开放域问答系统研究综述 *

何靖 [1]，陈翀 [2]，闫宏飞 [1]

（1.北京大学，信息科学技术学院，北京，100871；2.北京师范大学，管理学院信息管理系，北京，100875）

E-mail: hj@net.pku.edu.cn, chenchong@bnu.edu.cn, yhf@net.pku.edu.cn

摘　要：尽管搜索引擎能够满足网络用户的很多信息需求， 但是还有很多它无法满足。 原因之一是它严格的用户界面： 它的输入是关键词， 它的输出是相关文档集。对于很多信息需求， 更加合适的提问和回答方式是自然语言。 开放域问答系统就是用于解决这一问题的。 本文主要介绍开放域问答系统的系统框架， 主要技术和评测方法。

关键词：开放域问答系统；问题分析；信息检索；答案抽取

A Survey: Open-domain Question Answering System

HE Jing[1], CHEN Chong[2], YAN Hongfei[1]

[1](Department of Computer Science and Technology, Peking University, Beijing 100871, China)

[2](Department of Information Management, Beijing Normal University, Beijing, 100875, China)

E-mail: hj@net.pku.edu.cn, chenchong@bnu.edu.cn, yhf@net.pku.edu.cn

Abstract: Though Web users can find relative information with search engine, it cannot satify all information needs. One reason is that its interface to users is rigid: keywords as query and documents as output. However, for some information need, it's more suitable to be queried and answered by natural language. Open-domain question answering system is designed to solve this problem. In this survey paper, we will introduce the framework and critical techniques for an open-domain question answering system and present how to evaluate it.

Keywords: Open-domain question answering, question analysis, information retrieval, answer extraction

1　概述

1.1　研究背景

随着网络数据的快速增长，从海量网络数据中获得相关信息成为一个巨大挑战， 搜索引擎在一定程度上解决了这个问题。在分析搜索引擎日志时发现，它包含一些自然语言表述的查询如“如何安装 RedHat9”。这表明互联网用户更加习惯于用自然语言来表达他们的某些信息需求。搜索引擎的输入是一组关键词，但是有时用户的信息需求很难用关键词确切地表达。同时，有时用户所需信息的粒度并不是一篇文档，而是一个描述性的段落、句子、结论、人名或数字等，但是搜索引擎对于一个查询返回的是一个文档集合，用户还需从中找出相关的内容。这表明现有的搜索引擎服务和用户的实际信息需求之间存在着两个方面的“鸿沟”：系统要求的关键词表达方式与用户自然表达方式之间的鸿沟和系统返回信

*本文受到国家自然科学基金（70903008，60933004），CNGI 搜索引擎项目（CNGI2008-122），863 课题（2009AA01Z143），软件开发环境国家重点实验室开放课题（SKLSDE-2010KF-03)及 973 计划海量项目（2005CB321901）的共同支持。

息的方式和用户需要返回的方式之间的鸿沟。如果能使用户以一种更加自然的方式和系统交互，用户可以自然而精确地表达他们的信息需求，系统能直接返回用户想要知道的内容，就能填平这条鸿沟。基于这样的需求，开放域问答系统成为信息系统领域中继搜索引擎之后的又一个热点。从技术层面来看，计算机处理能力的提升，信息检索、自然语言处理、人工智能等相关领域研究的发展也为问答系统的构建创造了条件。

1.2 问答系统的历史

问答系统并不是一个年轻的研究方向。五十年代，Alan Turing 提出了著名的图灵测试。该测试的目的是测试计算机是否具有智能。为了鼓励进行图灵测试的研究，1991 年 Hugh Loebner 设立了 Loebner Prize，用于奖励第一个通过图灵测试的系统。十多年来，出现了 PC Therapist, Albert 等优秀的聊天机器人系统，它们的一些技术如问句答案的模式匹配，对开放域问答系统很有借鉴价值。除此之外，还有一些基于知识库的问答系统研究[32,33]，包括基于本体的问答系统，受限语言的数据库查询系统，问答式专家系统等。这些系统受限于一定的知识领域和语言表达方式，不具有可扩展性。本文讨论的开放域问答系统和它们不同，具有两个特性： 1、它能够回答的问题不局限于一个或几个特殊的领域，而是不限定领域的；2、它是基于一套文档数据库（可以是新闻集合， 也可以是整个 Web）， 而且它只能回答那些答案存在于这个文档数据库中的问题。因此它是可扩展的，随着文档数据库的增加，它具有了更多的“知识”，就能回答更多的问题。

最早的在线问答系统是由美国麻省理工大学 Boris Katz 等人开发 START 系统(http://start.csail.mit.edu/)，它自 1993 年 12 月正式提供服务至今，已经回答了数以百万计的问题，问题的类型包括位置，电影，人物，文化，历史，艺术，环境，词典定义等。 AskJeeves 也是一个优秀的开放域问答系统（http://www.ask.com），和 START 不同，它返回的结果并不是一个精确的答案，而是包含答案的一系列段落。其他比较著名的在线系统还包括 Brainboost(http://www.answers.com)和 AnswerBus(http://www.answerbus.com)，它们返回的是包含结果的句子。

为了推动开放域问答系统的发展，信息检索评测组织 TREC（Text REtrieval Conference）自 1999 年开始，设立了开放域问答的评测任务，已开展了 10 次评测，成为 TREC 中历时最长的评测任务。另外的一些著名评测组织如 NTCIR 和 CLEF 也已经设置问答系统评测的任务。可见，问答系统的研究已在领域内受到非常强烈的关注。

1.3 开放域问答系统的通用体系结构

由于自然语言处理、信息检索、人工智能等相关领域技术的局限性，问答系统的回答能力也是有限的。Moldovan [25]根据问答系统的能力，把它由弱到强分成了 5 类：

- 能回答事实问题的系统：回答的内容是一个事实，可以直接在文档里找到，一般是一个词或者一次词组。

- 能回答具有简单推理问题的系统：回答的问题可能是文档里面的一个片断，需要系统简单的推理能力。
- 能够多文档信息综合回答的系统：需要从多个文档中分别找出答案并且以一定的方式进行组合展示给用户。
- 交互式问答系统：答案是上下文相关的，即和用户已经提问的问题和系统已经返回的结果有关系。
- 具有类推能力的系统：答案需要系统进行推理获得，可能无法在文档集合中直接找到。

现有的开放域问答系统的能力， 一般是介于前四类之间， 主要处理那些能够通过直接从文档集中抽取答案就能回答的问题。这些问题主要包括事实类问题，列表类问题，定义类问题，关系类问题等。

当前，不同的问答式系统会具有不同的体系结构。如[8]的系统包括主题提取，主题定义，段落检索，答案抽取四个模块， [28]的系统包括问句分类，文档检索，句子抽取，答案抽取，排序，消除重复答案等模块。虽然模块划分不同，但一般来说，所有的系统的模块都可以纳入三个大的部分，即问句分析，文档和段落的检索和答案的提取和验证。

问句分析部分所需要完成的功能包括问句类型分析，问句主题识别，问句指代消解和问句语法分析等。问句分类是问答系统中一个很重要的环节，它需要把问句根据它的答案类型分到某一类别中，之后的检索和提取会根据问句类别采用不同的措施。在现有问答系统的解决方案中，很多都根据精细问句类型和精细实体答案的识别的对应关系来提取答案，所以他们尤其重视问句分类的性能。找出问句的主题，可以帮助检索部分首先找出和主题相关的文档和段落，便于进一步后续的处理。在某些系统交互式地回答用户的提问， 因此用户的问题中会出现一些指代词，因此需要根据上下文明确指代词在问题中具体所指。有的系统通过对问句进行语法分析，匹配问句的语法结构和包含答案句子的语法结构。

文档和段落检索部分的功能是根据问句构造查询，利用一定的检索模型找到可能包含答案的文档或者段落。这里会涉及到的问题包括：采用什么样的信息检索模型，如何构造查询，如何对这些段落进行排序，如何追求查全率和查准率之间的折衷，检索阶段的性能和最终的系统总性能有什么样的关系等。

答案的提取和验证是问答系统的最后一个部分，它的输出就是问句的最终答案。 它分析检索获得的文档或者段落，从中提出能够回答问题的答案。在提取答案时，问句类型直接决定如何生成候选答案集合。另外，某些问题的答案可能存在于知识库中或者 Web 上，这一步中可以通过察看知识库或者 Web 对答案进行验证。三大模块之间的流程和关系如图 1 所示：

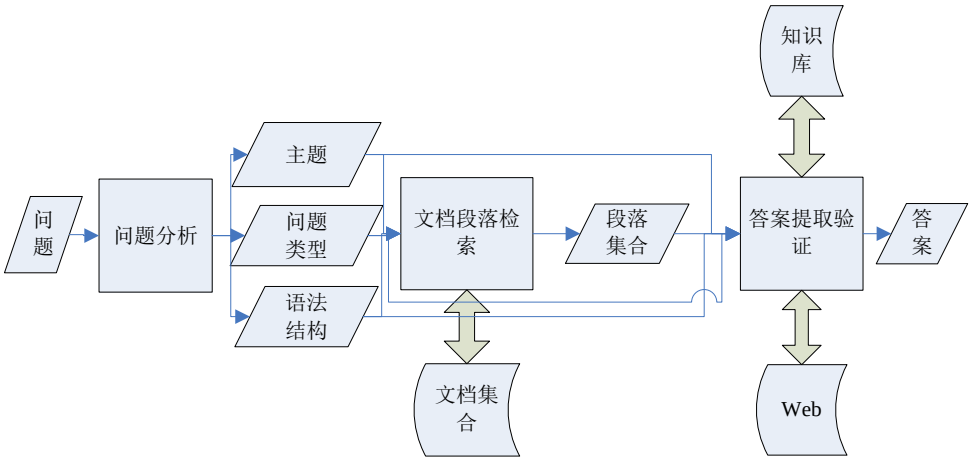

图 1 问答系统的通用体系结构

从上图可见，问题分析模块可以获得问句的主题、类型和语法结构。文档和段落检索模块从文档集合中获得相应的可能包括正确答案的段落集合。答案提取和验证模块根据前两个模块的输出以及通过知识库和 Web 上的知识获得最终的答案。下面就这三个模块进行详细的介绍。

2 问句分析

问句分析模块用于分析理解问题， 从而协助后续的检索和答案提取。 它的输入是由自然语言表述的问题， 输出是问句对应的答案类型和问句主题等。

2.1 问句分类

问句分类是根据问句所问的目标（答案）的类型对问句进行分类， 它是问句分析最重要的功能之一，因为答案类型直接影响后续步骤尤其是答案抽取的策略，比如对于问人物的问题，答案抽取模块首先就会把相关文档中出现的人物作为答案候选集合。最简单的问句分类通过疑问词直接决定问句的类型。在英语中，典型的有 5W1H（What, Who, When, Where, Why, How）的问题，在中文中，也有类似的“谁”，“哪儿”这样的疑问词。但这种方法粒度太粗，特别对 What，How 这样的疑问词，可以对应非常多的答案类型。 另外，有一些问句从句型上看是祈使句，不包含疑问词，如“列出长江流经的省份”，或者有一些问句包含了多个疑问词，如英语中含有定语从句，宾语从句的情况，这种方法无法处理这些情况。因此应该有更加精细的分类体系和相应的分类方法。一些研究者提出了问句的分

类体系结构，试图能够涵盖开放域中重要类型的问句，其中[34]的体系就是一个被广泛引用的问句分类体系（见表 1）。这个问句体系包括了 6 个大类，分别是缩略语，描述，实体，人物，地点，数量。为了能够更好的帮助找到正确的答案，在 6 个大类下面又分了 50 个小类，譬如在实体类里面又有动物，颜色，创造者等等，在数量类里面又有日期，距离，钱数等等。问句分类的任务就是通过分类算法，把一个问句分到这样一个分类体系结构的一个或几个类中去（某些问句比较模糊，可能属于一个以上的类别）。现有的问句分类的方法主要包括基于模式匹配的和机器学习分类算法两类。第一种方法是每一种问题类型会对应一个模式集合， 对于一个问句， 只要和某种问题类型对应的模式匹配， 就被认为是这种类型的问题。 第二种方法[30,34]类似于文本分类，它首先定义一个问题的特征集合， 这些特征可以包括：词，词组，表层词序列(n-gram), 词的解释，词的近义词，词的词性，语法树的词序列等。然后用一组（问题，类型）集合作为训练数据， 采用机器学习分类算法学习获得一个分类模型， 就可以对问句进行分类了。 [30]使用表层 n-gram 特征， 实验了 K 近邻， 决策树，朴素贝叶斯，支持向量机算法，实验结果表明支持向量机算法表现较好。[34]采用更深层次的特征，包括语法（词性，词组）和语义（解释，近义词）的信息，先用一个顶层分类器先把问句分到一个大的类别，然后根据分入的大类选用类内分类器把它分到小的类别，获得了不错的效果。

表 1 一种问题分类体系结构以及 TREC 问答任务中问题的分布

Class	#	Class	#
ABBREVIATION	18	term	19
abbreviation	2	vehicle	7
expression	16	word	0
DESCRIPTION	153	**HUMAN**	171
definition	126	group	24
description	13	individual	140
manner	7	title	4
reason	7	description	3
ENTITY	174	**LOCATION**	195
animal	27	city	44
body	5	country	21
color	12	mountain	5
creative	14	other	114
currency	8	state	11
disease/medicine	3	**NUMERIC**	289
event	6	code	1
food	7	count	22
instrument	1	date	146
lang	3	distance	38
letter	0	money	9
other	19	order	0
plant	7	other	24
product	9	period	18
religion	1	percent	7
sport	3	speed	9
substance	20	temp	7
symbol	2	vol.size	4
technique	1	weight	4

2.2 问句主题提取

问句分析的另一个主要方面是问句主题提取。 在后续的检索模块中， 需要选择问题中的一些关键词进行查询， 必要的时候会对查询进行调整， 但是无论如何， 都应该包含这个问题的主题。 通常可以通过对问句进行句法分析， 获得这个问句的中心词， 然后选取中心词和相关的修饰词成为问题的主题。如何选取合适粒度的中心词组成为这里的关键问题。Cui[7]提出了一种基于外部资源选取词组的方法。它把问句中的关键词提交给搜索引擎，从搜索引擎返回的答案中发现各种词的组合的点互信息，只有点互信息高于一定程度的中心词的组合才被认为是词组， 这个词序列就构成了问题的主题。

在一些系统的问题分析中还包含产生查询关键词，但是，由于关键词提取会依赖于检索模块所采用的模型，而且有的提取算法需要和检索模块的迭代过程从而和检索算法高度耦合，因此这里我们把这个子模块放在检索模块中介绍。

3 概述文档和段落检索

对于信息检索子系统， 最简单的方法是直接用已有的检索系统（如 Smart，Lemur，Lucene 等）或者搜索引擎（如 Google）对问题的非停用词进行全文索引，去掉问题中的停用词和问句相关的词（如疑问词生成查询， 把获得的文档或者段落作为答案提取和验证模块的输入，但这种简单的方法很难获得很好的效果。Thompson [5]通过实验，指出了文档检索的好坏会直接影响到问答系统的整体性能。当一个检索系统的查准率比较差的时候，可能会有大量的无关文档需要后续的模块处理，而一般来说，答案提取和验证模块需要比较复杂的自然语言处理的技术，因此大规模的无关文档会大大降低系统的效率。如果检索系统的查全率比较低，那么也就意味着有很多包含答案的文档或者段落没有被检索到。包括正确答案的文档或者段落越少，那么提取出正确答案的可能性也越小。在极端的情况下，如果所有包含答案的文档都没有检索获得，那么后续的模块无论如何也无法获得答案，因此在这一阶段，查全率比之查准率更为关键。这一模块需要选取合适的检索模型和查询，甚至需要强化现有的索引方式， 来获得较好的查准率和查全率。

3.1 检索模型的选用

首先我们讨论一下可以采用的信息检索模型。信息检索领域常用的模型包括布尔模型、向量空间模型、概率模型、语言模型等。其中布尔模型是最简单的一种，它把关键词组织成一个布尔表达式，使得文档中出现的关键词需要满足这个布尔表达式。它的优点在于简单高效以及可以通过查询直接控制返回的文档集的大小，而不需要像其余的排序模型那样选取一个截断阈值，但是它没有提供对文档和段落进行排序的功能。但是某些答案提取算法需要这样的功能， 因为某些算法会使用一个段落的相关性得分来估计这个段落中出现答案的可能性。一些搜索引擎会直接会提供布尔查询的界面（允许用户输入 AND，OR，NOT，括号等）。向量空间模型把文档和查询都表示成向量，根据查询和文档对应向量的相似度（通常是两个向量夹角的余弦值）对文档进行排序。概率模型估计计算文档和查询相关的概率，即计算 P(R|D,Q)，并按照相关性概率对文档进行排序。语言模型是把查询和文档分别表示成语言模型（即词或者 ngram 的多项分布）， 通过计算文档的语言模型到查询的似然或者两个语言模型之间的 KL 距离来估计两者之间的相关性。以上的四种模型均有问答系统采用。 [25,27]通过实验发现在问答系统的文档检索中，简单的布尔模型的效果和概率模型以及改进了的向量空间模型相当。

3.2 查询生成

无论采用何种模型，检索系统的输入应该不是一个问句，而是由关键词和词组组成的查询。最简单的转换方法就是把问句中的停用词去掉，其余的词作为关键词进行检索。但

是这样的方法存在着几个问题：首先，问句是可长可短的，当问题很长的时候，关键词会很多，若采用布尔模型，检索获得文档太少，查全率很低；当问题很短，关键词很少，检索获得文档太多，影响查准率， 不利后期处理。另一个问题普遍的存在与信息检索领域中：查询和文档的词有可能不匹配，即在查询和文档中可能使用不相同但具有相同语义的词。

对于查询松紧不确定的问题，需要对查询关键词进行调整，若关键词太多查询过紧，应该去掉一些。若关键词太少查询过松，就应该加上一些。对于查询文档不匹配的问题，可以通过把词形变化和意义接近词加到查询中来，同时这也是一种使得查询变松的方法。Moldovan [35]就采用这种迭代式调整技术， 多次查询，根据返回文档的多少，调整查询，决定是否增删关键词以及是否采用词形，句法或者语义级别的扩展形式。

以上的方法从问句已有的关键词出发。但是问答系统的目的是要找出一个问题的答案，而不是找和一个问题相关的文档，因此如果从一个问题推测它的答案中可能包含那些关键词， 用这些关键词来进行查询， 会得到很好的效果。 对于一类问题，系统可以从训练数据中学习获得这类问题的回答模式，根据这个模式对问句进行重写，构造包含答案关键词的查询。[1]就采用这种方法。首先，基于一个问题－答案对组成的数据集， 它统计问句中的 n-gram， 把那些频繁出现的称为问句词组。然后，对于每一个问句词组，统计答案中频繁出现构成模式的 n-gram，并通过一个过滤器过滤一些和内容有关的 n-gram（如名词词组）， 剩下的那些，称之为答案词组。 一个问句词组与其对应的一个答案词组集合组成了一组候选重写规则。在获得了一些重写规则以后，系统就可以根据这些重写规则构造查询，譬如有一个问句词组是“What is a”，对应的重写规则包括“is used to”，“according to the”，“to use a ”，“is a ”，“of a”，“refers to”，“used”，“refers”，“usually”等。当有一个问句“What is a computer”的时候，就可以构造查询“computer AND “is used to” ”，“computer AND “according to the”” 等等，就有可能找到正确的答案。这种方法的好处在于，它在构造查询的时候就预先考虑了答案可能包含的与内容无关只与问题形式有关的关键词，但是这种根据问句词组到答案词组的映射是表层的，可能存在相同的问句词组在上下文中会有不同的语义的情况，造成映射的错误。另外，这种方法获取的重写规则的完备性取决于训练数据的大小。

3.3 查询结果排序

问答系统中检索模块的结果应该是段落[27]，因为问题的答案一般是局部的文本（词组，句子等），返回整个文档会降低答案抽取模块的准确率和执行效率。

最直接查询结果排序方法是根据文档的自然段落，采用一种信息检索的模型（如BM25），按照查询和段落的相关性，对所有的自然段落打分排序，但是这类算法并没有考虑到关键词的位置信息，出现了关键词越靠近的段落则越有可能包含问题的答案。[27]一文通过实验的方法细致地考察了各种段落检索的算法。实验结果表明，基于密度的算法可

以获得比较好的效果。所谓基于密度的算法，就是查询关键词在某个段落里的出现次数和邻近程度会决定这个段落的相关程度。表现比较好的段落检索算法包括：

MultiText 算法：这种算法会倾向于挑选比较短的包含尽可能多的高信息量（对应于比较大的 IDF 值）关键词的段落。该算法会检索出文档中查询关键词密集出现的段落，这种段落的定义是从一个查询关键词开始，到一个查询关键词结束，中间包含了尽可能多的查询关键词。因此，这种段落是不定长的，这里定义相关性和段落的长度成反比，因此倾向于获得更短的段落。

IBM 的算法：这是 IBM 在参与 TREC 评测中提出的算法。它提取了一些相关性的特征。包括：匹配的关键词特征，就是指同时在查询和段落文本中出现的关键词的 IDF 值的和。词典匹配关键词特征，就是在查询中的关键词，虽然没有在段落中出现，但是关键词的同义词在段落中出现了，这些关键词的 IDF 值的和。不匹配关键词特征，就是虽然在查询中出现了，但是在段落中没有出现的关键词的 IDF 值的和。分散程度特征，就是在匹配的关键词之间的间隔。聚类词特征，即同时在问题和段落中都邻近出现的词的数目。最后通过线形叠加累积这些特征对于相关度的影响。

SiteQ 算法：该算法规定检索获得段落是由 m 个句子构成的。对于每个句子，获得的分数由两部分组成，一部分是所有关键词的 IDF 值的和，另一部分是相邻关键词的距离的平方倒数和 IDF 和的乘积。段落的得分是 m 个句子得分的叠加。

考察上述的三个算法，虽然在设计和实现细节上有很大的差异，但是都使用了 IDF 值的和以及引入了邻近关键词之间的距离。所以，在进行问答式系统的段落检索时，基于密度的算法是有效的。

上面的算法只考虑了独立的关键词及其位置信息，没有考虑关键词在问题中的先后顺序，也没有考虑语法和语义的信息。考虑语法信息，最直观的想法就是把问句和答案都解析成语法树，从两者语法树的结构中找出一些相关性的信息。Cui[9]提出了一种基于模糊依赖关系匹配的算法。这种算法需要把问题和答案都解析成为语法树，并且从中提取词与词的依赖关系。由于同样的问句可能具有语义上相同但是语言表述上不同的回答形式，如果只考虑完全相同类型的依赖关系会导致查全率降低，因此需要度量依赖关系之间的匹配程度。在训练数据上，通过统计不同依赖关系在问句-答案中的共现，可以获得不同依赖关系的匹配程度。在检索排序的时候，通过依赖关系匹配度，就可以获得问句和候选答案句子在语法上的匹配度。实验结果表明，这种方法检索的效果比基于密度算法的 SiteQ 为优。

3.4 增强索引的功能

大多数的问答系统都基于传统信息检索系统的全文索引。 但是，和一般信息检索系统只需要处理关键词不同的是，问答系统需要处理更多的语法、语义信息。 因此，有一些系统把这些信息也放入到索引中去，以提高效率。 [36]把一些关键词或者词组的属性放入索

引，在组成布尔查询的时候，不仅需要包含某些关键词，还有答案或者关键词的属性要求。譬如，对于一个问时间的问题， 就要求返回的的段落中包含时间。 [2]把查询变成一个结构化的查询，表达查询词和段落中应该包含的某些词的属性。进一步的，还有一些系统索引了句子中不同的词或者词组的关系，[17]索引了段落中句子解析称三元组的形式， 索引这样的三元组。 [4]还索引了句子中词和词组的语义关系。

4 答案提取和验证模块

一个问答系统通过问句分析和文档段落检索模块可以获得问题答案的段落集合，答案提取和验证块从这些段落中获取正确的答案。为了提取答案， 一般首先需要通过精细实体识别，从段落中识别中符合问题类型的答案作为候选集合， 然后利用表层字符串特征，或者语法、语义、逻辑上的关系或者模式匹配来获取答案。 通常，获得答案还可以通过一些外部的资源，譬如 Web，WordNet，知识库等进行验证，对结果进行重排序。

4.1 生成候选答案集合

问题分析模块已经获得问句分类。观察表 1 中的问题类型可以发现，除了描述类（Description）和其他少数几个小类如 expression，order 等，大多数的问题类型对应的答案都会比较短，他们可能是实体名如人名、机构等，也可能是属于一定语义范畴的名词如食物、技术、树木、花朵，也可能数字度量如距离、速度等。对于这些类型的问题，可以通过找出相应类型的词、词组或者片断来回答。对于传统的命名实体如人名机构地点时间等，在自然语言处理领域已经有比较深入的研究， 只要通过一个训练获得的模型如隐马尔科夫模型(HMM)或者条件随机场模型(CRF)就可以从文本中识别。对于一定语义范畴的名词（如鸟类， 花朵等），可能需要事先收集获得属于该语义范畴的具体名词集合，只要在段落中寻找这些集合中的词作为候选答案即可。为了构建具体名词集合，可以借助机器可读的词典如 WordNet。但是 WordNet 主要是一个概念的词典，包含的具体名词会很少，不能够满足回答这类问题的需求。为了获取更多的具体名词，有一种方法是通过一些〈概念性名词，具体名词〉作为训练的种子，从大的文档集或者 Web 中找到这种连接概念性名词和具体名词的模式，再根据这种模式提取更多的具体名词，多次迭代可以发现更多的（概念名词，具体名词）对和相应的模式。这种方法称之为自举(Bootstrap)方法。 [23]就是采用这样的方法，在 WordNet 的概念名词的本体结构之下再建立一层具体名词的本体结构， 这层具体名词结构有助于从检索的段落里获得语义上匹配的短语。另一种简单的方法是自动地提取 Web 资源（如 wikipedia, CIA 等)中具体名词列表。对于可以用正则表达式表示的那种答案，可以手工定制一些正则表达式包含尽可能多的情况，譬如距离一般可能是一个数字后面跟上一个表示距离的单位。以上的三类识别方法， 能够覆盖大多数问题类型。通过在文本中匹配相应类型的短语，就构成了候选答案集。

4.2 答案提取

在获得候选答案集合以后，如何获得匹配问题的最佳答案？首先我们介绍三种可以利用的不同层次的特征以及相应的方法，最后介绍一组对针对答案提取建立的统计学模型。

4.2.1 基于表层特征的答案提取

早期的大多数系统只使用表层特征来抽取答案。 一类表层特征是答案周围段落的一些特征， 譬如段落和查询的相关程度，所有查询词之间的距离，查询词和候选答案的距离。一般来说，段落相关程度过高，查询词之间以及查询词和候选答案之间距离越接近，则该候选答案越可能是问题的答案。另一个特征是该候选答案出现的次数。对于一个比较大的文档集，一个问题的答案可能反复出现，出现的次数越多，则它越可能是正确答案[22]。同理，该问题的答案可能在更大的数据集（如 Web）反复出现，因此可以通过搜索更大数据集提取在该数据集上的候选答案，从而根据总体的重复度估计它的正确性，这是一种广泛采用的答案验证方法。可以通过一个模型综合这两类特征，获得候选答案的得分。

4.2.2 通过关系抽取答案

表层特征没有考虑语法、语义的因素，因此很有可能出错。[19]估计这种基于实体识别和表层特征的方法的性能上限是 70%。比如对于两个不同的问题“青蛙吃什么动物？”和“哪些动物吃青蛙？”，通过问句分析，它们对应的问句类型都是生物。 通过文档和段落检索， 由于查询关键词均为“青蛙”，“吃”，。检索出来的文档包括这样的一些句子：“成年的青蛙主要吃昆虫，偶尔也会吃小鱼，蚯蚓和蜘蛛”，“鳄鱼一般会吃水边的动物，譬如鱼，蛇，青蛙，乌龟和一些哺乳动物”。因此，一个基于表层特征的系统，是无法区分这两个问题的区别的。因此这类方法具有局限性。为了克服上述缺陷，不少研究者提出了不同的改进。START 系统[17]采用的办法是把问句和文本中的句子转换成三元组的形式，三元组基本的构成是〈主语，动词，宾语〉，去掉了句子中的一些修饰成分，譬如上面场景中的问句变成的三元组就是〈青蛙，吃，什么〉和〈什么，吃，青蛙〉，与第一个问句相匹配的文本三元组应该是〈青蛙，吃，昆虫〉，〈青蛙，吃，小鱼〉等，与第二个问句相匹配的文本三元组应该是〈鳄鱼，吃，青蛙〉，〈蝙蝠，吃，青蛙〉，很容易的，就可以从文本三元组中获得答案而不会产生混淆了。

另一种解决这种混淆的方法是采用逻辑建立问题到答案的关系[24]，在历届 TREC 问答系统评测中表现最好的语言计算公司（LCC），就是采用这样的方法。逻辑表示是介乎于于句法解析表示和深层语义表示之间的一种表示形式，它可以通过解析获得的句法树通过一些规则计算获得，表达了主语，宾语，前置词，复杂的名词性短语，附属的形容词或副词等。由于自然语言的复杂性，存在非常多这样的转换规则，但是通过实验，发现 10-90 的规律，即可以通过 10 条最频繁使用的规则覆盖 90%的情况。转换成逻辑表示以后，通过一个逻辑证明机，借助一些外界的知识如果可以通过答案来证明问题，那么这个答案就是正确答案。

4.2.3 通过模式匹配抽取答案

另一种方法是基于句子级别的模式匹配来进行答案抽取的， 模式可以通过手工定制或机器学习获得。一般来说，手工定制的模式扩展性和覆盖率都比较低，因此主流的方法是采用学习训练数据的方法。下面以[7]为例说明一下这种方法的过程。

该文提出了一种软模式的方法，来处理定义类的问题。它首先对答案语句进行词性标注（POS）。因为模式只是一种回答的方式，而需要尽量少的训练集的内容，因此需要替换包含内容语义的词汇为它的词性。以查询关键词为中心取一个窗口，设窗口的长度为 2w+1，其中在关键词前面有 w 个词，在关键词后面也有 w 个词：$< slot_{-w}, slot_{-w+1}, \cdots, slot_{-1}, 查询关键词, slot_1, \cdots, slot_w >$。

考虑在该窗口中词出现的两种规律，即绝对位置和相对位置的分布规律。统计绝对位置的信息，就是把窗口中第 i 个位置称为第 i 个槽（slot），考虑第 i 个槽的词汇出现的可能性，即 $\Pr(token_u \mid slot_i)$。统计相对位置的信息，就是考虑前后相邻词的相互依赖关系，统计 bigram 的情况，即 $\Pr(token_i \mid token_{i-1})$。

$$\Pr(token_u \mid slot_i) = \frac{f(token_u)}{\sum_{token_v IN slot_i} f(token_v)}$$

$$\Pr(token_i \mid token_{i-1}) = \frac{f(< token_{i-1} token_i >)}{f(token_{i-1})}$$

其中 $f(token_u)$ 是指该词在第 i 个位置出现的计数， $f(< token_{i-1} token_i >)$ 是指 $token_{i-1}$ 和 $token_i$ 连续出现的计数， $f(token_{i-1})$ 是指 $token_{i-1}$ 单个出现的计数。通过这些信息，就可以计算候选答案语句和模式的匹配程度。相应的，也是分别计算绝对位置的匹配程度 Pa_weight_{Slots} 和相对位置的匹配程度 Pa_weight_{Seq} 综合获得最后的总匹配程度。

$$Pattern\_weight = \frac{Pa\_weight_{Slots} \times Pa\_weight_{Seq}}{fragment\_length}$$

$$Pa\_weight_{Slots} = \Pr(S \mid Pa) = \prod_{i=-w}^{w} \Pr(token_i \mid slot_i)$$

$$Pa\_weight_{Seq} = (1-\alpha)\Pr(left\_seq \mid Pa) + \alpha \Pr(right\_seq \mid Pa)$$

$$\Pr(left\_seq \mid Pa) = \Pr(token_{-w}, token_{-w+1}, \cdots, token_{-1} \mid Pa)$$
$$= \Pr(token_{-w}) \Pr(token_{-w+1} \mid token_{-w}) \cdots \Pr(token_{-1} \mid token_{-2})$$

$$\Pr(right\_seq \mid Pa) = \Pr(token_1, token_2, \cdots, token_w \mid Pa)$$
$$= \Pr(token_1) \Pr(token_2 \mid token_1) \cdots \Pr(token_w \mid token_{w-1})$$

其中 $Pattern\_weight$ 表示候选答案语句和模式的匹配程度， $fragment\_length$ 所取得窗口的大小， $\Pr(left\_seq \mid Pa)$ 和 $\Pr(right\_seq \mid Pa)$ 分别代表左半部分相对位置匹配程度和右半部分相对位置匹配程度。通过以上的方法获得的模式匹配度和内容匹配度相结合，就可以获得相关性。一般可以选择相关性程度比较高的句子作为候选答案。

4.2.4 答案提取中的统计模型

相比于信息检索和自然语言处理领域中的其他问题， 问答系统由于发展时间不长及其本身的复杂性， 使得少有理论性较强的数学模型来对它进行建模。 近年来， 有研究者开始尝试用统计模型对答案提取模块进行建模。 这些统计模型比较通用， 可以利用以上描述的三类特征中的一类或者多类。 这里介绍两个有代表性的建模方法。

第一种模型是噪音信道模型[10]。该模型把问句看成是目标信息，把答案看成源信息，假设源信息需要通过一个包含噪音的信道， 现在需要估计 $\Pr(q|S_{i,a})$， 也就是一个包含候选答案的句子 $S_{i,a}$ 是变成问句 q 的的概率。 这里的问句和答案句子都表示成关键字及其类型（词性）的序列， 把噪音建模成一组随机操作，比如添加，删除，替换，对齐词或词组等。 因此该模型就是通过估计的是通过一组随机操作把 $S_{i,a}$ 变成 q 的可能性来近似一个包含答案的句子能回答问题的可能性。各种操作发生的概率可以通过一组训练数据（问题-答案对集合）训练获得。

另一种模型是采用无向图模型对它建模[18]。 图模型是近年机器学习领域非常热门的一个模型族。该模型同时考虑了答案和问题的相关性以及答案之间的相关性。对于一个问题的一组答案{A_1, ……, A_n}， 用一组二元变量 S={S_1, ……, S_n}来表示每个答案是否是正确答案， 若 S_i=TRUE， 则 A_i 就是正确答案， 反之， S_i=FALSE, A_i 就不是正确答案。Pr(S)表示用 S 这组二元变量的取值来判定答案正确与否的可能性，因此 S_0=argmax Pr(S)就是一种判断答案正确与否的最佳方式。Pr(S)由答案和问题的匹配程度以及答案之间的相关程度共同决定， 表示为：

$$\Pr(S) = \frac{1}{Z} \exp\left(\sum_{i<j} \theta_{ij} S_j S_j + \sum_i \theta_i S_i\right)$$

其中 θ_{ij} 表示答案 A_i 和 A_j 的相关性， θ_i 表示答案 A_i 与问题的相关性， 而这里答案之

间或者问题答案的相关性都也是有一组特征线性加权获得的，各个特征的权重是通过一组问答对训练数据训练获得的。

5 问答系统中的其他方法

以上介绍的问答系统体系结构的三大部分，可以回答很大部分的问句。这些研究都是通过从文档数据库中找到答案的内容来回答问句的。其实，除了文档数据库以外，还有一些其他的资源，可以在这些资源中更加方便地找到问句的答案，这些资源包括：网上问答库，wikipedia，wordnet 等。另外，对于某些特殊的问题，可以采用比较特殊的方法来提高它们的性能。

5.1 问答系统中的外部资源

事实上，在传统体系结构的三大部分中，已经有很多都利用了外部资源。譬如在问句到查询的转换中，我们采用 WordNet 做了查询扩展；在答案的提取和验证中，我们采用 Web 数据进行答案的验证。下面我们讨论这些外部资源的进一步利用，以及其他可用的外部资源。

5.1.1 问句答案库

问句答案库对问答系统的构建具有很大的帮助。一方面，它可以被用作学习问答系统中所需机器学习模型（如答案抽取中的答案排序模型，用于问题和答案匹配的模式等）的训练集，另一方面也可以直接从中找出相同语义的问题所对应的答案来回答问题。有些研究者致力于把网上的现有资源自动组织成这样的问答库的形式。这样的工作，其实是通过对文档集进行离线的处理，从而可以提高在线回答问题的效率。Cong[6]发现，很多网络论坛中存在很多这样的问题－解答对，因此希望把它们提取出来。首先它通过标注序列学习(Labeled Sequence Learner)的方法建立一个问题模式集合，论坛中能够和这些模式匹配的句子被认为是问题。论坛中一个问题的答案，如果存在的话，总是存在于它的后续帖子中的，为了找出答案，它构建了一个由后续帖子中的段落作为节点构成的图，相似的节点（是否相似以低于一个 KL 距离的阈值判断）之间存在有向边，边的权重是由两个节点的 KL 距离以及目标节点的相关性（相关性由问题和段落的距离以及作者的权威性加权和获得）决定的。在这个图的基础上，以段落和问题的相关性作为初始权重，使用类似于 pagerank 的随机游走，可以获得每个节点的权重，权重高的就是相应问题的答案。通过这样的方法，就可以从论坛出提取一些问题－答案对组成问答库。

其实很多现有的网络服务直接包含这样的问答库，如常见问题档案馆（http://www.faqs.org），Yahoo! 回答 (answers.yahoo.com) 和百度知道（http://zhidao.baidu.com）等。对于一个新的问题，如何在已有的问答库中找到匹配的问题答案对呢？最直接的方法是比较问句和问句的相似度，但是问句都很短，而且可能同样语义的问题会采用不同的词汇、句式表达出来，因此仅仅比较问句相似度可能会有很大的

误差。而在已回答的问题中，答案会提供比问句多得多的信息，如果把整个问题-答案对统一起来考虑，很有可能会得到更好的结果。[16]都采用了按照文档，问题-答案对和问题相似度叠加的方法，以获得比较好的性能。在问答社区里面，由于有些用户不习惯于搜索以前问过的问题，因此，同一语义的问题有时候是重复存在的，可以利用这些冗余的问题-答案对，发现同样词义不同表述的问题之间的联系。[37]在社区问答系统中，采用类比推理的方法从不同质量答案集中确定最好的答案。[38]利用分类的方法来确定问题和答案对。[39]在论坛中，采用基于分类的序列模式方法首先确定问题，然后在同一问题线程形成的图中，依照传播方法来确定答案。[15]把两个问句看作是不同表达方式的语句，计算由其中一个“翻译”到另外一个的概率。 为了计算这种翻译的概率，就需要估计词与词之间进行翻译的概率。 它首先通过比较答案的相似度，从问答库中找出可能同义的问答对。这些同义问答对的问题构成了一个估计翻译概率的训练集， 可以把它看成是机器翻译中多语言平行语料库。通过这个训练集，就可以估计出不同词汇之间转换的可能性，他们的转移概率定义为：

$$P(t|s)=\lambda^{-1}\sum_{i=1}^{N}c(t|s,J^i)$$

$$c(t|s,J^i)=\frac{P(t|s)}{P(t|s_1)+P(t|s_2)+\cdots+P(t|s_2)}\#(t,J^i)\#(s,J^i)$$

其中t和s表示两个词，$P(t|s)$表示从s转移到t的概率，λ是一个归一化因子，J^i表示一对相同语义的问答对，$c(t|s,J^i)$ 表示在J^i中s转换为t的可能性，$\{s_1,s_2,\cdots,s_n\}$表示出现在J^i中的词汇的集合，$\#(t,J^i)$和$\#(s,J^i)$分别表示t和s在J^i中出现的计数。可见，这是一个迭代的过程，初始化的转移概率是随机数，可以证明，这个过程是收敛的，最后可以获得恒定的转移概率。再通过训练获得词与词的转移概率以后，其实就在一定程度上得到了词和词之间的语义相似度，从而可以计算新的问句和已知问答对的相似度了。相似度的计算方法如下：

$$sim(Q,D)=P(Q|D)=\prod_{w\in Q}P(w|D)$$

$$P(w|D)=(1-\lambda)P_{ml}(w|D)+\lambda P_{ml}(w|C)$$

$$P_{ml}(w|D)=\sum_{t\in D}P(w|t)P_{ml}(t|D)$$

其中$sim(Q,D)$就表示问题和已知问答对的相似度，$P(w|D)$是由已知问答对推出某一个词的可能性，为了防止某些词在训练集中没有和问答对里的词形成相关性而导致总相关性为零，$\lambda P_{ml}(w|C)$起到了平滑的作用，$P_{ml}(w|D)$的表达式中就用到了训练中计算获得的转移概率。

这种基于翻译模型的方法考虑到词与词之间语义的相关性，允许使用不同的词表示相同的语义，因此效果会比简单的直接检索关键词好。实验证明，这样做的效果会比语言模型，Okapi BM25 和空间向量模型都好。[29]进一步发展了一种方法， 把问题和问答对的相似度定义为问句翻译概率， 同时也考虑了问句和问答对答案之间的相关性，采用语言模型的查询似然的方法计算， 然后把两者进行线性叠加得到最终的相似性。

5.1.2 其他外部资源

在问句分类中，我们介绍了一种分类体系结构。但是，也有的研究者认为，这种结构有点不伦不类，其中有的类别表示的是一种语义上的信息，但其他的一些类别表示的是结构形式上的信息。Pasca [26]把答案类型按照在 WordNet 中的语义类来进行划分，在答案提取的时候，要求答案与这种答案类型相匹配。

在 Web 上，存在着大量的整理好的结构化的数据，这些数据一般通过表格或者模板类网页的形式展现出来。由于它们具有比较固定的格式，比较容易从它们的格式中理解它们的语义，因此，这些数据可以事先抓取下来作为本地的知识库。这些知识库可以组织成为〈实体，属性，值〉这样的三元组的形式，我们可以通过分析问题的语法结构明确问题是不是也能够重写成这样的三元组的形式，如果可以，则做一些语义上的扩展并且从知识库中查询，就有可能获得正确的答案。由于这些数据来源比较权威，语义信息明确，因此可以在回答中赋予比较高的置信度。

在回答定义类的问题时，除了以上所说的采用模式的方法以外，也是有很多外部资源可以利用的。譬如在 WordNet 中，很多词汇已经有了非常好的定义可以直接拿来使用，而且词在这个语义网络中所处的拓扑结构位置也包含了它的定义信息。在 Web 上存在一些类似于百科全书的网站，如 Wikipedia 等，可以利用这些资源获得定义。另外，把查询关键词输入搜索引擎，也可以大致的获得这个关键词的一些定义信息。当然当需要输入搜索引擎的时候，就需要结合利用第五节中叙述的模式的方法了。Cui [7]比较了这三种方法对于回答定义类问题的效果，比较的结果是百科类网站结果优于 WordNet 的结果，而 WorNet 的结果优于搜索引擎返回的结果。很显然，前两者都是花费大量的人工整理的，在准确性方面比较强。而与 WordNet 相比，百科类网站包括了一些具体事物的定义，而且它的定义不是像 WordNet 中那样是一个简单的义项，而会是比较详细的介绍，因此信息比较全面，回答的质量也比较高。

5.2 寻找特殊类问题的解决方案

考察理解 TREC 中问答任务的结果，可以发现，各种不同类型的问题的回答效果差别很大，比如实体类的问题，回答的效果很好，但是描述类问题，原因类问题的效果并不是很好。因此，某些研究主要致力于某些特殊类问题的解决，它们一般是在传统的处理流程之外加上几个特殊的问题处理模块，当发现问题类型和处理模块可以处理的类型相匹配的

时候，就用该模块处理。主要研究的问题类型包括，“谁”问题（Who），原因问题（Why），方式问题（How）和定义类问题(Who, What)等。

5.2.1 人物问题

问“谁”的问题其实有两种截然相反的类型：第一种的典型例子是“谁是北京大学校长”，这种问题的已知是一个概念，未知的是这个概念的一个实例；第二种的典型例子是“谁是许智宏”，这种问题的已知是一个实例，未知是这个事例所隶属的一个概念。很显然，如果能够从文本中预先提取出概念－实例对，那么可以帮助回答这类问题。[11]一文为了提取这种概念实例对，考虑了英语中的两种语言现象，一种语言现象是在一个概念后面直接跟上一个实例，如“president Lincoln”，这种现象在中文中也同样存在，如“国家主席江泽民”，“北京大学校长许智宏”；另一种语言现象是同位语，如“Bill Gates, CEO of Microsoft”。该文通过构造两个正则表达式涵盖了这两种语言现象。但是，这里还存在一个准确率的问题，即并不是所有能够和这种模式匹配的语言现象肯定是人物的实例和概念，为了提高准确率，该文又提取了一些特征，如模式的可信度，概念词的结尾，概念词的类型，实例词的特征等等，标注了一些满足模式文本属于或者不属于概念－实例对，通过机器学习的方法训练出一个分类器，从而对满足模式的文本进一步做分类。这样，就可以在问题到来之前把文本中所有的概念－实例对都提出来组织成为知识库的形式，当遇到这类问题的时候，跳过问答系统的传统解决过程，直接到知识库里查找。

5.2.2 原因类问题

原因类问题也是颇具难度的一类问题。在很多情况下，原因和结果是通过动词相连接的，譬如 cause, lead to, bring about 等是最常见也是最明显的包含因果关系的动词，另外还有一些动词在语义中已经包含了结果，譬如 kill 当中包含了 death 的结果，dry 包含了 dryness 的结果等，还有一些动词甚至包含着事件的内容，如 poison 一个词表明结果是 death，手段是 with poisoning。[12]希望能够找出表示一定因果关系的动词以及它们的模式。该文首先借助 WordNet 寻找一个具有因果关系的名词对，方法是在 WordNet 的一个名词 A 的定义中如果出现“A cause B”或者“B caused by A”这样的短语，那么认为 A 和 B 具有因果关系，按照这种方法就可以收集这样的一个具有因果关系的词汇的集合。然后，通过这个集合可能找到包含了一定因果关系的动词，方法是在文档集中检索具有因果关系的词汇对，对连接它们的动词进行排序，获得出现频率比较高的动词作为候选动词。下面就要知道这些动词的适用范围，因此在语言中往往有搭配的概念，某个词和某些词在一起出现表达了因果关系的意义，而和另外一些词出现可能会表示另外的含义。在此，为了区别不同的名词，该文依旧采用 WordNet 中的词的分类，把名词分为 9 种类型：实体，心理活动，抽象名词，状态，事件，动作，组织，过程和现象。这样，对于〈原因，动词，结果〉这三元组，每一项都会有一个取值。通过标注一些训练数据，就可以训练一个分类器（该文使用的是决策树算法 C4.5），根据取值可以知道在某个情境下，这是否表示一个因果关系。对于是因

果关系的文本，就可以用于回答因果类的问题。

5.2.3 方式和性状问题

问方式或者性状的问题（How）的解决方法和上述的方法是大同小异的，一般来说，对于这类问题，需要寻找的关系是动词和描述这个动作方式的副词或者副词词组。[13]采用机器学习的方法在文本中寻找满足描述方式关系得内容，提取的属性包括：在训练集合中的计数；在语法树中副词上层的结构，即该副词是修饰动词，名词或者形容词；作为副词的可能性；动词副词之间的距离；副词之前的词的词性；副词之后的词的词性；该词是否以“ly”结尾。通过一些训练数据，学习这些属性的分布，获得一个简单贝叶斯的分类器，从而判断文本中所有这类关系，帮助回答关于方式的问题。

以上的解决特殊类问题的方法其实都是大同小异，他们首先把这类问题对应到一种关系，然后通过机器学习的方法从文档集中找出满足这类关系的文本内容，从而和问题相匹配。所不同的是不同的关系需要根据关系的特有性质提取一些特殊的特征，并且采用合适的机器学习算法利用这些特征分类。

6 问答系统的评测

为了评价一个问答系统，和评价一个信息检索系统类似的， 需要一套测试集和一套评测指标。本节首先介绍 TREC 中的问答系统评测， 它是影响力最大的一个开放于问答系统的评测。 然后介绍一些重要的评测指标以及评测中的一些研究问题。

6.1 TREC的问答系统评测

在信息处理系统领域，很多有影响力的评测通过建立一套测试集来评测系统的好坏，推动领域的发展。譬如 MUC 对命名实体识别系统的评测，SUMMAC 对自动摘要的评测，SIGHAN 对中文分词的评测等。同样，在信息检索领域，TREC 评测是最为权威和具有影响力的。TREC 的是文本检索会议（Text REtrieval Conference）的简称，它是 1992 年由美国标准技术协会(NIST)发起举的办。1999 年，为了促进开放域问答系统研究研究的进展，该会议开始组织问答系统的评测。早期的评测（TREC8 和 TREC9），问题一般只包括事实性的问题，是从微软的百科全书 Encarta 和在线问答系统 START 中提取的。当时对于答案的要求不高，要求返回答案串的长度不得超过 50 字节/250 字节（分别是两个子任务），只要在答案串中包含了正确答案就可以了。从 TREC2001 开始，把整个问答任务划分成三个子任务：主任务，列表类任务和上下文相关任务。其中主任务包含事实性的问题和定义性的问题，列表类任务要求给出一系列答案构成的列表，上下文相关任务是把问题集结成为一个个的主题之下。从 TREC2003 开始，把任务分成主任务和段落任务，其中段落任务沿袭以前的传统，要求返回一定长度限制（250 字节）以内的答案串，而主任务中又有三类问题，即事实类问题，列表类问题和定类问题，对于前两类问题，要求回答的答案是准确

的字符串，不得包含其他的成分。到了 2004 年，一个明显的变化是所有的问题都是上下文相关的了，也就是集结于一个主题之下的。2005 年在以前的基础上增加了两个子任务，即段落检索子任务和关系类问题子任务，前者的提出是为了评估中间过程即段落检索模块的性能，后者的提出是为了鼓励对多个实体之间关系问题的研究。2006 年增加了复杂式交互式问答的评测。 2007 年数据集增加了博客数据的内容。 近年来在 TREC 问答类任务中表现比较好的研究机构包括语言计算公司（LCC），新加坡国立大学（NUS）等。

从 TREC 问答任务的发展历史，我们可以发现，评测和系统性能确实是一个互相促进的关系，最初的评测只要求能够包含答案的文本段落就可以了，而且问题类型只是最简单的事实类的问题，随着参加评测系统的性能的提高，系统都能获得比较好的这方面的性能了，因此，新的问题类型如定义类问题，列举类问题，关系类问题都加入评测，答案也由一个文本段落变成完全匹配的字串，问题的难度提高了，答案的要求也提高了。

6.2 问答系统评测指标

在 TREC 的问答任务中，主要任务的问题分为三类，即事实类问题，列表类问题和定义类问题。对于这三类问题，分别存在不同的评判方法和指标。

对于事实类问题，要求回答给出答案和它的出处（该答案从哪篇文档获得）。答案正确当且仅当答案的字符串和标准答案字符串完全匹配并且出处是属于合法的出处集合的。如果答案正确但是出处不正确，称这个答案为“不被支持的”；如果答案是包含了标准答案串的一个串或者只有包含标准答案串的一部分，那么认为这个答案“不精确匹配”；其他的情况就直接称答案为“不正确”。另一种情况是某个问题本身在文档集中是找不出正确答案的，这时候系统应该返回 NIL，如果系统返回的是 NIL，那么认为这个结果是正确的，如果系统还返回一个答案和相关出处，那么认为这个结果是“不正确”的。回答事实类问题的总体性能通过准确率（accuracy）来进行判断，即在参加评测的系统对每个事实性问题给出一个答案， 依据给出答案的正确的比例来评价系统的好坏。在 TREC8 和 TREC9 上， 曾经使用 MMR(返回的第一个正确答案的序号的平均值)来评价事实类问题的好坏。

对于列表类的问题，要求返回的格式和事实类问题是一样的，不同在于它需要返回多个这样的答案，因此需要把系统返回的结果集合和标准答案集合做一个比较，这和传统信息检索里面返回结果的评测非常相似，不同的在于信息检索返回的是文档集合，这里返回的是答案集合。类似的，通过查准率和查全率，可以判断对于列表类问题回答的好坏程度。查准率定义为返回正确的答案占返回答案数的比例，查全率定义为返回正确的答案占所有正确答案数的比例。为了统一表示查准率和查全率，采用这两者的调和平均数 F 表示。

对于定义类（其他类）问题，评测方法会复杂一些，与事实类和列表类的问题不同，它返回的结果是由很多文本的片断组合而成的。因此，不能用简单的字符匹配的方法对它进行评测，当前对于这类问题的评测主要还是采用人工的方法。TREC 对于这类问题评测

的基本流程表述如下：首先由各个参赛队提交结果，由工作人员从所有的结果中提出能够回答问题的一些“得分点”，并且把得分点分为两种，一种是包含了关键信息的“关键得分点”（vital nugget），另外一些是正确但是并不非常重要的“非关键得分点”（okay nugget）。对系统的评分主要由系统命中的关键得分点的数目和答案的长度综合而成。和信息检索评测类似的，也计算查准率和查全率，不过这两个概念和原始的概念会有一些出入，定义如下：

$$recall = \frac{|match\_vital\_nuggets|}{|vital\_nuggets|}$$

$$precision = \begin{cases} 1 & length < okay\_length \\ 1 - \dfrac{length - okay\_length}{length} \end{cases}$$

$$okay\_length = C \times |total\_nuggets|$$

$$F = \frac{(\beta^2 + 1) \times precision \times recall}{\beta^2 \times precision + recall}$$

其中 $match\_vital\_nuggets$ 表示在系统回答中能找到的“关键得分点”集合，$vital\_nuggets$ 表示总共“关键得分点”的集合，它们两者的商就是系统可以找到的关键得分点占总的比例。$length$ 表示返回答案的长度，C 是一个常数，这里取 100，表示一个得分点允许回答的字节数，$total\_nuggets$ 表示总共“得分点”的集合。$okay\_length$ 是一个对于一个问题的合理的回答长度，给每个“得分点”100 个字节的长度限制。如果回答不超过这个长度限制，那么认为准确率是 1，单纯从覆盖率来测试性能，如果超过这个长度限制，那么就会扣分。F 就是查准率和查全率的加权的调和平均值，β 取值为 5，即认为 $recall$ 更重要。

6.3 问答系统评测的研究

上面以 TREC 的评测为例，介绍了评测问答系统的方法。但是其中对于定义类问题的评测，并不是自动的，需要大量的手工参与。手工参与的评测一方面需要耗费大量的人力，另一方面也有可能带入一些主观的因素。因此，对于这类问题的评测，有的研究者试图通过自动的方法完成。

Lin [21]提出了一种简单的评测方法。它也需要事先构建一个“关键得分点”的集合，对于每个得分点，和系统返回的答案相比较，计算连续最大长度匹配。对于匹配的词。如

一个得分点是“A B C D”，系统返回是“B E B C D F”，那么最长长度匹配的串就是“B C D”。对于匹配串中的每一次词，计算一个分值，分值计算方式可以是计算匹配串内每一个词的 IDF 的叠加，因为 IDF 越大说明这个词汇越不可能出现，如果匹配说明系统的效果越好。最后叠加所有“得分点”的得分获得总得分。通过把该方法和手工评测方法比较，比较对各个参赛队排名后的序列相似度（Kendall’s τ）验证了这种方法的有效性。在实际评测的过程中发现， 不同的评审员对于同一个答案段落的“关键得分点”和“非关键得分点”的意见会有出入， 为了解决这个问题, Lin [20]借用自动摘要领域的一种办法， 对每个得分点给予一个权重， 这个权重是它被认为是“关键得分点”的次数和被认为“关键得分点”次数最多的“得分点”的次数的商， 因此， 任何一个“得分点”的权重都是 0 到 1 之间的。

当前， 组织进行问答系统评测的组织包括 TREC, NTCIR, CLEF, DUC 等， 每年都会有几十个组织参加这样的评测， 显示研究者对这个问题的广泛的兴趣和关注。

7 概述小结

本文主要介绍了开放域问答系统的历史起源和发展，应用背景，采用的方法和技术以及研究的主要方向。

开放域问答系统并不是一个新的问题和挑战，在计算机诞生之初已经被图灵提出来了。近年以来，随着信息科学技术的发展，尤其是网上信息的爆炸，信息的查找和获取需求的增大，以及信息检索和信息提取技术的发展，都推动着这一领域的发展。TREC 组织的问答任务评测，已经成为 TREC 保留时间最长的评测任务之一，可见它的重要性和困难程度。

大多数的开放域问答系统，都包括问题分析，文档段落的检索，答案的抽取和验证这三个顺序的过程。问题分析的主要任务包括指代消解，问句分类。文档段落检索就是要从问句到查询并且根据查询从文档集中得到可能包含正确答案的文档集或者段落集。答案的抽取和验证就从候选的文档集或者段落集中提取正确的答案。

除了主流的方法之外，还有一些其它的研究，主要包括：利用外部资源如 WordNet，Web 数据或者常见问题解答库帮助提高问答系统的性能，研究特定类型问题的特定解决方法，采用系综方法构造问答系统等。

系统设计和系统评测是互相促进的关系，合理的评测指标可以帮助设计出更加符合用户需求的系统，优秀的系统可以获得好的评测方法的认可。因此，问答系统的评测也是一项很重要的工作，TREC 就做出了大量的贡献。它从 1999 年开始对问答系统进行评测，改变评测的子任务和评测指标，使之逐渐接近现实系统的要求。有一些研究者对 TREC 评测提出了建议，比如提出了对于定义类问题的自动化评测方法代替 TREC 的手动评测。当前的问答式系统， 对于比较简单的问题比如问句较短的事实性问题， 已经表现的比较令人满意， 但是比较复杂的问题如关系型问题， 或者问句中出现比较复杂的关系， 则效果还

不理想， 因此它还有非常大的发展空间和潜力。

参 考 文 献

[1] Eugene Agichtein, Steve Lawrence, and Luis Gravano. Learning to find answers to questions on the web. ACM Trans. Internet Techn, 4(2):129－162, 2004.

[2] Matthew W. Bilotti, Paul Ogilvie, Jamie Callan, and Eric Nyberg. Structured retrieval for question answering. In SIGIR ’07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pages 351－358, New York, NY, USA, 2007

[3] Jennifer Chu-Carroll, Krzysztof Czuba, John M. Prager, and Abraham Ittycheriah. In question answering, two heads are better than one. In HLT-NAACL, 2003.

[4] Jennifer Chu-Carroll, John Prager, Krzysztof Czuba, David Ferrucci, and Pablo Duboue. Semantic search via xml fragments: a high-precision approach to ir. In SIGIR ’06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 445－452, New York, NY, USA, 2006.

[5] Kevyn Collins-Thompson, Jamie Callan, Egidio Terra, Clarke, and Charles L. A. The effect of document retrieval quality on factoid question answering performance. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Posters, pages 574－575, 2004.

[6] Gao Cong, Long Wang, Chin-Yew Lin, Young-In Song, and Yueheng Sun. Finding question-answer pairs from online forums. In SIGIR 08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 467－474, New York, NY, USA, 2008.

[7] Hang Cui, Min-Yen Kan, and Tat-Seng Chua. Unsupervised learning of soft patterns for generating definitions from online news. In Stuart I. Feldman, Mike Uretsky, Marc Najork, and Craig E. Wills, editors, Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004, pages 90－99. ACM, 2004.

[8] Hang Cui, Keya Li, Renxu Sun, Tat seng Chua, and Min yen Kan. National university of singapore at the trec 13 question answering main task. In In Proceedings of the 13 th TREC, 2005.

[9] Hang Cui, Renxu Sun, Keya Li, Kan, Min-Yen, and Chua, Tat-Seng. Question answering passage retrieval using dependency relations. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Question answering, pages 400－407, 2005.

[10] Abdessamad Echihabi and Daniel Marcu. A noisy-channel approach to question answering. In ACL ’03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, pages 16－23, Morristown, NJ, USA, 2003.

[11] Michael Fleischman, Eduard H. Hovy, and Abdessamad Echihabi. Offline strategies for online question answering: Answering questions before they are asked. In ACL, pages 1－7, 2003.

[12] Roxana Girju. Automatic detection of causal relations for question answering. In Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering, pages 76－83, Morristown, NJ, USA, 2003.

[13] Roxana Girju, Manju Putcha, and Dan Moldovan. Discovery of manner relations and their applicability to question answering. In Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering, pages 54－60, Morristown, NJ, USA, 2003.

[14] Jing Xiao Hang Cui, Tat-Seng Chua. A comparative study on sentence retrieval for definitional question answering. In in the proceeding of SIGIR Workshop on Information Retrieval for Question Answering, pages 43－50, 2004.

[15] Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. Finding similar questions in large question and answer archives. In Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 – November 5, 2005, pages 84 – 90. ACM, 2005.

[16] Valentin Jijkoun and Maarten de Rijke. Retrieving answers from frequently asked questions pages on the web. In CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management, pages 76 – 83, New York, NY, USA, 2005.

[17] Boris Katz and Jimmy Lin. Selectively using relations to improve precision in question answering. In In Proceedings of the EACL-2003 Workshop on Natural Language Processing for Question Answering, pages 43 – 50, 2003.

[18] Jeongwoo Ko, Eric Nyberg, and Luo Si. A probabilistic graphical model for joint answer ranking in question answering. In Proc. 30st SIGIR, pages 343 – 350, New York, NY, USA, 2007.

[19] Marc Light, Gideon S. Mann, Ellen Riloff, and Eric Breck. Analyses for elucidating current question answering technology. Nat. Lang. Eng., 7(4):325 – 342, 2001.

[20] Jimmy Lin and Dina Demner-Fushman. Will pyramids built of nuggets topple over? In Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, pages 383 – 390, Morristown, NJ, USA, 2006.

[21] Jimmy J. Lin and Dina Demner-Fushman. Automatically evaluating answers to definition questions. In HLT/EMNLP. The Association for Computational Linguistics, 2005.

[22] Jimmy J. Lin and Boris Katz. Question answering from the web using knowledge annotation and knowledge mining techniques. In CIKM, pages 116 – 123. ACM, 2003.

[23] Gideon S. Mann. Fine-grained proper noun ontologies for question answering. In COLING-02 on SEMANET, pages 1 – 7, Morristown, NJ, USA, 2002.

[24] D. Moldovan, S. Harabagiu, R. Girju, P. Morarescu, F. Lacatusu, A. Novischi, A. Badulescu, and O. Bolohan. LCC tools for question answering. In Text REtrieval Conference (TREC) TREC 2002 Proceedings. Department of Commerce, National Institute of Standards and Technology, 2002.

[25] Dan Moldovan, Marius Pasca, Sanda Harabagiu, and Mihai Surdeanu. Performance issues and error analysis in an open-domain question answering system. ACM Transactions on Information Systems, 21(2):133 – 154, 2003.

[26] Marius Pasca and Sanda M. Harabagiu. The informative role of wordnet in open-domain question answering. In Proceeding of ECACL 2001 Workshop on WordNet and Other Lexical Resources, 2001.

[27] Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In Proc. 26st SIGIR, pages 41 – 47, New York, NY, USA, 2003.

[28] Jinxi Xu, Ana Licuanan, and Ralph Weischedel. Trec 2003 qa at bbn: Answering definitional questions. In Proceedings of TREC 2003, 2004.

[29] Xiaobing Xue, Jiwoon Jeon, and W. Bruce Croft. Retrieval models for question and answer archives. In Proc. 31st SIGIR, pages 475 – 482, New York, NY, USA, 2008.

[30] Dell Zhang and Wee Sun Lee. Question classification using support vector machines. In SIGIR, pages 26 – 32. ACM, 2003.

[31] Y. Zhou, X. Yuan, J. Cao, X. Huang, and L. Wu. FDUQA on TREC 2006 QA track. In Text REtrieval Conference (TREC) 2006 Proceedings. Department of Commerce, National Institute of Standards and Technology, 2006.

[32] Gary G. Hendrix, Earl D. Sacerdoti, Daniel Sagalowicz, and Jonathan Slocum. Developing a natural language interface to complex data. ACM Trans. Database Syst., 3(2):105–147, 1978.

[33] W.A. Woods. Progress in natural language understanding – an application to lunar geology. In AFIPS Conference Proceedings, 1973

[34] Xin Li and Dan Roth. Learning question classifiers. In COLING, 2002.

[35] Dan I. Moldovan, Sanda M. Harabagiu, Marius Pasca, Rada Mihalcea, Roxana Girju, Richard Goodrum, and Vasile Rus. The structure and performance of an open-domain question answering system. In ACL, 2000.

[36] Dragomir R. Radev, John M. Prager, and Valerie Samn. Ranking suspected answers to natural language questions using predictive annotation. In ANLP, pages 150–157, 2000.

[37] Xin-Jing Wang, Xudong Tu, Dan Feng, and Lei Zhang. Ranking community answers by modeling question-answer relationships via analogical reasoning. In Proc. 32st SIGIR, pages 179-186, 2009.

[38] Liangjie Hong,Brian D. Davison. A classification-based approach to question answering in discussion boards. In Proc. 32st SIGIR, pages 171-178, 2009.

[39] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun. Finding question-answer pairs from online forums. In Proc. 31st SIGIR, pages 467-474, Singapore, July 2008.