ 是用来体现不连续情况下的衰减因子。式(10)中第1个求和符号代表所有长度为 n 的字符串序列。目前, 计算字符串核的最快的算法是基于 Suffix Tree 或 Suffix Array 的方法, 可以成功地把复杂度降为线性的, 具体参见文献[10]。

由于字符串核函数的计算量依赖于子序列的长度而不是符号集的长度, 因此文献[11]提出了词序列核(word sequence kernel)。本文将要处理的问句表示成词序列(word sequence)和词性序列(part-of-speech sequence), 采用文献[11]提出的词序列核分别设计了用于计算问答系统中问句相似性的词序列核(Word Kernel, 简记为 WK)和词性序列核(Part-of-speech Kernel, 简记为 PK)。

根据 2.2 节已定义的基于权值的树核 WTK 和基于字符串核的词性序列核 PK 和词性序列核 WK, 定义如下复合核函数:

$$\text{CK}(s_1, s_2) = \alpha_1 \text{WTK}(\text{tree}(s_1), \text{tree}(s_2)) + \alpha_2 \text{PK}(\text{pos}(s_1), \text{pos}(s_2)) + \alpha_3 \text{WK}(\text{word}(s_1), \text{word}(s_2)) \quad (11)$$

其中 s_i , $i=1, 2$ 表示一个问句, $\text{tree}(s_i)$ 表示 s_i 的句法树, $\text{pos}(s_i)$ 表示 s_i 的词性序列, $\text{word}(s_i)$ 表示 s_i 的词序列。 α_1 , α_2 和 α_3 分别是核函数 WTK, PK 和 WK 的参数, 在本文实验中, α_1 , α_2 和 α_3 的值分别为 0.4, 0.3, 0.3。依照文献[12], 核函数集合在线性合并运算下是封闭的, 由于 WTK, PK, WK 是合法的核函数(前面已证明), 因此复合核 CK 也是合法的核函数。问句 s_1 , s_2 的相似度定义为

$$\text{sim}(s_1, s_2) = \frac{\text{CK}(s_1, s_2)}{\sqrt{\text{CK}(s_1, s_1)\text{CK}(s_2, s_2)}} \quad (12)$$

3 实验结果与分析

3.1 数据集的建立和评价标准

为了测试本文提出的方法的有效性, 本文在 UIUC 问句测试集¹⁾和从 Yahoo!Answer 下载的问答对数据集上进行了测试。UIUC 数据集中包含了 500 条 TREC_10 中的问句, 对于其中每个问句

$q_i (i=0, 1, \dots, 499)$, 用 Yahoo!Answer 提供的问句搜索接口 questionSearch²⁾进行相关问句检索。例如问句“How far is it from Denver to Aspen?(从丹佛到奥斯本有多远?)”, Yahoo!Answer 的 question Search 返回的前 5 个结果如表 1 所示。

表 1 Yahoo!Answer 的 question Search 返回的结果

-
- (1) How do i apply my makeup and mascara correctly if this is my first time wearing makeup?(第1次化妆中怎样正确的使用粉底和睫毛膏?)
 - (2) How can I prepare to be a running back for sophomore football?(要想做好一个第2年的橄榄球的跑卫, 我应该怎么准备?)
 - (3) How to set up a macro that will open several different programs at a time?(如何安装一个宏, 能同时打开几个不同的程序?)
 - (4) How do single camera products and the way they are produced affect the film industry?(单镜头产品和单镜头拍摄是怎样影响电影工业的?)
 - (5) How big is Chateau Chenonceau and how many rooms are there?(雪浓梭堡有多大, 那有多少间房子?)
-

从上例看出, 虽然 Yahoo!Answer 提供了问句问句搜索接口, 但是检索结果和查询语句的相关性很低, 因此需要进一步的挖掘。本文按 Yahoo!Answer 提供的查询结果, 将前 1000 条记录作为该 q_i 的候选相似问句集 C_i 。令 $\text{rel}(q_i, C_{ij})$ 表示查询问句 q_i 和候选相似问句集中的问句 $C_{ij} (j=0, 1, \dots, 999)$ 的相似度, 值域为 $\{0, 1\}$ 。我们首先采用基于传统的向量空间模型的向量余弦方法, 进行相似度评分, 相似度的值记为 $\text{sim}_{\cos}(q_i, C_{ij})$, 如果 $\text{sim}_{\cos}(q_i, C_{ij}) > 0.5$, 则 $\text{rel}(q_i, C_{ij})=1$, 否则 $\text{rel}(q_i, C_{ij})=0$ 。在此基础上, 再采用人工方法, 对自动判断结果进行确认和更正, 并将人工判断的结果作为本文实验的标准测试集, 记为 Cdataset。

对于每个查询问句 q_i 和它的候选相似性问句集 C_i , 采用不同的相似性度量方法, 对候选相似性问句集中的问句进行相似度判断, 并根据相似度的值按从高到低的顺序进行排序, 采用 MRR(Mean Reciprocal Rank), Precision@n 和 MAP(Mean Average of Precision)3 种评价标准对所采用的相似性度量方法进行评价。MRR, Precision@n 和 MAP 的计算方法分别如下:

(1)MRR:

$$\text{MRR} = \frac{1}{|Q_r|} \sum_{q \in Q_r} \frac{1}{r_q} \quad (13)$$

¹⁾ http://l2r.cs.nyu.edu:80/cogcomp/Data/QA/QC/TREC_10.label

²⁾ <http://developer.yahoo.com/answers/V1/questionSearch.html>

其中 Q_r 表示查询测问句试集, r_q 是第一个相关问句的顺序。

(2) Precision@n:

$$\text{Precision@n} = \frac{\sum_{j=1}^n \text{rel}(j)}{n} \quad (14)$$

其中 $\text{rel}(j)$ 表示第 j 个候选问句和查询问句是否相关, 值域为 $\{0,1\}$ 。Precision@n 表示前 n 个候选相似问句中相关的问句的个数所占的比例。

(3) MAP:

$$\text{MAP} = \frac{1}{|Q_r|} \sum_{q \in Q_r} \frac{\sum_{r=1}^n P(r) \times \text{rel}(r)}{|R_q|} \quad (15)$$

其中 Q_r 表示查询问句集, R_q 表示和查询问句相关的问句, r 是其排列次序, N 是检索的问句的个数, $\text{rel}(r)$ 表示第 r 个候选问句和查询问句是否相关, 值域为 $\{0,1\}$, $P(r)$ 表示前 r 个检索的问句的相关问句所占比例。

3.2 加权树核(WTK)参数设置

本文提出的树核函数 WTK 与文献[7]中提出的树核 TK 的区别是引入了深度影响因子 μ , 节点权值影响因子 ν_Q 和 ν_{NV} , 当 $\mu = 1$, $\nu_Q = \nu_{NV} = 0.1$ 时, WTK 等价于 TK。本节分析比较了上述 3 个因子对问句搜索性能的影响, 其中句法解析器采用 stanford parser³⁾。

(1) 权重因子 ν_Q 和 ν_{NV} 图 2(a) 显示了在 UIUC 数据集和 Cdatasets 数据集上进行相似问句搜索时, $\mu = 1$, $\nu_{NV} = 0.1$, ν_Q 取值从 0.05-0.19 时所对应的 MAP 值。该实验测试了和疑问词相关的节点的权重因子 ν_Q 对实验性能的影响。图中, 横坐标表示 ν_Q 的取值, 曲线 TK 表示采用未改进的树核作为相似性度量标准时所对应的 MAP 值, 曲线 TK+ ν_Q 表示采用本文提出的对和疑问词相关节点进行权值计算的树核函数作为相似性度量标准时所对应的 MAP 值。图 2(a) 显示, 随着 ν_Q 取值的增加, 相应的 MAP 值逐渐下降。当 $\nu_Q \in \{0.05, 0.07\}$ 时, MAP 取得最大值。图 2(a) 说明, 当和疑问词相关的节点的权重小于其它节点的权重时, 问句检索的性能得到提高。出现这一现象的原因可能是因为检索的数据中每个(或大多数)候选问句中都包含有查询问句中出现的疑问词, 因此降低了疑问词这一特征的分类别能力。因此, 降低和疑问词相关的节点的权重, 对于提高检索性能是有用的。

图 2(b) 显示了在 UIUC 数据集和 Cdatasets 数据集上进行相似问句搜索时, $\mu = 1$, $\nu_Q = 0.1$, ν_{NV} 取值从 0.05-0.19 时所对应的 MAP 值。该实验测试

了和名词或动词相关的节点的权重因子 ν_{NV} 对实验性能的影响。图中, 横坐标表示 ν_{NV} 的取值, 曲线 TK 表示采用未改进的树核作为相似性度量标准时所对应的 MAP 值, 曲线 TK+ ν_{NV} 表示采用本文提出的对和名词或动词相关节点进行权值计算的树核函数作为相似性度量标准时所对应的 MAP 值。图 2(b) 显示, 随着 ν_Q 取值的增加, 相应的 MAP 值逐渐下降。当 $\nu_{NV} = 0.13$ 时, MAP 取得最大值。当 $\nu_{NV} \leq 0.11$ 或 $\nu_{NV} \geq 0.16$ 时, TK+ ν_{NV} 曲线所示的树核对应的 MAP 值小于曲线 TK 所示的树核对应的 MAP 值。实验结果说明适当地增加和主要成分(如名词、动词)相关的节点的权重有助于提高问句检索的性能。

(2) 深度影响因子 μ 对深度影响因子 μ , 进行了类似实验, 实验表明, 当 $\mu = 0.9$ 时, 本文提出的树核(不考虑节点权重因子)取得最好的实验结果。

表 2 中, TK 表示没有改进的树核, TK+ μ 表示带有深度影响因子的树核。Impr. 表示改进率。该表列出了两种树核分别取得 Precision@10, MRR 和 MAP 的值。和没有改进的树核相比, 改进的树核在 3 种评价标准下分别提高了 21.56%, 3.48% 和 2.45%。这说明句法树中树片段的深度及其影响因子在计算问句相似性上是有用的, 同树核相比, 带有深度影响因子的树核在捕获句子的结构信息上更加有效。

表 2 深度影响因子对实验性能的影响

核函数	Precision@10(Impr.)	MRR(Impr.)	MAP(Impr.)
TK	0.285(N.A)	0.383(N.A)	0.364(N.A)
TK+ μ	0.346(+21.56%)	0.396(+3.48%)	0.372(+2.45%)

3.3 复合核函数性能评价

为了测试本文提出的复合核函数在问句搜索上的性能, 本文分别采用 7 种独立方法进行比较。表 3 列出了这 7 种方法的名称和描述, 其中带星号(*)的为本文提出方法。每种方法参数设置如下: TK_{tree} 中 $\mu = 0.9$, WTK_{tree} 中 $\mu = 0.9, \nu_Q = 0.05, \nu_{NV} = 0.13$, WK_{word} 中 $\lambda = 0.9, n = 1$, PK_{POS} 中 $\lambda = 0.9, n = 3$, CK_{word+POS+tree} 中 $\alpha_1 = 0.4, \alpha_2 = 0.4, \alpha_3 = 0.4$, 其它方法均采用默认参数设置。表 4 列出了实验结果, 表中括号里的数值是相对于 VSM_{BOW} 的相对提高幅度。

表 4 说明:

(1) 本文提出的加权树核 WTK_{tree} 在问句搜索性能上优于没有实现加权机制的树核 TK_{tree}。其 TK_{tree} Precision@10, MRR, MAP 相比分别提高 39.65%,

³⁾<http://nlp.stanford.edu/software/lex-parser.shtml>

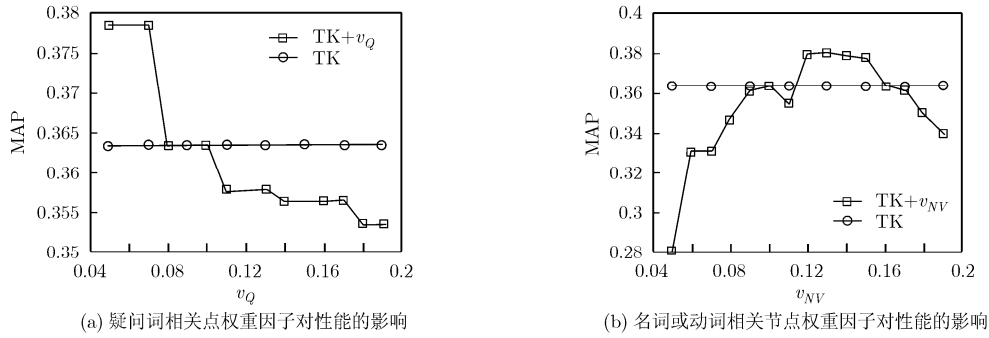


图2 MAP 与 v_Q, v_{NV} 的关系

表3 实验方法和描述

方法名称	方法描述
VSM _{BOW}	基于词袋(bag-of-words)表示的空间向量模型法
LM _{BOW}	基于词袋(bag-of-words)表示的语言模型法
TK _{tree}	基于语法树的树核函数法
WTK _{tree} *	基于语法树的加权树核函数法(本文提出方法)
WK _{word}	基于词序列的词序列核函数法
PK _{POS}	基于词性序列的词序列核函数法
CK _{word+POS+tree} *	基于词序列, 词性序列和语法树的复合核函数法(本文提出方法)

2.87%和 9.07%。这说明本文提出的加权机制是有效的, 在捕获句子的结构信息上, WTK_{tree} 比 TK_{tree} 更加有效。

(2)基于树核的方法 TK_{tree} 和 WTK_{tree} 总体评价上性能略低于 VSM_{BOW} 和 LM_{BOW}, 出现这一现象的原因之一可能是在句法解析过程中错误的解析造成的。由于本文中直接使用了 Stanford Parser 用于进行问句的句法解析, 由于该解析器是离线解析器, 训练集不是专门针对问答系统的问句, 因此, 降低了其解析准确率, 从而影响了基于句法树的树核方法在问句检索中的性能。

(3)词序列核 WK_{word} 和词性序列核 PK_{POS} 性能好于基于词袋表示的空间向量模型 VSM_{BOW} 方法和语言模型 LM_{BOW} 方法; 说明基于字符串的核在计算问句相似性上是有效的。

(4)本文提出的复合核 CK_{word+POS+tree} 性能上明显好于上述几种方法, 说明混合的 3 种核相互补充, 充分利用了句子的词序, 词性序列, 和句子结构信息。

4 结论

本文在原有树核的基础上引入了加权机制, 提出了一种加权树核函数, 区分不同成分节点在句子中的重要性, 从而能更加有效捕获句子的句法结构信息。此外, 在加权树核, 词序列核和词性序列核的基础上, 本文还提出了一种复合核, 利用词序, 词性等简单特征与句子结构特征的融合, 降低句法解析器的性能对检索性能的影响。实验表明, 复合核能充分利用句子的词序、词性、和句法信息, 在计算句子相似度, 用于进行基于问答系统的问句搜索任务中, 检索性能取得了明显改进。

本文中提到的检索性能是针对检索准确率而言的, 对于检索时间效率没有考虑, 如何在提高检索性能的同时提高时间效率是本文今后进一步研究方向。

表4 实验结果

方法	Precision@10(Impr.)	MRR(Impr.)	MAP(Impr.)
VSM _{BOW}	0.268(N.A)	0.439(N.A.)	0.403(N.A.)
LM _{BOW}	0.330(+23.12%)	0.472(+7.64%)	0.445(+10.37%)
TK _{tree}	0.285(+6.20%)	0.383(-12.72%)	0.364(-9.78%)
WTK _{tree} *	0.398(+48.51%)	0.394(-10.17%)	0.397(-1.36%)
WK _{word}	0.359(+34.33%)	0.4889(+11.21%)	0.465(+15.45%)
PK _{POS}	0.350(+30.56%)	0.5189(+18.22%)	0.483(+19.76%)
CK _{word+POS+tree} *	0.400(+49.25%)	0.506(+15.35%)	0.499(+24.02%)

参 考 文 献

- [1] Burke R D, Hammond K J, and Kulyukin V A, *et al.* Question answering from frequently asked question files: experiments with the faq finder system[J]. *AI Magazine*, 1997, 18(2): 57-66.
- [2] Jijkoun V and De Rijke M. Retrieving answers from frequently asked questions pages on the web [C]. In CIKM'05: Proceedings of the 14th ACM international conference on Information and knowledge management, Bremen, Germany, 2005: 84-90.
- [3] Cao Xin, Cong Gao, and Cui Bin, *et al.* The use of categorization information in language models for question retrieval [C]. In CIKM'09: Proceeding of the 18th ACM conference on Information and knowledge management, Hong Kong, China, 2009: 256-274.
- [4] Duan Hui-zhong, Cao Yun-bo, and Lin Chin-yew, *et al.* Searching questions by identifying questions topic and question focus [C]. In ACL-08: HLT: Proceeding of the 46th annual meeting of the association for computational linguistics: Human Language Technologies, Columbus, OH, USA, 2008: 156-164.
- [5] Xue Xiao-bing, Jeon J, and Croft W B. Retrieval models for question and answer archives [C]. In SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, 2008: 475-482.
- [6] Wang Kai, Ming Zhao-yan, and Chua Tat-seng. A syntactic tree matching approach to finding similar questions in community-based QA services [C]. In SIGIR'09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, Boston, MA, USA, 2009: 187-194.
- [7] Collins M and Duffy N. Convolution Kernels for Natural Language [M]. *Advances in Neural Information Processing Systems 14*, MIT press, 2001: 625-632.
- [8] Zhao Shu-bin and Grishman R. Extracting relations with integrated information using kernel methods [C]. In ACL'05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Ann Arbor, Michigan, USA, 2005: 419-426.
- [9] Lodhi H, Saunders G, and Shawe-Taylor J, *et al.* Text classification using string kernels [J]. *Journal of Machine Learning Research*, 2002, 2(Feb): 419-444.
- [10] Choon Hui-teo and Vishwanathan S V N. Fast and space efficient string kernels using suffix arrays [C]. In ICML'06: Proceedings of the 23rd international conference on Machine learning, Pittsburgh, Pennsylvania, USA, 2006: 929-936.
- [11] Cancedda N, Gaussier E, and Goutte C, *et al.* Word sequence kernels [J]. *The Journal of Machine Learning Research*, 2003, 3(Feb): 1059-1082.
- [12] Joachims T, De Thorsten J, and Cristianini N, *et al.* Composite kernels for hypertext categorization. In ICML: International Conference on Machine Learning, Williams College, USA, 2001: 250-257.
- 王 君: 女, 1981 年生, 博士, 研究方向为文本挖掘、自然语言处理、问答系统。
- 李舟军: 男, 1963 年生, 教授, 博士生导师, 研究方向为智能信息处理、信息安全。
- 胡 侠: 男, 1984 年生, 硕士, 研究方向为文本挖掘、自然语言处理。