结合传统计数特征和基于词嵌入特征的中文问答方法

A Chinese Question Answering Approach Integrating Count-based and Embedding-based Features

学科专业：模式识别与智能系统
研　究　生：王本友
指导教师：宋大为 教授

天津大学计算机科学与技术学院

二〇一六 年 十一 月

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 **天津大学** 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名:　　签字日期:　　年　　月　　日

学位论文版权使用授权书

本学位论文作者完全了解 **天津大学** 有关保留、使用学位论文的规定。特授权 **天津大学** 可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

(保密的学位论文在解密后适用本授权说明)

学位论文作者签名:　　导师签名:

签字日期:　　年　　月　　日　签字日期:　　年　　月　　日

摘　　要

计算机对文本的表示和理解一直是一个我们亟待解决并一直在尝试的问题。在以前的信息检索任务中，被检索的文档比较长，已经包含了非常丰富的信息。但是在流行的文本问答等任务中，问题和答案文本都是比较短，对这种短文本做匹配任务就需要更加精细的建模。基于词分布式假设发展而来的词向量技术，使得语音数据和图像数据大放异彩的深度学习也能够在文本领域有用武之地。深度学习在自动学习端对端的输入到输出之间的复杂非线性关系尤为出色，但是却要完全放弃我们已有的丰富的领域知识构建而成的丰富特征。为了利用深度学习的巨大潜力和传统的文本建模方法，融合了传统的基于计数的特征和深度网络学习的特征来预测问答对的匹配程度。除此以外，中文由于其很多独特的地方，例如需要分词、字词不同级别的语义以及不同的语法和表达习惯，集合英文问答任务积累的丰富的研究工作给的借鉴，我们还结合了中文的特点去建模到传统计数模型和嵌入模型，最后的特征融合和模型融合的策略取得了较好的效果，显著优于基础的基线模型。

关键词： 问答，语义匹配，词向量，文本建模

ABSTRACT

Document-based Question Answering system, which needs to match semantically the short text pairs, has gradually become an important topic in the fields of natural language processing and information retrieval. Question Answering system based on English corpus has developed rapidly with the utilization of the deep learning technology, whereas an effective Chinese-customized system needs to be paid more attention. Thus, we explore a Question Answering system which is characterized in Chinese for the QA task of NLPCC. In our approach, the ordered sequential information of text and deep matching of semantics of Chinese textual pairs have been captured by our count-based traditional methods and embedding-based neural network. The ensemble strategy has achieved a good performance which is much stronger than the provided baselines.

KEY WORDS: Question Answering, Semantic Matching, Word Embedding, text modeling

目　录

第1章 绪论

1.1 研究背景

随着数据量的规模增长和计算模型的不断发展，深度学习[1]已经在语音数据[2]和图像数据[3]大放异彩，很大原因是基于语音数据和图像数据是底层的原始输入信号，深度的神经网络适于一层一层去抽象出其深层（且有利于相应标签预测的）的表征。但是文本数据中的小粒度的字或是词已经是高度抽象的认知实体，因此文本数据曾被认为深度网络并没有多少潜力的领域。

分布式假设[4]构建词向量的应用[5,6]改变了这一观点，大量的基于词向量的深度模型涌入文本处理的领域。

自然语言处理的任务根据输入输出不同主要有以下几种类别：

- 文本分类: 例如话题分类和情感分类。
- 文本生成: 例如语言模型。
- 文本匹配: 例如搜索和问答系统。
- 文本转换: 例如翻译、改写、摘要和单轮对话。
- 结构预测: 例如分词和依存分析。

以上自然语言处理任务的区别在于端对端的输入输出的不一样。基于候选文档集的问答匹配任务中，模型的输入是一对文本串，分别是问题和答案。由于问题文本和答案文本的长度较短，传统的基于词袋模型的信息检索模型无法克服稀疏问题。一个好的问答模型需要同时去建模以下两列关键信息

- 句法语法等结构信息: 文本的的句法分析信息，特别是问题文本一般有着丰富的句子结构，不同结果的语义贡献不一样。
- 语义匹配: 克服词项独立性假设，能够语义匹配两个句子，即使两个句子没有相同的词语或者字。

而一些人工挑选的基于同义词和同义表达很难较完备地覆盖所有的改写模式，由外部数据训练的分布式词（字）向量的引入一定程度上缓解了稀疏问题。词向量把有着相同上下文的词汇嵌入到低维语义空间的近邻位置。然后通过神经网络再去组合这些嵌入向量最终得到文本的表示。完全由神经网络得到的文本匹配分数完全由数据驱动，通过领域知识提出的丰富特征并不能完美地覆盖了网络中去。我们将神经网络得到的特征和传统模型得到的特征同时加入到一个复杂的回归器，得到的结果当成是最终的预测的结果。

1.2 论文结构

本文分一下几章：

- 第二章-文本表征: 讲述传统的和基于词向量的文本表征方法。
- 第三章-回归预测: 具体地介绍了回归问题中的集成学习策略，去融合所有的特征做到最好的预测效果。
- 第四章-中文问答方法: 介绍我们做问答的具体做法。
- 第五章-结论和思考: 对我们的做法的一些总结和对未来工作的展望。

第2章 文本表征

在计算机科学的很多领域，表示学习是一个非常重要的问题[7]。在图像和语音领域，为了完成一些模式识别的任务，一般通过经验和领域知识去对原始数据输入进行人工总结，根据最后的目标任务的效果去手工调整特征的构造和组合。目标任务预测和中间表示被分开成两个独立的任务，只有通过人工的手动桥接起两个过程，一种经验的靠近最优的方式得到一个中间表示，最后服务于目标预测任务。最近一种基于神经网络的误差反向传播的做法替代了手工地准备数据特征表示，力求做到端对端的输入输出，中间的所有过程都交给数据驱动的自动调节。但是对于文本而言，由于输入是一堆经过高度抽象的词汇实体，神经网络也无法在不借助外部语料的前提下，仅仅从当前文本里面的词汇本身来加以抽象其具体信息。分布式假设[4]根据外部词汇共现信息将文本还原成词汇更细粒度的表示，让神经网络能够在文本中也能够大展身手。在分布式表示之前，研究者们还尝试过以计数为基础的语义表示,比如基于TDIDF向量空间模型、基于矩阵分解的潜在语义索引以及基于潜在话题的概率模型pLSA和LDA。

2.1 传统的基于计数的文本表示

传统的布尔模型和向量空间模型都把两个不同的词当作向量中的两个正交的维度，但是这个“词项独立性假设”直接忽略了词与词之间可能存在的语义关联（譬如“一义多词”），以及一个词在不同上下文语境下的不同含义被当成了单一语义（“一词多义”）。由于要维护整个词汇表的计数信息，会导致“维度灾难”,此项独立性假设没有考虑词项与词项之间的“信息熵”，忽略了词项之间的可能存在的语义关联。LSI一定程度上降低了维度，但是奇异值分解的算法时间复杂度较高，于是主题模型应运而生，主要有pLSI和LDA，pLSI是一个三层贝叶斯生成模型，在文档和词项之间加了一层主题，潜在主题可以去抽象出词的高层话题层面的含义，让语义相关的词通过话题来链接起来；LDA模型在pLDA的基础上加了一个超参数，该超参数是话题生成和话题词语分布的共轭先验分布（原分布系多项分布，其共轭先验分布式Dirichlet分布），使得参数估计能够有一个可以人工经验指导的先验。

2.1.1 布尔模型和向量空间模型

信息检索的最开始的模型就是布尔模型，模型的特点就是文档中是否有你要的关键词，有还是没有。我们来两个例子：第一个查询词是“霸王别姬”（要分词和去除停用词）。

- 第一篇文档：“项羽挥别红颜，自刎而死”
- 第二篇文档：“霸王：今日里败阵归心神不定，虞姬：劝大王休愁闷且放宽心。霸王：怎奈他十面敌如何接应！虞姬：且忍耐守阵地等候救兵。霸王：没奈何饮琼浆消愁解闷。”
- 第三篇文档：“人生苦短，‘霸王别姬’这种生离死别、气势恢宏的故事永远不会发生在我们的身边，更多的是柴米油盐酱醋茶还有隔壁的鸡毛蒜皮。所以人这一生就是要开心……”

布尔模型的缺点很明显明显，

- 无法排序：只是无法排序结果，只是简单判断检索词出现了没有，文档三的主题并不是讲述霸王和虞姬的故事，但是被检索出来了。
- 词义多样性：完全漠视“一词多义”和“多词同义”的中文特点。

后来成熟一点的是向量空间模型（Vector Space Model）,比较行之有效的基于TF-IDF的衡量方法，TF-IDF是信息检索最重要的概念之一，信息论上给出了TF-IDF的很多数学上的解释，因为他们不是基于纯粹的经验得出来的，而是有着深厚的数学基础的奠基。文档可以看作是一些词的集合（无序词袋模型），我们把查询词在该文档中出现的频率（TF）和词本身在整个文档集中的区分度（IDF）乘起来作为衡量文档与词相关性的一个重要指标。TF（term frequency）是一个归一化的计数指标，与布尔空间的计数并无二致。简单地定义为：

$$tf_i = \frac{\#\{w_i\}}{\sum_{w_i \in D} \#\{w_i\}} \tag{2-1}$$

$\#\{w_i\}$是w_i在当前文档出现的次数，分母是当前文档的词的总个数（重复计多个），为了避免向量内积计算中的0值问题，我们会对tf_i做一个平滑，平滑的方式[8]有很多。最经典的平滑方式之一是拉普拉斯平滑，给每一个词给一个较低的频率，然后加上一个折扣之后的原始词频。

对于布尔模型或者TF模型，只考虑词频也会遇到另外一个问题，那就每个词对整句话的贡献其实不一样。对于文章中的语气助词，即使另外一篇文章中大量出现相同的语气助词，也不能认为两篇文档有很高地相关性。词的权重也是一个非常需要考虑的信息，也就是每个的区分能力。有着信息论的基础，逆文档频率（inverse docuemnt frequency）应运而生。它反映了一个词在整个文档集上“露

面”的次数。具体定义为

$$idf_i = log(\frac{N}{df_i}) \tag{2-2}$$

N是文档集总数，df_i是包含词w_i的文档数量。

向量空间模型的每一个词的值表示为

$$tdidf_i = tf_i \cdot idf_i \tag{2-3}$$

有了TFIDF这样一个连续值域的向量表示，我们简单的计算机两个文档的TFIDF向量就可以得到一个实数值，表示文档之间的相关性（我们把查询也看做是一个简单的文档）。这个好处是能够对结果进行排序了。但是第二个对“一词多义”和“多词同义”的情况解决的不是很好。问题是这两个模型都是基于一个很重要的独立性假设，就是著名的“词项独立假设”，我们很容易去知道“霸王别姬”“项羽挥别颜”是一个意思，但是我们的量空间模型认“霸王”和“项羽”这两个词是完全正交的两维，没有考虑他们的互信息，没有考虑他们的潜在语义上的联系。那么问题来了，“怎么去挖掘词语和词语间的潜在语义?

2.1.2 潜在语义索引

潜在语义索引(LSI)[9]又叫潜在语义分析(LSA), 利用奇异值分解的技术（SVD）将文档词汇矩阵分解成

$$M = U^T \cdot \varepsilon \cdot V \tag{2-4}$$

U^T的含义是主题的文档与潜在主题的关联矩阵, ε对角阵就是各潜在主题的重要性（类似于主题的IDF）, V的含义是主题与词项的关联矩阵。通过分解之后的三个矩阵，其中中间的对角阵中较小和等于0的元素会被剔除，然后再去重构整个矩阵，在重构误差和降维之间的tradeoff。直观上看，原本每一个文档对应整个词汇表上的一个技术信息，一般是成千上万维，但是我们最后的主题向量一般会少好几个数量级，这就是一个降维的过程。除去降维以外，还可以对原始共现矩阵的一个降噪的功能，对于奇异值比较小的潜在语义一般认为是整个文档集的噪音，一定程度上被模型予以忽略了。

在文档词项共现矩阵M中，如果词汇表的长度为m,文档矩阵的长度是n。矩阵一共有m*n项，由于文档中包含的词有限，所以这个矩阵的稀疏性不言而喻，造成维数灾难；甚至在web环境下，文档集还会更新，一段时间之后需要重新计算词的权重；进行奇异值分解之后得到的矩阵的含义更加明确，去除了很多值为0 或者接近0的项（类似于降噪了），一定程度进行反映了潜在语义。LSI模型还是有些问题的，第一是算法的潜在主题的含义难以理解，缺乏有效的解释，第二是奇异值分解的算法的时间复杂度有点高，第三对新的文档无法进行推断

2.1.3 概率化潜在语义索引

概率化潜在语义索引（PLSI）[10]又称为概率化潜在语义分析（PLSA）。PLSI模型是一个生成模型。文档中出现一个单词是这样一个生成过程：

- 以P_d的概率选择一篇文档d
- 确定文档后以$P(z|d)$的概率选择主题z
- 选择主题后，以$P(w|z)$的概率选择词w

PLSI也基于一个独立性假设，就是文档选择主题和主题选择词语的两个过程是完全独立的。选定一篇文档后，文档-主题的z的分布θ是一个多项分布，即$\theta = (p_1, p_2, p_3, ...p_n)$,˘$\sum_{0i<n} p_i = 1$,且$\sum(p_i) = 1$，同样确定主题后选择词语的分布也是一个多项分布。

$$P(d, w) = p(d) * p(w|d) \tag{2-5}$$

被转化为

$$P(d, w) = \sum_{z \in Z} p(z) * p(w|z) * p(w|z) \tag{2-6}$$

把文档当作是主题的集合。Z是一个隐含的变量，我们可以通过EM算法去估计参数。pLSI的问题也有，主要是两个方面，第一是EM算法的时间复杂度不低。第二是文档和词项增加，PLSI的参数也呈线性增加，如果有m篇文档、k个主题和n个词，文档选择主题的z的分布θ有$m * k$个参数，确定主题后选择词语的分布有$n * k$个参数。第三是会出现过度拟合的情况。

2.1.4 潜在狄利克雷分布

潜在狄利克雷分布(LDA)[11]是在pLSA的基础加上了两个先验。LDA方法使生成的文档可以包含多个主题，该模型使用下面方法生成1个文档：

- 采样$\theta \sim Dir(\theta)$;
- 对于文档d里面的每一个词w_n;
- 采样$z_n \sim p(z|\phi)$;
- 采样$w_n \sim p(w|z)$;

其中θ是一个主题向量，是一个服从多项分布的概率分布，及各项维当前文档在该主题上的分布，向量中每维非负且和为1；$p(\theta)$是θ的分布，是一个超分布。取其共轭先验分布即Dirichlet分布。除开其加上的先验，其生成模型同上一节的pLSI。我们设定一个θ 和ϕ的先验之后，需要遍历所有文本的文本去推断潜在的主题信息，进而求出所有的文本的主题分布θ 和对应主题的词语分布ϕ LDA模型作者Blei的原文使用变分推断参数，但是现在一般采用基于蒙塔克罗的Gibbs采样法。LDA模型也有以下几个缺点

- 计算文档之间的相关性（语义相似性）。训练好的模型可以推断出文档所属的主题的分布，进而两个文档的相关性可以有两个多项分布的距离来计算，比如交叉熵或者一个余弦距离。
- 缓解多义词的问题。比如“苹果”可以是一种生长在温带的一种水果，也是一家注册在硅谷的一个卖手机的公司，也可以就是一部手机。有了训练好的模型之后，我们不需要仅仅停留在词语级别去判断两篇文档的相似性，从高度抽象出的主题层次，因为其主题层次是对词语级别做了一个聚类，已经去能够根据上下文鉴别出了当前一词多义的词的一个所属含义。
- 一定程度上降噪。文本数据里面的噪音一般是一些次要的主题。训练过程中会被过滤掉，只保留有着丰富语义的主题。
- 无监督。只是根据文本的共现数据，并不需要任何人工标记的数据。
- 不是特定语言敏感的，是一个通用的做法。任何能够分成一个词序列的语言都可以做这样的主题模型综上所述，LDA确实是一个有效的语言高阶主题建模的方法，基于吉布斯采样的模型已经被越来越多的接受了。由于不需要人工标注的监督数据，LDA主题模型可以用来建模大规模的语料，但是由于其还是基于词袋模型，并没有很好的建模词序。而且主题的各维度之间的弱相关性（归一化和为1）与实际的潜在主题之间的可能的关系并不一致。

2.2 词向量技术

词向量，又称为词嵌入（word embedding）。相较于上一章节的词汇共现信息，来挖掘文本中的浅层语义（话题），词向量是一个基于分布式假设[4]的技术，该假设认为有着相同上下文的词应该有着更加接近的表示。得益于神经网络的灵活性和完全的数据驱动，通过大规模语料学习到一个能够表征字词的低维表示。基于这样的一个表示，再去构建复杂的任务。

以下各章节介绍了几种经典的词向量模型。

2.2.1 神经网络语言模型

词向量本来是神经网络语言模型[5]的附属产物，语言模型是一个用来预测当前上下文的下一个词的模型。如下

$$\begin{aligned} P(w_1, w_2, ..., w_m) =& P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \\ & P(w_i|w_1, w_2, ..., w_(i-1))...P(w_m|w_1, w_2, ..., w_(m-1)) \end{aligned} \tag{2-7}$$

对于较长的文本，上述的语言模型会非常复杂。设置不同的上下文文本的长

度得到规模大小不一样语言模型，如N元语言模型近似为

$$P(w_i|w_1, w_2, ..., w_{i-1}) = P(w_i|w_{i-(n-1)}, w_{i-(n-2)}, ..., w_{i-1}) \tag{2-8}$$

超过N个词的上下文距离的词会被忽略，由此可以大幅地降低参数的规模和避免系数问题。当$n = 1$时被称为一元语言模型（unigram model），$n = 2$时被称为二元语言模型（bigram model），以及常用的三元语言模型（trigram model）。较大的N可以保留了词的序列信息，但是同时会增加参数规模和稀疏问题。随着计算机存储越来越易得廉价，存储并不算太大的问题，稀疏问题主要由好的平滑算法[8]去一定程度上缓解。

神经网络语言模型是一个通过神经网络，由上下文信息来预测下一个词的语言模型，其结构如图2-1:

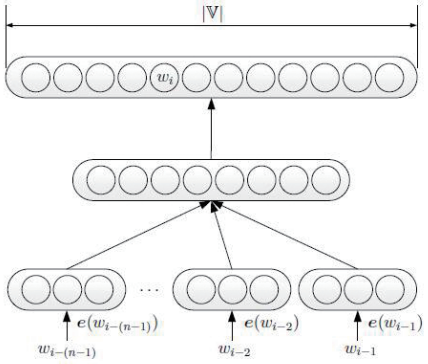

图 2-1　　神经网络语言模型

对于当前的上下文序列$\{w_{i-(n-1)}, w_{i-(n-2)}, ..., w_{i-1}\}$，神经网络最后的优化目标是想让输出层$w_i$对应位置的值为1，其他非当前上下文的位置接近0。神经网络的三层分别是

- 输入层查表找到词向量，将其拼接起来$x = [e(w_{i-(n-1)}); e(w_{i-(n-2)}); ..., e(w_{i-1})]$
- 隐层 $\mathbf{h} = tanh(b^{(1)} + H\mathbf{x})$
- 输出层 $\mathbf{y} = b^{(2)} + W\mathbf{x} + U\mathbf{h}$

最后的输出层的数据需要规范到一个0到1的区间(概率值的归一化要求)里，在最后的输出层加一个softmax层

$$P(w_i|w_1, w_2, ..., w_{i-1}) = \frac{e^{y_i}}{\sum_{0<k<|\mathcal{V}|} e^{y_k}} \tag{2-9}$$

$H \in \mathcal{R}^{|h|\times(n-1)|e|}$, 是一个输入层到隐藏层的权重矩阵，$b^{(1)}$使其偏执项，$|e|$ 是词向量的长度，$|h|$是隐层维度；$U \in \mathcal{R}^{|\mathcal{V}|\times(n-1)|e|}$，$|\mathcal{V}|$是词汇表的长度。$W \in \mathcal{R}^{|h|\times(n-1)|e|}$是一个输入层直接到输出层的权重矩阵，这一项不同于传统的多层神经网络，是

为了加快迭代的速度，但是也同时降低了网络的非线性建模能力。

由上述所知，神经网络语言模型的参数规模非常大，比如U 的长度是$|\mathcal{V}| \times (n-1)|e|$, $|\mathcal{V}|$的长度一般都是非常大，动辄十万百万级，后续有很多优化这一方法的做法。

2.2.2 C&W模型

由于神经网络语言模型需要预测整个词汇表上的生成概率,最后一个概率归一化层的计算量非常大，C&W[12] 改为对一个N元短语打分的方式，分数值以为着这样的短语是一个语料里面很可能出现的短语。网络结构如下图所示 对于整

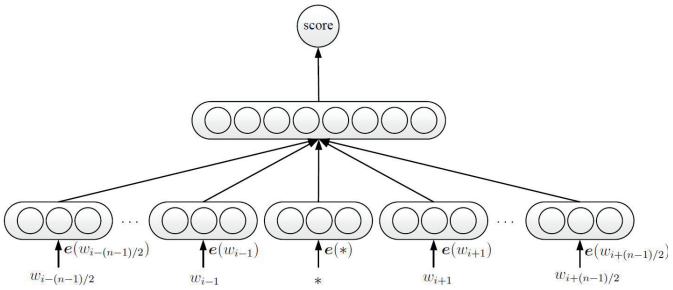

图 2-2 C&W模型

个语料我们是为了最小化下面这一个损失函数。

$$loss(D) = \sum_{(w,c)\in D} \sum_{w'\in\mathcal{V}} max(0, 1 - score(w,c) + score(w',c)) \tag{2-10}$$

(w,c)为从语料中选出的一个n元短语$w_{i-(n-1)/2}, ..., w_{i+(n-1)/2}$ 为奇数，w为当前奇数数列的一个中间词，前$(n-1)/2$和后$(n-1)/2$为其上下文，由其上下文加上一个中间词作为正样本，加上一个错误的词作为负样本，我们希望最后的模型最后得分正样本和负样本有一个间隔。

在替换中间词的过程中，很小的概率会替换出一个正确的中间词，一般替换策略根据所有词的词频来采样，高频词更有可能被选为候选的错误目标词。相比如神经网络语言模型，C&W模型将目标词从输出层拿到了输入层，同时输出的节点的个数从$|\mathcal{V}|$降低至1，复杂的softmax归一化的计算也省略掉，每次前馈时该操作至少计算$|\mathcal{V}|$词。

2.2.3 Cbow模型和Skip-gram模型

Word2Vec[6]的发布，通过很多技巧大幅降低了训练词向量的开销，对之前的模型做了大量的简化。主要分成Cbow模型和Skip-gram模型。

2.2.3.1 CBOW模型

在NNLM的基础上，也是由上下文来预测目标词，借鉴了一些工作[12,13], 做了以下几个简化:

- 删去隐层：CBOW 没有隐藏层，去掉隐藏层之后，模型从神经网络结构直接转化为log 线性结构，与Logistic 回归一致。log 线性结构比三层神经网络结构少了一个矩阵运算，大幅度地提升了模型的训练速度。
- 输入拼接换成相加求平均：CBOW去除了上下文各词的词序信息，使用上下文各词词向量的平均值，代替神经网络语言模型使用的上文各词词向量的拼接
- 输出层由线性变成了一个层级结构：这样计算y_i时不需要计算$O(|\mathcal{V}|)$次，而是$O(log(|\mathcal{V}|))$
- 随机负采样：在对输出进行概率归一化的时候，分母是$\sum_{0<k<|\mathcal{V}|} e^{y_k}$（见公式2-9），采样出部分的负例（采样个数一般远小于$|\mathcal{V}|$）之后，不需要计算$|\mathcal{V}|$。

最后的结构图如图 2-3:

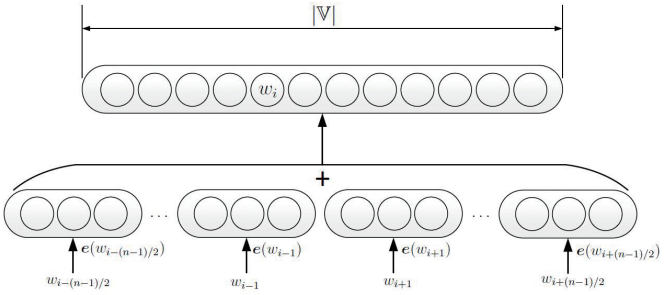

图 2-3 CBOW模型

2.2.3.2 Skip-gram

Skip-gram模型是另外一种模型，结构如图2-4: 与CBOW 模型一样，Skip-gram模型中也没有隐藏层。和CBOW 模型不同的是，Skip-gram 模型每次从目标词w 的上下文c中选择一个词，将其词向量作为模型的输入x，也就是上下文的表示。Skip-gram 模型同样通过上下文预测目标词，预测目标词的概率是

$$p(w_i|C) = \frac{exp(e'(w)^T x)}{\sum_{w' \in (V)} exp(e'(w')^T x)} \tag{2-11}$$

此处用来替换目标词的w'是采样自整个词汇表，每一个目标词对应多个负例（例如5个左右），而不需要计算整个词汇表上的归一化的分母。实际上训练的时

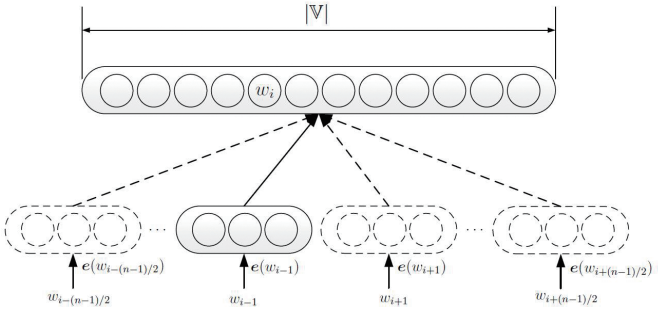

图 2-4 Skip-gram模型

候，因为一些高频词出现次数太多了，但是其并没有太丰富的语义，模型的训练开销大部分放在高频词训练上。二次采样技术会以一定概率丢掉高频词的样本，词在语料中出现的次数越多，被丢掉的概率越大。

2.2.4 词向量技术的比较

当前词向量上下文的定义不一样，或直接拼接有序的上下文单词的词向量，或求一个上下文词向量的平均，有的只是选取一个词的上下文。还有模型层数不一样，由于刚开始NNLM做的会增加一个隐藏层，后续的模型渐渐地去掉了这样一个隐藏层。由于隐层层和上下文建模的复杂度，模型也呈现出不一样的复杂程度，根据模型的复杂程度，也以为需要“喂”不同量级的数据。一般来说自然语言处理领域，由于词汇已经是高度抽象的语义实体，所以并不需要太深的网络，训练词向量过程中的隐藏层一般是不需要的。暂时流行的Cbow和Skip-gram模型中，由于Cbow模型使用了更加复杂的上下文，更适合训练更大语料规模的词向量，如果语料少一点就使用skip-gram模型。Cbow对单词的词性也更加敏感一些。

2.2.4.1 常用评价指标

- 语言学特性：语义相关性、同义词判别和单词类比
- NLP基础任务：基于平均词向量的文本分类、命名实体识别，卷积神经网络的文本分类，词性标注

语义相关性最经典的是WordSim353数据集，每个词对都有若干标注者进行标注对应的相关性，分数越高表示这两个词越相关，实验得出的分数与标准得分的数据计算皮尔逊系数，即是评分的优劣；同义词检测数据集的一个经典的来自托福考试，每一个题目包含一个目标词和四个候选的同义词；单词类比任务主要有语义类比和语法类比[6], 在Word2Vec文章中，该类比任务由词向量的加减法来

完成。如这两类类比的例子

$$\overrightarrow{king} - \overrightarrow{queen} = \overrightarrow{man} - \overrightarrow{woman} \tag{2-12}$$

以及语法类比的任务

$$\overrightarrow{amazing} - \overrightarrow{amazingly} = \overrightarrow{apparent} - \overrightarrow{apparently} \tag{2-13}$$

当前越来越多的研究不再执拗于任务这些语言学任务的效果，因为类比得最好的词向量并不能在上游任务中得到好的结果。语言学特性是词向量锦上添花的特性，并不能算上雪中送炭。在NLP的基础任务中，文本分类任务既可以采用平均句子或者文本的词向量，也可以以词向量为初值的基础上架设卷积网络来服务于最后的预测任务。以及其他的命名实体识别和词性标注等结构预测任务的初值。

2.3 基于词向量技术的文本建模

我们的词向量之后，怎么去组合词向量的语义来得到句子向量是一个很大的问题。一个粗糙的方式去直接求一个词向量相加之后求均值。

$$\vec{D} = \frac{1}{|D|} \sum_{w_i \in D} \vec{w_i} \tag{2-14}$$

这种直接的线性平均，一定程度上忽略了语义的非线性组合，缺少了和标签协同的学习机制。一般现在都会通过网络去学习文档的表示，这样的表示是直接和标签数据有着直接的关系，所以能够表现出更好的性能。根据网络的类型，主要分为卷积神经网络和循环神经网络（以及后续的LSTM和GRU）。

2.3.1 经典词向量表示方法

Mikolov[14]等人提出了一个Distributed Memory Model of Paragraph Vectors (PV-DM).

在我们的段落向量模型中，每个句子被映射到单个向量，其是矩阵D的列。同时，每个词也被映射到独立向量，矩阵W的列。句子向量和这些字矢量被平均或端对端连接以预测文本中的下一个字。在本研究的实验中，我们选择在开始和结束时组合这些矩阵。句子的标记被视为另一个词。它扮演一个“记忆”角色，用于记住当前文本或文章主题中的缺失。因此，我们将该模型称为“段向量的分布式存储器模型”（PV-DM）。上下文是固定长度的，并且从句子的滑动窗口采样。句子向量限于句子的所有上下文，但不超出句子。但是单词向量矩阵W超越了句子。例如，“强大”词向量也对所有句子有效。我们使用随机梯度下降来训练这些句子向量和单词向量，其中通过反向传播获得梯度。在随机梯度下降的每个步骤，可以从随机语句中提取固定长度的上下文，如图所示，从网络计算梯度

误差，并更新模型的参数。在预测阶段时推断计算新的句子向量，也通过梯度上升获得，其余的模型参数、词向量矩阵W和softmax权重是固定的。

在训练之后，这些句子向量可以用作句子的特征。我们可以直接在传统的机器学习技术中使用这些功能，例如逻辑回归，支持向量机或K均值聚类。总之，该算法具有两个关键阶段：

- 通过训练从已知语句获得单词向量矩阵W，softmax权重U，b和句子向量D;
- 第二阶段是推理阶段。通过增加矩阵D中的列数并保持W，U，b不变来减少新句子的句子向量D（不存在）。我们使用D通过基本分类器来标记句子。句子向量的优点：句子向量的一个重要优点是它们的训练集是未被标记的数据，因此它可以用于具有不足的训练样本标签的任务。

句子向量还解决了袋子模型的一些关键弱点。首先，它继承了单词向量的一个重要特征- 单词和单词之间的语义。在语义上，“强”比“巴黎”和“强”更接近。句子向量的第二个优点是它考虑了“字顺序”，并且n元语法模型需要被设置为大的n。这是重要的，因为n元模型在句子中保留大量的信息，包括词序。换句话说，我们的模型优于袋装n元模型，因为后者将呈现极高的尺寸，这将影响效率。

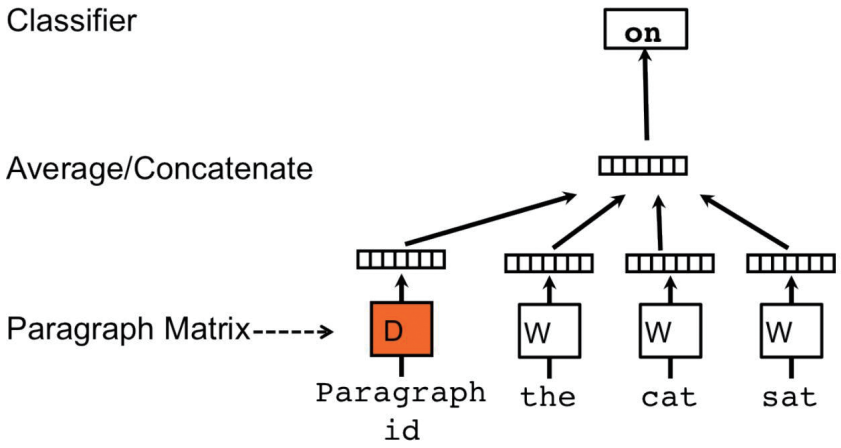

图 2-5　　PV-DM模型

上述方法通过句子向量和单词向量的端到端连接来讨论文本窗口中下一个单词的预测。另一种方法不使用上下文中的单词作为输入，而是强制模型从输出中的句子随机采样出词汇预测词汇。事实上，在每个随机梯度更新的迭代epoch中，选择一个一个文本窗口，然后从文本窗口中采样一个词，然后通过分类任务获得句子向量。如下图所示。 与上一节中提到的PV-DM版本相比，我

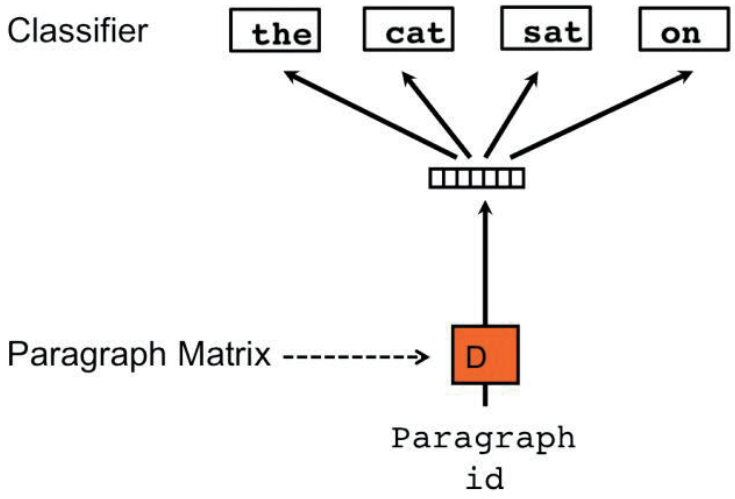

图 2-6 PV-DBOW模型

们将此版本称为段落向量的PV-DBOW：分布式单词包版本。除了在概念上简单之外，模型只需要存储少量的数据。与要存储softmax权重和字矢量的先前模型相比，该模型仅需要存储softmax权重。类似地，该模型类似于Skip-gram模型。在我们的实验中，每个句子向量是两个向量的组合：一个由PV-DM训练，另一个由PV-DBOW训练。PV-DM有着很好的性能，但是当与PV-DBOW结合时，结果更佳。

Collobert[15] 等人提出一个第一个基于卷积神经网络的句子建模。先把一堆词语映射成一组特征，再根据这些特征构建更高阶的特征，其结构图如图所示。

继Collobert之后，Kim等人提出了句子分类模型[16]，Kim模型的特点有

- 三层网络：一次卷积一次pooling 加一个全连接
- 卷积层：卷积核采用二维卷积，宽度固定，卷积窗口选了多个不同的长度，
- pooling：单个feature map 直接max-pooling成一个值
- 全连接：全连接层伴随着dropout和softmax激活函数

该实验使用了多个不同版本的词向量，

- 随机 随机初始化，并在训练过程中被更新
- 静态 由word2vec预训练好的词向量，不断训练过程中更新
- 动态 加载一个预训练好的词向量，并在训练过程微调
- 多通道 多组不同的词向量，当做不同的通道

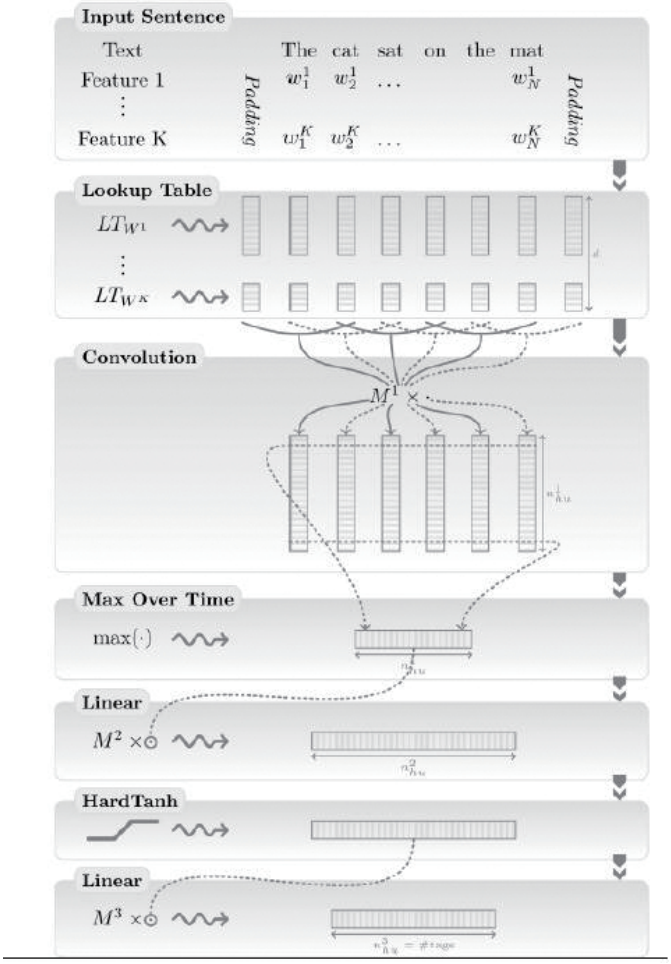

图 2-7 collobert模型

该模型具有所有随机初始化词的baseline模型本身表现并不好。通过使用预训练的向量的性能增益，效果非常好，即使是一个带有静态向量的简单模型也能表现得非常好。这些结果表明预训练的向量是好的，通过的特征提取器（词向量），可以在跨数据集使用。

除了这些使用灵活的卷积神经网络进行建模的方法，一些本来就能够去建模序列化数据的循环神经网络也有很大的潜力去建模句子。

2.3.2 match过程中的文本建模问题

自然语言处理中又很多成对匹配的任务。例如

- 机器翻译：源语言到目标语言的转化
- 文档摘要：长文档到它的短摘要
- 单轮对话：对话系统中的回复
- 问答系统：从候选答案里精准地找到问题的答案

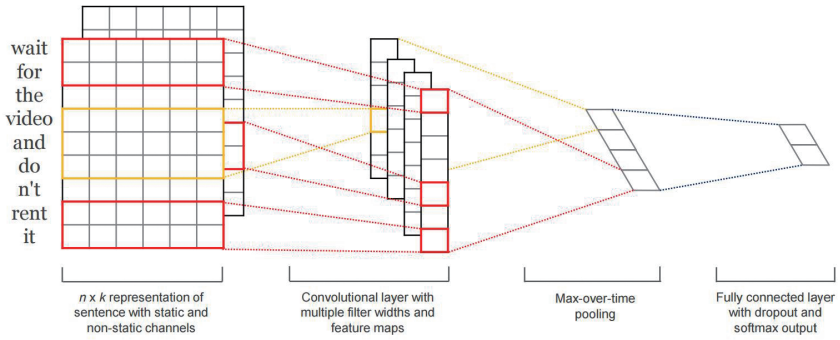

图 2-8 kim模型

- 机器阅读：从问题材料和候选答案里面的短句子之间的匹配
- 文档查询：查询和文档之间的匹配

对于这类成对匹配的任务，固然上一节介绍的很多一维表示模型也可以做到不错的效果，主要做法就是把当前的文档表示为一个单个的向量，比如一些CNN或者是RNN的做法，这类方法比较直观，但是基于这样的一维表示之上的匹配关系却难以体现在网络里，例如

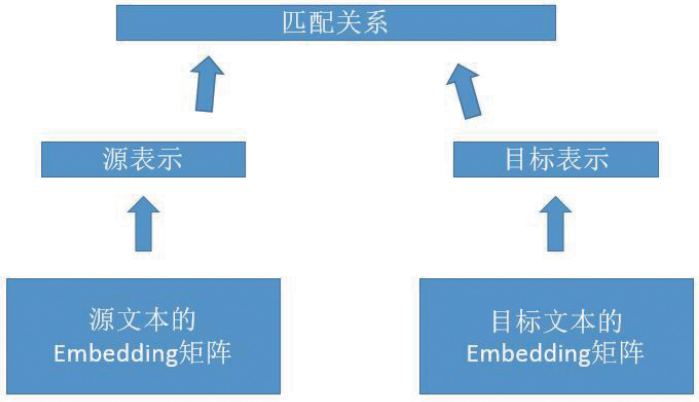

图 2-9 一维匹配模型

这种架构并不能很好的解决两个表示之间的复杂交互，一种基于二维匹配的技术，体现了Attention机制的方式越来越多引起人们的注意。在我们把两个配对的对象分开去建模，之间的复杂交互构建在他们高度抽象之后，因为已经高度抽象的信息是更加低维和高阶的，所以他们不可避免地丢掉了很多有效信息。我

们要在输入是底层的信号（文本里是embedding）时就去建模两者的交互。这种二维匹配的技术就是去做这样的工作。直接用一个矩阵直接建模变长的文本对。

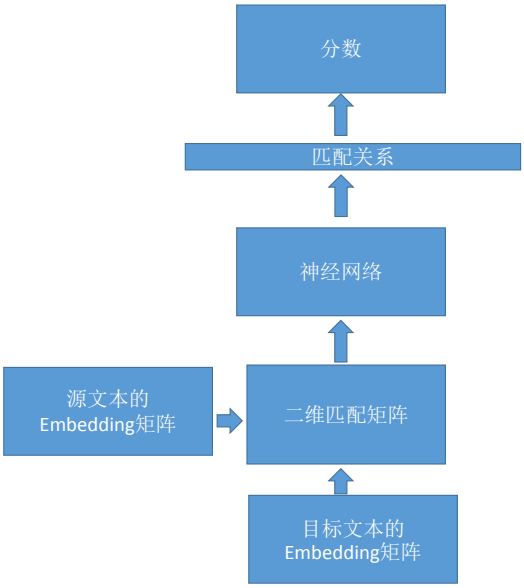

图 2-10　　二维匹配模型

Attention机制起源计算机视觉，盛行于Google Mind的基于Attention机制循环神经网络的工作[17]，该工作在循环神经网络的基础上，加上了一个Attention机制，在学习图像的一个局部时，每次当前状态的学习会直接跟前一个状态学习得到的attention的位置和当前输入的图像相关，只需要去去处理attention位置的像素，可以减少任务的计算量。

后来Attention机制也被引入到自然语言处理的的诸多任务上，如机器翻译[18]，Bahdanau等人在基于神经网络的翻译模型（Nerual machine Translation）上，也就是一个典型的序列到序列的Encoder-Decoder的过程，加上了注意力机制，就是在翻译解码的时候，会有权重直接从源语言的隐层表示传过来，直接联系和对其源语言和目标语言。另外一个神经机器翻译机的工作[19]，尝试结合全局和局部的Attention机制。除了RNN上的工作，Yin 等人建模句子对过程中将注意力机制扩展到CNN[20]。

在一个结对匹配的任务中，对于Attention机制的用法有很多种方式，其本质是为了将两个文本串加上一个显示的权重边联系起来，进行自动加权。

主流的Attention构建方式有下面几种

- 点乘：$f(m_s, m_t) = m_t^T m_s$
- 通用形式：$f(m_s, m_t) = m_t^T W_a m_s$

- 拼接： $f(m_s, m_t) = m_t^T [m_t; m_s]$
- 感知机： $f(m_s, m_t) = v_a^T tanh(W_a m_t + U_a m_S)$

计算完attention之后归一化操作如下：

$$score_t = align(m_s, m_t) = \frac{exp(f(m_s, m_t))}{\sum_{m_{s'} \in S} exp(f(m_s, m_{s'}))} \tag{2-15}$$

2.4 预测方式和技术方式的区别和优劣

前面探讨的基于文档-词汇计数共现矩阵得到文档、主题和词汇之间的联系，主题和词向量都是语义空间的低维表示，其实有很多共通之处，一些人愿意认为其实是两种不同的降维方式。但是他们有着一个区别是，主题是一个更高度层次上内容层次的抽象，而词向量是词汇级别的稍低层次的抽象，类比于图像里面，词向量可能是原始像素，而主题是原始像素构建的网络的里面的后续隐层，包含着更接近人类认知的语义抽象。

Glove [1] 是斯坦福大学一个基于计数的词向量构建方式[21]，能够同时结合较长的上下文和较短的上下文。通过计数的矩阵模型和预测的方式都可以得到词的低维向量表示，Levy和Goldberg[22] 证明了词汇矩阵的奇异值分解与Skip-gram模型（采用负采样技术）具有相同的最优解。Li等人[23] 证明了词汇共现矩阵分解与Skip-gram 模型可以完全等价（用对数概率误差代替了均方根误差，作为SVD的重构误差），相关印证印证了这个观点。以上证明词汇共现矩阵与Skip-gram模型的联系，该模型是计数矩阵和基于预测的神经网络的两个经典特例，都使用了词作为上下文，但是神经网络的优势在于可以进行进一步的深层次的组合，建模更加复杂的上下文。Baroni等人[24]也论证了神经网络的预测模型的优势。

基于词汇共现矩阵或者文档词汇共现矩阵，使用的上下文是一个整个文本，可以包含更加全局和通用的主题层次含义。而词向量一般选取几个词范围内的上下文，得到的是一个局部的词汇层次上的底层抽象的低维表示。对于神经网络而言更加需要的是一个包含更多信息的低抽象层次的语义表示，用以后面的线性或者非线性的组合。计数方式的上下文太宽泛，不适合我们进一步的语义组合。但是一个很重要的问题是，怎么在基于短上下文的神经网络中去表示一词多义的问题，一些工作开始尝试让词向量能够依赖长的上下文[25]，甚至可以一定程度上缓解一词多义的问题[26]。

[1] http://nlp.stanford.edu/projects/glove/

2.5 中文字词向量技术

一般来说，由于中文连续书写的问题，必须先去对文本进行一步切分，才能分成词。词是中文语言语义的最小单位，例如“会议”，单个的“会”和“议”表达的是两种宽泛的动词，但是组合在一起的“会议”才有着较具体的含义。早期中文处理中，一般会去先分词，分完词之后再进一步地进行训练词向量。而越到后来研究者越重视字级别的信息，甚至有结合偏旁部首级别的信息[27]。直观上看，字虽然语义上不够完整，但是有着更底层的含义，更符合神经网络去深度组织语义的原始输入。另一方面由于由字组合起来的词汇量太多，难免会遇到稀疏的问题，而常见的汉字只有3000多个，一定程度上会缓解词向量的未登录词的问题，当出现了未登录词的问题，我们可以考虑用对应的字代替或推测出其含义。而且在一些基本的自然语言处理的任务中，比如分词、命名实体识别等序列标注的问题，本来就是需要字向量来构建的。

相比西文有26个字母组合而成的，中文符号系统有着其很多特色，比如有着很丰富偏旁部首二维组合而成，有着不一样的组合成分，例如字和偏旁部首。通过词向量的训练过程直接考虑到字与字之间的关系、字与词之间的关系甚至其与偏旁部首的关系，从动机上考虑了更丰富的中文特点，应该是有助于后续任务的性能。。大多数词嵌入方法将一个词作为基本单元，并根据词的外部上下文学习嵌入，忽略词的内部结构。然而，在某些语言（如中文）中，单词通常由多个字符组成，并包含丰富的内部信息。一个单词的语义意义也与其组成字符的含义相关。Chen 等人[28] 通过假设词的语义由其中字的意思以及词特有的意思融合而成，并提出一个字符增强字嵌入模型（CWE）。为了解决字符模糊性和非组成字的问题，他们提出了多原型字符嵌入和有效的字选择方法。评估CWE对词相关性计算和类推推理的有效性的结果表明，CWE优于忽略内部字符信息的其他baseline方法。

在诸多的实践中，大家逐渐接受在中文词向量训练过程中利用上字层面的信息。任何一些系统都蕴含了”garbage in, garbage out”的道理，在不引入外部知识的前提下，输入的是什么的信息，模型能够建模的最大能力就已经确定。输入确定了模型能够做到的上限，而我们当前能做的就是设计一个高效的模型能够让输入的信息能够充分被模型所吸收和提取，也就是让模型更能接近它的上限。一个更加有效的中文字词联合模型或许还有很多潜力。

英文文本数据的处理积累了很多的经验，但是中文本身有很多的特点。如何去在中文文本数据本身的特点去构建一个跟任务相关的模型是一个非常直观且也应该行之有效的做法。直接把英文的做法往中文上套并不算是一个非常明智的做法，当然确实有很多通用的东西可以让大家借鉴，但是抱着应用本身去提高

效果的创新新乏善可陈。更多的结合中文特点的做的深度学习模型应该越来越被中文自然语言处理和计算语言学的工作者为之努力。

第3章 回归和排序方法

传统机器学习任务，一般分为监督学习、无监督学习和增强学习。监督学习方法的根据已标记数据的特点去推断没有标记的数据，监督学习方法根据其预测是连续值和离散值的区别，又有回归和分类两种任务。回归任务主要有几种方法

- 线性模型
- 决策树系列算法
- 朴素贝叶斯
- 基于实例的算法，如近邻算法
- 神经网络
- 基于核的算法，如支持向量机

除了以上基本算法，基于boosting和bagging的集成学习算法也是提升预测效果的杀手锏。由于决策树的灵活，由其构造的集成学习算法也被大家广泛使用。最后介绍一下排序学习的相关算法。

3.1 线性回归

一个典型的有监督机器学习过程是去根据有标签数据学习内在的分布，然后对新数据进行推断。例如一个样本有着d个属性$\underline{x} = (x_1; x_2; ...; x_d)$, 线性模型尝试用一个对当前的属性的线性组合来预测最后的结果。

$$f(\underline{x}) = w_1x_1 + w_2x_2 + ... + w_dx_d + b \tag{3-1}$$

由于本身的形式简单，线性模型的效率很高，而且有着直观的可解释性。诸多非线性结构都是在其基础上引入了层级结构和高维映射[29]。当我们选取差平方作为损失函数，我们需要优化的最后的损失函数是

$$loss(w, b) = \sum_{i=1}^{m}(f(x_i) - y_i)^2 \tag{3-2}$$

3.2 集成学习

3.2.1 Boosting 和bagging

分类或者回归这样的预测任务，好的结果常常不是来自单一模型，

往往来自于对当前的若干分类器（回归器）的集成。集成策略主要分成boosting和bagging两种方式。这些单一模型一般称为基学习器（base learner），集成同一类型的学习器属于同质(homogeneous)集成，集成不同类型的学习器称为异质(heterogenous)集成。

PAC学习模型中[30,31]（Probably Approximately Correct），如果有一个多项式级学习算法来识别一组概念，并且能够达到的识别率非常高，那么这组概念是强可学习的。如果该学习算法识别的概念的正确率只是稍微比随机的结果好一点，那么这组概念本事是弱可学习的。PAC学习模型提出了这两种学习本身的等价性问题，能否可以将一个弱学习算法通过某种策略“提升”为强学习算法。如果强学习算法和弱学习算法等价，只有稍微低代价地找到了一个弱学习策略，再根据一些集成学习的方法将其提升成强学习算法，似乎就可以不用去直接找一个强的base学习算法，因为学习一个强学习算法的成本要大得多。

现在具体介绍一下bootstraps: 名字源自“pull up by your own bootstraps”，即使用自己的力量。假如有一个弱学习器（同样适宜于一个强学习器），通过自身对抽样出来的多个子样本集进行训练，根据上一次结果调整下次抽样的子样本，最后多次训练的结果能够增强学习器的学习能力。类似一个差等生，把同样一个题目做多遍，特别是那些做错的题目，最后学习出来的策略一定程度上能够达到勤能补拙的效果。其中一个经典的算法是AdaBoost(Adaptive Boosting)，训练之前初始化赋予每个样本相同的权重，然后根据后续训练的模型的预测结果改变所有样本本身的权重。最后回归问题采取多个学习器的预测的值的加权平均值，权重计算自这些分类器在验证集上的表现。Adaboost按分类结果，分配不同的权重，让模型对分错的数据更敏感。Bootstrap利用分错的信息来对训练样本集做重采样，让错分的样本被进一步学习。后续的GBDT（MART）中的boost的概念跟上述的不一样，GBDT中每步自举的样本集都是整个大的样本集，且其权重都是均匀的，变化的是当前分类回归树的标签数据是更新，依赖于前面所有树的误差（或者说是梯度或残差）

bagging: 名字源自“bootstrap aggregating”的缩写。每次训练的时候随机地从训练集中采样出一个子样本，区别是每次采样的的概率都是不变的均匀分布，而不关心样本是否分类被错，这样利于去做并行。这样我们并行训练好的多个学习器，在最终的预测任务中去做投票，在分类器数量多的情况下，可以做一个均匀的加权结果，即直接同等对待所有学习器的结果，也可以按照学习器在验证集上的准确率（以及回归任务中的各种准确率）来进行按权平均。

Bagging与Boosting的区别：

- Bagging每次样本的采样是基于随机的概率分布，各轮训练相互独立，而Boosting的训练的时候样本的采样与之前训练的结果息息相关。

- Bagging的各个样本没有权重，而Boosting是有权重的，其权重随着训练的过程自动更新，该样本学习的结果离标签值越远，越应该被后续学习器所重视。
- Bagging方法由于其多次训练之间相互独立，所以能够并行训练；Boosting的训练与之前预测的结果息息相关，所以只能串行生成训练。Bagging这种可以并行训练的特点可以节省大量时间开销，特别是现在并行训练的算法和工具越来越成熟。

bagging采用粗糙的均匀采样，Boosting复杂地根据前面的预测结果进行下一次的抽样，因此在大部分dataset中，Boosting的学习性能稍优于Bagging。源自Boosting方法改进的AdaBoost方法也具良好的性能。但是Boosting可能导致退化，导致过拟合。分类器投票方法（Voting组合分类器）就是一种典型的集成机器学习方法，它通过组合多个弱分类器的投票结果来得到一个强分类器。投票分类法因为需要组合多种分类器（不管是同质的还是异质的），所有会有额外的时间开销。比较经典的随机森林也是一个经典的集成学习算法，采用一种行采样和列采样的技术来训练多个决策树，构成一个大的随机森林。

3.2.2 MART

MART（Multiple Additive Regression Tree)[32]，又叫GBDT(Gradient Boosting Decision Tree)，是一种用于回归的机器学习算法，该算法由多棵回归决策树组成。当替换掉目标函数后，该算法可也适用于分类或排序。

由于MART是一个基于boost的树模型，它的输出是多课回归树的输出的线性组合，其单棵回归树会去预测所有样本的结果（但是每棵树的学习目标并不一致，会根据梯度逐步求精）。假设我们给出一个数据集$\{\mathbf{x_i}, y_j\}, i = 1, ..., m$，其中$\mathbf{x_i} \in R^d$且对用的标签$y_i \in R$，对于每一个给定的向量$\mathbf{x_i}$，其特征为$x_{ij}$,其中$j = 1, ...d$，首先考虑一个回归桩（stump），由根节点和两个叶节点组成，其中有向弧将根连接到每个叶子节点。我们首先将所有数据视为驻留在根节点上，对于给定的特征，我们循环遍历所有样本，找到阈值t，使得如果所有样本具有$x_{ij} \le t$落到左子节点，其余落入右子节点，我们定义

$$S_j \equiv \sum_{i \in \mathbf{L}} (y_i - \mu_{\mathbf{L}})^2 + \sum_{i \in \mathbf{R}} (y_i - \mu_{\mathbf{R}})^2 \tag{3-3}$$

其中，$\mathbf{L} = \{i \,|x_{ij} \le t\}$ 以及$\mathbf{R} = \{i \,|x_{ij} > t\}, \mu_{\mathbf{L}}$ 和$\mu_{\mathbf{R}}$，是$\mathbf{L}$ 和$\mathbf{R}$ 集合中的均值，每一个对应的特征j 和对应的阈值t都有一个S_j，全局最小的S_j就是当前的树桩划分方式。就这样一直划分下去就是一个回归树。

MART是一类可以被视为使用回归树在函数空间中执行梯度下降的增强算法。最终模型再次将输入特征向量$\mathbf{x_i} \in \mathbf{R}^d$映射到得分$F(\mathbf{x_i}) \in \mathbf{R}$. MART是一类算

法，而不是单个算法，因为它训练是为了最小化通过的损失函数（例如，求解分类，回归或排序问题）。

$$F_N(\mathbf{x_i}) = \sum_{j=1}^{N} \alpha_j f_j(\mathbf{x_i}) \tag{3-4}$$

其中$f_j(\mathbf{x_i}) \in \mathbf{R}$是一个单个回归树，$\alpha_j \in \mathbf{R}$是对应的回归树的权重。$f_j$和$\alpha_j$都是在训练过程中学习到。

当N棵树已经训练好了，下一棵树怎么训练呢，MART使用gradient下降去降低损失，具体就是

$$\delta C \approx \frac{\partial C(F_n)}{\partial F_n}\delta F \tag{3-5}$$

MART在不同的文献中又两个不同的版本。残差版本认为MART是一个残差迭代树，第N棵回归树是在学习前N-1棵树的残差。Gradient版本把MART说成一个梯度迭代树，使用梯度下降法求解，第N棵回归树是在学习前N-1棵树的梯度下降值。两个版泵都是迭代回归树，都是累加每颗树结果作为最终结果（Multiple Additive Regression Tree)，每棵树都在学习前N-1棵树尚存的不足；两者的不同主要在于每步迭代时，优化目标是最小化残差还是最小化Gradient。残差是全局最优值，Gradient是局部最优方向，即前者每一步都在试图让结果变成最好，后者则每步试图让结果更好一点。直观上看残差是全局优化目标，但是其不够灵活，损失函数固定为残差的均方差，只能去处理通用意义上的回归问题，后者使用Gradient，可以对任意的损失函数。

我们下文里面的lambdaMART就是用的Gradient版本，

3.3 排序学习

3.3.1 RankNet

RankNet[33] 是一个pairwise模型，将排序问题转化成对一对元素排序先后的概率问题，例如比较文档d_i 排在d_j前面的概率。底层模型可以是任何输出模型的模型参数是可微函数的模型（通常我们使用的神经网络，也有基于boost树实现的）。RankNet训练工作如下。对于一个给定的查询，两个文档d_i和d_j有着对应的相关性分数（例如分数相关性是5个等级，等级之间的比较只有相对意义，没有绝对意义，比如4级的文档的相关性不是二级的两倍）.RankNet能够将输入的文档d对应的特征$\mathbf{x} \in \mathcal{R}$映射成一个分数$f(\mathbf{x}) = \mathbf{w}\mathbf{x} + b$，$s_i = f(\mathbf{x_i}), s_j = f(\mathbf{x_j})$。我们把$d_i \triangleright d_j$ 表示当前模型认为d_i 比d_j更相关，其分数的差值$s_i - s_j$用一个sigmoid函数规范化到0到1之间的实数，以满足概率值的归一化要求。

$$P_{ij} = P(d_i \triangleright d_j) \equiv \frac{1}{1 + e^{-\sigma(s_i - s_j)}} \tag{3-6}$$

对数据标签而言，文档d_i和d_j的相关性判断$\bar{S}_{ij}$可以是{1,0,-1}，其中1表示d_i更相关，-1表示d_j更相关，0表示同样相关。由于表示数据里面的相关性是一个离散的相关性标注，一般来说是由主观的标注者标注的，上文叙述也说明这一点，其分数的绝对值的意义不是很明显，而且比较的相对值是更有意义的，所以我们对其目标概率的归一化也只关心两个的文档相关性比较的最后结果，是强弱还是相同，而不关心其差值的绝对值大小。$d_i \triangleright d_j$的概率的标签也需要将其归一化到一个0-1之间的一个概率测度$\bar{P}_{ij} = \frac{1}{2}(1 + \bar{S}_{ij})$

优化的损失函数为d_i比d_j的相关的模型估计值P_{ij}和其标注值$\bar{P}_{ij}$的交叉熵

$$C = -\bar{P}_{ij} log P_{ij} - (1 - \bar{P}_{ij}) log(1 - P_{ij}) \tag{3-7}$$

带入上面的公式有

$$C = \frac{1}{2}(1 - S_{ij})\sigma(s_i - s_j) + log(1 + e^{-\sigma(s_i - s_j)}) \tag{3-8}$$

这样一个损失函数是对称的，易得$S_{ij} = 1$时

$$C = log(1 + e^{-\sigma(s_i - s_j)}) \tag{3-9}$$

$S_{ij} = -1$时

$$C = log(1 + e^{-\sigma(s_j - s_i)}) \tag{3-10}$$

3.3.2 LambdaRank

LambdaRank对RankNet的方法主要改进就是对损失函数进行了因式分解，在损失函数里面考虑了评价函数。

损失函数对$\mathbf{w}$中里面每一个分量w_i求导。

$$\begin{aligned} \frac{\partial C}{\partial w_k} &= \frac{\partial C}{\partial s_i}\frac{\partial s_i}{\partial w_k} + \frac{\partial C}{\partial s_j}\frac{\partial s_j}{\partial w_k} \\ &= \sigma(\frac{1}{2}(1 - S_{ij}) + \frac{1}{1 + e^{-\sigma(s_i - s_j)}})(\frac{\partial s_i}{\partial w_k} - \frac{\partial s_j}{\partial w_k}) \\ &= \lambda_{ij}\left(\frac{\partial s_i}{\partial w_k} - \frac{\partial s_j}{\partial w_k}\right) \end{aligned} \tag{3-11}$$

其中λ_{ij}的值为

$$\lambda_{ij} \equiv \frac{\partial C(s_i - s_j)}{\partial s_i} = \sigma\left(\frac{1}{2}(1 - S_{ij}) + \frac{1}{1 + e^{-\sigma(s_i - s_j)}}\right) \tag{3-12}$$

如果我们要更新w_k，每一个文档对都会对w_k有影响。我们需要更新w_k的总量为

$$\delta w_k = \eta \sum\nolimits_{i,j \in I} \left(\lambda_{ij}\frac{\partial s_i}{\partial w_k} - \lambda_{ij}\frac{\partial s_j}{\partial w_k}\right) \tag{3-13}$$

由于我们需要遍历所有的文档对$\{s_i, s_j\}$，且不能重复。不妨我们令所有的文档对组合为$\mathbf{I}$，$\mathbf{I} = \{(i, j) | d_i \triangleright d_j\}$，即所有的$S_{ij} \equiv 1$

最后我们的λ_i 定义为

$$\lambda_i = \sum_{j:(i,j)\in \mathbf{I}} \lambda_{ij} - \sum_{j:(j,i)\in \mathbf{I}} \lambda_{ij} \tag{3-14}$$

λ_i 是与文档d_i 一一对应的，而且λ_i 的正负也以为着文档d_i的相关性往哪一个方向更新，大小为应该更新的大小程度。以前实现RankNet时，我们使用一个严格意义上的随机梯度下降，按照梯度每对文档（具有不同的标签）更新权重。上面显示，我们可以为每一个文档计算λ，来代替梯度。将所有对的文档（其中一对由具有不同标签的两个文档）的权重更新量求和，然后进行更新。这是mini-batch学习，对于给定的查询，我们一次计算出所有的权重更新大小，让RankNet训练非常显着的加速（因为权重更新是大的开销，例如反向传播的神经网络模型）。

不仅如此，LambdaRank还让评价指标能够与最后的评价指标直接相关，在公式 3-17的基础上，由于上文已经令$S_{ij} \equiv 1$，易得$\frac{1}{2}(1 - S_{ij}) \equiv 0$，再直接乘以一个$|\Delta NDCG|$

$$\lambda_{ij} \equiv \frac{\partial C(s_i - s_j)}{\partial s_i} = \frac{\sigma|\Delta NDCG|}{1 + e^{\sigma(s_i - s_j)}} \tag{3-15}$$

$|\Delta NDCG|$是我们交换文档d_i和d_j的位置后NDCG变化的大小，这样我们优化过程会重点去优化哪些让NDCG提高得更多的方向，最后的评价指标和优化目标是一致的。

搜索排序关注文档列表的顺序而不是绝对值，所以需要一个新的损失函数。排序问题的评价指标一般有NDCG、ERR、MAP、MRR等，这些指标的特点是不平滑、不连续，无法求梯度，因此无法直接用梯度下降法求解。RankNet的创新点在于没有直接对这些指标进行优化，而是间接把优化目标转换为可以求梯度的基于概率的交叉熵损失函数进行求解。因此任何用梯度下降法优化目标函数的模型都可以采用该方法跟RankNet相比，LambdaRank分解因式后训练速度变得更快，同时考虑了评价指标，能够优化到最后评分函数（NDCG评价指标不仅只跟顺序有关，还对顺序在前面的排序更加敏感），效果更好。

3.3.3 LambdaMART

实际的搜索排序会考虑LambdaMART算法，因为其跟排序场景中的损失函数直接先关，在我们学习各个特征本身的权重的时候直接去考虑到了排序结果，甚至是我们对排序结果特化的一个评价指标。LambdaMART迭代Lambda代替残差，乘以步长以逐步接近目标解。在RankNet 和LambdaRank的想法启发下，结合MART，提出了LambdaMART方法，用跟NDCG相关的lambda来代替梯度，当

做当前树的优化目标。

$$\lambda_{ij} = \frac{-\sigma|\Delta Z_{ij}|}{1 + e^{\sigma(s_i - s_j)}} \tag{3-16}$$

$|\Delta Z_{ij}|$是交换d_i和d_j之后评价指标的变换，此处评价指标可以是NDCG也可以是MAP、MRR等等。当前回归树的文档d_i的优化目标同3-17，如下：

$$\lambda_i = \sum_{j:(i,j)\in \mathbf{I}} \lambda_{ij} - \sum_{j:(j,i)\in \mathbf{I}} \lambda_{ij} \tag{3-17}$$

最后，比较LambdaRank和LambdaMART如何更新其参数很有用。LambdaRank在检查每个查询后更新所有权重。另一方面，使用落在节点上的所有数据计算LambdaMART中的决定（节点处的分裂），因此LambdaMART每次仅更新一些参数（即，当前叶节点的分割值），但使用所有数据每个xi落在一些叶子）。这意味着LambdaMART能够选择可能的拆分和叶值LambdaMART有很多优点：

- 排序场景：一般传统方式通过分类或者回归的中间手段去解决排序任务，现在端对端地直接去优化最后的排序指标
- 数学解释非常漂亮：直接将类似于NDCG这样不能导出IR评价指数加入到优化目标里，并让其有了梯度的物理意义，即使NDCG本身不能求导，可以导出函数可以导出，而梯度的实际物理意义，数学解释非常漂亮
- 增量在线学习：由于每个训练都可以在现有模型中继续在线训练
- 特征组合：得益于树模型的灵活性，可以方便地去组合不同的特征
- 特征选择：得益于MART模型，可以学习特征不同重要程度
- 类别均衡：对于具有正和负样本不平衡的数据：由于模型的训练对象具有一对具有不同标签的文档，而不是预测每个文档的标签，所以它对正样本和负样本的不平衡不敏感

第4章　中文问答方法

随着自然语言处理（NLP）和信息检索（IR）技术的发展，Quesion Answering（QA）任务由于其更短的文章需要更精确的匹配，引起了相关领域研究者极大的关注。典型任务之一的名为基于文档的问答任务（DBQA）的侧重于从问题的给定文档候选中找到答案。与传统的文档检索任务相比，DBQA系统通常使用流畅的自然语言来表达查询意图，并且希望得到准确的结果，而且其丢弃了大多数不匹配的候选，一般只保留最好的结果。

由于DBQA任务中文本的长度较短，数据稀疏性已成为比传统检索任务的问题更严重的问题。基于相关性的IR方法如TFIDF或BM-25不能有效地解决这些语义匹配问题。因此，克服稀疏问题的词嵌入技术已经应用于一些英语QA系统以及中文QA系统。此外，问题文本是具有完整语法结构的自然语言，而不是文档检索任务中的一些关键字。句子应被视为序列或树而不是无序的单词袋，并且每个组成部分对整个句子具有不同的语义贡献。总而言之，有效的问答方法需保证同时考虑以下问题。

1）匹配语义相似但是可能被同义改写的文本。

2）考虑问题文本的顺序信息，而不是无序的单词集合。

对于第一个问题，枚举所有的文本的重写规则似乎是不可能的。我们通常采用基于嵌入的方法，其中两个词具有闭合的嵌入表示，当它们通常出现在类似的上下文中。这些表示可以在一定程度上以分布式方式捕获独立术语之间的语义关联。对于第二个问题，人们更可能首先阐述前提，然后根据中国表情习惯在这样的前提下提出相关问题。在词袋模型中，问题中无序的单词集将失去信息以区分前提和问题。我们利用基于计数的模型中的位置感知信息，并在神经网络前向传播操作期间保持一定程度上词或字符序列在神经网络中的顺序。

任务的目的是设计一个方法去从给定文档候选集合中问题的答案，这样的问题就成千上万个。目标答案只应从问题的给定文档其中包含一个答案句子中选择。结果将最终通过评估指标来评估，以确定我们的系统的性能。例如，在训练集中对于给定问题“俄罗斯贝加尔湖的面积有多大？”，参与者应该从候选答案找到正确答案“贝加尔湖长636公里，平均宽48公里，最宽79.4公里，面积3.15 万平方公里”。模型应该在问题和每个答案语句之间生成一组相关性分数，包括测试集中的正确答案。具有标记为问题的标准答案注释的测试集的评估工具箱将根据MRR分数和MAP分数对所有答复句子排序。本文详细阐述了

在NLPCC-ICCPOL 2016中回答基于文档的QA任务的共享子任务之一开放领域问答系统的方法。我们将基于计数和基于嵌入的方法与集合策略相结合。为了适应中文表达习惯，我们将中文的特征集成到基于计数的方法和基于嵌入的方法中，这在最终评估中基线上有显着的改进。

4.1 相关工作

QA任务集中于自动理解自然语言问题以及选择或生成一个或多个可以语义匹配问题的答案。在基于文档的英文QA系统中，已经有很多工作已经有了较好的性能。由于文档比文档检索的传统任务更短，结构化语法信息和词汇鸿沟是QA系统的两个关键点。对于结合句法语法信息，已经提出树状结构[34]和顺序结构[35]来利用句法信息，而不是无序的词袋模型。对于词汇鸿沟而言，如词汇语义[36]，概率改写[37]等工作提出以减轻词汇鸿沟的问题的影响。此外，基于与传统特征的结合的方法[38]试图结合语义和句法信息，通过数据驱动的学习机制对答案排名。

最近，端到端策略激励研究人员建立一个深层次的匹配模型，它也可以建模顺序文本。随着基于词嵌入的神经网络的发展，深度学习在QA任务中获得了良好的性能[39]。Severyn 等人[40]提出一种将有序重叠信息组合成隐层的简单卷积神经网络（CNN）。循环神经网络（RNN）和可以对顺序文本建模的以下长期短期记忆神经网络（LSTM）[41,42]也可应用于文本表示和问题和答案的匹配。Santos等人提出了一种具有双向注意机制的神经网络，用于对两个连续文本之间的交互进行建模，这可以容易地将CNN或RNN网络整合到一起[43]。

4.1.1 问答任务简介

通常问答任务（与对话任务不一样），被分成三种形式

- 基于候选文档的答案选择任务
- 基于知识库和事实的问答
- 基于生成的问答

基于文档的问答任务先去找到一个候选答案文档集合，该集合有或者没有这样的一个问题的答案。任务的关键在于对问题的所有的候选答案做语义匹配然后做排序，一般来说，top1的文档的答案会是最终的答案，也有可能存在问题并没有答案或者有多个答案，此类情况就需要一个类似检索系统的评价指标去评价，例如MAP或者MRR。其一个问题类似以下的形式:

基于知识库的问答任务，知识库已经存在了很大的事实（fact），我们的问题如果能在知识库中找到答案，就会直接返回最后的答案。例如问题像

表 4-1 基于文档的问答的示例.

问题	候选答案	标签
张旭(明代知县)的代表作是什么?	散体诸文，大抵应俗之作矣。	0
张旭(明代知县)的代表作是什么?	《点绛唇》	1
张旭(明代知县)的代表作是什么?	卜尽金钱，闲愁暗逐游思起。	0
张旭(明代知县)的代表作是什么?	云松螺髻。	0
张旭(明代知县)的代表作是什么?	眉减春山翠。	0
张旭(明代知县)的代表作是什么?	试卷珠帘，风香生袂	0
张旭(明代知县)的代表作是什么?	凉如水。	0
张旭(明代知县)的代表作是什么?	惜花心碎。	0
张旭(明代知县)的代表作是什么?	忍把阑干倚。	0

表 4-2 基于知识库的问答的示例.

问题	答案
山羊是属于哪个亚纲的?	真兽亚纲
郑州驱逐舰型号是什么有人知道吗	国产052c型
历史上的刘瑜是哪个年代的人	金
《红楼梦》的导演是哪位?	胡雪扬
体育粉们，我想问一下李娜主要获得过哪些奖项	2011法网女单冠军2014澳网女单冠军wta单打冠军头衔：9
孙悟空在那部轩辕剑作品中出现过	《轩辕剑叁外传：天之痕》

对应的事实知识库类似于有这样的实体-属性-属性值三元组。

表 4-3 基于知识库的问答的示例.

问题	属性	属性值
《红楼梦》(2005上海越剧版)	别名	《红楼梦》
《红楼梦》(2005上海越剧版)	主演	钱惠丽；单仰萍；陈颖；方亚芬
《红楼梦》(2005上海越剧版)	上映时间	2005年
《红楼梦》(2005上海越剧版)	类别	剧情古装伦理
《红楼梦》(2005上海越剧版)	导演	胡雪扬
《红楼梦》(2005上海越剧版)	编剧	徐进
《红楼梦》(2005上海越剧版)	上映地区	中国大陆
《红楼梦》(2005上海越剧版)	语言版本	越剧
《红楼梦》(2005上海越剧版)	画面颜色	彩色

更复杂的基于知识库的问答任务还有复杂的实体与实体之间的关系，需要有一种推理的机制。推理机制一般也在基于生成的问答任务中比较重要，也是我们需要做到的终极目标。要求机器“理解”问题并从合适的数据源中找到答案，并组合成一个合适的句子来回答目标问题。

李航和鲁正东[44] 认为，基于文档的问答任务是较为简单，暂时已被解决得较好的任务。而基于知识库和生成的问答系统还需要更多的努力，也更需要深度挖掘深度学习的潜力的地方。除了以上的问题还有类似的任务，像图像的文本摘

要和文本搜图的任务并不在本文的介绍范围内。

4.1.2 相关问答数据集

本章介绍了几个传统的问答数据集以及相关阅读理解的数据集，他们的任务以及特点。

4.1.2.1 TREC QA

对于回答一个问题而言，问答系统系统返回实际答案，而不是文档的排序列表。TREC自1999年以来一直就有QA track [1]，其任务被定义系统需要检索小片段的文本，其包含内容来自开放域但是问题是闭集的答案。评价使用MRR评估QA任务运行。单个问题的得分是返回第一个正确答案的排名的倒数或返回无正确答案的0。然后，最后结果是测试中的问题集合的平均值。TREC-9开始，报告两个版本的分数：“严格”评价和“宽松评价”，前者候选集无正确答案被计为错误，后者计为正确。

4.1.2.2 Insurance QA

Insurance QA [2] 任务需要在开发集，test1和test2中的为每个问题指定答案候选池。发布的语料库共包含24981个唯一答案，可以使用整个答案空间作为候选池，因此每个问题必须与24 981个答案候选者进行比较。由于计算耗时，这是不切实际的。Feng等人[45] 将池大小设置为500，以让其成一个实际的具有挑战性的任务。当然我们将ground truth也要答案放入池中，从答案空间随机抽取错误答案，直到池大小达到500。

4.1.2.3 Wiki QA

WikiQA语料库[3]是一个新的公开可用的问题和句子对，收集和标注用于开放域问题回答的研究[46]。为了反映一般用户的真实信息需求，我们使用Bing查询日志作为问题源。每个问题都链接到一个可能有答案的维基百科页面。因为维基百科页面的摘要部分提供了关于主题的基本和通常最重要的信息，所以我们在本节中使用句子作为候选答案。在众包的帮助下，我们在数据集中包括3,047个问题和29,258个句子，其中1,473个句子被标记为对应的问题的回答句子。这个任务需要你从维基百科识别句子，可以回答一个问题。

整个任务由三部分组成。第一部分询问给定的问题是否是一个合理的问题，以及您是否期望答案可以在维基百科中找到。第二部分提供了一个来自维基百科的短的段落，并询问是否可以在本段中找到该问题的答案。如果答案是肯定

[1] http://trec.nist.gov/data/qa.html
[2] https://github.com/shuzi/insuranceQA
[3] http://aka.ms/WikiQA.

的，那么它将让你选择本段中哪些句子可以单独回答这个问题。

4.1.2.4 Simple QA

Simple QA[47]，它包括总共108,442个由英语母语标注者以自然语言编写的问题，每个问题都与对应的事实[4] 配对，提供答案且说明完整。所有的事实均从Freebase中抽取的。随机地打乱这些问题的顺序，使用其中70%（75910）作为训练集，10%作为验证集（10845），剩余的20%作为测试集。收集这个数据集采用Freebase知识库，去除掉未定义关系的事实和有太多客体（object）的（主体，关系）对。剩下的事实将会随机采样出一部分给标记者去生成问题。

4.1.2.5 阅读理解数据集

MCTest[48] 是一个阅读理解的数据集，其提供了500个故事，每个故事包含有4个问题，每个问题是一个多项选择（候选答案四个）。要求机器在故事的背景下正确回答相关的四个问题。

Story Cloze Test[49] 是一个评估故事理解、故事生成和脚本学习的新框架。该任务要求系统选择一个正确的结尾曲结束一个四句话的句子。每一个样本包含6个句子，其中前四个是正常顺序，最后两个中有一个是正确答案，需要被系统判别去正确的答案。

4.2 数据探测

这一节我们统计一下基于文档的问答任务NLPCC的训练数据和测试数据的基本统计特性。

4.2.1 基本统计

在训练集中有181882个问答对，有8772个问题，在测试集中有122532个问题对和5779个问题。每个问题在训练集中平均有20个候选答案，在测试集中平均有21个候选答案。

4.2.1.1 问题分类

不同的问题将有不同的信息需求。问题句子的类型通常是一种重要的信号。当区分给定句子是否是正确答案时，这是先决条件但不是充分条件。例如问题“中央大学的首任校长是谁？”，候选答案的一个必要的条件是包含有一个至少一个人物命名实体。我们将问题分为以下几类：

有标记数据的前提下，我们一般会通过一种学习的机制去对问题做分类。在

[4]格式化为（主体，关系，客体），也就是(subject,relationship,object)

我们定义好确定的闭集的问题，然后分类器得到最后的类别。但是由于缺少高质量的标记数据，所以我们采取一个粗略的正则表达式的模板匹配来做问题分类

在对答案能够回答哪一部分问题的角度上看，我们对常见的机构、任务和地点的做一个命名实体抽取的方式来识别，对于数字和时间来说，还是接着使用模板匹配解决这个问题，由于时间中也包含有数字信息，所以同时匹配上此两种类别是以时间计。

4.2.1.2 字词重叠信息

对于像“佛罗伦萨什么时候降水比较多？”这样的问题，唯一的答案“降水主要集中在冬季”有着同样的部分“降水”。大多数中文分词工具，“降水”可以分割为单个单词，这通常被认为是语义单元的最小粒度。除了一些改写的案例，答案将通过将一些词语与问题重叠来涵盖问题。在整个训练测试中，我们得到如图中所示的趋势。很容易发现在x轴的0到13的范围内，问答配对之间的重叠字越多，问答配对的概率越大。此外，字符级重叠的信息将覆盖许多中文的改写模式。例如问题“‘年’字有多少笔？”，正确答案“笔划：6”，对答案进行分词的时候[“笔画”，“：”，“6”]，并且没有与问题重叠的单词。

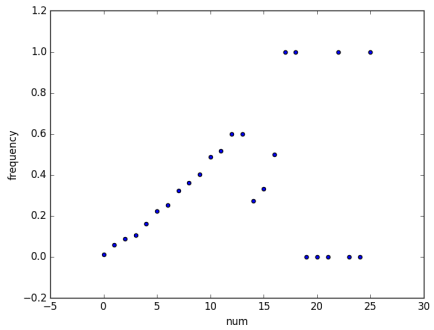

图 4-1 x轴表示问答句子中的重叠单词的数量。y轴是指目标答案的可能出现频率。数据散布在14个之后缺少单词样本。

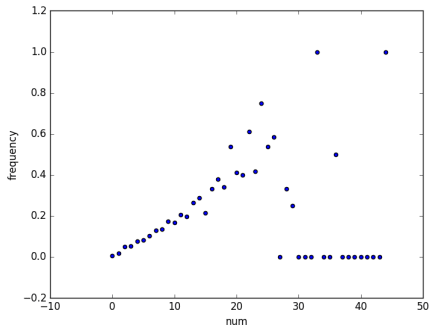

图 4-2 x轴表示问答句子中的重叠字符的数量。而y轴指的是成为目标答案的概率。数据分散在22和50之间的范围内，因为样本不够。

除了字词的重叠信息，我们还基于Embedding的相似度做了以下的统计。

表 4-4 训练集和测试集的基本信息.

	训练集	**测试集**
问答对	181882	122532
问题数量	8772	5779
平均候选答案	20.7	21.2
问题平均字数	46.3	46.2
答案平均字数	106.0	106.5
平均正确答案个数	1.05	1.06

表 4-5 问题种类.

	训练集	测试集
时间	950	861
数字	2135	1433
人物	1049	526
地点	583	396
组织	185	137
其他	3870	2646

表 4-6 问题匹配模板.

类别	匹配串
数字	.*(第)?(几|(多(少(?!年)|大|小|(长(?!时间))|短|高|低|矮|远|近|厚|薄))).*
时间	.*((哪一?(年|月|天|日)|(什么时(候|间))|(多久)|(时间是)|(多长时间)|(多少年))).*
机构	.*(((什么)|(哪(一|两|三|几|些)?(个|家)?)).{0,3}(班底|前身|团|公司|企业|组织|集团|机构|学校|基地|媒体|开发商|商场|购物中心|工作室)|((班底|前身|团|公司|企业|组织|集团|机构|学校|大学|基地|媒体|开发商|商场|购物中心|工作室)(是|叫))).*
人物	.*((什么名字)|((?¡!开发商是)谁)|((哪(一|两|三|几|些)?)(个|位)?(人才?|作家|作者|演员|皇帝|主持人|统治者|角色|主角|名字?|子|儿)(是|((叫|演)什么))))).*
地点	.*(((什么| 哪)(一|两|三|几|些)?(个|座)?(里|地 方|地 区|国 家?|城?市|县|村|州|洲|行 政区))|(在哪(?!((一|两|三|几|些)?(场|次|个|集|批|级|部|播出|网站))))|(是哪(?![\u4e00-\u9fa5]))).*

一般而言问答对的matching应该是和文本的relevance应该有一些差异的，但是通常意义上，越相关其更容易是匹配的问答对，毕竟都是对同一个issue进行阐述的。我们简单地直接计算正确答案和错误答案的所有词语的之间平均的相似度。例如对于问题包含有m个字/词$q_i \in Q.0 < i < m$，答案中包含有n个字/词$a_j \in A, 0 < j < b$，于是我们最后的粗略的相似度为

$$rel_{meanEmbedding}(Q,A) = \frac{1}{|Q||A|} \sum_{0<i<m} \sum_{0<j<n} sim(q_i, a_j) \tag{4-1}$$

得到下表

因为字词级别的Embedding会引入更多的噪音，尤其是字向量，字向量的个数更少，大概常用汉字在3000左右，出现的频率也比较高，所以汉字之间更容易互成上下文，所以字与字之间的相似度都比较大。更容易引入噪音。再者没有考虑权重和序列化信息所以并没有太好的结果。但是词向量确实有着较好的区分正确答案的能力，也是我们后面能够使用词向量的内在动机所在。

4.2.1.3 序列化信息

像TF-IDF或BM25模型的传统IR模型将查询或文档视为一个词袋，其中忽略结构的顺序信息。在具有较短长度的问答的QA系统的场景中，词的顺序信息可以有助于问答对的匹配，并且需要考虑到顺序信息的更复杂的模型。粗略地说，句子的不同位置中的单词可以反映不同的句法和语法结构。对于问题“中央大学的首任校长是谁? 的例子，“前中央大学”一词是前面位置中的“首任校长” 。后面的单词“首任校长”可能与问题更相关，这将更有助于问答匹配。在训练和测试集中，我们发现重叠位置和其对应的问答匹配的概率之间的统计相关性，如图所

表 4-7 答案匹配模板.

类别	匹配串
数字	.*(([]+)([零一二两俩三仨四五六七八九十百千万亿壹贰叁肆伍陆柒捌玖拾佰仟萬]+)).*
时间	.*(([春夏秋冬])(((([\d]+)([零一二两三四五六七八九十百千万亿]+))[年月日天时点刻分秒])))).*

表 4-8 基于Embedding的相关性.

	正确答案	错误答案
字向量skip-gram	0.347891	0.328757
字向量cbow	0.056105	0.034594
词向量	0.198798	0.105795

示的对于词级别的重叠和用于字符级的重叠。

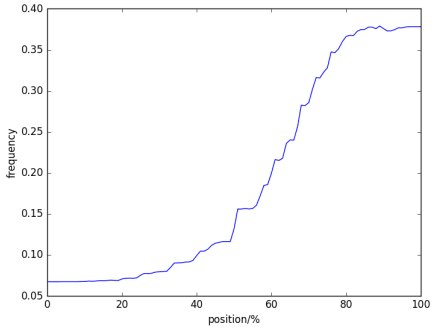

图 4-3 x轴是指问题句子中重叠字符的位置。x = 0表示重叠字符在句子的前面。x=%100 表示重叠字符在句子的后面.y轴表示正确答案的发生频率。

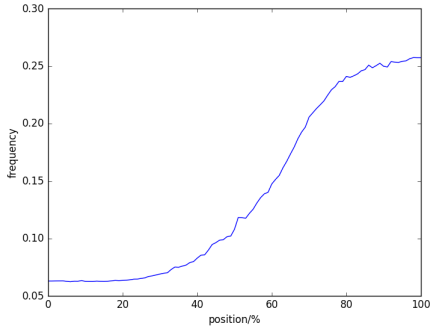

图 4-4 与图 4-3类似, 该图示出了问题句子中的重叠单词的位置与正确答案的出现频率之间的关系.

勿论是中文还是英文，对于问题句子，我么对其各组成成分（词或者是更细粒度的字）的重视程度其实是不一样的，有一些问题的前提词或者本来没有任何语义含义的副词或者是疑问词，并不具有区分出答案的能力，即使答案更多与其语义匹配的内容也没有足够的说服力认为其实有成为正确答案的潜力。树形的句法结构更能反映句子本身，但是基于树形的句法结构的复杂度很高，序列结构一定程度被发挥了很多潜力。

4.3 数据预处理

由于缺乏中文文本的明显界限，我们使用pynlpir包[5] [50]来分割问句语句和答案文本。移除用于删除没有歧义且具有很少语义含义的无用高频词的Stopwords [6]。为了得到答案的分类信息，我们采用LTP在线API [7]的命名实体识别技术

[5]https://github.com/tsroten/pynlpir
[6]停用词http://tcci.ccf.org.cn/conference/2016/pages/page05_evadata.html
[7]http://www.ltp-cloud.com/

（NER），来识别其中可能存在的命名实体。

NLPCC官方提供了一个300维度的词向量。另外我们还自己补充了两组词向量，第一组词向量由包含260000个实体的维基百科页面构成（原始页面都经过繁转简的操作）。第二组词向量来自于我们自己爬的百科页面[8]，与比赛中问题相关的所有实体会被我们爬取，该语料集合更小，但是与我们的问题领域更相关。同时我们使用了CBow和skip-gram两种训练方式。训练的代码来自word2vec的python版本gensim项目[9]。

4.4 特征工程

4.4.1 问题种类和答案分类

在我们的论文中，问题分为5类。其中3个涉及名称实体，它们是人物，地点和机构。其余两个是时间和数字。由于缺乏大规模的标签数据，我们不能采用基于学习的问题分类器[51]，基于模板的问题分类器可以覆盖简化分类法的大多数情况。

答案的分类可能是一个多标签任务，这意味着答案可以属于许多类别。前三个基于NER的类别可以被LTP在线API[10]识别。同时，我们使用正则表达式的模板来区分数字和时间的类型。在实践中，采用两种方法。第一个是dummp Number，5个类别被视为5维二进制特征向量，其默认值是不匹配。如果问题被识别为一种类别，则相应的维度标记为匹配。第二种方法是采用一维特征，反映了问答类型的匹配与否。

4.4.2 字词重叠

在 Sec. 4.2, 在匹配概率和重叠信息之间存在统计相关性对于中文文本，我们很难找到可以包含所有近同义词对的字典。另一种方法是使用基于字符的度量，因为许多同义改写模式对共享相同的中文字。我们计算词汇级别和字符级别的重叠分数如下：

$$Score_{overlap}(Q,A) = \sum_{q_i \in Q}^{n} freq_A(q_i) \cdot weight(q_i) \tag{4-2}$$

其中问题语句Q具有n字（字符），答案语句A具有m字（字符）。加权模型基于句子中q_i的位置和整个文本集合的q_j的IDF（反向文档频率）。$freq_A(q_i)$ 表示为q_i在回答A中的平滑之后的频率。

[8]http://baike.baidu.com/
[9]http://radimrehurek.com/gensim/models/word2vec.html
[10]http://www.ltp-cloud.com/

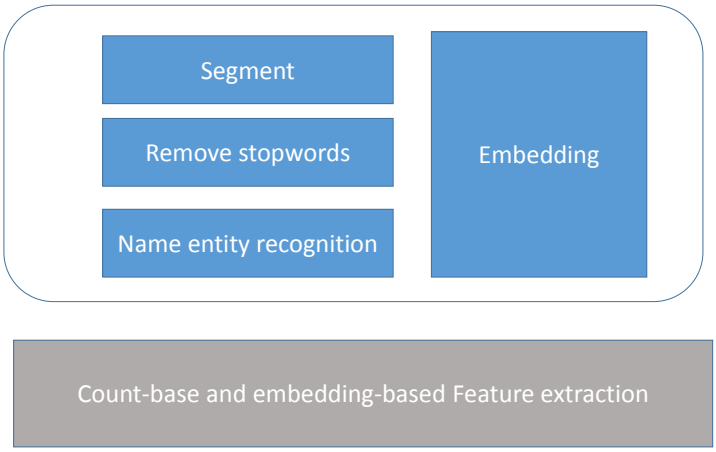

图 4-5 预处理框架

4.4.3 BM25

传统的BM25模型被实现如下

$$Score_{bm25}(Q,A)=\sum_{q_i\in Q}^{n} IDF(q_i)\cdot\frac{freq_A(q_i)\cdot(k+1)}{freq_A(q_i)+k\cdot(1-b+b\cdot\frac{Length_A}{Lenght_{avg}})} \tag{4-3}$$

其中$freq_A(q_i)$是q_i在答案A中的频率。k和b是特定任务的可调参数。$Length_A$和$Lenght_{avg}$分别是答案的长度A和整个答案的平均长度。

4.4.4 加权词向量

嵌入技术将单词嵌入到统一的语义空间中，这使得可以找到单词之间的关系。句子被认为是线性地被词语叠加。不同的词可以对句子的整体含义贡献不同的权重，这取决于他们的位置，语义结构和IDF。我们得到一个单词或一个汉字的表示如下：

$$Representation(S)=\frac{\sum_{i=0}^{n} weight(s_i)\cdot\overrightarrow{embedding(s_i)}}{\sum_{i=0}^{n} weight(s_i)} \tag{4-4}$$

s_i是句子中的字符或单词（问题或答案），$embedding(s_i)$是相应的嵌入向量。然后，我们计算问题和答案表示之间的内积作为最终得分。

4.4.5 神经网络

除了词嵌入的加权组合，我们构建了一个神经网络，如图 4-6所示。.

在我们的方法中，已经采用了词级嵌入和字符级嵌入来形成问题和答案的句子矩阵。可训练矩阵U用于桥接问题嵌入矩阵和答案嵌入矩阵。以下$tanh$函数可以避免以前激活的值的爆炸。通过row-pooling和col-pooling而不是max-

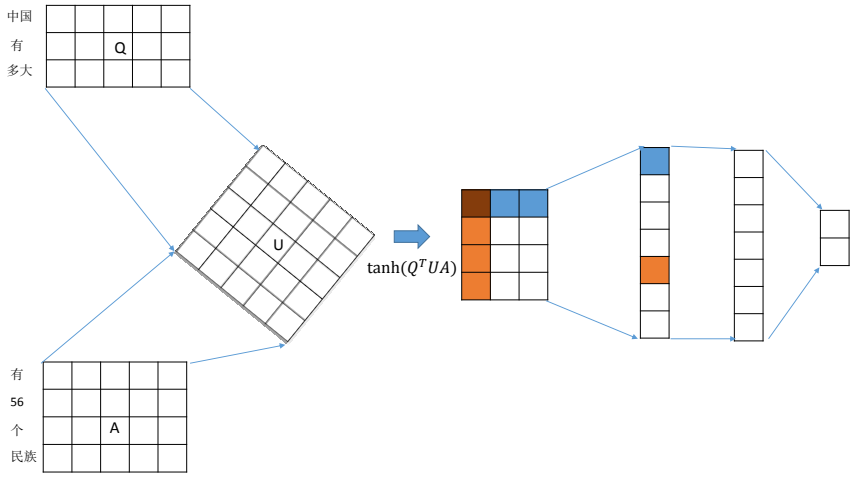

图 4-6　　神经网络结构

pooling[43]的操作，有序位置的信息仍然可以保留在完全连接层中。在softmax层之后，最后一个输出层包含两个浮点数，分别表示问答匹配和不匹配的概率。交叉熵损失函数用于优化过程。

4.4.5.1　Embedding层

问题和答案都是一个字/词串，通过查找Embedding表$(x) \in R^{|V|\times d}$，得到每一个字/词的向量，分别拼接成一个Embedding矩阵(依然按照原来字/词的顺序拼接)。

4.4.5.2　交互层

根据第2.3.2章所述，一维match模型由于孤立去建模单个表示，一组文本对之间组合他们的表示之前完全没有任何交互，实际上是与我们直觉上的理解是相悖的。换句说，我们要更好去做一个文本对的match，就希望在我们建模它们的表示的时候就能够考虑到对方的影响，例如问答中，问题的信息可以指导答案表示的建立。二维match模型的关键就是这样一个交互层，能够把两个embedding矩阵直接通过一个U矩阵联系起来，类似更多的联系两个矩阵的方式如第2.3.2所示，矩阵U是一个更加通用的形式。

$$M = tanh(Q^T UA) \tag{4-5}$$

4.4.5.3　Pooling层

我们得到问答对的Embedding交互矩阵之后，简单地看，最后M矩阵的长宽分别是问题文本和答案文本的长度，预示着问答对各个部件之间的matching关

系。pooling操作要做的就是把这样一个不定长的match矩阵M映射到一个相同的特征空间，在这样的一个特征空间中维度的总数是一致的，而且每一个维度在不同文档对的含义是一致的。这样一种区别于卷积网络的pooling操作应运而生。就是对该矩阵的行和列分别做max-pooling，我们截断掉异常长的特征，对于的短的特征向量将其以0补齐到一致的长度。这样的一种做法也是为了保留次序信息，有研究表明词序能够贡献20%的句子语义[52]，甚至更多。

4.4.5.4 全连接层和softmax层

Pooling之后全连接层需要作进一步的特征提取，最后特征将会被softmax归一化到两个概率值，分别表示问答对匹配的概率。当前预测的概率值和最终的标签的交叉熵是我们模型的损失函数。损失通过反向传播更新网络的参数。

4.4.6 其他特征

编辑距离通常用于测量文本字符串的相似性。Jaccard指数给出了问题和答案之间的语素集的相似性。回答句的长度通常被认为是一个重要特征。

4.5 模型融合

基于传统模型的得分，以及基于Embedding模型的得分，都直接影响问答的匹配程度。在对上述特征进行一个Z-socre 的归一化之后，我们考虑最直观且较快的线性回归得到一个最终分数。除了线性模型，我们还利用学习排序模型 [11]和基于树的集成回归模型[53]，来做最后的预测。

4.6 模型评价

我们的问答系统将通过MRP和MAP进行评估。MRR主要根据正确答案的等级表示回报结果的质量，即正确答案排名越高，结果越好。

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{4-6}$$

$|Q|$ 代表测试集合的问题的个数. Q_i是其第i个问题，$rank_i$ 是正确答案在候选集C_i中的位置. 如果正确答案不在候选集中，$\frac{1}{rank_i} = 0$ 。

[11]https://sourceforge.net/p/lemur/wiki/RankLib/

另一个评估度量是平均精度（MAP），其可以如下定义

$$MAP = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{\sum_{k=1}^{n}(P(k)\cdot rel(k))}{min(m,n)} \tag{4-7}$$

MAP主要是指结果的平均精度。如果正确的答案在检索的答案集中排列得很高，则MAP的值也将相应地高。m表示正确答案句子的数量，n是检索的答复句子的数量。$P(k)$是截止值的精度k，这是检索的回答句子的顺序中的排名。$rel(k)$如果在等级k的项目是回答句子，则等于1，如果不是，则等于0。

4.7 实验结果

表 4-9 结果比较.

method	**MAP**	**MRR**
Average Word Embedding	0.4610	0.4610
Machine Translation	0.2410	0.2412
Paraphrase	0.4886	0.4906
Word Overlap	0.5114	0.5134
计数特征	0.7938	0.7944
Embedding特征	0.7467	0.7470
所有特征	0.8005	0.8008

由于上述baseline是基于词袋模型并且没有基于学习的机制的事实，所以性能相当差。在最终评估中，我们的方法获得的MRR为0.8008，在18个提交中排名第5（15个团队中的第四个）。

在我们的方法中，一些模型的最终分数被视为整体方法的特征。如在第4.2章中所提到的，可以有效地对语法和语义信息建模的特征可能更可能与QA数据的匹配标签相关。在中文语法中，例如，与问题问题相关的关键词通常出现在问句的后面位置，而前面位置的词与无区别的前提更相关，后者被饱和大多数候选答案。此外，还需要有效的语义匹配策略。我们在我们的方法中采用字符级和词级模型，而深层神经网络可能会帮助很多，而我们的网络有点浅而紧凑。

在我们的实验中，像BM-25模型没有潜在的语义匹配，而基于字符的模型胜过中文的字字符。具有端到端策略的位置感知深层神经网络可能是QA任务的趋势。由于低维特征空间，与学习排序方法或基于树的增强方法相比，线性回归已经实现了相当好的性能。LambdaMART的排序方法为什么没有最好的结果，一部分是归因于问答系统更加关注的是top1文档的效果，LambdaMART的优化还是坚持去优化整个排序列表的所有文档，有悖于其训练过程中优化目标一致的动

机。而普通的XgBoost的方法在优化过程中也并没有取得好的结果，很多原因来自于我们的特征的总数不是很多，大概是10个左右，而且有两三个特征的效果极其显著，其他的特征基本很难影响到最终的效果，这也是我们复杂的Xgboost的方法并不比普通的线性回归的方法强。

整个实验也正说明，我们整体的特征提取方法，一部分来自于研究者前期对文本数据的一些经验和做法。一部分来自于我们神经网络直接对文本数据建模，完全数据驱动地学习到原始数据本身和最后标签的关系，最后我们通过学习机制把两种方法得到的所有的特征融合到一起。我们最后的效果还有很多优化的空间。一部分来自我们传统的特征并没有非常好地挖掘数据本身的规律，还有一些结合外部数据的潜力。另一方面来自我们的神经网络模型不是很成功，没有很好运用好神经网络的强大学习能力。

第5章 结论和思考

5.1 总结

在本文中，我们报告我们的方法为NLPCC 2016共享任务开放领域问答子任务的技术细节。已经提出了一些传统方法和基于神经网络的方法。在我们的方法中，我们将中文文本的特征与我们的模型相结合，并通过集合学习策略实现良好的性能。由于神经网络的浅结构，我们的最终性能不是那么好。在我们的观点中，包含短文本的顺序（或基于树的）信息和相应的有效语义匹配的有效重写是QA系统的两个关键因素。可以直接建模顺序文本的RNN网络和更灵活的CNN网络具有在文本数据中的一些适应之后获得更好性能的潜力。此外，尽管中文和英文文本之间有许多共同的特征，但是专门适用于中文的端到端系统也可以是中文问答系统的趋势。

5.2 展望

用深度学习解决问答的探索历史时间并不长，经过最近的探索，应该说很多模型相对最初的技术方案来说，在性能提升方面进展明显，在很多数据集上性能都有大幅度的提高。但是总体而言，这距离让机器像人一样能够理解文本并回答问题还有非常遥远的距离。最后我们对这个研究领域目前面临的问题进行简述并对一些发展趋势进行展望。

随着近期的探索，应该说很多模型相对于最初的技术方案而言，性能有着显着的提高，在许多数据集中都大大提高了性能。但是我们离人类真正去理解自然语言的含义，并流畅地回答问题的的距离还有好远

- 需要对大规模数据集构建更加复杂完备的语料：神经网络由于其包含有大量的参数，需要喂大量的数据进去，才能充分学习好其中的规律。但是对人工构造的数据集的成本太大，难以有非常大数据集供神经网络训练。第一个我们可以尝试去构建一下极限学习的方式，就是喂进不太大数据就可以做到够好的效果，这方面在自然语言处理方面还没有足够出色的工作出来。第二方面我们尝试去尽量多地收集高质量的数据（即使启发式地生成的），只有数据量多起来，我们才有这种可能性在这方面做出突破，毕竟数据是我们做学习任务的命脉。

- 神经网络模型结构的探索：目前对问答任务的理解，神经网络结构的解决方案还有很多潜力，需要探索各种新的网络结构和新的模型，促进快速发展的研究领域。例如考虑到Attention的二维匹配模型，甚至其他考虑到更复杂句子结构的树形结构匹配模型，甚至在里面结合上推理机制和外部知识。
- 引入外部知识：人能够理解文本的重要原因是，人相比于机器，有着更加丰富的上下文建模能力，很多conmmon sense一直作为人类阅读文章的上下文。我们在阅读过程肯定会借助记忆机制中的诸多联想推理，再去构建当前语言的一个理解，甚至于去发散到更加广泛的语义空间中去，只要我们能够在机器中能够缓存或者记忆一些通用的知识和跟当前领域特化的知识，才能更好地去理解当前文本。
- 开发更复杂的推理机制：我们在基于文档的问答系统里，其实并没有任何推理的机制，还是一种简单的数据依赖地做一个内容和上下文不太敏感的匹配任务，更谈不上任何复杂的推理，因为推理是人类赖以强大的重要能力，特别是有着一个复杂大规模的通用常识作为一个背景时。因为当前的主题范围内并不是包含了整个人们处理信息的所有范围，借助一些更加长期记忆的复杂上下文，人们容易联想或者推理出更加丰富的含义。而这也是我们希望机器能够借助外部知识进行推理的一个直观上的理由。
- 端对端策略：端对端策略现在已经成为一个事实上大家都接受的一种处理问题的方式，把所有特征提取、特征选择、模型选择和模型融合放到同一个神经网络中去。但是神经网络如何结合本来已经有很容易捕捉的文本信息是一个很大的难点。比如，如何显性地去找到一个较好策略让网络去能够对问题类别比较敏感，让一个问“是谁？”的问题能够对答案中的所有的人物命名实体有着更好的attention。

5.3 思考

随着大数据场景的自然语言处理业务的需求，越来越多的研究工作倾心于从对话系统来入手，探究机器处理语言和信息理解的能力。一个经典的思路的是从计算机出发，计算机擅长处理什么，我们就发挥其优势去处理文本数据。第二种思路是我们先探究人类是怎么处理信息的，然后将这个机制移植到计算机的处理方法中。越来越多认知和生理方面的灵感指导了计算模型的设计。最近的研究发现量子认知有可能很好的解释人类的信息处理方式，我们在问答和对话任务中，是不是也有很多量子的机制在其中[54]。

参考文献

[1] Lecun Y, Bengio Y, Hinton G. Deep learning. [J]. Nature, 2015, 521 (7553): 436–44.

[2] Hinton G, Deng L, Yu D, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups [J]. IEEE Signal Processing Magazine, 2012, 29 (6): 82–97.

[3] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks [C]. In Advances in neural information processing systems, 2012: 1097–1105.

[4] Harris Z S. Distributional Structure [J], 1954, 10 (2-3): 146–162.

[5] Bengio Y, Schwenk H, Senécal J S, et al. Neural Probabilistic Language Models [J]. Journal of Machine Learning Research, 2001, 3 (6): 1137–1155.

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

[7] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives [J]. IEEE transactions on pattern analysis and machine intelligence, 2013, 35 (8): 1798–1828.

[8] Zhai C, Lafferty J. A study of smoothing methods for language models applied to Ad Hoc information retrieval [C]. In International ACM SIGIR Conference on Research and Development in Information Retrieval, 2004: 179–214.

[9] Deerwester S, Dumais S T, Furnas G W, et al. Indexing by latent semantic analysis [J]. Journal of the American society for information science, 1990, 41 (6): 391.

[10] Hofmann T. Probabilistic latent semantic indexing [C]. In International ACM SIGIR Conference on Research and Development in Information Retrieval, 1999: 56–73.

[11] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation [J]. Journal of Machine Learning Research, 2003, 3: 993–1022.

[12] Collobert, Ronan, Weston, et al. A Unified Architecture for Natural Language Processing [J]. Journal of Parallel and Distributed Computing, 2008.

[13] Mnih A, Hinton G. Three new graphical models for statistical language modelling [C]. In International Conference on Machine Learning, 2007: 641–648.

[14] Le Q V, Mikolov T. Distributed Representations of Sentences and Documents. [C]. In ICML, 2014: 1188–1196.

[15] Collobert R, Weston J, Bottou L, et al. Natural Language Processing (Almost) from Scratch [J]. Journal of Machine Learning Research, 2011, 12 (1): 2493–2537.

[16] Kim Y. Convolutional Neural Networks for Sentence Classification [J]. Eprint Arxiv, 2014.

[17] Mnih V, Heess N, Graves A, et al. Recurrent models of visual attention [C]. In Advances in Neural Information Processing Systems, 2014: 2204–2212.

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

[19] Luong M-T, Pham H, Manning C D. Effective approaches to attention-based neural machine translation [J]. arXiv preprint arXiv:1508.04025, 2015.

[20] Yin W, Schütze H, Xiang B, et al. Abcnn: Attention-based convolutional neural network for modeling sentence pairs [J]. arXiv preprint arXiv:1512.05193, 2015.

[21] Pennington J, Socher R, Manning C. Glove: Global Vectors for Word Representation [C]. In Conference on Empirical Methods in Natural Language Processing, 2014.

[22] Levy O, Goldberg Y. Neural word embedding as implicit matrix factorization [J]. Advances in Neural Information Processing Systems, 2014, 3: 2177–2185.

[23] Li Y, Xu L, Tian F, et al. Word embedding revisited: a new representation learning and explicit matrix factorization perspective [C]. In International Conference on Artificial Intelligence, 2015.

[24] Baroni M, Dinu G, Kruszewski G. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors [C]. In Meeting of the Association for Computational Linguistics, 2014: 238–247.

[25] Choi H, Cho K, Bengio Y. Context-Dependent Word Representation for Neural Machine Translation [J]. arXiv preprint arXiv:1607.00578, 2016.

[26] Jin P, Zhang Y, Chen X, et al. Bag-of-Embeddings for Text Classification [J], 2016.

[27] Sun Y, Lin L, Yang N, et al. Radical-Enhanced Chinese Character Embedding [M]. Springer International Publishing, 2014.

[28] Chen X, Xu L, Liu Z, et al. Joint learning of character and word embeddings [C]. In International Conference on Artificial Intelligence, 2015.

[29] 周志华. 线性模型 [M]. 清华大学出版社, 2016.

[30] Kearns M J. The computational complexity of machine learning [M]. The MIT Press, 1990.

[31] Valiant L G. A theory of the learnable [J]. Communications of the Acm, 1984, 27 (11): 1134–1142.

[32] Friedman J H. Greedy Function Approximation: A Gradient Boosting Machine [J]. Annals of Statistics, 2001, 29 (5): 1189–1232.

[33] Burges, Chris, Shaked, et al. Learning to rank using gradient descent [J]. Inproceedings, 2005: 89–96.

[34] Yao X, Durme B V, Callison-Burch C, et al. Answer Extraction as Sequence Tagging with Tree Edit Distance [C]. In Conference of the North American Chapter of the Association for Computational Linguistics, 2013.

[35] Wang Z, Ittycheriah A. FAQ-based Question Answering via Word Alignment [J], 2015.

[36] Yih W T, Chang M W, Meek C, et al. Question Answering Using Enhanced Lexical Semantic Models [C]. In Meeting of the Association for Computational Linguistics, 2013: 1744–1753.

[37] Zhou G, Cai L, Zhao J, et al. Phrase-Based Translation Model for Question Retrieval in Community Question Answer Archives [C]. In The Meeting of the Association for Computational Linguistics, 2011: 653–662.

[38] Severyn A. Automatic Feature Engineering for Answer Selection and Extraction [C]. In EMNLP, 2013.

[39] Yu L, Hermann K M, Blunsom P, et al. Deep learning for answer sentence selection [J]. arXiv preprint arXiv:1412.1632, 2014.

[40] Severyn A, Moschitti A. Learning to rank short text pairs with convolutional deep neural networks [C]. In SIGIR, 2015: 373–382.

[41] Wang D, Nyberg E. A Long Short-Term Memory Model for Answer Sentence Selection in Question Answering [C]. In Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing, 2015.

[42] Tan M, Xiang B, Zhou B. LSTM-based Deep Learning Models for non-factoid answer selection [J]. arXiv preprint arXiv:1511.04108, 2015.

[43] Santos C D, Tan M, Xiang B, et al. Attentive Pooling Networks [J], 2016.

[44] Li H, Lu z. Deep Learning for Information Retrieval [J]. SIGIR 2016 Tutorial, 2016.

[45] Feng M, Xiang B, Glass M R, et al. Applying deep learning to answer selection: A study and an open task [C]. In IEEE Workshop on Automatic Speech Recognition and Understanding, 2015.

[46] Yang Y, Yih W-t, Meek C. Wikiqa: A challenge dataset for open-domain question answering [C]. In Proceedings of EMNLP, 2015: 2013–2018.

[47] Bordes A, Usunier N, Chopra S, et al. Large-scale simple question answering with memory networks [J]. arXiv preprint arXiv:1506.02075, 2015.

[48] Richardson M, Burges C J, Renshaw E. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. [C]. In EMNLP, 2013: 4.

[49] Mostafazadeh N, Chambers N, He X, et al. A corpus and cloze evaluation for deeper understanding of commonsense stories [J]. Proceedings of NAACL HLT, San Diego, California, June. Association for Computational Linguistics, 2016.

[50] Liu T, Che W, Zhenghua L I. Language Technology Platform [C]. In COLING 2010, 2010: 13–16.

[51] Li X, Roth D. Learning Question Classifiers [J]. Coling, 2003, 12 (24): 556–562.

[52] Landauer T K. On the computational basis of learning and cognition: Arguments from LSA [J]. Psychology of Learning and Motivation, 2002, 41 (41): 43–84.

[53] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System [J], 2016.

[54] Wang B, Zhang P, Li J, et al. Exploration of Quantum Interference in Document Relevance Judgement Discrepancy [J]. Entropy, 2016, 18 (4): 144.

[55] Papadimitriou C H, Tamaki H, Raghavan P, et al. Latent semantic indexing: a probabilistic analysis [C]. In Seventeenth ACM Sigact-Sigmod-Sigart Symposium on Principles of Database Systems, 1998: 217–235.

[56] Zhai C, Lafferty J. A study of smoothing methods for language models applied to information retrieval [J]. Acm Transactions on Information Systems, 2001, 22 (2): 179–214.

[57] Burges C J. From ranknet to lambdarank to lambdamart: An overview [J]. Learning, 2010, 11: 23–581.

[58] Wang M, Smith N A, Mitamura T. What is the Jeopardy Model? A Quasi-Synchronous Grammar for QA. [C]. In EMNLP-CoNLL, 2007: 22–32.

[59] Heilman M, Smith N A. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions [C]. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010: 1011–1019.

发表论文和参加科研情况说明

（一）发表的学术论文

[1] Wang B, Niu J, Ma L, et al. A Chinese Question Answering Approach Integrating Count-based and Embedding-based Features[C]. NLPCC 2016 accepted.

[2] Wang B, Zhang P, Li J, et al. Exploration of Quantum Interference in Document Relevance Judgement Discrepancy[J]. Entropy, 2016, 18(4): 144.

[3] Zhang P, Li J, Wang B, et al. A Quantum Query Expansion Approach for Session Search[J]. Entropy, 2016, 18(4): 146.

[4] Chen Y, Zhang P, Song D, Wang B. A Real-Time Eye Tracking Based Query Expansion Approach via Latent Topic Modeling[C]. CIKM 2015. ACM, 2015: 1719-1722.

致　谢

本论文工作在我的导师宋大为老师和张鹏老师的悉心指导下完成的，首先给予他们最大的感谢。一来感谢其科研治学之精神，二来对学生学习科研生涯之拳拳关爱。

宋老师有着最好的国际视野，邀请国际一流的学者，比如Chang Yi老师来实验室交流，同时提供最好的平台让我们出去交流学习。经常需要国内国外一直奔波，不倒时差就开始工作，有时候我们已经离开了实验室回去休息，宋老师仍然在实验室耕耘，负担着整个实验室的兴衰。同时科研指导的只言片语，可以让我们推敲很久。

张老师们寒暑相守，尤其是在腊月二十多，陪着学生们一起熬夜写论文。对着审稿意见句句予以揣摩推敲，不曾放过一个细节。同时对很多培训学习机会，不吝予以支持。甚至外地出差看到相关的insight，第一时间电话通知，回来第一时间到实验室就喊着一起讨论交流，此番科研的热情和执着却是比科研内容指导本身更加让人受益终生。

除开宋老师和张老师，竞飞师兄、晓朝师兄和永强师兄也给我相当多的科研指导和精神鼓励。竞飞师兄陪着我完成了第一篇论文的投稿工作，兢兢业业，不吝批评，亦不吝鼓励，从的工作上无微不至的指导，到最后一直带着我各地跑马拉松，言传身教地教我懂得去敬畏“生命”和“生活”，去为团队奉献；晓朝师兄聪慧谦虚，科研上有一股韧劲儿，同时在科研工作和生活上不骗自己，坚守自己的阵地，同时对家庭的责任和对团队的付出是我们整个实验室的财富。永强师兄也带着我投了第一篇顶会的论文，第一次在学术论文上能够印上我的名字，我知道顶会的论文有多远，一流的研究工作，我们并不是遥不可及，缺的是坚持和“敬畏”。

除了师兄们，我谨以时间轴为顺序，感谢我们一同毕业和进步的小伙伴们。一起互相鼓励参加复试并一同上课的李博古同学；王亚如同学本身的坚持和努力，以及对我非常多的鼓励和支持；陈云老大哥给我们生活上支持和精神上的鼓励；李鹏晓龙同学一直的关怀和关照；一起享受生活挥霍生命，并互相鼓励进步的李朝李健同学；在宿舍一起交流很多科研生活和恋爱想法的于广亮同学；在一篇论文工作中积极找人帮我做实验的李健铨和商振国同学；闺蜜团杨小丽、樊双

琳和韩金晶同学；陪我论文写到深夜的徐奇同学。以及实验室的诸多师弟师妹，给我提供了很多解决问题的原始题材，以及给我的诸多鼓励和信心，他们分别是牛嘉斌、王天舒、宋玲玲、刘小康、张胜男等，以及我们新来的张于华、苏展、马立群等同学，给我们的生活输入了新鲜的血液。

另外感谢我的父母，他们含辛茹苦地把我培养成了一个能够完成硕士学位的人，在他们的理解和支持下，我能够心无旁骛地专心去做科研本身，能够在正常生活之余排除其他的干扰。同时感谢我们家的小朋友胡宁轩，舅舅这么努力也是为了以后可以言传身教地让你去做一个幸福有担当的人。

本篇论文的原始工作来自NLPCC会议的一篇Poster，该论文的实验部分马立群牛嘉斌同学付出了巨大的努力，论文撰写部分感谢张于华同学。同时在本毕业论文撰写过程中感谢王晓静同学的陪伴。