文章编号：1003-0077(2009)02-0023-05

基于语法分析和统计方法的答案排序模型

李 波，高文君，邱锡鹏

（复旦大学 计算机科学与工程系，上海 200433）

摘 要：该文描述了一种构建问答式检索系统中答案排序模型的新方法。该方法结合了基于密度方法的度量特征和外部知识库，并且引入了基于语法分析方法的语法关键路径的新特征，使用支持向量机回归模型训练评估函数。实验证明，引入了上述语法关键路径特征后的新答案排序模型的排序性能有了明显提高。

关键词：计算机应用；中文信息处理；自动问题回答；语法关键路径；答案排序；支持向量机

中图分类号：TP391　　**文献标识码**：A

Constructing an Answer Ranking Model Using Semantic Analysis and Statistical Method for Question Answering

LI Bo, GAO Wen-jun, QIU Xi-peng

(Dept. of Computer Science and Engineering, Fudan University, Shanghai 200433, China)

Abstract: This paper describes a new method to construct the answer ranking model for Question Answering System. The method leverages knowledge density-based features used in answer ranking and introduces a new feature--syntactic path--by using parsing analysis and establishes an evaluation function by using supporting vector machine regression model. The experiments show that the new model which involves the syntactic path feature achieves substantial improvements.

Key words: computer application; Chinese information processing; question answering; syntactic path; answer ranking; Support Vector Machine

1 介绍

答案排序是开放领域问答式检索系统需要解决的重要问题[1]，答案排序效果的好坏直接决定了整个系统的性能。答案排序的核心是评判候选答案与问题的相似度。

一种直观的方法是使用外部知识库。例如“Who's the president of USA in 2007?”，使用已整理好的知识库，如文献[2]中介绍的经过结构化处理的 Wikipedia，可以很容易地找到“George W. Bush”作为答案。这种方法的缺点是要求使用的知识库已包含了问题的答案以及答案的相关属性，如上面的例子中的时间、国家、职务。有了结构化的知识库，还需要能较准确地自动分析和理解问题，这对于复杂的问句处理比较困难。

Web 作为现存最大的知识库，提供了一条有效的评判途径。文献[3]中描述了使用 Google 评价两个命名实体间相似度的方法。使用类似的方法，在答案排序的任务中，可以将候选答案与原始问题一起作为查询提交给搜索引擎。如果答案是正确，则应能得到较多的返回文档数，返回文档数成为评价答案置信度的一个特征。

评判答案置信度时可以考虑的因素，除了答案本身外，还包括了候选答案所在的上下文。一般来说，如果候选答案所在的上下文与问句的相似度较高，则候选答案的置信度较高。

在分析相关文档时，常用的方法有基于密度的

收稿日期：2008-08-11　**定稿日期**：2008-10-29

基金项目：国家自然科学基金资助项目(60435020)

作者简介：李波(1983—)，男，硕士生，主要研究方向为自然语言处理，信息检索；高文君(1986—)，男，硕士生，主要研究方向为自然语言处理，信息检索；邱锡鹏(1983—)，男，博士，讲师，主要研究方向为自然语言处理，信息检索，机器学习。

方法[4]和基于语法分析[5-6]的方法。

基于密度的方法将问句与相关文档表示为词袋(bag-of-words),分析关键词、词组的重叠度和距离信息,从而评判答案与问句的相似度 。

基于语法分析的方法将问句与相关文档表示为语法分析树。如文献[5-6]提出的基于深层语法分析的方法。首先使用自然语言处理工具分析问句和候选答案上下文,将它们表示成语法分析树或是分析片断。然后根据语法成分和关系类型计算候选答案所在上下文与问句的相似度。这种方法计算量较大,处理也比较复杂。

本文提出了一种新的答案排序方法,结合了基于密度和语法分析的方法,使用统计机器学习方法训练评价函数,为复旦大学问答系统(FDUQA)[7]构建了一个高效的答案排序模型。本文的另一个贡献是提出了一种新的基于语法分析的特征:语法关键路径。FDUQA 系统在 2007 年 TREC 会议组织的 Question Answering 评测中取得了很好的成绩[8]。

本文组织如下:第二部分是 FDUQA 问答系统架构介绍,第三部分详细介绍答案排序的方法,第四部分是实验与分析,最后是小结和展望。

2 系统架构

FDUQA 是一个面向开放领域的问答式检索系统,包含事实性问题回答、列表问题回答、定义类问题回答三个子系统。本文主要关注的是事实性问题的答案排序,系统架构如下:

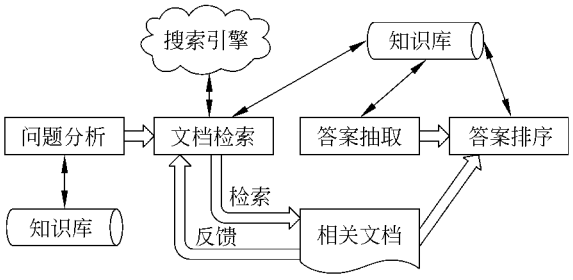

图 1 事实性问题回答系统的架构

1) 问题分析模块:使用 MiniParser[9]等自然语言处理工具分析问句,抽取问句关键词、名词短语或命名实体以及答案类型,并且分析问句句法成分和语法结构。

2) 文档检索模块:生成查询、检索文档。对文档进行简单的过滤和排序。使用查询扩展技术改进文档检索效果。

3) 答案抽取模块:使用基于条件最大熵算法训练的命名实体识别器,抽取与答案类型匹配的候选答案。

4) 答案排序和验证模块:考察候选答案及所在上下文的特征、结合外部知识,对候选答案过滤、计算置信度并排序。

3 构建答案排序模型

答案排序模型需要解决的是,对于给定问题 Q,候选答案集合 A,相关文档集合 D 等信息,对 A 中的候选答案排序。在排序过程中,一般按照候选答案的置信度排序。所以,答案排序模型的关键在于如何计算每个候选答案的置信度。

为了计算候选答案的置信度,需要选取合适的特征。基于密度方法使用的特征一般比较简单,如词频,由于自然语言的复杂性和不确定性,这些特征有其局限性。基于语法分析的方法引入了更高层次的语法特征,能够帮助系统更好地分析和理解候选答案所处的语境。

选取了合适的特征后,计算候选答案的置信度就是一个回归问题。在 FDUQA 中使用了支持向量机回归模型来解决。

3.1 支持向量机的回归模型

为了解决回归问题,在支持向量机中引入新的损失函数,这个损失函数必须包含距离的度量。常用的损失函数包括:Quadratic、Laplace、Huber、ε-Insensitive[10]。

对于线性回归,考虑训练样本:

$$D = \{(x^1, y^1), \cdots, (x^l, y^l)\}, x \in R^n, y \in R \tag{1}$$

最优回归函数

$$f(x) = \langle \omega, x \rangle + b \tag{2}$$

通过最小化目标函数:

$$\Phi(\omega, \xi) = \frac{1}{2} \| \omega \|^2 + C \sum_i (\xi_i^- + \xi_i^+) \tag{3}$$

其中 C 是预先给定的值,ζ^- 和 ξ^+ 是松弛向量,分别表示系统输出的上限和下限。

考虑 loss 函数为 ε-Insensitive 的情况:

$$L_\varepsilon(y) = \begin{cases} 0 & \text{for } | f(x) - y | < \varepsilon \\ | f(x) - y | - \varepsilon & \text{otherwise} \end{cases} \tag{4}$$

将 $L_\epsilon(y)$代入目标函数公式(3)，得到：

$$\bar{\alpha},\bar{\alpha}^* = \arg\min_{\bar{\alpha},\bar{\alpha}^*} \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*) \times\langle x_i,x_j\rangle - \sum_{i=1}^{l}(\alpha_i-\alpha_i^*)y_i + \sum_{i=1}^{l}(\alpha_i+\alpha_i^*)\varepsilon \quad (5)$$

限制条件为：

$$0\leqslant\alpha_i-\alpha_i^*\leqslant C,\quad \sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0 \quad (6)$$

考虑 *Karush-Kuhn-Tucker*(*KKT*)条件，有

$$\alpha_i\alpha_i^* = 0,\quad i=1,\cdots,l \quad (7)$$

在 *KKT* 条件下，公式(5)可以简化为：

$$\min_{\beta}\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\beta_i\beta_j\langle x_i,x_j\rangle - \sum_{i=1}^{l}\beta_i y_i \quad (8)$$

限制条件为：

$$-C\leqslant\beta_i\leqslant C,\quad \sum_{i=1}^{l}\beta_i=0$$

解式(8)，回归函数由式(2)给出，其中

$$\bar{\omega}=\sum_{i=1}^{l}\beta_i x_i,\bar{b}=-\frac{1}{2}\langle\bar{\omega},(x_r+x_s)\rangle \quad (9)$$

3.2　特征选取

3.2.1　基于密度方法的特征

基于密度方法的候选答案评价，将问句和相关文档表示为词的集合，分析问句中词与候选答案所在上下文的词、名词词组共同出现的次数，答案与问句词或词组之间的距离等。

系统中的排序模型选取并计算以下特征：

1) SC-Sentence Confidence：包含候选答案的句子置信度。通过问句关键词或词组与相关文档句子关键词或词组的重叠度以及与 *target* 的重叠度计算。定义为：

$$SC=\lambda * count(keyword)/length(sentence) + (1-\lambda) * count(target)$$

其中，*target* 是 TREC 组织的 Question Answering 评测中给出的与问句相关的信息，例如对于问题“Who is Warren Moon's agent”的 target 是“Warren Moon”，keyword 为“Warren Moon”和“agent”。λ 是调节因子，在系统中取 $\lambda=0.5$，*count*(*keyword*)、*count*(*target*)分别为相关句中出现的问句关键词数和 *target* 词数。

2) DC-Document Confidence：包含候选答案的文档段落的分数。通过选取相关文档中置信度最高的句子的置信度计算。定义为：$DC=\mathrm{MAX}(SC_i)$

3) AC-Answer Count：候选答案出现次数。统计候选答案在相关文档中出现的次数即频数。

4) DS-Distance Score：相关文档中，候选答案与问句关键词或词组的距离。对每一句包含候选答案的句子计算距离，并取加权平均值。

定义为：

$$DS=avg\sum_{i}\frac{1}{abs(pos(keyword_i)-pos(answer))}$$

$keyword_i$ 属于问句关键词的集合。*pos*(*keyword*$_i$)和 *pos*(*answer*)分别是问句关键词和候选答案在相关文档的句子中所处的位置。*DS* 值对求得的距离值取平均，然后归一化。

3.2.2　基于语法分析的特征

基于语法分析的方法将问句与相关文档表示成语法分析树，通过比较这两棵语法分析树或分析片断获取答案与问句的相似度。在系统中使用 MiniParser 进行语法分析，由 MiniParser 的分析结果，引入了两种基于语法分析树的度量特征。

第一种是文献[6]中介绍的基于语法三元组：Triple 的方法。给定一颗语法分析树，一个语法三元组表示为(Slot1，Rel，Slot2)，Slot1 和 Slot2 是词或是词组，Rel 是 Slot1 和 Slot2 的管辖关系。通过三元组的匹配，计算问句与相关文档的相关度，并由此得到候选答案的置信度。

3.2.2.1　语法关键路径

这里更进一步提出一种新的基于语法分析的度量特征。上面的三元组特征，由于需要严格匹配 Rel，即使将一些 Rel 作为等价关系组而允许模糊匹配，能够匹配的模式仍然较少。所以，考虑使用语法树上的关键路径弥补三元组特征的较小的覆盖率。

将语法关键路径定义为，给定相关文档的语法分析树 T、问句关键词集合 W 和一个候选答案 A，选取 W 中任一个关键词 Wi，都可以从 T 中找到一条从 A 到 Wi 的最短路径，把这条路径称为语法关键路径。一般的，关键路径经过的节点数越少，即关键路径的长度越短，关键路径两端的词或词组的相关度越高。

例如：对问题：“Who's Warren Moon's agent?”相关文档：“Leigh Steinberg has represented such greats as Troy Aikman, Steve Young, Warren Moon”，候选答案“Leigh Steinberg”。如图 2 所示，将相关文档表示为语法分析树。

在语法分析树上，抽取从候选答案“Leigh

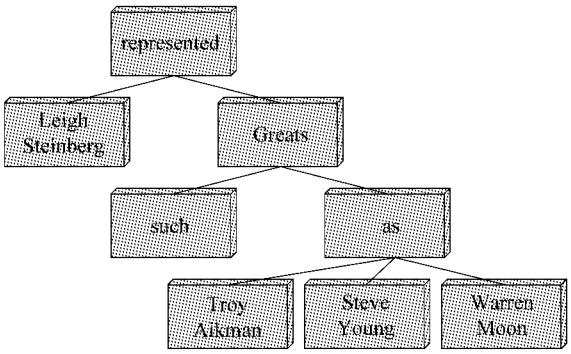

图2 例句"Leigh Steinberg has represented..."的语法分析树

Steinberg"到关键词 Warren Moon 的语法关键路径为"Leigh Steinberg-represented-greats-as-Warren Moon"。

因为语法关键路径不考虑语法分析树上的节点之间的具体语法关系,所以与语法三元组相比,语法关键路径特征具有较高的覆盖度。在计算基于密度方法的 *DS* 值时,可能因为某一句法成分过长,而使 *DS* 值降低。在上面的例子中,问题关键词"Warren Moon"与候选答案"Leigh Steinberg"之间许多无关的词降低了 *DS* 值,减弱了两者之间的联系。而在语法关键路径上,这些"Troy Aikman","Steve Young"作为相同的句法成分被压缩至同一个语法分析子树下。在语法关键路径的表示下,问句关键词与候选答案之间的距离特征更能表现词或词组间的联系。在上面的例子中"Leigh Steinberg"与"Warren Moon"在原句中的距离为9,两者在语法关键路径上的距离为3。

定义特征度量:语法路径距离 SDS-Syntactic Distance Score 为问句关键词与候选答案在语法关键路径上的距离。*SDS* 考虑问句关键词在问题中重要性的不同,对"问题关键词—候选答案"在语法关键路径表示下的距离求加权平均,具体计算方法:

$$SDS = avg\sqrt{\frac{\sum \alpha_j w_j}{\sum (w_i \times Syntactic\_path\_dist_i)^2}},$$

$$\alpha_j = \begin{cases} 0, & \text{otherwise} \\ 1, & \text{keyword}_j \text{ occurs in syntactic path} \end{cases}$$

$\sum \alpha_j w_j$ 是在相关文档中出现的问句关键词的权重之和。$syntantic\_path\_dist_i$ 是语法关键路径上候选答案与问句关键词的距离,即从候选答案出发到达指定问句关键词所要经过的节点数。

α_i 是问句关键词的权重,首先按照词性赋予初始权重。然后使用 MiniParser 分析修饰关系,加大被修饰词的权重。具体算法如下:

关键词集合记为 Term,对应的权重集合记为 Weight。

1) 根据词性赋予权重,名词为5、动词为7、形容词为1,数词为10,其他词性为0;

2) 如果关键词在问句 target 中出现,此关键词的权重乘以2;

3) 根据分析结果中的 MOD、DET、GEN 等修饰关系,被修饰词的权重乘以2;

4) 归一化,使权重值属于[0,1]。

3.3 答案排序模型

在答案排序模型中,使用置信分数对候选答案打分。将置信度定义为 Confidence = $H(f_1, f_2, \cdots f_n)$,其中 f_i 是上文中描述的特征值。H 是评价函数,使用支持向量机回归模型训练得到。训练时使用程序库 LibSVM[11],训练语料为 Trec2004、Trec2005 的事实性问题。答案排序模型的具体算法如下:

```
训练 H 评价函数
对给定问题的一组候选答案、相关文档集合 snippet 和问题 question
进行答案过滤;
For 每一个候选答案 candidate
    F1 = ExtractDensityFeature(candidate, snippet);
    F2 = ExtractSyntacticFeature (candidate, question, snippet);
    Confidence(candidate) = H(F1,F2);
End For
SortBy(Confidence);
```

ExtractDensityFeature 抽取统计特征,得到特征向量 $F_1 = (AC, SC, DC, DS)$

ExtractSyntacticFeature 抽取词法特征,得到特征向量 $F_2 = (Triple, SDS)$

答案过滤使用一些启发式规则和外部知识库进行。

1) 候选答案中不应出现停用词、问句或 target 中出现的词。这样的候选答案会被过滤。

2) 对于提问时间的问题,答案必须满足"{month}{day},{year}"的模式。如果问题使用"What year"或"What month"提问则答案必须是表示年份的4位数字或是特定月份。

3) 对于一般的问题,如果能抽取问题的中心词

(Focus)，则要求答案与 Focus 有上下位关系。这里在系统中使用了 WordNet[12] 来判断两个词之间的上下位关系。例如，问句使用“What city”提问，那么候选答案必须是 city 的下位词或在 WordNet 的解释中有 city 的义项。

4　实验与分析

4.1　实验一：特征选取的比较

测试集合为 Trec2006 的 403 道事实性问题。选择不同的特征集合，比较不同特征对排序性能的影响。核函数选取线性核函数。Baseline 系统使用的是 FDUQA2006 的排序模型。排序过程为首先按照答案抽取次数排序，将排序好的答案再依次按照查询置信度、句子置信度、答案类型置信度排序。最后一次排序完成后，得到排序好的候选答案列表。具体可参考文献[7]。

表 1 给出的是对给定问题的候选答案列表，使用不同特征的答案排序模型对答案排序后，排在列表首位的是正确答案的问题的比例，例如对于问题“Who's Warren Moon's agent?”的候选答案排序后，若排在首位的是正确答案“Leigh Steinberg”则认为这次排序是正确的。如表 1 所示，加入基于语法分析的语义关键路径的引入提高了排序性能。

表 1　答案排序结果与特征选取

	首位正确率	改进百分比
Baseline 系统	23.0%	—
线性 f_1，f_2，f_3	24.8%	7.8%
线性 f_1，f_2，f_3，f_4	25.3%	10.0%
线性 f_1，f_2，f_3，f_4，f_5	26.7%	16.1%

f_1：Sentence Score；f_2：Document Score；f_3：Triple；f_4：Answer Count；f_5：SDS

4.2　实验二：评价函数 H 对答案的评分性能

测试集合为所有 Trec2006 的 403 道事实性问题的 1 737 个候选答案上进行，其中正确的候选答案有 187 个。训练 H 时，使用所有的特征。对每一个候选答案，使用评价函数 H 计算置信度，置信度大于等于 0.7 判为正确答案，否则为错误答案。

表 2　评价函数 H 的评分性能，置信度大于 0.7 为正例

	True Positive	False Positive
线性核函数	27.5%	2.4%
多项式核函数	27.1%	2.9%
RBF 核函数	26.7%	3.4%

True Positive 是候选答案为正确答案，且置信度对于 0.7 的样例。False Positive 是候选答案为错误答案，但置信度大于 0.7 的。例如，对于候选答案的“Troy Aikman”，“Leigh Steinberg”都获得了大于 0.7 的置信度。例如正确答案“Leign Seinberg”为 True Positive，而“Troy Aikman”是 false positive。如表 2 所示，SVM 回归模型引入了语法关键路径等特征后，能够较好的给出候选答案置信度。

4.3　实验三：核函数选取

使用所有基于密度和基于语法分析的特征，分别使用线性核函数、多项式核函数、RBF 核函数训练 H 评价函数。如表 3 所示，经过排序后的结果比分类结果有所降低。这是因为，对同一问题可能有有多个候选答案被判为正例，即它们的置信度都大于 0.7。而 false positive 的候选答案的置信度较高，因此导致了排序错误。整体上看，排序效果较原始系统均有了一定的提高。

表 3　答案排序结果与核函数选取

	首位正确率	改进百分比
Baseline 系统	23.0%	—
线性核函数	26.7%	16.1%
多项式核函数	26.0%	13.0%
RBF 核函数	25.8%	12.2%

5　小结和展望

本文介绍的通过结合知识库、基于密度和基于语法分析的特征分析，使用支持向量机训练，构建的答案排序模型有效地提高了答案排序的效果。接下去的工作，我们将考虑如何结合语法分析树构建更加合理的 Kernel 函数，改进支持向量机训练模型，并加入结构化的外部知识。

（下转第 47 页）