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
myth1111@yahoo.com, archzhou@yahoo.com, xjhuang@fudan.edu.cn, ldwu@fudan.edu.cn

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。本文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，我们的 QA 系统用该算法计算每个问题答案的置信度，并依此排序，最后获得了不错的结果。

关键词：问题回答（QA），最大熵模型，信息检索

ME Based Confidence Scoring Algorithm for QA

You Lan, Zhou Yaqian, Huang Xuanjing, Wu Lide
Department of Computer Science and Engineering, Fudan University
myth1111@yahoo.com, archzhou@yahoo.com, xjhuang@fudan.edu.cn, ldwu@fudan.edu.cn

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

Keywords: Question Answering, Maximum Entropy Model, Information Retrieval

1. 引言

问题回答（QA）任务的目的是要为一个问题找到一个确切的答案。用户可以用自然语言向一个问题回答系统（QA 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一系列答案。通常情况下，QA 系统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”这一标准[1]。“置信度”综合考虑了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 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

*本项目受国家自然科学基金项目（60103014）和 863 计划（2001AA114120，2002AA142090）资助

题回答的置信度。也有一些系统就直接使用了答案评分的结果。

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。该算法包括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。笔者将在后面的篇幅里详细描述该算法的实现步骤以及实验的设计。

本文组织如下：第 2 部分主要介绍基于最大熵模型的置信度评分算法。其中 2.1 节描述了最大熵模型的原理，2.2 节详细分析我们提出的算法。第 3 部分主要介绍实验的设计并分析实验结果。最后一部分是对我们这方面工作的一个小结。

2. 置信度评分算法

2.1 最大熵模型

2.1.1 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[2]。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，且不受任何未知的因素的影响。

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。这样，概率 p(y|X)就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：

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

称 $f_i(X,y)$ 为最大熵模型的特征。其中，N 是训练样本集的大小。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：

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

可以求出上式的解为[2]：

$$p^*(y \mid X) = \frac{1}{Z(X)} \exp(\sum_i \lambda_i f_i(X, y))$$

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

其中 λ_i 是每个特征的权重。

2.1.2 建立最大熵模型

p(1|X)表示在因素向量为 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，因此我们可以用下式来计算置信度：

$$confidence = p^*(1 \mid X) = \frac{1}{Z(X)} \exp(\sum_i \lambda_i f_i(X,1))$$

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

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，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。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。

另外，各约束条件之间显然并非完全独立，它们对置信度的影响程度也各不相同。因此，我们尝试了几种不同的因素组合方式来训练最大熵模型。

2.2 问题回答过程因素

2.2.1 FDQA 系统介绍

为了更好地说明本算法中用到的各种因素，笔者首先简略地介绍一下我们的 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的“置信度”从高到低排序。类似于大多数 QA 系统，我们的系统主要由四个模块组成：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。（具体流程图见图 1）

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。每个问题类型又由三个部分组成：问题模板，答案类型模板（又称内部模板），上下文模板（又称外部模板）。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，首先采用严格的答案类型和上下文模板匹配方式，得到的答案是一步取得的。如果这样无法取得答案，再改用宽松的匹配方式，也即是内外两个模板一松一严，得到的答案就是两步取得的。

我们把上下文模板中的每一项，包括答案本身都看作一个概念。系统首先在目标句子中定位所有概念，然后尝试匹配模板，如果在第一或第二步匹配成功，就将答案抽取出来。

限于篇幅，有关 FDQA 系统的详细情况笔者就不一一说明了。详情可参阅"FDU at TREC2002: Filtering, Q&A, Web and Video Tasks"[3].

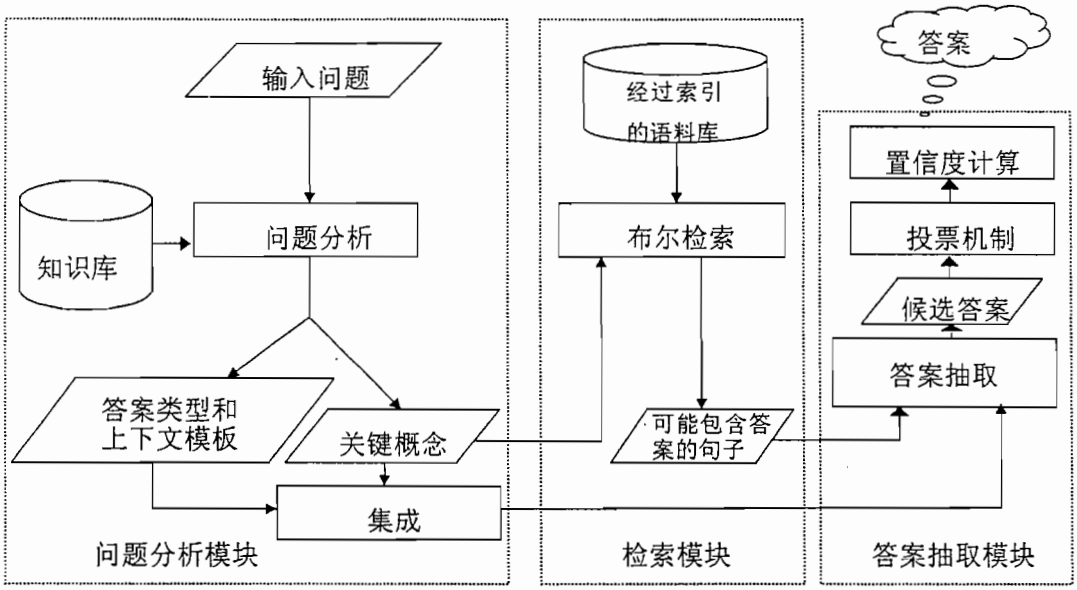


图 1：FDQA 系统在线部分流程图

2.2.2 因素

为计算置信度，我们考虑了系统处理问题过程中的 5 个因素：

- 答案评分：

为确定一个答案的分数，首先要就以下 4 个方面打分：上下文模板（context template），答案类型模板（answer type），句子匹配（sentence matching），概念匹配（concepts matching）。而答案的分数正是关于这 4 个分数的函数：

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

从上式可以看到，答案的分数是由具体的问题决定的，因此不能直接用来衡量置信度。

- 步数：

一个答案可能是一步抽取出来的，也可能是两步抽取出来的。由于第一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显然要比两步抽取出来的更可靠。

- 最佳票数：

对每个问题来说，在系统处理的过程中会找到许多的候选答案。我们采用了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。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：

最佳得票率也就是最佳票数与所有答案的总票数的比。

- 答案类型：

对于一类特定问题，它们会有一定的答案类型。比如“who”问题，答案往往是人。由于系统对各类问题的处理能力不同，因此答案类型也可以作为有关置信度的一个考量因素。

3. 实验

3.1 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 5 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，FDQA 系统答对了 124 题。我们对每个问题的答案估计系统对其的置信度，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。

3.2 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：

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

可以看到，置信度权重分数在 0 到 1 之间变化。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。

3.3 实验结果

3.3.1 基准方法

首先，基准方法 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。这样排序后，置信度权重分数为 0.234。

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，为了同基于最大熵模型的算法作个比较，我们将这两种方法作为实验中的另外两个基准方法。

方法 2 以问题类型作为排序的依据。我们将问题分成六种类型：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）。

表 1: 不同类型问题的回答准确率

问题类型	准确率
where	0.692
when/what year	0.410
who	0.340
what/which	0.168
其他	0.154

How	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，为了将尽可能多正确答案排在前面，我们也用与表 1 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。这样，得到最后结果的置信度权重分数为 0.261。这个分数比方法 1 的分数要稍高一些，但改善不多。这是因为我们的 QA 系统对各种不同类型问题的处理能力相差不多的原故。

最后，方法 3 是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，从而得到 0.367 的置信度权重分数。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。它的分数优于方法 1 和方法 2。但是正如笔者在前文中所述，答案评分应该是置信度所考虑的一个因素。所以仅以此来衡量系统对答案的置信度是有所偏颇的。

在后面的部分中，笔者将详细分析采用基于最大熵模型的置信度评分算法得到的结果。

3.3.2 使用基于最大熵模型算法

在训练最大熵模型的过程中，我们尝试了前文提到的 5 种因素的各种组合。最终发现，同时使用全部的 5 个因素并不能得到最好的效果。而在各种组合里，同时使用以下 4 个因素则能达到最佳效果：答案评分，步数，最佳票数，最佳得票率。

我们将应用全部 5 个因素的结果和只使用以上 4 个因素进行了对比（表 2）：

表 2：不同因素组合方式的对比

排序方式	置信度权重分数
置信度（使用 5 个因素）	0.426
置信度（使用 4 个因素）	0.434

从表 2 可以看到，使用 4 个因素的结果要优于使用全部的 5 个因素。在后面的叙述中，凡提到用基于最大熵模型算法计算的置信度都是是指只使用 4 个因素计算出来的置信度。

图2：采用基于最大熵模型算法的输出答案的分布

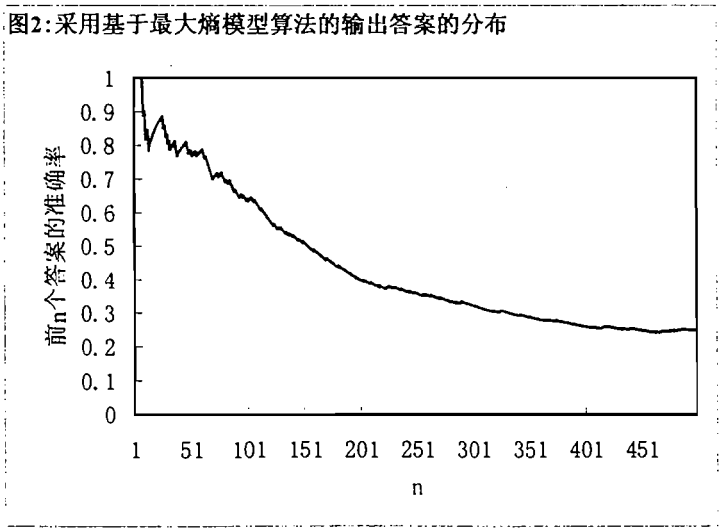


图 2 描述了用这个置信度排序后输出答案的分布情况。横坐标是答案个数 n，纵坐标是前 n 个答案的准确率。

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，就是有一半以上的正确答案排在了所有答案的前 1/5 中。

3.3.3 结果比较

下面，比较一下 3 个基准方法和基于最大熵模型算法的结果。（表 3）

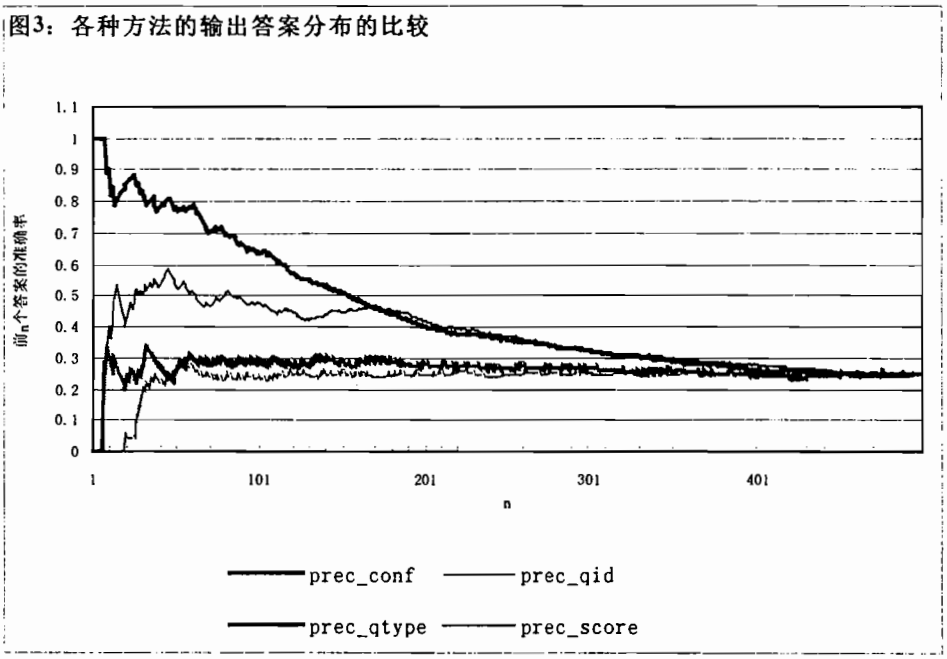
表 3：基于最大熵模型算法和基准算法的比较

输出答案排序方式	置信度权重分数
基准方法 1	0.234
基准方法 2	0.261
基准方法 3	0.367
基于最大熵模型算法	0.434

从表 3 可以看到，用最大熵模型计算出来的置信度显然优于 3 个基准方法。

图 3 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 2 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）。

图3：各种方法的输出答案分布的比较



可以看到，在基准方法 1 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，准确率则总在 0.2 到 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，这个精度并不理想。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。

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 比方法 1 稍好一点。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 才能表现其优势。而我们的系统显然

对不同类型问题的处理能力差别不是太大。

基准方法 3 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。

总之，采用基于最大熵模型算法的结果明显优于 3 个基准方法的结果。

4. 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，FDQA 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。“DRAFT Overview of the TREC 2002 Question Answering Track” [1] 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 FDQA 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。

FDQA 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。

参考文献

[1] Ellen M. Voorhees, "DRAFT Overview of the TREC 2002 Question Answering Track"

[2] Adam L. Berger, Stephen A. Della Pietra and Vincent J. Della Pietra, "A Maximum Entropy Approach to Natural Language Processing", Computational Linguistics Volume 22

[3] Lide Wu, Xuanjing Huang, Junyu Niu, Yingju Xia, Zhe Feng, Yaqian Zhou, "FDU at TREC2002: Filtering, Q&A, Web and Video Tasks"

[4] C.L.A. Clarke, G.V. Cormack, M. Laszlo, T.R. Lynam and E.L. Terra, "The Impact of Corpus Size on Question Answering Performance", 25th SIGIR

[5] Cody Kwok, Oren Etzioni and Daniel S. Weld, "Scaling Question Answering to the Web", 10th World Wide Web Conference

[6] Jennifer Chu-Carrolll, John Prager, Christopher Welty, Krzysztof Czuba and David Ferrucci, "A Multi-Strategy and Multi-Source Approach to Quesiton Answering"

[7] L. Hirschman and R. Gaizauskas, "Natural Language Question Answering: The View form Here", Natural Language Engineering 2001

[8] Marc Light, Gideon S. Mann, Ellen Riloff and Eric Breck, "Anlyses for Elucidating Current Question Answering Technology", Natural Language Engineering 2001

[9] S.D. Pietra, V.D.Pietra and J.Lafferty, "Inducing Features of Random Fields", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 4, pp. 380-