电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

硕 士 学 位 论 文

MASTER THESIS

论文题目 在线问答社区推荐算法研究

学 科 专 业 软件工程

学 号 **201521220110**

作 者 姓 名 薛 浩

指 导 教 师 刘梦娟 副教授

分类号 _____ 密级 _____

UDC [注1] _____

学 位 论 文

在线问答社区推荐算法研究

（题名和副题名）

薛浩

（作者姓名）

指导教师 刘梦娟 副教授

电子科技大学 成都

（姓名、职称、单位名称）

申请学位级别 硕士 学科专业 软件工程

提交论文日期 2018. 03 论文答辩日期 2018.06.01

学位授予单位和日期 电子科技大学 2018 年 06 月

答辩委员会主席 _____

评阅人 _____

注 1：注明《国际十进分类法 UDC》的类号。

Research on Recommendation Algorithms in Online Community Question Answering

A Master Thesis Submitted to

University of Electronic Science and Technology of China

Discipline: **Software Engineering**

Author: **Hao Xue**

Supervisor: **Mengjuan Liu**

School: **School of Information and Software Engineering**

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及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它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谢意。

作者签名：薛浩　　日期：2018年 6月 5日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技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，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名：薛浩　　导师签名：刘楼娟

日期：2018年 6月 5日

摘 要

问答社区已经逐渐发展成为人们分享并获取知识和信息的平台，每天都有大量的新问题被用户提出来，等待其余用户回答和讨论。但是随着社区的发展积累了海量的问题、答案及用户数据，问答社区开始面临“信息过载问题”。一方面是用户难以快速找到自己感兴趣的相关问题，另一方面是很多新问题被埋没在海量的数据里，无法及时获得高质量的答案，同时新问题缺少能够准确描述问题信息的话题标签，导致很难被其余用户检索发现。

本文针对问答社区面临的数据挑战难题，研究并设计解决标签推荐和专家用户推荐的算法模型。论文的研究工作主要分为两个部分。

第一部分提出了基于深度学习的标签推荐算法。根据问题的多标签属性，首先将标签推荐定义为一个多标签文本分类问题，然后结合双向长短期记忆网络和卷积神经网络提取问题文本的语义特征信息，并在训练数据集上进行有监督的多标签分类训练。为了提升算法模型的性能，本文在双向长短期记忆网络中引入了基于传统注意力机制的单词注意力机制和句子注意力机制。

论文的第二个研究工作是针对问答社区新问题的专家用户推荐，本文将专家用户推荐定义为一个对级排序学习问题，即对于每一个问题，按照答案的质量优劣，构造每两个回答者之间的相对偏序关系作为训练样本进行模型学习。为了缓解用户行为稀疏性以及为了增强用户和问题的匹配质量，算法构造一个基于用户回答问题的行为和用户社区关注关系的异构图，通过在异构图中进行随机游走发现更多的用户-问题潜在关系。在进行排序学习训练时，需要以数学形式表示问题和用户并计算问题和用户的相关性，本文使用双向长短期记忆网络进行问题文本的表示学习，同时学习一个用户嵌入矩阵表示用户。为了增强神经网络的表示学习能力，针对问答社区问题的多话题属性特点，提出一个多话题注意力机制。

本文在知乎的真实问答数据集上验证了提出的两个算法模型的性能，实验结果表明本文提出的算法模型优于传统的标签推荐和专家用户推荐算法，其中标签推荐算法的 F1-Score 指标相较于传统的基于内容的方法提升了 30%，比基于单个深度学习模型的算法提升了 10%。专家用户推荐算法在 NDCG 和 MRR 这两个指标上比传统算法提升了 10%左右，在 F1-Score 指标上提升超过了 3%。

关键词：推荐系统，问答社区，深度学习，专家推荐，标签推荐

ABSTRACT

The community question answering community has gradually developed into a platform for people to share and acquire knowledge and information. Every day, a large number of new questions are asked by users, waiting for the answers and discussions of the other users. However, as the development of the community has accumulated a huge amount of problem, answer and user data. CQA websites has begun to face the problem of "information overload". On the one hand, users are difficult to quickly find problems they are interested in. On the other hand, a lot of new problems have been hidden in huge amounts of data, people can't get high quality answers in time. Moreover, many new problems lack of topic labels which can accurately describe the problem, as a result, they are hard to be retrieved by the rest of the users.

This thesis aims to solve the problem of data challenges faced by CQA websites, and proposes the algorithm model to solve tag recommendation and expert user recommendation. The research work of this thesis include two parts.

The first part puts forward the label recommendation algorithm based on deep learning. According to the multi-labels of problem, firstly define tag recommendation as a multi-label text classification, and then combine bi-directional long short-term memory network (Bi-LSTM) and convolutional neural network (CNN) to extract text semantic characteristics of the information, and train model in a supervised way from the training data. In order to improve the performance of the model, this thesis introduced the word attention and sentence attention mechanism based on the traditional attention mechanism in the bidirectional long short-term memory network.

The second work of this thesis is propose an algorithm of expert user recommendation for new questions in CQA websites, this article defines expert users recommend as a pairwise learning problems, that means for each problem, according to the quality of the answers, create the partial order for every two answers to train model. In order to alleviate the sparse of user behavior and to enhance the quality of user-problem matching, this thesis also construct a heterogeneous graph based on user behaviors of answering the question and the user community social relationship, and then find more user-problem matching relationship through random walk in heterogeneous graph. This part apply Bi-LSTM learn represenation of problem, at same time learn a user embedding

matrix to learn to rank. In order to enhance the expression learning ability of neural network, a multi-topic attention mechanism is proposed for the multi-topic attribute of CQA websites.

Experimental results on zhihu datasets demonstate proposed algorithms outperform previous work of label recommendation and expert user recommendation, F1 – Score of the label recommendation outperform the traditional based on the content method outperform by 30% and 10% than single deep learning model. The expert user recommendation algorithm improved by about 10% on NDCG and MRR, and increased by more than 3% on the F1-score index than previous works.

Keywords: Recommendation System，Community Question Answering，Deep Learning，Expert Recommendation，Label Recommendation

目 录

第一章 绪论

1.1 研究背景与意义

在线问答社区诞生之前，搜索引擎是人们获取信息的主要路径。搜索引擎的出现在一定程度上缓解了人们面临的信息过载问题，极大的加快了信息检索和传播速度。用户通过搜索引擎获取信息是一种主动的方式，即用户产生了需求，然后借助搜索引擎进行检索，但是随着网络技术的发展，互联网上的数据呈爆炸式的增长，这种主动获取信息的方式已经越来越难以满足用户日益增长的信息需求：1）当用户无法准确描述查询意图时，搜索引擎可能返回与用户需求不相关的搜索结果；2）由于互联网保存着海量数据，即使用户能够提供准确代表其意图的关键字，搜索引擎也可能返回大量与用户意图不相关的检索结果，用户还是面临着二次筛选的问题。由于搜索引擎技术发展面临的难题越来突出，推荐系统的研究逐渐引起许多研究人员的注意，利用推荐系统解决信息过滤及推送任务相对于用户来说是一种被动地获取信息的方式，目的是解决用户和信息之间匹配问题，即通过分析用户的历史行为兴趣，将用户可能感兴趣的信息主动地呈现在用户面前，这种用户被动获取信息的方式有效地解决了搜索引擎所面临的两个突出问题，但是随之而来也有新的难题需要去解决，其中最为核心的就是解决用户和信息之间的匹配问题，为此研究者花费了大量的时间和精力开发了各种推荐算法。推荐系统研究发展到如今，已经产生了许多成熟的算法，比如协同过滤推荐算法，基于内容的推荐算法、图模型、混合推荐算法等等。

随着 web2.0 的发展，用户逐渐成为网络内容信息的生产者，随之而来的是在线知识问答社区（Community Question Aswering，CQA）也进入到了大众的视野，传统的知识问答平台有 Yahoo! Answer、新浪问问、百度知道等；新型的知识问答，或者称之为知识共享平台，包括 quora，stackoverflow，知乎等。知识问答社区的基本形式是用户在平台上提问，然后社区其他用户根据自己的知识、经验进行解答，用户之间可以就问题进行观点讨论，这种形式的信息交互具有很多优点：1）社区用户可以提出各种类型的问题，并且可以从其余用户那里获得答案，整个过程所有用户都可以参与；2）社区的信息以文本的形式表示，比如提问、回答、评论都是以自然语言为载体；3）作为知识共享平台，用户之间贡献知识、经验，能够有效的促进社区的良性发展，增加用户粘性；4）建立答案评价体系，允许用户对问题的答案进行评价，比如点赞，反对等；5）随着社区逐渐的发展壮大，以及更多领

域专家的加入，用户可以更加高效地获得更准确的答案。

知识问答社区作为知识共享的平台也起到了搜索引擎的作用，即便有上述诸多优点，但在逐渐发展壮大的同时也面临着诸多挑战：1）随着社区问题数量急速增加，越来越多的新问题无法及时得到有效的解答；2）随着大量用户的参与，社区用户质量参差不齐，导致社区充斥着许多劣质的问题和答案；3）社区数据和用户增加同样带来了搜索引擎所面临的问题，在某些情况下用户需要主动的去搜索自己感兴趣的问题及内容。面对这些挑战，问答社区同样需要引入了推荐系统进行信息筛选与推送。问答社区需要解决的推荐难题有个性化问题推荐、专家推荐、标签推荐、最佳答案排序等等。个性化问题推荐是指系统给社区用户推荐其可能感兴趣的问题，从而使用户参与到这些问题的讨论、解答过程中，这不仅可以促进社区知识、信息的流动，还可以增强用户粘性；专家推荐是指用户在提出一个新问题后，系统根据问题内容自动推荐一些专家用户，然后提问者就可以主动邀请这些专家来解答问题，这样不仅能够快速有效地解决新的问题，还能增加用户对社区的依赖性，有助于社区良性发展；标签推荐是指用户在提问时，社区根据用户的问题内容推荐一些能够有效反映问题主题的词语作为问题的标签，标签可以有效提升社区检索信息的速度，并且有助于分类问题。

1.2 国内外研究现状

随着 quora，stackoverflow，知乎等新型问答社区的兴起与流行，许多研究人员将研究目标从 Yahoo! Answer，百度知道等传统的问答社区转移到这些新兴的问答社区。StackOverflow 作为计算机科学领域的知识问答平台，具有很强的专业性，用户的兴趣领域相对比较集中且容易建模分析，同时问题一般都属于事实类型，比较容易判断答案的正确与否；知乎，quora 作为国内外知识问答共享平台的优秀代表，具有丰富的社区属性和研究价值。不同于 stackoverflow，这类社区用户类型比较广泛，包含各行各业的人，因此社区复杂性较高，用户的兴趣行为难以准确的判断，另一方面社区既包括事实类型的问题，也有非常主观性的问题，因此用户提供的答案质量高低难以判断。

在线问答社区推荐算法研究按照推荐场景可以分为：相似问题检索、个性化问题推荐、专家推荐、最佳答案发现、标签推荐等。

1.2.1 相似问题检索

在线问答社区中，用户在描述问题的时候，系统会自动检索出相似的问题供用户参考，通过相似问题检索一方面有助于用户快速获得答案，另一方面可以有效的

减小社区相似问题的冗余度。如图 1-1 所示的知乎社区相似问题检索。目前的相似问题检索研究主要采用的是基于问题文本相似度的方法。论文[1]提出了基于语言模型的翻译模型 TransLM，对于给定的两个问题 q_1, q_1，该模型首先计算两个条件生成概率 $p(q_1 \mid q_2)$ 和 $p(q_2 \mid q_1)$，即给定一个问题，计算生成另一个问题的概率，然后用这两个概率的平均值表示问题的相似性。概率定义如公式(1-1)和(1-2)所示：

$$p(q_1 \mid q_2) = \prod_{w \in q_1} p(w \mid q_2) \tag{1-1}$$

$$p(w \mid q_2) = \frac{|q_2|}{|q_2| + \lambda} \cdot p(w \mid q_2) + \frac{|q_2|}{|q_2| + \lambda} \cdot p(w \mid c) \tag{1-2}$$

论文[2]提出基于单词嵌入和标记对齐的神经网络模型，对于给定的两个问题，首先利用词向量嵌入方法 word2vec 学习问题的词向量表示，然后根据两个问题的词向量表示通过内积计算得到一个辅助矩阵，并对辅助矩阵做行正则化和列正则化，最后将辅助矩阵作为神经网络模型的输入来计算问题的相似度。近年来深度学习技术在计算机视觉、自然语言处理、推荐系统等领域取得了显著的成果。文献[3]采用长短期记忆网络（Long Short-Term Memory，LSTM）网络解决 CQA 网站的相似问题检索，特别是使用注意力（attention）机制选择问题句子的子集来学习问题的语义表示。

图 1-1 知乎相似问题检索

1.2.2 专家推荐

专家推荐是在线问答社区的一大挑战，如何发现领域专家并邀请其提供高质量的答案是问答社区一直面临的难题。在线问答社区的专家推荐算法通常是建立问题主题与社区用户兴趣之间的关联关系模型，关联关系的强弱代表了用户在某个主题领域的权威性。时至今日，研究人员已经开发了许多问答社区的专家推荐模型。早期的专家推荐算法模型主要是利用信息检索技术发现特定问题领域的专家组[4]。Li 等人[5]研究如何组合问题类别来发现问题到潜在专家的路径。Zhou 等人[6]提出了一种联合学习方法，该方法联合学习答案质量和该答案与问题的相关性来发现问题路由（Question Routing）。Riahi 等人[7]采用一种分段主题模型（Segement Topic Model）来预测可能提供最优答案的用户，实验结果表明，统计主题模型比较适合专家推荐。Mandal[8]提出一种基于主题查询的似然语言模型进行专家推荐，查询的主题是基于单词的词性统计的。Yang 等人[9]提出一个主题经验模型（Topic Expertise Model），该模型通过集成文本内容模型和链接结构分析来联合学习主题和专家经验。除了许多基于主题的模型，其它研究者提出了许多基于矩阵补全（Matrix Completion）的专家推荐算法。论文[10]将社区用户的专业知识用标签来表示，用户标签对（user,tag）分数表示了用户在该标签上的回答质量。另外也有研究者[11]通过引入结合用户社交网络信息和矩阵补全技术来提升专家推荐模型的性能。

近年来，随着深度学习技术的发展，已有研究者将深度学习技术应用于问答社区推荐算法中。论文[12]提出利用卷积神经网络模型进行专家推荐。该算法将问题和用户的兴趣偏好用单词嵌入向量来表示，然后将单词嵌入作为卷积神经网络的输入特征进行学习，网络的输出层使用 softamx 层表示预测专家的概率。论文[13]提出了一种排序度量网络学习框架，该框架模型探索用户与问题的相关性质量排序和用户的社交关系。

许多已有的专家推荐模型主要是根据用户历史问答行为学习用户的兴趣模型，但是这种方法受限于 CQA 网站用户行为数据稀疏问题[14]以及文本的特征表示[15]的影响，效果并不理想。当前，大多数已有的专家推荐算法主要侧重于学习问题内容的语义表示，例如文献[16]通过词袋（bags-of-words）来发现问题的语义表示，但是这种方法丢弃了句子中单词之间的顺序关系，并不能完全用来表示句子的语义。随着论文[17]提出 word2vec 模型，研究人员开始研究单词的分布式表示来表示句子的语义。Word2vec 模型学习相似单词的语义信息并将单词编码成低维连续空间中的实值向量，因此语义相似的单词具有相似的向量表示，使用 word2vec 模型可以提升了句子和段落的语义表示效果。

1.2.3 个性化问题推荐

问答社区每天会有数以万计的新问题在系统上被提出，然而有相当多数量的问题没有获得答案或没有获得高质量的答案。随着未被回答的问题越来越多，那些提出问题的用户对社区的依赖度会大幅度下降，因此 CQA 面临的一个挑战就是如何有效的将问题推荐给可能感兴趣的用户并获得用户提供的答案，增加用户与社区的交互性，减少提问者等待答案的时间，提升用户参与社区行为的积极性。

论文[18]使用概率隐含语义分析（PLSA）进行个性化问题推荐，算法根据用户过去回答过的问题建立用户兴趣概率分布模型，同时由于在实际的场景下用户只可能回答整个社区的一部分问题，因此为了缓解用户行为稀疏性，模型引入了用户-单词联合分布模型来增强特定领取的用户兴趣发现。论文[19]引入了语言模型对用户兴趣进行建模，作者认为用户兴趣在一个较长时间段内是稳定的并且用户通常只对一小部分特定主题感兴趣。论文根据用户回答过的问题采用语言模型建立用户-问题相似度模型，对于那些用户未知的问题，通过这个相似度模型就可以计算出用户对问题的感兴趣程度。也有许多研究者提出了使用贝叶斯生成模型进行个性化问题推荐，论文[20]提出了 User-Question-Answer 模型，该模型属于贝叶斯生成模型，模型以主题混合表示用户兴趣偏好，同时使用问题分类来提升推荐性能。文献[21]提出了基于主题的用户兴趣模型，发现问题与最佳答案的匹配模型，以此来促进用户提供高质量的答案。论文[22]提出 RankSLDA 算法（Rank Supervised Latent Dirichlet Allocation），这是一个有监督的概率主题模型，RankSLDA 是基于贝叶斯推理框架，该模型通过引入学习排序扩展了监督的隐含狄利克雷分布模型（Latent Dirichlet Allocation Model），引入学习排序可以发现用户更倾向于回答的问题类型。

1.2.4 最佳答案发现

在 CQA 网站，用户在提出问题之后，其余用户可以贡献答案，但是由于用户知识水平的差异性导致答案的质量参差不齐，如果由提问者主动去浏览问题的所有答案并发现最佳答案，这是非常耗时耗力的。因此就需要一种最佳答案发现机制从所有的答案列表中找到一个与问题最匹配且质量最高的答案。

从一个答案列表里找到最佳答案可以看做是一个答案排序问题。已经有许多排序技术被用于最佳答案的排序发现。论文[23]提出的 ExpertiseRank 算法是 PageRank[24]算法的变体，该算法通过构造网络节点图来计算用户在问题领域的经验分数，另外该算法还引入了基于用户提问数量和回答数量的 Z-score 度量指标来

度量用户经验。算法 CQARank[25]使用主题经验模型来评测用户在不同主题下的兴趣和经验分数，以此判断用户提供的答案是否是最佳答案，该模型是一种高斯混合模型。论文[26]提出的算法模型，不仅考虑问题-答案相关性，而且也考虑了回答者的相关信息，以此作为用户是否是问题领域的专家的一个判断标准。该论文基于 stackoverflow 问答数据集分别研究了：1）采用 LDA 模型发现用户在不同领域的经验；2）基于贝叶斯网络判断答案和问题的相关性，最后结合 1）和 2）对问题的答案进行排序，由此发现最佳答案。

基于内容的最佳答案发现研究可以分为传统的方法和基于深度学习的方法。论文[27]假设问题的每个答案都是独立的，对于事实类问题，根据问题和答案的相似度，计算某个答案是最佳答案的概率。论文[28]假设答案和问题之间存在多条潜在的链接，正向链接表示答案质量较高，负向链接表示答案错误或者没有意义，由此将问题转换为二分类问题，然后采用逻辑回归模型进行答案为正或负链接。基于正负链接的预测，提出一种类比推理方法衡量正向链接是最佳答案的概率。

随着深度学习研究的兴起，许多研究人员开始将深度学习技术用于问答网站的答案排序场景。论文[29]提出基于卷积神经网络的答案排序算法，首先由卷积神经网络学习生成问题和答案的向量表示，然后通过最大间隔方法学习正确的答案排序。论文[30]提出使用深度卷积神经网络生成答案和问题的嵌入表示。为了使网络更深，该模型使用多个不同的卷积核，并且在每一个卷积维度上进行 k-max 池化，k-max 池化操作选择每一维度最大的 k 个值，并且保持相对顺序不变。问题-答案排序函数采用线性张量内积，以此挖掘用户、问题、答案之间的多维度交互信息。由于深度学习技术在最佳答案发现场景中的成功应用，文献[31]尝试在两个方面来提升的最佳答案发现算法的性能。首先网络模型不仅仅使用单词的嵌入表示，比如利用 word2vec 训练的词向量，而且将问题和答案中单词的嵌入表示传递给一个双向长短期记忆网络（Bi-directional Long Short-Term Memory Networ, Bi-LSTM）层，这样可以发掘到更多的上下文信息。然后通过一个卷积层和最大池化层学习问题、答案的高纬度特征表示。该模型通过引入 LSTM 网络可以捕获到句子中更多的长范围依赖关系。

1.2.5 标签推荐

在问答社区中，问题标签能够表示问题的语义，并且有助于用户检索问题和发现其感兴趣的话题，同时标签能够增强用户社区参与程度，增强内容消费。给新问题推荐标签面临着许多挑战，一方面问题和标签都是比较短的文本，难以准确提取主题分布；另一方面 CQA 存在着大量的长尾标签，这些长尾标签由于出现的次数

很少，所以难以被发现。

标签推荐的研究工作可以分为两类：基于用户的方法和基于内容的方法。基于内容的方法主要利用用户、标签、问题之间的关系进行个性标签推荐，具体的算法有论文[32]提出的张量分解和基于图的方法[33]。基于内容的方法是通过文本内容信息计算问题-标签的相似度，论文[34]将标签推荐形式化的定义为一个分类问题。Liu[35]等人使用统计机器翻译技术将文本翻译为标签。Yu wu 等人[36]在文献[33]提出的算法基础之上进行了扩展，考虑了相似问题和相似标签。对于一个新问题，首先计算它的相似问题，将相似问题的标签作为候选标签集，并通过语言模型进行筛选；然后再以同样的方式找筛选过后的标签的相似标签。通过两步相似性发现，可以发现更多的长尾标签。概率主题模型同样被用于标签推荐。Stanley 等人在文献[37]中提出一个贝叶斯概率模型来预测 stackoverflow 社区问题的标签。

1.3 本文的主要研究内容

本文主要针对在线问答社区推荐算法展开研究，论文首先结合问答社区特点及推荐系统理论方法分析了问答社区目前面临的信息冗余难题以及可以应用推荐算法的场景，并针对新问题的标签推荐和专家推荐这两个场景设计了新的算法模型，论文的目标是基于真实场景的数据设计出有效可行的推荐算法来提升问答社区推荐算法的性能。论文的主体部分主要围绕标签推荐和专家推荐这两个场景展开，分析并设计有效的推荐算法，并在知乎真实的问答数据集上进行实验验证算法的可行性。

论文首先阐述本文设计的基于神经网络的标签推荐算法。该算法模型将问答社区的标签推荐定义为多标签分类问题，即对于新问题的标签推荐转换为预测新问题的标签。算法分为两个部分，第一部分是基于卷积神经网络（Convolutional Neural Network ，CNN）提取新问题文本信息的 N-gram 特征；第二部分是基于双向长短期记忆网络（Bi-directional Long Short-Term Memory，Bi-LSTM）提取新问题文本信息的上下文语义信息，最后结合提取到的两种类型特征进行多标签预测。

为了给问答社区的新问题推荐专家，本文分析了专家推荐的特点，将专家推荐定义为一个信息检索领域的对级排序学习（Pairwise）问题。算法首先根据用户回答问题的行为及用户在社区的关注关系构造一个异构图，通过在异构图上进行随机游走得到许多节点序列，这些序列是由用户节点和问题节点构成的，利用这些节点序列构造对级排序学习所需的三元组作为训练样本。在进行对级排序学习时，为了计算用户和问题的相似性，分别利用神经网络和用户嵌入矩阵得到用户和问题

的表示向量。因此本文的专家推荐算法核心点是通过训练学习到能够表示问题信息的神经网络和能够表示用户的嵌入矩阵，当对新问题进行专家推荐时，只需要得到问题和用户的向量表示，就可以根据相似性计算函数得到一个专家排序列表作为推荐结果。

为了验证本文设计的推荐算法的效果，本文在知乎真实的问答数据集上进行实验验证，实验结果表明，本文提出的算法模型相对于传统的算法模型在性能上有显著提升，具体实验及结果分析在第五章进行了详细介绍。

本文的主要贡献有：

（1）以问题推荐、专家推荐、标签推荐、最佳答案排序等推荐场景总结了问答社区推荐算法的研究现状与方法。

（2）分析新型的问答社区标签系统的优劣势，设计了结合 Bi-LSTM 和 CNN 网络的标签推荐算法，并利用知乎平台的问题-标签数据集实验验证了算法的可行性和有效性。

（3）设计了结合随机游走和深度学习技术的专家推荐算法。该算法利用用户回答问题的行为记录构建异构图，通过随机游走算法采样节点构造 pairwise 方法所需的三元组训练样本，并结合 Bi-LSTM 网络学习问题的语义向量表示，最后通过排序学习训练网络模型以及用户嵌入矩阵。

1.4 本文的结构安排

本文总共分为六个章节，全文较为全面的总结了在线问答社区推荐算法的研究现状，并设计了针对社区新问题的标签推荐算法和专家推荐算法。论文的具体组织结构如下：

第一章首先介绍了问答社区发展面临的困境，并分析了问答社区需要解决的推荐难题。按照推荐场景，将问答社区的推荐任务分为相似问题检索、个性化问题推荐、最佳答案发现、专家推荐和标签推荐，并详细的介绍了目前的研究现状和研究方法。

第二章是相关理论基础，详细的介绍了目前的问答社区推荐算法采用的理论基础，比如语言模型、概率主题模型、PageRank 及其变体；同时介绍了本文提出的推荐算法模型需要用到的一些理论技术，比如 Bi-LSTM,、CNN 和注意力机制（attention）等深度学习技术以及排序学习相关理论基础。

第三章详细阐述了本文提出的标签推荐算法模型。首先介绍了问答社区的标签特点，然后将问题的标签推荐任务定义为多类别标签分类问题，最后详细介绍了本章提出的 Bi-LSTM 与 CNN 相结合的标签推荐算法。

第四章提出了问答社区的专家推荐模型。首先根据用户提供的答案质量提出用户在不同问题领域的权威性计算方法，然后详细介绍专家推荐算法模型。该算法首先根据用户回答问题的行为和社区关注关系行为构造加权异构图，然后在异构图上进行随机游走采样用户、问题三元组对作为训练样本，并分别通过 Bi-LSTM 网络和用户嵌入矩阵来学习表示问题、用户的嵌入向量，最后使用 pairwise 方法学习用户之间关于问题的强弱排序关系。

论文第五章在知乎的问答数据集上进行实验验证本文提出的标签推荐算法和专家推荐算法的效果，并与已有的算法进行对比，分析本文的算法的优劣性。

第六章对本文进行了归纳总结，并针对本文的算法提出几点改进意见。

第二章 相关理论基础

2.1 语言模型

语言模型可以分为两类：一类是基于形式的语言模型，通过定义一系列的语法和句法规则对句子进行建模；另一种是统计语言模型，将自然语言看作是一个随机现象，然后借助概率统计的方法，统计计算语言的规律。在自然语言处理研究早期，形式语言模型占据着主导地位，专家学者根据语言学理论建立丰富的语法、文法等语言规则来解决自然语言领域的相关问题，但是由于没有取得突出的成果，一些研究人员开始将研究方向转向统计语言模型，随着统计语言模型在机器翻译等领域取得突破性进展，大量的学者开始转向研究统计语言模型。

统计语言模型把句子（单词的序列）看作是一个随机事件，并以一个概率值来描述句子属于某种语言的可能性。语言模型的两个基本功能是：1）判断一段文本是否符合一种语言的语法和语义规则；2）生成符合一种语言语法或语义规则的文本。例如给定一个句子 $s=\{w_1,w_2,...,w_T\}$，根据句子中单词序列的联合概率判断句子是否是一个合法句子：

$$\begin{aligned} p(s) &= p(w_1,w_2,...,w_T) \\ &= p(w_1)p(w_2 \mid w_1)p(w_3 \mid w_1w_2)\cdots p(w_T \mid w_{1:T-1}) \\ &= \prod_{t=1}^{T} p(w_t \mid w_{1:t-1}) \end{aligned} \tag{1-1}$$

公式(2-1)所示的语言模型在计算概率乘积时有两个缺点：一是多个概率乘积导致模型参数众多，使得计算量非常大，另一点是由于数据的稀疏性问题，大部分单词序列不会出现在训练语料库中，导致某些条件概率值为 0，从而使最终的句子概率为 0。为了缓解这种问题，研究人员提出了 N-gram 语言模型。

N-gram 语言模型基于 n 阶马尔可夫性质，假设一个词的概率只依赖于其前面的 n-1 个词，因此有：

$$p(w_t \mid w_{1:t-1}) = p(w_t \mid w_{t-n+1:t-1}) \tag{1-2}$$

当 n=1 时，称为一元（unigram）语言模型。在 n 元语言模型中，条件概率 $p(w_t \mid w_{t-n+1:t-1})$ 也可以通过最大似然函数来计算，如公式(2-3)所示：

$$p(w_t \mid w_{t-n+1:t-1}) = \frac{count(w_{t-n+1:t})}{count(w_{t-n+1:t-1})} \tag{1-3}$$

其中 $count(w_{t-n+1:t})$ 为序列 $w_{t-n+1:t}$ 在语料库中出现的次数。

然而遗憾的是 N-gram 语言模型的最大似然估计也存在数据稀疏问题。因为数据稀疏问题在基于统计的机器学习方法中是一个常见的问题，主要是由于训练样本不足而导致很多单词序列没有出现在训练语料中。在一元语言模型中，如果一个词 w 在语料库里不存在，这就会导致任何包含 w 的句子的概率都为 0。在 n 元语言模型中，当一个 n 元单词组合没有出现在训练语料库时，其概率为 0。为了避免这种情况发生，需要给一些没有出现的单词组合赋予一定概率。数据稀疏问题的一种常用解决方法是平滑技术（Smoothing）。平滑技术是传统统计语言模型中一项必不可少的技术，其思想是增加低频词的频率，降低高频词的频率。比如加法平滑如公式(2-4)所示：

$$p(w_t \mid w_{t-n+1:t-1}) = \frac{count(w_{t-n+1:t})+\delta}{count(w_{t-n+1:t-1})+\delta|\mathrm{V}|} \tag{1-4}$$

其中，$\delta \in (0,1]$为常数。一般δ=1，也叫加 1 平滑。除了加法平滑，还有很多平滑技术，比如 Good-Turing 平滑，Kneser-Ney 平滑等。

统计语言模型广泛应用于各种自然语言处理问题，如语音识别、机器翻译、拼音输入法，字符识别等。在推荐系统领域，基于内容的推荐算法可以根据用户兴趣和物品的内容信息建立统计语言模型分析用户兴趣并进行推荐，因此语言模型在推荐系统领域也有广泛的应用。

2.2 概率主题模型

在机器学习和自然语言处理中，主题模型是一种统计模型，可以发现出现在文档集合中的抽象“主题”，常常作为一种发现隐藏在文本中的语义结构的文本挖掘工具。由于主题模型通常是以概率统计的形式表示文档的主题分布，因此通常也称作概率主题模型，最早的概率主题模型是由 Papadimitriou 等人提出的 LSI（Latent Semantic Indexing）[38]。经典的 PLSA（Probabilistic Latent Semantic Analysis）[39]算法是由 Thomas Hofmann 在 1999 年提出的，模型结构如图 2-1 所示。

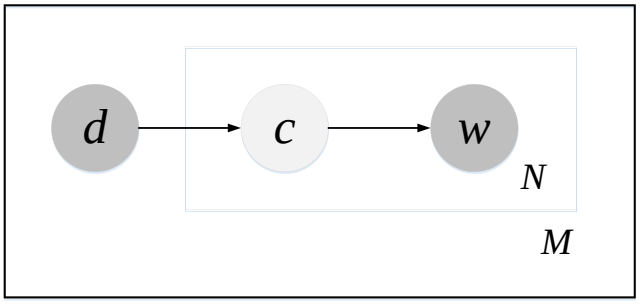


图 2-1 PLSA模型结构图[39]

模型公式如(2-5)所示：

$$p(w|d) = p(d)\sum_{c} p(c|d)\,p(w|c) \tag{1-5}$$

d 是文档索引， c 是从 $p(c|d)$ 中抽取的单词的主题， w 是从 $p(w|c)$ 中生成的单词。 $p(c|d)$ 和 $p(w|c)$ 都是多项式分布。PLSA 有两个缺点，一个是不知道 $p(d)$ 的参数，因此对于一篇新文档无法计算它的概率，另一方面 $p(c|d)$ 的参数个数随着文档数量的增加而线性增加，这可能导致模型过拟合。

迄今为止最常使用的主题模型是隐含狄利克雷分布（Latent Dirichlet allocation，LDA）[40]，它是 PLSA 的泛化版。LDA 引入了 document-topic 和 topic-word 分布的稀疏狄利克雷分布,对文档所包含的少量主题以及主题经常使用的少量单词进行编码表示。在 LDA 中,每个文档被看作是多个主题的混合, 即每个文档有一组通过隐含狄利克雷分布分配给它的主题，这与在 PLSA 中的假设一样，不同点是 LDA 中的主题分布服从稀疏狄利克雷先验，稀疏狄利克雷先验对文件只包含少量的主题以及主题经常只使用少量的单词这两个信息进行建模。LDA 图模型如 2-2 图所示。

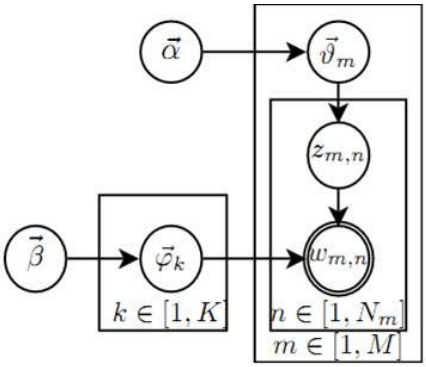


图 2-2 LDA模型图[40]

LDA 的主题建模可以分为两个步骤：

（1）$\vec{\alpha} \to \vec{\upsilon}_m \to z_{m,n}$。这个过程可以看做是生成第 m 篇文档时，首先从狄利克雷分布 $\vec{\alpha}$ 中采样生成第 m 篇文档的主题分布 $\vec{\upsilon}_m$，然后从主题的多项式分布 $\vec{\upsilon}_m$ 中生成第 m 篇文档的第 n 个主题。

（2）$\vec{\beta} \to \vec{\varphi}_k \to w_{m,n}|k = z_{m,n}$。这个过程是从狄利克雷分布 $\vec{\beta}$ 中生成主题 $z_{m,n}$ 对应的单词分布 $\vec{\varphi}_k$，然后从单词的多项式分布 $\vec{\varphi}_k$ 中生成最终的词语 $w_{m,n}$。

将 LDA 的主题建模概率概括就是：一篇文档包含若干个主题的概率分布，同时每个主题包含若干个单词的概率分布。

2.3 链接预测技术

链接分析技术最初应用在搜索引擎领域解决网页排序问题。受网页排序启发，很多研究者也将链接分析技术用于推荐领域，比如论文[23]提出的ExpertiseRank算法是PageRank[24]的变体，该算法通过构造网络图来计算用户在问题领域的经验分数。论文[41]中介绍的Personalized PageRank算法是最早提出将PageRank算法思想用于推荐系统领域的算法，随后很多学者基于Personalized PageRank算法提出了许多改进的算法模型。

PageRank[24]算法是由谷歌提出来的，该算法根据互联网网页之间的链接关系构建一张图，网页用节点表示，网页之间的链接关系用节点之间的边表示，然后在图中进行随机游走来模拟用户浏览网页的行为，以此来对网页进行全局重要性排序。该算法在计算网页重要性的基于两个假设：一是，如果目标页面被其余很多网页所链接，则说明该目标网页很重要；另一方面，由于网页的质量不同，因此其余网页质量越高，则其指向的目标页面质量也会越高。PageRank算法通过迭代的方式在图中进行随机游走计算每个页面的PR值，直到PR值趋于收敛。PageRank的计算方法如公式(2-6)所示：

$$PR(v_i) = (1-d) + d * \sum_{j \in in(v_i)} \frac{PR(v_j)}{|out(v_j)|} \tag{1-6}$$

d是阻尼系数，它表示从一个顶点随机跳转到下一个顶点的概率。$in(v_i)$表示所有指向顶点v_i的顶点，即v_i的入边集合，$|out(v_j)|$表示顶点v_j的出边个数。

除了PageRank之外，还有许多经典的链接预测算法。HITS算法由Kleinberg在论文[42]中提出，作为一个优秀算法，至今仍然被众多研究者使用。在HITS算法中，每个页面被赋予两个属性：hub属性和authority属性。同时网页被分为hub页面和authority页面这两类，hub指的是那些包含了很多指向authority页面的链接的页面，比如一些门户网站；authority页面则指那些包含实际内容的网页。HITS算法的目标是当用户查询网页时，返回那些高质量的authority网页。HITS算法基于下面两个假设：

（1）一个高质量的authority页面会被很多高质量的hub页面所指向。

（2）一个高质量的hub页面会指向很多高质量的authority页面。

SALSA[43]是另一个优秀的链接分析算法，SALSA可以看做是HITS算法的一个扩展，与HITS的不同之处在于他的authority集合和hub集合是相对独立的，两者之间不会相互增强。SALSA算法基于的假设是，在计算转移概率时，两个节点之间只通过一个节点相连，即这两个节点属于文献计量学里面的共被引或耦合关

系。SALSA 计算过程可以分为两步：首先是对网页集合进行扩充，并构建无向二分图；然后采用 PageRank 的随机游走模型进行链接关系的传播。

2.4 排序学习技术

排序学习是信息检索、协同过滤、在线广告等领域的核心技术，主要目标是借助于机器学习理论，学习训练数据中文档列表中文档之间的相对顺序。早期的排序算法可以分为两种：一种是只基于文档之间的拓扑结构关系来判断文档的重要程度，而不考虑查询的内容，例如 PageRank[24]，HITS[42]等网络结构链接分析方法等；另一种排序算法是基于查询语句和系统中文档之间的相似度，返回文档列表，比如 BM25[44]和基于向量空间模型的余弦相似度值等。第一种方法利用网络拓扑结构计算文档的全局重要性时忽略了文档与特定查询之间的关系，返回的查询结果很可能与查询不相符。对于第二种基于相似度的方法来说，由于查询语句较短，基于向量空间模型的特征表示向量太稀疏，导致计算准确率不高。

目前的排序算法研究主要是利用机器学习技术自动学习排序模型，即排序学习（Learning to Rank，L2R）。根据训练数据集的不同形式，可以将排序学习方法分为三类：点级排序（Pointwise），对级排序（Pairwise），列表排序（Listwise）。

Pointwise 方法假设训练数据中的每个 query-document 对有一个数值或排序得分，因此可以将排序学习问题近似为回归问题，将每一个 query-document 对看作是一个数据样本来预测它的分数。许多现有的监督机器学习算法可以方便地来解决这个回归问题。序数回归和分类算法也可以解决 pointwise 问题。因为对于排序问题而言最终的目的是得到一个准确的文档排序，对于 query-docoment 对的准确分数要求不高，当用序数回归解决 pointwise 问题时，其输出空间为有序的类目标；用分类算法时，输出的是一个无序的多分类标签值，如非常相关、相关、不相关等。常见的 pointwise 学习算法有: Constraint Ordinal Regression[45]，Ranking with Large Margin Principle[46]，OAP-BPM[47]等。

Pairwise 方法与 pointwise 方法不同，它不计算文档与查询的相关性分数，而是对于一个特定查询，计算每两个文档之间的相对有序关系。Pairwise 方法通常将排序问题转化为对文档有序对的分类问题，比如三元组 (q,d_1,d_2) 表示对于查询 q，文档 d_1 比 d_2 更相关，因此学习目标就是对于一个给定查询，判断文档对在三元组中的相对位置，其优化目标是最小化误分类文档对的数量。虽然 pairwise 相对于 pointwise 可以获得更准确的排序结果，但是 pairwise 本身也存在一些缺陷，比如对于某一个查询 q，文档 d_1 是强相关的，文档 d_2 是弱相关的，文档 d_3 是不相关的，而在训练时，三元组 (q,d_1,d_2)，(q,d_1,d_3)，(q,d_2,d_3) 只体现了文档间的相对顺序，

而丢失了相对顺序的强弱关系。常见的pairwise学习算法有：GBRank[48]，RankNet[49]，Ranking SVM[50]等。

Listwise 方法是对于一个给定查询，将整个文档集合作为输入，然后在整个输入集合上优化得到一个最优的排序结果，常用的方法是直接优化排序的评分函数，比如正确排序结果与预测排序结果之间的 KL 散度。Listwise 在损失函数中考虑了文档排序的位置因素，这是 pointwise 和 pairwise 这两种方法所不具备的，listwise 方法一般情况下也比前两种方法具有更好的性能。由于要对排序列表进行直接优化，因此 Listwise 的训练复杂杂度非常高。常见的 listwise 排序学习算法有：ListNet[51]，LambdaRank[52]等。

2.5 深度学习模型

深度学习是机器学习算法的一个分支，它的本质是一种通过使用包含复杂结构或由多重非线性变换构成的多层网络对数据进行高层特征抽象的算法，是机器学习中对数据进行表示学习的一种算法。深度学习模型，特别是基于人工神经网络的模型最早是 1980 年提出的新感知机算法。1989 年扬·勒丘恩（Yann LeCun）等人将标准的反向传播算法[53]应用于深度神经网络的训练并取得了成功。尽管算法可以成功执行，但以当时的数据量和计算性能来说模型性能有限且计算代价昂贵，因此对深度神经网络的研究以及比较低迷。

2007 年，hinton 等人[54]提出了一种训练前馈神经网络中的有效算法。这个算法将网络中的每一层神经元看作是无监督的受限玻尔兹曼机，并使用有监督的反向传播算法进行模型参数的调优。此时的深度学习由于大数据以及高性能图形处理器的出现，训练速度和模型效果有了显著的提升，因此越来越多的研究人员开始研究深度学习。自深度学习出现以来，它已经成功应用在很多领域，以卷积神经网络、循环神经网络为代表的模型在计算机视觉和语音识别领域中已经成熟应用。近年来，深度学习技术开始在自然语言处理领域大放异彩，例如在神经机器翻译[55,56]、机器阅读[57,58]都取得突破性的进展。

2.5.1 长短期记忆网络

循环神经网络（Recurrent Neural Networks, RNN）[59]是人工神经网络的一种，它的特点是同一个隐藏层之间的神经元也有连接，即每个隐藏层神经元的输入同时包括前一层神经元的输出和前一时刻该隐藏层的输出，这里的时刻是序列中前后顺序的概念，不是时间概念。典型的循环神经网络如图 2-3 所示。

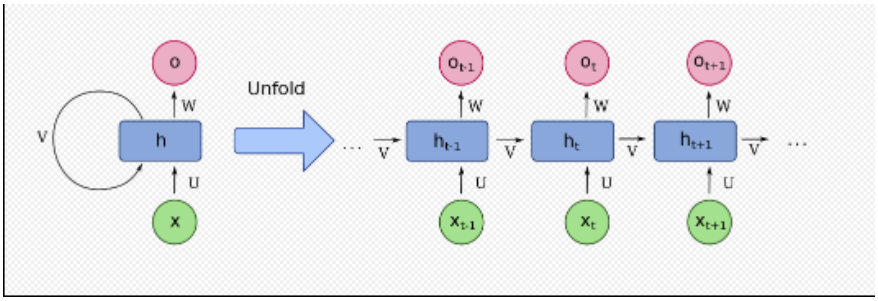


图 2-3 循环神经网络结构图[59]

基本的 RNN 的隐藏层和输出层公式(2-7)和(2-8)所示：

$$h_t = f(Ux_t + Vh_{t-1}) \tag{1-7}$$

$$o_t = g(Wh_t) \tag{1-8}$$

h_t 是 t 时刻隐藏层状态向量，它的输入包括输入层输入 x_t 和 t-1时刻隐藏层状态向量 h_{t-1}， o_t 是 t 时刻的输出向量。

RNN 被广泛应用在处理变长输入的场景中，例如文本、语音等，但是当序列很长时 RNN 的性能就会大大降低，主要原因是对于长序列而言，RNN 的隐藏层保存的历史序列信息会丢失，同时隐藏层之间的连接权重在序列的每一步都会被重复使用，导致该权重对隐藏层状态的影响逐渐增大，一旦发生变化则对整个网络影响很大，这种情况导致 RNN 训练过程中会出现梯度消失（gradient vanish）的情况。

LSTM[60]是 RNN 的一种非常流行的变体，它通过引入多个门控的概念代替了 RNN 中重复使用隐藏层连接权重的情况，有效地缓解了 RNN 的梯度消失问题，同时引入遗忘门可以根据实际情况保留序列的历史信息，起到了长距离依赖的作用，这一点是传统的 RNN 网络不具备的特点。

给定一个输入句子 $X = \{x_1, x_2, ..., x_n\}$， x_t 是一个 N 维的词向量，则隐藏层向量 h_t（维度为 H）在时间步 t 的更新方式如下：

$$i_t = \sigma(W_i \cdot [x_t, h_{t-1}] + b_i) \tag{1-9}$$

$$f_t = \sigma(W_f \cdot [x_t, h_{t-1}] + b_f) \tag{1-10}$$

$$\overline{C}_t = \tanh(W_c \cdot [x_t, h_{t-1}] + b_c) \tag{1-11}$$

$$C_t = i_t \cdot \overline{C}_t + f_t \cdot C_{t-1} \tag{1-12}$$

$$o_t = \sigma(W_o \cdot [x_t, h_{t-1}, C_{t-1}] + b_o) \tag{1-13}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{1-14}$$

在 LSTM 架构中有三个门，分别是输入门 i_t、遗忘门 f_t、输出门 o_t，还有一个 C_t（cell memeory vector），激活函数 σ 采用的是 sigmoid 函数。输入门决定输入向量 x_t 如何改变 cell memory 的状态，输出门决定 cell memory 的值如何影响隐藏层的输出，遗忘门决定 cell memory 记忆或者遗忘神经网络之前的状态。

与单向 LSTM 相比，Bi-LSTM[61]在两个方向上同时处理输入句子，生成两个独立的向量序列，一个是前向处理输入句子，另一个是按照相反方向处理输入句子。Bi-LSTM 神经网络每个时间步的输出是两个方向的两个输出向量的连接。Bi-LSTM 在学习单词或句子的语义向量时可以同步捕捉到它们的上下文信息，从而获得更多的语义信息。

2.5.2 卷积神经网络

20 世纪 60 年代，Hubel[62]在研究猫脑皮层中用于局部敏感和方向选择的神经元时发现这些神经元的网络结构可以有效地降低反馈神经网络的复杂性，受其启发提出了卷积神经网络（Convolutional Neural Network, CNN），此时的 CNN 研究才处于理论阶段。1998 年，Lecun 等人提出了 LeNet-5 模型[63]，该模型使用了基于梯度的反向传播算法对网络进行有监督的训练，网络模型的基本结构包括卷积层、下采样层、全连接层和 softmax。卷积层和下采样层的交替连接是为了对原始图像进行特征映射和特征提取，模型最后使用 softmax 对图像进行多分类操作。卷积层可以对图像的局部特征信息进行提取并构成高层次的图像特征，下采样层可以挖掘高层次特征中最重要的特征并达到了降维的目的。LeNet-5 在手写字符识别领域的成功应用引起了研究人员对 CNN 的关注。此时对 CNN 的研究从理论研究阶段进入了模型实现阶段，并且引起了各个领域的研究工作，例如在语音识别[64]、物体检测[65]、人脸识别[66]等领域。近年来随着深度学习研究热潮的兴起，CNN 成为了众多科学领域的研究热点之一，特别是新的 CNN 网络模型的提出激发了众多科学家的研究热情。2012 年，Krizhevsky 等人提出的 AlexNet [67]算法模型在大型图像数据库 ImageNet[68]的图像分类竞赛中以巨大优势夺得了冠军，使得 CNN 再一次成为学术界的研究热潮。从此之后，许多模型结构更复杂，性能更优的 CNN 被提出来，比如牛津大学的 VGG（Visual Geometry Group）[69]、谷歌的 GoogLeNet[70]、微软的 ResNet[71]等。经典的 LeNet-5 网络模型如图 2-4 所示。

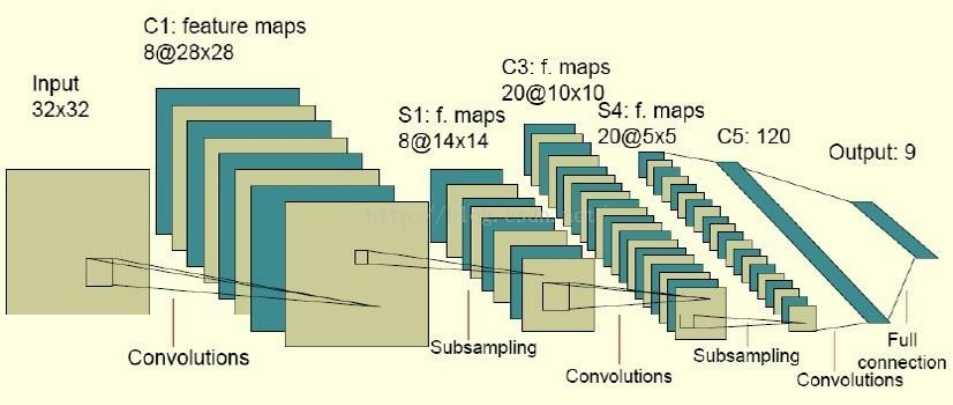


图 2-4 卷积神经网络识别手写数字[63]

典型的卷积神经网络主要由输入层、卷积层、下采样层（池化层）、全连接层和输出层组成。

（1）输入层：卷积神经网络的输入通常是图像，即为二维矩阵（黑白图像）或为三维矩阵（彩图）。

（2）卷积层：卷积层的神经元是利用卷积核和输入层的局部神经元通过 element-wise 乘积得到的，不同于传统的前馈神经网络，这里的每个卷积层神经元只与输入层的部分神经元连接，这可以有效减少网络的参数个数。卷积公式如下：

$$c = f(w \otimes x + b) \tag{1-15}$$

$\otimes$表示卷积操作，w是卷积核，x是输入层的局部区域神经元，c是卷积层神经元，b是偏置，f是激活函数。

（3）池化层：对卷积操作得到的特征进行下采样，通常方法有最大池化、最小池化和平均池化等。例如过滤器大小为 2×2，步长为 2 的最大池化操作如图 2-5 所示。

（4）全连接层：经过多个卷积层和下采样层的交替连接，CNN 最终通过一个全连接层对图像进行分类。

卷积神经网络不同于传统的前馈神经网络，它有诸多特点：

（1）局部连接：卷积层的每个神经元只与输入层的部分神经元连接，而非普通前馈神经网络那样全连接。

（2）共享参数：由同一个卷积核卷积得到的神经元共享同一个参数矩阵，即卷积核。

（3）下采样层可以有效的减少网络模型的参数。

由于以上特点使得卷积神经网络的参数大大减少，减小了模型训练难度。

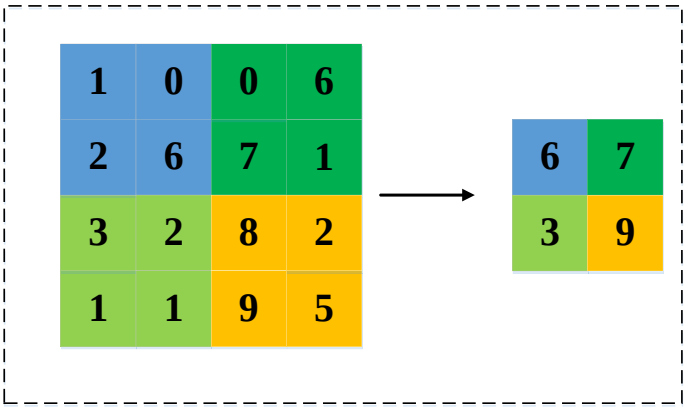


图 2-5 最大池化操作

2.5.3 注意力机制

注意力（Attention）机制最早是在 90 年代被提出来的用于处理视觉问题，最近随着深度学习技术的快速发展，基于注意力机制的神经网络研究成为了一个热点，谷歌在论文[72]中将注意力机制引入到 RNN 模型上进行图像分类。随后 bahdanau 等人在论文[73]中引入注意力机制解决机器翻译问题。在后来的研究中，众多研究人员将注意力机制应用了计算机视觉、自然语言处理等领域，并且取得了较为显著的成果。

注意力机制思想的提出受到了人类注意力机制的启发。人类在观察图像的时候，并不是一次把整个图像都认真看一遍，而是根据观察目的将注意力集中在图像的特定区域，这个区域往往代表了这个图像所反映的主题含义，同理在文本分类等任务中，文档的某些段落、句子或者单词相对于其它部分更能代表文档的主题，因此我们在阅读一篇文档时会将更多的注意力聚焦到这些内容上。

将注意力机制应用在自然语言处理领域的开创性工作是由论文[73]提出的，这篇文章使用一个典型的 Sequence-to-Sequence 框架解决机器翻译问题，也即编码器-解码器（Encoder-to-Decoder）框架，编码器使用一个 RNN 网络对源语言进行编码得到一个表示其语义信息的向量，解码器也是一个 RNN 网络，它将编码器输出的语义向量进行解码，得到翻译结果。如图 2-6 是一个典型的利用编码器-解码器解决机器翻译问题的框架图。

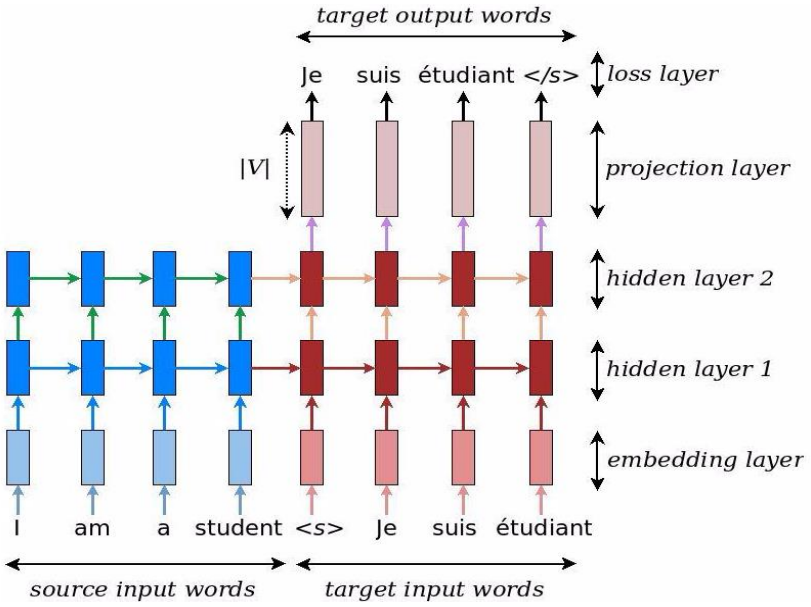


图 2-6 Encoder-to-Decoder架[74]

这种传统的解码器-编码器框架在解决机器翻译问题时取得了不错的结果，但是同时也存在一些问题。首先输入序列不论长短都要被编码成一个固定长度的向量，而解码器在解码的时候所有信息都来源于这个固定长度的向量，这个问题限制了模型的性能，尤其是当输入序列比较长时，这个向量不足以完全表示输入序列的所有信息，使得模型的性能变得更差。注意力机制的思想打破了传统的编码器-解码器框架在编解码时都依赖于一个固定长度的向量的限制，该机制将编码器对输入序列进行编码过程中的每一个中间向量都保存起来，在解码阶段通过选择性的学习这些中间向量来重点关注那些可能影响解码输出结果的输入向量。将注意力机制应用在机器翻译领域，我们可以简单的认为目标语言的每个单词都以一定的概率对应于源语言中的每个单词，因此在解码器翻译阶段，我们不仅考虑源语言的语义向量表示，而且还关注那些与当前输出单词对应概率更高的那些源语言单词。编码器-解码器框架中的注意力机制架构如图 2-7 所示。

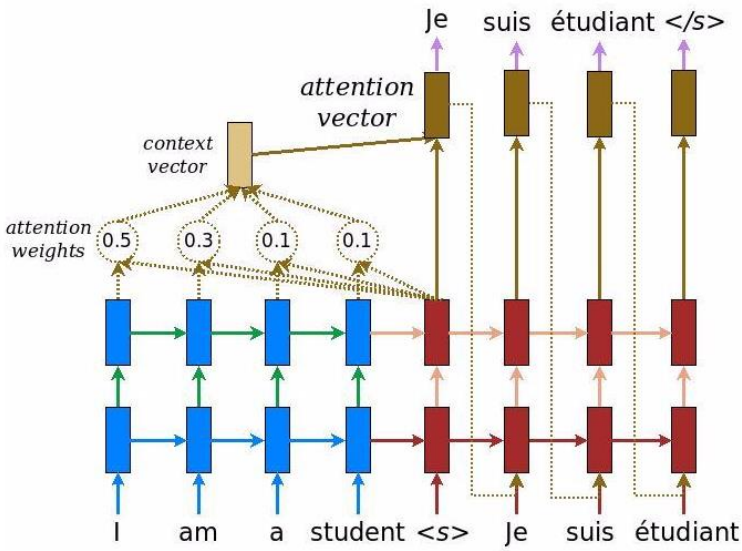


图 2-7 编码器-解码器框架的注意力原理图[74]

图 2-7 所示的注意力机制的计算公式如下：

$$\alpha_{ts}=\frac{\exp(score(h_t,\bar{h}_s)}{\sum_{s=1}^{S}\exp(score(h_t,\bar{h}_s)} \tag{1-16}$$

$$c_t=\sum_s \alpha_{ts}\bar{h}_s \tag{1-17}$$

$$a_t=f\left(c_t,h_t\right)=\tanh(W_c[c_t,h_t]) \tag{1-18}$$

公式(2-16)是解码阶段第 t 个时间步，编码器的第 s 个中间向量的全局注意力权重计算公式，其中 h_t 是解码器第 t 个时间步的隐藏层状态向量，$\bar{h}_s$ 是编码器第 s 个时间步隐藏层状态向量。

公式(2-17)是上下文向量，它是通过编码器每个时间步隐藏层状态向量的全局注意力权重加权求得。

公式(2-18)是解码器第 t 个时间步的注意力输出向量，该向量以上下文向量和隐藏层状态向量为输入，通过 tanh 函数进行非线性映射得到的概率分布向量。

注意力机制的引入虽然会增加神经网络的计算量，但是它显著的提升了网络模型的效果。

2.5.4 Word2vec

Word2vec[17]是谷歌在 2013 年提出的一个用低维实数向量表示单词分布式嵌入的算法。Word2vec 包含的单词嵌入表示模型有连续词袋模型（Continuous Bag-

of-Words，CBOW）和 Skip-gram 两种，模型结构如图 2-8 和 2-9 所示。

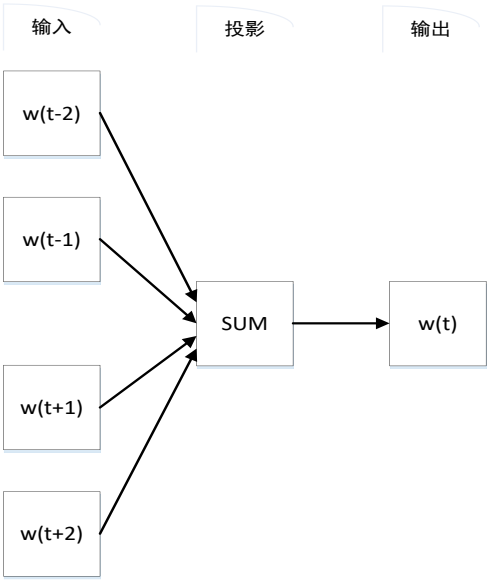


图 2-8 CBOW模型图[17]

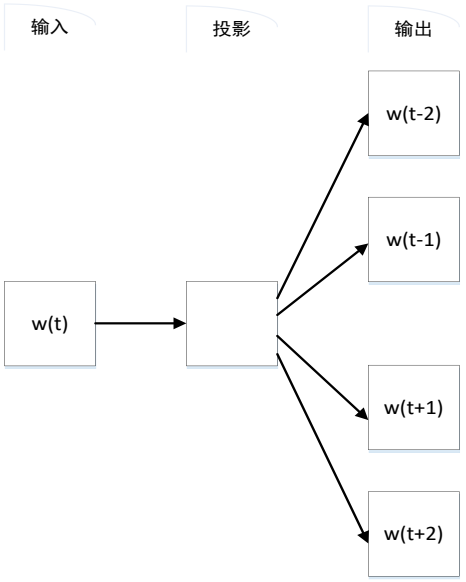


图 2-9 Skip-gram模型图[17]

Word2vec 采用的模型是一个两层神经网络，通过训练可以学习到语料库中每个单词的低维实数向量表示，而向量空间上的相似度可以用来表示单词在语义空间上的相似度，因此很多自然语言处理任务都使用 word2vec 学习单词的语义向量表示。

CBOW 模型的思想是在给定周围上下文词来预测当前词，而 Skip-Gram 模型的思想是给定当前词，预测其上下文窗口内的其余单词。在 Word2vec 被提出之前，最常用的词向量表示模型有：

（1）One-hot 表示

One-hot 是一种非常简单的词向量表示方法，该方法首先创建一个词表，并对每个词进行编号，词表中每个词的向量表示就是一个维度和词表长度相同的稀疏向量，只有当前单词的编号对应的向量元素为 1，其余为 0。这种简单的向量表示方法的一个缺点是向量维度非常大，不便于计算；另一个缺点是无法捕捉词与词之间的相似度，因为这种表示方法丢失了词序关系，即使意义相近的单词，也无法从向量表示中体现出任何关系。

（2）基于传统的 softmax 的神经网络模型

由于 one-hot 表示法的缺点，研究人员开始研究单词的分布式表示方法，分布式表示的思想最早由 Hinton 在 1986 年提出来。基本思想是通过模型训练将每个词映射为一个低维实数向量，并且词语之间的语义相似度可以通过实数向量的距离来表示。Bengio 提出经典的（Neural Network Language Model, NNLM）算法模型

[75]，它采用分布式表示每个词，NNLM 算法模型的目的是给定上下文单词序列，预测下一个单词，模型目标公式为：i-th output = $p(w_t = i \mid conext)$，该算法在预测单词出现的概率时采用的是经典的 N-gram 语言模型，模型首先通过嵌入矩阵 C 得到每个单词的向量表示，然后输出到网络中进行前向传播，嵌入矩阵 C 可以通过训练来更新。NNLM 模型的不足之处是在输出层采用 softmax 层，这大大增加了模型训练过程中的矩阵计算量，使得模型效率非常低下。但是它具有很大的开创性价值，在其之后许多研究人员开始研究基于神经网络的单词分布式表示，但这些模型的一个缺点是，都无法摆脱 softmax 带来的计算代价问题。

Word2vec 相对于 one-hot 表示的优点是其采用了分布式表示的思想，相对于 NNLM 等基于传统的 softmax 的神经网络模型来说它大大增加了训练速度。word2vec 在训练速度上的高效性体现在模型的输出层分别采用了层次 softmax 和负采样这两种方式来加速训练过程。

2.6 文本分类

文本分类是自然语言处理领域中一个非常热门的研究方向，并广泛应用在信息检索、基于内容的推荐算法、计算广告等场景。早期的文本分类方法主要是基于专家规则和专家系统。利用专家系统可以快速解决一些迫切的分类问题，但是由于时间、人力等成本昂贵，无法建立更大领域范围的专家系统，使得很多文本分类问题无法得到解决。随着统计学习方法的发展，特别是机器学习算法研究的兴起，专家学者开始利用统计机器学习方法解决分类问题，同时由于这一阶段互联网的快速发展产生了大量的文本数据，研究人员开发出了一系列解决大规模文本分类问题的统计机器学习方法。至此，基于统计机器学习的方法已经可以解决绝大多数领域的问题。基于传统机器学习方法的文本分类基本流程如图 2-10 所示，分为人工提取特征工程、文本表示、训练分类器这几个阶段。

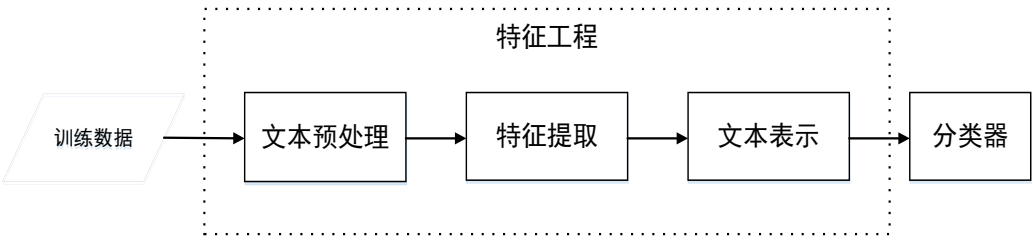


图 2-10 传统的文本分类流程

利用传统的机器学习算法解决问题时，特征工程往往是最重要的一步，需要花费时间以及根据经验去提取、组合有助于解决问题的特征，同时特征工程不同于分

类器模型，不具备很强的通用性，它与问题领域息息相关。抽象的来说，可以把机器学习解决问题的过程看做是从数据中提取信息，然后从信息中提取知识，这两步分别对应特征工程和训练分类器。

利用传统机器学习方法解决文本分类问题时，由于文本数据的复杂性，使得特征工程也变得更加困难。对于文本内容的特征提取，可以分为三个步骤，分别是文本预处理、特征工程、文本表示三个部分，最终是把文本转换成数学表示形式进行算法模型的训练。

2.6.1 文本预处理

中文文本预处理主要可以分为以下几个方面：编码、分词、去除停止词、特殊格式处理。不同于英文单词之间有天然的空格进行分割，处理中文文本的核心步骤是分词，很多研究表明特征粒度为词粒度远好于字粒度，因为基于字粒度的分词损失了过多“N-gram”信息。传统的分词算法主要有基于条件随机场、基于字符串最大匹配、基于句法和语义分析消歧方法等，目前流行的分词方法是基于基于深度学习和条件随机场的方法。目前常用的分词工具有：结巴[76]、哈工大语言云（LTP-cloud）[77]、中科院计算所 NLPIR[78]、清华大学 THULAC[79]。去除停止词也是中文文本处理不可或缺的一步，停止词是指文本中一些对文本处理任务无意义的词，比如介词、连词代词等，去除停止词通常的方法是维护一个停止词表，在特征提取过程中删除文本中出现在停止词表中的词。

2.6.2 文本特征提取

文本特征提取的主要目的是发现那些能够表征文本特征，并对文本分类有作用的单词，同时通过特征提取也可以实现对文档的降维。常用的文本特征提取方法有 TF-IDF、卡方统计、信息增益。

（1）TF-IDF。TF-IDF（Term Frequency-Inverse Document Frequency）[80]叫做词频-逆文档频率，作为一种统计方法，它用来评估一个单词或字在一篇文档或者一个语料库中的重要性。单词的重要性与它在文档中出现的频率成正比，同时与它在语料库中出现的频率成反比。单词在文档中的词频可以通过公式(2-19)来计算：

$$TF_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \tag{2-19}$$

其中 n_{ij} 表示词语 i 在文档 j 中出现的次数，$\sum_k n_{kj}$ 表示文档 j 中的所有词语的个数。

逆向文档频率（IDF）通过公式(2-20)来计算：

$$IDF_i = \log \frac{|D|}{|j : t_i \in d_j|} \tag{2-20}$$

其中$|D|$表示整个语料库中文档的个数，$|j : t_i \in d_j|$表示整个语料库中包含词语 i 的文档个数。

通过词频和逆向文档频率这两个值的乘积得到最终的文档中词语的TF-IDF值，如公式(2-21)：

$$\text{TF-IDF}_{ij} = TF_{ij} * IDF_i \tag{2-21}$$

通过以上方法计算出来的高权重的词，就能作为该文档的关键字。但是 TF-IDF 倾向于筛选掉常用词而保留重要的词语，并且选择出的特征可能不具备类别区分度。主要是因为 TF-IDF 没有考虑单词在不同类别文档中的分布差异，而这个分布差异正好能够体现单词是否具有类别区分度。

（2）卡方检验（Chi-square test）。卡方是一个统计检验量，经常被用来比较实际观察数据和期望从特定假设所获得的数据。卡方值可以用来定量判断预期值和观察值的差异，卡方值越大，说明实际观察值和预期值差距越大，反之亦然。卡方值的计算公式如公式(2-22）所示：

$$\chi^2 = \frac{(o-e)^2}{e} \tag{2-22}$$

其中o是观察值，e是预期值。

利用卡方检验进行文本特征选择时，通常以“单词w与类别c不相关”作为原假设，然后通过实际统计数据，通过计算卡方值来判断原假设是否成立，如果不成立，则说明词w与类别c是相关的，那么词w就可以作为分类特征。因此，使用卡方检验进行文本特征选择就是为每个词计算它与类别c的卡方值，从大到小排序取前k个词作为分类特征。

（3）信息增益。信息熵是由香农提出来用于解决信息度量问题的方法，香农用信息熵的概念来描述信源的不确定度，通俗来讲可以用熵来表示一个系统或者变量的不确定性程度，熵越大，不确定性程度越大。信息熵的公式(2-23)所示：

$$H(S) = -\sum_{i=1}^{C} p_i \log_2(p_i) \tag{2-23}$$

其中p_i表示第i个类别的概率。

信息增益是指信息熵的减少量。在特征选择中，根据某个特征对样本进行分组，分组后样本集合的信息熵相对于分组前信息熵的减少量表示了这个特征的重要性，

例如在决策树算法中，每次按照信息增益最大选择最优划分属性作为分类特征。特征T带来的信息增益计算方式如公式(2-24)所示：

$$IG(\mathrm{T}) = H(C) - H(C|T) \tag{2-24}$$

其中$H(C|T)$为条件熵，公式如下：

$$H(C|T) = p(t)H(C|t) + p(\overline{t})H(C|\overline{t}) \tag{2-25}$$

其中$H(C|t)$表示当特征T出现时的信息熵，$H(C|\overline{t})$表示特征T不出现时样本的信息熵。

在进行文本特征选择时，将每个特征单词按照信息增益降序排列，然后就选择前k个作为分类特征。

2.6.3 文本表示

文本特征工程的最后一步是文本表示，文本表示的目的是把经过特征提取的文本信息转换成计算机可以处理的数学表示形式，这是决定文本分类结果非常重要的部分，传统的文本表示方法有词袋模型（Bag Of Words, BOW）和向量空间模型（Vector Space Model, VSM）。

BOW 模型的思想是仅仅把文档看作是一系列词汇的集合，而忽略单词的顺序、语法和句法等信息，文档中每个单词之间都是独立的，一个单词出现与否不依赖其它单词是否出现。使用 BOW 时，首先从语料库中构建词典，并对每个词赋予一个唯一索引；然后将文档表示成一个维度值和词典大小相同的向量，向量中每个位置元素值表示词典中相同索引位置单词在文档中出现的次数。BOW 模型的优点是简单高效，缺点是在构造文档向量时忽略了单词在句子中出现的次序，同时由于词典维度很大，使得文档的表示向量是一个高纬度且稀疏的向量。

向量空间模型分为两步，首先是进行特征单词选择，比如上节介绍的 TF-IDF，卡方检验和信息增益。第二步是计算特征单词权重，常用的方法是 TF-IDF。

另一类常用的文本表示是主题提取模型。比如 LDA、LSI、PLSA 等概率主题模型等方法。这些方法属于无监督的机器学习算法，区别于基于人工特征的文档向量表示方法，可以通过概率模型发现文本隐含的主题分布，并以此作为文本特征，这些方法得到的文本表示可以认为是文本的深层语义表示。

2.6.4 分类器

文本分类任务的最后一步是训练分类器，传统的文本分类器都是基于机器学习算法的，比如支持向量机（Support Vector Machine），朴素贝叶斯分类算法（Naïve

Bayes）、最近邻、最大熵等等。

2.6.5 基于深度学习的文本分类

前面小节详细介绍了传统的文本分类方法的各个环节，其中最复杂也最重要的是特征工程，但是基于人工的特征工程不仅费时费力，而且提取到的特征质量取决于经验和提取方法，另一方面，基于词带的模型或者向量空间模型的文本表示向量维度非常高，并且非常稀疏，最终影响模型分类效果。深度学习模型的一个本质特点是特征表示学习，对于输入数据，通过深层网络结构可以学习到高层次的特征表示。因此可以利用深度学习模型，例如 CNN，RNN 提取文本语义特征。但是要将深度学习技术应用到自然语言处理领域的前提是如何有效的解决文本表示问题，一个好的文本表示方法不会过多丢失原始文本的信息，有助于模型提取真实可靠的特征信息。本文 2.5.4 节详细介绍了目前最为常用的分布式向量表示方法，用低维稠密的向量表示单词之后，文本数据就可以转换成类似于图像的二维矩阵形式，矩阵行对应文本中的每一个单词，列对应单词向量表示的每一个维度，然后就可以使用 CNN、RNN 等网络模型处理文本数据了。

（1）CNN

论文[81]提出 TextCNN 模型来解决文本分类问题，该模型是 CNN 在图像领域取得成功之后，在自然语言处理领域的又一次成功应用。模型利用 CNN 的卷积核操作提取句子的 N-gram 信息。模型架构如图 2-11 所示。

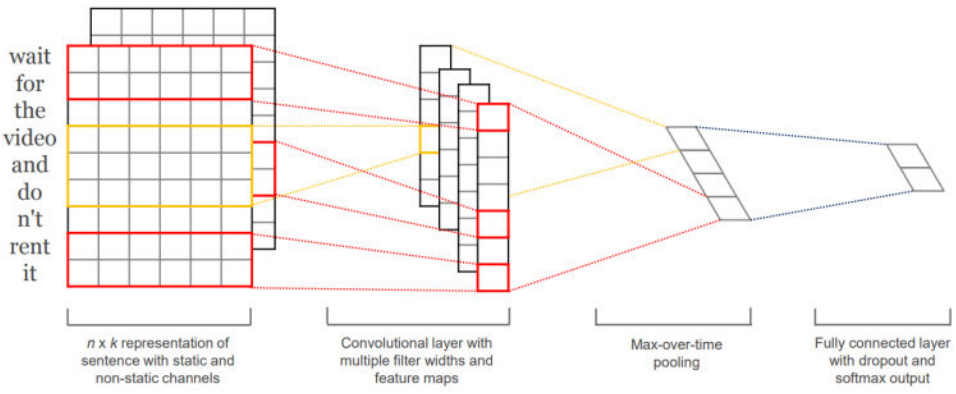


图 2-11 TextCNN模型架构图[81]

该算法的核心部分是双通道（channel）词向量、一维卷积和池化层。对于词向量表示，一种形式是固定利用 word2vec 训练得到的词向量，另一种通道是以 Word2vec 训练的词向量作为初始值，然后在训练过程中通过后向传播进行更新。设置两种词向量表示方法是为了有效抑制模型过拟合。对于一维卷积而言，如图 2-

11 所示，采用多个不同大小的卷积核进行特征提取，并且要将每个卷积核同等作用于双通道词向量表示上。

该算法模型验证了即使采用一层卷积操作，也能取得显著的结果，同时也说明了无监督的词向量预训练技术对于深度学习在自然语言处理领域的使用非常重要。

（2）RNN

尽管使用 CNN 在解决自然语言处理任务中取得了非常不错的结果，但 CNN 的一个固有的缺点是只接受固定长度的输入，同时卷积核的大小也是固定的。由此导致的问题是，CNN 难以同时有效地对长或短句子进行序列信息建模，另一方面在训练模型时，需要仔细的对卷积核的大小进行调节。在自然语言处理领域，比较常用的深度学习模型是 RNN，因为 RNN 可以接受变长序列的输入，并且能够对上下文信息进行建模。双向循环神经网络，特别是 Bi-LSTM 作为 RNN 的一种变体，能够对同时从两个方向对文本序列的上下文信息建模。论文[82]使用双向 LSTM 网络进行文本分类是这方面的代表性工作。使用双向 LSTM 进行文本分时，首先通过一个嵌入层（embedding）得到输入单词的嵌入向量表示，然后将其输入到 LSTM 得到一个隐藏层状态向量，这里采用的是双向 LSTM，因此是同时运行一个前向的 LSTM（从前向后处理文本序列）和后向的 LSTM（从后向前处理文本），并且把由同一个单词得到两个隐藏层状态向量连接形成一个上下文特征向量，最后通过全连接层和 softmax 层进行分类。

2.7 本章小结

本章对问答社区推荐算法使用到的相关理论基础进行了详细的阐述，主要包括理论原理及应用场景。这些理论技术可以分类传统方法和深度学习模型。语言模型和概率主题模型在文本信息建模方面有非常重要的应用价值，基于网络结构的链接分析技术在网页排序，物品推荐、社区专家发现等场景应用广泛。分布式词向量表示及基于深度学习的算法模型在计算机视觉、文本处理、推荐系统等领域取得了显著成果，并且相较于传统算法模型的效果更优。文本分类方法可以分为传统的机器学习方法和基于深度学习的方法，并在信息检索、基于内容的推荐系统、计算广告等场景下有广泛的应用。

第三章 基于深度学习的标签推荐算法

3.1 引言

类似于知乎、quora 这样的新型问答社区的一个重要特点是引入了标签系统，用户在提出问题后，可以手动给问题添加注解型的标签，这些标签代表了问题的语义信息，并且有助于社区搜索和专家发现。如图 3-1 的例子展示了知乎平台的问题“python sklearn 随机森林如何设置每一个特征的分割次数（每个特征的候选阈值个数）？”的标签是“Python，机器学习，集成学习，sklearn”。分析问题的标题及描述信息可以直接提取出“python”、“sklearn”“随机森林”这三个标签，但是通过仔细分析可知，这个问题属于机器学习领域，那么“机器学习”也是一个标签，同时随机森林算法是一种集成学习方法，因此“集成学习”也可以看做是该问题的一个标签。由该例子可知，仅仅根据问题文本的字面含义来提取标签可能会忽略问题隐含的某些话题标签，这种情况下如果标签推荐系统能够发掘问题的隐含语义信息，比如上面问题的“机器学习”和“集成学习”，那么就可以进行准确全面的标签推荐。

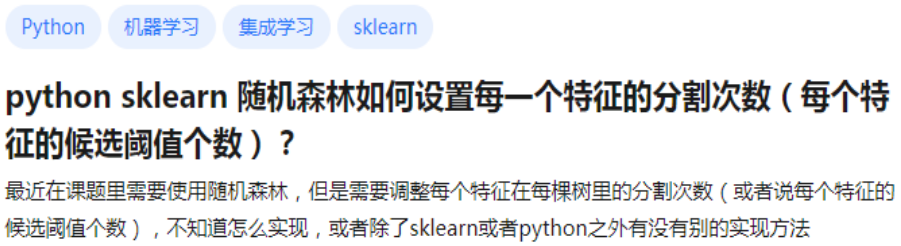


图 3-1 知乎网站问题标签

标签系统的引入很大程度上提升了 CQA 生态系统的用户体验，促进了社区的发展。标签能够使得问题更容易理解，因此可以加速问题被解决的进程。将标签作为关键字，社区用户可以很快发现其感兴趣的问题并可能提供高质量的答案。除了以上诸多好处，一个准确的并且完整的标签系统有助于 CQA 社区的内容消费，例如标签可以促进 CQA 社区内容索引、搜索和知识挖掘。

尽管标签非常重要，但是已有的 CQA 平台的标签系统依然存在问题。根据[83]，知乎平台的一个问题子集里超过 50%的问题的标签少于 3 个，它们不足以代表问题的整个话题范围，这种情况主要由于问题和标签都是短文本内容，难以从中发现隐含的话题，同时很大一部分标签是长尾标签，它们容易被忽略。因此在问答社区

研究系统如何自动提取问题的标签变得非常重要，近年来随着深度学习技术在自然语言处理领域的成功应用，已有研究者将 RNN，CNN 等深度神经网络技术应用到标签推荐。

3.2 标签推荐问题定义

CQA 网站的每一个问题都会对应一组话题标签，这组标签代表了问题所属的话题领域以及问题的语义范围。假设有一组问题 $Q=\{q_1,q_2,...,q_n\}$，Q 中的每个问题 q_i 都有一组标签 $T_i=\{t_{i1},t_{i2},...,t_{im}\}$，本章的目标是对于用户提出的每一个新问题，系统自动给问题推荐一组能够描述问题话题信息的标签。传统的预测文本内容标签是一个多分类问题，即文本只能属于多个标签类别中的一个，但是问答社区的每个问题一般包含多个标签，因此可以将在线问答社区的问题标签推荐看成是一个多标签文本分类问题，即社区中用户提出的问题可以同时属于多个标签。由此，本章研究的问答社区问题标签推荐算法就转换为：对于一个特定问题 q_i，预测它属于标签集合中每个标签的概率，即 $p(q_i \in t_j) = p(t_j \mid content(q_i))$。

3.3 标签推荐算法设计

本章 3.2 节将问答社区问题标签推荐任务定义为多标签分类问题，本节将详细介绍本章提出的在线问答社区标签推荐算法模型。由 2.6 节介绍可知深度学习技术在文本分类领域效果突出，但是已有的模型都是仅仅基于 CNN 或者 RNN，CNN 可以提取文本的 n-gram 特征，但是无法对变长的序列信息进行建模；RNN，特别是 Bi-LSTM 能够对长下文序列信息进行建模，因此本章提出融合 CNN 和 Bi-LSTM 网络特点的标签推荐模型。这两部分分别是基于 Bi-LSTM 的特征提取和基于 CNN 的特征提取。

对问题的文本信息进行特征提取得到两个向量，然后将两个向量连接起来作为问题的特征向量进行分类。具体分别为基于注意力机制的 Bi-LSTM 网络和 CNN。网络结构如图 3-2 所示，模型的输入是变长的单词序列，输出是问题属于每个类别标签的概率。

3.3.1 基于双向长短期记忆网络的特征提取

论文[82]在利用神经网络模型解决文本分类任务时引入了层次注意力机制，分别是单词注意力和句子注意力，单词注意力是指在学习句子的语义向量时计算每个单词对当前句子的重要性，句子注意力是在学习文档的语义向量时衡量每个句子在文档中的重要程度。受其启发，本节利用 Bi-LSTM 对问答社区用户提出的问

题的文本信息进行建模时也使用层次注意力机制。在线问答社区的用户问题分为三个结构：问题标题、问题描述信息、问题所属话题。而问题标题是问题的整体信息概括，为了方便处理，将问题标题和问题描述信息看成一整篇文档。

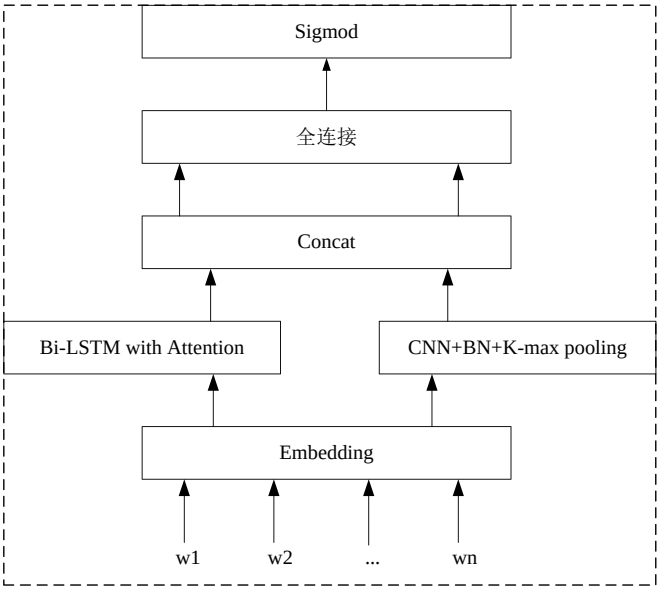


图 3-2 标签推荐算法模型结构图

标签推荐算法模型中基于注意力机制的 Bi-LSTM 网络模型特征提取架构如图 3-3 所示，包括四个层次：单词编码、单词注意力、句子编码、句子注意力。

假设问题有 L 个句子，第 i 个句子 S_i 有 T_i 个单词，w_{it} 表示第 i 个句子的第 t 个单词。本文提出的模型是要将问题映射成语义向量表示，因此需要对单词、句子做向量化表示学习。

单词编码：给定一个包含了 T 个单词的句子，首先通过单词嵌入矩阵 W_e 将单词向量化，即 $x_{it}=W_e w_{it}$，注意这里的嵌入矩阵 W_e 是利用 Word2vec 预训练好的；然后使用一个 Bi-LSTM 网络来学习句子中每个单词的编码信息，Bi-LSTM 网络包含了单词的上下文信息。Bi-LSTM 包含前向 LSTM $\vec{f}$ 和后向 LSTM $\overleftarrow{f}$，它们分别从前向后和从后向前依此读取句子中的单词进行网络表示学习，网络模型如公式(3-1)所示：

$$\begin{aligned} x_{it} &= W_e w_{it}, t \in [1,T], \\ \vec{h}_{it} &= \vec{f}(x_{it}), t \in [1,T], \\ \overleftarrow{h}_{it} &= \overleftarrow{f}(x_{it}), t \in [T,1], \end{aligned} \tag{3-1}$$

然后就可以通过拼接前向隐藏层状态向量 $\vec{h}_{it}$ 和后向隐藏层状态向量 $\overleftarrow{h}_{it}$，得到单词 w_{it} 在当前句子中的语义向量，即 $h_{it}=[\vec{h}_{it},\overleftarrow{h}_{it}]$。

单词注意力：在学习句子的语义表示时，并不是所有的单词贡献完全相同，因此引入单词注意力机制来提取那些对句子的含义影响最重要的单词，并集成这些单词的向量表示形成句子的向量表示。单词注意力公式如下：

$$u_{it} = \tanh(W_w h_{it} + b_w) \tag{3-2}$$

$$\alpha_{it} = \frac{\exp(u_{it}^T u_w)}{\sum_t \exp(u_{it}^T u_w)} \tag{3-3}$$

$$s_i = \sum_t \alpha_{it} h_{it} \tag{3-4}$$

如公式(3-2)所示，首先将单词 w_{it} 的语义表示 h_{it} 输入到一个单层感知器得到 u_{it}，然后通过公式(3-3)计算 u_{it} 和单词级别的上下文向量 u_w 的相似度得到一个正则化权重 α_{it}，以此来衡量单词 w_{it} 的重要性。最后加权计算句子 s_i 的语义表示。上下文语义向量 u_w 可以看成是一个高级的语义表示，首先随机初始化它，然后在训练过程中学习。

句子编码：学习到句子 s_i 的语义表示之后，就可以以同样的方式获得文档的向量表示。同样使用一个 Bi-LSTM 网络来对句子进行编码，公式如(3-5)和(3-6)所示：

$$\vec{h}_i = \overrightarrow{LSTM}(s_i), \quad i \in [1, L] \tag{3-5}$$

$$\overleftarrow{h}_i = \overleftarrow{LSTM}(s_i), \quad i \in [L, 1] \tag{3-6}$$

同样的通过拼接 $\vec{h}_i$ 和 $\overleftarrow{h}_i$ 得到句子 s_i 的文档向量表示 $h_i = [\vec{h}_i, \overleftarrow{h}_i]$。

句子注意力：与单词注意力机制相同，学习文档的向量表示时也引入注意力机制来重点关注那些在文档中比较重要的句子，句子注意力机制公式如下：

$$u_i = \tanh(W_s h_i + b_s) \tag{3-7}$$

$$\alpha_i = \frac{\exp(u_i^T u_s)}{\sum_t \exp(u_i^T u_s)} \tag{3-8}$$

$$q = \sum_i \alpha_i h_i \tag{3-9}$$

其中 u_s 是衡量每个句子重要性的特征向量，它起到了对句子进行重要性衡量的作用，并且是在网络训练过程中学习的。

通过以上基于注意力机制的 Bi-LSTM 网络就可以学习到问题文本的语义特征向量。

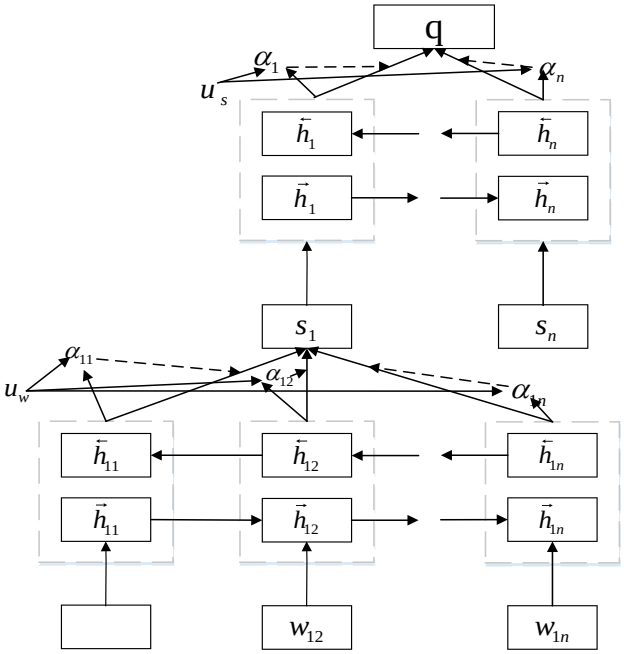


图 3-3 基于注意力机制的Bi-LSTM模型

3.3.2 基于卷积神经网络的特征提取

为了挖掘更多的文本特征，本章的推荐算法模型也引入了卷积神经网络进行特征提取，如 2.5.2 节所述，CNN 中的卷积操作可以发掘输入信息的局部特征，不同的卷积核可以发现不同类型的特征。池化操作能够在选择更重要的特征的同时可以达到降维的目的，从而减少模型的复杂度。

对于本章要处理的问题文本内容信息而言，引入 CNN 的主要原因有：1）卷积操作可以发现文本中的多个不同的 N-gram 特征，并且对于一个 N-gram 可以通过多个卷积核从不同的角度去提取重要信息；2）通过引入不同窗口大小的卷积核可以发现更多的高级特征，模型不容易过拟合。基于 CNN 的特征提取模型细节如下。

（1）卷积特征提取：

假设句子长度为 n 个单词（长度不够时需要填充），则句子可以表示为:

$$x_{1:n} = x_1 \oplus x_2 \oplus \cdots \oplus x_n \tag{3-10}$$

其中$\oplus$是连接操作符。用 CNN 提取图像特征时卷积核的维度可以调节，但是在对文本序列的做卷积时，卷积核的维度要和单词的嵌入向量维度相同，例如使用卷积核 $w \in R^{hd}$ 提取 h 个单词的特征，其中 h 是卷积核窗口大小，d 是单词嵌入向量的维度，卷积特征提取如图 3-4 所示。

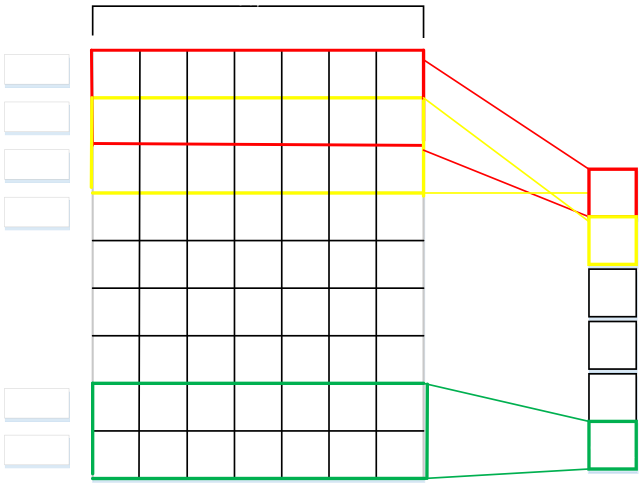


图 3-4 卷积特征提取

卷积映射特征 c_{n-h+1} 是以 h 为卷积窗口大小对单词序列 $x_{n-h+1:n}$ 卷积得到的，如公式(3-11)所示：

$$c_{n-h+1} = f(w \cdot x_{n-h+1:n} + b) \tag{3-11}$$

其中 b 是偏置项， f 是非线性函数，本节使用激活函数的是线性整流函数（Rectified Linear Unit, ReLU），函数图像如图 3-5 所示。

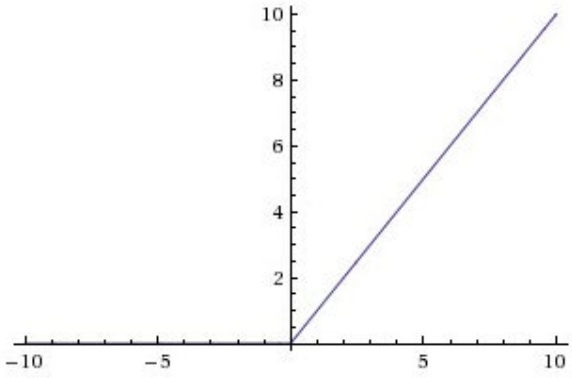


图 3-5 ReLU函数图像

函数形式为：

$$f(x) = \max(0, x) \tag{3-12}$$

该函数被证明在卷积神经网络中比 sigmoid 函数更有效。它有很多优点：

（1）可以提供稀疏激活。在一个随机初始化的网络中，大约有 50%的隐藏层神经网络被激活（非 0 输出），因此使用该函数可以提供神经网络的稀疏表示能力。

（2）没有梯度消失问题。对于 sigmoid，tanh 等激活函数而言，当自变量的值

趋近于无穷大时，函数值几乎不变，也即发生饱和，因此当自变量取值很大时，函数的导数为 0，发生梯度消失问题。

（3）计算效率高。由于 ReLU 只有比较、求和以及乘法计算，因此计算相对于 sigmoid 和 tanh 函数更高效。

ReLU 的缺点是随着训练的进行可能出现神经元死亡，即神经元的值永远为 0，不会改变，导致参数梯度为 0，权重无法更新。出现这种现象的主要原因是学习速率太大，因此在训练时尽量选择较小的学习速率。

对于句子 $x_{1:n}$ 来说，可以使用卷积核 $w \in R^{hd}$，以步长为 1，对其所有窗口序列 $\{x_{1:h}, x_{2:h+1}, ..., x_{n-h+1:n}\}$ 进行卷积操作得到如下一组特征映射：$c = \{c_1, c_2, ..., c_{n-h+1}\}$。

在 CNN 网络中，通过一个卷积核可以提取得到一组同一类型 N-gram 的特征，为了挖掘多种类型的特征，本章基于 CNN 的特征提取模型使用多个不同的卷积核，每个卷积核的窗口大小也不同。

（2）基于批规范化的 CNN：

训练神经网络的过程中，由于前面网络层参数的改变，使得随后每一层的输入分布发生变化，这种情况导致训练深度神经网络变的更加复杂，为了训练一个神经网络，需要一个更小的学习速率和进行非常仔细的参数初始化，但是这会让训练过程变得很慢。为了解决这个问题，论文[84]提出了批规范化（Batch Normalization，BN）技术，基本思想是在训练时，通过 mini-batch 对每一层神经元的激活做规范化处理操作，使得神经元的输出值的均值为 0，方差为 1。

BN 允许使用更大的学习速率，并且不需要仔细地进行参数初始化，同时它还可以作为正则化项，代替 dropout 的作用。BN 算法流程如表 3-1 所示。

表 3-1 Batch Normalizing算法

Input: Values of x over min-batch: $B = \{x_{1....m}\}$, parameters to be learned: γ , β
Output: $\{y_i = BN_{\gamma,\beta}(x_i)\}$
$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i,$ # mini-batch mean
$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_b)^2,$ # mini-batch variance
$\widehat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}},$ # normalize
$y_i = \gamma \widehat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i),$ # scale and shift

为了加速卷积神经网络的训练过程并且减小学习速率和参数初始化方式对网

络模型的影响，本节的特征提取模型使用 BN。在卷积层，通过相同特征映射得到的不同元素采用同样的规范化操作，即共享同一个卷积核的卷积层神经元采用统一的规范化方式。对每一个特征映射学习一组参数 alpha, beta。卷积神经网络中使用 BN 的方式如公式(3-13)所示:

$$c = f(BN(w \cdot x_{i:i+h-1})) \tag{3-13}$$

其中 $BN(\cdot)$ 是 Batch-Normalized，这里只对仿射 $w \cdot x_{i:i+h-1}$ 做 normalization 而不对偏置 b 做 normalization，是因为偏置 b 在 mean substraction 时没有影响。

（3）k-max 池化：

卷积神经网络模型中引入池化操作一方面能够提取最重要的特征，另一方面可以有效降低特征维度，减少模型参数。对于常用的最大池化（max pooling）操作来说，它只提取一组特征映射中取值最大的特征值，而抛弃其余特征值，基于的假设是越大的特征值越能代表这个特征，其余较小的特征值不足以表达该特征。而在自然语言处理任务中，对于一个句子来说，通过一个卷积核提取一组特征映射，这组特征映射值反映了句子主语、宾语等元素的语义信息，如果只取值最大的，那么就可能丢失其余比较重要的信息，这对于分类任务来说是非常重要的。

论文[85]提出 k-max pooling 的思想，给定一个 k 值，对于序列 $S \in R^p$，$p \geq k$，k-max pooling 选择这个序列中 k 个最大的值，并且保持原始的顺序。本章采用 k-max pooling 思想对卷积得到的特征映射进行筛选，k-max pooling 可以保留卷积操作提取到的 k 个映射值，而不像 max pooling 那样只保留最大的特征值而丢失一些可能非常重要的特征信息。

本节介绍的基于 CNN 的特征提取算法模型如图 3-6 所示，首先通过一个嵌入层（Embedding）将输入的单词序列向量化，然后通过多个不同类型的卷积核进行特征提取，如图中所示 Conv(11)，Conv(21)，Conv(31)这三种窗口大小不同的卷积核；在卷积操作之后，通过 BN 做归一化处理，然后通过 ReLU 激活函数进行线性变化，并使用 k-max pooling 进行特征筛选；为了提升模型性能，如图所示采用了多层网络结构。

在使用 CNN 处理文本序列时，需要固定输入长度，因此为了方便处理，将问题文本的多个句子合并成一个长句子进行处理，并且根据句子长度进行截断或者补全。

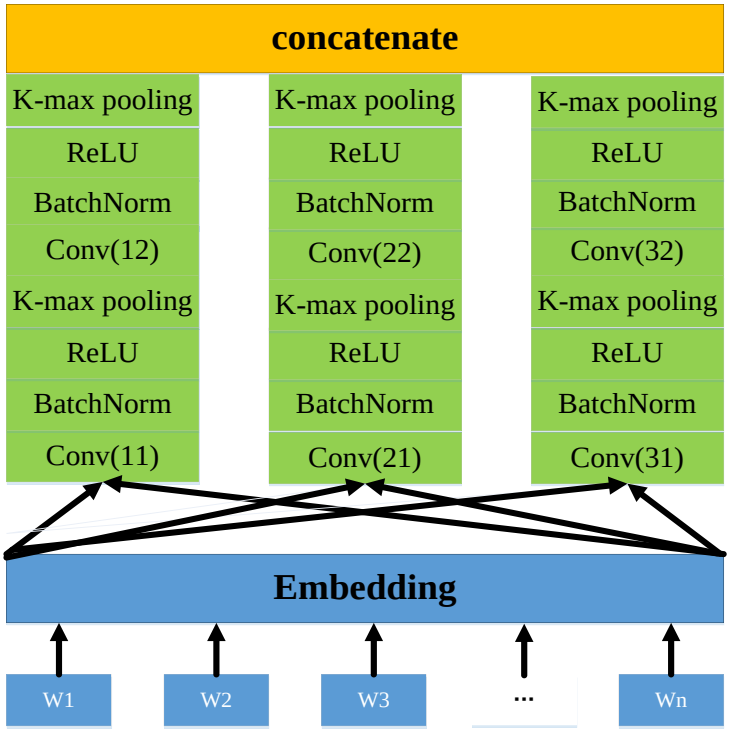


图 3-6 基于CNN的文本特征提取

3.3.3 连接

通过 3.3.1 介绍的基于注意力机制的 Bi-LSTM 网络可以提取问题文本的语义特征信息，由 3.3.2 节介绍的卷积神经网络可以发掘问题文本的多种特征，最后将语义向量信息和特征向量进行连接得到一个维度较高的问题文本表示向量。

3.3.4 输出层

对于多分类问题，每个样本只能属于其中一个类别，这多个类别之间是互斥的，输出层最常使用的是 softmax 函数，它也被称为归一指数函数，输出向量的每个元素都表示一个概率值，并且所有元素的和为 1，softmax 函数形式如公式(3-14)所示：

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \tag{3-14}$$

而对于多标签分类问题，一个样本可以同时属于多个类别标签，因此就不能使用 softmax 函数的概率值来表示样本类别，通常的做法是使用 sigmoid 函数。本章解决在线问答社区问题标签推荐的多标签分类算法模型以 tensorflow 提供的 sigmoiod_cross_entropy_with_logits 作为损失函数，函数形式如公式(3-15)所示：

$$\begin{aligned} l(x,z) &= z * -\log(sigmoid(x)) + (1-z) * -\log(1-sigmoid(x)) \\ &= (1-z) * x + \log(1+\exp(-x)) \\ &= x - x * z + \log(1+\exp(-x)) \end{aligned} \tag{3-15}$$

3.4 本章小结

本章主要研究的是问答社区标签推荐算法。首先分析了问答社区的标签系统，将标签推荐问题转化为一个文本分类领域的多标签分类问题，然后详细阐述了本文提出的标签推荐算法，该算法模型包含两部分，一个是通过基于注意力机制的 Bi-LSTM 网络提取问题文本内容的上下文语义特征，另一部分是通过卷积神经网络提取问题的 N-gram 特征，最后将两部分提取到的特征向量进行连接得到一个表示问题语义及 N-gram 特征的特征向量，然后对其进行多标签分类。

第四章 基于排序学习的专家推荐算法

4.1 引言

在线问答社区的一个核心功能是用户提出新问题，然后其余用户提供答案，但是随着社区逐渐发展壮大，用户提出了许多新的新问题，而很多新问题并没有及时获得答案，为了能够让新问题及时获得高质量答案，在线问答社区发展到如今产生了专家推荐系统，即当用户在社区提出新问题后，系统会自动推荐一些专家用户供提问者邀请来回答问题，这种方式不仅可以使新问题能够及时得到解答，还能获取高质量的答案。如图 4-1 展示了知乎平台新问题的专家推荐结果，用户在提出新问题后，系统会根据问题的话题领域以及系统用户的兴趣偏好及权威性推荐专家，图中红色方框标出了“更多推荐结果”。

图 4-1 知乎专家用户推荐

在线问答社区的专家推荐模型中，核心的一点就是发现新问题所在领域的专家用户，因此系统首先需要确定用户在各个问题领域的权威性，即领域权威性计算，其次需要将用户提出的问题和相应专家用户进行匹配度计算，然后推荐匹配度高并且回答行为较为活跃的专家用户。关于专家的权威性，根据问答社区的实际情况可以分为两点进行考虑：一是用户在某个问题领域下提供的答案获得了数量较多的赞数，这代表用户提供的答案质量较高，反映了用户在该问题领域下具有较高的权威；另一点是如果用户在社区的行为只专注于个别领域，并且在这些领域下面提供了若干高质量的答案，则用户在这些领域下的累计权威性代表了其为专家用户。

本章后面内容会重点讲述本文提出的用户权威性计算方法。在得到用户在各个问题领域下的权威性之后，专家推荐的下一步就是匹配问题和专家，本章提出基于用户权威性的神经网络排序学习算法解决问答社区专家用户推荐问题，该算法借助于深度学习技术学习问题的语义向量表示，然后根据用户在问题领域的权威性高低进行排序度量学习并推荐专家用户，算法的具体细节在本章节后面会详细介绍。

4.2 用户权威性计算

在 CQA 网站给新问题进行专家推荐时，重点考虑的是用户在各个问题领域的权威性，而用户的权威性主要体现在他在社区中回答的答案质量的高低。在 CQA 网站，用户会给他们认可的答案投票，票数的高低代表了其余用户对答案的认可度，也反映出了答案质量的高低。

对于用户提供的每一个答案来说，如果只通过该答案获得的票数来评判答案质量的高低可能存在偏差（由于回答者的知名度或者说其余用户跟风投票），并不能完全反映用户答案的质量以及用户的权威性，本文认为同时考虑用户过去回答的答案的质量和当前答案的质量来计算用户在当前问题下的权威性，即对于用户回答过的每一个问题，根据获得的票数以及历史贡献度加权考虑该用户在每个问题下的权威性。因此本文针对每个问题，提出一个线性累计加权方法计算每个答案的质量，以此表示用户在每个问题下的权威性。

由于用户的经验、兴趣只会集中在一小部分领域，因此在计算用户在每个问题下的权威性时，需要根据问题类型进行领域的划分，并分领域进行权威性计算。在类似于知乎、quora 这样的在线知识问答平台，每个问题都有话题属性，因此可以根据话题将问题进行分类，然后计算用户在每个问题下的权威性时只考虑用户在与当前问题属于同一话题的问题下的历史贡献。

本文提出的 CQA 网站用户权威性计算方法细节如下：

首先将用户回答过的问题根据问题的话题类别进行分类，并且按照时间顺序进行增序排序，然后计算用户在每个问题下的领域权威性。假设用户 u 回答了 n 个属于领域 t 的问题，因此用户 u 在这 n 个问题中每一个问题下的权威性有以下方式计算：

$$a(u,q,t) = (1-\beta)v_l(u,q,t) + \beta a_h(u,t) \tag{4-1}$$

其中 $a(u,q,t)$ 表示用户 u 在当前问题 q 下权威性，并且有 $q \in t$，$a_h(u,t)$ 表示用户 u 在领域t的累计权威性，$v_l(u,q,t)$ 表示用户 u 在问题 q 下提供的答案获得的归一化票数，并且有 $q \in t$，$a_h(u,t)$ 的更新公式如表 4-1 所示。

表 4-1 用户权威计算方法

初始时 $a_h(u,t)=0$， $h=0$
For $q=1\to n,\ q\in t$ do: $a=(1-\beta)v_l(u,q,t)+\beta a_h(u,t)$，其中 $h=q-1$ $a_h(u,t)=a$，其中 $h=q$

β 是超参，可以通过交叉验证选择最佳值。

通过表 4-1 的算法，可以计算出每一个问题下面所有答案的质量，以此来反映用户在问题领域的权威性，将答案按照质量高低降序排序，对于每一个问题，就可以得到一个答案排序列表，对应也就得到用户的排序列表。

4.3 基于用户权威和循环神经网络的排序学习

4.3.1 专家推荐问题定义

在描述具体的专家推荐算法之前，本节先定义所要解决的推荐问题。CQA 网站的专家推荐是指，当用户在网站提出问题之后，系统会根据问题的信息自动推荐一些比较权威的用户，提问者可以邀请这些用户回答问题。因此 CQA 网站的专家推荐核心目的就是找到新问题所属领域的活跃专家，因此这是一个问题-专家匹配问题。

对于社区中的每一个问题都可以根据 4.2 节提出的用户权威性计算方法计算出回答者的权威性，并将回答者按照权威性降序排序，那么每一个问题，都可以根据其回答者列表构造若干个三元组 (q_i,u_j,u_k)，该三元组表示用户 u_j 在问题 q_i 上的权威强于用户 u_k。因此 CQA 网站的专家推荐问题可以看成是一个 pairwise 排序学习问题，通过优化每两个用户之间的偏序关系得到一个用户排序列表。用户之间的偏序关系通过一个排序函数来衡量，假设函数 $sim(u_j,q_i)$ 可以量化用户 u_j 在问题 q_i 上的权威，所以对于三元组 (q_i,u_j,u_k) 来说就有 $sim(u_j,q_i)>sim(u_k,q_i)$，这里的 u_j，u_k 和 q_i 是用户和问题的向量表示。因此专家推荐算法的目标就是通过排序学习得到用户及问题的向量表示方法，然后通过 sim 函数进行用户排序计算得到一个用户列表作为问题的专家推荐结果。

4.3.2 专家推荐算法设计

本节介绍本文的专家推荐算法：基于用户权威和循环神经网络的排序学习，算法流程如图 4-2 所示。算法分为四个部分，分别是异构图构造、局部随机游走、问题及用户向量表示和排序学习，分别如图 4-2(a)、4-2(b)、4-2(c)、4-2(d)所示。

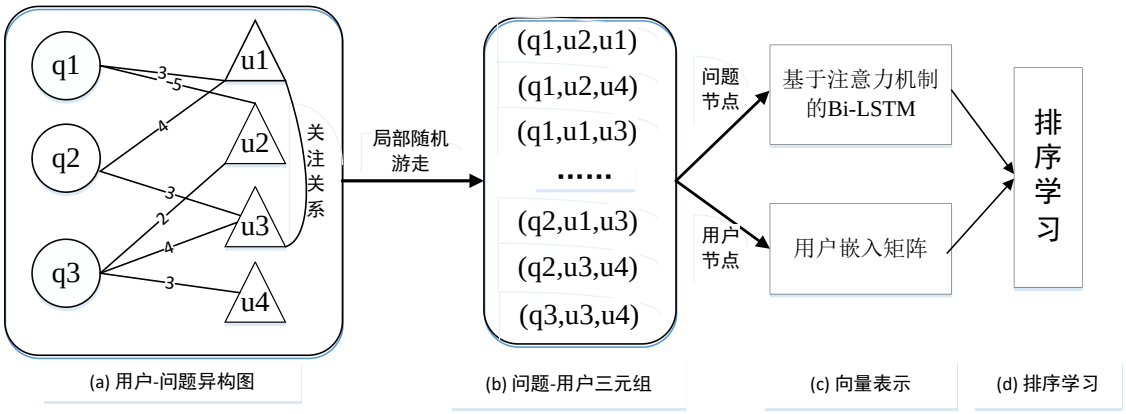


图 4-2 专家推荐模型

（1）异构图构造

在 CQA 网站，用户不仅会和问题进行交互，用户之间也有交互行为，比如用户之间会有关注关系，用户的这些交互行为也反映了用户的兴趣。通过 4.3.1 节的介绍，我们知道可以构造若干个三元组 (q_i,u_j,u_k) 作为训练样本进行排序学习，但是这种学习方式只考虑了用户对问题的行为以及用户在问题领域的权威性，没有考虑用户在网站内的社交关系和问题的语义信息。为了考虑用户的社交行为和回答问题的质量，构造一个异构网络，该网络集成了 CQA 网站中的用户关注关系和用户回答问题的行为。网络用 $G=(V,E)$ 表示，V 是节点集合，由问题集合 Q 和用户集合 U 组成，E 是边的集合，由用户社交关系 S 和问题用户相关性组成，在建立用户与问题的边时，利用 4.2 节介绍的用户权威性计算方法得到的结果表示边的权重，而用户之间的关注边权重取值相同。CQA 网络异构图如图 4-2(a)所示，该图表明，用户 $u1$，$u2$ 都回答了问题 $q1$，但是 $u2$ 提供的答案质量高于 $u1$（$u2$ 与 $q1$ 连接边的权重更大），表明对于问题 $q1$ 来说，$u2$ 比 $u1$ 更具权威，同时 $u1$ 和 $u3$ 有边相连，即 $u1$ 和 $u3$ 有关注关系，引入用户关注关系不仅可以发掘相似用户，而且还能够有效缓解数据稀疏性问题。

（2）随机游走

前面将专家推荐问题定义为 pairwise 排序学习，因此需要这样的三元组 (q_i,u_j,u_k) 作为训练样本，而本章前一小节构造的异构图不仅包含用户回答问题的信息，而且还包含用户的社区关注关系。为了从异构图中获取三元组信息并且利用用户的社区关注关系提升结果，在异构图上采用随机游走模型。随机游走模型最早被用作网页重要性排序和关键字词提取,也在内容推荐[86]和社区检测[87]中作为相似度度量方法，同时随机游走也可以发现网络局部结构的信息。

正是随机游走这种不仅可以作为相似度度量方法，又能够发现网络局部结构特征的特点，本文引入随机游走来发现 CQA 社区用户的隐含行为。在网络中采用随机游走有两个优点。一是对于一个很大异构网络图，可以并行的进行多个随机游走过程，同时发现网络图中不同部分的局部结构；二是可以在图上进行局部随机游走，这样不仅不会丢失太多的信息，而且还会很大程度上减小全局随机游走带来的计算开销。

由于异构图中距离较远的节点之间关联性较小，因此本文采用局部随机游走的方式，在不损失信息的情况下减小全局随机游走带来的计算开销。首先给定局部随机游走路径长度 w，然后以图中的任意一个节点 v_i 为游走起始点开始随机游走，在收敛后，按照游走概率得到一个与 v_i 相似的且长度为 w 的节点列表。列表中的节点类型有用户节点和问题节点，v_i 既可以是用户节点也可以是问题节点。如果 v_i 是问题节点，那么对于列表中的用户节点，就可以根据用户权威或者游走概率构造相应的三元组，例如 (v_i, v_j, v_k)，其中 v_j 、 v_k 是用户节点，表明 v_j 在问题 v_i 的权威强于 v_k ，或者 v_i 游走到 v_j 的概率比游走到 v_k 的更大。而如果 v_i 是用户节点，那么 v_i 与列表中的其余用户节点都是相似节点。

通过上面的局部随机游走，不仅可以得到所需的三元组，而且还可以利用用户的关注关系来提升推荐效果。

$$W = \{v_i, v_{i+1}..., v_{i+w}\} \tag{4-2}$$

（3）问题及用户向量表示

通过前一小节介绍的局部随机游走构造三元组之后进行排序学习，因此需要相应的数学表示方法来表示用户节点和物品节点，由于问题是以文本的形式呈现的，所以可以学习问题的语义向量表示，同时也可以学习能够反映用户兴趣特点的用户嵌入向量，然后通过向量计算的方式来计算排序损失。对于文档的向量表示，传统的表示方法有向量空间模型，TF-IDF，N-gram 等模型，近年来随着深度学习技术的快速发展，众多研究者将其应用到 NLP 领域并取得不错的成就，比如机器翻译、文档分类等等。本文使用在 2.5.1 节介绍的 Bi-LSTM 网络模型结构来学习问题的语义向量表示。而对于用户的向量表示，首先初始化一个维度为 $m*d$ 的用户嵌入矩阵，其中 m 表示用户数，d 表示用户的向量嵌入维度。用户和问题的向量表示是在训练过程中同时学习的。

本节的目的是学习问题（可以看做是文档）及用户的语义向量表示，文档的主题语义主要体现在某些单词和某些句子上，因此在学习文档的语义向量表示时更

多的关注这些单词和句子反映文档主题的部分往往能获得更好的表示结果。首先，文档是有层次结构的，即单词形成句子，句子构成文档，因为首先构造句子的向量表示，然后结合句子的向量表示构成文档的向量表示。其次不同的单词句子在文档中的作用也不一样，而且相同的单词或句子在不同的文档中所反映的含义也不一样。在神经网络模型中引入 attention 机制可以带来两个好处，一是提升了模型的性能；二是提供了一种视角，即哪些单词和句子在文档向量表示学习中占据了更重要的作用。在本文所要解决的 CQA 网站专家推荐场景中，把用户提出的问题看成文档，因此需要发掘问题描述中重要的句子和单词。在文档语义向量表示中不同于 Encoder-to-Decoder 的是，它只有 Encoder，而没有 Decoder，因此如果要采用 attention 机制，就要在 Encoder 阶段使用。

本小节使用 Bi-LSTM 网络来学习问题的上下文语义信息，模型架构类似于本文 3.3.1 节介绍的基于注意力机制的双向长短期记忆网络，但不同点是给问题推荐专家用户时，问题的话题标签是已知的，因此本章提出基于话题标签 attention 的问题文本语义表示。

话题标签注意力机制：由于在 CQA 网站，话题是整个问题含义的总结，因此在学习问题文档的语义向量表示时赋予话题更多的关注，首先利用 word2vec 工具预先学习话题单词的向量表示，假设问题 q_i 包含 m 个话题，u_t 是问题的第 t 个话题单词的向量表示，因此话题 attention 如公式(4-3)，(4-4)，(4-5)所示：

$$u_i = \tanh(W_s h_i + b_s) \tag{4-3}$$

$$\alpha_i = \frac{\exp(\sum_t u_i^T u_t)}{\sum_t \exp(\sum_t u_i^T u_t)} \tag{4-4}$$

$$q = \sum_i \alpha_i h_i \tag{4-5}$$

这里的话题标签 attention 重点关注的是问题的话题属性，公式(4-4)中的 h_i 表示的是句子 s_i 的向量表示，在公式(4-4)中计算每个句子和所有话题单词相似度，然后通过 softmax 函数得到句子的重要性权重 α_i，q 是问题文档的综合语义向量表示。

（4）排序学习

在得到用户和问题的向量表示之后，就可以根据公式(4-6)计算排序损失，并更新模型的参数。

本章设计的基于排序学习的专家推荐算法（Experts Recommendation with Learning to Rank，ERLR）如表 4-2 所示，其中 $lrw(G, v_i, w)$ 表示在图 G 中以节点 v_i

为起始点进行局部随机游走。算法细节介绍如下：

（1）以 CQA 网站的问题、用户为节点，根据用户回答问题计算得到的权威以及用户的关注关系构建异构图，如图 4-2(a)所示。

（2）在图中以某个节点为起始节点开始局部随机游走并按照游走概率选择与该节点相似的 w 个节点，然后按照用户权威及游走概率构造三元组训练样本。

（3）通过用户嵌入矩阵和 Bi-LSTM 网络得到三元组训练样本中用户和问题的向量表示。

（4）根据公式(4-6)计算排序损失，并更新模型参数。

表 4-2 专家推荐算法

算法：基于排序学习的问答社区专家推荐算法（ERLR）
输入：异构图 $G=(V,E)$，局部随机游走长度 w，嵌入维度 d，游走迭代 T，问题-专家列表
输出：基于注意力机制的神经网络，用户嵌入矩阵 $U \in R^{d\times m}$
1. For $t=1\rightarrow T$ do:
3. For each $v_i \in V$ do:
4. $W = lrw(G, v_i, w)$
5. 依据用户权威及游走概率构造三元组
6. 通过 Bi-LSTM 网络和用户嵌入矩阵学习三元组中节点的嵌入向量
7. 根据公式(4-6)计算排序损失
8. 通过后向传播算法更新用户和问题的嵌入向量

$$l(v_i)=\begin{cases}\sum\limits_{\substack{u_+,u_-\in W\\ u_+,u_-\in U}} \max(0, m-(sim(u_+,v_i)-sim(u_-,v_i))), & v_i \in Q\\ \sum\limits_{\substack{u\in W\\ u\in U}} \| u-v_i \|^2 , & v_i \in U\end{cases} \tag{4-6}$$

u_+ 表示对于问题 v_i 提供了获得更多点赞数量、高质量答案的专家用户，u_- 表示对于问题 v_i 来说权威性小于用户 u_+ 的用户。相似性函数 sim 定义为内积，超参 m（0<m<1）控制了损失函数的间隔大小，同时 Q，U 分别是问题集合和用户集合。

根据上面提出的方法，本文将 CQA 网站的问题文本内容信息和用户权威性排序结合在一起构成一个网络排序框架来进行专家用户的发现。根据此方法学习到的模型可以对每一个新问题推荐出一个专家列表。

以上游走过程进行 T 轮，每一次随机选择一个顶点为起始点进行随机游走。可以将上述随机游走过程看作是数据样本选择过程，即每一次游走都会选择若干个节点并构造相应的排序三元组，然后通过最小化排序损失函数来更新模型参数。

需要注意的是问题表示向量矩阵 $Q \in R^{d \times n}$ 和用户嵌入矩阵 $U \in R^{d \times m}$ 是在模型的训练过程中学习的，该模型涉及到的参数有 LSTM 网络模型参数，用户嵌入矩阵 U，本文将这些学习参数统称为 Θ，因此该模型的目标函数如公式(4-7)所示：

$$\min_{\Theta} L(\Theta) = \sum L(v_i) + \lambda \| \Theta \|^2 \tag{4-7}$$

上式中 λ 是正则化系数用来权衡损失和正则化项。

目标函数的优化算法选择了随机梯度下降算法的变体 RMSProp 算法。在 t 时间步，参数 Θ 的更新公式如下：

$$\Theta_{\mathrm{t}} \leftarrow \Theta_{t-1} - \frac{\varepsilon}{\sqrt{\delta + \mathrm{r}}} \bullet g \tag{4-8}$$

其中 $r \leftarrow \rho r + (1-\rho) g \bullet g$，$\rho$ 是衰减率，ε 是全局学习率，g 是梯度。

4.4 生成专家推荐结果

通过训练可以得到一个基于 attention 的 Bi-LSTM 的网络和一个用户嵌入矩阵，由 Bi-LSTM 网络模型可以计算得到任何一个新问题的语义向量表示，通过用户嵌入矩阵可以获得每个用户的嵌入向量。因此在给新问题推荐专家时，首先根据问题所属领域找到在该领域下回答过问题的用户，然后按照排序度量函数 *sim* 来计算用户和问题之间的相对排序并产生最终的专家推荐列表。

4.5 本章小结

本章详细阐述了本文提出的专家推荐算法。首先提出了一个用户权威计算方法，该方法计算用户在一个问题下面的权威性时，综合考虑他在这个问题下提供的答案质量（获得的票数高低）和他过去回答过的答案的质量，这样计算的好处是一方面考虑了用户的历史行为，另一方面缓解了用户当前答案的票数存在虚假的可能性。为了能够有效的给新问题推荐专家用户，将专家推荐问题定义为一个 pairwise 排序学习问题，即对于每一个问题，构造回答者两两之间的偏序关系来表示用户在这个问题下的权威性。为了通过数学计算表示用户之间关于问题的偏序关系，本章使用双向长短期记忆网络学习表示问题的语义向量，同时构造一个用户嵌入矩阵表示用户特征向量。本章的算法模型通过训练得到一个能够学习表示问题特征信息的神经网络和一个能够表示用户兴趣偏好信息的用户嵌入矩阵，因此

给新问题推荐专家用户时，只需要通过神经网络和用户嵌入矩阵得到问题的向量表示和用户的向量表示，然后通过相似度计算函数计算问题和用户的相似度，得到一个用户排序列表作为推荐结果。

第五章 实验设计与结果分析

5.1 引言

本章通过实验验证本文设计的在线问答社区标签推荐算法和专家推荐算法的可行性和有效性，并验证模型参数对算法效果的影响，同时将本文的算法和已有的算法模型进行实验对比并分析结果。

5.2 数据集介绍

本文第三章设计的问答社区标签推荐算法采用的是“知乎看山杯机器学习挑战赛”的官方公开数据集[88]，该数据集包含 300 万个问题，每个问题有一个或多个标签，整个数据集共有 1999 个标签，每个标签对应知乎的一个话题，数据完全符合本文标签推荐算法所要解决的问题。

考虑到用户隐私及数据安全方面的问题，该数据集不包含问题和话题描述的原始文本，而是使用字符编号及分词后的单词编号来表示文本信息。同时，由于词向量的分布式表示技术在自然语言处理领域的广泛应用，数据集提供了字符的嵌入向量和单词的嵌入向量，这些嵌入向量是通过 google word2vec 工具在知乎的海量文本语料训练得到的。

数据集的下载链接 https://pan.baidu.com/s/1c1NXHXA，提取码 5f3q，提供的数据文件包括：

（1）char_embedding.txt 和 word_embedding.txt：分别是字符级别的 256 维的嵌入向量及词语级别的 256 维的嵌入向量。以上两个文件都使用 google word2vec 训练得到的，并保存为 txt 格式。词汇表中省略掉了出现频次低于 5 次的字符或者词语，因此在训练和验证语料中出现的词汇有可能没有对应的单词向量，所以在训练模型的时需要对这些单词的嵌入向量做特殊处理。

（2）question_train_set.txt：训练集中包含的问题信息；总共 5 列，各个列之间用 \t 分割。格式是：question_id ct1,ct2,ct3,...,ctn wt1,wt2,wt3,...,wtn cd1,cd2,cd3,...cdn wd1,wd2,wd3,...,wdn。其中第二列为 title 的字符编号序列；第三列是 title 的词语编号序列；第四列是描述的字符编号序列；第五列是描述的词语标号序列。

（3）question_topic_train_set.txt：问题与话题标签的对应关系。总共有两列，列之间用\t 分割。需要注意的是，如果一个问题对应多个话题标签，这些标签是无序的。数据格式是：question_id topic_id1,topic_id2...topic_idn。

（4）question_eval_set.tx：该文件格式和 question_train_set.txt 一致。

（5）topic_info.txt: 话题信息文件；由该文件可以统计出整个数据集共包含 1999 个标签，因此网络模型的输入向量维度为 1999 维。

由于官方提供的测试数据只包含问题的信息，没有标签信息，故无法离线验证本文的标签推荐算法及对比算法的性能，因此本文第三章的算法模型不使用上面（4）中的测试数据，而是从（2）和（3）中提取出 50 万的问题-话题对应关系作为本文标签推荐算法的测试数据集。

由于知乎的公开数据集只包含问题信息和标签信息，而没有用户行为数据以及用户关注关系，因此本文第四章的专家推荐算法采用的数据集是从知乎爬取的。本文爬取了 2016 年 11 月至 2017 年 7 月知乎网站的问题及用户数据，如表 5-1 所示数据统计结果。

表 5-1 实验数据统计结果

问题个数	答案数	点赞数	用户数	关注关系
1056782	3533101	2536757	302970	4609812

在这 302970 个用户中，只有 23647 位用户回答过问题，其余用户只是评论和点赞，这说明大部分用户并不具备相关领域的知识和经验，他们在社区主要是为了获取相关信息和知识。这 23647 个用户回答的问题数目分布如图 5-1 所示，由图可知，少量用户贡献了大量的答案，即用户回答问题数服从长尾分布。

通过以上统计分析可知，数据集中有大量的问题只有少量答案，同时大量用户没有回答过问题。由于专家推荐算法是根据用户的回答行为进行专家用户推荐，因此就需要对原始数据集进行筛选，本文选择了回答问题数最多的 1000 个用户和它们总共回答过的 634069 个问题，并且保留了这 1000 个用户之间的关注关系。

5.3 数据集分析及预处理

本文第三章实验所采用的数据集，除了对原始文本进行大小写转换、全半角转换及去除一些特殊字符，训练数据和预测数据都没有经过任何清洗。因此在使用该数据集验证第三章设计的标签推荐算法模型时，需要先进行相应的数据处理再进行实验验证。

（1）由于词汇表中省略掉了出现频次低于 5 次的字符或者词语，这些字符和词没有对应的嵌入向量，因此需要对这些词的嵌入向量做特殊处理，首先用特殊标记<UNK>来代替这些低频词，然后以范围[-0.25, 0.25]随机初始化这些低频词的词

向量，这样做是为了保留文本的语义信息，不会造成信息的丢失。

（2）表 5-2 和图 5-2 展示了实验数据集中问题的标签个数分布情况，有 999984 个问题只有一个标签，只有 2300 个问题包含了 6 个标签，大部分问题的标签个数集中在 2 到 4 个，统计计算可得，每个问题平均只有 2.34 个标签，由此可知知乎平台问题的标签分布非常稀疏。

（3）在利用 CNN 神经网络处理文本数据时，需要对文本进行补全或截断来保证网络的输入长度是固定的，在处理问题文本信息时，对问题的标题和描述信息进行截断和补齐，即保证 CNN 网络输入长度固定，如表 5-3 所示，统计结果显示问题的标题单词平均个数是 13，描述单词的平均个数是 58，标签推荐算法实验做截断补齐的长度分别是 30 和 150。

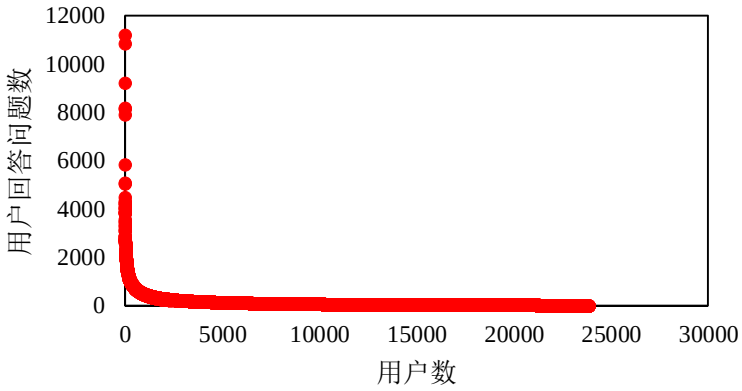


图 5-1 用户回答问题数分布

表 5-2 问题标签个数分布

标签个数	1	2	3	4	5	6
问题数	999984	770310	638821	396903	189585	2300

表 5-3 问题平均单词个数及截断补齐

	平均个数	截断补齐长度
标题单词	13	30
问题描述单词	58	150

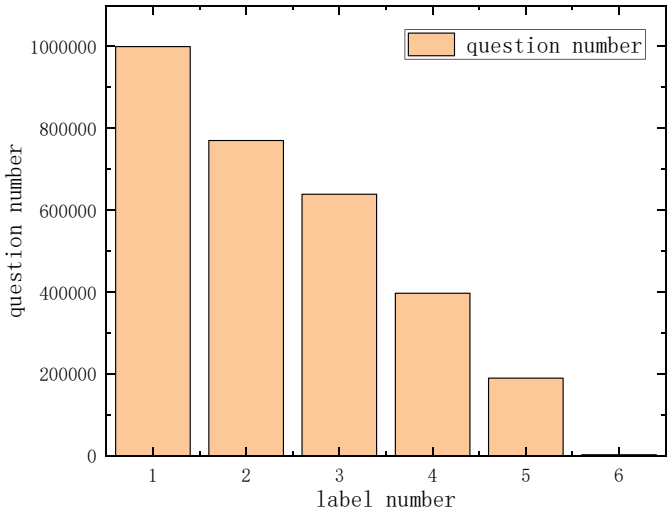


图 5-2 问题标签个数分布

对于专家推荐算法的数据集而言，同样进行文本大小写转换，字符替换，去除停止词及分词处理，并且使用 word2vec 预训练单词的向量表示。

5.4 标签推荐算法实验设计

实验的硬件环境是一台配有英伟达 GTX 1070 显卡，内存为 16GB 的服务器，软件环境如表 5-4 所示。

表 5-4 实验软件环境

环境/开发库	版本
Ubuntu	14.04.5 LTS
python	2.7.12
tensorflow-gpu	1.4.1
pandas	0.20.3
numpy	1.14.0
matplotlib	2.0.2
word2vec	0.9.1
tqdm	4.19.7

原始数据集提供的测试集无法离线验证试验效果。因此在进行试验之前，需要对数据进行划分，得到训练集、验证集、测试集。首先将 question_topic_train_set.txt 中的样本随机打乱，抽取前 50 万条数据作为测试集，然后将剩余数据以 8:2 的比

例划分为训练集和验证集。最后根据对 question_topic_train_set.txt 的划分结果，根据 question_id 的对应关系将 question_train_set.txt 中的数据划也分为训练集、验证集、测试集，并建立问题信息和话题标签的对应关系。

字符及单词的嵌入向量通过 python 第三方库 word2vec 读取 char_embedding.txt 和 word_embedding.txt 这两个文件来获得。

通过统计分析训练数据集中问题的标签个数分布可知，每个问题平均有 2.34 个标签，并且大部分问题只有一个标签。本文实验验证 top1，top3，top5 推荐下的效果。

神经网络模型中影响网络训练速度和模型性能的参数很多，在设计算法验证实验时，主要考虑以下几个对模型训练速度和结果有影响的因素。

（1）Mini-batch size。对于较大的 mini-batch size 值，可以充分利用矩阵和线性代数库来加速计算过程，但是如果 batch-size 太大，会导致权重的更新频率降低，优化过程较长，而 mini-batch size 太小的话，则加速效果不太明显，并且参数更新会更频繁。因此 mini-batch size 的大小要根据数据集合规模、计算能力等因素选择最优值。mini-batch size 权重更新规则如公式(5-1)所示：

$$w \leftarrow w - \eta \frac{1}{n} \sum_{x} \nabla C_x \tag{5-1}$$

其中 n 是 mini batch-size，上面的更新规则也可以看做是 n 个样本的梯度求均值。

（2）正则化参数对模型的影响。在训练机器学习算法模型时，正则化非常重要，它可以有效的抑制模型过拟合，在神经网络模型中，最常用的正则化技术是 dropout。本文的标签推荐算法模型的 CNN 部分采用了 Batch normalized，它可以代替 dropout；Bi-LSTM 部分将采用 dropout。

（3）学习率。如果学习率太小会导致网络模型训练速度非常慢，学习率太大的话则会导致代价函数震荡。除了仔细的选择初始学习率之外，可以使用动态调整学习率的优化算法，比如 RMSProp，AdaGrad。

（4）Early stopping。所谓 Early stopping，即在训练的每一个轮（epoch）结束时，计算验证集的准确率，当准确率不再提高时，就停止模型训练，early stopping 不仅有助于提升模型训练效率，还能有效降低过拟合的风险。但是在具体操作时，要合理把握 early stopping 的时机，不能仅仅根据一个 epoch 的结果就停止训练，而是在训练过程中记录最优的验证集准确率，并且连续若干次训练都没有得到最优值时，可以认为训练效果不会再提升，就可以停止训练，这种策略称之为“no-improvement-in-n”，n 表示的是 epoch 的次数。

（5）词向量的维度。词向量的维度也会影响网络模型的性能；由于本节实验所采用的数据集已经预先训练好了单词的嵌入向量，因此就不需要通过实验验证词向量嵌入维度对模型结果的影响。

（6）卷积核大小及个数。在 CNN 中，卷积核的大小和个数是一个非常重要的超参，为了挖掘更多不同层次的文本特征，本节的实验采用多个不同大小的卷积核。

（7）参数初始化。从均匀分布[-0.1，0.1]中随机采样来初始化网络模型参数。

（8）优化算法。网络使用后向传播算法进行训练，优化算法选择基于梯度的自适应学习率方法 RMSProp[89]。

5.5 标签推荐算法实验结果与分析

实验评测指标采用 topN 推荐常用的三个指标：

（1）准确率（Accuracy）：给测试集每个问题推荐 N 个标签，推荐准确的标签个数占总推荐个数的比例，并在测试集上取均值。准确率计算方法如公式(5-2)所示：

$$accuracy = \frac{1}{|\mathrm{Q}|}\sum_{q\in Q}\frac{|R(q)\cap T(q)|}{|R(q)|} \tag{5-2}$$

其中，$|R(q)|$表示给测试集 Q 中的问题 q 推荐的标签个数，$T(q)$ 表示问题 q 的真实标签列表。

（2）召回率（Recall）：算法正确推荐的标签数占测试集中总标签数的比例。

$$recall = \frac{1}{|\mathrm{Q}|}\sum_{q\in Q}\frac{|R(q)\cap T(q)|}{|T(q)|} \tag{5-3}$$

公式(5-3)的符号含义和公式(5-2)相同

（3）F1 值：F1 指标为 Accuracy 和 Recall 的调和平均数，如公式(5-4)所示：

$$F1 = \frac{2*accuracy*recall}{accuracy+recall} \tag{5-4}$$

本文设计的问答社区标签推荐算法是融合循环神经网络和卷积神经网络的算法模型预测问题所属的真实标签，这个推荐场景也可以看作是文本内容分类，为了验证算法模型的可行性和有效性，选择以下一些算法进行对比实验。

（1）TT。论文[36]提出一个问答社区标签推荐算法，该算法通过计算问题相似度、标签相似度为新问题推荐其相似问题的标签以及这些标签的相似标签，本节以该算法作为基准算法，并将其命名为 TT，实验采用论文提供的原始代码[90]。

（2）TextCNN 和 TextRNN。论文[81]和[82]分别提出使用 CNN 和 RNN 进行文本分类，模型的分类器采用的是 softmax，如果将分类器换成 sigmoid 就可以处理多标签分类问题，而本文所要解决的标签推荐正好可以看成是多标签分类问题，因此在原始的 TextCNN 和 TextRNN 算法模型基础之上，将 softmax 换成 sigmoid 就可以进行多标签分类，并作为本节算法的对比算法来验证本文提出的基于神经网络的标签推荐算法和已有的神经网络模型的效果，TextCNN 和 TextRNN 的代码采用[91]提供的实现。

5.5.1 卷积核窗口对算法性能的影响

在卷积神经网络中，卷积核窗口大小是一个非常重要超参。为了探讨卷积核窗口大小对本节的算法模型的影响，通过实验验证算法在多组不同卷积核窗口下性能变化情况。如表 5-5 所示算法在不同的卷积核下的预测性能，其中(2,3,4)表示窗口大小分别是 2，3，4 的三种类型卷积核。由表可知，当 CNN 模型部分使用窗口大小为(2,3,4,5)这四种类型的卷积核时模型效果最好，而当增加窗口大小为 6 和 7 这两种类型的卷积核，即使用(2,3,4,5,6,7)这六种卷积核时效果最差，并且相对最优结果，F1-score 下降了 10%。从图 5-3 中可以观察到，当在(2,3,4)的基础之上增加一个窗口大小为 5 的卷积核时模型效果有显著提升，而再依次增加窗口为 6 和 7 的核时，模型效果开始下降。通过实验结果可知卷积核窗口大小对本文的标签推荐算法模型性能影响较大，从原理角度来分析，卷积操作可以提取输入数据的局部 N-gram 信息作为高层次特征表示，如果卷积窗口的值太大，窗口内单词语义关联性不强，卷积操作得到的高层次特征不足以表达 N-gram 序列中单词之间的语义相关性，并且还会引入噪声信息影响影响神经网络的学习表示能力。

表 5-5 卷积核窗口大小对模型的影响

卷积核	Accuracy@3	Recall@3	F1-score@3
(2,3,4)	0.5020	0.4815	0.4915
(2,3,4,5)	0.5990	0.5536	0.5754
(2,3,4,5,6)	0.5236	0.5093	0.5164
(2,3,4,5,6,7)	0.4712	0.4505	0.4606

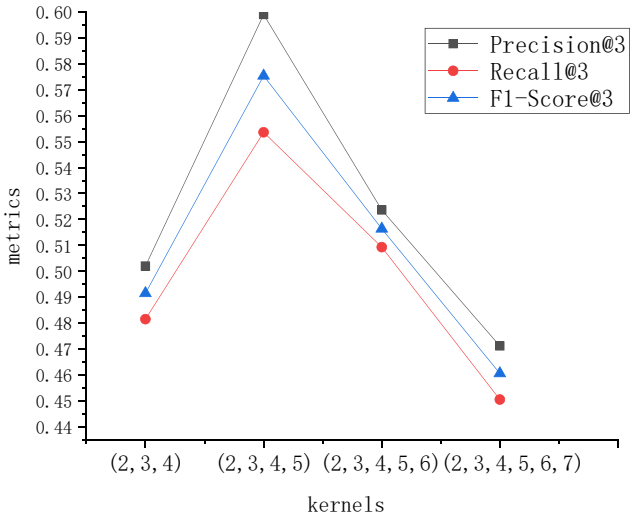


图 5-3 卷积核对模型性能的影响

5.5.2 k-max 池化对算法性能的影响

在 CNN 网络中使用 k-max pooling，可以对同一类型特征的 topk 取值进行提取，并且保留这些特征值的相对顺序，也即保留了输入数据的相对位置信息；k-max pooling 在进行特征值选择时也过滤掉了一些不重要的特征信息，达到了降维的目的。k 的取值会影响提取到的特征多少，设置 k 的取值时需要注意它的值不能大于特征映射的长度。当在 CNN 中使用多个不同大小的卷积核时，k 个取值不能大于窗口最大的那个卷积核得到的特征映射向量长度。例如，假设输入序列的长度是 15，使用的最大卷积窗口是 6，卷积之后得到的特征映射长度是 10，则 k 的取值不能大于 10。为了验证 k-max pooling 对模型的影响，本节通过实验验证 top3 推荐下 k 取 2，4，6，8 时对算法性能的影响。实验结果如表 5-6 所示，由表可知，当 k 取值为 4 时算法效果最好，而随着 k 值的增大，例如 6-max pooling 和 8-max pooling 时算法性能都在下降，实验结果表明，k 的取值太小或者太大都会影响模型效果。k 值对算法有影响的原因也来自于实验所采用的数据集和卷积核，本文实验使用的数据集中问题文本的长度都比较短，通过卷积操作得到的特征映射向量维度也比较小，因此当 k 值太大时，选择的 topK 特征值会包含那些没有表达能力的特征值，达不到特征选择的目的，最终影响模型性能；另一方面如果卷积核窗口太小，而 k 值也太小时，就有可能会丢失掉一些比较重要的特征值，因此在 CNN 网络中使用 k-max pooling 时要综合训练数据和卷积核来选择最优的 k 值。

表 5-6 k-max poolling 对模型效果的影响

K	Accuracy@3	Recall@3	F1-Score@3
2	0.5072	0.4550	0.4797
4	0.5990	0.5536	0.5754
6	0.5146	0.4608	0.4862
8	0.4602	0.4381	0.4489

5.5.3 对比算法实验结果及分析

将本文设计的标签推荐算法模型命名为（Label Recommendation with Deep Learning，LRDL），LRDL 的超参通过验证集进行调优选择。LSTM 的维度是 100，则前向 LSTM 和后向 LSTM 组合之后的向量维度是 200。在基于注意力机制的层次 Bi-LSTM 模型部分，第一层利用 Bi-LSTM 网络通过单词的嵌入向量学习句子的语义向量表示时，输入是 256 维的单词嵌入，输出是 200 维的句子语义向量；第二层利用 Bi-LSTM 通过句子的语义向量学习问题文本的语义向量表示时，输入是第一层输出的 200 维的句子向量，输出也是 200 维的问题文本语义向量。在 CNN 模型部分，使用的是(2,3,4,5)这四种类型卷积核，k-max pooling 的 k 值是 4。

训练时 mini-batch size 大小是 128，并且为了防止由于数据量太大导致内存超载，事先将训练数据集以每 128 个样本为一个 batch 分为若干份，在训练过程中依次加载每个 bach 进行训练，dropout 值采用经验值 0.5，初始学习率是 0.001。

表 5-7 算法对比结果

Metrics	TT	TextCNN	TextRNN	LRDL
Accuracy@1	0.3957	0.5695	0.5952	0.6736
Recall@1	0.2523	0.3402	0.3614	0.3905
F1-score@1	0.3081	0.42595	0.4497	0.49439
Accuracy@3	0.3015	0.4512	0.4703	0.5990
Recall@3	0.2933	0.4097	0.4250	0.5536
F1-score@3	0.2973	0.4294	0.4465	0.5754
Accuracy@5	0.2201	0.3117	0.3206	0.3912
Recall@5	0.3378	0.4301	0.4510	0.5808
F1-score@5	0.2665	0.3615	0.3748	0.4675

本文设计的标签推荐算法模型及对比算法模型在测试集中的 top1、top3、top5 推荐的测试结果如表 5-7 所示。由表可以发现，对于所有模型来说，top5 推荐的准

确率最低，但同时召回率也最高，这是因为给问题推荐的标签越多，从测试集中召回的也会更多，而由于数据集中每个问题平均只有 2.34 个标签，因此推荐结果里会包含一些噪声标签，导致命中率就会大大降低；top1 推荐获得了最高的准确率，因为推荐列表越短，则只会给问题推荐最可能的标签，命中率也会大大提高，但是这样就无法发现那些长尾标签，即那些属于问题，但是出现的次数很少的标签。Top3 推荐的准确率和召回率介于 top1 和 top5 之间。由此得出结论，随着推荐列表的增大，准确率呈下降，而召回率则开始上升。

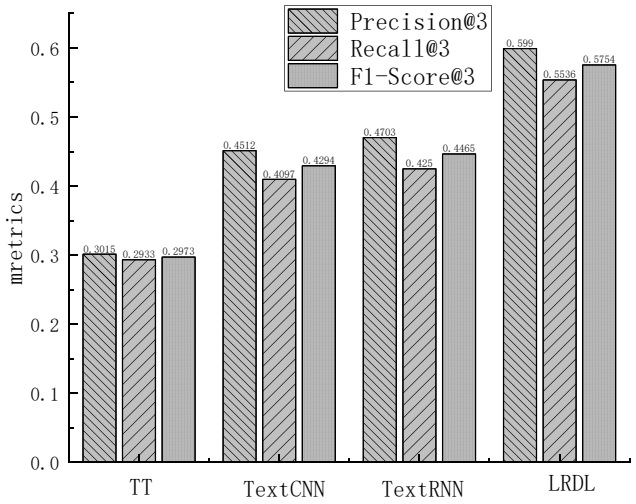


图 5-4 top3 推荐算法结果

图 5-4 更加直观的展示了 top3 推荐下，各个算法模型的性能，在进行模型之间的对比时可以发现，在这些模型中 TT 的效果最差，因为它作为传统的算法模型要进行大量的人工特征提取，比如 unigram，bi-gram，n-gram，而提取到的特征质量决定了模型的性能好坏。TextCNN 和 TextRNN 作为深度学习模型，F1 指标相对于 TT 模型提高了超过 10%。TextCNN 采用多个不同大小的卷积核可以提取多种不同类型的 n-gram 特征，同时通过多层网络可以发现更多高层次的表示特征。TextRNN 模型可以提取文本序列的长下文语义信息，具有非常强的特征表示学习能力。本文的算法模型结合了 RNN 和 CNN 的优点，并且引入了 attention 机制和 k-max pooling，使得模型的特征表示能力大幅度提高。但是需要说明的一点时，本文的模型相对于其他模型都更复杂，因此需要花费更多的训练和计算时间，同时对训练数据量的要求也高。

图 5-5 直观说明本文的标签推荐算法模型随着推荐列表长度的变化，模型性能的变化趋势，由图可知，随着列表变长，准确率呈直线下滑，特别是进行 top5 推

荐时，准确率下降幅度更大；与准确率相反的是，召回率在提高，这是因为推荐列表越长，从测试集中召回的标签也越多。

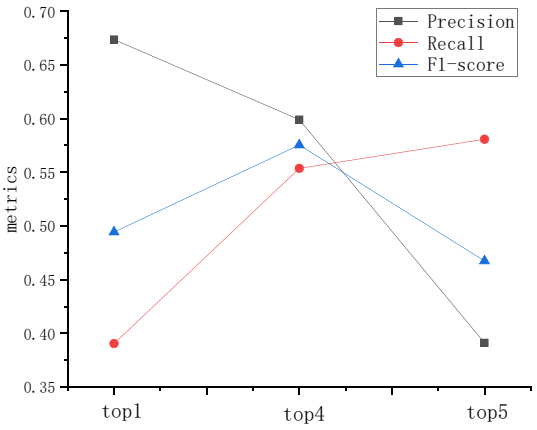


图 5-5 LRDL性能随推荐列表长度变化趋势

本文的算法及对比算法性能较差的一个非常重要的原因是数据集中标签之间存在父子关系，这些具有父子关系的标签之间语义比较相似，因此在预测问题的标签时，受这种语义相似情况的影响，可能会预测出真实标签的父标签或者子标签，而无法预测真正的标签，导致预测准确率下降。另一方面数据集中的标签存在噪声，即问题的标签有误，不能代表问题的真实话题属性和语义信息，也可能影响模型效果。

5.6 专家推荐算法实验设计

为了验证本文设计的专家推荐算法模型，将训练数据集中的问题按照时间戳排序，并以 8:1:1 的比例将问题划分为训练集、验证集和测试集。因为对于每个问题，能够提供高质量答案的用户较少，因此按照 top5，top10，top15 给问题推荐专家用户来验证算法的效果，并且和已有的专家推荐算法进行实验对比，并分析结果。

本文设计的专家用户推荐算法中，影响模型效果的因素有：

（1）算法 4-1 的超参 β，它决定了用户权威性计算结果，最终体现在每个问题的回答者的排序结果上。

（2）局部随机游走长度 w。模型的第二步是在异构图中进行局部随机游走，并采样节点构造三元组作为训练集，因此随机游走长度 w 的大小决定了构造的三元组质量的高低。

（3）dropout 值。dropout 作为网络模型的一种正则化方法，可以有效抑制过拟合。

（4）嵌入向量维度 d。用户嵌入向量及问题的向量表示维度作为一个超参对模型有很大的影响，如果维度太低，就不能完全表示问题的语义信息或者用户的偏好，而维度太高的话，向量就会变得稀疏。

对于用户提出的新问题，专家推荐的目标是能够找到一组能够提供高质量答案的专家列表，本文使用在 CQA 平台最流行的两种专家发现评价指标 *NDCG*（Normalized Discount Cumulative Gain）和 *MRR*（Mean Reciprocal Rank）以及 topN 推荐常用的评价指标准确率，召回率，F1-Score 来度量本文设计的专家推荐算法性能，其中准确率，召回率，F1-Score 已在 5.5 节定义。

（1） *NDCG*

在信息检索领域，衡量排序质量的指标主要指标是 *NDCG*，在 CQA 网站的专家推荐问题中我们可以把用户提出的问题看作是查询，而系统产生的专家列表看作是信息检索领域的文档，因此可以使用 *NDCG* 有效衡量专家推荐列表的排序效果。用 R^q 表示算法给问题 q 推荐的专家用户排序列表，R_i^q 表示排在第 i 个位置的用户，$|R^q|$ 表示给问题 q 推荐的专家用户个数，用户 j 与问题 q 的相关度由专家权威性计算方法得到的值 y_{qj} 表示，因此给问题 q 推荐的专家用户排序 *NDCG* 指标如公式(5-5)，(5-6)所示：

$$DCG = y_{q,R_1^q} + \sum_{i=2}^{|R^q|} \frac{y_{q,R_i^q}}{\log_2 i} \tag{5-5}$$

$$NDCG = \frac{DCG}{IDCG} \tag{5-6}$$

IDCG 是 *DCG* 的理想顺序，如公式(5-7)所示：

$$IDCG = \sum_{i=1}^{|REL|} \frac{rel_i}{\log_2 i} \tag{5-7}$$

REL 根据相关性的正确排序列表。

（2） *MRR*

用 r_{best}^q 表示算法给问题 q 推荐的专家用户列表中，最佳回答者的排序。*MRR* 衡量的是专家推荐的排序质量，公式如下：

$$MRR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{r_{best}^q} \tag{5-8}$$

其中 $|Q|$ 表示测试集中问题的数量，由公式可知，当 $r_{best}^q = 1$ 时推荐质量最高，如果最佳回答者不在推荐列表中则，$\frac{1}{r_{best}^q} = 0$，*MRR* 的最大值是 1，表示算法对测

试集的所有问题都得到了准确的推荐结果。

以上实验指标中，*NDCG*评测的是整个推荐列表的排序质量，*MRR*指标衡量的是最佳专家用户在推荐列表中的排序质量，而 accuracy，recall，f1-score 作为 topN 推荐的评价指标，只是计算推荐的用户是否回答了当前的问题，以及测试集有多少用户被召回，而不考虑推荐列表的排序质量，这三个指标相对于 *NDCG* 和 *MRR* 来说对算法的要求比较低，因为在问答社区大部分用户的行为比较稀疏。

5.7 专家推荐算法实验结果及分析

影响算法模型的参数有：用户历史行为权衡因子 β，局部随机游走长度 w，随机游走轮数 T、用户嵌入（问题嵌入）维度和 dropout 值。神经网络模型部分的参数通过验证集来选择，对于嵌入维度的大小选择 100,200,300,400,500，通过实验发现，当嵌入维度超过 300、游走次数超过 10 次时模型性能趋于稳定。对于嵌入维度这个因素，它一方面会影响网络模型的复杂度和训练速度，同时它也代表了问题及用户主题分布维度，对于知乎而言，话题标签众多，但是它们之间存在包含与被包含关系，所以对于问题、用户的嵌入表示而言，一个维度可以代表一个大类别的话题分布信息，因此即使嵌入维度远小于数据集的话题分布个数，但模型的性能也趋于稳定。

Dropout 的值根据经验取值为 0.5。Attention 机制已被证明能有效增强神经网络的表示学习能力，因此本文的算法模型直接使用 attention 机制，而不验证 attention 机制是否能提升模型性能。

5.7.1 用户历史行为权衡因子 β 对算法性能的影响

在本文提出的算法 4-1 中，公式(4-1)在计算用户在某个问题下面的权威性时考虑到了用户历史回答答案的质量，β 影响着用户历史回答的答案质量的贡献程度，$\beta=0$ 时表示只考虑用户在当前答案的质量，而不考虑他历史回答的答案。本节实验验证进行 top10 推荐时参数 β 对模型性能的影响，β 取值分别是 0，0.2，0.4，0.6，0.8，模型其余参数取值：dropout = 0.5, d=300, $w=6$。

由表 5-8 可以看出，随着 β 值的增大，MRR,NDCG 这两个指标先是增加，然后再减小，当 β=0.4 时这两个指标值达到最优，而当 β 取 0.6，0.8 时结果降低了 2%，β 取 0 和 0.2 时效果也较优。因此可以知道，用户的经验和历史行为能够客观反映用户答案质量高低。Accuracy，Recall，F1-Score 这三个 topN 指标不考虑推荐排序质量高低，而是衡量算法是否能准确给问题推荐专家用户，β 对这三个指标的影响程度相对于 MRR 和 NDCG 来说要小一些，F1 指标的最差结果比最优值降

低了 1%。

表 5-8 β 对算法模型的影响

β	MRR	NDCG	Accuracy	Recall	F1-Score
0	0.5745	0.7723	0.1184	0.1015	0.1093
0.2	0.5810	0.7758	0.1203	0.1128	0.1164
0.4	0.5892	0.7862	0.1156	0.1036	0.1093
0.6	0.5670	0.7675	0.1174	0.1102	0.1137
0.8	0.5677	0.7664	0.1020	0.1087	0.1052

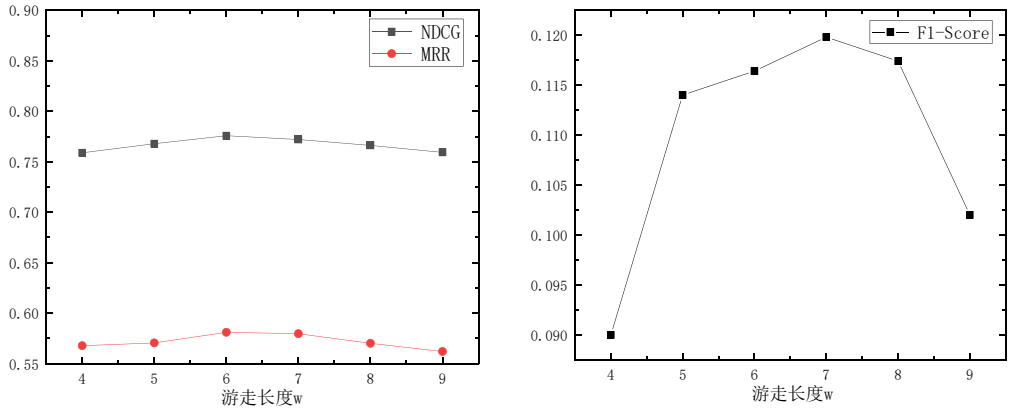
5.7.2 局部随机游走长度 w 对算法性能的影响

本文的算法模型根据用户行为以及社区关注关系构造了异构图，并通过在图中进行随机游走扩充三元组。本节实验研究局部随机游走长度 w 的值取 4，5，6，7，8，9 时模型性能如何变化。如表 5-9 和图 5-6(a)、(b)所示 w 值对 NDCG, MRR, F1-Score 这三个指标的影响。图(a)反映了随着 w 值的增大，NDCG 和 MRR 这两个排序指标的变化趋势，由图可知，随着 w 的值增大，这两个指标先是增加然后减小，并且在 w 取值为 6 时达到最大，分别为 0.7758，0.581。当 w 取值为 4 时，NDCG 的值最小为 0.7589，相对于最优值下降了 2%；而 w 取值为 9 时，MRR 的值最小且为 0.562，比最优结果小了 2%。图(b)反映了 w 对 F1-Score 的影响，由图(b)及表 5-9 可知，w 取值为 4 时，F1-Score 的值是 0.09，而当 w 取值为 7 时，该指标取得最大值 0.1198，比最差结果提升了 2%；而当 w 大于 7 时，F1-Score 开始下降，但是与最优结果的差距在 1%范围以内。由以上实验结果可以知道，NDCG 和 MRR 作为衡量排序质量的指标，随着 w 的改变变化趋势相同，且容易受 w 的影响，而 F1-Score 作为衡量推荐准确率和召回率的指标，w 值的改变对其影响相较于 NDCG 和 MRR 较小。

在异构图上进行随机游走时，如果游走范围太大，采样的节点就会引入一些噪声数据，如果游走范围太小，就无法发现相关联节点之间的三元组关系，这两种情况都会影响模型最终的排序质量。对于不同的数据集，这个采样长度不同，最优取值要根据实际实验结果来取。

表 5-9 随机游走长度 w 对模型的影响

w	NDCG	MRR	F1-Score
4	0.7589	0.5677	0.09
5	0.7679	0.5705	0.114
6	0.7758	0.581	0.1164
7	0.772	0.5796	0.1198
8	0.7662	0.5701	0.1174
9	0.7593	0.562	0.102



(a) 游走长度对NDCG和MRR的影响　　(b) 游走长度对F1-Score的影响

图 5-6 随机游走长度对模型各项指标的影响

5.7.3 对比算法实验及结果分析

本节将本文的专家推荐算法与已有的算法进行对比。

（1）TF-IDF。TF-IDF 是词频-逆向文档频率的简称，目的是计算一个单词在文档中的重要性，可以作为特征提取方法提取问题的特征并构造问题的特征向量表示；对于专家推荐模型来说，通过 TF-IDF 不仅可以提取问题的特征，并且可以根据用户回答问题的文本信息，提取用户的特征表示，然后通过计算问题特征向量和用户特征向量的余弦距离进行推荐。

（2）ExpertsRank[23]。ExpertsRank 算法使用问题-答案的相关关系构造用户行为图，并通过链接结构分析发现专家用户，该算法的主要思想是基于 PageRank 算法。

（3）TSPM[20]。该算法是一个主题概率模型，它通过基于 LDA 的概率模型对问题、用户的主题信息进行建模。

以上三种对比算法代表了三种不同的研究方法，TF-IDF 是传统的特征提取方法，ExpertsRank 是基于网络结构链接分析的方法，TSPM 属于主题概率模型，基于问题文本、用户获得的点赞、反对数等数据对用户偏好建模。本文提出的专家推荐算法模型包括了链接分析、神经网络表示学习以及排序学习研究方法。在本文的方法中，网络的输入是通过 word2vec 预先训练好的词嵌入向量，同时 Bi-LSTM 的权重是使用标准正态分布进行随机初始化。

为了对比不同方法的在五个指标上的性能，算法模型的超参通过交叉验证选择最优值。表 5-10 显示了本文的算法及对比算法在 NDCG，MRR，Accuracy，Recall，F1-Score 这五个指标上的结果，同时本文的算法参数的最优取值如下：$\beta = 0.4$，dropout=0.5，$d = 300$，$w = 6$。由表可知，本文提出的专家推荐算法在各项指标上都最优，图 5-7 更直观的展示了算法性能，由图可知，TF-IDF 效果最差，因为 TF-IDF 更主要是作为一种特征词提取方法，在计算问题、用户特征向量相似度时是基于单词匹配，而无法捕捉到任何文本语义信息；ExpertsRank，作为基于链接分析的方法效果好于 TF-IDF，因为链接分析方法基于用户的历史行为和社交关系可以发现用户-问题之间更多潜在的关联关系。主题提取模型 TSPM 效果优于链接分析法，因为主题提取模型根据文本信息可以挖掘出用户、问题的语义分布信息。在所有的实验例子中，本文提出的算法效果最优，因为首先它基于链接分析方法可以挖掘出用户-问题之间更多潜在的关联关系，同时利用深度学习模型可以学习问题的语义表示。

表 5-10 算法实验结果

Methods	MRR	NDCG	Accuracy	Recall	F1-Score
TF-IDF	0.2054	0.3874	0.0276	0.0201	0.0233
ExpertsRank	0.4715	0.6658	0.0553	0.0489	0.0520
TSPM	0.4890	0.6814	0.0843	0.0795	0.0820
ERLR	0.5892	0.7862	0.1156	0.1036	0.1093

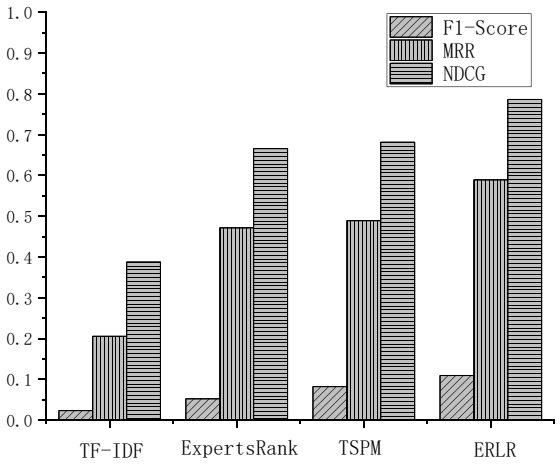


图 5-7 算法性能对比图

由本文 5.2 节对实验数据集的介绍可知，知乎上用户行为非常稀疏，同时每个问题的答案数也非常少，即使本文的算法模型引入了关注关系，并通过随机游走构造了更多的训练三元组，但这也无法从根本上解决数据稀疏性对模型性能的影响，因为即使用户是某个问题领域的专家，但是如果该用户没有回答该问题，算法也不会将其作为专家用户，导致推荐结果也不准确。解决该类问题的一种思想是探索与利用（Explore and exploit，E&E），即在利用已有的数据情况下，探索一些可能增强数据表现和模型效果的推荐结果，即给问题推荐那些可能的专家用户，以此增强用户的行为缓解问题答案等面临的稀疏性问题。

5.8 本章小结

本章通过实验验证了本文设计的标签推荐算法和专家推荐算法，并与已有的算法进行了实验对比分析。首先详细介绍了本文实验使用的知乎真实的问答数据集，然后对数据集进行分析和预处理。本章第四节设计了标签推荐算法的实验方案，包括实验软硬件环境及训练数据集的处理。第五节通过实验分析了不同的算法参数对模型效果的影响，最后通过与对比算法进行实验比较，说明本文提出的标签推荐算法在性能上有较大提升。第六节详细介绍了专家推荐算法的实验方案，然后通过实验分析算法参数对模型性能的影响，并通过实验结果选择最优的参数，最后对比了本文的专家推荐算法和已有的算法优劣，实验结果显示，本文的算法在推荐准确率、召回率、NDCG、MRR 等指标上相较于传统的方法都有提升，证明了本文的算法的有效性。

第六章 总结与展望

6.1 总结

如今在线问答社区逐渐成为人们进行网络交流、学习的平台。例如 qoura，stackoverflow，知乎等流行的问答社区每天会产生数以万计的新问题，大量的用户在社区中提问、回答问题。随着积累的问题、答案、用户数据越来越多，很多新问题无法获得高质量的答案，同时用户也越来越难以发现自己感兴趣的问题，问答社区面临的“数据过载”挑战越来越严重。基于此，本文研究如何利用推荐系统，缓解问答社区的信息过载问题。

本文第一章首先全面研究阐述了在线问答社区中不同场景下的推荐算法，分析了这些算法的理论依据、研究方法。总的来说，目前已有的方法可以分为：基于链接分析的方法，比如基于 pagerRank 思想的 CQARank, ExpertsRank；基于主题提取模型的方法，比如 RanSLDA，PLSA；基于深度学习的模型，比如基于 CNN，LSTM 网络。在第二章详细的介绍了目前已有的问答社区推荐算法和本文的算法模型所使用的一些相关理论技术，包括语言模型、概率主题模型、排序学习、链接预测分析以及深度学习模型和文本分类模型。

本文第三章提出了在线问答社区的问题标签推荐算法。在线问答社区的每个问题包含一个或多个标签，这些标签不仅概括了问题的语义信息，还反映了问题所属的话题领域，为了研究方便，本文将标签推荐定义为一个多标签的文本分类问题，从而可以借鉴文本分类的相关理论技术。基于深度学习技术在自然语言处理任务中的成功应用，本文提出了结合长短期记忆网络和卷积神经网络的标签推荐算法模型。长短期记忆网络作为循环神经网络的一种变体，可以有效的对变长序列信息进行建模，特别是对于长序列而言，可以保存历史信息，双向长短期记忆网络能够同时从两个方向对序列信息建模建模，本文正是利用双向长短期记忆网络对问题文本进行建模，提取语义信息。卷积神经网络最早应用在图像处理领域并取得了显著成果，随后研究人员将其应用到自然语言处理任务中。本文用卷积神经网络提取问题的多层次 N-gram 特征，并结合长短期记忆网络提取的语义向量构造问题的特征表示，最后通过 sigmoid 激活函数预测问题属于每个标签的概率来获得推荐结果。

本文第四章提出了专家推荐算法模型，给问答社区的新问题推荐专家用户，方便提问者邀请这些专家用户回答问题。在进行专家用户推荐时要考虑两点，一是如何计算一个用户是否是某个问题领域的专家用户，另一点是如何计算问题和用户

的匹配程度。本文基于用户回答的答案质量提出了用户的权威性计算算法，并引入双向长短期记忆网络对问题的文本信息进行语义建模，为了计算用户、问题的匹配程度，模型采用了 pairwise 排序学习方法。需要说明的一点是，第三、四章的两个算法模型在利用双向长短期记忆网络（Bi-LSTM）进行语义特征提取时，采用了层次 attention 机制，通过引入该机制，可以提升语义特征的准确度，最终提升模型性能。

本文在知乎的问题数据集上验证了提出的标签推荐和专家推荐模型的有效性和性能，并与一些已有的算法进行对比。

6.2 展望

本文研究问答社区推荐算法，重点提出了问题标签推荐和专家推荐两个算法，所采用的理论依据主要是深度学习、随机游走、排序学习。但是通过随机游走分析用户行为兴趣时只考虑了用户回答的问题和关注关系这两个因素，而没有考虑的用户个人信息和提问的问题等其他信息，如果考虑这些信息，那么可能会提高模型的效果。另一方面，深度学习作为一个非常复杂的数学模型，要通过调节大量的参数才能获得最优的效果，本文通过交叉验证的方式选择最优的参数，超参的选择都是手工完成的，一种更优的参数选择方法是利用网格搜索等方法进行自动选择。

问答社区需要解决的推荐场景众多，由于工作量的限制，本文只研究了其中两个，因此对于本课题来说，还有很多需要继续研究的地方。

致 谢

三年时光，流沙滑落到时间沙漏的底部，预示着我的学生生涯即将结束。经过三年的学习和生活，我从懵懂本科生成长为一名研究生，到了毕业这个阶段，完成毕业论文意味着硕士生涯将告一段落，此时此刻，我的脑海里回放着过去的点点滴滴，有那些取得优异成绩时的喜悦、面对失败时的落魄、从幼稚中蜕变时的成熟，这所有的一切，离不开家人、老师、朋友和同学对我的帮助、鼓励和包容。

首先，要感谢我的导师刘梦娟老师，感谢刘老师这三年来对我在学习、生活上的帮助和关心，刘老师严谨的治学态度以及对学生极其负责的责任心深深的感染了我，在读研的三年时间里，我不仅在科研学习上有了很大的进步，也在为人处世方面学习了很多道理。如果没有刘老师的严格要求和认真指导，我也不会顺利高效的完成我的毕业论文。

我还要感谢教研室的同学，三年同门情谊，情深无价。感谢大家陪我度过了三年的科研生活。永远无法忘记无数个一起开会讨论的日子、一起奋战到夜晚的时光。感谢你们对我提供的帮助、带给我的美好记忆。

我还要感谢我的室友，感谢你们对我的包容，那些夜晚卧床一起畅谈的时光是我此生此世也难以忘记的，和你们在一起三年时光里，我学会了很多知识、道理。

最后，我要感谢我的父母、哥哥和女朋友在背后对我的默默支持，以便我可以继续深造。我的家人是平凡而伟大的，感谢他们付出了劳动和时间为我提供了强有力的支持，感谢他们无条件的支持我、信任我、鼓励我。

参考文献

[1] Xue X, Jeon J, Croft W B. Retrieval models for question and answer archives[C]. International Acm Sigir Conference on Research & Development in Information Retrieval. DBLP, 2008:475-482.

[2] Parikh A P, Täckström O, Das D, et al. A Decomposable Attention Model for Natural Language Inference[J]. 2016:2249-2255.

[3] Neural attention for learning to rank questions in community question answering.

[4] Liu X, Koll M, Koll M. Finding experts in community-based question-answering services[C]. ACM International Conference on Information and Knowledge Management. ACM, 2005:315-316.

[5] Li B, King I, Lyu M R. Question routing in community question answering: putting category in its place[C]. ACM International Conference on Information and Knowledge Management. ACM, 2011:2041-2044.

[6] Zhou G, Liu K, Zhao J. Joint relevance and answer quality learning for question routing in community QA[J]. 模式识别国家重点实验室, 2012:1492-1496.

[7] Zolaktaf Z, Riahi F, Milios E, et al. Finding Expert Users in Community Question Answering[J]. Topic Models Expert Recommender, 2012:791-798.

[8] Mandal D P, Kundu D, Maiti S. Finding experts in community question answering services: A theme based query likelihood language approach[C]. Computer Engineering and Applications. IEEE, 2015:423-427.

[9] Yang L, Qiu M, Gottipati S, et al. CQArank:jointly model topics and expertise in community question answering[C]. ACM International Conference on Conference on Information & Knowledge Management. ACM, 2013:99-108.

[10] Yang B, Manandhar S. Tag-based expert recommendation in community question answering[C]. Ieee/acm International Conference on Advances in Social Networks Analysis and Mining. IEEE, 2014:960-963.

[11] Zhao Z, Zhang L, He X, et al. Expert Finding for Question Answering via Graph Regularized Matrix Completion[J]. IEEE Transactions on Knowledge & Data Engineering, 2015, 27(4):993-1004.

[12] Wang J, Sun J, Lin H, et al. Convolutional neural networks for expert recommendation in community question answering[J]. Science China, 2017, 60(11):110102.

[13] Zhao Z, Yang Q, Cai D, et al. Expert finding for community-based question answering via ranking metric network learning[C]. International Joint Conference on Artificial Intelligence. AAAI Press, 2016:3000-3006.

[14] Li B, King I. Routing questions to appropriate answerers in community question answering services[J]. 2010:1585-1588.

[15] Qiu X, Huang X. Convolutional neural tensor network architecture for community-based question answering[C]. International Conference on Artificial Intelligence. AAAI Press, 2015:1305-1311.

[16] Xu F, Ji Z, Wang B. Dual role model for question recommendation in community question answering[C]. International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2012:771-780.

[17] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.

[18] Qu M, Qiu G, He X, et al. Probabilistic question recommendation for question answering communities[C]. International Conference on World Wide Web, WWW 2009, Madrid, Spain, April. DBLP, 2009:1229-1230.

[19] Yan Z, Zhou J. A New Approach to Answerer Recommendation in Community Question Answering Services[M] Advances in Information Retrieval. Springer Berlin Heidelberg, 2012:121-132.

[20] Guo J, Xu S, Bao S, et al. Tapping on the potential of q&a community by recommending answer providers[C]. Acm Conference on Information & Knowledge Management. ACM, 2008:921-930

[21] Ni X, Lu Y, Quan X, et al. User interest modeling and its application for question recommendation in user-interactive question answering systems[J]. Information Processing & Management, 2012, 48(2):218-233.

[22] San Pedro J, Karatzoglou A. Question recommendation for collaborative question answering systems with RankSLDA[J]. 2014:193-200.

[23] Zhang J, Ackerman M S, Adamic L. Expertise networks in online communities:structure and algorithms[C]. International Conference on World Wide Web. ACM, 2007:221-230.

[24] Page L. The PageRank citation ranking : Bringing order to the web[J]. Stanford Digital Libraries Working Paper, 1998, 9(1):1-14.

[25] Yang L, Qiu M, Gottipati S, et al. CQArank:jointly model topics and expertise in community question answering[C]. ACM International Conference on Conference on Information & Knowledge Management. ACM, 2013:99-108.

[26] Sahu T P, Nagwani N K, Verma S. Selecting Best Answer: An Empirical Analysis on Community Question Answering Sites[J]. IEEE Access, 2016, 4(99):4797-4808.

[27] Ko J, Si L, Nyberg E. A Probabilistic Framework for Answer Selection in Question Answering[C]. Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA. DBLP, 2007:524-531.

[28] Wang X J, Tu X, Feng D, et al. Ranking community answers by modeling question-answer relationships via analogical reasoning[J]. 2009, 72(3):179-186.

[29] Feng M, Xiang B, Glass M R, et al. Applying deep learning to answer selection: A study and an open task[C]. Automatic Speech Recognition and Understanding. IEEE, 2016:813-820.

[30] Qiu X, Huang X. Convolutional neural tensor network architecture for community-based question answering[C]. International Conference on Artificial Intelligence. AAAI Press, 2015:1305-1311.

[31] Tan M, Santos C D, Xiang B, et al. LSTM-based Deep Learning Models for Non-factoid Answer Selection[J]. Computer Science, 2015.

[32] Rendle S, Schmidt-Thieme L. Pairwise interaction tensor factorization for personalized tag recommendation[C]. ACM International Conference on Web Search and Data Mining. ACM, 2010:81-90.

[33] Feng W, Wang J. Incorporating heterogeneous information for personalized tag recommendation in social tagging systems[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2012:1276-1284.

[34] Song Y, Zhang L, Giles C L. A sparse gaussian processes classification framework for fast tag suggestions[J]. 2008:93-102.

[35] Liu Z, Chen X, Sun M. A Simple Word Trigger Method for Social Tag Suggestion.[C]. Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John Mcintyre Conference Centre, Edinburgh, Uk, A Meeting of Sigdat, A Special Interest Group of the ACL. DBLP, 2012:1577-1588.

[36] Wu Y, Wu W, Zhang X, et al. Improving recommendation of tail tags for questions in community question answering[C]. Thirtieth AAAI Conference on Artificial Intelligence. AAAI Press, 2016:3066-3072.

[37] Stanley, C., and Byrne, M. D. Predicting tags for stackoverflow posts. In Proceedings of ICCM, volume 2013.

[38] Papadimitriou C H, Raghavan P, Tamaki H, et al. Latent Semantic Indexing: A Probabilistic

Analysis[J]. Journal of Computer and System Sciences, 2000, 61(2):217-235.

[39] Hofmann T. Probabilistic latent semantic indexing[C]. International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 1999:50-57.

[40] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. J Machine Learning Research Archive, 2003, 3:993-1022.

[41] Haveliwala T H. Topic-sensitive PageRank: a context-sensitive ranking algorithm for Web search[J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(4):784-796.

[42] Kleinberg J M. Authoritative sources in a hyperlinked environment[J]. Journal of the Acm, 1998, 46(5):604-632.

[43] Lempel R, Moran S. The stochastic approach for link-structure analysis (SALSA) and the TKC effect 1[J]. Computer Networks, 2000, 33(1):387-401.

[44] Robertson S, Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond[J]. Foundations & Trends® in Information Retrieval, 2009, 3(4):333-389.

[45] Wei C, Keerthi S S. New approaches to support vector ordinal regression[C]. International Conference on Machine Learning. ACM, 2005:145-152.

[46] Shashua A, Levin A. Ranking with Large Margin Principle: Two Approaches[J]. 2003.

[47] Harrington E F. Online ranking/collaborative filtering using the perceptron algorithm[C]. Twentieth International Conference on International Conference on Machine Learning. AAAI Press, 2003:250-257.

[48] Zheng Z, Chen K, Sun G, et al. A regression framework for learning ranking functions using relative relevance judgments[C]. International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2007:287-294.

[49] Burges C, Shaked T, Renshaw E, et al. Learning to rank using gradient descent[C]. International Conference on Machine Learning. ACM, 2005:89-96.

[50] Herbrich R. Large margin rank boundaries for ordinal regression[J]. Advances in Large Margin Classifiers, 2000, 88.

[51] Cao Z, Qin T, Liu T Y, et al. Learning to rank:from pairwise approach to listwise approach[C]. International Conference on Machine Learning. ACM, 2007:129-136.

[52] Burges C J C, Ragno R, Le Q V. Learning to rank with nonsmooth cost functions[C]. International Conference on Neural Information Processing Systems. MIT Press, 2006:193-200.

[53] Lecun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural Computation, 2014, 1(4):541-551.

[54] Hinton, Geoffrey E. Learning multiple layers of representation[J]. Trends in Cognitive Sciences,

2007, 11(10):428-434.

[55] Sutskever I, Vinyals O, Le Q V. Sequence to Sequence Learning with Neural Networks[J]. 2014, 4:3104-3112.

[56] Cho K, Merrienboer B V, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. Computer Science, 2014.

[57] Cui Y, Liu T, Chen Z, et al. Consensus Attention-based Neural Networks for Chinese Reading Comprehension[J]. 2016.

[58] Cui Y, Chen Z, Wei S, et al. Attention-over-Attention Neural Networks for Reading Comprehension[J]. 2017.

[59] Lipton Z C, Berkowitz J, Elkan C. A Critical Review of Recurrent Neural Networks for Sequence Learning[J]. Computer Science, 2015.

[60] Gers F A, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM[J]. Neural Computation, 2000, 12(10):2451.

[61] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures[J]. Neural Netw, 2005, 18(5):602-610.

[62] Hubel D H, Wiesel T N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex.[J]. J Physiol, 1962, 160(1):106-154.

[63] Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[64] Waibel A, Hanazawa T, Hinton G, et al. Phoneme recognition using time-delay neural networks[C]. Backpropagation. L. Erlbaum Associates Inc. 1995:35-61.

[65] Vaillant R, Monrocq C, Cun Y L. An original approach for the localization of objects in images[C]. International Conference on Artificial Neural Networks. IET, 1993:26-30.

[66] Lawrence S, Giles C L, Tsoi A C, et al. Face recognition: a convolutional neural-network approach[J]. IEEE Transactions on Neural Networks, 1997, 8(1):98-113.

[67] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]. International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1097-1105.

[68] Deng J, Dong W, Socher R, et al. ImageNet: A large-scale hierarchical image database[C]. Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009:248-255.

[69] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.

[70] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]. IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2015:1-9.

[71] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. 2015:770-778.

[72] Mnih V, Heess N, Graves A, et al. Recurrent models of visual attention[J]. 2014, 3:2204-2212.

[73] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. Computer Science, 2014.

[74] https://github.com/tensorflow/nmt.

[75] Bengio Y. Neural net language models[J]. Scholarpedia, 2008, 3(1).

[76] https://github.com/yanyiwu/cppjieba.

[77] https://github.com/HIT-SCIR/ltp.

[78] https://github.com/NLPIR-team/NLPIR.

[79] https://github.com/thunlp/THULAC.

[80] Salton G, Buckley C. Term-weighting approaches in automatic text retrieval ☆[M]. Pergamon Press, Inc. 1988.

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

[82]Yang Z, Yang D, Dyer C, et al. Hierarchical Attention Networks for Document Classification[C]. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016:1480-1489.

[83] Nie L, Zhao Y L, Wang X, et al. Learning to Recommend Descriptive Tags for Questions in Social Forums[J]. Acm Transactions on Information Systems, 2014, 32(1):1-23.

[84] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[J]. 2015:448-456.

[85] Kalchbrenner N, Grefenstette E, Blunsom P. A Convolutional Neural Network for Modelling Sentences[J]. Eprint Arxiv, 2014, 1.

[86] Andersen R, Fan C, Lang K. Local Graph Partitioning using PageRank Vectors[C]. IEEE Symposium on Foundations of Computer Science. IEEE Computer Society, 2006:475-486.

[87] Duchi J, Hazan E, Singer Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization[J]. Journal of Machine Learning Research, 2011, 12(7):257-269.

[88] https://biendata.com/competition/zhihu/data/.

[89] Hinton, Geoffrey. Neural networks for machine learning. Coursera, video lectures, 2012.

[90] https://github.com/MarkWuNLP/QuestionTagging.

[91] https://github.com/brightmart/text_classification.

攻读硕士学位期间取得的成果

[1] Mengjuan Liu, Wei Wang, Fan Zhou, Hao Xue, Zhiguang Qin. ActiveRec: A Novel Context-Sensitive Ranking Method for Active Movie Recommendation[C]. International Conference on Advanced Cloud and Big Data. IEEE Computer Society, 2016:92-97.

[2] 卿勇，刘梦娟，薛浩，刘冰冰，秦志光. OPEN：一个基于评论的商品特征抽取及情感分析框架[J].计算机应用与软件, 2018(1):65-71.

[3] 刘梦娟，马小栓，薛浩. 一种针对单类协同过滤问题的负样本选择方法[P]. 中国，发明权利，201710285697.5.

[4] 基于社交网络模型的视频分享关键技术研究，国家自然科学基金青年项目，61202445，2013/01-2015/12.

[5] 融合用户行为与社交关系的推荐系统研究，中央高校基本业务费项目，ZYGX2016J096，2017/01-2018/12.