基于常问问题集的中文问答系统研究

秦兵　刘挺　王洋　郑实福　李生

(哈尔滨工业大学计算机学院信息检索实验室，哈尔滨 150001)

E-mail: { qinb,tliu }@ir.hit.edu.cn

摘要：首先根据用户的提问建立一个候选问题集，然后通过计算句子语义相似度，在候选问题集中找到相似的问句，并将答案返回给用户。该系统还能够自动地更新和维护 FAQ 库。实验表明，与基于关键词的句子相似度计算相比，基于语义的句子相似度计算提高了问题匹配的准确率。

关键词：问答系统；常问问题集；句子相似度

分类号：TP391

Research of Question Answering System Based on Frequently Asked Questions

Qinbing,Liuting,Wangyang ,Zhengshifu,Lisheng

(Information Retrieval Laboratory, Harbin Institute of Technology, Harbin 150001,China)

E-mail: {qinb, tliu}@icm.hit.edu.cn

Abstract: In this paper, the candidate question set is built according to query. Semantic similarities of sentences are computed between the user query and the candidate questions. In this way the corresponding answer with the most similar with query is returned to the user. This system can also automatically update and maintain FAQ. Experiment shows that the new computing method gets a better performance than the keywords-based approach.

Key words: question answering; FAQ; sentence similarity

基于常问问题集的问答系统是在已有的问题-答案对的集合中找到与用户提问相匹配的问题，并将其对应的答案直接返回给用户。问答系统是目前自然语言处理领域一个研究热点[1]，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。因此，问答系统和传统的依靠关键字匹配的搜索引擎相比，能够更好地满足用户的检索需求，更准确地找出用户所需的答案，具有方便、快捷、高效等特点。目前问答系统的研究大致可以分作三类：基于常问问题集的问答系统，基于百科知识的问答系统以及开放域的问答系统。基于常问问题集的问答系统又可以作为后两种问答系统的一个组成部分，如果用户的提问与以往的记录相符，可直接将对应的答案提交给用户，免去了重新组织答案的过程，可以提高系统的效率。

常问问题集 (FAQ)可以作为自动问答系统中的一个组成部分。它把用户经常提问的问题和相关答案保存起来。对于用户输入的问题，可以首先在常问问题库中查找答案。如果能够找到相应的问题，就可以直接将问题所对应的答案返回给用户，而不需要经过问题理解、信息检索、答案抽取等许多复杂的处理过程，提高了效率。国外近年来在该领域作了一些工作[2]，国内的这方面研究还很少。本文研究的 FAQ（*Frequently-Asked Question*）系统根据用户问题建立候选问题集的基础上，建立常问问题集的倒排索引，提高了系统的检索效率，同时，与传统的基于关键词的方法相比，采用基于语义的方法计算相似度提高了问题的匹配精度。文中描述的句子相似度的计算方法不仅能够用于 FAQ 的检索，还能够用于自动问答的其它阶段，以及信息检索和基于实例的机器翻译等领域，并且基于常问问题集的问答系统在远程教育等领域有很广泛的应用前景。

1 系统实现

该系统主要包含三个部分：候选问题集的建立，句子相似度计算，FAQ 库的更新。

1.1 候选问题集的建立

建立候选问题集的目的是缩小查找范围，使后续的相似度计算等较复杂的处理过程都在候选问题集这个相对较小的范围内进行。本系统选出 FAQ 中 50%的问句作为候选问题集。设用户输入的问句（目标问句）中共有 n 个词（W_1、W_2、…、W_n），FAQ 库中共有 m 个问句，第 $i(1 \le i \le m)$ 个问句含有 n_i 个词（Q_1、Q_2、…、Q_{n_i}），第 i 个问句和目标问句之间重叠的词个数记为 Num_i，即 $Num_i = |\{W_1, W_2, ..., W_n\} \cap \{Q_1, Q_2, ..., Q_n\}|$，则 Num_i 值最大的前 50%的 FAQ 问句就组成候选问题集。

计算 Num_i 时，如果将 FAQ 库中的问句一一读出来和目标问句进行比较，效率是比较低的。对于目标问句中的某个词，为了能够快速地统计 FAQ 库中究竟有多少问句含有这个词，设计了如图 1 所示的数据结构。

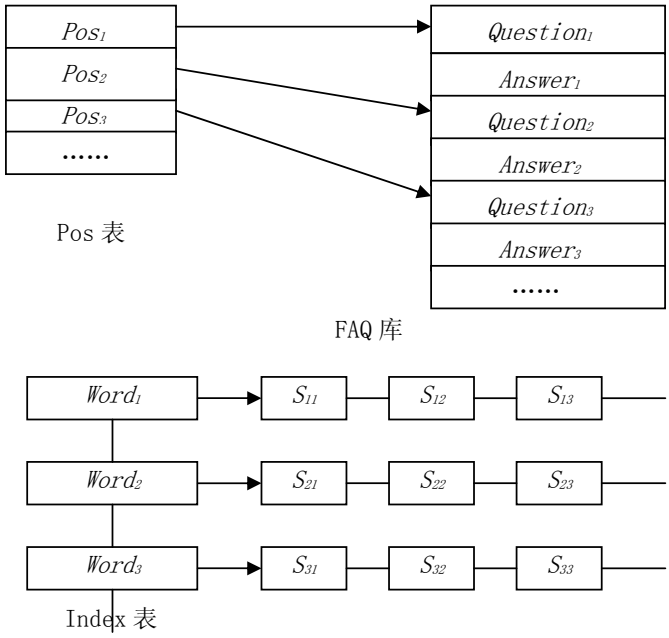

图 1 常问问题集的数据结构

Fig 1. Data structure of Frequently Asked question set

图 1 中的 FAQ 库记录了所有原始的问题-答案对，Pos 表记录了 FAQ 库中每个问句在库文件中的位置，Index 表中的 $Word_1$、$Word_2$……是 FAQ 库中的问句所包含的词经过排序后所形成的链表。每个 $Word_i$ 指向一个 S 链表，这个 S 链表中的每个节点记录 FAQ 库中含有 $Word_i$ 的一个问句的句子号。

在实际的检索过程中，对于目标句子中的一个词 w，寻找它在 Word 链表中的位置。由于 Word 链表是有序的，可以很容易地利用折半查找等方法在 O(logn)的时间复杂度内找到目标。不妨设找到的节点为 $Word_k$，沿着 $Word_k$ 所指向的 S 链表，可以统计出有哪些 FAQ 库中的问句包含 $Word_k$。对目标问句中的每一个词都进行这样的处理后，就可以进一步计算出 Num_i 的值。 最后，找出 Num 值最大的 50%个问句的句子号，通过 Pos 表将 FAQ 库的文件指针移到相应的问句处，并把问句读出。

1.2 句子相似度计算

候选问题集确定后，下一步是要从这个集合中找出和目标问句最相似的问句。所用的方法是计算候选问题集中每个问句和目标问句之间的相似度，对应的相似度最大的问句就是要找的句子。

计算相似度的方法有很多，本文综合运用了 2 种方法来计算句子的相似度：基于向量空间模型的 TFIDF 方法和基于语义的句子相似度计算方法。

1.2.1 基于向量空间模型的TFIDF方法

若 FAQ 中所有问句包含的所有的词为w_1、 w_2、…、 w_n,则 FAQ 中的每一个问句都可以用一个 n 维的向量$T =< T_1, T_2, ..., T_n >$来表示。$T_i\left(1 \le i \le n\right)$的计算方法为：设 n 为$w_i$在这个问句中出现的个数， m为 FAQ 中含有w_i的问句的个数， M 为 FAQ 中问句的总数，那么$T_i = n \times \log(M / m)$。可以看出，出现次数多的词将被赋予较高的 n 值，但这样的词并不一定具有较高的$\log(M / m)$值。例如，在汉语中“的”出现的频率非常高，即 TF 值(n 值)很大，但由于“的”在很多文档中都出现，它对于分辨各个文档并没有太大的帮助，它的 IDF 值$\left(\log(M / m)\right)$将是一个很小的数。因此，这种方法综合地考虑了一个词的出现频率和这个词对不同文档的分辨能力。

用同样的方法，可以计算目标问句的 n 维向量$T' =< T_1', T_2', ..., T_n' >$。得到$T$ 和T'后，它们所对应的 2 个句子之间的相似度就可以利用T 和T'这 2 个向量之间夹角的余弦值来表示，即

$$Similarity(T, T') = \frac{\sum_{i=1}^{n} T_i \times T_i'}{\sqrt{\sum_{i=1}^{n} T_i^2 \sum_{i=1}^{n} T_i'^2}}$$

TFIDF 方法综合考虑了不同的词在问句中的出现频率(TF 值)和这个词在整个 FAQ 库中对不同句子的分辨能力(IDF 值)。这种方法不需要任何对文本内容的深层理解，它能够在 FAQ 中应用，一个很重要的原因是 FAQ 库是开放域的自然语言文本，而且 FAQ 库通常都很大。

1.2.2 基于语义的相似度计算方法

在本系统中，单靠 TFIDF 方法还不能达到预期的结果。原因有 2 个：（1） TFIDF 是一种统计的方法，只有当句子包含的词数足够多时，相关的词才会重复出现，这种统计方法的效果才能体现出来，而 FAQ 中，面对的是单个的问句，问句中包含词的个数往往不足以体现这种方法的效果；（2）TFIDF 方法只考虑了词在上下文中的统计特性，而没有考虑词本身的语义信息，因此具有一定的局限性。基于上述原因，本文描述的系统引入了基于语义的相似度计算方法。

计算语义相似度，需要一定的语义知识资源作为基础。本文采用董振东和董强先生创建的知网(HowNet)作为系统的语义知识资源[3]。知网是一个以汉语和英语所代表的概念为描述对象，以揭示概念与概念之间以及概念所具有的属性之间的关系为基本内容的常识知识库，是一个网状的有机的知识系统。

语义词典是知网的基础文件，在这个文件中每一个词语的概念及其描述形成一个记录。每一个记录都包含词语、词语例子、词语词性、概念定义等 4 项。这里主要用到的是概念定义(即 DEF)这一项。

计算句子之间的语义相似度，要确定句子中的词在这个句子中所表达的语义。例如，“打毛衣”中的“打”作为“编织”的意思，而“打酱油”中的“打”作为“买”的意思，需要确定“打”这个词在不同的句子中的不同含义。语义消歧能够挖掘出一个词在上下文中确切的含义，而不是仅仅停留在词的表面。这为后面的工作奠定了基础。

除了语义词典，知网中还提供了义原分类树。义原分类树把各个义原及它们之间的联系以树的形式组织在一起，树中父节点和子节点的义原具有上下位的关系。可以利用义原分类树计算 2 个词之间的语义距离。知网中存在 Entity、Event、Attribute 等 11 棵义原树。但有些义原树，例如 Converse、Antonym 等，里面的义原没有父子关系，并不体现上述的词与词之间的上下位特征，因此无法使用。在 11 棵义原树中总共选取了以下 6 棵义原树用来计算词的语义距离：Entity、Event、Attribute、Attribute Value、Quantity、Quantity Value。

首先需要计算 2 个词之间的语义距离。这里，把语义距离定义为两个词对应的义原在义原树中的最短距离。如果 2 个词中有一个词的义原无法在 6 棵义原树中找到，或者 2 个词的义原分别处于 2 个不同的义原树，则认为这两个词之间的语义距离为∞。设 2 个词 U，V 之间的语义距离为 D，那么 U，V 之间的相似度可以为

$$s(U,V)=\begin{cases} D/p, & p\neq\infty; \\ 0, & p=\infty. \end{cases}$$

其中，$D=|T1\cup T2|-|T1\cap T2|$，$T_1,T_2$ 分别是 2 个词所在义原树从树根到该节点语义元素集合，$T1\cup T2$ 是义原树中从树根到 U，V 各自语义节点包括的所有义原的集合，$|T1\cup T2|$ 是该集合元素个数，$T1\cap T2$ 表示 U，V 对应语义树相同语义节点集合，$|T1\cap T2|$ 表示公共节点的个数，则 D 表示义原树中 U，V 这 2 个节点的路径最短距离。不同的义原树的长度不同，需做归一化处理，p 为义原树的总节点层次级数。因此，2 个词相似度取值在 0 到 1 之间。

有了词与词之间的语义相似度，就可以来计算句子间的语义相似度。设 2 个句子 A 和 B，A 包含的词为 A_1、A_2、…、A_m，B 包含的词为 B_1、B_2、…、B_n，则词 $A_i(1\le i\le m)$ 和 $B_j(1\le j\le n)$ 之间的相似度可用 $s(A_i,B_j)$ 来表示，这样就得到两个句子中任意 2 个词的相似度，A,B 句子之间的语义相似度 $s(A,B)$ 为

$$s(A,B)=\left(\frac{\sum_{i=1}^{m}a_i}{m}+\frac{\sum_{i=1}^{n}b_i}{n}\right)/2$$

式中：

$a_i=\max(s(A_i,B_1),s(A_i,B_2),...,s(A_i,B_n))$；$b_i=\max(s(B_i,A_1),s(B_i,A_2),...,s(B_i,A_n))$.

上述计算相似度的方法还用于问答系统中的其它阶段。例如，在对问题“西红柿是什么颜色的？”的答案查找过程中，有一个句子为“番茄是红色的。”，这两个句子通过上述的相似度计算方法就可以匹配上。而如果单纯地依靠关键词，“西红柿”和“番茄”是不可能匹配上的，而基于语义的句子相似度计算可以使这 2 个句子获得较高的相似度，从而找到正确答案。

1.3 FAQ库的更新

计算出用户所输入的目标问句和候选问题集中每个问句的相似度，如果所有这些计算出来的相似度的最大值大于一定的阈值，那么就认为最大的相似度所对应的问句和用户的目标问句是同一个问题。可以直接将这个问句对应的答案输出给用户。如果最大相似度的值小于阈值，系统就认为 FAQ 库中没有用户所问的问题，那么将用户所问的这个问题和对应的答案加入 FAQ 库，对其进行自动的更新。

2 实验与分析

为了验证方法的有效性，利用标准的测试集进行了测试。根据文献[4]定义的 2 个衡量聚类结果质量的指标：聚类精确度（Precision）和 聚类召回率（Recall），检测相似度的计算结果。

聚类精确度反映把相似文本单元和不相似的文本单元合并到同一个类的程度，聚类精度越高，得到的聚类结果的每个类中的内容越集中。它是反映聚类结果质量的重要指标，定义为

$$\Pr ecision = \frac{1}{N_{CL}} \sum \frac{R_i}{T_i}$$

式中：N_{CL} 是类数；R_i 是第 i 类中计算得到相似文本单元的数量；T_i 是第 i 类中标准文本单元的总数。

聚类召回率反映出将同一主题相似文本单元合并到一个类中的程度，聚类召回率越高，相似的文本单元就越集中，它们被拆分到不同的类中的情况就越少。聚类精确度反映的是对不同主题的区分能力，而聚类召回率反映了对相同主题的识别能力。因此，它也是反映聚类结果质量的一个重要指标，公式如下：

$$\text{Re}call = \frac{1}{N_{TOP}} \sum (\frac{R_i}{\sum R_j})$$

式中：N_{TOP} 是文本中描述内容的相对集中在的主题数；R_i 是第 i 类中与本类主题相似文本单元的数量（第 i 类是这个主题最集中的类）；R_j 是第 j 类中与正在考察的主题相似的文本单元的数量。

有了评价标准就可以对方法进行评价。标准的测试集是 60 类，每类包含若干个相似句子。不同类之间的句子是不相似的。用聚类分析方法对它们进行聚类操作，统计出这些聚类结果的精确率和召回率，如表 1 所示。实验结果表明，基于语义的相似度计算方法对包含词不相同而词的语义相近的句子有较好的效果，而对包含相同关键词较多的句子基于关键词的方法和基于语义的相似度计算方法差别不大。

表 1 实验结果

Table 1. The result of experiment

方法	准确率/%	召回率/%
基于 TFIDF 句子相似度计算方法	73.2	74.12
基于语义的句子相似度计算方法	80.78	76.31

3 结论

本系统能够根据用户以自然语言输入的问题自动地在 FAQ（Frequently-Asked Question）库中寻找候选问题集，通过计算句子相似度，将匹配的答案返回给用户，在一定程度上提高了获取信息的效率。同时，本文中描述的句子相似度的计算方法对中文信息处理的其他领域，如信息检索和基于实例的机器翻译等领域具有一定的借鉴作用 。

参考文献

[1]郑实福，刘挺，秦兵等.中文自动问答系统综述[J].中文信息学报，2002，6（16）：46-52

[2] Burke R D, Hammond K J, Kulyukin V, etal. *Question answering from frequently asked question files: Experiences with the FAQ Finder systemp[J]*. AI Magazine,1997, 18：57-66

[3]董振东，董强. 知网[EB/OL]. http://www.keenage.com

[4] Hatzivassiloglou V, Klavans J L, Holcombe M L, etal. *SIMFINDER: A flexible clustering tool for summarization*[A]. In NAACL Workshop on Automatic Summarization. Association for Computational Linguistics, 2001

删除的内容: