基于最大熵模型的 QA 系统置信度评分算法*

游 斓+, 周雅倩, 黄萱菁, 吴立德

(复旦大学 计算机科学与工程系,上海 200433)

A Maximum Entropy Model Based Confidence Scoring Algorithm for QA

YOU Lan+, ZHOU Ya-Qian, HUANG Xuan-Jing, WU Li-De

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

+ Corresponding author: Phn: +86-21-65642830 ext 315, E-mail: lan_you@fudan.edu.cn, http://www.fudan.edu.cn

Received 2004-02-25; Accepted 2004-07-06

You L, Zhou YQ, Huang XJ, Wu LD. A maximum entropy model based confidence scoring algorithm for QA. *Journal of Software*, 2005,16(8):1407−1414. DOI: 10.1360/jos161407

Abstract: Confidence score describes how confident a question-answering system is about its response. This paper presents a Maximum Entropy Model based algorithm which uses several factors to train an ME model, and then the ME model is used to calculate the confidence of other questions. Efficiency of this method has been proved by the TREC11's QA evaluation, where the performance of the system has been improved dramatically after confidence ranking.

Key words: natural language processing; information retrieval; question-answering system; maximum entropy model; confidence score

摘 要: 置信度指的是一个问题回答系统(QA 系统)对其所作回答的自信程度.描述了一种基于最大熵模型的算法.首先,从训练语料中提取若干因素来训练最大熵模型;然后应用训练好的模型在测试集上计算置信度.在 2002 年度的文本检索会议(TREC)中,QA 系统用该算法计算每个问题答案的置信度,并依此排序,获得了显著的成绩.

关键词: 自然语言处理;信息检索;问答系统;最大熵模型;置信度

中图法分类号: TP301 文献标识码: A

如何从堆积如山的电子文档中获取自己感兴趣的部分,迄今为止,最常用的工具便是搜索引擎.一般来说,人们在使用搜索引擎检索文档时,总会在带着一些问题,比如某人想知道犹他州的国家公园是什么(What is the National Park in Utah?),他也许就会向搜索引擎提交"Utah National Park"这样的查询.然而传统的搜索引擎进行的只是文档检索的工作,并非真正的信息检索.人们仍然要在搜索引擎返回的相关文档中寻找问题的答案.

问题回答(QA)任务的目的就是要为问题找到确切的答案.用户可以用自然语言向一个问题回答系统(QA

* Supported by the National Natural Science Foundation of China under Grant No.60435020 (国家自然科学基金); Key Project of Shanghai Science and Technology Committee of China under Grant No.035115028 (上海市科委重点项目)

作者简介: 游斓(1979－),女,上海人,硕士生,主要研究领域为自然语言处理,信息检索;周雅倩(1976－),女,博士生,主要研究领域为自然语言处理,信息检索;黄萱菁(1972－),女,博士,副教授,主要研究领域为自然语言处理,信息检索;吴立德(1937－),男,教授,博士生导师,主要研究领域为自然语言处理,信息检索,视频检索.

系统)提问,系统将会在庞大的语料库中找到关于这个问题的若干个答案,然后给所有的答案打分,并将最好的那个答案反馈给用户.图 1 用一个具体的例子描述了 QA 任务.用户向 QA 系统提问:What is the National Park in Utah?系统在语料库中搜索到答案 Zion,并返回给用户.

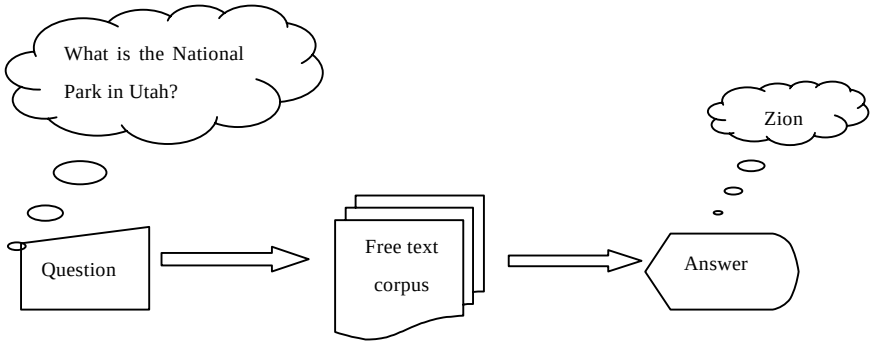

Fig.1 Description of QA task

图 1 QA 任务的描述

通常情况下,QA 系统往往会根据答案的确切程度来给它们打分,较好的答案将被赋予较高的分数.但这个分数仅仅告诉用户,对一个特定的问题来说,哪个答案会更好一些.然而用户更想知道的是,在输入的一系列问题中,系统对哪些回答的正确性更有把握一些.这就要求 QA 系统还应该能够了解其所作回答的正确程度.因此,为了衡量 QA 系统的这种能力,TREC2002 首次引进了“置信度”标准[1].“置信度”也就是 QA 系统对其所作回答的自信程度.

在 TREC2002 的 QA 任务中,许多系统仅使用答案类型作为考量的因素,以确定对不同问题答案的置信度[2].也有一些系统就直接使用了答案评分的结果[3].但用这两种方法算出的置信度往往不够精确.一个较好的置信度评分算法应该综合考虑系统处理各种不同类型问题的能力,以及在处理过程中系统所涉及的各项参数,当然也包括了前面提到的答案分数.

在 TREC2002 中,也有一些系统考虑到了这一点,但是这些系统只是用经验公式来综合计算所有的因素.排名前 10 位的单位中只有 BNN 使用经验公式来综合问题回答过程中涉及的各项因素,从而得到置信度分数[4].他们考虑了以下 3 个因素:问题中的动词、答案上下文与问题的匹配情况以及问题类型,并对这 3 个因素赋予相同的权重.而在我们的系统中,仅答案评分一项就已经涵盖了所有这 3 个因素.为了更加精确地计算置信度,我们使用了一种基于最大熵模型的算法,综合了答案分数和问题类型在内的若干因素,并在实验中取得了令人满意的效果.

本文第 1 节主要介绍基于最大熵模型的置信度评分算法,其中第 1.1 节描述了最大熵模型的原理,第 1.2 节详细分析我们提出的算法.第 2 节主要介绍实验的设计并分析实验结果.最后是对我们工作的一个小结.

1 置信度评分算法

1.1 最大熵模型

1.1.1 基本原理

建立最大熵模型的基本思想是为所有已知的因素建立模型,而把所有未知的因素排除在外[5].也就是说,要找到这样一个概率分布,它满足所有已知的事实,且不受任何未知因素的影响.

QA 系统在处理一个特定问题的过程中会涉及各种因素,假设 X 就是一个由这些因素构成的向量,变量 y 的值反映了答案的正确性,y=1 表示答案正确,y=0 表示答案错误.概率 $p(y|X)$是指系统对某个问题给出答案正确或错误的可能性.这个概率可以用上述思想来估计.最大熵模型要求 $p(y|X)$在满足一定约束的条件下,必须使得下面定义的熵取得最大值:

$$H(p)=-\sum_{X,y}p(y|X)\log p(y|X),$$

这里的约束条件实际上就是指所有已知的事实,一般可以用以下的方式来表述:

$$f_i(X,y)=\begin{cases}1, & \text{if } (X,y) \text{ satisfies certain condition}\\ 0, & \text{else}\end{cases},$$

i=1,2,3,…,n,称 $f_i(X,y)$为最大熵模型的特征,n 为所有特征的总数.可以看到,这些特征描述了向量 X 与答案正确性 y 之间的联系.

概率 $p(y|X)$必须满足上述特征的约束,由此可以定义一个受限的概率分布族为:

$$\wp=\{p(y|X):E_p\{f_i\}=E_{\tilde{p}}\{f_i\},1\le i\le n\},$$

其中,

$$E_p\{f_i\}=\sum_{X,y}f_i(X,y)p(X)p(y|X),$$

$$E_{\tilde{p}}\{f_i\}=\sum_{X,y}f_i(X,y)\tilde{p}(X)\tilde{p}(y|X),$$

$\tilde{p}(X)$ 和 $\tilde{p}(y|X)$ 都是在训练数据中观测到的经验分布.

现在的问题就是要在受限的概率分布族中找到一个具有最大熵的分布,即

$$p\times(y|X)=\arg\max_{p(y|X)\in\wp}\left\{-\sum_{X,y}(\tilde{p}(X)p(y|X))\log p(y|X)\right\},$$

可以求出上式的解为[5]

$$p\times(y|X)=\frac{1}{Z(X)}\exp\left(\sum_i\lambda_i f_i(X,y)\right),$$

$$Z(X)=\sum_y\exp\left(\sum_i\lambda_i f_i(X,y)\right),$$

其中λ_i是每个特征的权重.

1.1.2 建立最大熵模型

在我们的方法中,向量 X 由系统处理问题的过程中提取出的各种因素组成,由于最大熵模型要求 X 的各个分量取离散值,我们首先要把原本取连续值的因素离散化.X 描述了我们的系统是如何获得答案的,而 y 则描述了答案的正确性.因此,$p(y=1|X)$就表示在特定因素向量为 X 的情况下答案正确的概率,这也正是我们期望知道的系统对答案的置信度.因为 y 只有 0 和 1 两种取值,我们可以用下式来计算置信度:

$$confidence=p\times(y=1|X)=\frac{1}{Z(X)}\exp\left(\sum_i\lambda_i f_i(X,1)\right),$$

$$Z(X)=\exp\left(\sum_i\lambda_i f_i(X,1)\right)+\exp\left(\sum_i\lambda_i f_i(X,0)\right).$$

这里,我们只选择在训练数据中出现次数大于 3 的特征,以避免过拟合现象,然后使用 IIS 来计算特征参数.

1.2 问题回答过程因素

1.2.1 FDUQA 系统介绍

为了更好地说明在本算法中用到的各种因素,先简要介绍我们的 QA 系统.

系统的输入是一系列基于事实的问题,经过处理以后,系统输出这些问题的答案.并且,所有的答案将按系统对其“置信度”从高到低排序.

类似于大多数 QA 系统,我们的系统主要由 4 个模块组成:预处理和索引模块(离线模块)、问题分析模块、检索模块以及答案抽取模块(具体流程如图 2 所示).

在分析问题和抽取答案时,系统会用到一个知识库,该知识库包含了约 80 个问题类型.每个问题类型又由 3 个部分组成:问题模板、答案类型模板(又称内部模板)和上下文模板(又称外部模板).问题模板是指问题出现的

形式,答案类型模板是指该类问题的答案可能是何种类型(如时间、地名等),上下文模板是指答案可能出现的上下文的形式.为了获得一定的召回率和较高的准确率,系统将问题的答案同时限定在内部和外部模板中.目标文本与模板之间的匹配存在两种方式:宽松的或严格的.无论是答案类型模板还是上下文模板的匹配都可以是这两种方式中的任意一种,但不允许两者同时以宽松的方式匹配.抽取答案的时候,第 1 步先采用严格的答案类型和上下文模板匹配方式.如果这样无法取得答案,第 2 步再改用宽松的匹配方式,即是内外两个模板一松一严.如果第 1 步就能找到答案,就不进行第 2 步了.我们把上下文模板中的每一项,包括答案本身都看作是一个概念.系统首先在目标句子中定位所有的概念,然后尝试匹配模板,如果在第 1 或第 2 步匹配成功,就将答案抽取出来.

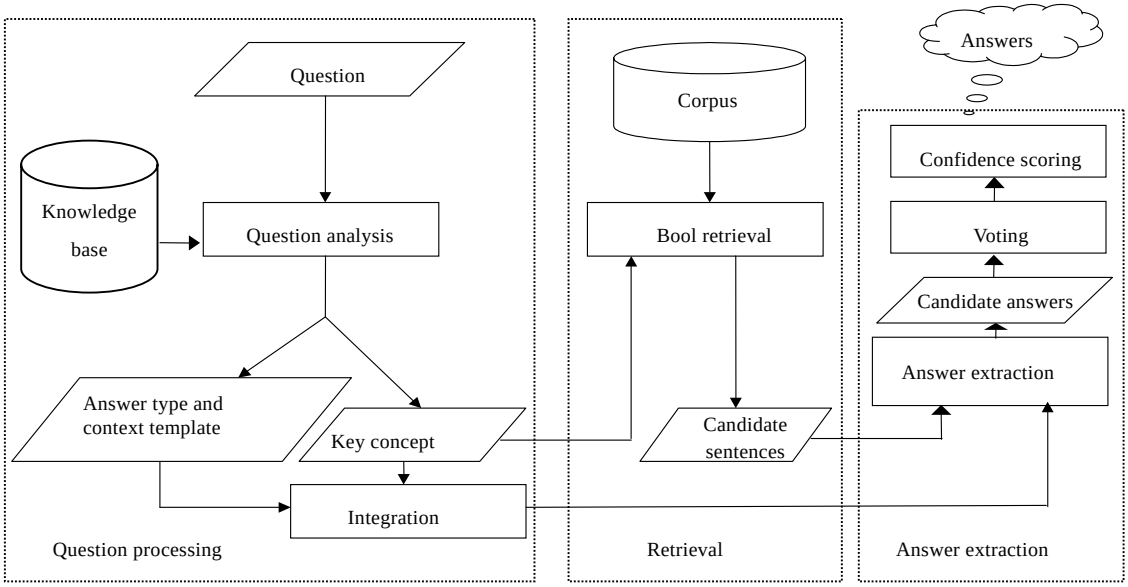

Fig.2 Flow chart of the online part of FDUQA system

图 2 FDUQA 系统在线部分流程图

图 3 是 FDUQA 系统处理问题过程的一个具体例子.在这个例子里,输入的问题是 What is the National Park in Utah?首先分析问题,确定问题的类型为“What VERB_BE1 NP1 in NP2?”,这里的 VERB_BE 是指 be 动词,NP 指名词词组.从知识库中找到这个问题的上下文模板和答案类型.这个问题的答案类型应该是名词词组(NP),它的第 1 个上下文模板表明答案可能出现在:Utah’s 〈Answer〉 National Park.其中〈Answer〉就是答案所在.然后在问题中提取出查询词 National Park 和 Utah,并从语料库中检索出所有可能包含答案的句子,再从中抽取答案.比如有这样一个句子“… A lightning-sparked fire in Utah’s *Zion* National Park spread out of control to 1,100 acres Tuesday ...”这个句子正好符合第 1 个上下文模板,Zion 又是一个 NP,符合答案类型.于是 Zion 就被抽取出来作为问题的答案.当然,在检索出来的句子群中可能会有多个句子符合模板匹配,这样就会找到不只一个候选答案.最后我们采取一种投票机制来找到最佳的答案.每个答案出现的次数就是它的票数,将票数最高的作为最佳答案.

限于篇幅,有关 FDUQA 系统的详细情况在此不一一说明.详情可参阅文献[6].

1.2.2 因 素

为计算置信度,我们考虑了系统处理问题过程中的 4 个因素:

- 答案评分

对于每一个问题,系统都可能会找到若干个答案.如何比较这些答案的好坏?这就要从以下 3 个方面为答案打分:

上下文模板(context template).对于某一类型的问题来说,其答案可能出现在不同的上下文中.在某些上下文模板中,答案出现的可能性相对较大,这类上下文模板就会获得较高的分数.

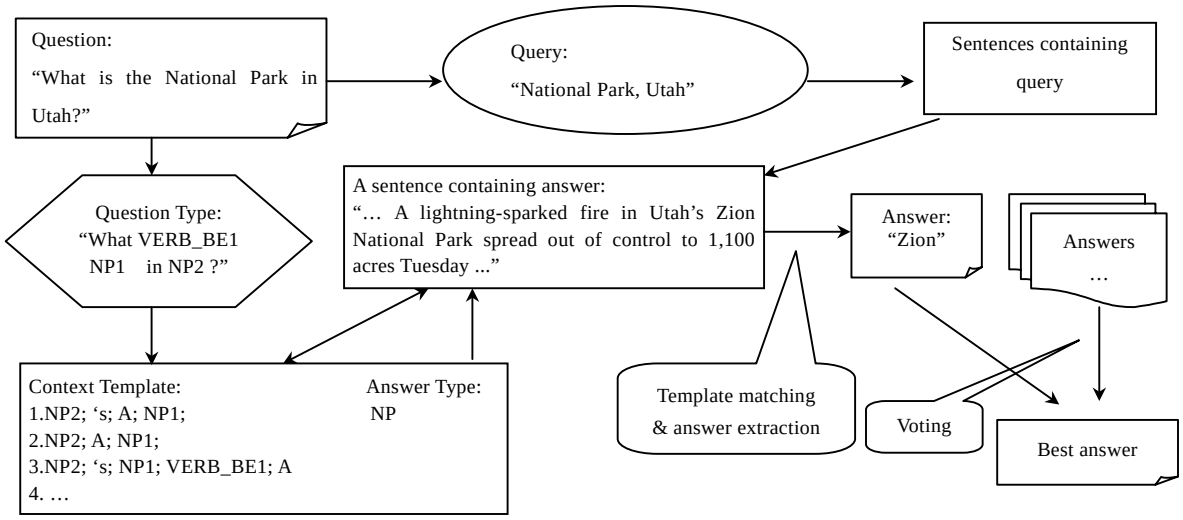

Fig.3 An example of question processing

图 3 问题处理过程实例

答案类型模板(answer type).一个问题可以有不同类型的答案,比如以 Who 开头的问题,答案类型可以是人名,也可以是组织名.而以人名作为答案的可能性要更大些,所以前一个答案类型模板的分数会比后一个要高.

查询词匹配(keywords matching).答案所在的句子可能并没有匹配所有的查询词.然而,查询词匹配得越多,答案就越好.因此这个分数表示了答案所在的句子与查询词的匹配程度.

一个答案的分数也就是关于这 3 个分数的函数:

$$S=f_{\phi}(\text{context template, answer type, keywords matching}),$$

从上式我们可以看出,答案的分数要视具体的问题而定,因此不能直接用来衡量置信度.

- 步数

在上一小节里,我们知道答案抽取的过程可以分为两步.在第 1 步找不到答案的情况下再进行第 2 步.由于第 1 步抽取使用的是严格的模板匹配,所以一步抽取出来的答案要比两步抽取出来的更为可靠.基于此,答案抽取的步数在一定程度上会影响到答案的正确性.

- 最佳票数

在投票机制中,每个答案出现的次数就是它的票数.一个答案出现得越多就越可能是正确答案.而作为最后输出的最佳答案,它的票数在衡量置信度的时候也具有一定的意义.

- 最佳得票率

然而,有时候最佳票数的多少并不能很好地说明问题.比如:对于问题 1,系统只找到了 1 个答案,并且这个答案在语料库中出现了 2 次.也就是说,这个答案就是最佳答案,且最佳票数为 2.对于问题 2,系统共找到了 3 个答案(A,B,C),其中答案 A 出现了 2 次,而答案 B 和 C 各出现了 1 次.那么答案 A 就是问题 2 的最佳答案,且最佳票数也为 2.虽然这两个问题的答案都具有同样的最佳票数,但显然问题 1 的答案正确度更高一些.为了能够对这类情况进行处理,我们定义了最佳得票率为最佳票数与所有答案的总票数的比,并把它也作为计算置信度时考虑的因素之一.

2 实 验

2.1 实验环境

我们使用 TREC10 的语料和问题作为训练数据.TREC10 的 QA 任务共有 500 个问题.对每个问题,我们从系统的处理过程中获取了上述 4 个因素,然后以此来训练最大熵模型.

另外,实验用的测试数据是 TREC11 的 500 个问题.在这 500 个问题中,FDUQA 系统答对了 124 道题.我们对每个问题的答案估计系统对它的置信度,并按照置信度从高到低排序输出的答案.我们希望在使用了基于最大熵模型的置信度评分算法后,可以将正确的答案尽可能地排在前面.

2.2 TREC11 QA任务的评估方式

在实验中,我们使用与 TREC11 的 QA 任务相同的评估方式.TREC11 定义了一种叫做"置信度权重分数(confidence-weighted score)"的量度标准.它类似于文本检索中的平均精度指标.具体表达式如下:

$$confidence-weightedscore=\frac{\sum_{i=1}^{500}\#correct\_up\_to\_question\_i\div i}{500},$$

其中*#correct_up_to_question_i* 是指从第 1 个问题到第 *i* 个问题的 *i* 个问题中系统答对的问题数.

可以看到,置信度权重分数在 0~1 之间变化,并且其最大值受问题回答准确率的约束.只有当所有的答案都正确时,该分数才为 1.而对于一定的准确率来说,正确的答案排得越是靠前,该分数就越高.

2.3 基准方法

方法 **1**. 直接按照问题的编号排序输出答案.这种方法完全不考虑问题本身以及系统处理该问题的过程.

下面介绍的两种方法分别都被其他参加 TREC2002 的 QA 系统使用过,为了与基于最大熵模型的算法作比较,我们将这两种方法作为实验中的另外两种基准方法.

方法 **2**. 以问题类型作为排序的依据.我们将问题分成 6 种类型:where,when/what year,what/which,who,how,以及其他.对每种问题类型,我们就系统对 TREC10 的 500 道问题的回答情况作了统计(见表 1).

Table 1 Precision of different question types

表 **1** 不同类型问题的回答准确率

Question type	where	when/what year	who	what/which	Other	How
Pricision	0.692	0.410	0.340	0.168	0.154	0

表 1 中,各种问题类型按回答准确率从高到低排列.在测试集上,为了将尽可能多的正确答案排在前面,我们也用与表 1 相同的顺序来排序输出答案.对于同一类型问题的答案则按问题编号排序.也就是说,系统对一个问题答案的置信度就相当于这个问题类型在训练集上的回答准确率.很容易理解的是,假如系统对某几类问题回答的准确率特别高,这种方法必定会取得不错的结果.

方法 **3**. 直接用答案评分来排序输出答案,将分数高的答案排在前面.这个方法将系统对答案的置信度的高低等同于答案评分的高低.但是正如前文中所述,答案评分应该是置信度所考虑的一个因素.仅以此来衡量系统对答案的置信度是有所偏颇的.

表 2 是系统处理两个不同的 when 问题时所涉及的各项因素的值.如果仅将答案评分作为答案的置信度,那么第 1 题答案的置信度应该高些.但实际上第 1 题答错了,而答案评分较低的第 2 题却答对了.第 1 题的最佳答案的评分只能说明它在这道题的所有 251 个答案中是最好的,不能光凭此来断定它的置信度会比第 2 题的答案要高.从表 2 也可以很直观地看到,如果再综合考虑其他 3 项因素,就不难得出第 2 题答案置信度较高的结论.

Table 2 An example of the values of factors

表 **2** 具体问题中各项因素的值

Question	Best answer	Correctly answered	Answer score	Step	Number of votes	Best vote ratio
When was the telegraph invented?	1 827	Wrong	5.0	2	3	3/251=0.012
When was the Oklahoma City bombing?	1 995	Correct	4.0	1	34	34/34=1.0

2.4 实验结果

2.4.1 基于最大熵模型算法的结果

我们使用基于最大熵模型算法来计算答案的置信度,然后以此对所有答案排序,从而得到置信度权重分数为 0.434.图 4 描述了用置信度排序后输出答案的分布情况.横坐标是答案个数 *n*,纵坐标是前 *n* 个答案的准确率.

从图中可以看到,前 8 个答案的准确率为 1,也就是说,前 8 个答案都是正确的.而前 100 个答案的准确率为 0.64,说明前 100 个答案中有 64 个正确答案.相对于总共 124 个正确答案,有一半以上的正确答案排在了所有答案的前 1/5 中.

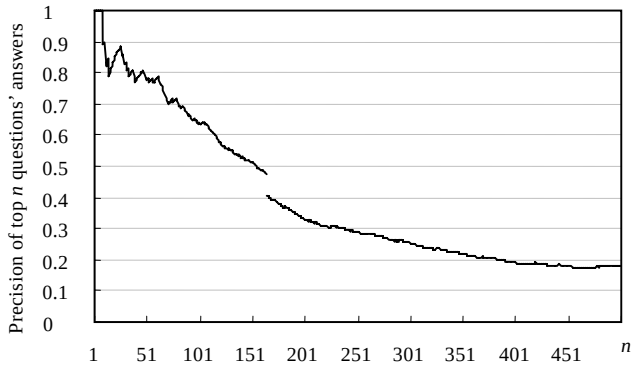

Fig.4 Distribution of answers when using ME-based algorithm

图 4 采用基于最大熵模型算法的输出答案的分布

从图中可以看到,前 8 个答案的准确率为 1,也就是说,前 8 个答案都是正确的.而前 100 个答案的准确率为 0.64,说明前 100 个答案中有 64 个正确答案.相对于总共 124 个正确答案,有一半以上的正确答案排在了所有答案的前 1/5 中.

2.4.2 结果比较

下面比较 3 种基准方法和基于最大熵模型算法的结果.表 3 显示了这几种方法的置信度权重分数.

Table 3 Comparison of different algorithms

表 **3** 基于最大熵模型算法和基准算法的比较

Algorithm	Confidence-Weighted score
Method 1	0.234
Method 2	0.261
Method 3	0.367
ME-Based algorithm	0.434

从表 3 我们可以看到,用最大熵模型计算出来的置信度显然优于 3 种基准方法.

图 5 描述了各基准方法输出答案的分布与基于最大熵模型算法的比较.其中横轴与纵轴的意义与图 4 相同.图中用粗实线表示采用最大熵模型算法的结果(prec_conf),细实线表示基准方法 3 的结果(prec_score),粗虚线表示基准方法 2 的结果(prec_qtype),细虚线表示基准方法 1 的结果(prec_qid).

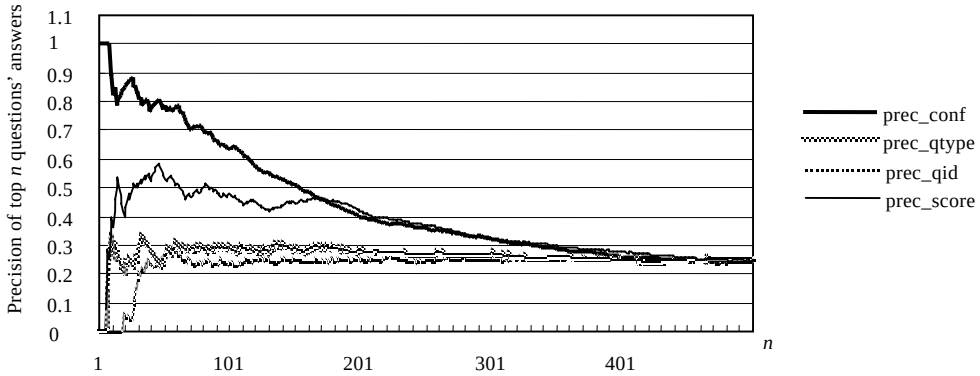

Fig.5 Distribution of answers when using different algorithms

图 5 各种方法的输出答案分布的比较

可以看到,由于基准方法 1 没有考虑任何影响答案置信度的因素,而只是按问题编号输出答案,因此在它的

结果中,前 18 个答案都是错的,即前 18 个答案的准确率为 0.在第 32 个答案以后,准确率总在 0.2~0.3 之间浮动.这是因为我们的系统在 TREC11 的 500 个问题上的精度是 0.248.可见正确答案在所有答案中的分布基本上是均匀的.这显然不是我们想要的结果,我们希望将正确的答案尽可能地往前排.

基准方法 2 按问题类型的准确率来输出答案,但只有当一个 QA 系统对某几类问题的处理能力特别好时,方法 2 才能表现出其优势.而我们的系统对不同类型问题的处理能力差别不是太大.因此,比较基准方法 1 和基准方法 2 的两条曲线可以看到,方法 2 只比方法 1 稍好一点.

基准方法 3 将答案评分作为答案的置信度,虽然它的结果明显比方法 1 和方法 2 要好,但它仍低于基于最大熵模型的算法.在第 170 个答案以后,方法 3 和最大熵模型算法的两条曲线几乎重合.这主要是因为如果答案的评分很低,我们的 QA 系统就会认为该问题没有答案.不过,当答案数小于 170 时,基于最大熵模型的算法显然比基准方法 3 要理想得多.这是因为我们的算法综合了问题处理过程中的 4 种因素.这种评分算法较之其他方法要合理许多.比如在表 2 的例子中,两题都是 when 类型的问题,而且第 1 题答案的评分高于第 2 题.如果用以上 3 种基准方法来计算置信度,结果都不会理想.但通过最大熵模型算法可以算出第 1 题答案的置信度为 0.338,而第 2 题答案的置信度为 1.0.

总之,采用基于最大熵模型算法的结果明显优于 3 种基准方法的结果.

3 总 结

当然,对一个 QA 系统来说,首要的任务是要获得较高的准确率.然而系统还应给出它对所作回答的置信度,以使用户了解系统对答案的准确性有多少把握.

为了更精确地计算这种置信度,我们设计了一个基于最大熵模型的算法.这个模型考虑了所有已知的因素,而把所有未知的因素排除在外.

这种方法为我们在 TREC11 的 QA 任务中取得了不错的成绩.尽管在所有参加的 67 个系统中,FDUQA 系统的精度仅排在第 30 位,但我们的置信度权重分数排到了第 13 位.“Overview of the TREC 2002 Question Answering Track”一文提及有些精度比我们高的系统最后分数却不如我们.TREC2002 的总结报告中也提到 FDUQA 系统是少数用自动学习方法来计算置信度的系统,并且获得了显著的成绩.TREC2002 中绝大多数的系统都使用问题类型来进行置信度评分,排名在我们系统之前的也只有 BNN 一家使用了经验公式.但显然在本文所讨论的算法中,仅答案评分一项就已涵盖了 BNN 的 3 个因素.因此他们的算法相当于本文实验中的方法 3.

FDUQA 系统在 TREC2002 中的表现以及其他学术报告对本系统的评价都证明了基于最大熵模型的置信度评分算法的优越性.在以后的工作中,我们还将进一步完善该算法,使其更好地为 QA 系统服务.

References:

[1] Voorhees EM. Overview of the TREC 2002. In: Voorhees EM, Buckland LP, eds. Proc. of the 11th Text Retrieval Conf. (TREC-11). Gaithersburg: NIST Special Publication, 2002. 115−123.

[2] Soubbotin MM, Soubbotin SM. Use of patterns for detection of likely answer strings: A systematic approach. In: Voorhees EM, Buckland LP, eds. Proc. of the 11th Text Retrieval Conf. (TREC-11). Gaithersburg1: NIST Special Publication, 2002. 325−331.

[3] Greenwood MA, Roberts I, Gaizauskas R. The University of Sheffield TREC 2002 Q&A system. In: Voorhees EM, Buckland LP, eds. Proc. of the 11th Text Retrieval Conf. (TREC-11). Gaithersburg: NIST Special Publication, 2002. 823–831.

[4] Xu J, Licuanan A, May J, Miller S, Weischedel R. TREC 2002 QA at BBN: Answer selection and confidence estimation. In: Voorhees EM, Buckland LP, eds. Proc. of the 11th Text Retrieval Conf. (TREC-11). Gaithersburg: NIST Special Publication, 2002. 96–101.

[5] Berger AL, Della Pietra SA, Della Pietra VJ. A maximum entropy approach to natural language processing. Computational Linguistics, 1996,22(1):39−71.

[6] Wu LD, Huang XJ, Niu JY, Xia YJ, Feng Z, Zhou YQ. FDU at TREC2002: Filtering, Q&A, Web and video tasks. In: Voorhees EM, Buckland LP, eds. Proc. of the 11th Text Retrieval Conf. (TREC-11). Gaithersburg: NIST Special Publication, 2002. 232−247.