计 算 机 研 究 与 发 展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202220837
61(12): 3168−3187, 2024

基于深度学习的查询建议综述

田 萱 徐泽洲 王子涵
(北京林业大学信息学院 北京 100083)
(国家林业草原林业智能信息处理工程技术研究中心(北京林业大学) 北京 100083)
(tianxuan@bjfu.edu.cn)

Review of Deep Learning Based Query Suggestion

Tian Xuan, Xu Zezhou, and Wang Zihan
(*School of Information Science and Technology*, *Beijing Forestry University*, *Beijing* 100083)
(*National Forestry Grassland Forestry Intelligent Information Processing Engineering Technology Research Center* (*Beijing Forestry University*), *Beijing* 100083)

Abstract Query suggestion (QS) is an indispensable part of search engines. It can provide query candidates before users entering a complete query to help express their information needs more accurately and more quickly. Deep learning helps to improve the accuracy of QS and it has become the mainstream technology to promote the development of QS in recent years. We mainly summarize, analyze and compare the research status of deep learning based QS (DQS). According to the different application stages of deep learning, DQS methods are divided into two categories: generative QS methods and ranking-based QS suggestion methods, and the modeling ideas of each model are analyzed. In addition, the data sets, baselines and evaluation indexes commonly used in the field of QS are introduced, and the technical characteristics and experimental results of different models are compared. Finally, the current challenges and future development trends of QS research based on deep learning are summarized.

Key words query suggestion (QS); deep learning; query auto-completion; encoder-decoder; neural language model

摘 要 查询建议是当今搜索引擎必不可少的一个组成部分,它可以在用户输入完整查询前提供查询候选项,帮助用户更准确、更快速地表达信息需求.深度学习技术有助于提升查询建议的准确度,成为近年来推动查询建议发展的主流技术.主要对基于深度学习的查询建议研究现状进行归纳整理与分析对比,根据深度学习应用阶段不同,把其分为生成式查询建议与排名式查询建议2类,分析其中每种模型的建模思路和处理特征.此外还介绍了查询建议领域常用的数据集、基线方法与评价指标,并对比其中不同模型的技术特点与实验结果.最后总结了基于深度学习的查询建议研究目前面临的挑战与未来发展趋势.

关键词 查询建议;深度学习;查询自动补全;编码器-解码器;神经语言模型

中图法分类号 TP391

随着互联网的广泛应用,用户已经习惯使用搜索引擎来满足信息检索需求.为了用户能够更高效准确获取信息,查询建议(query suggestion, QS)技术应运而生并成为搜索引擎的重要组成部分.在用户键入查询词之前,QS以搜索下拉框、推荐链接等形式为用户推荐一些完整查询词供用户选择.一方面,通过QS技术用户无需手工输入完整查询词即可轻易选择目标查询词.另一方面,当用户难以明确表达自己查询需求进行尝试性搜索时,QS可以更快更准地帮助用户表述自己的查询需求以获得满意的查询结果.

QS包括一类特殊的技术——查询自动补全

收稿日期: 2022-10-09; **修回日期**: 2024-05-11
通信作者: 田萱(tianxuan@bjfu.edu.cn)

(query auto-completion, QAC). QAC 特指在用户键入查询过程中而非键入前向用户提供查询建议. QS 与 QAC 本质上都是利用上下文信息为用户提供查询建议, 二者的区别主要体现在 3 个方面: 1)对查询建议问题建模不同. QS 将查询建议建模为在给定上下文情况下的文本生成问题; 而 QAC 中, 一些研究把前缀视作上下文的一部分, 同样以文本生成模型解决 QAC 问题; 另一些则利用前缀计算后续字符出现的条件概率, 把 QAC 建模为语言模型问题. 2)利用上下文因素略有不同. QAC 的特点决定了查询前缀是 QAC 中最核心的上下文因素; 而 QS 给出的查询建议不严格受限于前缀, 更多考虑查询会话因素. 3)呈现形式有所不同. QAC 大多以搜索下拉框形式随用户输入动态呈现; 而 QS 既可在下拉框中呈现, 也可以搜索发现、猜你喜欢等链接形式为用户提供推荐.

文献 [1] 提出了一种简单且符合直觉的 QAC 方法, 即基于最大似然估计的最流行补全法(most popular completion, MPC). MPC 依据搜索引擎中大多数用户的查询频率来推测每个具体用户的查询需求. 但该方法存在着个性化不足和长尾现象两大问题, 无法满足用户对于查询实时性与准确性的要求.

为了更好地适应各类用户的不同查询需求, 一些研究从时间、地点等多个维度将 QAC 方法进行了改良. 文献 [2] 和文献 [3] 分别对时间因素敏感的 QAC 方法和个性化 QS 方法进行了总结概括, 但这些研究大多是基于传统机器学习方法角度进行介绍, 缺乏对基于深度学习技术的 QS 和 QAC 的系统性分析.

深度学习是一种表征学习方法[4], 近年来已经广泛应用于自然语言处理领域. 通过深度学习模型的应用, QS 在感知上下文属性的同时能为用户生成个性化查询, 既保留了用户初始查询意图又能进一步多角度发现用户查询需求. 基于深度学习的 QS(deep learning based query suggestion, DQS)与 QAC(deep learning based query auto-completion, DQAC)研究与日俱增, 成为当前信息检索领域的研究热点之一, 获得了信息检索、数据挖掘和人工智能等领域国内外学者的高度关注.

如图 1 所示, DQS 过程可分为生成阶段和排名阶段 2 个步骤. 生成阶段主要是根据会话信息生成一个初始查询候选列表; 排名阶段则依据一定算法将查询候选列表进行重排序并呈现给用户供其选择. 一部分 DQS 研究将深度学习模型应用于生成阶段, 学习丰富的上下文信息, 合成不依赖日志的新查询, 以解决部分查询缺少日志导致的长尾现象. 此外, 另一部分研究将深度学习应用于排名阶段, 利用深度学习模型为候选查询的语义相关性进行打分, 根据语义相关性进行重排序, 以解决排名过程中过于依赖查询日志中附加信息的问题, 保证排序结果的可靠性. 当前 DQS 研究聚焦于利用各种模型结构建模不同维度的上下文信息. 随着大型预训练语言模型在自然语言处理领域的广泛应用, 一些研究开始将预训练语言模型应用于 DQS 领域并取得了良好效果.

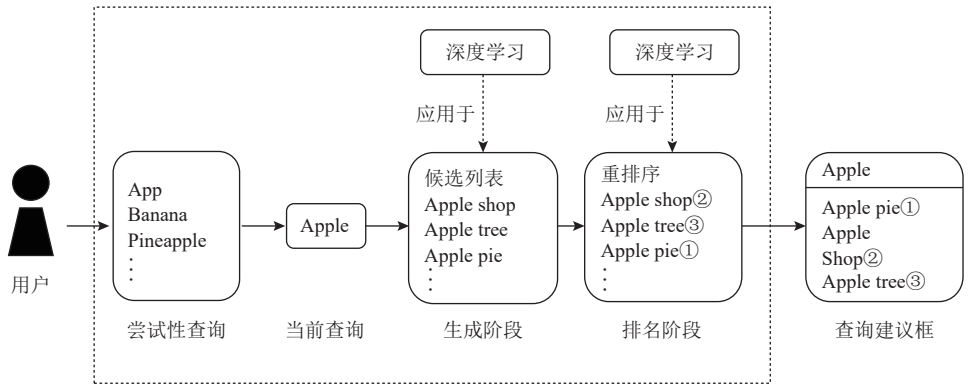

Fig. 1 Common process of DQS

图 1 DQS 的一般过程

和已有的综述文献 [2–3] 不同, 本文的主要贡献在于从深度学习技术角度全面、系统论述 DQS 研究进展, 并着重剖析这些深度学习模型在 QS 过程中对不同上下文因素的特征处理和建模特色. 依据深度学习技术的应用阶段不同, 下面将 DQS 划分为“生成式 DQS”和“排名式 DQS”2 类分别进行介绍.

1 相关背景简介

2011 年提出的基于日志共现的 MPC 是一种普

遍适用的 QS 方法，是目前 DQS 研究领域的主流基线算法. 该方法定义为：

$$Score_{\text{popularity}}(q)=\frac{f(q)}{\sum_{q'\in Q}f(q')}. \tag{1}$$

当 QAC 模块接收到某个查询前缀，MPC 会以该前缀为索引，在搜索日志Q中搜寻以该前缀开头的所有查询记录q'，$f(q)$表示查询q在Q中出现的总次数. 然后 MPC 将流行度得分$Score_{\text{popularity}}(q)$最高的若干个查询补全从高到低呈现给用户，供用户选择.

然而 MPC 仅考虑了流行度这一上下文因素，缺乏对查询会话中更多上下文信息的深入挖掘. 随后的 QS 研究中，研究者将不同维度上下文因素融入到 QS/QAC 生成阶段和排名阶段中. 下面对这些上下文因素进行归纳介绍.

1.1 上下文因素简介

用户在使用搜索引擎进行查询时，其查询的时间、地点等背景信息与输入的查询词、点击的链接等会话信息共同构成了查询的上下文. 已有研究中常用的上下文因素分类如图 2 所示. 鉴于用户发起查询的时间、地点及查询会话中的用户行为具有个性化特征，我们将时间因素、会话因素、地点因素总结为个性化上下文因素. 而查询在全体用户中的流行度和参与查询构成的查询前缀则是共性的，因此统一归纳于非个性化上下文因素.

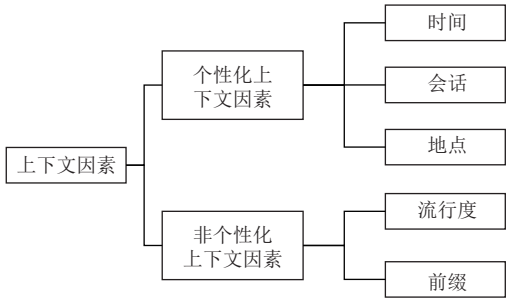

Fig. 2 Contextual factor classification

图 2 上下文因素分类

DQS 研究往往根据应用场景，将个性化上下文与非个性化上下文有机结合从而获得更好的用户反馈. 通过模型学习不同的上下文因素并赋予上下文因素不同的权重. DQS 可以由式(2)定义.

$$Score(q)=\arg\max\left(\sum_{i\in I}\alpha_i Score_i(q)\right), \tag{2}$$

$$I=\left\{I_{\text{popularity}},I_{\text{prefix}},I_{\text{context}},I_{\text{time}},I_{\text{location}}\right\}, \tag{3}$$

其中，I表示不同上下文因素的集合，如式(3)所示. 式(2)中$Score_i(q)$表示 DQS 模型对不同上下文因素的评分；α_i表示模型为不同上下文因素赋予的不同权重.

下面将分别介绍这些上下文因素.

1)时间因素. 在使用搜索引擎过程中，一些查询的出现频率与时间因素有很大的关系. 时间因素包含周期性因素和突发性因素. 周期性因素是指在长时间段内规律性重复的事件，例如与四季变化、每年法定节日有关的事件. 突发性因素是指短时间内搜索量急剧上升的事件，例如突发新闻等. 一个时间敏感的 QS 系统应当对上述特殊事件有所响应.

2)会话因素. 搜索意图产生后，会在搜索引擎上产生一系列查询和点击行为，在一个时间段内，用户的这些行为集合被叫做会话. 基于会话因素的 DQS 系统可以将一段时间窗口内的用户行为作为个性化依据，使得查询建议的结果更符合该用户本次查询中的意图.

3)地点因素. 查询意图同样与查询发起地点息息相关，例如用户对于附近餐馆的搜索等. 基于地点信息因素的 QS 主要采用了 2 种方式来挖掘地点信息. 第 1 种是使用包含位置标签的数据集[5-6]，第 2 种是从数据集中提取位置信息[7-8].

4)流行度因素. QS 研究中的流行度因素由文献 [1] 首先提出，它是 QS 研究中重要的非个性化上下文因素，在传统的个性化 QS 研究[9-10] 中，首先基于流行度得到初步的查询建议候选列表，随后通过一些预定义的个性化特征调整候选查询建议的排名. 在 DQS 研究中，也经常会将基于流行度的查询建议与基于深度模型的查询建议进行有机结合从而兼顾查询建议的可靠性与丰富度.

5)查询前缀. 用户键入的不完整查询字符串称为查询前缀. 基于搜索日志的流行度方法往往只把查询前缀当作一个搜索日志的索引使用，忽略了它们作为查询一部分的意义. 针对这一问题，文献 [11–13] 通过神经语言模型挖掘查询前缀中的语义信息，增强了模型对于罕见查询的响应能力.

1.2 DQS 方法的分类

DQS 的过程一般具有先生成后排序的 2 阶段特征. 根据深度学习模型在 DQS 过程中应用阶段不同，我们将 DQS 方法分为生成式 DQS 方法与排名式 DQS 方法两大类，如图 3 所示.

根据深度学习模型的不同，生成式 DQS 方法可分为基于神经语言模型的 DQS 方法、基于编码器-解码器模型的 DQS 方法和基于大型预训练语言模型的方法. 其中基于神经语言模型的方法将 QS 问题建模为字符级语言模型，根据有限的查询前缀利用神经

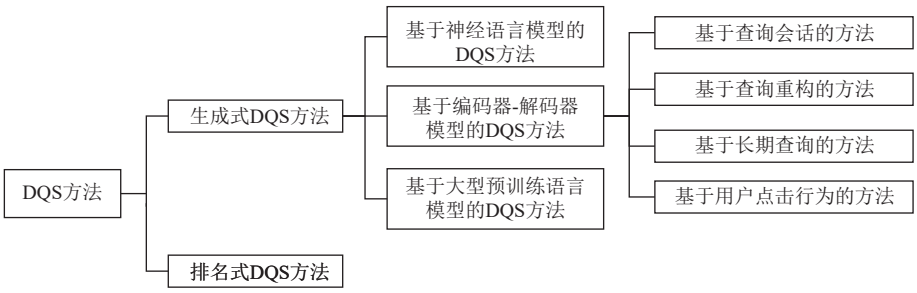

Fig. 3 The classification of DQS methods

图 3 DQS 方法分类

网络推理后续字母出现的概率. 而基于编码器-解码器模型的方法将 DQS 建模为一个序列到序列的文本生成问题, 通过对上下文因素的学习提供查询建议. 下面将按照上述分类对其进行详细介绍和分析.

2 生成式 DQS 方法

传统基于共现的 QS 方法主要依据查询在日志中的出现频率给出查询建议, 对于日志中罕见的查询无法给出有效的查询建议. 而生成式 DQS 利用神经网络模型提取当前会话中丰富的上下文信息, 经由模型直接生成不依赖日志的查询建议. 生成式 DQS 方法相较于传统方法, 对长尾查询有更好的响应, 在实际应用中能为基于日志的查询建议提供可靠的补充. 根据生成式 DQS 的模型结构不同, 本节将其分为基于神经语言模型的 DQS 方法、基于编码器-解码器模型的 DQS 方法和基于大型预训练语言模型的 DQS 方法 3 类进行介绍.

2.1 基于神经语言模型的 DQS 方法

神经语言模型是一种典型的文本生成模型, 将神经语言模型应用于 DQS 领域可以有效地提升模型对罕见查询的响应. QAC 作为 QS 的子领域, 同样存在长尾问题, 该问题在 QAC 领域表现为: 基于共现的 QAC 方法仅能为日志中存在的查询前缀输出补全, 缺乏对罕见的查询前缀的响应能力. QAC 依据前缀字符串补全后续字符的过程与利用神经语言模型利用部分文本生成完整句子的过程十分吻合, 因此一部分研究利用基于循环神经网络(recurrent neural networks, RNN)的神经语言模型来解决 DQAC 中的长尾问题.

RNN 是一种主流深度学习模型, 因其递归处理历史信息和建模历史记忆的特点, 常被用于处理语音、语言等序列化数据. 长短期记忆(long short-term memory, LSTM)网络和门控循环单元(gate recurrent unit, GRU)网络作为改进的 RNN, 解决了 RNN 中的长距离依赖问题. 基于 LSTM 和 GRU 的神经语言模型本质上是根据当前查询已键入的前缀字符串计算下一字符的可能性. 理论上, 普通模型构建完整查询需要遍历所有可能的字母组合, 但由于这种做法效率低下, 基于神经语言模型的 DQS 在生成阶段常用 beam search 的方法来寻找概率较高的字母组合构成查询项, 在排名阶段则依据上一步中字母组合出现概率的高低对候选项进行排名.

传统 QAC 方法难以处理罕见的前缀, 在查询日志较少的小型搜索引擎中基于流行度的 QAC 方法就失去了可靠性. 为了解决这一问题, 文献 [11] 将神经语言模型应用于 QAC 问题中, 提出了 NQLM(neural language model for QAC)模型, 该模型的结构如图 4 所示. 为了将字符信息与单词信息联合起来预测字符级文本, 该模型将每个查询中的字符映射到向量中, 并在每个完整单词的末尾提取词级的嵌入信息. 把二者连接后传递到双层 LSTM 中, 每个 LSTM 单元的输出上采用了一个 dropout 层来避免过拟合问题, 最后通过 softmax 函数输出字符的概率. 在得到计算补全字符可能性的语言模型后, 很自然地就能利用它生成候选查询. 该方法将神经语言模型应用于 QS 生成阶段, 相较于传统方法显著提升了 QS 对长尾查询前缀的响应能力.

文献 [13] 模型与文献 [11] 模型结构相似, 但对于前缀的编码方式有所不同. 它采用了标准的字符级语言模型, 采用独热编码将前缀字符输入到 256~1 024 个隐藏单元的双层 LSTM 中. 然后将隐藏层的输出经由 softmax 层输出下一个字符的概率, 最后在 beam search 期间计算概率并依据概率高低生成候选列表. 文献 [13] 的模型主要特点在于将神经语言模型与纠错框架相结合从而生成可以纠错的查询补全, 为实际应用中查询前缀的纠错提出了解决方案. 同时针对文献 [11] 中所忽略的效率问题, 该模型优化了

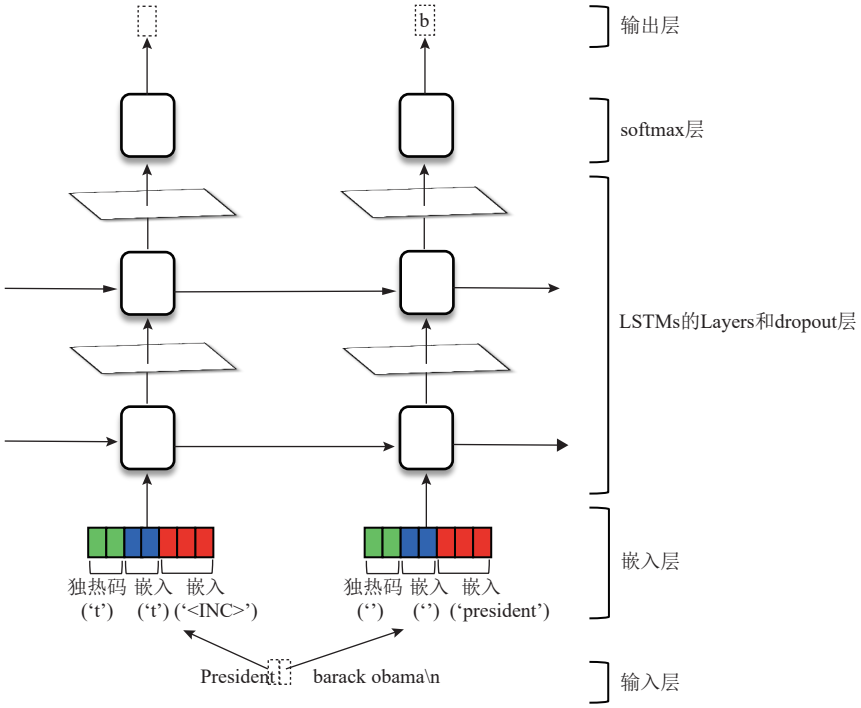

Fig. 4 NQLM model architecture

图 4 NQLM 模型结构

LSTM 的向前传播过程从而提升了查询补全的效率.

上述基于神经语言模型的 DQS 挖掘了查询前缀中字符级与词级的语义信息，但是缺乏对于个性化上下文因素的建模. 为了提高 DQS 个性化程度，文献 [14] 提出了一种考虑会话因素的个性化神经语言模型. 该模型不仅学习了查询前缀中字符级的嵌入，而且学习表示了用户长期的查询历史(即用户嵌入). 文献 [14] 的研究没有将用户嵌入作为字符嵌入的附加输入，而是采用文献 [15] 中的 FactorCell 方法，通过不同用户的查询嵌入表示控制神经语言模型中循环层权重矩阵的因子变换. 由于每个用户的 FactorCell 权重矩阵不同，所以最终经由神经语言模型生成个性化的查询候选项. 通过应用 FactorCell，该模型对训练中未出现的新用户也有具有较好的响应能力.

与基于 LSTM 语言模型的 DQS 不同，文献 [12] 提出了一种基于 GRU 的神经语言模型 NQAC(neural query auto completion). NQAC 同样从个性化角度出发，融入会话和时间 2 种个性化上下文因素. 对于会话因素，模型考虑了用户长期查询的单词并集，将每位用户的查询词汇表表示为用户向量. 对于时间因素，受文献 [16] 对查询时间因素建模的启发，NQAC 对查询发出的具体小时、分钟、秒进行编码，得到时间向量. 该模型结构如图 5 所示，NQAC 在嵌入策略上与文献 [11] 不同，它没有采用字符级的嵌入，而是将用户向量、时间向量与词级的串联拼接，送入双层 GRU 中，通过 softmax 层输出下一个字母的出现概率. 在查询候选的生成阶段，文献 [12] 研究受文献 [17] 的启发采用改进的 beam search 方法，以确保查询补全结果多样性. 与之前基于 LSTM 的 QAC 方法[11] 相比，基于 GRU 的 NQAC 对于罕见查询同样有较好的响应. 由于 GRU 相对于 LSTM 所需学习的参数更少，NQAC 计算的复杂度更低、效率更高，因此 NQAC 在专业系统开发中具有更好的扩展性.

上述基于神经语言模型的方法更偏重利用上下文完善用户的输入，但是在专业信息平台中，部分用户无法正确表达专业术语，导致无法检索到相关内容. 帮助用户提炼关键术语可能是一种更行之有效的查询建议方案. 2023 年文献 [18] 改进了 CBoW(continuous bag of words)模型，利用科技文献中的关键词生成关键词嵌入，为用户补全语义相关的文献关键词，帮助用户获取目标文献. 在类似的专业领域信息检索中，这种策略显然更有益于提升领域查询补全效果.

2.2 基于编码器-解码器模型的 DQS 方法

受制于模型结构较为单一的特点，基于神经语言模型的 DQS 方法往往难以捕获查询会话中更丰富广泛的上下文信息，导致模型准确性较低. 针对这一问题，另一些研究将 DQS 建模为序列到序列(sequence

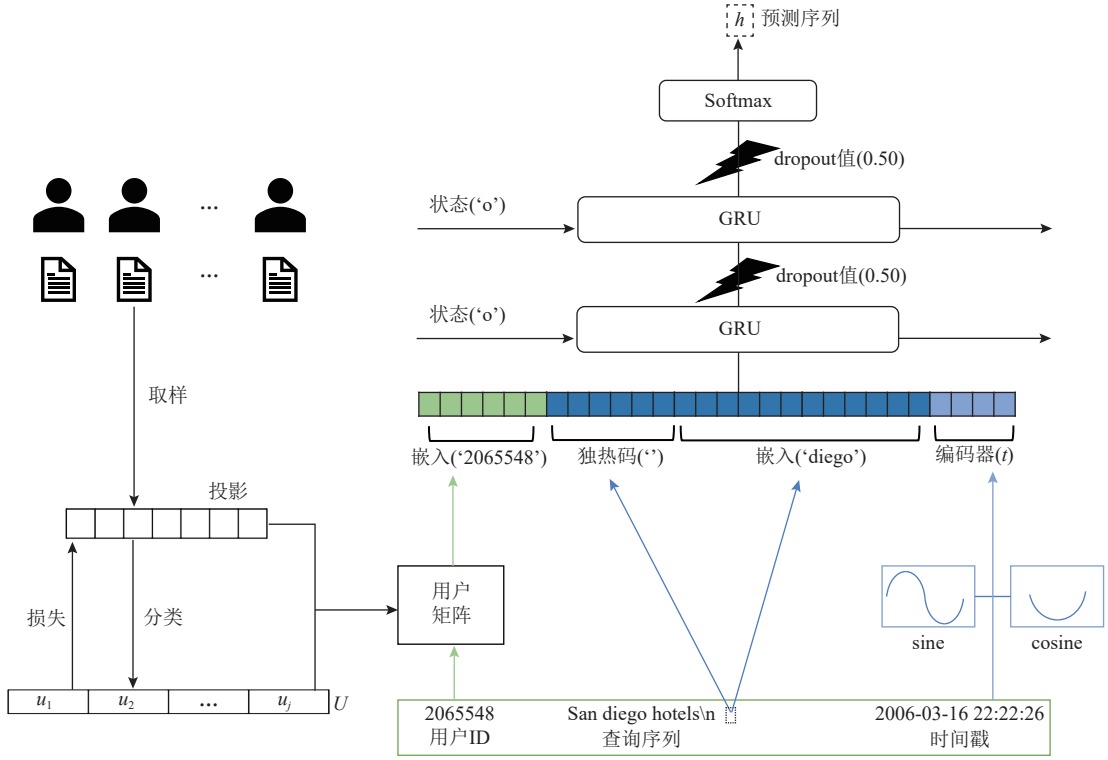

Fig. 5 NQAC model architecture

图 5 NQAC 模型结构

to sequence, Seq2Seq)问题，将编码器-解码器模型应用于 DQS 研究中. 目前编码器-解码器模型已被广泛应用于文本翻译、问答系统等 Seq2Seq 研究领域. 由于该模型能够捕获查询会话中更全面的上下文信息[19]，能为用户提供更加个性化的查询建议，近年来基于编码器-解码器模型的 DQS 成为了 QS 中主流研究方向.

会话上下文作为一种个性化上下文因素，在基于编码器-解码器模型的 DQS 研究中被进一步细分为查询会话、查询重构、用户点击行为、用户长期查询这 4 个子维度. 我们以一次用户真实的搜索苹果派食谱的全过程为例来说明上述子维度，如图 6 所示，用户一开始先后输入查询 app, apple 和 apple pie, 在每一次查询之后访问了若干页面，最后获得了满意的搜索结果. 图 6 中的多次查询内容共同构成了查询会话；每次搜索之后对于查询内容的修改构成了查询重构；用户每次搜索后访问的网页(文档)共同构成了用户点击行为；用户以往可能搜索过的其他内容则属于用户的长期查询历史.

在基于编码器-解码器模型的 DQS 方法中，多数方法是将查询会话送入编码器中，采用不同的编码策略对查询会话的各个细分维度进行嵌入，从而得到了不同维度个性化的查询建议. 根据编码策略的不同，本节将基于编码器-解码器模型的 DQS 方法分为基于查询会话的方法、基于查询重构的方法、基于长期查询的方法和基于用户点击行为的方法 4 类进行分析阐述.

2.2.1 基于查询会话的方法

查询会话包含了一个短时间窗口内用户多次查询内容，蕴含丰富的上下文信息，为了从中提炼出用户的真实查询意图，一些研究提出基于查询会话的 DQS 方法来建模一系列用户查询. 查询会话同时具有词级、会话级信息，为了从不同层次捕捉这些特征，对查询会话的建模往往采用层次化模型结构，查询会话由查询编码器和会话编码器组成. 其中较低层的查询编码器对单个查询进行建模，较高层的会话编码器则利用查询编码器的输出学习总结会话中的所有查询. 通过这种设计，模型保留了更多上下文信息，提升了 QS 对会话上下文的敏感性.

2015 年文献 [20] 提出一种基于编码器-解码器的层次化 DQS 模型 HRED(hierarchical recurrent encoder-decoder). HRED 使用 GRU 实现了 2 个编码器(查询编码器和会话编码器)和 1 个解码器，其结构如图 7 所示. 训练过程中，HRED 首先将用户的每个查询通过查询编码器进行编码，同时更新会话编码器的相应状态参数，使其适应训练集中的下一个查询并对

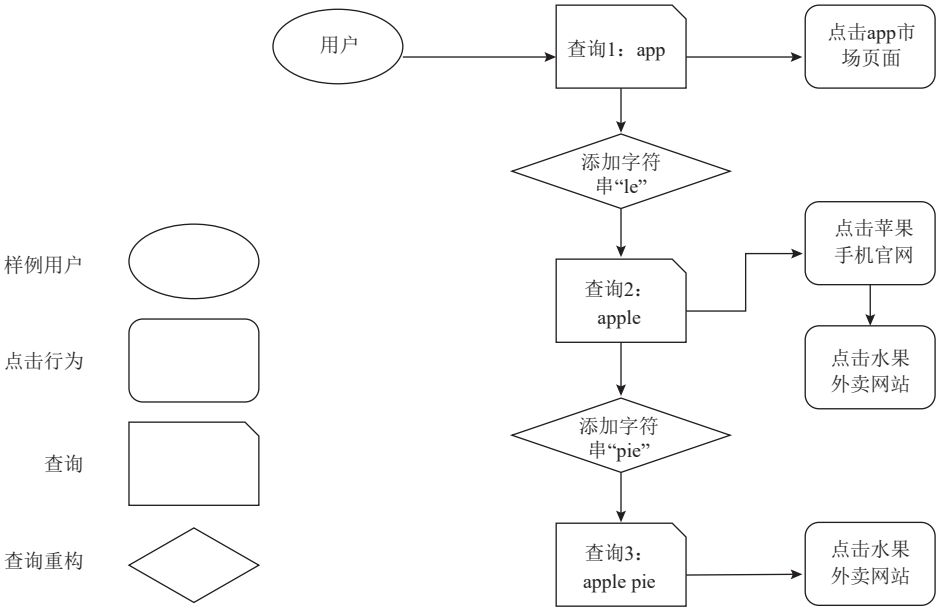

Fig. 6 Real search process of a user

图 6 某位用户真实的搜索过程

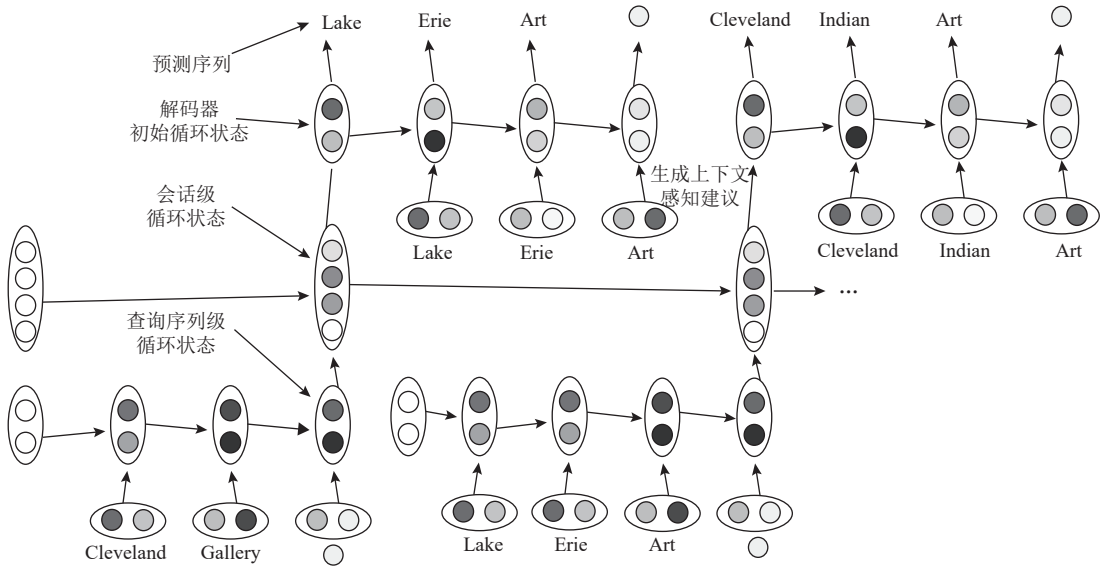

Fig. 7 HRED model architecture

图 7 HRED 模型结构

会话中所有查询不断重复这一过程. 在生成阶段，HRED 首先通过查询编码器将用户查询序列进行词级的嵌入表示，并同时更新会话编码器的状态，最后通过对会话编码器最终状态进行采样生成查询建议. HRED 是第一个生成式的 DQS 模型，这种层次化的模型结构为后续研究提供了一种标准化编码流程.

文献 [21] 在 HRED[20] 的基础上提出了一个多任务学习框架 M-NSRF(multi-task neural session relevance framework)，通过对文档排序和查询建议联合训练，该模型体系结构如图 8 所示，其中查询建议组件结构与 HRED 类似，并将文献 [20] 中的词级编码器改为具有最大池化层的双向 LSTM[22] 实现，从而提升了整体编码性能. 该研究的创新之处在于将查询建议组件的词级编码器和会话编码器的循环状态与文档排序组件共享，同时完成 QS 任务和文档排序任务. 这种多任务学习同时优化多组件的方法，为后续的多维度个性化 DQS 研究提供了一种思路.

在查询会话中，并不是所有查询都能代表用户的查询意图，一些尝试性查询往往会成为 QS 研究中的噪声查询，利用注意力机制能够较好学习不同查询的重

要程度以及克服噪声查询干扰. 文献 [23] 提出了一种应用了注意力机制的 DQS 模型 HCARNN(hierarchical contextual attention recurrent neural network)，该模型体系结构如图 9 所示，该模型同样是由查询编码器、会话编码器与解码器构成的层次化模型. 通过在会话编码器中应用注意力机制，该模型在捕获会话上下文的同时为会话中不同的查询分配权重，从而生成抗噪声的查询建议. 通过在层次化模型中引入注意力机制，该模型在地图软件的表现中取得了显著提升. 同时该模型具有通用性，可以迁移到其他类型的应用程序中.

在一些具体网站的检索系统中，用户往往具有标签特征，文献 [24] 提出了一种将用户标签和 Seq2Seq 模型结合的 DQS 策略，得到的模型没有应用层次化的查询会话建模方法，而是遵循文献 [25] 将 DQS 问题建模为一个机器翻译问题进行训练. 将查询会话通过具有注意力机制的编码器进行词级嵌入，最终通过解码器输出查询建议. 该文献针对求职网站上

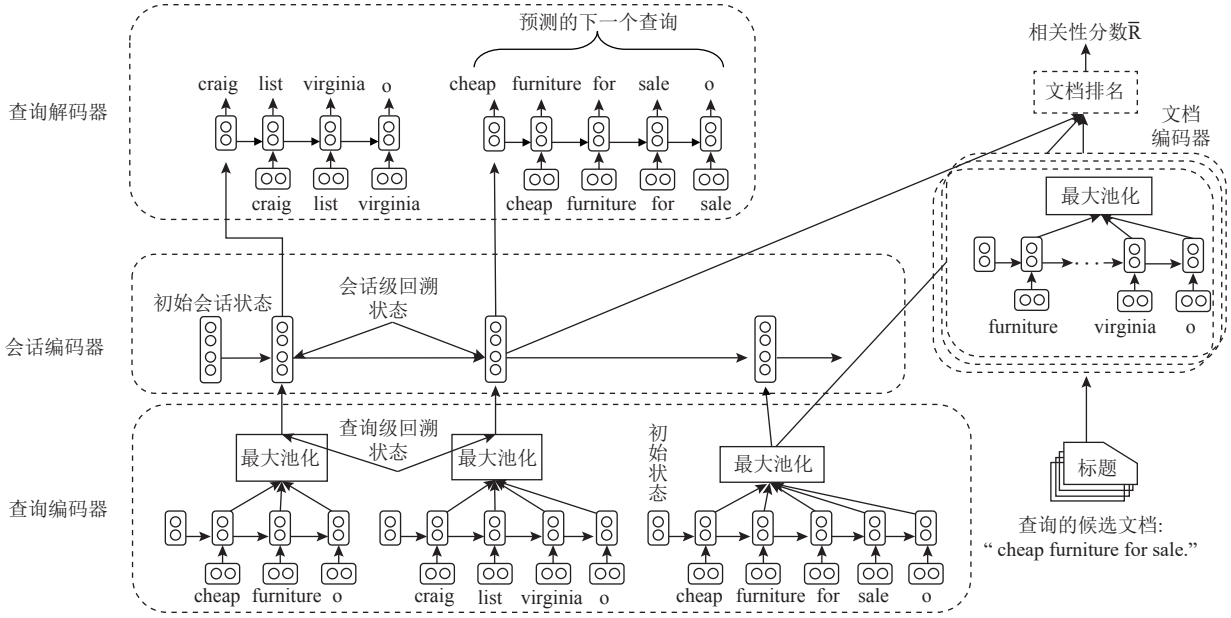

Fig. 8 M-NSRF model architecture

图 8 M-NSRF 模型结构

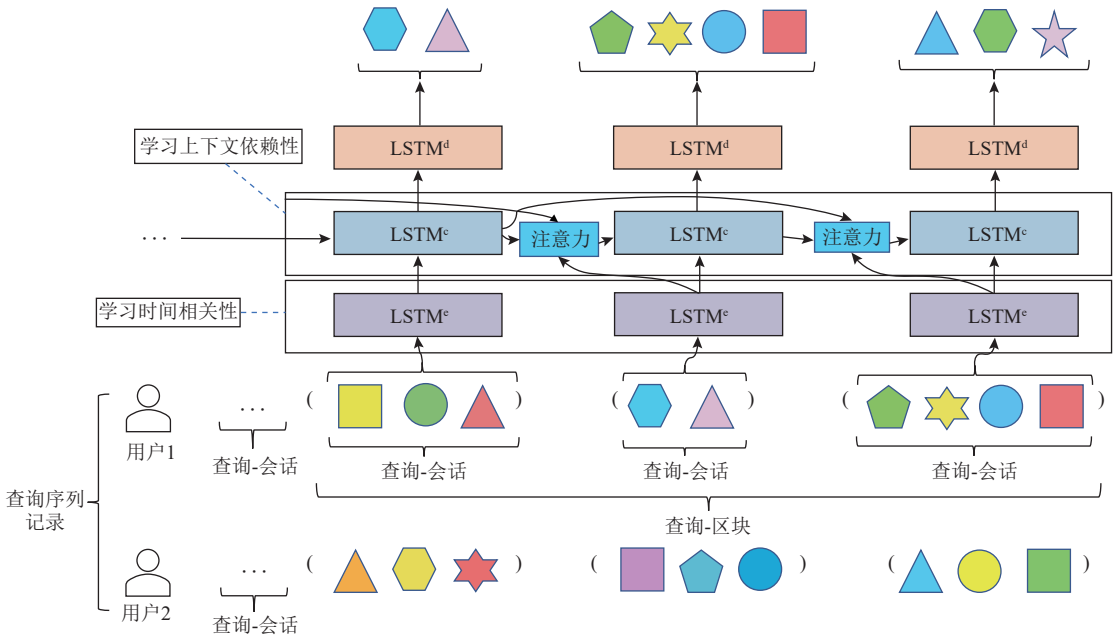

Fig. 9 HCARNN model architecture

图 9 HCARNN 模型结构

用户搜索职位的情景,将求职网站的用户标签特征整合到 DQS 模型中,提出了 2 种新的个性化编码策略附加词汇表和嵌入串联. 1)附加词汇表. 将用户标签特征附加在源查询的开头充当附加单词,与查询单词一起训练. 2)嵌入串联. 将同类别用户标签特征嵌入链接到该用户查询每个词语的开头. 这 2 种策略从用户标签的角度出发,有效提升了用户在该求职网站上查询建议的个性化程度,并可较为容易地迁移到其他具有用户标签特征的检索系统中.

2.2.2 基于查询重构的方法

当会话中存在多次查询时,查询之间一定存在对查询字符串的修改,我们把这些修改行为称为查询重构. 2.2.1 节中基于查询会话的方法偏重从整体上建模用户的意图,没有考虑查询之间的修改中蕴含着用户查询意图的转换. 而基于查询重构的方法着重于研究用户每次查询前对上一次查询字符串的增加、删除、修改这些重构操作. 查询重构的研究在整体建模查询会话基础上,更精细地建模了用户每次查询间的意图转化.

文献 [26] 以文献 [27] 中的机器翻译模型为基础,提出了一个具有 ACG(attend, copy and generate)机制的 DQS 模型,其模型结构如图 10 所示. 在编码器部分,该模型采用了分层注意力机制建模查询会话,为会话中的查询和查询中的单词分别赋予不同权重,得到会话级加权表示. 在解码器部分,该模型不同于以往 DQS 模型直接通过解码器生成新查询,而是将上一步编码器中的会话级加权表示,通过 ACG 机制最大化保留查询重构中有意义的词汇,再生成少量新词汇与之组合构成新查询建议. 在较长的查询会话中,得以保留的词汇往往具有代表性,因此 ACG 机制能够捕获长会话中的主题转换,弥补了 HRED 在长会话中性能不足的缺陷.

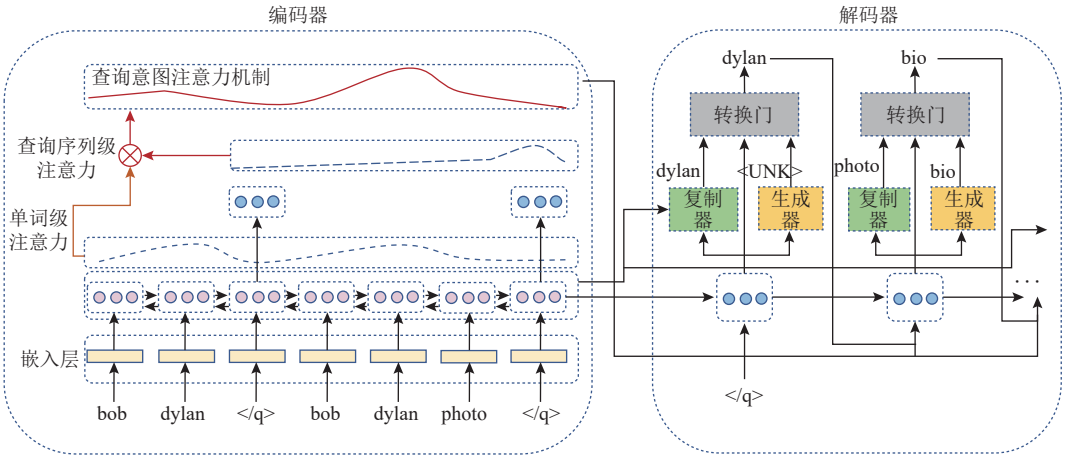

Fig. 10 ACG based DQS model architecture

图 10 基于 ACG 的 DQS 模型结构

文献 [26] 考虑到了查询重构的特点,但是缺乏对于查询重构过程的直接建模. 文献 [28] 提出了一种对查询重构本身进行表征学习的 DQS 模型 RIN(reformulation inference network),其模型结构如图 11 所示由层次化编码器、重构推理器、查询判别器、查询生成器 4 部分构成. 其中层次化编码器在以往研究中的双层编码器基础上加入了查询重构嵌入层,结合会话级注意力机制得到对查询重构敏感的加权表示. RIN 其余 3 部分则利用这种加权表示,分别完成对查询建议的生成、打分排序以及对下一次重构的预测. 最后, RIN 采用多任务学习联合优化上述 4 个组件. 通过对查询重构的多任务学习, RIN 在重构行为类似的查询中能够利用这部分知识生成新查询. 这种方法更细致地建模了查询之间的意图转化,从而在一定程度上解决了短会话中查询数据稀疏的问题.

2.2.3 基于长期查询的方法

当会话的时间窗口较长时,同一位用户在发起不同查询时会表现出一定共性,为了利用这种共性特征为用户提供符合其长期习惯的查询建议,一些研究通过对同一名用户较长时间窗口内的查询会话建模,使得 QS 结果更加符合该用户的长期习惯.

文献 [29] 提出了一种关注用户长期查询习惯的层次化 DQS 模型 AHNQS(attention-based hierarchical neural query suggestion),该模型学习了用户长期行为,其体系结构如图 12 所示. AHNQS 包含 2 部分:具有注意力机制的会话级 RNN 和用户级 RNN. 会话级 RNN 参考文献 [20] 中的方法,对用户的短期查询进行建模并利用注意力机制捕获用户短期偏好. 而用

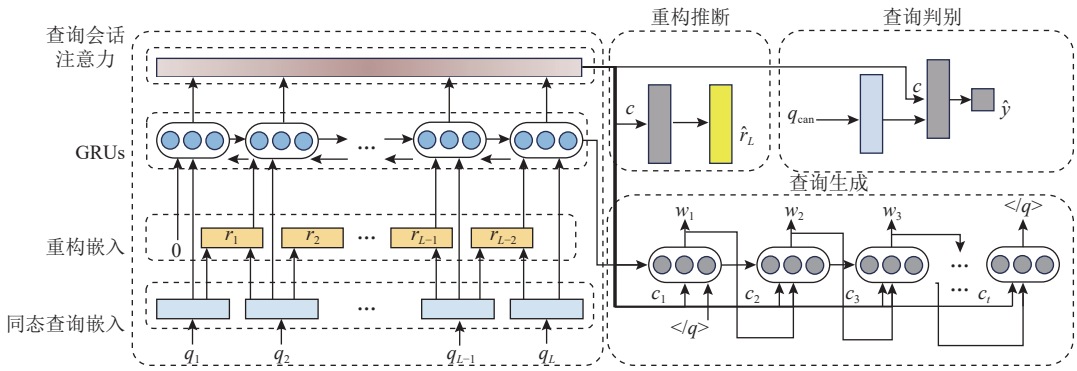

Fig. 11 RIN model architecture

图 11 RIN 模型结构

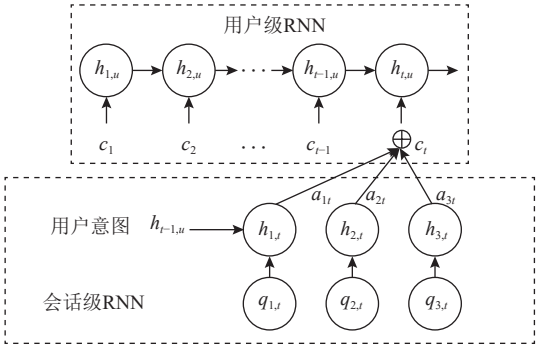

Fig. 12 AHNQS model architecture

图 12 AHNQS 模型结构

户级 RNN 则利用会话级 RNN 的隐藏状态，对用户长期查询行为进行建模. 这种设计使 AHNQS 能利用长期查询建模用户的偏好，在查询上下文较少的短会话中也能为用户提供高质量的查询建议.

2.2.4 基于用户点击行为的方法

文献 [20] 方法对会话上下文的研究往往聚焦于查询内容本身，而缺乏对用户查询后相关行为的建模，一些 DQS 研究对用户点击行为进行了挖掘，通过用户在查询会话中点击的查询结果反推用户的查询意图. 事实上，在查询过程中，用户在获得满意搜索结果前，往往会进行一系列的尝试性搜索并产生点击行为，虽然这些点击行为与目标搜索可能存在着偏差，但其作为查询会话的一部分，仍具有丰富的信息.

文献 [30] 提出了一种反馈记忆网络（feedback memory network，FMN）模型，该模型体系结构如图 13 所示. FMN 同时建模了用户点击行为和跳过行为，将用户点击的文档视为正反馈文档，跳过的文档视为负反馈文档，利用具有注意力机制的文档编码器和文档位置对文档的内容信息和排列位置进行建模，得到个性化的反馈记忆. 随后将这种反馈记忆集成到文献 [20] 中 HRED 的 Seq2Seq 模型中，生成对用户点击行为敏感的查询建议. 这种模型有效建模了用户点击行为中反映的查询偏好，提升了 HRED 模型的个性化程度.

将用户点击行为作为查询内容的一部分，并与查询在同一层次共同建模也是一种行之有效的方案. 文献 [31] 提出了一种层次化编码用户点击行为的模型 HAN（hierarchical attention network），该模型体系结构如图 14 所示. 其层次化结构与 HRED[20] 类似，并在词级编码器与会话编码器上分别应用注意力机制，即利用词级注意力机制寻找会话中最能代表信息需求的词汇以及利用会话级注意力机制克服噪声查询的干扰，并通过串联嵌入查询与点击文档的方式建模了用户查询后的点击行为，获得了比 HRED 更好的性能.

与上文介绍的查询会话建模方式类似，对于用户点击行为也可以采取分层建模方法. 文献 [32] 在文献 [21] 的基础上，提出了一种新模型 CARS（context attentive document ranking and query suggestion）. 该模型结构如图 15 所示，利用注意力机制对用户查询和点击行为同时进行分级建模. 在低层词级编码器中，查询与被点击的文档信息被转换为向量. 高层会话级编码器则利用这些向量，动态地总结会话中的上下文信息. 最后，查询建议器（即解码器）利用 2 类编码器学习到的嵌入表示可以为用户生成新查询建议. 此外，CARS 模型还利用多任务学习的手段将查询建议器和文档排序器 2 部分进行联合优化，成功建模了查询建议与文档排序 2 个相关联任务之间的关联性，为后续研究提供了新方向.

文献 [33] 将 Transformer 模型应用到 DQAC 领域，将查询前缀与查询会话、用户点击行为这些上下文因素有机结合起来，提出了 M^2A（multi-view multi-task attentive learning framework of discriminative and gener-

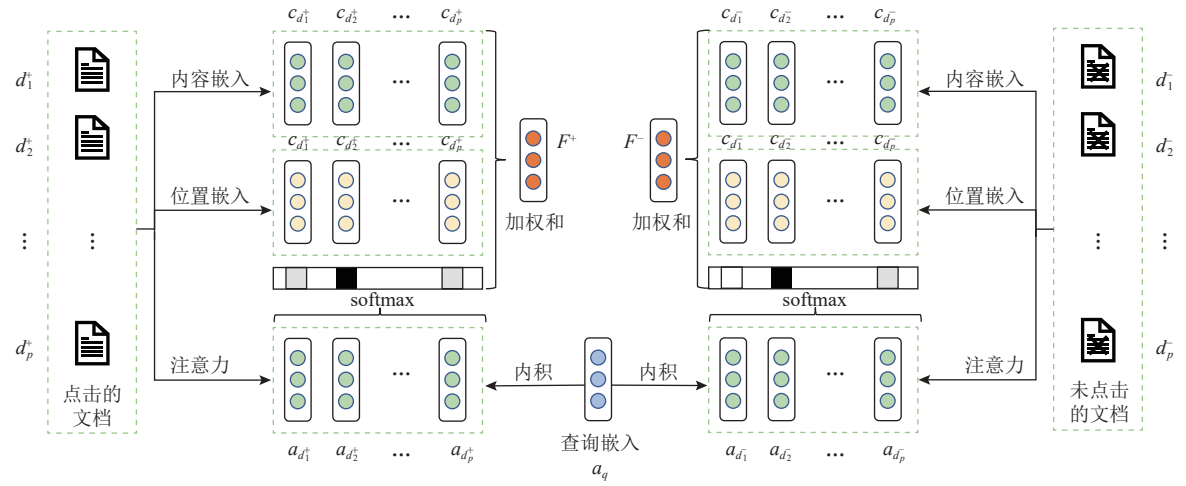

Fig. 13 FMN model architecture

图 13 FMN 模型结构

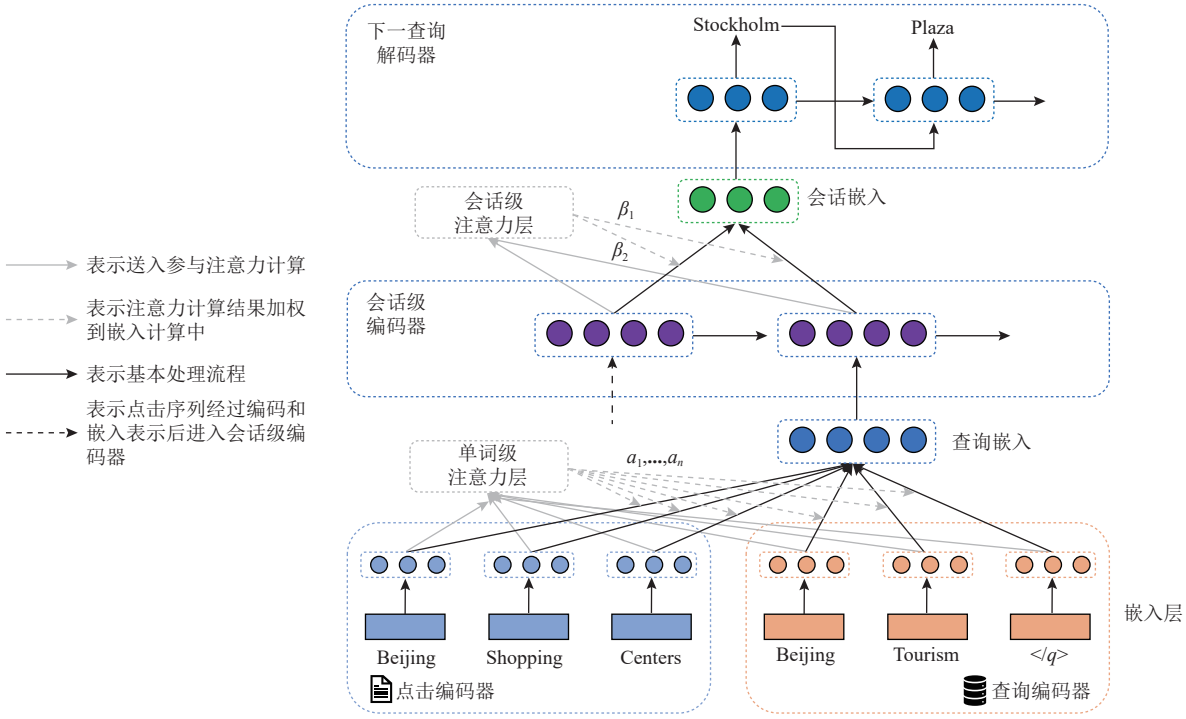

Fig. 14 HAN model architecture

图 14 HAN 模型结构

ative query auto-completion model)模型，该模型结构如图 16 所示. 其中的编码器模块由查询前缀编码器和用户行为编码器组成. 首先查询前缀编码器将用户键入的查询前缀通过带有多头池化(multi-head pooling)层的词级编码器获得前缀表示；用户行为编码器则将用户查询与查询后的点击项标题组合为一次用户行为，通过带有多头池化层的词级编码器获得用户行为向量. 然后将这些用户行为向量序列送入带有自注意力机制的会话级编码器，获得上下文感知的用户表示. 最后将前缀表示加权与用户表示加权结合起来，送入查询解码器和点击率(click through rate, CTR)预测器中. 该模型在不同组件之间共享多视图编码器，以多任务学习方式同时优化解码器与 CTR 预测器. 在生成阶段利用解码器输出完整查询建议，排名阶段利用 CTR 预测器对生成的查询建议预估其点击率高低并据此进行排名，利用 2 个不同组件的联合学习自然地完成了 DQS2 大阶段的任务. 该模型利用了比以往 DQS 更多维度的上下文信息实现了更优秀的性能，在淘宝数据集上取得了良好的效果.

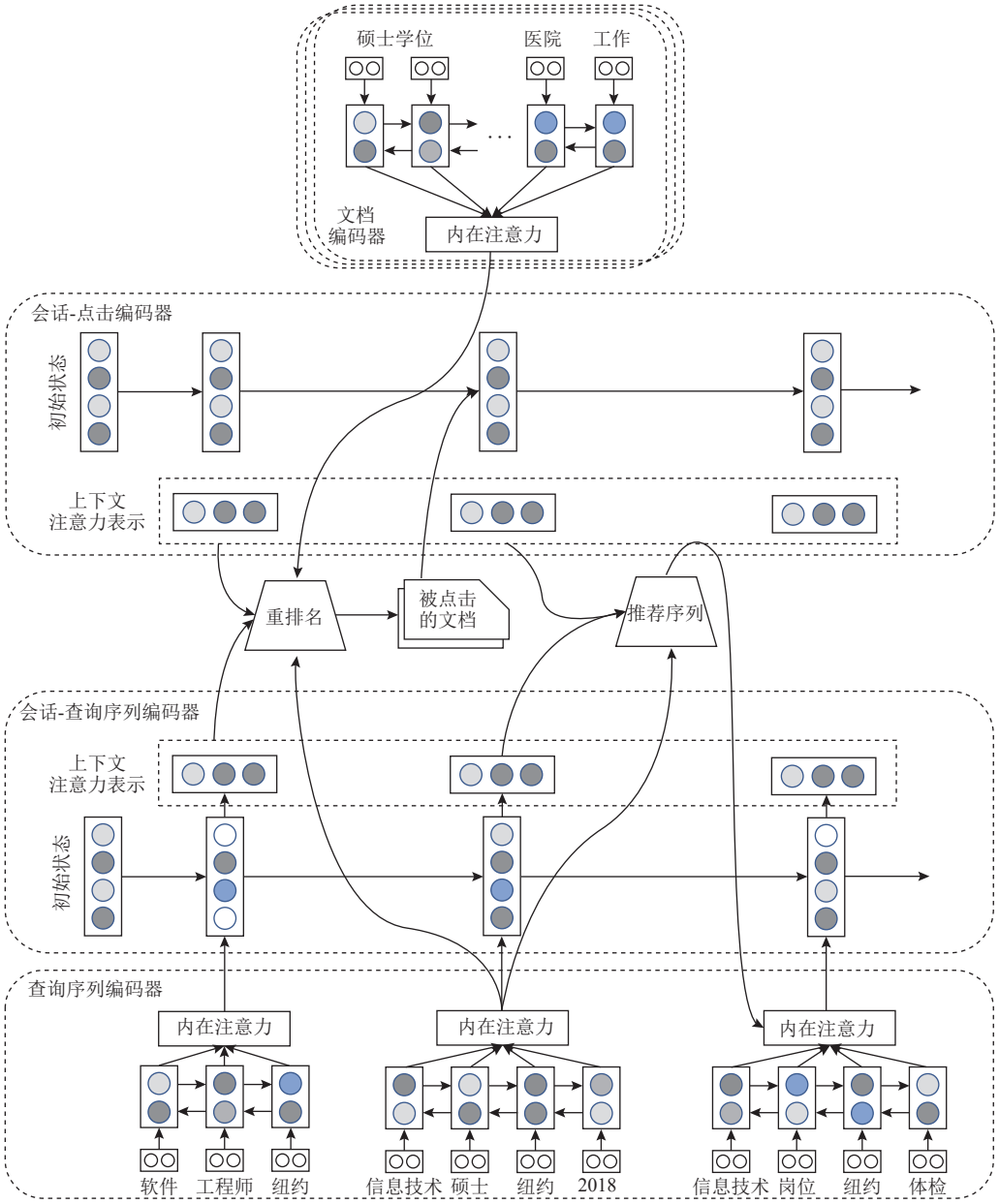

Fig. 15 CARS model architecture

图 15 CARS 模型结构

在绝大多数 DQS 研究中，查询建议往往以短语形式呈现给用户，但是短语一般具有歧义性. 为了更准确捕捉用户的信息需求，文献 [34] 提出一种以问题形式呈现的查询建议（question-formed query suggestion, QQS）. 首先，对基于维基百科的 SQuAD 数据集[35] 进行关键词提取、合并和定位，得到问题形式的查询建议数据集. 接下来，应用文献 [36] 中的问题生成模型对数据集中（查询关键词，长文本，问题）三元组进行训练，利用查询关键词生成若干个 QQS. 最后在排名阶段中，应用 K 聚类算法对生成的 QQS 进行去重、排名，得到最终的 QQS 候选列表. 该文利用维基百科数据集中用户点击的文档作为背景信息，提出了一种新颖的查询建议形式，为用户提供了更直观的查询建议，在人工评估中也取得了较好表现.

在探索性搜索过程中，用户的搜索意图是模糊不定的，此时更需要检索系统提供额外支持. 文献 [37] 提出一种在科技文献检索系统中的 DQS 模型 SERP（search engine results page），该模型由动态编码器和解码器组成. 当用户发起初始查询后，检索系统会为用户提供一个无限下滑的结果页面，随着用户不断滚动下滑页面，SERP 编码器动态接收一系列 Doc2vec 的文档嵌入表示并不断更新隐藏层状态，利用文档

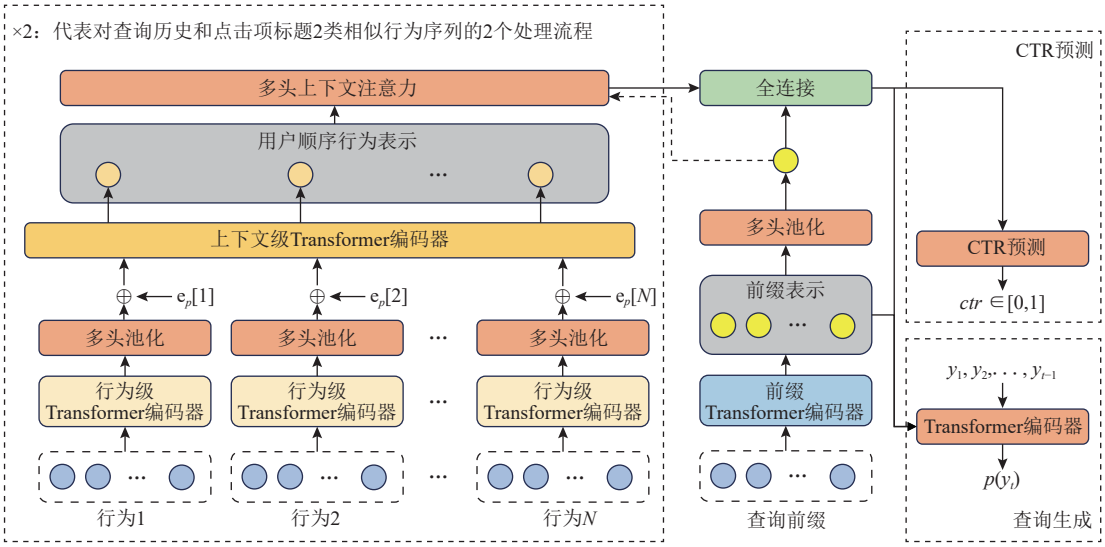

Fig. 16 M^2A model architecture

图 16 M^2A 模型结构

中的关键词为用户生成查询建议. 该模型成功建模了用户在结果页面的滚动行为，为用户提供独立于初始查询词的查询建议，其性能相较于过去伪相关反馈的 QS 方法有明显提升.

文献 [38] 面向求职中的职位搜索提出一种 DFSM (dynamic searching flow model)模型，将用户简历信息、历史会话信息、历史点击职位描述与用户的当前查询相结合，动态地对用户的意图进行建模，捕获了用户的实时意图和长期习惯，显著提升了职位搜索补全的个性化程度.

2.3 基于大型预训练语言模型的 DQS 方法

上述基于神经语言模型和基于编码器-解码器模型的 DQS 方法均为基于神经网络的全监督学习，这类方法针对不同的上下文因素，设计合适的结构来把归纳偏置引入模型中. 虽然该类方法能在解决长尾问题的同时为用户提供个性化的查询建议，但模型只能在特定情境下使用，泛用性较差. 为了解决这一问题，近年来一些研究将大型预训练语言模型应用于 DQS：首先在大规模无标注数据上进行模型预训练，再针对 DQS 任务对模型参数进行微调. 得益于大量语料的训练，该类模型很容易地迁移到不同应用场景下的 QS 中，具有良好的泛化能力；相较于传统生成式 DQS 方法模型的准确率与抗噪声能力也有显著提升.

近年来，大型预训练语言模型在自然语言处理 (NLP)领域常见问题，如机器问答和摘要生成等中均取得了较好表现. DQS 的研究范式也从以 RNN 及其变体模型为基础的结构工程转向了“预训练+微调” (pre-train and fine-tune)的范式. 文献 [39] 将预训练语言模型的思想应用于 DQS 领域. 该文献对比了完全训练的 Transformer 模型与 2 种微调的预训练语言模型，BERT[40] 和 BART[41] 在查询建议任务方面的表现. 为了使模型的输入输出形式更契合 DQS 任务，文献 [39] 依据模型特点不同对 3 种模型进行了相应处理：对 Transformer 模型，用分隔符将会话中的查询进行切分嵌入作为模型的输入，将查询会话的最后一次查询作为事实的训练模型；对 BERT 模型，利用在 BooksCorpus 数据集[42] 上预训练的编码器部分，结合针对 QS 任务微调的解码器输出查询建议；对 BART 模型，直接利用在 BooksCorpus 数据集预训练好的模型，结合经过新闻摘要数据集微调的模型权重为用户输出查询建议. 得益于 Transformer 擅长捕捉长距离语义相关性的特点，基于 BERT 和 BART 的 DQS 模型相较于以往基于 RNN 的模型能更好地处理长会话，能清晰地划分多主题会话中的主题边界，生成更多样化的查询建议结果.

基于 Transformer 的大型预训练语言模型有多种不同的变体模型. 以文献 [39] 为基础，文献 [19] 将多种大型预训练语言模型应用于 DQS，探究不同结构的基于 Transformer[43] 的预训练语言模型在 QS 任务方面的表现差异. 该研究针对 QS 任务分别微调了基于 BERT，ART，T5[44] 的扁平化架构(flat architecture)与层次化架构(hierarchical architecture). 这 2 类模型的解码模块均遵循标准化的 transformer 解码器，编码模块则略有差异：扁平化模型对会话中的查询仅进行词级嵌入，而层次化模型对查询分级进行词级嵌

入与字符级嵌入. 层次化模型得益于对查询词更细粒度的分级建模, 在短会话中表现较好; 而扁平化模型将查询按顺序连接, 能更好还原查询会话的位置结构, 因此在长会话、强噪声的情景下扁平化模型的健壮性更佳.

3 排名式 DQS 方法

传统 QS 在排名阶段往往将查询日志中的附加信息(如时间信息、地点信息等)通过人为定义的方式转化为特征信号, 结合学习排名(learning to rank)方法[45] 为特征信号赋予不同的权重, 从而对候选查询进行重排序. 但许多查询日志仅记录用户的查询内容, 并不记录额外的附加信息. 为了在重排序查询候选列表时降低对日志中附加信息的依赖, 一些研究采用排名式 DQS 方法, 利用深度学习模型评估候选查询的语义相关性, 并根据语义相关性对候选列表进行重排序, 在缺乏附加信息的情形下也能保证排序结果的可靠性. 在实际应用中这种语义相关性可以作为其他特征信号的补充.

排名式 DQS 方法一般首先会挖掘查询日志中常见的查询后缀, 将之与长尾查询组合生成新查询. 再通过训练神经网络模型来评估这种前缀-后缀组合的语义相关性, 并根据相关性对候选列表进行重排序. 这种生成方法虽然仍依赖于查询日志, 但也能一定程度上解决 QS 的长尾问题.

文献 [46] 针对 QAC 中的长尾问题, 首先在生成阶段改进了传统方法[1], 选取用户输入前缀中的末尾词, 检索日志中与之匹配频度最高的后缀, 再与查询前缀整体组合构成新查询. 在排名阶段中, 首先提取数据集中每个查询的前缀与后缀, 训练基于卷积神经网络的语义模型 CLSM(convolutional latent semantic model). 然后使用训练好的 CLSM 模型将前缀与后缀投影到公共的 128 维空间中并计算其余弦相似度, 得到表示相关程度的 CLSM 特征. 最后综合考虑 CLSM 特征与新查询的 N 元语法特征等对生成的新查询进行排序. 该方法与传统方法的区别在于它充分挖掘日志中的后缀信息构建新的查询, 并创新性地将建模文档相关性的 CNN 模型[47] 迁移到 QS 的排名阶段, 来衡量这种新查询构建的合理性. 相较于基于流行度的传统方法, 对于罕见查询的 MRR 有显著提升.

文献 [48] 受文献 [46] 启发, 对 2 个阶段均进行了改进. 在生成阶段, 同样通过查询前缀挖掘日志中的后缀信息, 但扩展了文献 [46] 中的前缀结束词, 由查询前缀的末尾开始, 从后向前将每个字母纳入前缀结束词中构成前缀结束短语, 并将每一个前缀结束短语作为索引检索日志中与之匹配频度最高的后缀, 从而扩大了后缀的检索范围, 为新查询的合成提供了更多的备选后缀. 在排名阶段, 为减小传统规范语言模型的计算开销, 利用一个基于 RNN 的非规范化语言模型(unnormalized language model, ULM)[49], 将规范化项近似为一个标量参数, 近似计算新查询中单词连续出现的概率, 从而建模前缀与后缀之间的语义连贯性, 解决了 CLSM 模型对查询长度不敏感的问题. 最后, 利用 FST 库(apache finite state transducer)优化了生成阶段中的日志检索过程, 提升了模型整体的运行效率, 同样对罕见查询有较好的响应能力.

4 DQS 的实验对比与分析

查询建议实验获得公平客观评价的充分条件在于使用合理的数据集和统一的评价指标. 本节主要介绍一些在 DQS 实验中常用的数据集、基线方法以及衡量算法性能的指标, 并对前文中一些方法的实验结果进行分析对比.

4.1 常用公共数据集

目前大多数 DQS 研究实验都基于 AOL 数据集[50]. AOL 是一个公开的查询日志数据集, 由从 2006 年开始的真实搜索引擎查询记录组成, 合计有 3 600 万次查询. 一些研究中也涉及其他基于搜索引擎的数据集, 如 Bing 和 Sogou 等. 还有一类基于应用程序网站的数据集, 包括 TaoBao 和 Amazon 购物网站数据集, Baidu 地图数据集和 LINKEDIN 求职网站数据集等.

大部分 DQS 研究者会根据需要将 AOL 数据集进行一定程度预处理. 例如对会话敏感的 DQS 研究往往将数据集按照时间窗口进行划分[20]; 以文献 [11] 为代表的基于神经语言模型的 DQS 方法则把 AOL 中少于 3 次的查询与过长的查询删除, 以减小其对模型训练的干扰. DQAC 的相关研究则将 AOL 中的完整查询拆分出前缀与后缀以便对前缀建模[13].

4.2 常用基线方法

基线(baseline)算法是研究论文中用于实验结果对比的前人算法. 如表 1 中所示, DQS 研究中最常用的基线算法可以分为 2 类: 一类是传统的基于日志流行度的方法, 包括 MPC(most popular completion)与 MPS(most popular suggestion); 另一类则是基于深度学习的方法, 包括 HRED(hierarchical recurrent encoder-

Table 1 Common Baseline Methods

表 1 常用基线方法

基线方法	说明	发表会议期刊
MPC[11–12,14,33,46,48]	基于流行度的查询补全方法	WWW'11
MPS[20,26,28–31,38]	基于流行度的查询建议方法	
HRED[12,19,21,26,28,30–33,39]	基于编码器-解码器模型的查询建议方法	CIKM'15
NQLM[12,14]	基于神经语言模型的查询补全方法	SIGIR'17

decoder)和 NQLM(neural language model for QAC).

4.3 常用评价指标

随着 DQS 研究不断深入,各 DQS 研究的评价指标也根据模型研究的侧重点不同呈现出多元化特点,这些评价指标包含了准确度、新颖性、满意度、效率等方面. 目前 DQS 中常用的评价指标有平均倒数排名(mean reciprocal rank, MRR)、归一化折扣累计效益(normalized discounted cumulative gain, NDCG)、时间、召回率(recall at K, Recall@K)、双语评估替补(bilingual evaluation understudy, BLEU)等. 其中 MRR 与 NDCG 是最常见的评价指标,二者都是通过考查候选项的排序位置来衡量用户的体验程度,二者区别在于计算方式不同: MRR 是通过排序位置次序号倒数值的累加而得,而 NDCG 则按照 log 调和级数进行计算. 时间是用于评价模型效率的重要指标,常用于 DQS 模型性能优化、排名算法改进的研究中. Recall@K 表示正确候选项出现在 top-K 候选列表中的次数. BLEU 是机器翻译领域的经典评价指标,在 DQS 研究中常被用于衡量模型生成的查询建议与实际数据的一致程度. 在实际研究中,为了能多方面评价模型生成查询建议的结果,通常采用 2 种或更多的评价指标综合分析查询建议的表现. 例如将 MRR 和 Recall@K 组合进行了评价[23], 将 MRR, NDCG 和 BLEU 三者组合进行了评价[32] 等.

4.4 实验对比分析

为方便说明算法效果,本节将对基于神经语言模型的 DQS、基于编码器-解码器模型的 DQS 和排名式 DQS 这 3 类方法的实验结果对比分析. 鉴于多数 DQS 研究所用指标大都不同,本节仅选取多数研究都采用的 MRR 评价对比分析不同模型的准确性.

此外,多数 DQS 研究虽然均使用了 AOL 数据集,但为了迎合不同模型应用场景,各研究对 AOL 数据集进行了不同的预处理、采用了不同的实验设置. 因此一些 DQS 研究之间无法单纯地通过 MRR 数值的大小评判其性能高低. 本节中也将对比介绍这些研究的实验对数据集的特殊处理.

4.4.1 基于神经语言模型的 DQS 方法的实验对比分析

基于神经语言模型的 DQS 方法的实验对比见表 2, 由于本文提及的基于神经语言模型的 DQS 方法均属于 QAC 领域,而 AOL 等数据集中缺少从查询前缀得到完整查询的真实记录,因此本节中方法均遵循文献 [47] 中的实验设置,即: 首先删除长尾查询以排除其对模型训练的干扰,然后将数据集中的完整查询人为拆分成前缀-完整查询的查询对,从而模拟真实的前缀补全过程.

Table 2 Experimental Comparison of Neural Language Model Based DQS Method

表 2 基于神经语言模型的 DQS 方法实验对比

方法	关键技术	数据集	数据集处理	评价指标	数值
NQLM[11]	词级语言模型	AOL	①删除出现少于 3 次的查询和过长的查询; ②遵循文献 [45] 对查询进行拆分	MRR	0.355
NQAC[12]	通过 GRU 建模用户因素、时间因素	AOL 生物医学数据集	①删除出现少于 3 次的查询和过长的查询; ②遵循文献 [45] 对查询进行拆分	MRR	0.382
文献 [13]	字符级语言模型	AOL Amazon	①删除过长的查询; ②将测试集中的完整查询拆分为查询前缀与后缀; ③按时间戳拆分查询以模拟真实情况	Hit Rate	0.448
Factor-Cell[14]	自适应权重矩阵	AOL	①将测试集中的完整查询拆分为查询前缀与后缀(前缀至少包含 2 个字符, 后缀至少包含 1 个字符)	MRR	0.309

注: 对于有多数据集的研究, 本表格仅记录 AOL 数据集上的实验数据.

在基于神经语言模型的 DQS 方法中, NQLM[11] 的 MRR 虽然不高,但为后续该类方法提供了一种"神经语言模型+beam search"的标准化思路,其他算法均在其基础上进行了一定程度改进: 文献 [13] 中的方法在神经语言模型中整合了查询纠错功能从而提升了查询补全的效果,同时利用 CPU 定制化实现了整个模型架构以提升 QAC 的效率,其响应时间相较于先前的 beam search 方法缩短了很多; NQAC[12] 在神经语言模型中整合了用户信息和时间信息,提升了神经语言模型的个性化程度,其 MRR 数值相较于 NQLM 提高了 7.3%.

4.4.2 基于编码器-解码器模型的 DQS 方法的实验对比分析

基于编码器-解码器模型的 DQS 方法的实验对比见表 3. 在数据集方面,根据方法应用场景的不同,各研究采用的数据集也有所不同. 但各方法对数据集都采取了类似处理,即按照一定时间窗口对数据集中所有的用户查询进行划分,从而模拟多段用户会话. 接下来本文将按照 2.2 节方法分类对不同模型

Table 3 Experimental Comparison of Encoder-Decoder Model Based DQS Method

表 3 基于编码器-解码器模型的 DQS 方法的实验对比

方法	分类	关键技术	数据集	数据集处理	评价指标	数值
HRED[20]	基于查询会话的方法	层次化建模会话因素	AOL	以 30 min 空闲时间为会话边界，将数据集划分为多个会话.	MRR	0.575
M-NSRF[21]		多任务学习、具有最大池化层的双向 LSTM	AOL	同文献 [19]，将数据集划分为多个会话.	MRR	0.238
HCARNN[23]		注意力机制	百度地图	将 1 位用户 1 天内的查询作为查询会话，对数据集进行划分.	MRR	0.138
文献 [24]		个性化编码策略	LINKEDIN	将 1 位用户较短时间内的查询作为查询会话，依此对数据集进行划分.	CTR	+5.62%
ACG[26]	基于查询重构的方法	注意力机制、ACG 机制	AOL	同文献 [19]，将数据集划分为多个会话.	MRR	0.594
RIN[28]		注意力机制、多学习	AOL	①同文献 [19]，将数据集划分为多个会话； ②按查询次数将测试集划分长中短会话.	MRR	0.825
AHNQS[29]	基于长期查询的方法	层次化模型、注意力机制	AOL	①同文献 [19]，将数据集划分为多个会话； ②删除出现次数少于 20 的查询，并保留数据集中长度大于 5 的会话以及至少有 5 个会话的用户.	MRR	0.851
FMN[30]	基于用户点击行为的方法	反馈记忆网络	Sogou	①同文献 [19]，将数据集划分为多个会话； ②将会话中最后一个查询视为正确的查询建议； ③将点击的标题视为文档内容.	MRR	0.581
HAN[31]		层次化模型+注意力机制		①同文献 [19]，将数据集划分为多个会话； ②将点击的标题视为文档内容.	MRR	0.604
CARS[32]		注意力机制、多任务学习	AOL	利用 BM25[51] 补充数据集中用户获取的文档列表.	MRR	0.542
M^2A[33]		Transformer、多任务学习	TaoBao	无特殊处理.	MRR	0.579
QQS[34]		Maxout pointer 机制	SQuAD	对关键词进行提取、合并、定位.	人工评估	0.33/0.4/0.67（Google/Bing/QQS）
SERP[37]		动态编码器	arXiv	①去除文档中的符号、数字； ②删除过于常见、过于罕见以及过短的查询.	人工评估（precision@5）	0.9
DFSM[38]		Dynamic Flow	Boss 直聘	同文献 [19]，将数据集划分为多个会话	MRR 和 NDCG 等	

注：对于有多数据集的研究，仅记录 AOL 数据集上的实验数据.

进行实验对比分析.

基于查询会话的方法中 M-NSRF[21] 将文档排序任务与查询建议任务联合学习，让二者共享会话级递归状态，AOL 数据集上的 MRR^2 数值相较于 HRED 提升了 3%.

基于查询重构的方法中，ACG[26] 利用复制机制保留具有价值的短语，优化了模型解码器，相较于 HRED 显著提升了罕见短语的查询建议能力，AOL 数据集上 MRR 值达到 3.7% 的提升；RIN[28] 引入重构嵌入层，以重构推理的方式提升了模型对下次查询的预测能力，其 MRR 值相较于 HRED 提升了 32.9%.

基于长期查询的方法中，AHNQS[29] 利用层次化注意力机制建模用户偏好，为日志中具有更丰富信息的查询会话赋予更高的重要性，相较于 MPS 在 AOL 数据集上的 MRR@10 提升达 21.86%.

基于用户点击行为的方法中，CARS[32] 同时对查询会话和会话中的点击文档进行分层建模，增强了过往查询会话中的点击行为分析，其 MRR 数值相较于 HRED 和 M-NSRF 分别提升了 6.6% 和 5.5%；QQS[34] 和 SERP[37] 应用场景较为特殊，二者各自设计了人工评估标准来衡量算法的优越性. 虽然二者无法与其他方法进行纵向比较，但是通过应用模型，二者在各自应用场景（阅读理解检索，科技文献检索）下，用户体验相较于不使用模型时均有所提升. QQS 在段落级问题生成模型的基础上引入门控自注意力编码器，以问题形式提供查询建议，有效减少了查询建议的歧义性，在人工评价中的表现相较于 Google 和 Bing 搜索引擎中的 QS 组件分别提升了 103% 和 67.5%. SERP 在探索性搜索中利用 DQS 模型总结会话中的文档主题，在多主题的探索性搜索中能够比伪相关反馈类的方法提供更多的语义信息，其 precision@5 相较于伪相关反馈方法提升了 11.1%~21.6%.

文献 [19, 39] 中涉及多种基于 Tranformer 的变体模型，在与 HRED 和 ACG 等基于 RNN 的传统模型的纵向对比中，得益于 Transformer 处理长文本的优势，基于 Transformer 的模型的 BLEU 数值表现显著优于 HRED 模型，其中基于 BART 的模型在 AOL 和 MS MARCRO 数据集下 BLEU-1 相较于 HRED 的提升最为显著，其数值分别提升了 34.9% 和 8.04%.

4.4.3 排名式 DQS 方法的实验对比分析

排名式 DQS 方法的实验对比见表 4，这些排名式 DQS 方法均属于 DQAC 领域. 实验数据处理方面，

同样采用了人为拆分方式将 AOL 数据集中的完整查询拆分为查询前缀和查询后缀.

Table 4 Experimental Comparison of Ranking-Based DQS Method

表 4 排名式 DQS 方法的实验对比

方法	关键技术	数据集	数据集处理	评价指标	数值
CLSM[46]	CNN，LWG	AOL，Bing	在单词边界处对数据集中的查询进行前缀和后缀的拆分	MRR	0.245
ULM[48]	LSTM，MCG	AOL	同文献 [46]	MRR	0.340

CLSM[46] 提出了前缀和后缀组合生成新查询的方法，并利用基于 CNN 的排名式框架对这些新查询进行排序，相较于 MPC 显著提升了处理罕见前缀时的 MRR 数值. 而 ULM[48] 在 CLSM 的基础上不仅改进了生成阶段的方法，而且在排名阶段进一步建模了前缀和后缀之间的语意连贯性，其 MRR 指标相较于 CLSM 提升了 38.8%.

5 总结与展望

如今，深度学习技术已经广泛应用于查询建议和查询补全领域. 本文针对现有基于深度学习的查询建议技术进行了分类、梳理和归纳. 根据深度学习在 DQS 中应用阶段的不同，将基于深度学习的个性化查询补全和查询建议分为生成式和排名式两类，对每类方法中的代表性模型进行了研究、分析与对比，并总结了每类方法的技术特点. 总结基于深度学习的查询建议研究领域的重点问题和发展趋势，认为该领域还存在 5 个具有挑战性的研究方向.

1)基于“prompt 范式”的 DQS

以前基于大型预训练语言模型的 DQS 研究[19] 大都遵循着预训练+微调参数的“微调范式”. 近年来，预训练+prompt(即“prompt 范式”)的流行打破了这一范式. “prompt 范式”无需微调大量参数，在预训练语言模型的基础上针对不同的情景找出最合适的 prompt，帮助语言模型完成不同的下游任务. 例如在查询扩展(query expansion)领域中，文献 [52] 利用少量 prompt 经由大型预训练语言模型生成伪文档进行查询扩展，从而帮助检索系统查询消歧. 文献 [53] 则在设计 prompt 丰富查询重写(query rewriting, QR)时，精心设计了正确性、清晰性、信息性和简要性 4 个关键属性融入到 prompt 中来约束和提升查询重写效果. 基于“prompt 范式”的 DQS 能赋予模型少量样本学习甚至是零样本学习的能力，使 DQS 模型适应标记数据很少的新场景，具备更强的泛化性. 因此“prompt 范式”是未来基于大型预训练语言模型 DQS 的重要研究方向.

2)对话式检索中的 QS

近年来，以 ChatGPT 为代表的智能对话 AI 引发了信息检索系统从临时检索到对话式检索的转变. 对话式检索实现了用户与检索系统之间的信息互动，通过多轮对话更准确地理解用户信息需求. 这种方式为用户提供了更直观的搜索结果与更人性化的搜索体验. 然而对话式检索目前面临着两大挑战：首先，对话往往较长，且包含着更多的语言现象(如省略、指代等)，因此对话式检索中的用户意图往往更加隐蔽，对检索系统的上下文敏感性提出了更严格的要求. 其次，对话中用户表达质量不一，低质量的查询往往无法为用户带来满意的结果. 针对以上问题，目前一些研究从查询重写的角度来提升对话式检索的质量. 文献 [54] 提出一种生成式框架 CGF(constrained generation framework)，利用上下文信息对语音对话系统中语音识别有误的查询进行重写，从而提升用户查询满意度；文献 [55] 提出一种基于强化学习的 CONQRR(conversational query rewriting for retrieval)框架直接集成在现有 CQA(conversational question answering)系统中，将用户的对话上下文重写为一个独立问题，改善用户查询的表达质量，有效提升 QA 检索结果的准确度；受对话式检索中常见语言现象的启发，文献 [56] 提出一种 CRDR(conversational query rewriting method for dense retrieval)模型，利用模型提取上下文中的关键词对原始查询进行字符级修改，从而明确查询的表意；文献 [57] 借鉴了机器阅读理解语言模型来解决原始查询中的常见歧义，在查询重写过程中无需从会话搜索数据获得监督即可加强用户意图理解和表达. 这些方法为在对话式检索中应用 QS 提升检索质量提供了有效途径. 在长对话中为用户适时提供查询建议，有助于修正用户当前的查询表述，明确用户查询意图，提升后续对话式检索结果的准确性.

3)基于多任务学习的 DQS

查询建议作为查询过程的一部分，包含候选生成与候选排名两大任务，与查询过程中其他任务，如文档排序、查询扩展等有着很强的关联性，因此多任务学习(multi-task learning)在 DQS 领域具有极大潜力. 文献 [58] 设计出一个多任务框架 ConvGQR(generative query reformulation framework for conversational search)，利用生成式预训练语言模型将查询重写与检索任务

关联起来，其中一个模型用于查询重写，另一个用于生成潜在答案，并辅以一种知识注入机制来共同优化查询改写和检索，提升了会话查询优化效果. 相比于单任务学习，相关联的多任务学习有更好的泛化效果，可以充分利用不同维度的上下文信息，实现不同任务之间的优势互补. 目前 DQS 领域中基于多任务学习的研究较少，是未来 DQS 领域值得深入研究的重要方向之一.

4)生成式方法的效率与个性化的权衡

效率问题一直是 DQS 研究领域的一大挑战. DQS 的效率与个性化对于用户体验都至关重要. 随着大语言模型时代的发展，“重写-检索-阅读”的大语言模型框架也将查询重写融入大语言模型，查询改写的效率影响愈发突出[59]. 事实上，在目前生成式 DQS 研究中，随着对个性化因素挖掘越来越深入，相应的模型结构也愈发复杂. 受制于神经网络模型的迭代计算速度，个性化程度高的复杂模型生成补全项的效率往往较低. 因此，如何在特定应用情景下提炼真正针对该情景有效的个性化因素，从而取得效率和个性化两方面的平衡，是未来 DQS 应用领域中重要且具有挑战性的问题.

5)更有效地建模用户长期行为序列

目前 DQS 研究中，主要利用注意力机制从用户短期行为序列中提取用户查询意图，缺乏对长期行为序列的建模和利用. 但是将注意力机制套用在长期行为序列上会导致巨大的时间开销. 针对推荐系统中的此类问题，文献 [60] 提出一种基于局部敏感哈希(locality-sensitive hashing)的端到端目标注意力机制(end-to-end target attention)有效地减少了计算开销，成功建模用户长期行为序列. 文献 [61] 将查询建议任务建模为一个异构图上的链接预测任务，利用图学习方法学习异构图上查询与文档间的关系，提升了查询建议生成的准确率和覆盖率. 在 DQS 研究中，也可以采用类似方法建模用户行为序列. 在 DQS 中应用相关技术取代传统注意力机制，从而建模长期查询和点击行为也是 DQS 未来发展方向之一.

作者贡献声明：田萱提供了论文的方向与思路，指导并修改论文；徐泽洲调研该领域研究，完成论文的主要写作；王子涵绘制了论文中的模型图.

参 考 文 献

[1] Bar-Yossef Z, Kraus N. Context-sensitive query auto-completion[C]// Proc of the 20th Int Conf on World Wide Web. New York: ACM, 2011: 107−116

[2] Tian Xuan, Zhang Xiao, Meng Xiangguang, et al. Research Review of time-sensitive query auto-completion technique[J]. Acta Electronica Sinica, 2015, 43(6): 1160−1168 (in Chinese)
(田萱，张骁，孟祥光，等. 时间敏感查询词补全关键技术研究综述[J]. 电子学报，2015, 43(6)：1160−1168)

[3] Tahery S, Farzi S, et al. Customized query auto-completion and suggestion – A review[J]. Information Systems, 2020, 87(1): 101415–101432

[4] Lecun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553): 436−444

[5] Shokouhi M. Learning to personalize query auto-completion[C]//Proc of the 36th Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2013: 103−112

[6] Hu Sheng, Xiao Chuan, Ishikawa Y. An efficient algorithm for location-aware query autocompletion[J]. IEICE Transactions on Information and Systems, 2018, 101(1): 181−192

[7] Huang Zhipeng, Mamoulis N. Location-aware query recommendation for search engines at scale[C]//Proc of the 15th Int Symp on Advances in Spatial and Temporal Databases. Berlin: Springer, 2017: 203−220

[8] Qi Shuyao, Wu Dingming, Mamoulis N. Location aware keyword query suggestion based on document proximity[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 28(1): 82−97

[9] Kannadasan M R, Aslanyan G. Personalized query auto-completion through a lightweight representation of the user context[J]. arXiv preprint, arXiv: 1905.01386, 2019

[10] Jiang J Y, Ke Y Y, Chien P Y, et al. Learning user reformulation behavior for query auto-completion[C]//Proc of the 37th Int ACM SIGIR Conf on Research & Development in Information Retrieval. New York: ACM, 2014: 445−454

[11] Park D H, Chiba R. A neural language model for query auto-completion[C]//Proc of the 40th Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2017: 1189−1192

[12] Fiorini N, Lu Zhiyong. Personalized neural language models for real-world query auto completion[J]. arXiv preprint, arXiv: 1804.06439, 2018

[13] Wang Powei, Zhang Huan, Mohan V, et al. Realtime query completion via deep language models[C]//Proc of the 41st Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2018, 2319−2327

[14] Jaech A, Ostendorf M. Personalized language model for query auto-completion[J]. arXiv preprint, arXiv: 1804.09661, 2018

[15] Jaech A, Ostendorf M. Low-rank RNN adaptation for context-aware language modeling[J]. Transactions of the Association for Computational Linguistics, 2017, 6(1): 497−510

[16] Shokouhi M, Radinsky K. Time-sensitive query auto-completion[C]// Proc of the 35th Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2012: 601−610

[17] Vijayakumar A K, Cogswell M, Selvaraju R R, et al. Diverse beam search: Decoding diverse solutions from neural sequence models[J].

arXiv preprint, arXiv: 1610.02424, 2016

[18] Gabín J, Ares M E, Parapar J. Keyword embeddings for query suggestion[C]//Proc of the 45th European Conf on Information Retrieval. Berlin: Springer, 2023: 346−360

[19] Mustar A, Lamprier S, Piwowarski B. On the study of transformers for query suggestion[J]. ACM Transactions on Information System, 2022, 40(1): 1−18, 27

[20] Sordoni A, Bengio Y, Vahabi H, et al. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion[C]// Proc of the 24th ACM Int Conf on Information and Knowledge Management. New York: ACM, 2015: 553−562

[21] Ahmad W U, Chang K W. Multi-task learning for document ranking and query suggestion[C/OL]//Proc of the 6th Int Conf on Learning Representations. 2018[2022-12-11]. https://openreview.net/pdf?id=SJ1nzBeA-

[22] Conneau A, Kiela D, Schwenk H, et al. Supervised learning of universal sentence representations from natural language inference data[C]//Proc of the 2017 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2017: 670−680

[23] Song Jun, Xiao Jun, Fei Wu, et al. Hierarchical contextual attention recurrent neural network for map query suggestion[J]. IEEE Transactions on Knowledge & Data Engineering, 2017, 29(9): 1888−1901

[24] Zhong Jianling, Guo Weiwei, Gao Huiji, et al. Personalized query suggestions[C]//Proc of the 43rd Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2020: 1645−1648

[25] Luong M T, Pham H, Manning C D. Effective approaches to attention-based neural machine translation[C]//Proc of the 2015 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2015: 1412−1421

[26] Dehghani M, Rothe S, Alfonseca E, et al. Learning to attend, copy, and generate for session-based query suggestion[C]//Proc of the 26th ACM on Conf on Information and Knowledge Management. New York: ACM, 2017: 1747−1756

[27] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[C/OL]//Proc of the 6th Int Conf on Learning Representations. 2015[2021-04-22]. https://arxiv.org/abs/1409.0473

[28] Jiang Jun Yu, Wang Wei. RIN: Reformulation inference network for context-aware query suggestion[C]//Proc of the 27th ACM Int Conf on Information and Knowledge Management. New York: ACM, 2018: 197−206

[29] Chen Wanyu, Cai Fei, Chen Honghui, et al. Attention-based hierarchical neural query suggestion[C]//Proc of the 41st Int ACM SIGIR Conf on Research & Development in Information Retrieval. New York: ACM, 2018: 1093−1096

[30] Wu Bin, Xiong Chenyan, Sun Maosong, et al. Query suggestion with feedback memory network[C]//Proc of the 26th World Wide Web Conf. New York: ACM, 2018: 1563−1571

[31] Li Xiangsheng, Liu Yiqun, Li Xin, et al. Hierarchical attention network for context-aware query suggestion[C]//Proc of the 14th Asia Information Retrieval Societies Conf. Berlin: Springer, 2018: 173−186

[32] Ahmad W U, Chang Kaiwei, Wang Hongning. Context attentive document ranking and query suggestion[C]//Proc of the 42nd Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2019: 385−394

[33] Yin Din, Tan Jiwei, Zhang Zhe, et al. Learning to generate personalized query auto-completions via a multi-view multi-task attentive approach[C]//Proc of the 26th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining. New York: ACM, 2020: 2998−3007

[34] He Yuxin, Mao Xianling, Wei Wei, et al. Question-formed query suggestion[C]//Proc of the 12th IEEE Int Conf on Big Knowledge. Piscataway, NJ: IEEE, 2021: 482−489

[35] Rajpurkar P, Zhang Jian, Lopyrev K, et al. SQuAD: 100, 000+ questions for machine comprehension of text[C]//Proc of the 2016 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2016: 2383–2392

[36] Zhao Yao, Ni Xiaochuan, Ding Yuanyuan, et al. Paragraph-level neural question generation with maxout pointer and gated self-attention networks[C]//Proc of the 2018 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2018: 3901−3910

[37] Medlar A, Li Jing, Głowacka D. Query suggestions as summarization in exploratory search[C]//Proc of the 6th Conf on Human Information Interaction and Retrieval. New York: ACM, 2021: 119−128

[38] Zhou Zile, Zhou Xiao, Li Mingzhe, et al. Personalized query suggestion with searching dynamic flow for online recruitment[C]// Proc of the 31st ACM Int Conf on Information & Knowledge Management. New York: ACM, 2022: 2773−2783

[39] Mustar A, Lamprier S, Piwowarski B. Using BERT and BART for query suggestion[C/OL]//Proc of the 1st Joint Conf of the Information Retrieval Communities in Europe. 2020[2023-02-01]. https://www.irit.fr/CIRCLE/wp-content/uploads/2020/06/CIRCLE20_06.pdf

[40] Kenton J, Toutanova L K. BERT: Pre-training of deep bidirectional transformers for language understanding[C]//Proc of the 2019 Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg, PA: ACL, 2019: 4171−4186

[41] Lewis M, Liu Yinhan, Goyal N, et al. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension[C]//Proc of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2020: 7871−7880

[42] Zhu Yukun, Kiros R, Zemel R, et al. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books[C]//Proc of the 2015 IEEE Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2015: 19−27

[43] Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer[J]. The Journal of Machine Learning Research, 2020, 21(1): 5485−5551

[44] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017, 30(1): 5998−6008

[45] Burges C J C. From ranknet to lambdarank to lambdamart: An

overview[C/OL]. 2010[2022-02-02]. https://www.microsoft.com/en-us/research/uploads/prod/2016/02/MSR-TR-2010-82.pdf

[46] Mitra B, Craswell N. Query auto-completion for rare prefixes[C]// Proc of the 24th ACM Int Conf on Information and Knowledge Management. New York: ACM, 2015: 1755−1758

[47] Shen Yelong, He Xiaodong, Gao Jianfeng, et al. Learning semantic representations using convolutional neural networks for web search[C]//Proc of the 23rd Int Conf on World Wide Web. New York: ACM, 2014: 373−374

[48] Wang Sida, Guo Weiwei, Gao Huiji, et al. Efficient neural query auto completion[C]//Proc of the 29th ACM Int Conf on Information & Knowledge Management. New York: ACM, 2020: 2797−2804

[49] Sethy A, Chen S, Arisoy E, et al. Unnormalized exponential and neural network language models[C]//Proc of the 40th IEEE Int Conf on Acoustics, Speech and Signal Processing (ICASSP). Piscataway, NJ: IEEE, 2015: 5416−5420

[50] Pass G, Chowdhury A, Torgeson C. A picture of search[C]//Proc of the 1st Int Conf on Scalable Information Systems. New York: ACM, 2006: 1−7

[51] Robertson S, Zaragoza H. The probabilistic relevance framework: BM25 and beyond[J]. Foundations & Trends in Information Retrieval, 2009, 3(4): 333−389

[52] Wang Liang, Yang Nan, Wei Furu. Query2doc: Query expansion with large language models[J]. arXiv preprint, arXiv: 2303.07678, 2023

[53] Ye Fanghua, Fang Meng, Li Shenghui, et al. Enhancing conversational search: Large language model-aided informative query rewriting[C]// Proc of the 2023 Conf on Empirical Methods in Natural Language Processing, Stroudsburg, PA: ACL, 2023: 5985–6006

[54] Hao Jie, Liu Yang, Fan Xing, et al. CGF: Constrained generation framework for query rewriting in conversational AI[C]//Proc of the 2022 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2022: 475−483

[55] Wu Zeqiu, Luan Yi, Rashkin H, et al. CONQRR: Conversational query rewriting for retrieval with reinforcement learning[C]//Proc of the 2022 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2022: 10000−10014

[56] Qian Hongjin, Dou Zhicheng. Explicit query rewriting for conversational dense retrieval[C]//Proc of the 2022 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2022: 4725−4737

[57] Yang Dayu, Zhang Yue, Fang Hui. Zero-shot query reformulation for conversational search[C]//Proc of the 2023 ACM SIGIR Int Conf on Theory of Information Retrieval. New York: ACM, 2023: 257−263

[58] Mo Fengran, Mao Kelong, Zhu Yutao, et al. ConvGQR: Generative query reformulation for conversational search[C]//Proc of the 61st Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2023: 4998–5012

[59] Ma Xinbei, Gong Yeyun, He Pengcheng, et al. Query rewriting in retrieval-augmented large language models.[C]//Proc of the 2023 Conf on Empirical Methods in Natural Language Processing, Stroudsburg, PA: ACL, 2023: 5303–5315

[60] Chen Qiwei, Pei Changhua, Lv Shanshan, et al. End-to-end user behavior retrieval in click-through rateprediction model[J]. arXiv preprint, arXiv: 2108.04468, 2021

[61] Palumbo E, Damianou A, Wang A, et al. Graph learning for exploratory query suggestions in an instant search system[C]//Proc of the 32nd ACM Int Conf on Information and Knowledge Management. New York: ACM, 2023: 4780−4786

Tian Xuan, born in 1976. PhD, associate professor. Senior member of CCF. Her main research interests include intelligent information processing, data mining, and machine learning.

田　萱, 1976 年生. 博士. 副教授. CCF 高级会员. 主要研究方向为智能信息处理、文本挖掘、机器学习.

Xu Zezhou, born in 1998. Master. Student member of CCF. His main research interests include intelligent information processing, data mining, and machine learning.

徐泽洲, 1998 年生. 硕士. CCF 学生会员. 主要研究方向为智能信息处理、数据挖掘、机器学习.

Wang Zihan, born in 2000. Master. Student member of CCF. His main research interests include intelligent information processing, data mining, and machine learning.

王子涵, 2000 年生. 硕士. CCF 学生会员. 主要研究方向为智能信息处理、数据挖掘、机器学习.