清 华 大 学

综 合 论 文 训 练

题目：基于双向 Attention 机制的中文问题答案抽取方法研究

系　　别：计算机科学与技术系

专　　业：计算机科学与技术

姓　　名：周建宇

指导教师：徐华　副教授

2017 年 6 月 13 日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签　名：_____导师签名：_____日　期：_____

中文摘要

机器问答作为自然语言处理领域中最重要的研究方向之一，一直得到计算机科学界的高度关注。机器问答也被学界一直认为是下一代搜索引擎的发展趋势，高效、精准的自动问答对信息的高效获取和传播具有重要意义。

自计算机诞生依赖，对机器问答的研究就从未间断。机器问答的核心是自然语言处理，其发展方向也随自然语言处理技术的发展而不断更新。从早期的基于计算语言学的统计模型发展为如今的基于数据驱动的深度学习模型，问答效果也在不断提升。目前学界绝大部分研究都是基于英文问答的，而中文问答领域的研究与应用仍有很多不足。

问答的种类繁多，本文专注于根据文本并从中抽取问题答案（也称作机器阅读理解）的方法研究。本文借鉴了目前该领域应用效果最好的基于双向 Attention 机制的英文问答（阅读理解）算法，并将其加以改进和优化，以应用到中文问答场景。为了完成这一目标，本文的主要工作有：

1. 实现了基于双向 Attention 机制的英文问答算法。
2. 设计并实现了基于翻译机制的可应用于中文问答场景的双向 Attention 算法。
3. 设计并实现了基于中文训练语料库的中文双向 Attention 算法。
4. 对比了基于翻译与基于中文训练语料库的两种 Attention 算法在不同中文问答场景下的优劣并分别对两种算法进行了改进优化。
5. 实现了基于双向 Attention 算法的中文问答平台，该平台支持用户上传和编辑存在问题答案的文本，平台可基于该文本针对用户问题产生答案。

关键词：问题答案抽取；中文问答；双向 Attention；机器阅读理解

ABSTRACT

As one of the most important research field for Natural Language Processing(NLP), Question Answering has always been a hot topic in computer science. Question Answering is also regarded as the next generation search engine. Offering precise answer effectively has a great significance on the effective acquirement and spread of information.

Ever since the birth of computer, the research for Question Answering has never been stopped. The core technique of Question Answering is NLP. As a result, the development of Question Answering is closely related to the improvement of NLP. From the initial statistical language computing method to the deep learning method, the performance of Question Answering task has been improved greatly. However, most research of Question Answering lies on English field, so there are still a lot of word needed to be done in Chinese Question Answering.

Question Answering is a broad field with various types. To be more specific, this paper aims at the research of extract answers based on given contexts, which is also known as Machine Reading. This paper learns from the most advanced method for English Machine Reading, which is known as the bi-directional attention flow, and propose some improvements, to make it better for Chinese Machine Reading. To achieve this goal, this paper has the following main contributions:

1. We implement an English Question Answering algorithm based on bi-directional attention flow mechanism.
2. We design and implement a Chinese Question Answering algorithm based on translation and bi-directional attention flow mechanism.
3. We design and implement a Chinese Question Answering algorithm based on original Chinese training corpus.
4. We compare these two Chinese Question Answering algorithm in different application scenario and propose improvements respectively.

5. We implement a bi-directional-attention-flow based Chinese Question Answering platform, which supports users upload and edit contexts and answer questions based on them.

Keywords： Answer Extraction; Chinese Question Answering; Bi-directional Attention Flow; Machine Reading

目 录

中文摘要 I

ABSTRACT II

第 1 章 引 言 1

1.1 研究背景 1

1.1.1 问答概述 1

1.1.2 问答发展历程 1

1.1.3 问答系统与问题分类 2

1.2 研究现状 3

1.2.1 问答范式概述 3

1.2.2 基于信息检索的问答范式 4

1.2.3 基于知识库的问答范式 7

1.3 本文主要贡献 8

第 2 章 预备知识 9

2.1 卷积神经网络 9

2.2 循环神经网络 11

2.3 长短期记忆神经网络 12

第 3 章 双向 Attention 算法详述 15

3.1 模型概述 15

3.2 字符编码层 16

3.3 词语编码层 17

3.4 短语编码层 17

3.5 双向 Attention 层 17

3.6 建模层 18

3.7 输出层 19

3.8 模型训练 19

3.9 模型测试 20

第 4 章 基于中英翻译机制的中文问题答案抽取方法 21

4.1 算法总体流程.....21
4.2 数据预处理.....21
4.3 算法详述.....22
第 5 章 基于中文语料训练的中文问题答案抽取方法.....24
5.1 算法总体流程.....24
5.2 数据预处理.....24
5.3 算法详述.....25
第 6 章 中英翻译方法的实验结果与分析.....26
6.1 实验参数设定.....26
6.2 词向量编码维度对准确率的影响.....26
6.3 卷积核大小对准确率的影响.....28
6.4 融合函数对准确率的影响.....29
第 7 章 中文语料训练方法的实验结果与分析.....30
7.1 实验参数设定.....30
7.2 文本语序对准确率的影响.....30
第 8 章 总结与展望.....32
8.1 本文工作的总结.....32
8.2 未来工作的展望.....33
插图索引.....34
表格索引.....35
参考文献.....36
致 谢.....38
声 明.....39
附录 A 外文文献书面翻译.....40

第1章 引 言

1.1 研究背景

1.1.1 问答概述

问答（Question Answering）是计算机科学领域的一个重要研究方向，与信息检索、自然语言处理等技术密切相关。问答的最终目标是构建一个能够自动回答人类以自然语言提出的各种问题的系统。

传统的问答系统的工作机制是根据问题，从一个结构化的数据库（通常是知识库）中抽取和组织答案。更一般的问答系统还能够从非结构化的知识文档语料中抽取答案。而常见的非结构化知识文档语料包括维基百科、新闻网页等。

问答领域的研究致力于自动回答多种多样的问题，包括事实类、列表类、定义类等等。而按照问答系统的知识获取方式，又可将问答系统分为封闭领域问答系统和开放领域问答系统两类。封闭领域问答系统着重于回答某个特定领域（如医疗领域）的各类问题，这类问答系统的任务相对来说比较简单，由于问题范围较窄，只需通过自然语言处理的方法从该特定领域的本体中挖掘答案即可。另一方面，封闭领域问答系统常常只接受特定种类的问题，如只接受请求描述类的问题而不接受询问步骤类的问题。而随着机器阅读的方法在问答系统中的应用，一些领域（如医疗）已经有了该领域的问答系统，如询问有关阿兹海默症的问答系统。

开放领域问答系统则几乎负责回答一切问题，其回答问题是基于本体于各类已经存在的知识的。这类系统通常有十分丰富的知识预料可供挖掘，但随着问题种类和范围的大大增加，回答问题的难度也越来越高。另一方面如何从庞大繁杂的知识库中高效快速搜寻组织答案也是一大考验。

1.1.2 问答发展历程

最早的问答系统当属 BASEBALL 和 LUNAR，二者都为封闭领域问答系统。BASEBALL 能够回答关于一年某个时间段内关于美国棒球联盟的问题。LUNAR 则能够回答关于月球岩石的地理信息，这些信息由阿波罗探月计划收集。由于当时互联网还没有十分普及，问答领域也很窄，因此尽管当时的硬件

资源落后，但 BASEBALL 和 LUNAR 在问题回答准确率上还算令人满意。接下来若干年里，封闭领域问答系统得到了长足发展，这类问答系统几乎都有共同的特征——以数据库或某种特定领域的知识系统为核心，将用户的询问转化为可供数据库查询的 SQL 语句，最终根据 SQL 语句返回查询结果。

SHRDLU 是第一个获得巨大成功的问答系统，它于 60 年代末 70 年代初由 Terry Winograd 开发。其与之前的问答系统最大的不同在于对机器人行为的模拟，可以说是最早的人机对话系统，它实现了早期的人机交互，人可用自然语言提问和发出指令，SHRDLU 会依据人的指令做出相应动作或解答问题。当然，问题仅限于特定种类和特定领域，指令的种类也较少，但其意义是重大的。

到了 70 年代，问答系统更加集中于封闭领域的细化，问答的专业性越来越强，并有了知识库的概念。此时问答系统开始与专家系统对接，致力于针对特定问题产生更可靠且重复性较高的答案。专家系统与现代问答系统已经十分相似，只是内部工作机制不同。专家系统基于高度结构化组织的专家知识库，而现代问答系统则基于对海量非结构化自然语言语料库的统计学方法。

八十年代左右，计算语言学理论的不断完善极大促进了问答系统的发展，使其在自然语言理解方面的能力大大增强。其中的代表如由加州大学伯克利分校的 Robert Wilensky 的 Unix Consultant(UC)系统。该系统负责回答有关 Unix 操作系统的各类问题。UC 依赖与一个十分庞大完整的 Unix 知识库，几乎涵盖了包含 Unix 的一切知识，根据用户不同种类的询问，UC 可尝试从相关知识点中抽取组成答案。

目前，针对特定领域的高度面向自然语言的问答系统也发展起来，如生命健康领域的 EAGLi 系统。另一方面开放领域问答系统也加速发展，如微软小冰、苹果 Siri、麻省理工问答系统 Start、IBM 的沃森。值得一提的是，在 2011 年，沃森参加问答类综艺节目《危险边缘》并击败了该节目两位最强选手 Brad Rutter 和 Ken Jennings，堪称问答系统发展的一座里程碑。

1.1.3 问答系统与问题分类

目前主流的分类主要依据为问题答案的来源，主要分为“数据库问答”、“常问问题问答”（Frequently Asked Questions, FAQs）、“新闻问答”、“互联网问答”等。由于数据库数据存储组织的高效性，数据库问答系统首先发展起来，其依赖结构化的查询语句与用户进行交互，但用户使用该类问答系统的学习成本较

高。FAQ 问答系统在企业客服中应用十分普遍，其主要思想是将一些提问频率很高的问题答案统一整理、高效组织，依据用户问题与系统中已有问题的相似度给出系统中存在的答案，这类问答系统的优点是查询速度快，缺点是回答的问题数量比较有限。另一类重要的系统是新闻问答系统，该类系统之所以脱颖而出最主要的原因是数字新闻媒体的普及，如今每天互联网上涌现的海量新闻，其蕴藏的信息量是十分可观的，也是目前公认的作为开放领域问答系统的最好数据来源。关于互联网问答系统，其核心是利用搜索引擎，然后根据用户询问返回若干包含答案信息的相关文档并从中抽取答案，其表现第一依赖于搜索引擎的返回结果，第二依赖于对答案的精确检索，目前还面临很多挑战。

随着问答领域研究的不断深入，对问题的分类也不断细化，目前形成了包括“仿真陈述类问题”（Factoid Question）、“清单类问题”（List Question）、“定义类问题”(Definition Question)、“时间限制类问题”(Temporally Restricted Question)、“序列类问题”(Series of Question)在内等多类问题。其中最为普遍和基本的是“仿真陈述类问题”，这类问题询问有关一段预先给定语料的问题，并从该段语料中抽取若干文字片段组成答案。“清单类问题”顾名思义，即能回答诸如“请列举中国由哪些省份”一类的问题。“定义类”、“时间限制类”、“序列类”问题与字面意思相近，不再赘述。本文研究的问题类型为“仿真陈述类”，即回答一系列基于简单事实、并能用简短精炼的语言回答的问题。

1.2 研究现状

1.2.1 问答范式概述

现代问答系统按照回答问题的方法可分为**基于信息检索的问答**（IR-based question answering）和**基于知识库的问答**两种范式。本文的研究重点是“仿真陈述类”问答，因此下文重点均为两类范式在该类问答中的应用。

基于信息检索的问答范式也可以说是基于文本的。这种问答依赖的是互联网海量的文本数据。根据用户询问，利用信息检索技术从海量文档中抽取出与问题答案相关的文本段落。更具体地，这种方法会首先对用户以自然语言提出的问题进行分析，确定最可能的问题类型（通常是诸如人物、地点、时间等），再形成可供搜索引擎接受的询问（query）。搜索引擎根据询问会返回一个依据答案相关度排序的文档列表，最终系统会将抽取可能的候选答案文本并依据相关程度返回给用户。

第二种基于知识库的问答范式，我们则首先需要对用户询问进行一种形式化的语义表示，是的用户询问变成一种可计算的表达。形式化语义表示的方式多种多样，但其最终目标都是利用这种表示去进行数据库查询。数据库可以是多种多样的，如科学事实数据库或地理信息数据库。各种数据库都需要符合一定语法规则的、逻辑性较强的查询（如 SQL 语句）。

1.2.2 基于信息检索的问答范式

信息检索式问答的目标是从互联网文本中抽取小段文本作为问题答案返回给用户，它能够回答的问题大致如表 1.1 所示：

问题	答案
卢浮宫在哪儿？	法国巴黎
问答系统的英文说法是什么？	Question Answering
中国的流通货币名称是什么？	人民币
杏仁蛋白奶糖中用到的坚果是什么？	杏仁
吕思清演奏什么乐器而出名？	小提琴
中国的国土面积是多少？	960 万平方公里
世界上海拔最高的山峰是什么？	珠穆朗玛峰

表 1.1 信息检索式问答的常见问题与答案

通过上表能够看出，此类问答范式比较适合回答的大部分问题都是“仿真陈述类”问题。

如前文所述，基于信息检索的问答范式回答问题的流程大致分为三步：对问题的解析，信息检索、对候选答案的再加工。问题解析部分的主要任务为**形成询问**和**答案类型确定**，询问主要由问题中的关键词构成，这些关键词能够提高搜索引擎检索效率和结果准确性。答案类型确定的主要作用是确定产生答案的命名实体类型，该步骤可以有效降低信息检索缓解的搜索空间。信息检索的主要功能是排序，从海量文档中将文档排序，并将文档内部的段落排序。对候选答案的再加工指的是从已排序的文档和段落中抽取最可能的答案片段的过程，该步骤将最终产生返回给用户的答案。具体流程如图 1.1 所示。

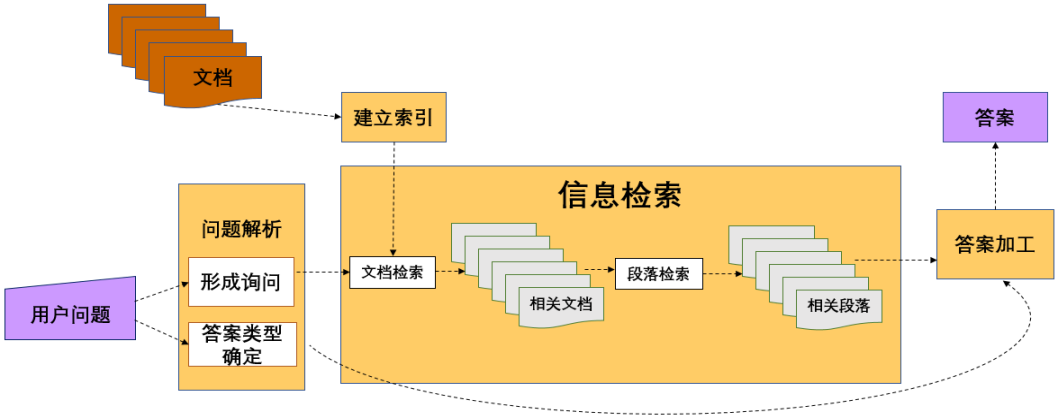

图 1.1 问答系统的工作流程图

问题解析步骤的第一个任务是形成询问。形成询问通常是从抽取问题中抽取一系列关键词，并在需要的情况下进行扩展得到。形成问询的另外一种方法是句子改写，Lin, J.提出了一系列改写规则[1]，其核心是将疑问词去掉并改为待填空陈述句，这种方式可以最大程度在文档中匹配到与答案相关的文本。对于答案类型的解析，通常采用的是分层归类的方法。Li and Roth 建立了一套标签式分类体系[2]，在这种分层的标签分类体系下，每一个问题都会首先被赋予一个粗粒度的标签如人物，或是一种复合式细粒度的标签如人物：描述、人物：分组等，具体分类方法如图 1.2 所示。问题分类的方式很多，既可以通过既定规则、也可以通过监督式机器学习或者融合二者的方法。但现代问题分类的主流方法还是在已经经过人工标注的数据集上进行训练最终产生问题分类器的[2] 。

信息检索的核心是搜索引擎，可以是面向一系列文档的检索系统，也可以是通用的互联网搜索引擎。若采用的是文档检索系统，则这一步首先进行粗粒度的文档相关度排序。但排序结果在前的文档未必就存在对问题的解答，这是因为系统最终要返回的是一小段文本答案，而不是整个文档，如此粗粒度的相关度排序有可能对接下来的答案抽取产生误导，因此在文档排序的基础上，我们还要进行更细粒度的排序，这通常是章节、段落或者是句子层面的排序。一种简单的方法是，我们采用某种分割算法，将一个文档划分为若干段落，再利用tf-idf①算法进行相关度排序。另外一种常用的计算相关度的方法是根据段落包含问

① 一种应用于信息检索和数据挖掘的常用加权技术

题关键词的多少来决定，如果能够再更短的句子中包含更多的关键词（即关键词密度大），则相关度较高[3]。还有一种比较常见的做法是采用N-gram overlap[4]。 其思想是计算问题和段落文本的在n个词语中的最大匹配数。如果采用的是通用互联网搜索引擎（如谷歌），一种普遍的做法是直接将问题输入搜索引擎，依据搜索引擎返回的文档和关键词匹配结果直接抽取相关句子。

接下来最关键的是答案抽取。传统的基于规则的答案抽取方法主要有两类，分别是基于答案类型的抽取（answer-type pattern extraction）和基于N-gram tiling的抽取。基于答案类型的抽取是根据问题解析部分判断的答案所属类型，生成正则表达式，从而从段落中匹配出答案。例如，一个问题的答案是人物类，那么接下来就可以对于候选答案文本进行标签搜索，将所有标签为人物的实体全部提取出来，再利用正则表达式进行进一步匹配最终产生答案。N-gram tiling[4]方法主要应用于互联网搜索引擎检索返回的结果中。第一步对于返回结果中包含关键词的片段，我们赋予所有片段中的单词（unigram）、双词(bigram)、三词(trigram)一定的权重，权重与这些gram在所有包含关键词片段中出现的频率有关。接下来是给每一个gram打分，分数与gram跟问题类型的匹配程度有关。最后一步是将得分高的gram拼接起来组成候选答案，一种常见的做法是贪心，即按照得分由高到低依次将有overlap的N-gram拼接产生候选答案，并将候选答案递归拼接，直到产生最终答案，在此过程中会不断淘汰掉组合后得分低的候选答案。

而现在的趋势则是基于端到端的监督式机器学习直接抽取答案，这类方法比之前基于规则的方法在答案准确率上有较大提高且不需要引入大量人为规则，因此近年来受到学界追捧并逐渐发展成为一个相对独立的研究领域——机器阅读。基于神经网络的机器阅读中第一步也是最关键的步骤是字词编码（word/character embedding）。起初使用的是单纯的循环神经网络[6]，这种网络结构十分适合对语义建模并具有一定的理解能力。随后一种特殊的循环神经网络——长短期记忆神经网络[7]因其对语言良好的记忆特性被广泛适用于字词编码和语义理解中。为了进一步提高答案抽取效果，一种是在结合LSTM[②]字词编码的基础上同时使用CNN[③]进行字符层面的编码[8]，并将二者结果融合作为字词编

[②] 一种时间递归神经网络，论文首次发表于 1997 年。由于独特的设计结构，LSTM 适合于处理和预测时间序列中间隔和延迟相对较长的重要事件。

[③] 一种前馈神经网络，人工神经元可以响应周围单元，可以大型图像处理。它包括卷积层和池层。

码结果。另一种是结合语法分析树的语义编码[9]。两种方法原理不同，但在实际应用中都取得了不错的效果。除了语义编码，在问题和候选语料相关度的计算方面也创造性地运用了一种叫做注意机制（Attention）的方法[10]，并在此基础上产生了问题到语料与语料文本到问题的双向注意机制[11]。受双向注意机制的启发，在语义编码层面也产生了双向编码的方法[12]。目前该方法已经在斯坦福问答数据集（Stanford Question Answering Dataset, SQuAD）上取得了很好的效果。

Tag	Example
ABBREVIATION	
abb	What's the abbreviation for limited partnership?
exp	What does the "c" stand for in the equation E=mc2?
DESCRIPTION	
definition	What are tannins?
description	What are the words to the Canadian National anthem?
manner	How can you get rust stains out of clothing?
reason	What caused the Titanic to sink ?
ENTITY	
animal	What are the names of Odin's ravens?
body	What part of your body contains the corpus callosum?
color	What colors make up a rainbow ?
creative	In what book can I find the story of Aladdin?
currency	What currency is used in China?
disease/medicine	What does Salk vaccine prevent?
event	What war involved the battle of Chapultepec?
food	What kind of nuts are used in marzipan?
instrument	What instrument does Max Roach play?
lang	What's the official language of Algeria?
letter	What letter appears on the cold-water tap in Spain?
other	What is the name of King Arthur's sword?
plant	What are some fragrant white climbing roses?
product	What is the fastest computer?
religion	What religion has the most members?
sport	What was the name of the ball game played by the Mayans?
substance	What fuel do airplanes use?
symbol	What is the chemical symbol for nitrogen?
technique	What is the best way to remove wallpaper?
term	How do you say " Grandma " in Irish?
vehicle	What was the name of Captain Bligh's ship?
word	What's the singular of dice?
HUMAN	
description	Who was Confucius?
group	What are the major companies that are part of Dow Jones?
ind	Who was the first Russian astronaut to do a spacewalk?
title	What was Queen Victoria's title regarding India?
LOCATION	
city	What's the oldest capital city in the Americas?
country	What country borders the most others?
mountain	What is the highest peak in Africa?
other	What river runs through Liverpool?
state	What states do not have state income tax?
NUMERIC	
code	What is the telephone number for the University of Colorado?
count	About how many soldiers died in World War II?
date	What is the date of Boxing Day?
distance	How long was Mao's 1930s Long March?
money	How much did a McDonald's hamburger cost in 1963?
order	Where does Shanghai rank among world cities in population?
other	What is the population of Mexico?
period	What was the average life expectancy during the Stone Age?
percent	What fraction of a beaver's life is spent swimming?
speed	What is the speed of the Mississippi River?
temp	How fast must a spacecraft travel to escape Earth's gravity?
size	What is the size of Argentina?
weight	How many pounds are there in a stone?

图 1.2 基于层次标签化的问题分类图

1.2.3 基于知识库的问答范式

基于知识库的问答是指在从数据库中查询答案的问答。这里的数据库通常是关系型数据库（relational database）或者是简单的 RDF 三元组（RDF triples）[④]数据库，目前知名度比较高的基于此类问答的应用有 Freebase[13]和 DBpedia [14]。

[④] 资源描述框架（Resource Description Framework），一种用于描述 Web 资源的标记语言。

一种简单的问答方法是填补三元组中的缺失项。如下面的 RDF 三元组：

Subject	**predicate**	**object**
中华人民共和国	诞生时间	公元 1949 年

这样的一个三元组可以用来回答如“中华人民共和国是何时成立的？”或者“哪个国家于1949年成立？”一类的问题。我们能从该问题中挖掘出“……国家是何时诞生”这样的模式。更一般地，我们可以总结出更多常见的模式。若我们已经有大量已经标注过的问题数据，则也可以采用监督学习的方式来学习更多更复杂的模式[15]。鉴于很难寻找大规模的训练语料库，也有许多采用半监督或非监督的方法来提取模式的[16]。另外在扩大模式提取范围的基础上，还产生了同义模式扩充等方法[17]，用于最大程度地进行模式匹配。

1.3 本文主要贡献

本文的研究集中于基于信息检索类的问答，且着重于分析目前表现最好的基于神经网络答案抽取方法在中文语境下的应用。本文将首先实现基于双向注意机制的问答抽取算法[18]，并结合中文的语言特性对算法进行调整和优化，最终实现一个性能良好的中文问答抽取算法。本文主要选定了两条优化途径，一种是**基于翻译模式的中文问答**，该方法仍旧采用英文语料库进行模型训练，但在算法应用阶段会进行两次中英翻译，即将中文问题和翻译为英文并输入给系统，再将系统产生的答案翻译成中文返回给用户。另外一种则是直接**采用中文语料进行模型训练**，直接产生中文答案，省去了中间翻译环节。这两种方法各有利弊第一种方法中间需要两部翻译转换，增加了算法的开销，同时采用机器翻译具有一定的不准确性，可能会对问题理解产生偏差。第二种方法更为直观，理论上应该会取得更好的效果，但由于缺乏大规模中文训练语料库，因此本实验采用的是由英文翻译为中文的斯坦福问答语料库以及采用填空式生成技术产生的中文问答语料库，训练数据质量必然有所下降，导致对模型的性能产生影响。

本文探索目前主流的基于机器学习的问题答案抽取方法在中文场景下运用的可能性，并取得了一定成绩。同时我们也开发了一个小型的中文问答平台，供有兴趣的研究者测试并提出意见。

第2章 预备知识

2.1 卷积神经网络

卷积神经网络（Convolutional Neural Network, CNN）本质是一种前馈神经网络，其核心思想借鉴了数学中卷积的概念。对于二维输入，借助卷积的帮助，它具有强大的特征抽取能力，因此在具有天然二维输入的图像领域取得了很好的效果。

卷积神经网络主要分为两种操作，一部分是卷积（Convolution），另一部分是池化（Pooling）。卷积是整个网络结构种最重要的操作，由卷积层实现。卷积层包含若干含有训练参数的过滤器（或称为卷积核），每个过滤器包含的参数数量有限，并且能够在整个输入上滑动抽取特征，整个过程如图2.1所示。池化操作将每个过滤器抽取的特征进行再次计算，一般池化分为两种：最大池化（max-pooling）和平均池化（average-pooling）。顾名思义，最大池化操作是选取一个过滤器抽取的特征向量种元素值最大的来代表整个特征向量。平均池化即计算整个特征向量的平均值并将其作为给过滤器抽取的特征值。在实际应用中，最大池化应用更多且效果更好，尤其是在图像领域。

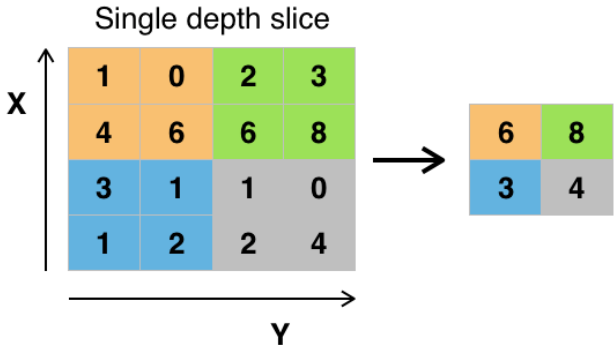

图 2.1 卷积神经网络的卷积特征抽取过程示意图

一个简单而典型的卷积神经网络一般分为3层：卷积层、池化层和全连接层。输入信息首先经过卷积层产生特征图（feature map），池化层对特征图进行

下采样形成稠密特征图，之后稠密特征图通过全连接层产生最终输出。一个典型的运用于图像的卷积神经网络工作过程如图2.2所示，图示过程采用了两个卷积层和两个池化层。

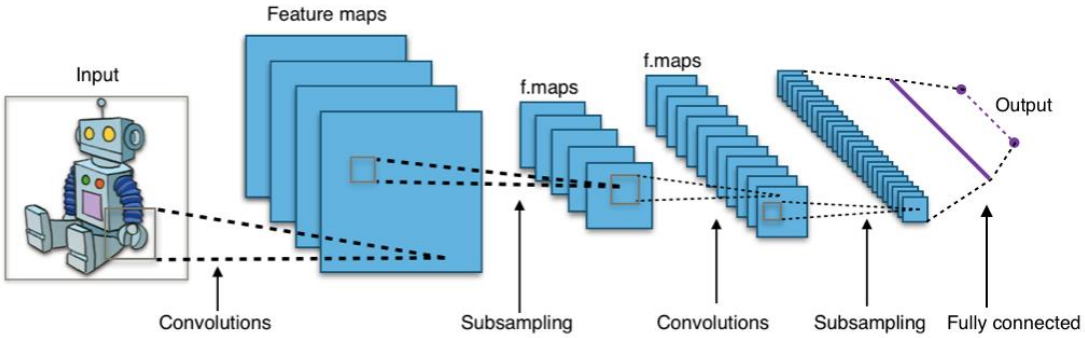

图 2.2　基于卷积神经网络的图像识别过程示意图

鉴于CNN在图像识别领域的突出表现，近些年来在自然语言处理领域也越来越多的使用CNN来处理各类问题，其中句向量编码就是一个典型的应用场景。我们将自然语言中的一句话看作一张二维“图片”，该图片的长度为单词数量，宽度为单词编码长度。然后通过多个卷积核在不同层次上对句子进行特征抽取，最终形成句向量。一个典型的句向量形成过程如图2.3所示，图示过程采用了6种卷积核，分别有三种不同的大小，最终通过最大池化和softmax函数[⑤]产生输出。

⑤ 是逻辑函数的一种推广。它能将一个含任意实数的 K 维的向量 z 的“压缩”到另一个 K 维实向量 σ(z)中，使得每一个元素的范围都在(0,1)之间，并且所有元素的和为 1。

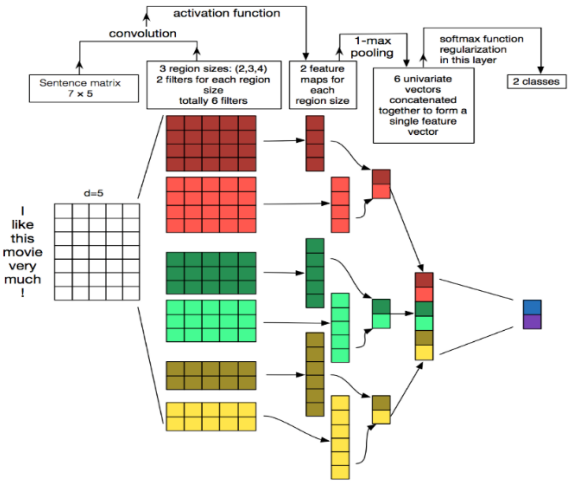

图 2.3 CNN 在句向量编码中的工作流程示意图

2.2 循环神经网络

循环神经网络（Recurrent Neural Network, RNN）是一种递归的神经网络。与前馈神经网络不同，RNN 能够根据模型上一时时刻、之前若干时间段的输入输、和本时刻的输入来确定本时刻的输出，是一种记忆能力的体现，一个典型的 RNN 结构如图 2.4 所示。其递归的网络结构天然地适合处理序列类型数据，典型的应用场景如语言模型和文本生成以及机器翻译。

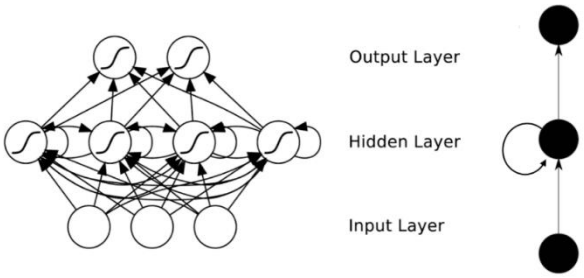

图 2.4 循环神经网络结构示意图

RNN 的结构主要分为三次部分，输入单元、输出单元和隐藏单元。隐藏单元是决定记忆能力的关键。一个简单的隐藏单元如图 2.5 所示，其中 x 代表输入，s 为隐藏单元状态，W 为对输入的权重，o 代表输出状态。为了便于理解，在表达 RNN 时我们也经常使用隐藏单元的展开图，图 2.5 的展开图如图 2.6 所示，图中我们分别展示了整个序列在 t-1、t、t+1 三个时刻的状态和输出。

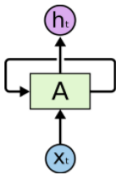

图 2.5 循环神经网络神经元示意图

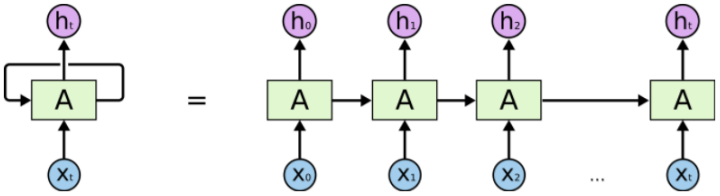

图 2.6 循环神经网络神经元展开图

通过图 2.6 我们也能看到，除了隐藏单元，RNN 的另一大特点是参数共享，不同时间状态下不同的输入共享同一权值矩阵，这极大降低了模型的参数训练量和复杂度，因此在实际训练效率上 RNN 与 CNN 相同具有很大优势。

2.3 长短期记忆神经网络

长短期记忆神经网络（Long Short Term Memory network, LSTM）是一种特殊的循环神经网络，它在处理输入数据的长期依赖问题上具有十分突出的表现。

LSTM 与普通 RNN 最大的不同在于隐藏单元结构。普通的 RNN 隐藏单元内部仅有一个激活门限（通常是 tanh 函数[6]）来处理上一隐藏层状态的输出、当前状态输入和当前输出的关系，LSTM 则复杂的多，具有若干激活门限，并采用多种多样的连接结构使其对长期依赖的记忆能力大大增强，一个典型的 LSTM 隐藏单元如图 2.7 所示。

[6] 双曲函数中的一个，tanh 为双曲正切。在数学中，双曲正切“tanh”是由基本双曲函数双曲正弦和双曲余弦，推导而来。

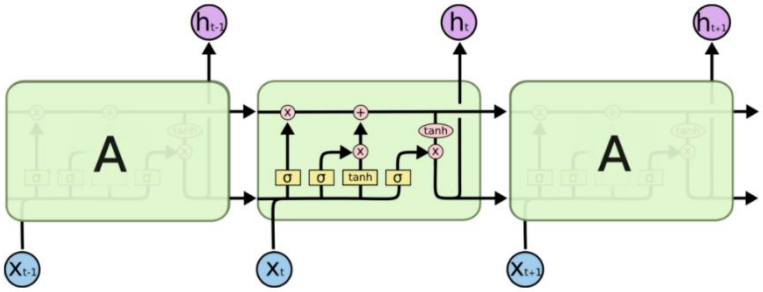

图 2.7 LSTM 神经元示意图

下面我们重点介绍一下 LSTM 背后的核心思想，这将帮助我们理解 LSTM 能够处理长期依赖的原因。

LSTM 的核心是单元状态。在处理序列数据时，我们可以直观地理解为序列数据缓慢地流经 LSTM 的一个个经过展开的隐藏单元。无论在 LSTM 中发生的什么样的计算，序列数据始终是单向流动的。在图 2.7 中我们已经看到，LSTM 单元中有许多门限，这些门限将对信息进行选择，重要的信息将通过门限参与输出部分的计算，非重要信息则会被门限截断不能继续流动，这就是选择遗忘机制。正是因为有这种机制，保证了模型不会过拟合，且具有一定的自主选择记忆和推理能力。根据图 2.7，我们可以总结出 LSTM 单元中发生的一系列计算过程。

首先对于原始的当前时刻输入数据x_t和上一时刻隐藏状态h_{t-1},我们初步计算中间状态f_t，f_t的计算方法如下：

$$f_t = \sigma(W_f \cdot \left[h_{t-1}, x_t\right] + b_f) \tag{2-1}$$

其中W_f为待训练参数矩阵，b_f为待训练偏置向量。

接下来我们同样利用h_{t-1}和x_t来计算我们具体需要存储哪些信息，首先通过 sigmoid 函数来决定更新的信息范围，得到i_t矩阵，计算方法如下：

$$i_t = \sigma(W_i \cdot \left[h_{t-1}, x_t\right] + b_f) \tag{2-2}$$

其中W_i、b_f的含义与公式（2-1）相同。

然后需要计算哪些信息需要更新到当前时刻的单元状态C_t中，我们将即将更新入C_t的信息称为$C_t^{'}$，$C_t^{'}$的计算方法如下：

$$C_t^{'} = \tanh(W_C \cdot \left[h_{t-1}, x_t\right] + b_C) \tag{2-3}$$

其中W_C、b_C的含义与公式（2-1）、（2-2）相同。

最关键的一步我们需要计算当前时刻单元状态C_t，其计算方法如下：

$$C_t = f_t * C_{t-1} + i_t * C_t^{'} \tag{2-4}$$

最后我们计算整个隐藏单元的当前状态输出o_t以及当前隐藏状态h_t，计算方法如下：

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2\text{-}5)$$

$$h_t = o_t * \tanh(C_t) \quad (2\text{-}6)$$

以上便是 LSTM 的核心计算原理，当然针对不同问题，很多学者也提出了许多不同的 LSTM 变种（如 Gated Recurrent Unit, GRU），此处不再展开。

第3章 双向 Attention 算法详述

3.1 模型概述

本文将实现一种基于双向注意机制（Bi-Directional Attention Flow）的神经网络结构。这种分层次的网络结构将在不同层次、不同粒度下对文本进行表示，详见见图 3.1。其中包括字符、词语、短语在内的三个编码层，其主要作用是对问题和答案候选文本进行不同层次的表示。之后我们利用双向 Attention 层来产生一种对问题敏感的候选答案所在文本（上下文）表示（query-aware context representation）。这里我们对 Attention 机制的实现相比于之前主流的方法有了一些改进。首先我们不再将问题和上下文完全转化为单一向量后再计算相关度，而是在每一个生成向量的过程中就进行 Attention 计算，这样可以减少因为过早地产生编码向量而带来的信息损失。另外，我们采用了双向 Attention 计算，既计算从问题到上下文的 Attention，也计算从上下文到问题的 Attention。这样可以避免只进行前者的单项计算而产生的偏差。

该模型于 2017 年初在斯坦福问答数据集（Stanford Question Answering Dataset, SQuAD）取得了最高准确率，同时也在 CNN/DailyMail 等数据集上由良好的表现。模型核心为六层神经网络：

1. **字符编码层（Character Embedding Layer）**将用一个接受字符输入的卷积神经网络将问题和上下文中出现的所有词语映射到一个高维向量空间。
2. **词语编码层（Word Embedding Layer）**使用经过预训练的词语编码模型同样将所有词语映射到高维向量空间。
3. **短语编码层（Phrase Embedding Layer）**考虑到相邻若干词语间的作用关系，并结合前两层编码结果对词语编码进行优化表示。
4. **注意流层（Attention Flow Layer）**综合前三层的词语编码表示，并将二者融合产生基于上下文信息的问题表示。
5. **建模层（Modeling Layer）**利用循环神经网络并结合注意流层产生的问题表示对上下文再次扫描。
6. **输出层（Output Layer）**计算候选答案与问题的相关概率，并最终产生答案。

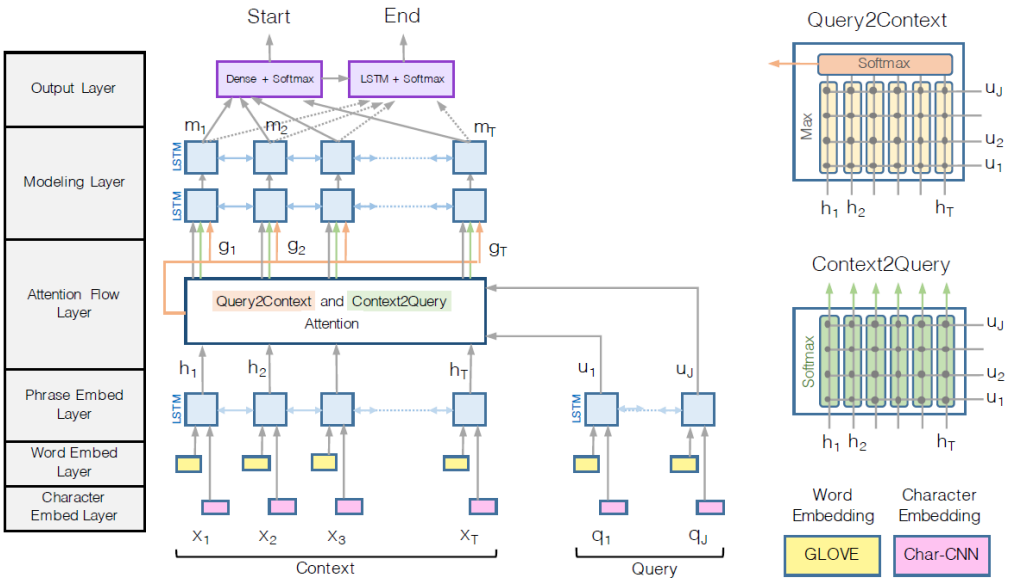

图 3.1 基于双向 attention 的神经网络算法结构图

3.2 字符编码层

我们考虑对问题和上下文文本的形式化表示，令 $\{x_1, \ldots x_T\}$ 和 $\{q_1, \ldots q_T\}$ 分别表示上下文和问题中的单词，利用字符粒度编码的卷积神经网络[19]，能够产生对词语语义高度抽象的单词向量。具体地，我们首先将 26 个英文字母和其他符号进行 one-hot 编码，并将其作为卷积神经网络（CNN）的一维输入，然后采用多层卷积操作和一次最大池化（max pooling）操作产生对单词的稠密向量表示，具体如图 3.2 所示。

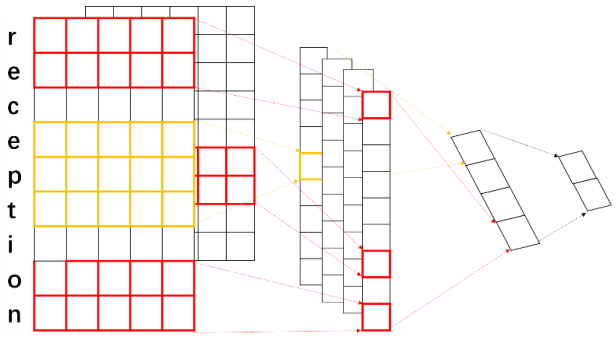

图 3.2 CNN 字符编码示意图

3.3 词语编码层

词语编码层的工作与字符编码层相同，均将所有单词映射到高维向量空间，只不过采用的方法有所不同。这里我们采用经过预训练的单词向量GloVe[20]直接获得定长向量。至此我们便获得了由两种不同方法产生的问题文本矩阵Q_1和Q_2以及上下文文本矩阵C_1和C_2。接下来的工作是融合，我们将上述四个矩阵通过一个两层的高速网络（Highway Network）[21]，该高速网络的输出是经过融合的d维的问题文本矩阵 $Q \in R^{d \times J}$和d维的上下文文本矩阵$X \in R^{d \times T}$，其中J和T分别代表问题单词数量和上下文单词数量。

3.4 短语编码层

该层将接受 X 和 Q 矩阵作为输入，利用长短期记忆神经网络(Long Short-Term Memory Network, LSTM) [22]处理文本相邻词的交互关系。这里我们使用了两个双向的 LSTM，这样做很大程度上保留了文本的原始信息，最终我们将双向 LSTMs 的输出进行合并获得矩阵$H \in \mathbb{R}^{2d \times T}$和矩阵$U \in \mathbb{R}^{2d \times J}$，由于我们将两个 LSTM 的输出按行进行合并，因此 H 和 U 的维度均是 2d。

短语编码层和字符编码层、词语编码层一样，都是对整个输入文本进行编码，三者只是在编码粒度上有差异，因此实质表现的是文本在不同粒度下的矩阵表示，这与多层卷积神经网络在抽取图像特征方面的方法类似。

3.5 双向 Attention 层

该层的输入是短语编码层产生的上下文矩阵 H 以及问题矩阵 U，本层的目的是将二者融合，产生一种基于问题的上下文表示（query-aware context representation），该表示可表达为矩阵 G。为了生成可计算矩阵 G，我们需要分别计算从上下文到问题、从问题到上下文两个方向的 attention，为了双向计算我们首先需要获得一个相似度矩阵（similarity matrix）$S \in \mathbb{R}^{T \times J}$，该矩阵表达的是短语编码层生成的矩阵 H 和 U 中每一个词的相关关系。具体地，S_{tj}表达的是上下文中第 t 个单词和问题中第 j 个单词的相似度。相似度矩阵 S 的计算方法为：

$$S_{tj} = \alpha(H_{:t}, U_{:j}) \in \mathbb{R} \tag{3-1}$$

其中α是一个标量函数，$H_{:t}$表示上下文矩阵第 t 个单词所代表的向量，$U_{:j}$表示问题矩阵第 j 个单词所代表的向量。对于α的解析式表达并没有一个明确的定义，通常我们可选取$\alpha(h,u)=w_{(S)}^{T}[\mathrm{h};\mathrm{u};\mathrm{h}\circ\mathrm{u}]$，其中$w_{(S)}\in\mathbb{R}^{6d}$，其元素具体数值可通过训练产生，∘代表基于矩阵元素的乘法。[;]表示向量按行拼接。

从上下文到问题的 attention(Context-to-Query Attention)表征了对于上下文中的每一个单词，问题中哪一个单词与之相关度最高。我们令$a_t\in\mathbb{R}^J$表示上下文中第 t 个单词对于问题中所有单词的 attention 权重，则直观地我们有 $\sum a_{tj}=1,\forall \mathrm{t}\in[0,\mathrm{T})\wedge \mathrm{t}\in \mathrm{N}$, 且$a_{tj}=\ softmax(S_{t:})\in\mathbb{R}^J$，相应地，我们接下来获得的基于问题的上下文表示矩阵$U_{:t}^{'}=\sum_j a_{tj}U_{:j}$，这样$U_{:t}^{'}$即是一个2d×T规模的矩阵，该矩阵是基于问题的上下文表示。

从问题到上下文的 attention（Query-to-Context Attention）表征了对于问题中的每一个单词，上下文中哪一个单词与之相似度最高，这也是该网络最关键的部分。与计算 Context-to-Query Attention 类似，相应的权重$\mathrm{b}=\mathrm{softmax}(max_{col}(\mathrm{S}))\in\mathbb{R}^J$,其中$max_{col}$函数是取矩阵中最大元素所在列的列向量。接下来我们就得到了基于上下文的问题表示$h^{'}=\sum_t b_t H_{:t}\in\mathbb{R}^{2d}$，该向量将上下文中关于问题最重要的单词进行了加权求和，最终为了计算方便，我们将$h^{'}$按列拼接 T 次，最终得到矩阵$H'\in\mathbb{R}^{2d\times T}$。

最后，结合短语编码层生成的矩阵 H，我们最终可以得到对于问题敏感的上下文表示矩阵 G，

$$G_{:t}=\beta(H_{:t},U'_{:t},H'_{:t})\in\mathbb{R}^{d_G} \tag{3-2}$$

$G_{:t}$对应与上下文中的第 t 个单词。对于β函数，这里的处理是将其简单看做一若干有关向量的按行拼接，如 $\beta(h,u',h')=[h;u';h\circ u';h\circ h']\in\mathbb{R}^{8d\times T}$。当然，一种更好的做法是将$\beta$看作一个可训练的带参函数（如多层感知机），但简单的矩阵拼接再英文数据集上已经取得了不错的效果。

3.6 建模层

得到矩阵 G 后，建模层将进一步捕捉问题与上下文之间的交互关系，可以直观的理解成对带着问题对上下文的再次扫描。我们采用在机器阅读中应用广泛的双向 LSTM（Bi-LSTM）扫描矩阵 G, 并产生对回答问题最有帮助的矩阵表

示$M \in \mathbb{R}^{2d \times T}$，M 的每一列代表一个单词，但此时的单词向量既包含上下文信息，也包含问题信息。

3.7 输出层

该层的结构功能依应用场景（问答、阅读理解）而定。此处应用于仿真陈述类问答，我们的目标是从所给上下文中抽取片段作为答案返回。因此我们要确定该片段的起止位置。我们首先计算片段开始位置的概率分布：

$$p^1 = softmax(w_{(p^1)}^T [G; M]) \tag{3-3}$$

其中$w_{(p^1)}^T \in \mathbb{R}^{10d}$，是一个权重可训练矩阵。对于结束位置，我们将矩阵 M 再次通过一个双向的 LSTM 得到$M^2 \in \mathbb{R}^{2d \times T}$，接下来我们计算结束位置的概率分布：

$$p^2 = softmax(w_{(p^2)}^T [G; M^2]) \tag{3-4}$$

至此我们只需选出概率最大的p^1和p^2中的元素直接将截取答案并返回即可。

3.8 模型训练

对于神经网络的模型训练我们首先要定义训练的损失函数，由于我们采用的是监督学习，将采用直观的概率分布损失之和作为损失函数$L(\theta)$，具体表达为：

$$L(\theta) = -\frac{1}{N}\sum_i^n \log\left(p_{y_i^1}^1\right) + \log(p_{y_i^2}^2) \tag{3-5}$$

这里θ表示该模型中所有可以训练的参数，N 代表训练集的数据规模，y_i^1和y_i^2代表第 i 个样本的真正答案实际的起止位置。

最终我们选取答案文本范围为(k, l)，使其$p_k^1 p_l^2$的值最大。

3.9 模型测试

我们在斯坦福问答数据集（SQuAD）、哈工大填空式中文阅读理解数据集和微软机器阅读理解数据集（Microsoft Machine Reading Comprehension Dataset, MS-MARCO）上对模型进行了评测。

对于前两种数据集，我们采用 Exact Match(EM) score 和 F1 score 作为模型评测指标。EM score 衡量模型预测的答案文本与实际答案文本的实际单词匹配率。F1 score 是召回率和准确率的调和平均，具体计算方法为：

$$\text{F1 score} = 2 \cdot \frac{precision \cdot \text{recall}}{precision + recall} \tag{3-6}$$

其中 precision 表示预测答案文本中正确的单词数与文本总单词数的比率，recall 表示正确单词数和实际答案单词数的比率。

对于 MS-MARCO 数据集，其数据特性与前两种数据集不同，其数据全部从微软必应搜索引擎获取，所有问题和答案均来源于现实世界，答案也全部为人工手动填写，因此答案文本很可能并非候选文档中的文本片段。这里我们分别采用 ROUGE-L[⑦]两个指标衡量从候选文本片段选出文本片段作为答案，该片段应该与实际人工填写的答案具有最高的 ROUGE-L 和 BLEU1 值。通过这种方法，我们仍旧能够采用 EM score 和 F1 score 对模型进行评价。

⑦ 一种自动文档摘要效果评价方法，也用于机器翻译等自然语言处理领域。

第4章 基于中英翻译机制的中文问题答案抽取方法

4.1 算法总体流程

第三章我们详细介绍了基于双向 Attention 的英文问题答案抽取算法，下面我们将尝试用这种方法来解决中文问答。一种直观的想法是采用翻译的方式，即对于用户输入的中文问题，我们首先将其翻译为英文，再将对应的英文问题输入给前述模型，同时对于上下文，我们也要将原始的中文上下文转换成英文，经过翻译后的问题于上下文才能作为前述模型的输入，进而进行各层编码，产生对应于翻译文本的答案片段，最后再将答案片段翻译为中文返回给用户。大致流程如图 4.1 所示。

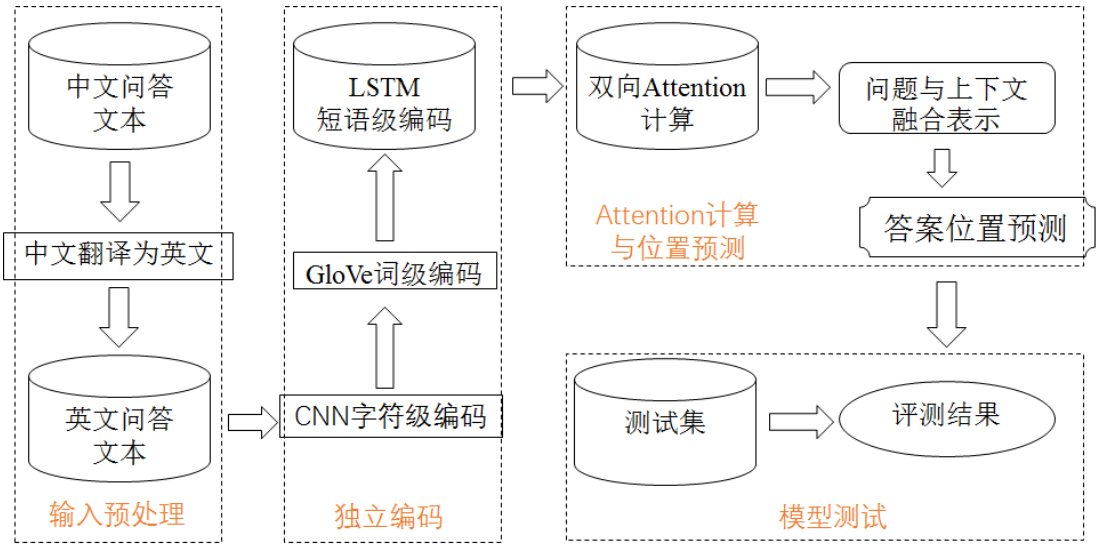

图 4.1 基于中英翻译机制的问答抽取算法流程图

4.2 数据预处理

数据预处理阶段首先要解决的问题是特殊词，即不在词表内的词。对与 SQuAD 数据集，我们需要首先进行词频统计，并选出词频最高的前五万个单词组成词表，对于超出该词表的词语我们均统一用 UNK 表示，并用零向量统一编码。同时，我们需要对非词语的特殊字符进行处理，主要包括逗号、句号等标

点符号，在此我们选取了十五个常用的符号进行编码，对其他特殊字符同样采取 UNK 表示。

在此基础上，我们对 SQuAD 训练集进行清洗，将训练集中的指针位置有字符位置转换为单词位置，这样就避免出现了答案中会有半个单词的问题，同时所有 UNK 均占一个单词的位置。

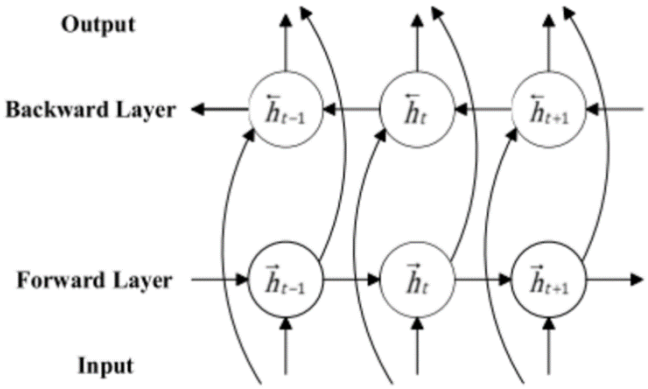

图 4.2 双向 LSTM 工作示意图

4.3 算法详述

算法的第一步是处理用户输入。对于用户输入的问题和上下文，我们首先将用户问题和上下文进行翻译，在此我们采用的是谷歌翻译。然后对其中的每一个单词（包括特殊符号）进行扫描，对于在我们预先筛选的词表中不存在的词直接进行 UNK 标记。

首先在字符编码层，如第三章所述，我们对于输入的所有单词和符号进行分割，所有的单词都转换为字符的集合输入给字符编码层。由于英文有 26 个英文字母，我们将采用六位二进制编码对这 26 个英文字母进行初始编码，并将整个单词按照字符组成顺序组成单词矩阵，之后便是采用 CNN 的卷积和池化操作对单词进行特征抽取，最终形成向量。对于 10 个特殊字符，我们采用同样的六位二进制向量编码处理，只不过我们把这些特殊字符当作字母数量为 1 的单词，即单词矩阵的行数为 1。

在词语编码层，我们的做法与字符级编码层相同，首先对词表外的词进行UNK 标记，并根据 GloVe 词向量字典直接查找得到每一个单词对应的数值向量。

在短语编码层中，我们将前两层的单词向量进行拼接，作为该层输入。其中问题和上下文的单词输入是独立进行的。为了产生词与词之间的交互，表达词之间的关联性，我们使用两个互为反方向的 LSTM 对输入单词进行编码，具体地，我们首先按照正序让所有单词通过 LSTM，其所有隐藏层的输出反向作为另一个 LSTM 的输入，我们最终将反向 LSTM 最后一个隐藏层的输出作为短与编码层的输出，双向 LSTM 的工作流程如图 4.2 所示。

双向 Attention 层、建模层和输出层的所有细节已在第三章阐述，不再赘述。

第5章 基于中文语料训练的中文问题答案抽取方法

5.1 算法总体流程

与第四章所介绍的基于翻译机制的中文问答抽取方法不同，本章所介绍的方法无需经过翻译，将直接产生中文答案。这种思想借鉴了原始的双向 Attention 机制，这种机制本质上对语言不敏感的，理论上应该能将完成各种语言的问答。因此我们在原始双向 Attention 算法的基础上进行了适当改进，使得整个算法能够像处理英文单词一样处理中文字符。

这种方法将直接接受中文问题和上下文作为输入，直接进行中文的字符级和词级编码，之后各层如第三章所述进行信息汇总和特征抽取，最终产生预测答案文本。具体流程如图 5.1 所示。

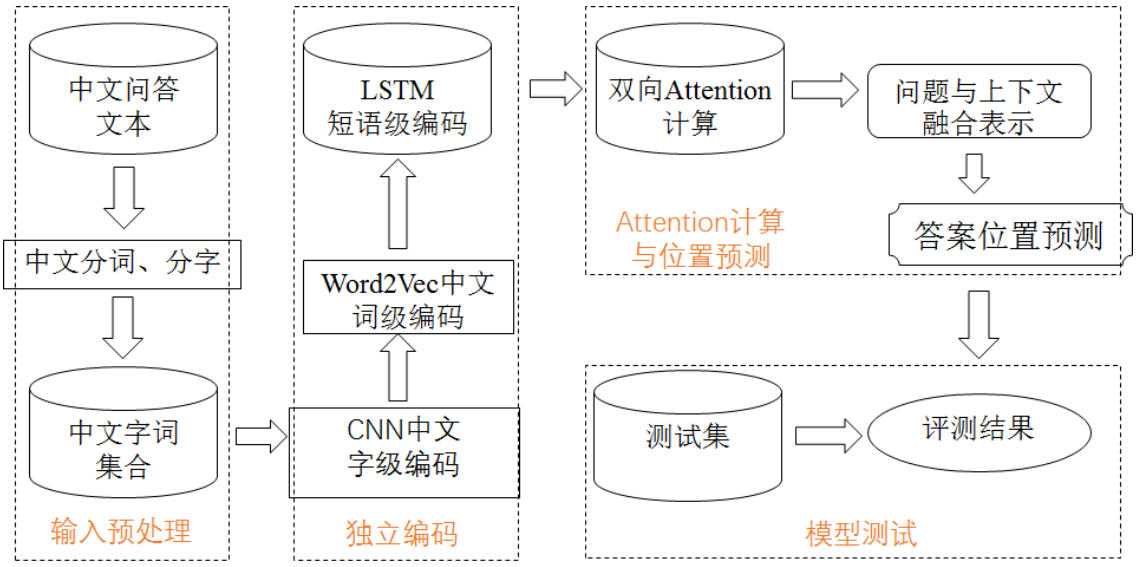

图 5.1 基于中文语料库训练的问题答案抽取算法流程图

5.2 数据预处理

我们需要解决的首要任务就是中文问答数据的获取。我们采取了很多方法，如产生基于填空式的问答数据，利用这种方法，我们爬取了百度百科 700 篇明星档案数据并声称了 7000 多个与这些文章相关的问题，但其中最重要的问题是

问题种类单一，答案必须是实体，因此如果用该数据作为训练集可以预见，我们的模型的回答能力是比较有限的。因此我们仍采用的是经过人工标注的SQuAD 数据集，采用谷歌翻译的方式将所有文本翻译为中文。这里我们需要注意，翻译过后，文章的语序、答案所在位置都会发生变化，因此我们需要重新统计获得 SQuAD 训练集中标准答案文本在文章中的位置。同第四章描述的一样，我们将字符级的位置改为中文字级表示。这里我们定位的方法是，先将答案片段翻译为中文，再回到原文中进行全匹配，若无法匹配则计算采用上下文中文本片段的 BLEU 值，将 BLEU 值最高的片段作为问题答案，经统计 95%以上的答案均可通过全匹配的方式定位到中文答案。

在词表处理上，与基于翻译的方法不同，我们不仅统计词频，还要统计字频，对于词语，根据经验值，我们将词表的大小设置为六万，对于字表，我们设置为一万。二者均是根据词频和字频由高到低选取。

5.3 算法详述

算法的大致流程与第三章所介绍的双向 Attention 算法相似，主要区别在于前两层。

字符编码层在此编程了字级编码层，结合中文特性，我们将中文当中的单字看作英文当中的字母。二者的主要区别在于集合规模。英文仅有 26 个字母，而汉字却有成千上万个，因此为了能够采用二进制编码，我们将字符编码的维度从 5 扩大到了 17 来对字表中的所有单字进行编码。并将单字编码表示按照所组成的词语进行拼接，形成词语矩阵。由于中文中存在大量的单字词语，而很少存在多字（三个或以上）词语，因此我们呢在卷积神经网络的卷积核大小上统一设置为 1 并不做更改。

在词语编码层，我们没有如 GloVe 一样预先训练好的词向量作为输出，为了解决这个问题，我们采用了谷歌的开源项目 Word2Vec, 该项目提供了一个词向量训练工具，理论上借助语料库，可以完成对任何一种语言的词向量编码。因此我们使用了搜狗发布的全文新闻语料库 SoGouCA 进行训练，得到了中文词级向量表示。

第6章 中英翻译方法的实验结果与分析

6.1 实验参数设定

在基于中英翻译的方法下，我们分别在 SQuAD 数据集和 MS-MARCO 数据集上进行了测试，实验中的参数设定如表 6.1 所示。

参数名称	值（value）
CNN 层卷积核数量(filter num)	100
CNN 层次卷积核大小（filter size）	1×5
英文词向量编码宽度(d_e)	100
中文词向量编码宽度(d_c)	100
GPU 数量（GPU num）	1
批大小(batch size)	60
学习率（learning rate）	0.5
样本训练周期（epoch）	12
遗忘率（dropout rate）	0.2

表 6.1 翻译方法下实验参数设定表

6.2 词向量编码维度对准确率的影响

根据第三章介绍的神经网络模型，模型的前三层分别对问题和上下文进行不同粒度的编码（embedding），而词语编码层输出的向量（word embedding）维度从根本上决定了语义表达的精准度，也会持续影响到后续各层的效率，因此我们首先研究 Embedding 维度对模型效果的影响。

对于基于翻译机制的问答抽取算法，我们对字符编码层的过滤器（filter）数量进行调整，filter 数量直接决定了字符编码层输出向量维度。前文也曾提到，原生的问答算法在词语编码层采用了经过预训练的 GloVe，依据 GloVe 所具备向量维度：50、100、200 和 300，我们分别选取第二层输出维度（d）为 50、100、200 和 300，分别得到模型在 SQuAD 和 MS-MARCO 数据集上的表现分别如图 4.1 和 4.2 所示。能够看到，模型在维度在 d=100 的情况下表现最好，在

SQuAD 数据集上 EM（Exact Match，完全匹配率）[⑧]值达到了 70%，F1 值[⑨]达到了 70%。分析原因不难看出，当 d=50 维度较低，可能无法将语义充分表达，而维度过高一方面会给计算增加负担，另一方面存储的冗余信息会使需要训练的参数量大大增加，导致模型不宜收敛，效果大大折扣。因此 d=100 是一个比较符合实际的结果。

在 d=100 的情况下，我们也统计了不同上下文本长度对回答准确率的影响，如图 4.3 和图 4.4 所示。随着文档长度的上升，准确率下降很快，在词数小于 100 时能够达到 90%以上，而当词数超过 300 以后只能维持在 60%左右，说明该模型对于回答需要浏览大量文档的问题效果仍有待提升。

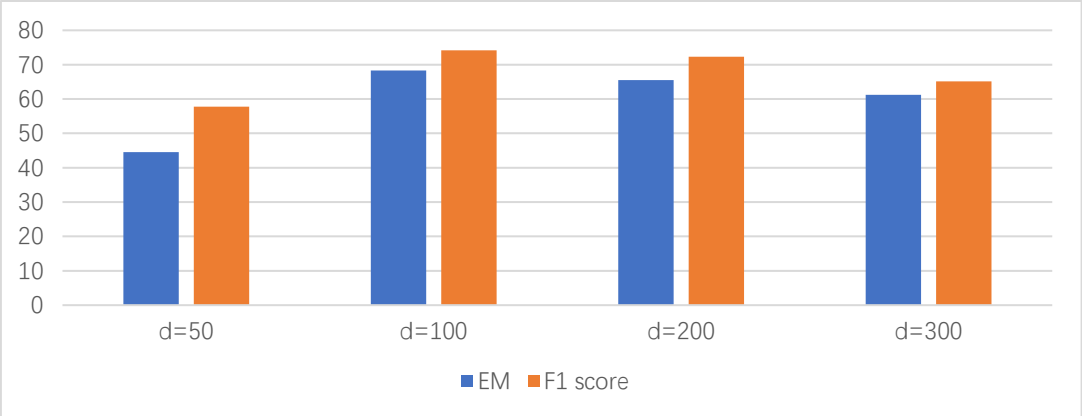

图 6.1 翻译机制下词向量编码维度对 EM 和 F1 分数影响（SQuAD 数据集）

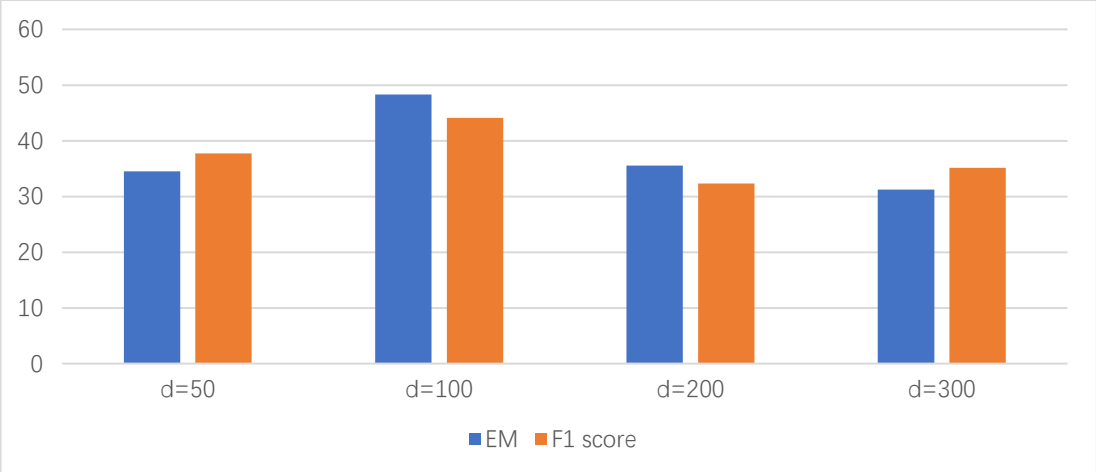

图 6.2 翻译机制下词向量编码维度对 EM 和 F1 分数影响（MS-MARCO 数据集）

⑧ 一种衡量文本相似度的评价方法，计算文本之间的完全匹配单词数占总单词数百分比
⑨ 一种衡量文本相似度的评价方法，综合了文本间的准确率和召回率，是二者的调和平均。

6.3 卷积核大小对准确率的影响

在字符编码层中我们默认选取的 filter size 为1×5，这主要考虑到英文中字符较少，用五位元素取值为 0/1 的向量足够对大部分英文字符编码。在 Seo M[18]等人的论文中也提到将 filter size 设为1×5是一个比较明智的做法，既能够保证向量不会过长导致增加计算负担，也能够充分对字符进行稠密编码，同时效果良好。但当进行基于中文语料库训练时，汉字数目繁多，仍然采用 5 位二进制编码不现实，因此我们最终采用较为常见的 17 位二进制编码，这样即可实现对九万多个汉字的稠密编码同时计算开销不会太大。在固定过滤器宽度的基础上，我们适当调整过滤器高度，过滤器高度不同代表对特征抓取的粒度不同，我们分别选择 filter size 为1×17、2×17和3×17的过滤器分别测试模型效果，测试结果如图 4.5 和图 4.6 所示。

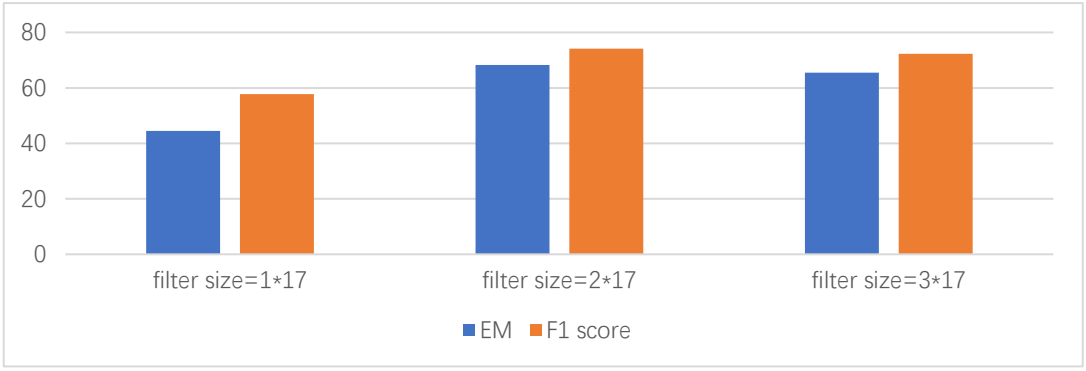

图 6.3 翻译机制下卷积核大小对 EM 和 F1 分数影响（SQuAD 数据集）

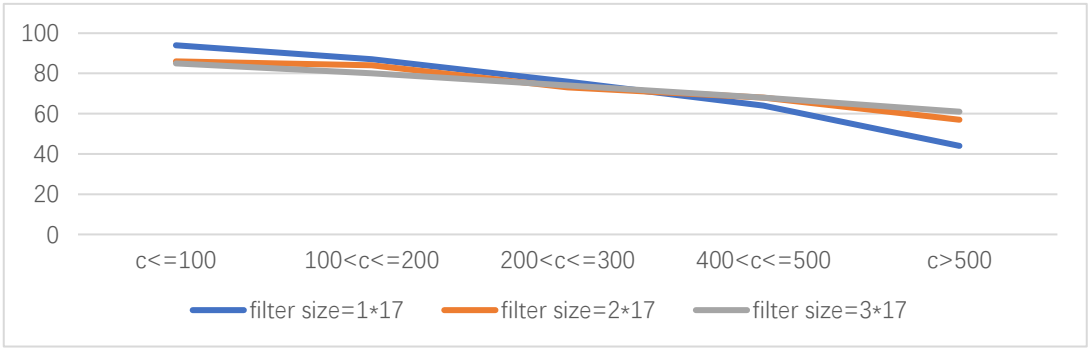

图 6.4 卷积核大小在不同文本长度下对 EM 和 F1 分数影响（SQuAD 数据集）

通过结果能够看出对于 filter size=1 的情况下处理短文本的准确率很好，相比 filter size=2 和 3 的情况则明显下降。但是当文本长度上升值超过 400 词时，长度更长的过滤器效果要优于 filter size=1 的情况，表现出对长文本更强的特征抽取能力。

6.4 融合函数对准确率的影响

第四章算法详述中我们提到过在第四层双向 Attention 层中，我们最终需要将 attention 权重向量和短语编码层输出向量进行融合产生矩阵 G。我们也提到了两种融合函数，一种是较为简单的矩阵拼接，例如 $\beta(h,u',h') = [h;u';h\circ u';h\circ h'] \in \mathbb{R}^{8d\times T}$，另一种则使用可以训练的神经网络模型，这里我们选取即简单又具有强大非线性拟合能力的多层感知机。

接下来我们分别采用上述提到的两种方法对模型进行评测，其中多层感知机的隐藏层数为 1，评测结果如图 4.6 所示。能够看到两种方法的表现差别不大，而相比之下较为简单的矩阵拼接无需参数训练，效率更高，因此可作为首选融合函数。当然我们这里只是采用了最为简单的隐藏层数为 1 的多层感知机，增加隐藏层数和模型复杂度或许会带来准确率上的提升。

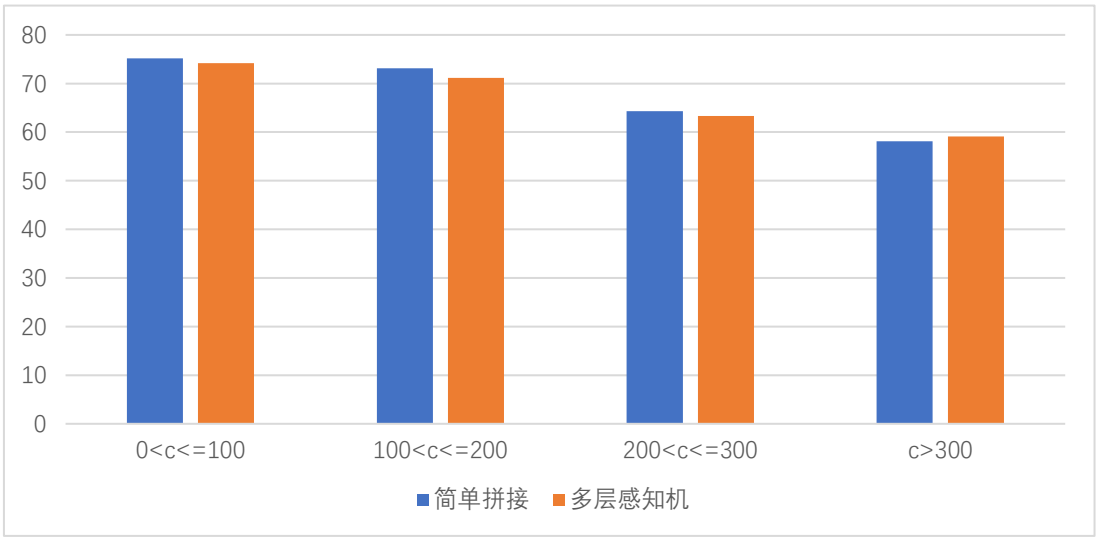

图 6.5 翻译机制下不同融合函数对 F1 分数影响（SQuAD 数据集）

第7章 中文语料训练方法的实验结果与分析

7.1 实验参数设定

在基于中文语料训练的方法下，我们仅在 SQuAD 数据集上进行了测试，实验中的参数设定如表 7.1 所示。

参数名称	值（value）
CNN 层卷积核数量(filter num)	100
CNN 层次卷积核大小（filter size）	1×17
英文词向量编码宽度(d_e)	100
中文词向量编码宽度(d_c)	100
GPU 数量（GPU num）	1
批大小(batch size)	60
学习率（learning rate）	0.5
样本训练周期（epoch）	12
遗忘率（dropout rate）	0.2

表 7.1 基于中文语料训练下的实验参数设定表

7.2 文本语序对准确率的影响

谷歌在机器翻译研究中[24]，曾尝试将训练集中的待翻译文本语序颠倒进行训练，发现能够更充分地利用 LSTM 的记忆能力，产生了更好的翻译效果，因此在本文的研究中也进行了类似尝试，我们将训练集中的文章文本和问题文本分别进行了逆序处理，并对比了其与正序的实验结果，具体如图 4.8 所示。

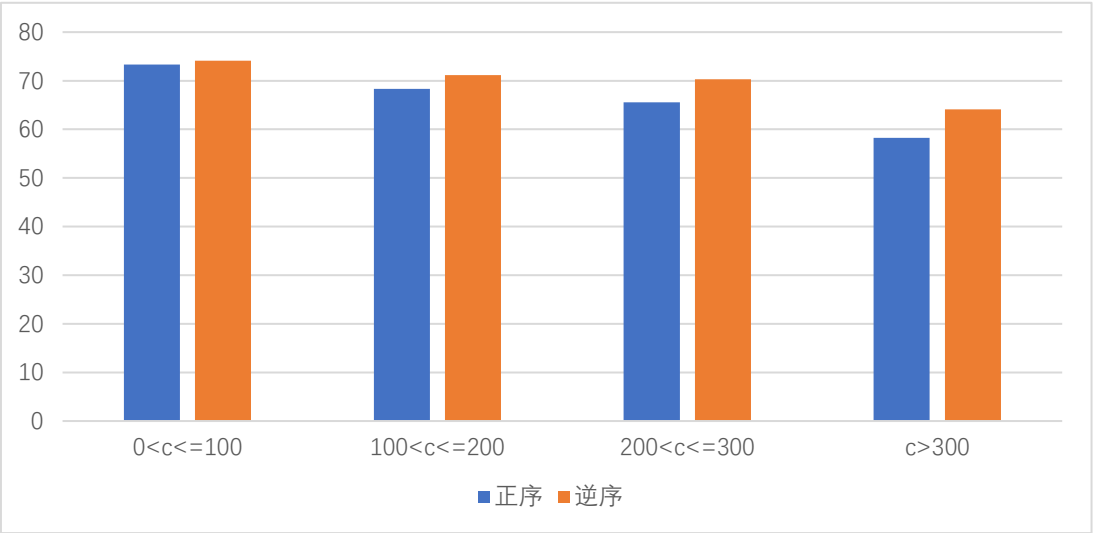

图 7.1 中文语料训练下不同文本语序对 F1 分数影响（SQuAD 数据集）

实验结果表明，文本语序对于基于中文语料训练的方法是有一定效果的，我们初步分析这可能与中英语言特性有关。中文中前置定语较多，因此一句话的主要信息可能集中在前半部分，通过颠倒语序能够使文本在通过双向 LSTM 层编码时能够赋予前置信息更大的权值，有利于在答案提取截断更准确地抓取重要信息。另外我们也发现，通过颠倒语序的方式模型在文本长度上的稳定性也有一定提高，这与谷歌的结论是一致的。对于语言和语序在基于机器学习的问答中所起的作用仍需要进一步深入研究。

第8章 总结与展望

8.1 本文工作的总结

机器问答是目前自然语言处理和人工智能技术结合最紧密的方向之一，也是机器智能的重要体现。本文结合了 LSTM、Attention 机制等在自然语言处理取得良好表现的算法，提出并实现了两种基于双向 Attention 机制的中文问题答案抽取方法。该方法以基于双向 Attention 机制的英文问答算法为基础，分别采用了翻译方法和中文语料训练方法，并比较了两种算法在不同情况下的优劣。

本文的主要贡献在于将最新的基于机器学习的问答抽取算法应用到中文领域，并结合中文的特性对算法做出一系列优化。

基于监督学习的问答抽取算法其性能很大程度上取决于训练数据集，因此将该算法应用到中文问答首要的问题便是训练数据集的获取。在缺少大规模中文问答数据集的情况下，本文首先提出了基于翻译机制的中文问答算法，即利用目前现有的高质量英文问答数据集进行训练，以产生高质量的模型，再利用中英互译的方法理解中文问题并给出中文答案。这种方法直观的好处是无需寻找中文训练数据，且原理通俗易懂，但由于经过了两层翻译，尤其是对问题和上下文的翻译很大程度决定了模型输入是否准确，倘若翻译出现问题，那么整个系统在问题输入阶段就已经产生了很大偏差，之后的各种语义编码和计算也就无异于白费力气，因此系统的鲁棒性不强。

为了增强系统的鲁棒性，我们提出了第二种直接基于中文语料库的训练方法。本文我们直接采用了翻译的方法将斯坦福问答数据集通过谷歌翻译完全翻译为中文，再进行模型训练。该方法虽然在数据预处理的翻译阶段可能产生偏差，但由于训练过程直接接受中文输入，因此实际应用时对中文的鲁棒性比较强，进行优化时也具有一定针对性。

本文对比了两种方法，发现在处理短文本时，第一种基于翻译的方法具有很好的表现，几乎与英文问答算法表现相同，但在处理长文本时准确率下降很快。而第二种方法在处理长文本时更具优势。

我们分别对两种算法在各层进行了参数优化，并尝试通过颠倒输入语序的方法来增强模型表现，取得了一定成果。

在此基础上本文针对第二种方法实现了一个中文问答测试平台，主要供测试和展示使用。

8.2 未来工作的展望

通过本次研究发现最大的困难在于目前尚无大规模高质量的中文问答数据集。所谓大规模，主要指要有针对文本的大量由人提出的问题；所谓高质量，主要指问题和文本的关联性要强。SQuAD 数据集和 MS-MARCO 数据集均采用众包的方式利用人工标注产生，因此质量很高。哈工大讯飞联合实验室发布的中文阅读理解数据集是一个很好的尝试，但整个语料库是基于命名实体识别技术的填空式问答，因此与人类提问和回答方式还存在一定差距，希望接下来能够国内也能拥有高质量的中文问答数据集，相信有了数据集，整个模型在中文的表现会更加出色。

当前最主流的机器问答方法大部分均是基于双向 attention 的，最近微软亚洲研究院也在机器阅读理解领域做出了新的尝试，创造性地提出了 R-NET 网络结构[25]，实现了目前在 SQuAD 数据集上的最佳表现。因此将本文研究的双向 attention 机制和 R-NET 结合可能会成为机器问答领域的一个研究方向。

目前基于机器学习的问答方法仍具有很多局限性，对于长文本的信息抽取仍是一大难题，另外，在日常生活中更常见的问答常常不从候选文档中抽取文本片段作为答案，而是采用了记忆和推理机制，这也是目前机器人问答的核心技术。因此如果要真正实现能够应用的问答系统，一方面要提升信息检索技术和答案抽取算法的准确率，另一方面要融合 Memory 机制[26]，使机器真正产生记忆和推理能力，相信会取得更好的效果。

插图索引

图 1.1 问答系统的工作流程图 5
图 1.2 基于层次标签化的问题分类图 7
图 2.1 卷积神经网络的卷积特征抽取过程示意图 9
图 2.2 基于卷积神经网络的图像识别过程示意图 10
图 2.3 CNN 在句向量编码中的工作流程示意图 11
图 2.4 循环神经网络结构示意图 11
图 2.5 循环神经网络神经元示意图 12
图 2.6 循环神经网络神经元展开图 12
图 2.7 LSTM 神经元示意图 13
图 3.1 基于双向 attention 的神经网络算法结构图 16
图 3.2 CNN 字符编码示意图 16
图 4.1 基于中英翻译机制的问答抽取算法流程图 21
图 4.2 双向 LSTM 工作示意图 22
图 5.1 基于中文语料库训练的问题答案抽取算法流程图 24
图 6.1 翻译机制下词向量编码维度对 EM 和 F1 分数影响（SQuAD 数据集） 27
图 6.2 翻译机制下词向量编码维度对 EM 和 F1 分数影响（MS-MARCO 数据集） 27
图 6.3 翻译机制下卷积核大小对 EM 和 F1 分数影响（SQuAD 数据集） 28
图 6.4 卷积核大小在不同文本长度下对 EM 和 F1 分数影响（SQuAD 数据集） 28
图 6.5 翻译机制下不同融合函数对 F1 分数影响（SQuAD 数据集） 29
图 7.1 中文语料训练下不同文本语序对 F1 分数影响（SQuAD 数据集） 31

表格索引

表 1.1 信息检索式问答的常见问题与答案 1 4
表 6.1 翻译方法下实验参数设定表 26
表 7.1 基于中文语料训练下的实验参数设定表 30

参考文献

[1] Lin J. An exploration of the principles underlying redundancy-based factoid question answering[J]. ACM Transactions on Information Systems (TOIS), 2007, 25(2): 6.

[2] Li X, Roth D. Learning question classifiers[C]//Proceedings of the 19th international conference on Computational linguistics-Volume 1. Association for Computational Linguistics, 2002: 1-7.

[3] Monz C. Minimal span weighting retrieval for question answering[C]//Proceedings of the SIGIR Workshop on Information Retrieval for Question Answering. 2004, 2.

[4] Brill E, Dumais S, Banko M. An analysis of the AskMSR question-answering system[C]//Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002: 257-264.

[5] Lin J. An exploration of the principles underlying redundancy-based factoid question answering[J]. ACM Transactions on Information Systems (TOIS), 2007, 25(2): 6.

[6] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model[C]//Interspeech. 2010, 2: 3.

[7] Cheng J, Dong L, Lapata M. Long short-term memory-networks for machine reading[J]. arXiv preprint arXiv:1601.06733, 2015.

[8] Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification[C]//Advances in neural information processing systems. 2015: 649-657.

[9] Liu R, Hu J, Wei W, et al. Structural Embedding of Syntactic Trees for Machine Comprehension[J]. arXiv preprint arXiv:1703.00572, 2017.

[10] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[11] Xiong C, Zhong V, Socher R. Dynamic Coattention Networks For Question Answering[J]. arXiv preprint arXiv:1611.01604, 2016.

[12] Seo M, Kembhavi A, Farhadi A, et al. Bidirectional Attention Flow for Machine Comprehension[J]. arXiv preprint arXiv:1611.01603, 2016.

[13] Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]//Proceedings of the 2008 ACM SIGMOD international conference on Management of data. AcM, 2008: 1247-1250

[14] Bizer C, Lehmann J, Kobilarov G, et al. DBpedia-A crystallization point for the Web of Data[J]. Web Semantics: science, services and agents on the world wide web, 2009, 7(3): 154-165.

[15] Zettlemoyer L S, Collins M. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars[J]. arXiv preprint arXiv:1207.1420, 2012.Fader A, Soderland S, Etzioni O. Identifying relations for open information extraction[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011: 1535-1545.

[16] Fader A, Soderland S, Etzioni O. Identifying relations for open information extraction[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011: 1535-1545.

[17] Berant J, Liang P. Semantic Parsing via Paraphrasing[C]//ACL (1). 2014: 1415-1425.

[18] Seo M, Kembhavi A, Farhadi A, et al. Bidirectional Attention Flow for Machine Comprehension[J]. arXiv preprint arXiv:1611.01603, 2016.

[19] Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014.

[20] Pennington J, Socher R, Manning C D. Glove: Global Vectors for Word Representation[C]//EMNLP. 2014, 14: 1532-1543.

[21] Rupesh Kumar Srivastava, Klaus Greff, and J¨urgen Schmidhuber. Highway networks. arXiv preprint arXiv:1505.00387, 2015.

[22] Schmidhuber J, Hochreiter S. Long short-term memory[J]. Neural Comput, 1997, 9(8): 1735-1780.

[23] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[24] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]//Advances in neural information processing systems. 2014: 3104-3112.

[25] Palangi H, Smolensky P, He X, et al. Deep Learning of Grammatically-Interpretable Representations Through Question-Answering[J]. arXiv preprint arXiv:1705.08432, 2017.

[26] Weston J, Chopra S, Bordes A. Memory networks[J]. arXiv preprint arXiv:1410.3916, 2014.

致 谢

感谢徐华老师在这次毕业设计中对我的细心指导。徐老师帮助我完成了论文选题，并在完成毕业设计的整个过程中一直关心我的进度并给我提出了很多宝贵的建议。

感谢实验室的几位师兄为我答疑解惑，帮助我解决了毕业设计中遇到的许多困难。

感谢我的本科室友对我毕业设计实验设计思路的启发，帮助我开拓思路并更高效地完成了毕业设计中的所有工作。

感谢我的父母对我一直以来的关心和支持，他们十分关心我的毕业设计乃至整个本科四年的学习情况，一直鼓励我、促我上进。

最后，感谢清华对我的培养，帮助我形成了严谨扎实的求学态度和锲而不舍的钻研精神。再次感谢帮助过我的各位老师和同学们，也希望自己能够进一步探索自身的学术兴趣和人生志向，争取实现人生理想，为社会创造更大价值。

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____　　日　期：_____

附录 A 外文文献书面翻译

R-NET:自匹配网络在机器阅读理解中应用

摘要：在本文中，我们介绍了 R-NET 网络结构，这是一种为机器阅读理解而设计的端到端的神经网络模型，其目的是从给定的文章中回答问题。我们首先匹配问题和与文章基于门控制注意递归网络的问题与文章表示形式。然后我们提出自匹配的注意机制，以完善文章与自身的匹配，它有效地以编码表示的形式来表达整篇文章的信息。我们最后采用了到指针网络找到答案在段落中的位置。我们进行广泛的实验对球队和 MS-MARCO 数据集和我们的模型达到最佳结果在两个数据集之间所有发表的结果。

A.1 引言

在本文中，我们专注于阅读理解风格的问题回答，旨在回答一系列根据一段文本或文章提出的问题。我们主要使用斯坦福问题回答数据集（SQuAD）（Rajpurkar 等，2016）和 Microsoft MAchine Reading COmprehension（MS-MARCO）数据集两个用于阅读理解和问答的大型数据集，这两个数据集都是通过众包的方式人工创建的。 SQuAD 需要根据所给的文章内容提出问题，这就限定了答案必须是文章内容中的一段文本。这与填空式阅读理解数据集不同（Hermann 等，2015; Hill，2016），填空式阅读理解数据集其中的答案是单个单词或实体。此外，SQuAD 需要不同的形式逻辑推理来推断答案（Rajpurkar 等，2016）。另一个从真实问答环境产生的数据集 MS-MARCO 提供从必应搜索引擎收集的若干针对于某一问题的相关文件。由于 MS-MARCO 数据集中的问题是由人提出的，答案不局限于所给文档中的片段，更多的可能来自于用户自己推理和表达。

自发布 SQuAD 数据集以来，机器阅读领域取得飞速发展。 Wang 和 Jiang（2016b）使用匹配 LSTM 构建针对特定问题的文章表示（Wang&Jiang, 2016a），并用指针网络进行答案边界预测（Vinyals et al.，2015）。 Seo 等人（2016）引入双向注意力流动网络，对多层次的问题通道对进行建模的粒度。熊等 （2016）提出了出现这个问题的动态共同注意网络并通过同时和迭代地提炼答案预测。 Lee 等人（2016）和 Yu 等人（2016）通过对段落内的连续文本跨度进行排序来预测答案。

受 Wang 和 Jiang（2016b）的启发，我们引入了如图 1 所示的 R-NET 端到端神经网络模型，用于阅读理解和问答。 我们的模型包括四部分：1）循环网络编码器，其作用是分别对问题和文章进行编码。2）门控匹配层，其作用是将问题和文章进行初步匹配。3）自匹配层，作用是汇总和聚合整个篇章的信息。4）基于指针网络的答案边界预测层，作用是将答案所在的起止下标输出产生答案。这项工作的主要贡献有三点。

Passage: Tesla later approached Morgan to ask for more funds to build a more powerful transmitter. **When asked where all the money had gone, Tesla responded by saying that he was affected by the Panic of 1901**, which he (Morgan) had caused. Morgan was shocked by the reminder of his part in the stock market crash and by Tesla's breach of contract by asking for more funds. Tesla wrote another plea to Morgan, but it was also fruitless. Morgan still owed Tesla money on the original agreement, and Tesla had been facing foreclosure even before construction of the tower began.
Question: On what did Tesla blame for the loss of the initial money?
Answer: Panic of 1901

表 1　SQuAD 数据集中的一个例子

首先，我们提出了一个基于门限注意力的循环神经网络，为基于注意力的网络结构增加了一个门限（Bahdanau 等人，2014;Rocktaschelet 等人，2015; Wang&Jiang，2016a）。我们考虑到在进行机器阅读理解和问答中，一段文字中每句话对回答一个问题的重要性不同，在参考了 Wang 和 Jiang（2016a）论文工作的基础上，我们同样采用带有注意机制的方法对文章和问题进行编码，并将二者信息相互融合，最终产生针对问题的文章编码表示通过引入基于注意力的门限控机制，我们的循环网络结构能够根据与问题的相关程度的不同来分配不同的权重给文章的各个部分。降低不相关的文章部分的权重并加强与问题关联紧密的文章中的重要部分。

其次，我们引入了自我匹配机制，可有效聚集整个段落的重要信息并推断答案。通过门限控制匹配层，我们首先对问题进行编码，编码的依据是通过比对问题中的每个单词与文章中每个单词的关联度。然而，在实际应用过程中循环网络只能在一定长度的文本内保持记忆能力，尽管理论上它可以记忆任意长度的文本信息。候选的某些答案往往是在文章中其它部分的线索。为了解决这个问题，我们设计了一个自匹配层来动态地通过自匹配的方式来改进文章表示。在已经获得针对特定问题的文章表示的基础上，我们采用基于门限控制的注意力机制，使文章能够自己关注与自己相关的部分，通过不断地扫描文章本身来提取信息，这种基于门限控制的注意力机制大大丰富了文章编码所能表示的信息，也更好地量化了问题，使得后续各层各种操作能够更好地搜索和预测答案。

最后，我们所提出的方法达到了目前世界上的最好的结果。 我们的单一模型在 SQuAD 测试集上的完全匹配精确度达到了 72.3%，而集成模型则进一步将这一指标提升到了 76.9%，这一结果目前在 SQuAD 排行榜上居于首位。此外，我们的模型也在 MS-MARCO 数据集上获得了目前公开发表的最好结果（Nguyen 等人，2016）。

A.2 任务描述

对于阅读理解风格的问答，我们的任务是给出一个段落 P 和问题 Q 是根据 P 中发现的信息来预测 A 的问题答案 A. SQuAD 数据集进一步将答案 A 限制为通道 P 的连续子跨度。答案 A 通常包括非实体并且可以是更长的短语。 这个设置挑战了我们了解和理解两者问题和通过，以推断答案。 表 1 显示了 SQuAD 中的一个简单示例数据集。 对于 MS-MARCO 数据集，提供了 Bing Index 的几个相关段落 P 一个问题 Q.此外，MS-MARCO 中的答案 A 是由人生成的，不能是连续跨越通道。

A.3 R-NET 结构

图 1 给出了 R-NET 模型的结构概述。首先，问题和文章分别通过一个双向的循环神经网络（Mikolov 等人，2010）。 然后我们用匹配基于门限的注意力循环神经网络对问题和文章进行匹配，获得针对特定问题的文章表示。 除此之外，我们利用自匹配层对文章信息进行进一步聚合，形成更丰富的文章表示，然后将其传递到输出层以预测答案的起止位置。

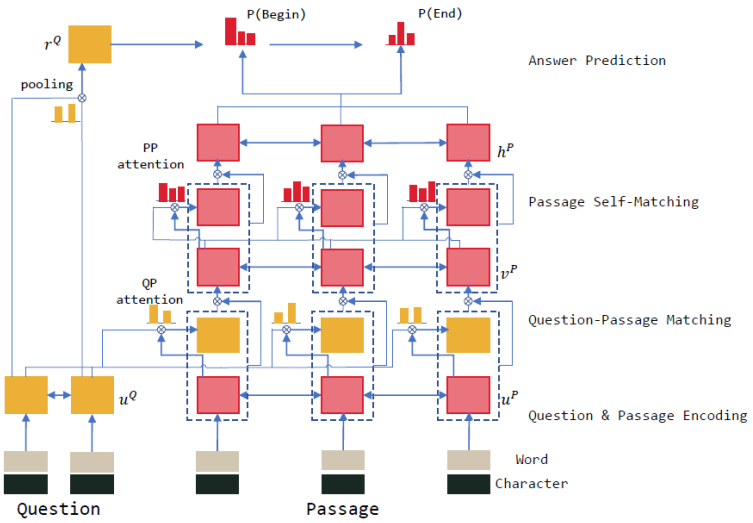

图 1　R-NET 整体结构示意图

A.3.1 问题与篇章编码器

考虑问题 $\mathrm{Q}=\{w_t^Q\}_{t=1}^m$ 和一篇文章 $\mathrm{P}=\{w_t^P\}_{t=1}^n$。我们首先将单词转换为他们各自的词级表示（$\{e_t^Q\}_{t=1}^m$ 和 $\{e_t^P\}_{t=1}^n$）和字符级表示（$\{c_t^Q\}_{t=1}^m$ 和 $\{c_t^P\}_{t=1}^n$）。字符级编码是通过将原始语料以字符为单位输入给循环神经网络（RNN）并抽取 RNN 的最后一个隐藏层的输出产生的。这样的字符级编码已经被证明是有助于处理词表以外的词汇，而且能够防止模型过拟合。 然后，我们使用双向 RNN 来产生新的问题和文章的表示 $u_1^Q,\ldots,u_m^Q$ 和 $u_1^P,\ldots,u_n^P$，它们分别表示问题和文章中的所有单词：

$$u_t^Q = BiRNN_Q(u_{t-1}^Q,[e_t^Q,c_t^Q]) \quad (1)$$
$$u_t^P = BiRNN_P(u_{t-1}^P,[e_t^P,c_t^P]) \quad (2)$$

在我们的实验中，我们选择使用门控循环单元（GRU）（Cho 等人，2014）来代替 LSTM（Hochreiter＆Schmidhuber，1997），因为 GRU 的计算过程和原理类似于 LSTM，但是在计算上开销更小。

A.3.2 基于注意门限的的循环神经网络

我们提出了一个基于注意力的循环网络，将问题信息纳入文章表示。 它是一种基于注意力的循环神经网络的变体，因为它有一个门限来确定信息在有关问题段落中的重要性。我们将问题和文章分别表示为 $\{u_t^Q\}_{t=1}^m$ 和 $\{u_t^P\}_{t=1}^n$，Rocktaschel 等人提出了通过自动对齐的方式生成句子对的表示$\{v_t^P\}_{t=1}^n$：

$$v_t^P = RNN(v_{t-1}^P,c_t) \quad (3)$$

其中$c_t = att(u^Q,[u_t^P,u_{t-1}^P])$，是对于整个问题进行池化过的 attention 向量：

$$s_j^t = v^T\tanh(W_u^Q u_j^Q + W_u^P u_t^P + W_v^P v_{t-1}^P)$$
$$a_j^t = \exp(s_j^t)\Big/\sum_{j-1}^{m}\exp(s_j^t)$$
$$c_t = \textstyle\sum_{i=1}^m a_j^t u_i^Q \quad (4)$$

每个段落代表 v_t^P 动态合并来自的汇总匹配信息整个问题。Wang&Jiang（2016a）介绍了基于匹配的 LSTM 网络，它采用 u_t^P作为一个额外的输入投入到网络：

$$v_t^P = RNN(v_{t-1}^P, [u_t^P, c_t]) \quad (5)$$

为了确定文章每一部分的于问题的相关程度和重要性，我们给输入再加入一个门（$[u_t^P, c_t]$）进入 RNN：

$$g_t = sigmoid(W_g[u_t^P, c_t])$$
$$[u_t^P, c_t]^* = g_t \odot [u_t^P, c_t] \quad (6)$$

与 LSTM 或 GRU 中的门不同，这个附加门的通过与否是基于当前的文章信息和其对应的池化 attention 向量的，其重点是问题之间的关系和当前通行词。添加的这一门限有效地模拟了这样一种场景，即一篇文章中只有部分于所问问题相关。我们在接下来的计算中将使用$[u_t^P, c_t]^*$而不是$[u_t^P, c_t]$ 。我们称这为基于门限的注意力循环网络。

A.3.3 自匹配注意机制

数据通过基于门限的注意力循环网络后，我们获得的针对问题的文章表示 $\{v_t^P\}_{t=1}^n$ 已经可以确定文章中的哪些部分于回答问题相关。 这种表示的一个问题是它的语境知识非常有限。一个候选的答案经常能够给回答某一问题提供重要的线索，而且这种线索与该答案相邻的文本关联不大。而且，有时候在问题和所给文章中或多或少存在一些语法或句法上的分歧（Rajpurkar 等人，2016），因此通过语境来推断答案是必要的。为了解决这个问题，我们提出了让已经具有与问题相关表示的文章进行自匹配的方法。 它动态收集整个段落中的重要信息，并对与之相关的信息进行了编码，最终我们可以获得文章的更高阶表示h_t^P：

$$h_t^P = BiRNN(h_{t-1}^P, [v_t^P, c_t]) \quad (7)$$

其中$c_t = att(v^P, v_t^P)$是整个文章的经过池化的注意力向量（v^P）：

$$s_j^t = v^T \tanh(W_v^P v_j^P + W_v^{P'} v_t^P)$$
$$a_j^t = \exp(s_j^t) / \sum_{j=1}^{n} \exp(s_j^t)$$
$$c_t = \sum_{i=1}^{n} a_j^t v_i^P \quad (8)$$

我们在计算$[v_t^P, c_t]$时将用基于门限控制的注意力循环网络中附加门，以自适应控制 RNN 的输入。

A.3.4 输出层

我们借鉴了 Wang&Jiang（2016b）的方法并使用了指针网络（Vinyals 等人，2015）来预测答案的开始和结束位置。 此外，我们使用经过池化的 attention 向量来通过问题向量最终产生指针网络的初始隐藏层输入向量。我们给定文章的集合表示$\{h_t^P\}_{t=1}^n$，注意机制这里被用作从文章选择答案起始位置（p^1）和结束位置（p^2），整个过程可形式化表达如下：

$$s_j^t = v^T \tanh(W_h^P h_j^P + W_h^\alpha h_{t-1}^\alpha)$$
$$a_i^t = \exp(s_i^t) / \sum_{j=1}^{n} \exp(s_j^t)$$
$$p^t = argmax(a_1^t, \dots, a_n^t) \quad (9)$$

这里h_{t-1}^a表示循环神经网络的最后一个隐藏状态（即指针网络）。循环神经网络的输入是一个基于预测概率分布a_t的池化 attention 向量：

$$c_t = \sum_{i=1}^{n} a_i^t h_i^P$$
$$h_t^a = RNN(h_{t-1}^a, c_t) \quad (10)$$

当预测答案的起始位置时，h_{t-1}^a表示循环神经网络的初始隐藏层状态。我们利用问题向量r^Q作为循环神经网络的初始状态。$r^Q = att(u^Q, V_r^Q)$是基于参数V_r^Q的问题的注意集合向量：

$$s_j = v^T \tanh(W_u^Q u_j^Q + W_v^Q V_r^Q)$$
$$a_i = \exp(s_i) / \sum_{j=1}^{m} \exp(s_j)$$
$$r^Q = \sum_{i=1}^{m} a_i u_i^Q \quad (11)$$

为了训练网络，我们采用最小化负对数概率的总和的方法，其中概率分布是输出层所给出的答案起止位置概率分布。

A.4 实验

A.4.1 实现细节

我们主要关注 SQuAD 数据集，并用它训练和评估我们的模型，在过去几个月中，该数据集获得了巨大关注。 SQuAD 包含 10 万多个问题，全部通过众包的方式由人工完成，语料来自 536 篇维基百科文章。数据集随机分为训练集（80％），开发集（10％）和测试集（10％）。每个问题的答案都是文章中的一小段文本。

我们使用斯坦福 CoreNLP（Manning 等人，2014 年）的分词器来预处理每个段落和问题。在我们的整个过程中都使用了 LSTM 的变体模型门循环单元（Cho 等人，2014）。对于词向量编码，我们对于问题和文章都使用了经过预处理的区分大小写的 GloVe 词向量（Pennington 等人，2014），这样就保证了训练期间向量长度是固定的。另外我们使用零向量来代表所有词表以外的词。我们利用 1 层双向 GRU 来计算字符级向量表示和 3 层双向 GRU 来编码问题和文章，在我们的实验中，基于门限控制的注意循环网络也同样对问题和文章进行了双向编码。所有层的隐藏向量长度设置为 75，用于计算注意力权重向量的隐藏层大小也是 75。同时我们也在层之间应用 0.2 的遗忘率（Srivastava 等人，2014）。该模型采用 AdaDelta 优化方法（Zeiler，2012）进行了学习率为 1 的优化学习。另外，在 AdaDelta 中使用的ρ和ε分别为 0.95 和$1e^{-6}$ 。

A.4.2 SQuAD 实验结果

我们使用两个指标来评估 SQuAD：完全匹配率（EM）和 F1 分数。EM 衡量模型预测的答案与实际答案完全匹配的词语数所占的百分比。F1 衡量预测的答案和实际答案之间的最大重叠比例。在开发集上的评测结果由官方的评测脚本给出。由于测试集被隐藏，我们需要将模型提交给斯坦福自然语言处理组才能获得测试结果。

表 2 显示了我们的模型在开发集和测试集上的 EM 和 F1 值，并给出了与其他方法的效果对比。我们的复合模型由 18 个具有相同结构和训练参数的单一

模型组成的。在测试时，我们选择在 18 个模型中具有最高的置信度得分的答案作为最终答案。我们可以看到，我们的方法明显优于基准线，并且超过了所有之前相比很先进方法，在单一模型和复合系统都大大超越了之前的方法。在原来自匹配层的基础上，我们利用双向 GRU 深度整合匹配文章信息并将结果输入给答案指针层。这将有助于进一步进行信息整合和无损传播。

	Dev Set	Test Set
Single model	**EM / F1**	**EM / F1**
LR Baseline (Rajpurkar et al., 2016)	40.0 / 51.0	40.4 / 51.0
Dynamic Chunk Reader (Yu et al., 2016)	62.5 / 71.2	62.5 / 71.0
Attentive CNN context with LSTM (NLPR, CASIA)	- / -	63.3 / 73.5
Match-LSTM with Ans-Ptr (Wang & Jiang, 2016b)	64.1 / 73.9	64.7 / 73.7
Dynamic Coattention Networks (Xiong et al., 2016)	65.4 / 75.6	66.2 / 75.9
Iterative Coattention Network (Fudan University)	- / -	67.5 / 76.8
FastQA (Weissenborn et al., 2017)	- / -	68.4 / 77.1
BiDAF (Seo et al., 2016)	68.0 / 77.3	68.0 / 77.3
T-gating (Peking University)	- / -	68.1 / 77.6
RaSoR (Lee et al., 2016)	- / -	69.6 / 77.7
SEDT+BiDAF (Liu et al., 2017)	- / -	68.5 / 78.0
Multi-Perspective Matching (Wang et al., 2016)	- / -	70.4 / 78.8
FastQAExt (Weissenborn et al., 2017)	- / -	70.8 / 78.9
Mnemonic Reader (NUDT & Fudan University)	- / -	69.9 / 79.2
Document Reader (Chen et al., 2017)	- / -	70.7 / 79.4
ReasoNet (Shen et al., 2016)	- / -	70.6 / 79.4
Ruminating Reader (Gong & Bowman, 2017)	- / -	70.6 / 79.5
jNet (Zhang et al., 2017)	- / -	70.6 / 79.8
Interactive AoA Reader (Joint Laboratory of HIT and iFLYTEK Research)	- / -	71.2 / 79.9
R-NET (Wang et al., 2017)	**71.1 / 79.5**	**71.3 / 79.7**
R-NET (March 2017)	**72.3 / 80.6**	**72.3 / 80.7**
Ensemble model		
Fine-Grained Gating (Yang et al., 2016)	62.4 / 73.4	62.5 / 73.3
Match-LSTM with Ans-Ptr (Wang & Jiang, 2016b)	67.6 / 76.8	67.9 / 77.0
QFASE (NUS)	- / -	71.9 / 80.0
Dynamic Coattention Networks (Xiong et al., 2016)	70.3 / 79.4	71.6 / 80.4
T-gating (Peking University)	- / -	72.8 / 81.0
Multi-Perspective Matching (Wang et al., 2016)	- / -	73.8 / 81.3
jNet (Zhang et al., 2017)	- / -	73.0 / 81.5
BiDAF (Seo et al., 2016)	- / -	73.7 / 81.5
SEDT+BiDAF (Liu et al., 2017)	- / -	73.7 / 81.5
Mnemonic Reader (NUDT & Fudan University)	- / -	73.7 / 81.7
ReasoNet (Shen et al., 2016)	- / -	75.0 / 82.6
R-NET (Wang et al., 2017)	**75.6 / 82.8**	**75.9 / 82.9**
R-NET (March 2017)	**76.7 / 83.7**	**76.9 / 84.0**
Human Performance (Rajpurkar et al., 2016)	- / -	82.3 / 91.2

表 2　　SQuAD 数据集下 R-NET 与其他方法的效果比较

Single Model	**ROUGE-L / BLEU1**
FastQAExt (Weissenborn et al., 2017)	33.7 / 33.9
Prediction (Wang & Jiang, 2016b)	37.3 / 40.7
ReasoNet (Shen et al., 2016)	38.8 / 39.9
R-NET	42.9 / 42.2

表 3　MS-MARCO 数据集下 R-NET 与他方法的效果比较

A.4.3 MS-MARCO 实验结果

我们还将我们的方法应用于 MS-MARCO 数据集（Nguye 等人，2016）。MS-MARCO 是另一个机器阅读理解数据集，与 SQuAD 主要有两个区别。在 MS-MARCO 数据集中，每一个问题有几个相应的段落，所以我们按照在数据集中给出的顺序简单地连接一个问题的所有段落形成一个大段落。其次，MS-MARCO 的答案不一定是文章段落中的某一文本片段，因此 MS-MARCO 数据集

中的官方评估指标是 BLEU 和 ROUGE-L，这两个指标广泛应用于许多与自然语言处理相关的领域。在本次实验中，我们选择与参考答案具有最高 ROUGE-L 得分的答案作为训练中的标准答案，并将预测最高得分最高的文本片段作为预测答案。我们在 MS-MARCO 数据集上训练了我们的模型，结果如表 3 所示，我们的方法在该数据集上也超过了之前的所有方法。

A.4.4 讨论

在本节中，我们将重点介绍一些我们尝试提高模型效果但最终毫无作用的方法。根据 SQuAD 数据集的经验，我们测试产生的最好结果仅仅在于我们完全严格按照所述参数设定的情况下才能获得。所以这些总结出来的参数设定对于其他数据集不一定行之有效。我们相信探索对于问答数据集通用的一些关于参数设定的规律总结是有价值的研究课题，我们正在用不同的模型来测试这些想法。

句子排名。在 SQuAD 数据集中，一段文字往往由几句话组成。而最终和问题相关的答案往往是某一句话其中的一个片段。因此很自然的我们就考虑句子的排序是否将有助于帮我们找到最终答案。我们尝试了两种方式来整合句子排名：（a）我们单独训练了一个句子排名模型，并将这个模型与我们之前讨论的模型结合起来; （b）我们将答案范围预测和句子预测视为两个相关任务，并训练了一个多任务模型。然而这两种方法都没有改进最终的结果。分析表明，句子模型在句子预测上的精准度竟然不即原模型预测模型。我们最好的句子模型实现了 86％的准确率，而我们的原始预测模型预测句子的准确率超过了 92％。这表明精确的答案跨度信息预测在句子预测中确实是至关重要的。

语法信息。我们尝试了三种方法来将语法信息集成到我们的模型中。首先，我们尝试在编码层中添加一些语法特征作为输入。这些语法功能包括 POS 标签，NER 结果，线性化 PCFG 树标签和依赖关系标签等。其次，我们尝试在使用编码后的集成树-LSTM 模块层。我们使用多输入 LSTM 来构建依赖树路径，隐藏层状态既包括之自上而下的路径也包括自下而上路径。最后，我们尝试给模型添加依赖关系解析并将其作为附加任务。然而以上所有这些做法都不能在 SQuAD 数据集上为我们的模型带来任何改进。

多跳推理。我们尝试在指针层中添加多跳推理模块，但仍未能在当前的 R-NET 网络结构模型上获得改进。一个原因可能是需要这样的问题推论太复杂，无法在当前设置下有效地学习，特别是考虑到在 SQuAD 数据集 中没有关于显式推理过程的标注。

问题生成。对于数据驱动的学习方法，人工标注数据可能成为制约模型效果的瓶颈。虽然互联网中的文本语料很丰富，但也不容易找到类似于 SQuAD 数据集这种风格的文章答案对。为了产生更多的数据，我们使用 SQuAD 数据集训练了一个序列到序列的问题生成模型（Zho 等人，2017），并利用该模型生成了来自英语维基百科语料的大量伪问题文章对。我们训练这个伪语料库的 R-NET 模型与 SQuAD 训练数据一起，我们给自动生成样本分配了的权重较小，使伪语料库的总权重和真正语料大致相等。到目前为止，这种做法在最终结果上没有取得任何改进。分析表明，我们产生的伪问题的质量还需要改进。

A.5 相关工作

阅读理解和问题回答数据集。对数据集的基准测试在近期阅读理解和问答回答研究进展中发挥了重要作用。现有的数据集可以根据是否人工标注分为两类。通过人工标注的数据总是高质量的（Richardson 等人，2013; Berant 等人，2014;Yang 等人，2015），但是对于训练现代数据密集型模型来说太小了。那些是自动的由自然发生的数据产生的数据可能非常大（Hill 等人，2016; Hermann 等人，2015），这允许训练的模型更具描述能力。但是，他们是在填空式风格下产生的，其目的是预测一个段落中的空缺词（通常是命名实体）。而且，Chen 等人（2016）显示，CNN / Daily News 数据集（Hermann 等人，2015）所需要的推理比以前想到的要少很多，并得出结论，在这种数据既下模型的表现几乎饱和。

与上述数据集不同，SQuAD 提供了大量高质量的问答数据。答案在 SQuad 通常包括非实体，可以是更长的短语，这比填空式数据集更具挑战性。此外，Rajpurkar 等人 （2016）显示数据集保留了不同的集合答案，需要不同形式的逻辑推理，包括多句推理。MS-MARCO（Nguyen 等人，2016）也是一个大规模的数据集。数据集中的问题是真实的通过 Bing 或 Cortana 存储的匿名查询得到的，所有文章都是与问题相关的网页。数据集中的每个问题与几个段落或文章相关联。但答案是由人生成的，这与 SQuAD 数据集不同，答案必须是文章文本中的某一文本片段。

端到端阅读理解神经网络模型与填空式数据集相融合，形成了几个强大的深度学习模式（Hermann 等人，2015; Hill 等人，2016; Chen 等人，2016; Kadlec 等人：2016; Sordoni 等人，2016; Cui 等人，2016; Trischler 等人，2016; Dhingra 等人，2016;Chen 等人，2016），这些模型都已经被应用来解决问答问题。Hermann 等人（2015）首先提出了注意机制的阅读理解。Hill 等人（2016）针对 CBT 数据集提出了一个基于窗口的记忆网络。Kadlec 等人

（2016）引入了一个关注缺失项的指针网络来预测缺失的实体。Sordoni 等人（2016）提出了一个迭代的注意机制能够更好地模拟问题和文章之间的关系。Trischler 等人（2016）解决通过结合注意力模型与重新排列模型来解决问题的答案任务。 Dhingra 等人 （2016）提出通过乘法门控迭代选择文章的重要部分功能与问题表示。Cui 等人（2016）提出了针对文章和问题双向关注机制相互编码。 Shen 等人（2016）提出迭代推测用动态数量的推理步骤回答，并接受强化学习的训练。

基于神经网络的模型在 SQuAD 数据集上证明了有效性。Wang&Jiang（2016b）结合了匹配 LSTM 和指针网络来产生答案的边界。Xiong 等人（2016）和 Seo 等人 （2016）采用变体的互注意机制来匹配问题与答案。Xiong 等人（2016）提出了一种迭代推断的动态指针网络。Yu 等人（2016）和 Lee 等人（2016）通过对连续的文本片段进行排名来测试 SQuAD 数据集。Yang 等人（2016）提出了一种细粒度的门控机制，动态结合词级和字符级表示，并模拟问题和段落之间的相互作用。Wang 等人（2016）提出将文章的上下文与问题问题进行多角度匹配的方法。

与上述模型不同，我们在模型中引入了自匹配注意机制。它动态地通过查看整个段落和总结文章信息来提炼文章代表与当前文章中的词语和问题相关，让我们的模式充分利用文章信息。在几篇论文中，很多人已经提出了加强对词语语境的重视。Ling 等人（2015）提出，要考虑到基于窗口的上下文单词，同时考虑单词和其相对位置。Cheng 等人（2016）提出了一种新颖的 LSTM 网络来对一个句子中的单词进行编码其中考虑正在处理的当前令牌与其过去令牌之间的关系记忆。 Parikh 等人（2016）将此方法应用于根据单词对单词进行编码形式及其距离。由于与问题相关的通过信息对推断有帮助在阅读理解中，我们应用基于问题表征的自我匹配并注意注意力循环网络。它有助于我们的模型主要关注问题相关通知中的证据，并动态地查看整个段落来汇总证据。

我们模型的另一个关键组成部分是注意力循环网络，这已经证明了在广泛的任务中取得了不错的成绩。Bahdanau 等人（2014 年）首先提出注意力循环网络，并将其应用在在产生目标字词、词语推断和文本对齐中。Hillmann 等人（2015）将阅读理解引入词汇层面的注意力来模拟相互作用问题和段落 Rocktaschel 等人 （2015）和 Wang&Jiang（2016a）提出确定通过逐字匹配来实现该目的。基于注意力的循环网络是一种基于注意力的循环网络的变体，具有额外的门限，模拟文章不同部分所包含的信息对不同问题的重要性不同。

A.6 结论

在本文中，我们提出了 R-NET 阅读理解和问答模型。介绍了基于注意力的循环网络和自适应关注机制获取问题和段落的表示，然后使用指针网络来定位回答边界。我们的模型在 SQuAD 和 MS-MARCO 数据集上都获得了最很好的成果，超过之前许多表现很好方法。对于未来的工作，我们将尝试让语法和知识库信息输入我们的系统。此外，我们也在设计新的网络结构来处理需要复杂推论的问题。