分类号_____ 学号 M202073411
学校代码 10487 密级_____

华中科技大学

硕士学位论文

（学术型□ 专业型☑）

基于预训练语言模型的多轮检索式对话研究

学位申请人：李申瑞

学 科 专 业：计算机技术

指 导 教 师：胡迎松 副教授

答 辩 日 期：2022 年 5 月 24 日

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Professional Master Degree

Research on Multi-turn Retrieval Dialogue Based on Pre-trained Language Model

Candidate : LI Shenrui

Major : Computer Technology

Supervisor : Associate Prof. HU Yingsong

Huazhong University of Science and Technology

Wuhan 430074, P. R. China

May, 2022

独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：李申瑞

日期：2022年 5月 29日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本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和汇编本学位论文。

保　密□，在_____年解密后适用本授权书。

本论文　不保密☑。

（请在以上方框内打“√”）

学位论文作者签名：李申瑞　　指导教师签名：胡迎松

日期：2022 年 5 月29 日　　日期：2022年5月29日

摘 要

人机对话系统已被广泛应用于社会的各个方面。预训练语言模型在各项自然语言处理任务上不断取得突破，其在对话系统中的应用得到重点关注。检索式对话将复杂的对话问题转换为搜索问题，在实际的场景中预先设计大量的候选回复，系统根据用户输入检索出合适的回复，流程清晰且易于实现。

由于预训练语言模型是在大规模通用数据上训练得到的，在对话领域适应性较弱，因此提出了一种利用对话匹配训练任务进行领域后训练的方法，通过使用对话匹配任务构造的上下文话语和回复样本对预训练语言模型进行后训练，提高了预训练语言模型的对话领域适应性。现有的检索式方法没有充分提取上下文话语和回复之间的匹配信息，因此构建了一种多级匹配的检索式对话模型，该模型从话语级别和单词级别同时建模，分别提取上下文话语中的局部匹配信息和单词序列中的全局匹配信息并将其融合，以计算出最终的匹配分数。为解决对话模型外部知识不足的问题，抽取数据集中的知识三元组，并融合成知识图谱，使用键值存储模块将其引入对话模型，丰富了模型的外部知识信息。

在数据集 Ubuntu、Douban 和 E-commerce 上进行了实验。实验结果表明，结合领域后训练方法得到的 BERT 模型相比于基本的 BERT 模型在 Ubuntu 数据集上核心指标$R_{10}@1$提升了 4.7%；多级匹配检索式对话模型相比于 BERT 在三个数据集上性能分别提升了 5.9%、2.5%和 11.3%，且该模型结构应用于最新的 BERT-FP 模型之后性能仍提升了 0.5%、1.0%和 0.4%；在外部知识引入之后对话模型三个数据集上的性能分别提升了 0.5%、0.3%和 0.2%。

关键词： 自然语言处理；对话系统；预训练语言模型；知识图谱；检索式对话

Abstract

Human-machine dialogue systems have been widely used in all aspects of society. The pre-trained language model has continuously made breakthroughs in various natural language processing tasks, and its application in dialogue systems has received much attention. Retrieval dialogue converts complex dialogue questions into search questions. A large number of candidate responses are pre-designed in actual scenarios, and the system retrieves appropriate responses based on user input. The process is clear and easy to implement.

Since the pre-trained language model is trained on large-scale general data and has weak adaptability in the dialogue domain, a method for domain post-training is proposed using the dialogue matching training task. The pre-trained language model is post-trained with the response samples by using the pairs of contextual utterances and response constructed by the dialogue matching task, which improves the adaptability of the pre-trained language model to the dialogue domain. Existing retrieval-based methods do not fully extract matching information between contextual utterances and response, so a multi-level matching retrieval-based dialogue model is constructed. The local matching information of the word sequence and the global matching information in the word sequence are fused to calculate the final matching score. In order to solve the problem of insufficient external knowledge of the dialogue model, the knowledge triples in the data set are extracted and fused into a knowledge graph, and the key-value storage module is used to introduce them into the dialogue model, which enriches the external knowledge information of the model.

Experiments are conducted on datasets Ubuntu, Douban and E-commerce. The experimental results show that compared with the basic BERT model, the core index $R_{10}@1$ of the Ubuntu corpus is improved by 4.7% in the BERT model obtained by combining the domain post-training method; compared with BERT, the performance of the multi-level matching retrieval dialogue model is improved by 5.9%, 2.5% and 11.3% on the three datasets respectively, and the performance of the model is still improved by 0.5%, 1.0% and 0.4% after the structure is applied to the latest BERT-FP model; after the external knowledge is introduced, the performance of dialogue model on the three datasets is

respectively improved by 0.5%, 0.3% and 0.2%.

Key words: Natural Language Processing, Dialogue System, Pretrained Language Model, Knowledge Graph, Retrieval-based Dialogue

目 录

1 绪论

人机对话技术在科研与工业界具有很高的研究价值，在社会上也有很高的应用价值。随着科技的快速发展，对话技术被广泛地应用在生活的方方面面，技术也愈发成熟。国内外涌现了非常多的关于对话系统的研究，极大地促进了对话系统的发展。

1.1 研究背景与意义

人机对话技术[1]在科研与工业界一直都是热点研究问题，旨在尽量模仿人类之间的对话交互方式。在过去，拥有一个足够智能的聊天系统或者虚拟助手较为困难，似乎只能在高科技电影中才能看到。随着时代的快速发展，自然语言处理领域已取得重大的技术突破，让人们看到拥有一个可以与人类进行智能对话的虚拟助手不再是一件难事。

从时间线上来看，对话系统历经三个阶段：第一阶段为人工阶段，对话系统主要依靠手工制定规则和设计模板；第二阶段，随着机器学习的发展，基于统计学习的对话系统模型被提出；第三阶段，随着深度学习的发展，各种基于深层神经网络的对话系统模型被提出。ELIZA[2]是对话系统历史上第一代的聊天机器人，其主要功能是与用户进行心理交谈，但此时的对话系统还主要依靠人工设计语法、规则和模板，虽简单易懂，但由于该项工作费时费力，且在各个领域的泛化与扩展能力不足。第二阶段的人机对话系统摒弃了第一代的缺点，主要依靠机器学习的统计学习方法，但是这种方法学习能力较差且晦涩难懂，不易被理解，相对来说规模不易扩大。随着时代发展，自然语言处理领域包括对话系统在内的多个任务的表现都得到了有效的提升。第三代基于深度学习和数据驱动的对话系统逐渐成为主流，取代了传统方法，成为科研与工业界热门研究的课题。

对话系统按照不同的应用场景可分为两种类型：任务型对话系统和非任务型对话系统。任务型对话系统为解决用户提出的需求或问题而设计，因此，任务型对话系统也被称为问答系统。闲聊型对话系统面向领域广泛，可支持与用户的闲聊对话，没有规定的内容与主题，主要为用户提供休闲娱乐等服务。

根据实现方式的不同，可将对话系统分为生成式对话系统和检索式对话系统。一般来说，生成式对话模型主要是在大量的对话数据上进行训练，通过神经网络中的隐藏单元学习上下文中的语义信息，从而当用户输入一段话语时就可以针对性地输出合适的回复。基于生成式的方法大都为端到端，不像管道结构需要划分多个模块，不需要人工去设计模块并且可以减少模块之间误差的累计，进一步提升回复质量。但是人与人之间的对话类似一个“主观题”，回答并没有标准的答案，只要保证内容不偏离主题即可，因此生成式对话模型的评估一直是难以解决的问题。此外，模型容易产生单调、无聊的回复，想要搭建一个优秀的生成式对话模型具有一定的挑战。基于检索的对话方法其主要思想是在知识库中去寻找与对话历史匹配的候选回复，然后通过计算上下文话语和候选回复之间的匹配分数，选择出最合适的回答作为输出返回给用户。这种方法关键之处在于检索候选回复与计算匹配分数的算法，因为选择的回复都在语料中，故在流畅性、连贯性和信息量往往优于生成式方法。然而目前在多轮会话方面的研究仍存在多个问题：多轮对话之间通常存在层次结构，而这必然呈现出更复杂的语义信息；多轮对话通常以简短、非正式的形式表达，故想要更好地理解语义必然要了解上下文信息；多轮对话不仅关注上下文信息，还需要考虑对话之间的逻辑性和一致性；关于多轮对话的优质语料不易获取，并且在特定领域的数据非常匮乏，导致无法构建优质的对话系统。因此，解决多轮对话中存在的上述问题将是一个非常具有挑战性的任务。

综上所述，多轮对话在科研和工业界具有很高的应用价值，而如何构建一个高效优质的检索式多轮对话系统具有很大的研究意义和挑战。本文将着重于检索式多轮对话的工作，解决目前存在的多个问题，以提高对话模型的性能。

1.2 国内外研究现状

国内外早期关于对话系统的研究工作重心在任务型对话系统上，随着海量数据的积累和深度学习的发展，越来越多的研究集中在开放领域的非任务型对话系统上，而检索式对话方法由于其易于实现和评价的特点被广大研究者备受关注。预训练语言模型的提出更是进一步促进了检索式对话方法的发展。

1.2.1 早期对话系统工作

在对话系统历史上，第一款用于实际工作的人机对话系统由麻省理工学院于 20 世纪 60 年代提出，该机器人主要扮演一位心理治疗师的角色，用来倾听用户或患者的心声并给出积极的回复。自此之后，多个人机对话系统相继提出，例如用于闲聊的 ALICE 系统[3]，可以对关于棒球问题进行回答的 BASEBALL 系统[4]，甚至可以回答知识脑力竞赛问题的 Waston 对话机器人[5]，且 Waston 首次战胜了人类挑战者，这一发现轰动世界并极大地推动了人机对话技术的发展。但是早期的对话系统使用人工设计的模板与规则进行回答，或者使用 AIML 的标记语言实现匹配回答，导致对话系统的性能完全取决于规则与模型定义的质量，并且需要花费大量的人力物力去进行对话系统规则的设计与完善工作。

在应用上，对话系统可分为两类：任务型对话系统和非任务型对话系统，后者也称为聊天机器人。面向任务的对话系统[6]旨在为用户解决特定的需求，例如预定餐厅、产品推荐、天气或位置查询和车票预定等等。通常来说，构建任务型对话系统的方法可大致分为两种：管道方法与端到端方法。管道方法相对简单，易于理解，可以分开独立解决各自的问题。但是管道方法存在三方面问题：数据问题，每个模块之间独立，这就需要为每个模块提供大量标注数据，费时费力；依赖问题，每个模块都难免产生错误，并且这种错误会随管道向下传播，当下游发现错误时并不能及时发现其错误来源；领域问题，管道方法对于领域敏感，需要为每个域单独设计语义槽、候选动作和策略。

随着深度学习的飞速发展，端到端方法逐渐取代管道方法在对话系统中的应用。基于端到端的任务型对话系统主要模型结构为编码器-解码器结构，用户给定输入，系统可直接给出输出，不需要经过多个模块的处理。Wen 等人为解决模型缺少领域信息的问题，在端到端模型中加入了历史信息和具体的领域信息[7]。为解决序列到序列模型在任务型对话系统中难以构建的问题，Eric 等人提出一个键值索引网络，该模型将注意力与索引机制相结合，从而使得可以查询出更多信息[8]。Williams 等人提出一个混合编码网络模型，该模型将循环神经网络与特定领域知识的软件和系统动作模板相结合，减少了训练所需的数据量[9]。Lee 等人构建了一个开源的多域端到端对

话系统平台 ConvLab，旨在使研究人员能够在公共环境中快速建立可重用组件的实验，并且该平台提供完整的注释数据和预训练参考模型，用户可以通过简单地设置来进行复杂实验[10]。

非任务型对话系统，是一种面向多个领域、用于与用户进行闲聊的人机对话系统。在真实场景下，非任务型对话系统要求与用户进行连贯性一致、有逻辑地对话，并且当用户转移话题时系统可以灵活稳定地回复。按照构建方式的不同，非任务型对话系统可分为生成式对话系统和检索式对话系统。生成式方法大都采用序列到序列（Sequence-to-Sequence, Seq2Seq）模型[11]，通过大规模的无标注数据集端到端训练编码器-解码器结构，从而使得模型可以学习语义特征进而生成回复。由于循环神经网络（Recurrent Neural Network，RNN）的结构问题会导致前一个节点信息的丢失，因此 Seq2Seq 模型在生成长句子或者多轮对话方面有较大的问题。Bahdanau 等人提出了注意力机制，可以用来计算输入话语与生成的回复之间的匹配程度，匹配程度越高，分数越高，由此可以提高话语的长度和轮数[12]。生成式方法可以生成一些自由组合、语法正确的话语，但是对于一些没有学习过的问题经常会生成一些无聊的单调性回复，并且生成式模型难以评估，因为用户的输入话语通常是个“主观题”，候选答案可能有多个，无法评价孰优孰劣。为此，Wiseman 等人提出了束搜索（Beam Search）帮助 Seq2Seq 模型生成更多灵活多变的回复[13]。随后的生成式对话系统的研究大都集中在预训练语言模型上，Zhang 等人提出一个大型、可微调的神经对话回复生成模型 DialoGPT，该模型使用大规模的通用语料 Reddit 进行训练，在自动和人工评估方面实现了最优的性能，生成的回复更加丰富和上下文一致[14]。为解决对话数据和背景知识缺乏的问题，Zhou 等人提出一个中文的数据集 KdConv，该数据集由多个领域的对话语料构成。整个数据集由知识三元组构成，形成了完整的知识图谱。此外，KdConv 将生成式模型与检索式模型在数据集上进行了性能对比，实验结果表明了引入背景知识可以增强模型[15]。

检索式对话方法通过在候选项中选择出最合适的话语进行回复，不需要通过系统自动生成话语，这些候选回复通常是预先定义或者收集的优质对话语料，因此模型输出的话语在流畅性、一致性和信息量上要优于基于生成式的模型，因此检索式对话被

广泛应用于目前工业界的聊天助手或者商业客服。早期基于检索式对话的工作使用简单的神经网络来构造单轮对话模型，这种模型只关注当前用户输入的话语，而不注重对历史对话信息的挖掘，这就导致模型的输出常常出现上下文不一致、主题不连续的问题。检索式多轮对话方法不仅关注当前用户的输入，还注重对话历史中的上下文信息，帮助系统选择更合适的回复。神经表征学习和预训练方法的不断发展为基于检索的对话方法奠定了坚实的技术基础，极大地推动了检索式对话方法的发展。

1.2.2 检索式多轮对话相关工作

从时间线上看，检索式多轮对话[16]的发展可以分为多个阶段，从手工设计规则和模版、机器学习的统计方法和深度学习方法。手工设计规则费时费力，且模型泛化能力弱；基于统计的多轮对话方法对于话语的理解能力较弱；基于神经网络的检索式多轮对话方法已经取得较为不错的成果，在科研界与工业界有着非常广泛的应用。传统的用于回复选择的上下文-回复选择匹配模型通常分为两个框架：基于表征的匹配框架和基于交互的匹配框架。前者基于句子嵌入进行匹配，后者基于上下文-回复交互进行匹配。随着预训练语言模型在自然语言处理领域的多个任务上不断取得突破，许多研究人员开始尝试使用基于预训练语言模型的匹配框架去建模上下文-回复匹配模型，这种方法建模的多轮对话模型可以表现出优秀的性能。接下来将介绍这三种框架国内外主要的研究工作。

（1） 基于表征的方法

基于表征的匹配方法通常遵循表征匹配范式，由表征层和匹配层组成。在表征层中，上下文和回复通过表征函数表示成向量，再通过聚合函数将上下文中所有语句的向量融合到一起，形成上下文级别的向量。最后，匹配层通过计算上下文级别的向量与回复向量之间的匹配分数。该框架的实现包括表示函数、聚合函数和匹配函数的定义，可以根据不同的函数来构造不同的对话模型。在检索式对话方法发展初期，Lowe 等人提出了一个基于 RNN 的对话模型[17]，该模型将对话上下文以拼接的方式作为输入，然后利用 TF-IDF、RNN 或长短期记忆网络（Long Short-Term Memory, LSTM）作为编码器，将对话中的话语和回复表示成向量，最后计算表示向量之间的匹配分数并进行排序输出。Inaba 等人将对话中的话语和回复的向量表示和交互都交给 RNN

完成，进一步提高了对话模型的表现[18]。Zhou 等人提出了一个多视角（Multi-view）模型[19]，该模型综合了词序视角和话语序列视角两种不同视角的信息进行联合建模，在公共语料库上进行的实验表明显著优于其他基准模型。

（2） 基于交互的方法

基于交互的对话方法是将对话中的话语和候选项中的回复进行交互计算，充分提取对话历史的特征信息。该框架通常遵循表征匹配聚合范式，模型结构大致可分为编码层、交互层与聚合层。

在基于交互的匹配方法上，最早的工作是 Wu 等人提出的顺序匹配网络模型[20]，作者注意到只是将话语进行连接或者将回复与高度抽象的上下文向量相匹配会丢失话语之间的依赖或者重要的上下文信息。顺序匹配网络模型首先将应答与上下文中的每个句子在多粒度的级别上进行交互，然后利用卷积和池化运算计算出上下文中的特征信息。接着将这些信息按顺序输入 RNN 中得到匹配向量的聚合表示，最后根据 RNN 的隐藏状态表示计算出对应的得分。在此工作之后，Zhang 等人提出一个深层话语聚合模型[21]，该模型注意到上下文中最后一轮话语的重要性，将其在输入时与每个话语进行连接，然后输入至注意力机制中计算出语义信息，将应答与对话历史进行交互，利用 RNN 进行匹配信息的累积，最后得出具体的匹配分数。随着 Transformer[22]在机器翻译任务中大放异彩，Zhou 等人受其启发，提出了深度注意力匹配模型[23]。该模型首先在不同级别上给出了对话中每个话语的向量表示，然后利用多个自注意力机制计算出话语之间的匹配信息，将两部分信息输入三维卷积神经网络和池化层进行特征聚合。Gu 等人提出了一个交互式匹配网络模型[24]来建模多轮对话任务。该模型首先从三个方面构建词汇表征，其次设计了分层循环注意力编码器，其可以对历史对话进行分层编码，并通过注意力机制生成更多描述性表征。Tao 等人提出了基于深度交互网络模型[25]，旨在进一步提取对话历史中的匹配信息。该模型将上下文中每一个话语和候选回复进行配对，然后送入堆叠的交互块中进行匹配信息的累积，最后将这些匹配信息沿交互块链传播，得出最终的匹配分数。Tao 等人将词级、n-gram 级和句子级表征融入到深层神经网络中进行匹配，并研究每个表征为匹配做的贡献。根据不同表征进行融合和匹配计算的时间顺序提出了多表征融

合网络模型[26]，并在早期、中期或最后阶段将表示融合至匹配中。评估结果表明后期融合总是优于早期融合。Yuan 等人提出一种多跳选择器网络模型[27]来解决过多的语境信息造成模型性能下降的问题。该模型首先使用多跳选择器来选择相关的话语作为上下文，然后将选择好的话语与应答进行交互计算，最后获得对应的得分。实验结果表明，模型在三个公共多回合对话数据集上的表现优于一些最先进的方法。

（3） 基于预训练语言模型的方法

最近，预训练语言模型[28]由于其强大的语言表示和理解能力，在各种下游任务中显示出了令人印象深刻的优势。目前比较有代表性的预训练模型为：ELMo[29]、BERT[30]、XLNet[31]、GPT-2[32]、ELECTRA[33]和 ALBERT[34]等。研究人员通常基于输入表示、编码器结构、预训练任务等多个方面对预训练模型进行改进以提出更优的模型，而其工作流程可大致分为预训练和微调两个阶段。预训练语言模型因其通用的输入表示和经过简单微调即可显著的提高下游任务的表现而大受欢迎。

一些研究人员试图将预训练语言模型应用于多轮对话。通过将上下文中的所有话语和候选回复连接到预先训练好的多层自注意力网络，可以通过模型中的注意力机制执行表示、交互和聚合等操作。Whang 等人在将 BERT 应用到多轮对话的基础上，提出了一种高效的基于特定语料库的后训练方法[35]，可以有效帮助模型训练一般语料库中从未出现的上下文表示和单词。Lu 等人提出一种说话者区分方法旨在区分不同的说话者信息，还提出一种对话增强的方法，在不同时间点切断真实对话以扩充训练语料，两种方法有效地提高了模型的表现[36]。Gu 等人提出了一种新的说话者感知 BERT（Speaker-Aware BERT，SA-BERT）模型[37]，可以感知对话人的变化信息。文章还提出一种对话人解缠策略用来处理纠缠的对话，该策略根据对话人的信息选择少量最重要的话语作为过滤上下文，最后进行领域自适应，将领域内的知识整合到预训练语言模型中。Whang 等人提出一个话语操作策略（Utterance Manipulation Strategies, UMS）[38]来学习话语之间的顺序性和时间依赖关系。其中 UMS 是一种自监督的方法，包含插入、删除和查询三种策略，有助于回复选择模型保持对话的连贯性。Xu 等人为对话模型设计了四个自监督任务：下一个会话预测、话语恢复、非相关检测和一致性判别，以多任务的方式与这四个自监督任务联合训练基于预训练语

言模型的对话模型[39]。Han 等人提出了一种新的细粒度的后训练（Fine-grained Post-training, FP）方法[40]，该方法通过训练每个简短的上下文话语回复对来学习话语层面的交互，并通过一个新的训练目标-话语关系分类来帮助模型理解话语之间的语义关系和连贯性。实验结果表明，模型在多个数据集上取得了最好的表现，这意味着细粒度后训练方法对于回复选择任务非常有效。

综上所述，基于交互的模型在性能上通常优于基于表征的模型，这是因为基于交互的模型允许上下文信息和回复在输入时就进行交互，因此匹配的信息可以得到充分保留。此外，基于预训练模型的方法在上下文和候选回复之间进行全面交互，并通过多个自监督的预训练任务在大规模语料库上进行预训练，因此，通常显著优于传统的基于表征和基于交互的模型。但是，基于预训练模型的方法由于参数规模庞大，导致其计算量大，就效率和成本而言要比基于表征和基于交互的方法高。目前基于预训练模型的方法大都简单地将上下文和回复拼接作为编码器的输入，没有考虑从多个角度对上下文中的信息进行提取，导致缺少重要的匹配信息。此外现有方法大都采取简单策略，在预训练模型顶层添加分类器，未对上下文话语和回复之间的信息进一步地提取，模型效果有待提升。

1.3 主要研究内容

本文在预训练语言模型的基础上，从模型的对话适应训练、多轮对话模型架构和外部知识的引入等多个角度，构建一个基于预训练语言模型的知识驱动的多级匹配检索式对话模型。为充分证明本文方法和模型的有效性，本文在三个公开的数据集上进行了多组对比和消融实验。

综上所述，本文主要研究内容可分为以下三点：

（1）考虑到预训练语言模型在特定领域的对话语料库表示上下文信息有一定的局限性，故本文提出新的训练任务-对话匹配来帮助预训练语言模型进一步理解和适应对话领域。

（2）本文基于预训练语言模型的基本结构从话语和单词级别同时对多轮检索式对话进行建模，并为两个模型设计了合适的匹配网络框架，融合多个级别的上下文回

复之间匹配信息，以此提高模型的性能。

（3）考虑到对话模型缺少相关外部知识的引入，故本文在对话语料之上构建了知识图谱，为对话模型提供外部知识，进一步提升模型性能。

1.4 论文组织结构

本论文的组织结构如下所示，共分为五个章节：

第一章为绪论。首先介绍了人机对话技术的研究背景及意义，其次阐述了国内外关于对话系统研究的相关工作，根据检索式多轮对话现有的问题提出本文主要的研究内容，最后列出论文的组织结构。

第二章为面向对话领域的预训练语言模型分析。首先从输入的词向量表示、编码器结构、预训练任务三个方面对预训练语言模型进行介绍，然后根据预训练模型存在的问题对预训练任务进行改进使其适应对话语料，以提高对话模型的整体性能。

第三章为知识驱动的多级匹配检索式对话。首先对多轮检索式对话任务进行了形式化定义，然后分别从话语和单词级别两个角度介绍对话模型具体的流程。最后融合两种级别的匹配信息以此提高对话系统的性能。此外，在对话语料之上构建了知识图谱，为检索式多轮对话系统引入外部知识。

第四章为实验数据和结果分析。首先介绍了实验中使用到的公开数据集、评价指标和基准模型，其次对实验环境和模型训练参数作了具体阐述，最后列出多组对比和消融实验数据，并对其进行分析总结，以此验证本文所提的方法的有效性。

第五章为总结与展望。主要对整篇文章的工作内容进行了总结，并结合本文工作的不足之处对未来检索式对话方法的研究进行了展望。

2 面向对话领域的预训练语言模型分析

预训练语言模型一经出现，便在自然语言处理的多个任务上取得了最好的效果。大量研究表明，预训练语言模型在大量的文本数据上进行预训练之后可以学习强大的表示和理解能力，只要简单的调整就可以提高下游任务的表现。虽然预训练语言模型非常强大，但因为是在通用文本上训练得到的，例如图书语料、维基百科等，对于特定领域的对话语料库的上下文信息提取仍有一定局限性。例如，Ubuntu 语料库是用于评估多轮检索式对话系统的最常用语料库，包含了一些通常不会出现在通用语料库中的术语和 Ubuntu 命令，例如 apt-get、lsmod 和 grep 等。因此，为提高预训练语言模型在特定领域语料库上的上下文信息表示能力，本章考虑在 Ubuntu、Douban 和 E-commerce 语料库上结合预训练任务进行预训练语言模型的后训练，以此来提高多轮检索式对话模型的表现。对话语料经过语言表征的学习可以转换为词嵌入向量，这些向量输入上下文编码器就可以形成对应的上下文表示。现有预训练语言模型的预训练任务仍存在一些问题，对于领域的适应性帮助不足，因此，利用一种新的对话匹配训练任务来帮助预训练语言模型很好地适应对话领域，从而提高模型在回复选择任务上的表现。

2.1 不同嵌入方式对模型性能的影响

自然语言处理中的下游任务需要一个好的表征来进行更有效的学习，这些表征在自然语言处理领域通常称为嵌入（Embedding）或者分布式向量（Distributed Vectors）。早期的向量表示使用一个维数为所有单词数量的独热编码，这样会导致矩阵维度过大，计算复杂。词嵌入就是将这种高维的向量嵌入到低维中，简化了矩阵的计算复杂度。这些表征的学习，在早期研究中是通过由人工标注的数据集来进行监督训练进行的，但是有标注的数据集资源短缺并且耗时耗力，对于一个复杂的模型来说容易过拟合且泛化能力弱。在预训练语言模型问世之后，这个问题得到解决。对于预训练语言模型来说，输入文本的向量表征从最初的浅层的词嵌入逐渐发展到后来的深度编码，表征中包含的信息越来越丰富。接下来本节将主要按照这两种阶段介绍预训练语言

模型的两大范式：浅层词嵌入（Non-Contextual Embeddings）与上下文嵌入（Contextual Embeddings）。图 2.1 展示了自然语言处理的通用神经网络结构。

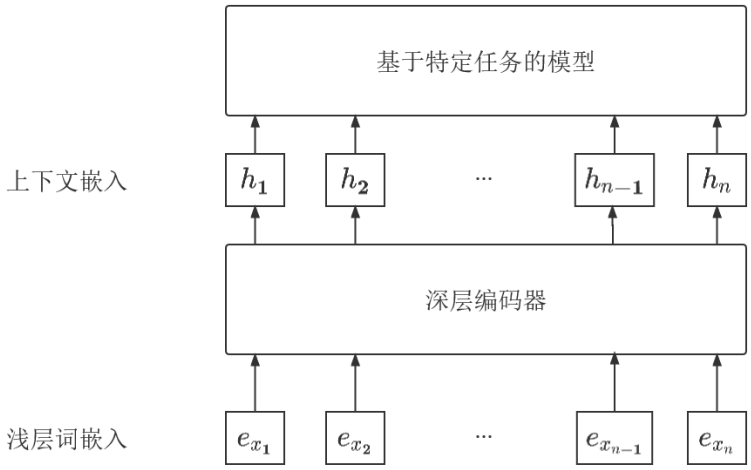

图 2.1 自然语言处理的通用神经网络结构

2.1.1 浅层词嵌入

浅层词嵌入通过所有单词的数量和单词出现的频率计算出单词的向量表示。这种计算方式得到的单词向量表示与整个序列的依赖关系不大。虽然这种方法能够捕获词的语义信息，但是无法根据上下文进行动态变换，因此称词向量为静态、上下文无关。具体做法是将词汇表中的每一个词汇映射到一个查询表中，而进行预训练的过程就是查询表生成的过程。这样，每个词汇的独热编码（One-hot Encoding）通过乘以查询表即可得到属于该词的词向量。形式上，给定一个文本序列$\mathrm{X} = \{x_1, x_2, \ldots, x_N\}$，对于属于词汇表 V 的每个单词 x，通过查找表$\mathrm{E} \in \mathbb{R}^{D_e \times |V|}$将其映射为词向量$\mathrm{e}_x \in \mathbb{R}^{D_e}$，其中$\mathrm{D}_e$是一个超参数，代表词嵌入的维度。通过这种操作，文本序列就可以转换为词嵌入序列$\mathrm{e} = \{e_{x_1}, e_{x_2}, \ldots, e_{x_N}\}$。

浅层词嵌入是第一代预训练语言模型的产物，有着两个明显的缺陷：第一个是无法处理一词多义的情况，在文本序列中同一个单词虽然出现在不同位置，但是对应的向量表示是相同的；第二个就是出现 OOV（Out of Vocabulary）的情况，因为语料中不可能包含所有的词汇，一旦遇到陌生的词，就无法通过查找表来得到词向量。为解决此问题，字符级表示或者子单词表示被应用于许多自然语言处理任务。

2.1.2 上下文嵌入

为了解决一词多义和单词对于上下文的依赖性，单词的向量表示需要根据在不同语境而动态改变。给定一个长度为 N 的文本序列$X = \{x_1, x_2, \dots, x_N\}$，$x_i$的向量表示由整个文本序列来决定。

$$[h_1, h_2, \dots, h_N] = f_{enc}(x_1, x_2, \dots, x_N) \tag{2.1}$$

其中$f_{enc}(.)$函数就是图 2-1 中的深层编码器，而输出的h_i为词元x_i的上下文嵌入或者说是动态嵌入，因为x_i会根据上下文信息而动态改变。

预训练语言模型的经典代表有：ELMo、GPT 和 BERT 等。语言模型嵌入（Embedding from Language Models，ELMo）对于一词多义给出了很好的解决方案。之所以存在一词多义问题，就是因为词嵌入是静态表示，当具体使用时就会受限。ELMo 的基本思想是先学习一个和原始词嵌入一样的词向量，然后根据不同语境来动态地改变向量表示，这样最终形成的向量表示就不存在一词多义了。ELMo 结构上是由两个长短期记忆网络构成，具体结构如图 2.2 所示。

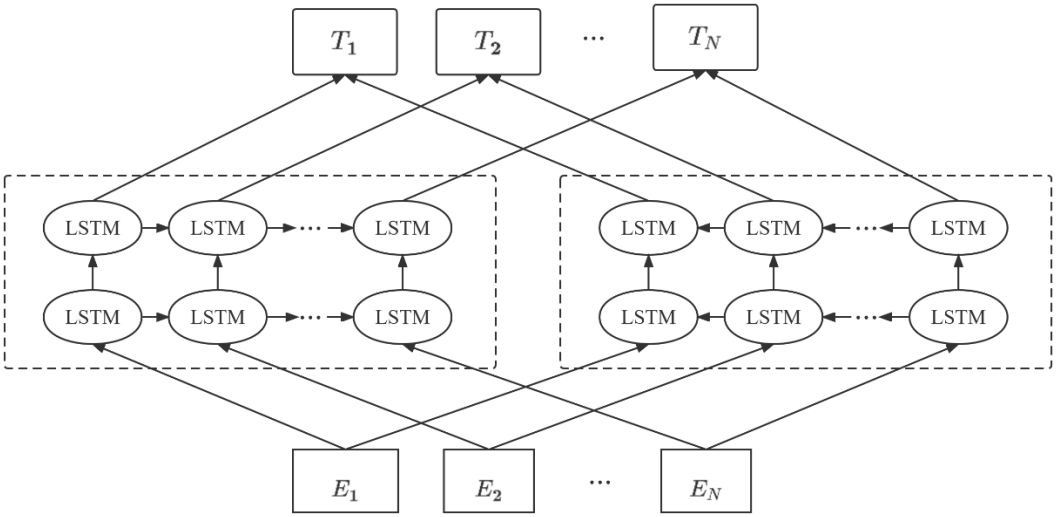

图 2.2 ELMo 结构

模型左端为正向长短期记忆网络编码器，与之对应的右端的反向长短期记忆网络编码器。ELMo 在经过训练之后，输入的文本序列经过两个长短期记忆网络编码形成对应的词向量表示，这就可以提供给下游的自然语言处理任务。

GPT 主要结构由 Transformer 的解码器组成，具体如图 2.3 所示。GPT 虽然进一

步提取了文本的特征信息，但是只正向地对文本进行了处理，未考虑到逆向的文本信息，这也就带来了一定的信息损失。

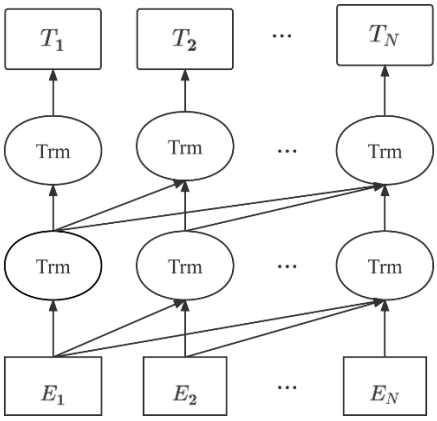

图 2.3　GPT 结构

基于 Transformer 的双向编码器表征（Bidirectional Encoder Representations from Transformer, BERT）结合了 ELMo 和 GPT 两者的优点，不仅使用了双向结构，并且使用了 Transformer 的编码器来抽取文本中的信息，本质上就是双向的 GPT，结构如图 2.4 所示。预训练语言模型 BERT 提高了多个下游任务的表现，是一项具有里程碑式的工作。GPT 和 BERT 工作流程大致相同，即首先在大量文本数据上进行模型预训练，然后在微调阶段根据具体的下游的自然语言处理任务来进行调整。两者的模型结构都基于 Transformer，但是存在单向和双向结构之分。BERT 作为一个双向模型适合做文本分类的相关任务，而 GPT 作为单向更加适合做语言生成的任务。

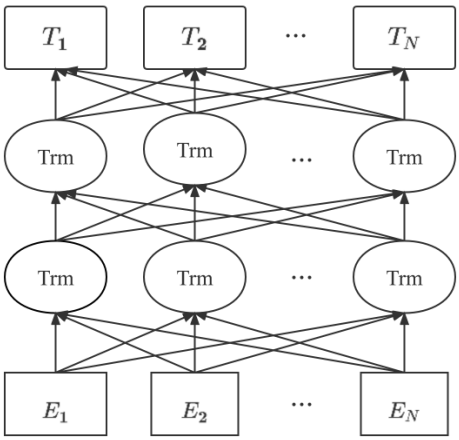

图 2.4　Bert 结构

2.2 不同编码方式对模型性能的影响

在自然语言处理任务中，编码器的作用是将一个长度不定的文本序列转变成一个定长的隐藏变量。经过上一节的介绍，上下文无关的词向量经过深度编码器之后就会根据上下文而动态改变，形成不同意义的词嵌入表示。接着，本节将具体分析输入文本的不同编码方式对于模型性能的影响，并介绍编码器的具体结构。

2.2.1 编码方式

一般情况下，预训练语言模型的输入为一段长序列文本，该文本通常利用[CLS]和[SEP]进行连接。在检索式对话方法中，具体的编码方式有三种，分别是表征式编码、交互式编码和预训练式编码。给定一段对话历史$c = \{u_1, u_2, \dots, u_n, r\}$，其中$u_i$为上下文中的一段话语，$r$ 为对应的回复。具体地，表征式编码将对话上下文和回复分别输入至编码器中进行编码表示，可表示为$[CLS]u_1[SEP]u_2 \dots [SEP]u_n[SEP]$和$[CLS]r[SEP]$；交互式编码将对话上下文中的每一个话语和回复分别输入至编码器中进行编码表示，可表示为$[CLS]u_i[SEP]$和$[CLS]r[SEP]$，其中$0 < i \leq n$；预训练式编码将上下文话语和回复利用特殊符号[CLS]和[SEP]连接成一个长序列，可表示为：$[CLS]u_1[SEP]u_2 \dots [SEP]u_n[SEP]r[SEP]$。

表征式编码考虑到了上下文话语和回复中单独的语义信息，但未考虑到每一轮话语中独特的语义信息，而交互式编码弥补了这一缺陷，其将上下文中每一轮话语单独作为输入，提取了局部的语义信息，但也未考虑到全局的语义信息。上述两者编码方式均未考虑上下文和话语之间的语义关系，只是单独提取两者的语义信息，因此，预训练式编码进一步弥补了这种缺陷，其将上下文话语和回复连接之后进行编码，这样对话历史中的每一个词都可以和整个文本中的任何一个词进行交互，可以获取到更加全面的语义或者句法特征信息。

2.2.2 编码器结构

预训练语言模型BERT主要结构由多层的Transformer编码器组成，而Transformer编码器结构主要由多层相互叠加而成，每个层都包括两个子层，分别是多头注意力机制和前向反馈层，具体结构如图 2.5 所示。

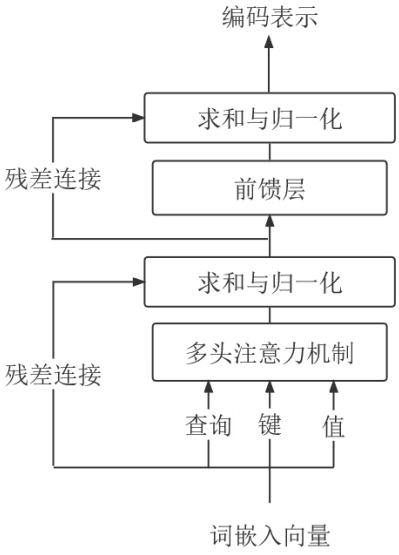

图 2.5　Transformer 编码器结构

Transformer 通过不同的线性映射分别将查询（Query）、键（Key）和值（Value）映射到d_k、d_k和d_v维度。然后在每个 Query、Key 和 Value 的映射上同时执行注意力机制，最后得到一个d_v的输出矩阵。通过将这些矩阵进行级联和再次线性映射，可得到最终的编码表示。这种设计称为多头注意力机制，结构如图 2.6 所示。

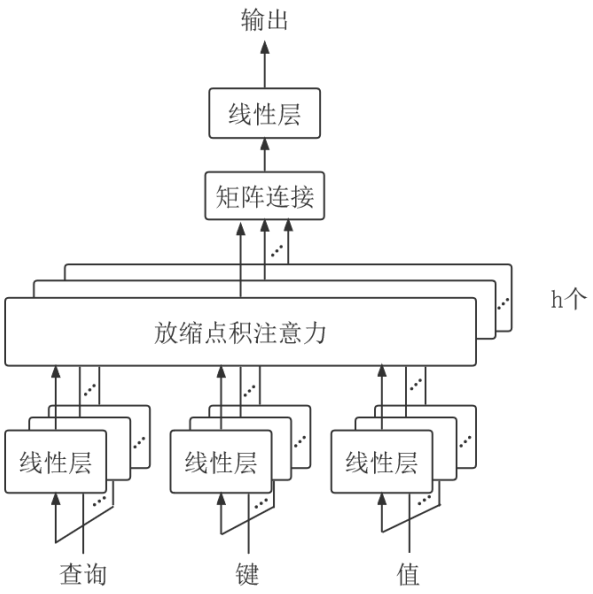

图 2.6 多头注意力机制

多头注意力机制捕获不同表征潜在空间中不同位置的特征，若只有单个注意力头，

平均化的信息无法体现这一点。

$$\begin{aligned}\text{MultiHead}(Q,K,V) &= Concat(head_1,\dots,head_h)W^O \\ where\ \ head_i &= Attention\left(QW_i^Q, KW_i^K, VW_i^V\right)\end{aligned} \tag{2.2}$$

其中输入中的 Q、K、V 和输出对应的学习矩阵为$W_i^Q \in \mathbb{R}^{d_{model}\times d_k}$，$W_i^K \in \mathbb{R}^{d_{model}\times d_k}$，$W_i^V \in \mathbb{R}^{d_{model}\times d_v}$和$W^O \in \mathbb{R}^{hd_v\times d_{model}}$，这里$d_{model}$代表输出矩阵的维度，Concat 表示矩阵连接，Attention 表示注意力函数，而 h 代表注意力头部的数量。

注意力机制的主要功能是将输入的键（Key）、值（Value）和查找（Query）计算为输出，其中的 Key、Value、Query 和输出是以单维矩阵形式进行计算的。Query 和相应的 Key 进行计算得到 Value，再通过将 Value 进行求和计算得到输出。Transformer 中较为重要的结构为放缩点乘注意力，结构如图 2.7 所示。

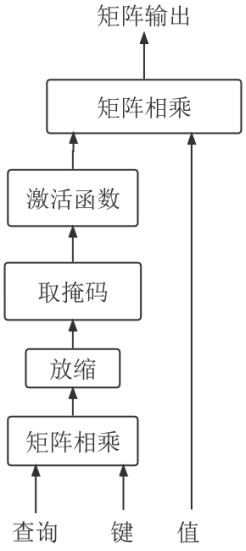

图 2.7 放缩点积注意力

注意力的输入由多个 Query、Key 和 Value 组成，其中 Query 和 Key 向量的维度为d_k，Value 的维度为d_v。通过使用全部的 Key 与 Query 进行点乘，再除以$\sqrt{d_k}$进行维度缩减，让梯度更加稳定，然后应用 SoftMax 激活函数来获得关于 Value 的权重。在实际的计算过程中，将 Query、Key 和 Value 都表示成矩阵 Q、K 和 V，同时来计算注意力函数：

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.3}$$

点乘注意力与加法注意力是两种最常用的注意力机制，在实际计算的适用性上加法注意力与点乘注意力大致相同，但是在实现上点乘注意力可以更好地使用矩阵运算，不仅速度更快，且空间复杂度更低。对于较小的维度d_k，两种注意力机制的效果大致相同，当d_k增大时，加法注意力要比点乘注意力好，而为了减轻这种因为维度增大带来的噪声，对点积通过除以$\sqrt{d_k}$进行了缩放。

编码器第二层包含一个前向反馈层（FFN），其中包含两个线性变换，而中间还经过激活函数 ReLU 的转换。

$$\mathrm{FFN(x)} = \max(0, \mathrm{x}W_1 + b_1)W_2 + b_2 \tag{2.4}$$

文本序列经过嵌入层学习成为词嵌入向量，由编码器进行双向编码学习文本的上下文信息，经过多个预训练任务训练之后，编码器的最后隐藏状态对应于上下文嵌入。接下来将介绍预训练语言模型训练过程中用到的两个预训练任务。

2.3 对话领域后训练

由于预训练语言模型是在大量文本数据上训练得到的，对于特定的对话领域适应性不足，因此需要利用其原有的两个预训练任务对其进行对话领域的后训练。但预训练任务仍存在一些问题，对领域适应的帮助不大，因此提出对话匹配训练任务来帮助预训练语言模型充分适应对话领域。

2.3.1 预训练任务

预训练语言模型 BERT 提出两个预训练任务来帮助模型训练以提高模型在下游任务中的表现，这两个预训练任务分别是掩码语言建模（Masked Language Model, MLM）和下一句预测（Next Sentence Prediction, NSP）。为了帮助预训练语言模型适应对话领域，在三个公开对话数据集上利用 MLM 和 NSP 来再次训练预训练语言模型，这个过程称为领域后训练（Domain Post-training, DPT）。接下来，本小节将具体介绍这两个预训练任务的内容，对这两个任务存在的问题进行总结分析。

（1） 掩码语言建模

掩码语言建模任务类似于完形填空，通过预先设置好的百分比使用 Mask 替换词元序列中的一些词元，然后推测这些被替换词元。该任务的训练样本结构如图 2.8 所

示。这种推测方式通过同时考虑文本正向和反向的信息来推测出被替换的词元，可以很好地学习文本中的信息。由于在预训练阶段输入的词元序列存在屏蔽词，而在微调阶段输入的词元序列是完整的，这就造成预训练微调两阶段不一致的问题。为此，预训练语言模型 BERT 使用这样的屏蔽方式：首先在随机选择 15%的词元当中，选择80%的词元进行掩码，10%的词元替换为文本中的任意一个词元，10%的词元不进行任何操作。在随机选择 15%的词元中 10%的词元被替换为文本中的任意一个词元，当推测原本这个位置上的词元的时候，这个过程就类似文本纠错。在随机选择 15%的词元中 10%的词元不进行任何操作，微调阶段不需要进行预测，这就缓解了微调时仍需预测词元的问题。

图 2.8 掩码语言建模任务训练样本

掩码语言建模任务通过直接对单个词元进行随机掩码，这对于中文语料来说会丢失很多实体和短语信息，并且该任务仅预测被随机掩码的词元，而其他的词元并未参与预测过程。除此之外，该任务的掩码策略是静态的，即在数据处理阶段将训练数据进行屏蔽，这种屏蔽方式导致学习的上下文信息有限。由于该任务只能预测被屏蔽的词元，导致训练的效率过低，计算算力消耗过大，模型的收敛速度过慢。

（2） 下一句预测

自然语言处理领域存在多个句子级任务中，例如问答和自然语言推理都是需要学习句子间的顺序关系，但是单纯的依靠掩码语言建模是不足以解决这个问题的，故预训练语言模型 BERT 提出 NSP 预测任务，该任务的目的是为了得到一个可以区分句子顺序的模型，并且该任务可以基于单个句子的数据集进行。具体来说，下一句预测任务就是在语料库中为训练样本选择出两个句子，其中 50%的概率选择的第二个句子是第一句后面真实的下一句，记为 isNext。而另外一半的概率是在数据文档中任意选择一个作为第一句的下一句，记为 notNext。该任务的训练样本结构如图 2.9 所示。编码器输出的词元序列首位的词元 C 就代表预测的概率。通过预测句子乙是否为句

子甲的下一句来帮助预训练语言模型学习句子级别的信息。

图 2.9 下一句预测任务训练样本

下一句预测任务作为二分类任务，负样本构造方式过于简单，导致预训练语言模型无法充分训练，并且该任务对于下游任务的作用要比 MLM 任务小。

2.3.2 对话匹配任务

掩码语言建模与下一句预测两个预训练任务分别从单词级别和句子级别帮助预训练语言模型进行训练，充分理解通用文本语料中的语义信息。但是由于多轮对话中包含许多潜在的上下文信息，预训练语言模型想要直接利用通用文本来表达特定领域的对话上下文信息还是有一定局限性。因此，本文在三个对话语料库 Ubuntu、Douban 和 E-commerce 上结合一种改进后的预训练任务-对话匹配（Dialogue Matching, DM）来对预训练语言模型进行后训练，与从头开始训练不同，后训练是在已经训练好的预训练模型之上继续训练，这可以帮助模型更好地适应对话领域，提高模型在多轮对话任务上的表现。

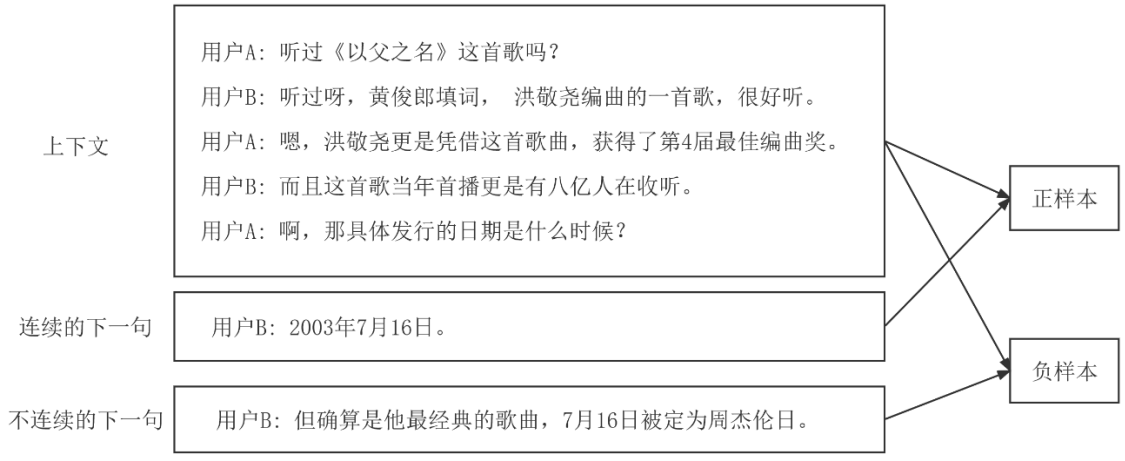

图 2.10 对话匹配样本选择

具体地，对话匹配预训练任务是通过在一段多回合对话中选择多个正负样本，其中正样本为对话中连续的一段，而负样本为对话中的两段上下文不一致对话，其主要

目的是预测样本中的回复是否为给定上下文的下一句，具体形式如图 2.10 所示。

形式上，给定一段多回合对话$c = \{u_1, u_2, \ldots, u_n\}$，通过选取正负样本：正样本P = $\{u_1, u_2, \ldots, u_i\}$，其中$i \leq n$；负样本$N = \{u_1, \ldots, u_j, u_{j+2}\}$，其中$j + 2 \leq n$。当样本进行输入表示时，需要将最后一句话作为回复，具体形式：[CLS]$u_1[SEP]u_2[SEP] \ldots u_k[SEP]r[SEP]$。经过编码器之后，将[CLS]对应的编码器最后隐藏状态送入全连接层和激活函数得到最后匹配的结果，结果为$y_{dm} = 1$表示回复与上文相关，反之无关。

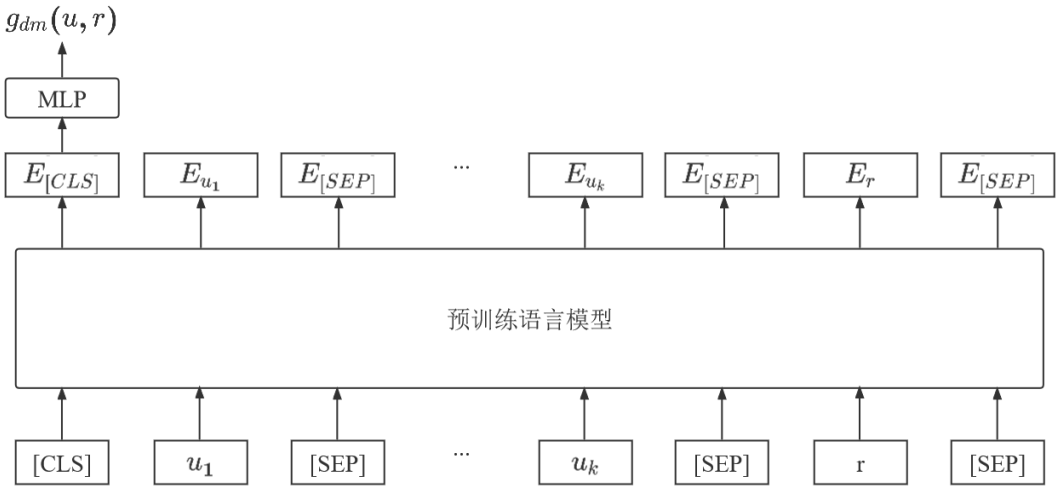

图 2.11 对话匹配任务训练过程示意图

预训练语言模型结合对话匹配预训练任务进行后训练的过程如图 2.11 所示。具体地，首先使用[CLS]和[SEP]将样本中的上下文话语和回复连接成一个长序列，然后经过词嵌入学习后送入预训练语言模型编码器中进行编码，再取出编码器中最后一层的潜在状态作为输入文本的语境表示 E。

得到整个序列的上下文表示 E 后，取出序列中第一个特殊符号[CLS]对应的上下文表示$E_{[CLS]}$。CLS 符号通过编码器中的自注意力机制来获取句子级别的信息表示，在经过训练之后，CLS 会聚合每个句子的上下文表示。因此，CLS 对应的上下文表示$E_{[CLS]}$可以直接用来作句子分类任务，而在对话匹配任务中也是如此，CLS 可以直接用来判断给定的话语是否为上下文的下一句，如果是，则是正确回复，反之不是。上下文话语回复对中的语义交互信息聚合成$E_{[CLS]}$表示向量，将该向量送入全连接层

和激活函数即可得到对应的匹配分数，该分数即可用作分类结果。

$$g_{dm}(c,r)=\sigma\left(WE_{[CLS]}+b\right) \tag{2.5}$$

其中σ(·)为 sigmoid 激活函数，W和b分别为全连接层的可训练参数矩阵和偏置，而g_{dm}就表示对话匹配预训练任务的匹配分数，取值范围为（0，1）。

对话匹配预训练任务的目标函数可以表示为：

$$\mathcal{L}_{dm}=-y_{dm}\log\left(g_{dm}(c,r)\right)-(1-y_{dm})\log\left(1-g_{dm}(c,r)\right) \tag{2.6}$$

对话匹配预训练任务可以有效地帮助预训练语言模型适应对话领域，在具体的训练过程中，将原始的 NSP 任务转变为对话匹配任务。NSP 预训练任务判断给定的两句话是否为连续的，其中正样本由数据集中的连续的两个句子构成，而负样本由数据集不同文档中的两个句子构成，正负样本的抽样概率相等。在关于预训练语言模型的研究[34]中发现，NSP 任务对下游任务的影响很小。该研究推测，与 MLM 任务相比，NSP 效率低下的原因是其作为训练任务缺乏一定难度，其将文本主旨预测和一致性计算合并在一个任务中。然而，文本主旨预测相较于一致性计算更容易学习，并且与 MLM 损失学习到的信息有一定的重叠。

由于 MLM 任务对于中文语料以汉字为基础进行随机屏蔽，这样会丢失很多短语与实体信息，因此引入两种级别的屏蔽方式：短语级别和实体级别。短语级别的屏蔽方式主要通过分词工具对文本序列进行分词来获得短语信息，进而随机屏蔽，而实体级别的屏蔽方式则是对文本序列中地点名称、人物名称和机构名称等实体进行随机屏蔽。通过这两种屏蔽方式可以为预训练语言模型提供很好的外部知识，进而提升模型在回复选择任务中的表现。此外，为充分提取对话中的上下文信息，本文将 MLM 中的静态掩码策略替换为动态掩码策略，主要区别是：静态掩码在数据处理时就已经确定好了掩码的位置，然后直接提供给模型进行训练；动态掩码在模型进行训练时动态地确定数据中掩码的位置。通过动态掩码的方式可以确保一个样本在每次训练过程中掩码位置都不同，这有助于模型学习到更多的上下文表示。

对于回复选择任务来说，句子级别的预训练任务是帮助预训练语言模型语义理解的一个重要影响因素。对话匹配任务需要模型预测给定的上下文和话语是否连续，这可以很好地提取出对话中的顺序信息和时间依赖关系。预训练语言模型在学习到这

些信息之后对于回复选择任务的适应性增强，有助于提高对话匹配模型的性能。

2.4 本章小结

本章首先分析了不同嵌入方式和不同编码方式对于模型性能的影响。接着为了解决预训练语言模型特定领域适应性不足的问题，本章提出在三个数据集上对预训练语言模型进行领域后训练的方法，并对两个任务的不足之处进行了详细分析。为了进一步适应对话领域，本章还提出了一个新的对话匹配训练任务，该任务预测给定的语句是否为给定上下文的下一句，这与回复选择任务非常相似。因此，利用对话匹配任务在数据集上对预训练语言模型进行后训练可以很好地帮助其适应回复选择任务。

3 知识驱动的多级匹配检索式对话

人机对话系统旨在实现自然、一致的对话以满足用户的需求。一般来说，对话系统可分为任务型对话系统和非任务型对话系统，而非任务型对话系统的实现方式可分为基于生成式的对话系统和基于检索式的对话系统。基于生成式的方法通过将文本输入至编码器-解码器结构中来学习文本信息，结构简单清晰，无需过多人力去构造模型，但是基于生成的方法通常评估较为困难且生成的对话会单调无趣，对数据集依赖性较强，因此使用这种方法构建对话系统有一定难度。本章将重点研究基于检索式的对话系统。

3.1 问题描述

基于检索式的对话方法主要过程是给定一段对话上下文，通过在多个候选回复中检索出一个最合适的回复作为输出。随着深层神经网络的快速发展，各种深度学习模型如卷积神经网络和循环神经网络被应用到单轮检索式对话研究上，使得计算对话回复匹配的准确率得到了很大的提高。之后的研究工作更多的关注多轮对话，但由于多轮对话存在丰富的上下文信息，如果直接使用单轮检索式对话模型就会丢失很多上下文信息导致模型效果变差，因此选择最符合上下文的回复成为了问题的关键。多轮检索式对话的构建方式可大致分为基于交互的方法与基于预训练语言模型的方法。在多轮对话上的大多数研究都是基于交互的方法，大都利用现有的神经网络设计出复杂的匹配模型，通过将上下文话语和回复以多种方式进行匹配，然后计算出匹配分数。但是由于 RNN 结构的简单性，无法记忆过长的文本序列，因此当对话上下文语句长度过大，模型的效果就会显著变差。

预训练语言模型由于其卓越的表示和理解能力在自然语言处理领域的多个任务上取得了全面突破，故越来越多的研究都在探索如何将预训练语言模型与检索式方法进行有效的结合。预训练语言模型是在大量通用的文本数据上训练得到的，只需要对其简单微调即可应用于下游任务，一般是在模型之后加上全连接层和激活函数来做分类或预测任务。但目前基于预训练语言模型的方法大都简单地拼接上下文话语

和候选回复，从单词或字级别的角度去构建模型，忽视了话语级别的信息，未将回复与上下文在每一轮话语进行信息匹配，造成了一定程度的信息丢失，也未尝试将两种级别的信息结合起来同时建模。本章分别从单词级别与话语级别两个角度同时建模，并结合 CNN、RNN 和注意力机制提取两种级别的匹配信息，最后将两者融合，得出上下文-回复匹配分数。这种构建的方法有望提高检索式多轮对话模型的性能。接下来给出检索式对话方法的形式化定义。

给定一个多轮对话数据集$\mathcal{D}=\{(c_1,r_1,y_1),(c_2,r_2,y_2),\ldots,(c_N,r_N,y_N)\}$，其中 N 为数据集中的样本数量，$c_i=\{u_1,u_2,\ldots,u_n\}$表示多轮对话上下文，n 为上下文中话语轮数，$r_i$表示对应$c_i$的候选回复，而$y_i\in\{0,1\}$为一个二元分类标签，代表$r_i$是否为$c_i$的正确回复。其中$y=1$表示$r_i$为合适的回复，反之$y=0$。检索式多轮对话任务的目的是在对话语料$\mathcal{D}$上训练出一个匹配模型$\mathrm{g(c,r)}$，对于任何一对上下文话语与候选回复，模型$\mathrm{g(c,r)}$都可以计算出匹配分数。表 3.1 显示了一个真实的多轮对话数据样例。

表 3.1 多轮对话数据样例

角色-对话轮数	上下文话语
用户 A-1	听过《以父之名》这首歌吗？
用户 B-2	听过呀，黄俊郎填词， 洪敬尧编曲的一首歌，很好听。
用户 A-3	嗯，洪敬尧更是凭借这首歌曲，获得了第 4 届最佳编曲奖。
用户 B-4	而且这首歌当年首播更是有八亿人在收听。
用户 A-5	啊，那具体发行的日期是什么时候？
标签类别	候选回复
正确	2003 年 7 月 16 日。
错误	但确算是他最经典的歌曲，7 月 16 日被定为周杰伦日。

3.2 多级匹配对话模型结构

本节基于预训练语言模型提出一个多级匹配检索式对话模型结构，简称多级匹配（Multi-level Matching，MM）模型。该模型具体可分为两个子模型：话语级别对话

匹配模型和单词级别对话匹配模型。模型结构如图 3.1 所示。整体模型的工作流程为：首先，上下文与候选回复以特定的方式进行组合并将其表示为三种嵌入向量，在相加之后输入至预训练语言模型中计算出对应的上下文表示；其次，将上下文与回复在编码器中对应的隐藏状态送入卷积神经网络或者注意力模块进行上下文与回复之间的信息匹配，将话语中重要的信息表示成匹配向量，送入循环神经网络对其特征进行累积或者聚合；最终将匹配向量对应的隐藏状态送入全连接层和激活函数得到两个子模型的匹配分数，将两者相加得到最终对话模型的匹配分数。接下来，本节将具体介绍两个子模型的实现细节和融合的具体过程。

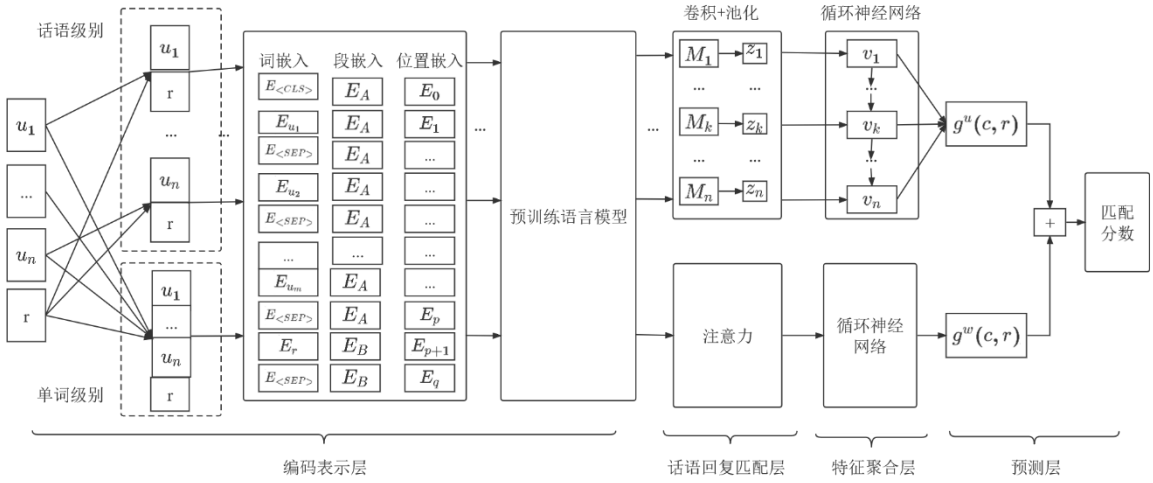

图 3.1 多级匹配模型结构

3.2.1 话语级别对话匹配

受基于交互式方法[20]的启发，本文将其与预训练语言模型结合重新对多轮检索式对话系统建模。首先介绍在话语级别上的建模，话语级别的对话匹配是将上下文中每一个话语与候选回复进行交互然后通过卷积池化进行特征提取，最后通过循环神经网络累积匹配信息。该方法考虑了上下文每个话语-回复对之间的重要局部匹配信息以及上下文中话语之间的关系。通过给定一段上下文$\mathrm{c}=\{u_1,u_2,\ldots,u_n\}$和候选回复r，其中$u_i$为上下文c中的第 i 个话语。通过以[CLS]$u_i$[SEP]和[CLS]r[SEP]的形式送入预训练语言模型 BERT 中进行编码形成上下文表示，其中[CLS]和[SEP]为预训练语言模型输入中的特定符号。当上下文-回复对经过编码器表示之后，就可形成隐藏状态：

$$H_U = f_{enc}([\text{CLS}]u[\text{SEP}]) = \left[h_1^u; h_2^u; \ldots; h_{l_u}^u\right]$$
$$H_R = f_{enc}([\text{CLS}]\text{r}[\text{SEP}]) = \left[h_1^r; h_2^r; \ldots; h_{l_r}^r\right] \tag{3.1}$$

其中H_U和H_R为预训练语言模型 BERT 学习到的上下文表示，对应于最后一层的隐藏状态，$H_U \in \mathbb{R}^{l_u \times d}$，$H_R \in \mathbb{R}^{l_r \times d}$，$h_j^u$和$h_k^r$分别代表话语和回复中第 j 和第 k 个单词对应的上下文向量，l_u和l_r分别是话语u_i和候选回复r的序列长度，d为维度大小。

在获得上下文和回复对应的上下文表示H_U和H_R之后，就可以计算出一个序列相似度矩阵M：

$$M = \left[e_{i,j}\right] = \left[(\text{h}_i^u)^T A h_j^r\right] \tag{3.2}$$

其中$0 \le \text{i} < l_u$，$0 \le \text{j} < l_r$，$\text{A} \in \mathbb{R}^{l_u \times l_r}$为一个线性转换矩阵，$e_{i,j}$为矩阵 M 中第 i 行第 j 列的元素。对于∀i，预训练语言模型对位置 i 前的词元之间的连续性及时间关系进行捕获直到第 i 个词元成为一个潜在向量。因此，序列相似度矩阵M是在话语级别上对话语u和候选回复r进行匹配得到的特征矩阵。

接下来将使用卷积神经网络和最大池化来处理序列相似度矩阵M以形成匹配向量v。卷积神经网络最早使用在计算机视觉上用来处理二维图像数据的局部特征，但是自然语言处理领域也随处可见其身影。通常来说，序列就类似一维的图像，以这种方式，卷积神经网络就可以用来提取文本中的特征。最大池化通过对文本中的特征点取最大值的方式来减少上一层参数损失带来的估算值的均值偏差，保留更多的信息。在话语级别模型中，卷积神经网络将序列相似度矩阵M看作是一个输入通道，对其进行卷积和最大池化操作。这里假设卷积神经网络第l层的矩阵M的特征图输出为：

$$z^l = [z_{i,j}^l] \tag{3.3}$$

其中$z^0 = \text{M}$，$0 \le \text{i} < l_u$，$0 \le \text{j} < l_r$，$0 < l \le 3$。通过在卷积层中使用窗口大小为$r_w^l \times r_\text{h}^l$的二维卷积运算将$z_{i,j}^l$定义为：

$$z_{i,j}^l = \sigma\left(\sum_{s=0}^{r_w^l}\sum_{t=0}^{r_h^l} W_{s,t}^l \cdot z_{i+s,j+t}^{l-1} + b^l\right) \tag{3.4}$$

其中$\sigma(\cdot)$为 ReLU 激活函数，$W^l \in \mathbb{R}^{r_w^l \times r_h^l}$和$b^l$为卷积层的训练参数。最大池化操作遵循卷积操作，可被定义为：

$$R_{i,j}^{l} = \max_{0\le s<P_w^l} \max_{0\le t<P_h^l} z_{i+s,j+t}^{l} \tag{3.5}$$

其中P_w^l和P_h^l分别为 2D 池的宽度和高度，$R_{i,j}^{l}$为池化操作后得到的结果。最后特征图的输出被级联并投影到一个维度较少的空间，使用一个线性转换将其计算为匹配向量v。根据上述公式可以了解到，通过从训练数据中得到上下文表示和预训练语言模型参数，有助于识别恰当回复的话语中的序列可能会与回复中的某些序列具有高度的相似性，并在序列相似性矩阵形成高峰点。这些点通过卷积和最大池化运算继续转换和选择，并将上下文中的语义信息传递到下一步需要操作的向量中。这对应了话语级别模型在对话历史中捕获重要信息的关键过程。图 3.2 展示了序列相似性矩阵M通过卷积和最大池化操作转换为匹配向量 v 的过程。

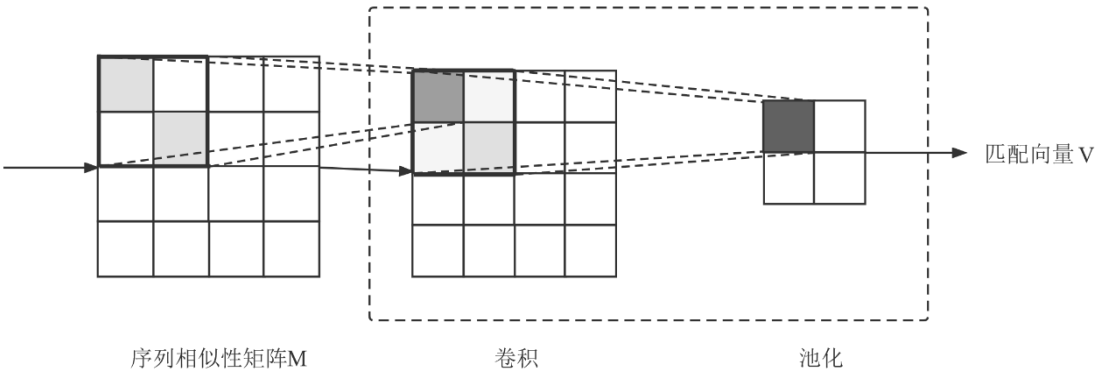

图 3.2 卷积池化过程

在深度学习中，一些经典的神经网络由于梯度消失或者梯度爆发导致很难学习到输入中时间间隔比较长的信息。虽然梯度剪裁的方法可以有效避免产生梯度爆发，但是无法避免梯度消失。而门控循环单元通过引入重置门（Reset）和更新门（Update）来控制信息流动，从而解决了时间序列中时间间隔较长的信息难以提取的弊端。Reset 可以帮助门控循环单元学习到时间序列中短距离的信息，而 Update 对于时间序列中长距离的信息更加敏感。

在得到匹配向量$v = [v_1, v_2, \ldots, v_n]$（这里的 n 对应话语回复对数量）之后，将其作为门控循环单元的输入，经过编码之后就可形成其对应的隐藏状态$H_m = [\mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_n]$。具体计算过程：

$$\begin{aligned} z_i &= \sigma(W_{zv}v_i + W_{zh}h_{i-1}) \\ r_i &= \sigma(W_{rv}v_i + W_{rh}h_{i-1}) \\ h_i' &= tanh\big(W_{hv}v_i + W_{hh}(r_i \odot h_{i-1})\big) \\ h_i &= z_i \odot h_i' + (1 - z_i) \odot h_{i-1} \end{aligned} \tag{3.6}$$

其中z_i、r_i、h_i'和h_i分别对应 Update、Reset、待选隐藏状态和隐藏状态，初始时$h_0 = 0$，σ(·)对应 sigmoid 激活函数，⊙代表点乘操作，1-代表 1 减某个变量，对应公式 3.6 中最后一行，W_{zv}、W_{zh}、W_{rv}、W_{rh}、W_{hv}和W_{hh}为门控循环单元网络中可学习的参数。图 3.3 展示了门控循环单元的具体结构。

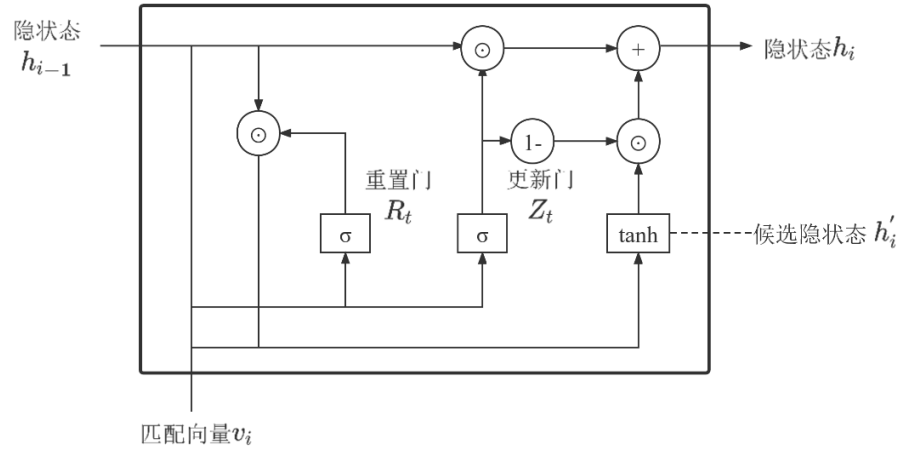

图 3.3 门控循环单元模型

这一层的门控循环单元主要对对话历史中的依赖和时间关系进行提取，并且其作为一个话语序列的匹配，利用时序来保证话语-回复对匹配的信息累积。更新门和重置门主要操纵从先前的潜在状态和之后的输入流向潜在状态的匹配信息量，因此对话历史中关键句子所形成的匹配向量可以被累积下来，并且向量中的负面因素可以被去除。

在经过上述模块处理之后，就可以利用最后的信息进行话语-回复匹配和预测。利用门控循环单元中的隐藏状态$H_m = [\mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_n]$中的最后一个隐藏状态就可以计算出话语级别的对话匹配分数为：

$$g^u(c, r) = \sigma(W_u h_n + b_u) \tag{3.7}$$

其中σ(·)对应 sigmoid 激活函数，W_u和b_u为匹配层的参数和偏置。在模型训练的过程中，通过在对话语料$\mathcal{D}$上最小化交叉熵损失来迭代整个对话模型的参数。使用Θ_u

代表话语级别对话匹配模型的参数，这样目标函数$\mathcal{L}(\mathcal{D},\Theta_u)$被定义为：

$$\mathcal{L}(\mathcal{D},\Theta_u)=-\sum_{i=1}^{N}\left[y_i\log\left(g^u(c_i,r_i)\right)+(1-y_i)\log\left(1-g^u(c_i,r_i)\right)\right] \tag{3.8}$$

其中$y_i\in\{0,1\}$表示回复r_i是否为上下文c_i的二进制标签。

3.2.2 单词级别对话匹配

话语级别的对话匹配考虑了每个上下文话语回复对之间的重要局部匹配信息以及上下文中话语之间的关系。与之不同的是，单词级别的对话匹配是通过将上下文连接成一个长单词序列，并将候选回复与这个长单词序列进行匹配，通过注意力机制来建模两者之间的细粒度交互，最后通过循环神经网络来聚合匹配信息。该方法可有效地关注到上下文与候选回复之间的全局匹配信息。给定一段上下文$\mathrm{c}=\{u_1,u_2,\ldots,u_n\}$和候选回复r，其中$u_i$为c的一个话语。与话语级别方法的输入方式不同，单词级别的对话匹配以$[\mathrm{CLS}]u_1[SEP]\ldots u_n[SEP]r[SEP]$的方式形成词元序列输入至预训练语言模型编码器：

$$\mathrm{H}=f_{enc}([\mathrm{CLS}]u_1[SEP]\ldots u_n[SEP]r[SEP])=[h_0,h_1,\ldots,h_l] \tag{3.9}$$

其中l对应整个词元序列的长度，h_0对应词元[CLS]的上下文表示。以往基于预训练语言模型的方法都是通过将h_0输入至分类器中得出最后的匹配结果，但简单地使用[CLS]上下文表示无法描述话语和回复之间复杂的匹配信息。受注意力机制在基于交互式方法[24]中成功应用的启发，本文尝试将注意力机制与预训练语言模型相结合提取出上下文和回复之间更多的匹配信息以增强回复选择。

根据上下文中的话语和回复在词元序列中的位置信息，从上下文表示H中取出对应的话语和回复表示。对于上下文c中的一个话语u和回复 r，其对应的表示为：

$$\begin{aligned}H_U&=\left[h_1^u,h_2^u,\ldots,h_{l_u}^u\right]\\H_R&=\left[h_1^r,h_2^r,\ldots,h_{l_r}^r\right]\end{aligned} \tag{3.10}$$

其中l_u和l_r分别是上下文中某一话语和回复的长度，h_j^u和h_k^r分别代表话语和回复中第 j 和第 k 个单词对应的上下文向量，$H_U\in\mathbb{R}^{l_u\times d}$，$H_R\in\mathbb{R}^{l_r\times d}$，d 表示维度大小。话语级别方法通过输入[CLS]u/r[SEP]可以直接得到H_U和H_R，而单词级别将话语和回复连接成词元序列输入至预训练语言模型编码器，由于词元序列同时包含上下文和

候选回复，因此可获得更多的上下文表示。

这里主要讨论点积注意力机制和多头注意力机制在单词级别对话匹配模型中的应用。点积注意力机制是用来计算向量相关性的一种简单方法，通过将话语和回复的上下文表示H_U和H_R投影到相同的隐空间中，计算其中每一对上下文表示向量h_j^u和h_k^r的相关性：

$$\theta_j^k = h_j^u (h_k^r)^T \tag{3.11}$$

其中θ_j^k为上下文表示向量h_j^u和h_k^r之间的相关度。之后利用 SoftMax 激活函数计算回复中每个词向量的注意力分数：

$$\eta_j^k = \frac{exp(\theta_j^k)}{\sum_{i=1}^{l_r} exp(\theta_j^i)} \tag{3.12}$$

其中η_j^k表示回复中第 k 个单词的注意力分数。通过将回复中每个单词和其对应的注意力分数相乘，即可得到话语中第 j 个单词对应的回复感知的上下文向量：

$$c_j = \sum_{i=1}^{l_r} \eta_j^i \, h_i^r \tag{3.13}$$

最后利用平均池化（mean）来获得回复感知的话语表示c^R：

$$c^R = mean(c_1, c_2, \dots, c_{l_u}) \tag{3.14}$$

多头注意力的输入由查询（Query）、键（Key）和值（Value）构成。当话语和回复的上下文表示作为这三种输入的集合时，模型可以在相同的缩放点乘注意力上学习到不同的信息和捕获时间序列上不同距离的依赖关系。多头注意力通过使用 h 个独立学习的线性投影来对查询、键和值进行变换，然后将其送入缩放点乘注意力进行信息汇聚，最后将得到的 h 个汇聚矩阵相互连接并输入至线性层中得到最后输出。其中每一个注意力汇聚被称作是一个头。给定话语和回复上下文表示H_U和H_R，话语中每个单词对应的回复感知上下文表示向量为：

$$\begin{gathered} \text{Attention(Q, K, V)} = \text{SoftMax}\left(\frac{QK^T}{\sqrt{d_k}}\right)\text{V} \\ \text{head}_i = Attention\left(H_U W_i^Q, H_R W_i^K, H_R W_i^V\right) \\ C_U = Concat(\text{head}_1, \dots, \text{head}_h) W^O \end{gathered} \tag{3.15}$$

其中Q、K和V分别代表查询、键和值，$W_i^Q \in \mathbb{R}^{d\times\frac{d}{\mathrm{h}}}$，$W_i^K \in \mathbb{R}^{d\times\frac{d}{\mathrm{h}}}$，$W_i^V \in \mathbb{R}^{d\times\frac{d}{\mathrm{h}}}$为多头注意力中第 i 个头的投影矩阵，$d_k = \frac{d}{\mathrm{h}}$，h 代表头的数量，$W^O \in \mathbb{R}^{d\times d}$为连接层对应的输出参数矩阵，作用是将多头注意力汇聚的特征转换为上下文特征，$C_U = [c_1, c_2, \dots, c_{l_u}]$。最后的聚合与点积注意力方式相同，使用公式（3.14）获得回复感知的话语表示c^R。

在获得回复感知的上下文表示$c^R = [c_1^R, c_2^R, \dots, c_n^R]$之后，通过利用每个话语在上下文中的时间位置，将$c^R$依次输入至门控循环单元来聚合上下文信息以提高回复选择的表现。

$$\begin{aligned} z_i &= \sigma(W_{zc}c_i^R + W_{z\mathrm{h}}h_{i-1}) \\ r_i &= \sigma(W_{rc}c_i^R + W_{r\mathrm{h}}h_{i-1}) \\ h_i' &= tanh\left(W_{hc}c_i^R + W_{hh}(r_i \odot h_{i-1})\right) \\ h_i &= z_i \odot h_i' + (1 - z_i) \odot h_{i-1} \end{aligned} \tag{3.16}$$

其中z_i、r_i、h_i'和h_i分别对应 Update、Reset、待选潜在状态和潜在状态，σ(·)对应 sigmoid 激活函数，W_{zc}、$W_{z\mathrm{h}}$、W_{rc}、$W_{r\mathrm{h}}$、W_{hc}和W_{hh}为门控循环单元网络中可学习的参数。上下文表示经过门控循环单元的处理就形成对应的隐藏状态$H^R = [h_1^R, \dots, h_n^R]$，其中最后的隐藏状态$h_n^R$对应于最终的相关性特征。

最后使用相关性特征h_n^R计算出单词级别的对话匹配分数：

$$g^w(c, r) = \sigma(W_w h_n^R + b_w) \tag{3.17}$$

单词级别对应的损失函数为：

$$\mathcal{L}(\mathcal{D}, \Theta_w) = -\sum_{i=1}^{N}\left[y_i \log\left(g^w(c_i, r_i)\right) + (1 - y_i)\log\left(1 - g^w(c_i, r_i)\right)\right] \tag{3.18}$$

3.2.3 模型融合

在推荐、表征学习和其他领域，融合不同角度的信息已被证明是非常有效的策略。在早期基于表示的检索式对话方法中 Multi-view 模型[19]通过将话语级别和单词级别两个视角的信息融合来提高对话模型的性能。输入的上下文话语和候选回复共同表示为语义词嵌入向量，这些向量被两个视角的模型共同使用，以这样的方式互补信息。然后使用两个独立的循环神经网络对两个视角的话语嵌入和单词嵌入进行编码，

分别捕获话语级别的语义与话语信息和单词级别的依赖关系。最后通过线性集成两个视角的信息，联合最小化两个因素，分别是每个视角信息的训练损失和互补视角之间的不一致。通过这种方法可以显著提高对话模型的性能。但是该方法仍存在一些问题：在话语视角中未将上下文话语和候选回复一一匹配，缺少了很重要的交互信息；循环神经网络对于文本序列的编码长度较短，无法记忆过长的文本序列。因此，本文提出的方法正好可以解决上述问题：在话语级别中将上下文话语和候选回复通过注意力机制一一交互匹配，提取出了重要的交互信息；利用预训练语言模型对输入的对话进行表征，这样得到的表示向量包含丰富的上下文信息。

话语级别的对话匹配模型通过将对话历史中每一个话语和候选回复进行匹配形成多个特征向量，然后将这些向量送入门控循环单元进行匹配信息的累积，最后使用最后一层的隐藏状态计算出话语级别的匹配分数。单词级别的对话匹配模型通过将上下文中话语和回复进行串联输入至预训练语言模型进行编码表示，形成的上下文表示中含有更多的上下文信息。这些语境信息经过注意力机制的进一步特征提取和循环神经网络的特征聚合最终形成单词级别的匹配分数。

因此，上下文c和候选回复r对应的话语级别的匹配分数和单词级别的匹配分数分别为$g^u(c,r)$和$g^w(c,r)$。通过将两者进行最后的融合形成最终模型的匹配分数：

$$g(c,r) = g^u(c,r) + g^w(c,r) \tag{3.19}$$

最终的匹配分数融合了在上下文-回复对中话语和单词级别的匹配信息。因此，同时对话语和单词级别的对话进行建模的效果要显著优于对两者单独建模的效果。

话语级别和单词级别的对话匹配分别对应一个交叉熵损失函数，融合模型通过最小化两者损失函数的总和来进行整体优化。使用Θ代表融合模型的参数，总目标函数由公式（3.8）和公式（3.18）相加构成，可被定义为：

$$\begin{aligned}\mathcal{L}(\mathcal{D},\Theta) = &-\sum_{i=1}^{N}\left[y_i \log\left(g^u(c_i,r_i)\right) + (1-y_i)\log\left(1-g^u(c_i,r_i)\right)\right] \\ &-\sum_{i=1}^{N}\left[y_i \log\left(g^w(c_i,r_i)\right) + (1-y_i)\log\left(1-g^w(c_i,r_i)\right)\right]\end{aligned} \tag{3.20}$$

在实际的应用系统中，基于检索式方法的聊天机器人要将匹配方法应用在回复选

择任务上，需要事先从索引中检索出多个候选回复。这里应用一种启发式方法来检索候选回复。给定一个带有前 n-1 轮话语$\{u_1,\ldots,u_{n-1}\}$的话语u_n，从前 n-1 轮话语中根据对应的 tf-idf 分数检索出前几个特定的词语，然后使用这些词语来扩展u_n，将其送入索引中，通过索引中内置的匹配算法找出合适的候选回复。最后使用匹配分数对候选回复进行排序，并返回最前面的候选项作为对应上下文的回复。以这样的方式可以高效地检索出合适的回复，适合在实际的场景中应用。

3.3 结合知识图谱的多轮对话

知识图谱（Knowledge Graph，KG）主要用来对搜索引擎进行一定的改善，当用户使用搜索引擎进行知识查询时，查询到的结果会以一定的组织形式呈现给用户。随着大数据时代的到来，知识和信息的日益丰富促进了知识图谱的发展，使得知识图谱在搜索、问答、推荐和推理等方面实现了广泛应用。通俗地说，知识图谱实质上是一种语义网络，由定义的实体、关系和实体三元组组成，通过将这些信息以图的形式展现出来，用以描述真实生活中事物的概念及其依赖。本文尝试在三个公开数据集上进行知识图谱的构建，为检索式多轮对话模型提供相应的外部知识，以提高模型在回复选择任务上的表现。

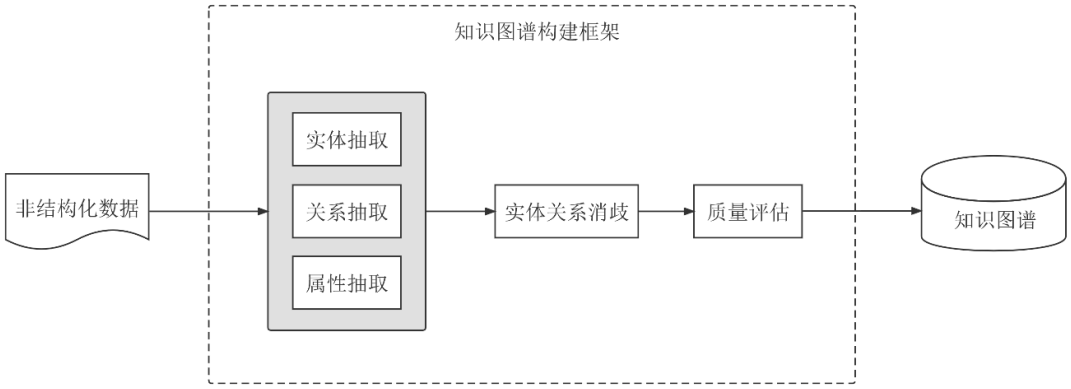

图 3.4 非结构化数据的知识图谱构建示意图

由于对话语料为文本形式的非结构化数据，不存在事先定义的实体、关系和属性，因此需要通过自然语言处理方法[41]对语料进行三方面信息的抽取：实体抽取、关系抽取和属性抽取。由于知识抽取到的信息之间缺乏联系，因此需要将实体、关系和属

性进行整合和实体关系消歧，最后进行质量评估，从而形成一个完整的知识图谱。本文构建知识图谱的流程如图 3.4 所示。

当知识图谱构建完成后，通过在对话匹配模型中添加一个键值储存模型，并将多轮会话中涉及到的全部三元组初始化为键值储存模型的知识信息。当对话的知识信息被引入后，对话匹配模型的性能得到进一步提高。接下来逐一介绍构建知识图谱的具体过程和在对话匹配模型中引入知识信息的过程。

3.3.1 对话语料的知识抽取

知识抽取通常分为多个子任务：实体抽取[42]、关系抽取[43]，而属性抽取通常可看成特殊的关系抽取。命名实体识别是将给定文本中的人名、地名等各种具有特定意义的实体抽取出来进行分类归纳为结构化信息，其方法大致可分为三类：基于模板的方法、基于机器学习的方法和基于深层神经网络的方法。前两种方法需要手工对数据集进行处理标签，费力耗时，成本代价高昂。随着深度学习的发展，利用深度学习中的各种神经网络来对命名实体识别进行端到端的建模已经成为研究热点，而预训练语言模型的出现更是极大地提高了该任务的表现。关系抽取是在上一步完成的基础上，对实体之间的关系进行提取。本文通过基于深度学习的方法来进行关系抽取，该方法主要判断两个实体是否存在某种联系，将计算得到的结果作为对应标签的概率。

由于本文使用的数据来自公开的数据集，因此不需要提前对数据进行采集和分析。这里简单地将语料中的多轮对话以文本的形式提取出来，然后通过自然语言处理方法将输入序列中的实体和实体之间的联系提取出来形成三元组进行保存。受研究[41]的启发，本文拟采用管道流水线方法进行知识抽取，首先采用 Bert+BiLSTM+CRF 的方法来进行实体抽取，在完成实体抽取任务之后，通过将文本以及文本中的实体输入至 BERT 中进行关系分类，最后形成知识三元组存入知识图谱。

实体抽取模型由三个模块组成，分别为预训练语言模型 BERT 模块、双向长短期记忆力网络（BiLSTM）和条件随机场（Condition Random Field，CRF）。文本经过模型处理可得到实体被标注好的序列。

首先将对话语料进行去噪和分词表示成输入序列$X = [x_1, x_2, \ldots, x_n]$，然后使用预训练语言模型 BERT 对输入序列进行表征以获取对应的隐藏表示$H = [h_1, h_2, \ldots, h_n]$。

接着，BiLSTM 来双向提取上下文表示中的特征，并通过线性层利用文本对应的最后一层隐状态计算出每个字对应类别标签的概率$Y = [y_1, y_2, \ldots, y_n]$。如果此时直接选取文本对应输出概率中分值最高的来进行分类，就考虑不到相邻位置的信息，无法得到最优结果。因此，这里引入条件随机场 CRF 来根据给定的X建立预测序列Y的条件概率分布模型：

$$P(y|x) = \frac{1}{z(x)} exp\left(\sum_{i=2}^{n}\sum_{p=1}^{P}\lambda_p t_p(y_{i-1}, x, i) \cdot \sum_{i=1}^{n}\sum_{q=1}^{Q}\beta_q s_q(y_i, x, i)\right) \tag{3.21}$$

其中i代表单词在文本中的位置，n 为文本长度，P 和Q代表目前结点的特征函数的数量。t_p和s_q分别代表转换和状态特征函数，λ_p和β_q分别为两个函数的权重系数，利用最大似然估计得到。$z(x)$为归一化因子。

模型最后的输出结果为输入序列对应的实体标注序列，其中 O 代表非实体词，而 LOC 代表地理位置实体，B 和 I 分别代表实体词的开始和延续。其他实体词及其标注符号包括：人名（PER）、命令名（COM）、机构名（ORG）、歌名（SIN）、奖项名（REW）、公司名（COR）和商标名（TRA）等。整体的模型如图 3.5 所示。

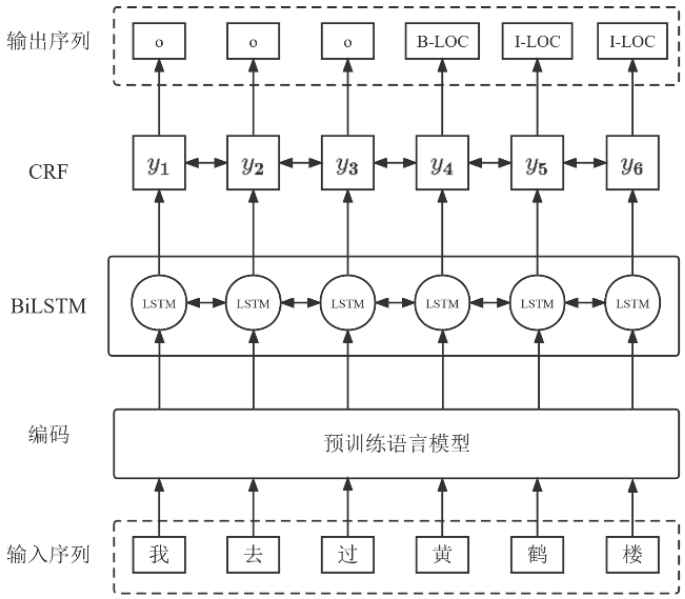

图 3.5 实体抽取模型

在完成实体抽取之后，文本中的多个实体被标注出来，接下来就需要对实体进行

关系分类。实体之间的关系标签包括：朋友、亲人、使用、参观、演唱、发行时间、获的和具体信息等多个。这里采用的关系抽取方法[44]是将文本和实体一起作为 BERT 的输入，通过‘$’和‘#’将文本中两个实体的首尾位置标记出来，并在序列的开头插入[CLS]特殊符号用来表示文本的上下文信息。关系抽取模型结构如图 3.6 所示。

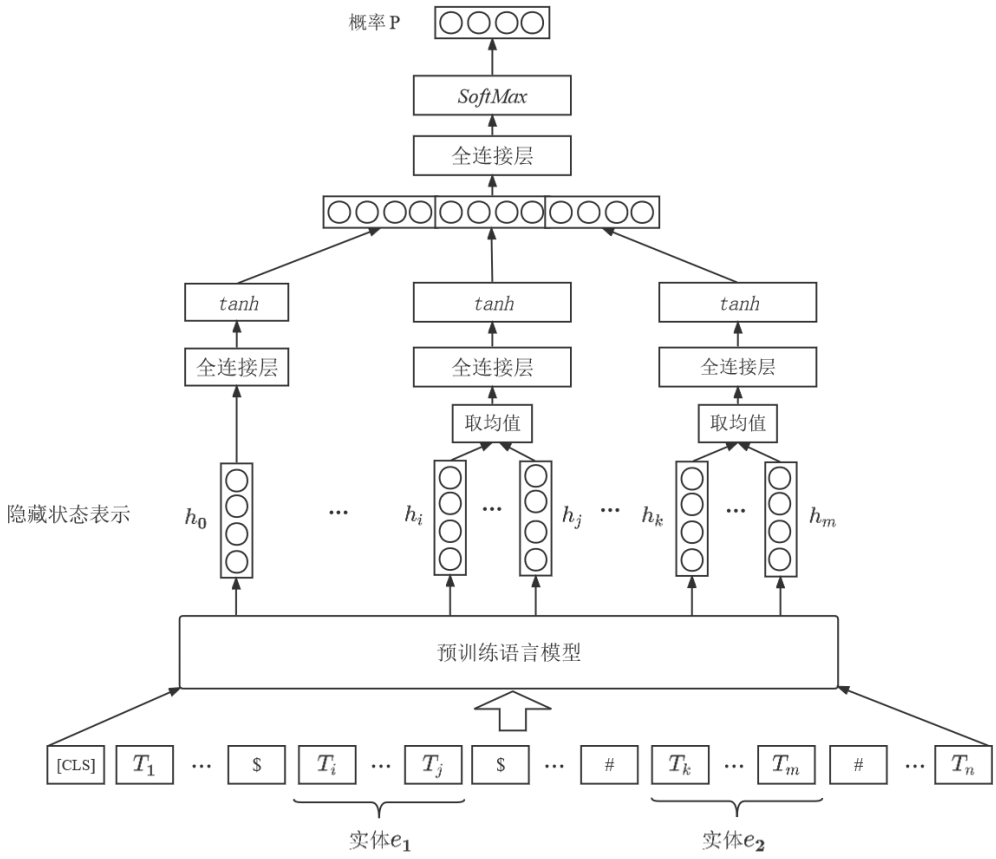

图 3.6 关系抽取模型

给定一段文本$T = [t_1, t_2, \ldots, t_n]$和文本中的两个实体$e_1$与$e_2$，经过预训练语言模型的深度编码之后文本对应的最终隐藏状态为$H = [\mathrm{h}_0, \mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_n]$，其中$\mathrm{h}_0$对应特殊符号[CLS]的上下文表示，$\mathrm{h}_i$代表文本中第 i 个单词对应的上下文向量，而$e_1$和$e_2$对应的最终隐藏状态分别为$[\mathrm{h}_i, \ldots, \mathrm{h}_j]$和$[\mathrm{h}_k, \ldots, \mathrm{h}_m]$，其中 i，j 和 k，m 分别代表两个实体在文本中的起始和结束下标，且$0 < \mathrm{i} < \mathrm{j}$，$\mathrm{j} < \mathrm{k} < \mathrm{m}$，$\mathrm{m} < \mathrm{n}$。接下来对两个实体的向量表示进行取均值操作，再通过激活函数和全连接层的处理得到两个实体的上下文表示h_1'和h_2'。

$$\begin{aligned} \mathrm{h}_1' &= W_1\left[\tanh\left(\frac{\sum_{t=i}^{j} h_t}{j-i+1}\right)\right]+b_1 \\ \mathrm{h}_2' &= W_2\left[\tanh\left(\frac{\sum_{t=k}^{m} h_t}{m-k+1}\right)\right]+b_2 \end{aligned} \tag{3.22}$$

其中W_1、b_1、W_2和b_2为对应实体e_1和e_2全连接层可训练的参数，并且这里设置$W_1 = W_2$，$b_1 = b_2$。同理，对于特殊标记[CLS]对应的隐藏状态h_0采用同样的操作：

$$\mathrm{h}_0' = W_0[\tanh(\mathrm{h}_0)] + b_0 \tag{3.23}$$

其中W_0、W_1和W_2具有相同的维度，$W_0, W_1, W_2 \in \mathbb{R}^{d\times d}$，$b_0$、$b_1$和$b_2$为偏置向量，$d$为预训练语言模型 BERT 的隐藏状态大小。

最终，将h_0'、h_1'和h_2'连接起来输入至全连接层，再经过 SoftMax 激活函数得到最后的分类概率。

$$\begin{aligned} h'' &= W_3[concat(\mathrm{h}_0', \mathrm{h}_1', \mathrm{h}_2')] + b_3 \\ p &= SoftMax(h'') \end{aligned} \tag{3.24}$$

其中$W_3 \in \mathbb{R}^{L\times 3d}$，$L$为关系类型的数量，$p$对应最后关系类别的概率，损失函数采用交叉熵损失函数。实体关系抽取任务完成之后，大量的实体-关系-实体三元组形成初步的知识表示，其中知识三元组样例如表 3.2 所示。

表 3.2 知识三元组样例

对话历史	知识三元组		
	头部实体	关系	尾部实体
用户 A：听过《以父之名》这首歌吗？	洪敬尧	编曲	以父之名
用户 B：听过呀，黄俊郎填词，洪敬尧编曲的一首，很好听。	黄俊郎	填词	以父之名
用户 A：嗯，洪敬尧更是凭借这首歌曲，获得了第 4 届最佳编曲奖。	洪敬尧	获得	第 4 届最佳编曲奖
用户 B：而且这首歌当年首播更是有八亿人在收听。			
用户 A：啊，那具体发行的日期是什么时候？			
用户 B：2003 年 7 月 16 日。	2003 年 7 月 16 日	发行时间	以父之名
用户 A：哦，十几年前的歌，算是周杰伦演唱的一首老歌了。	周杰伦	演唱	以父之名
用户 B：但确算是他最经典的歌曲，7 月 16 日被定为周杰伦日。	7 月 16 日	定为	周杰伦日

由于本文使用的对话语料相对简单，多轮对话中的实体语义相对单一，因此暂不考虑对其进行实体关系消歧，本章将重点放在如何更好地利用这些知识三元组来增强检索式对话系统的工作上。

3.3.2 对话模型的知识引入

在完成知识图谱的完整构建之后，本文在检索式对话系统模型中引入一个键值对存储模块[45]来充分利用知识图谱中的知识信息，而键值对存储模块中的知识信息是多轮对话中提及的所有知识三元组。本小节首先具体介绍该模块的具体功能，再将该模块应用到本文的对话模型。

键值对存储模型是建立在端到端内存网络架构之上的，主要的功能是从一个文档中提取给定问题的答案。该模型可具体分为三个部分：（1）键散列（Key Hashing）：给定一个查询，从知识库中检索出与这个查询有一定联系的信息。这里检索的方法可以是两个序列至少包含一个相同实体。然后在模型训练时将这些信息与训练数据作为一个整体送入模型中；（2）键寻址（Key Addressing）：在寻址过程中，将查询与每个键相互比较，为每个候选内存计算对应的概率；（3）值读取（Value Reading）：在最终的读取操作中，通过使用寻址的概率求加权和来读取内存的值，并将其作为输出返回。

键值对存储模型的优势在于可以灵活地对先验知识进行表示，这样可以将特定领域的背景知识编码至内存中，从而训练相应的模型。键值对存储模型的编码方式有多种，这里主要介绍本文使用的三元组方式。三元组的主要结构为主语、关系和宾语，键主要有左侧的主语和关系组成，对应于需要回答的问题，值由右侧的宾语组成，对应于问题的答案。此外，还将其进行了反向表示，即宾语、关系和主语，这对于回答不同类型的问题很重要。接下来，本文将此方法具体应用在对话模型中。

给定一个知识三元组$\mathrm{T}=\{(h_1,r_1,t_1),(h_2,r_2,t_2),\dots,(h_n,r_n,t_n)\}$，其中$h_i$、$r_i$和$t_i$分别为头部实体、关系和尾部实体，$1\le i\le n$。键值对存储模块中的键存储和值存储分别表示为键向量k_i和值向量v_i，其中k_i为头部实体h_i和关系r_i的平均词嵌入，v_i为尾部实体的词嵌入。接着，通过一个查询向量q来处理键向量k_i得到对应的分数α_i，然后将分数作为值向量v_i的权重得到最后的分值：

$$\alpha_i = \text{softmax}(q^T k_i)$$
$$v = \sum_{i=1}^{n} \alpha_i v_i \tag{3.25}$$

其中查询向量q初始化为特殊符号[CLS]对应的隐藏状态h_0。经过键值对存储模块的计算得到最终的值向量v之后，将其与h_0进行连接，然后输入至全连接层和激活函数进行最后的分类。

$$\text{p} = \sigma[\text{W}(h_0 \oplus \text{v}) + \text{b}] \tag{3.26}$$

其中σ为 sigmoid 激活函数，W和b为全连接层的参数和偏置，$\oplus$表示连接操作。

为了使得检索式对话模型利用到知识三元组信息，本文额外增加了一个注意力损失$\mathcal{L}_{att}$，该损失仅在正例上计算：

$$\mathcal{L}_{att} = -\frac{1}{|\{truth\}|} \sum_{i=1}^{|\{truth\}|} \log \alpha_i \tag{3.27}$$

其中$\{truth\}$为正确回复中使用的三元组索引集合，$|\{truth\}|$为集合中元素个数。

3.4 本章小结

本章基于预训练语言模型提出了一个新的多级匹配检索式对话模型，该模型考虑到了话语级别的局部匹配信息和单词级别的全局匹配信息，并将两种信息融合在一起提高了检索式对话模型的性能。此外，本章在三个公开数据集上构建了知识图谱来为对话模型引入外部知识，首先利用实体和关系抽取模型将语料对话中包含的知识三元组抽取出来，然后使用键值存储模型将知识信息引入至对话模型，以此提高模型的性能。

4 实验数据和结果分析

本章主要对多级匹配检索式对话系统在多个数据集上开展了多组实验，并对实验数据进行详细分析，以此来验证模型设计的可行性和改进方法的有效性。本章首先介绍实验需要使用到的多个数据集以及模型运行的实验环境。然后具体介绍模型评价指标、基线模型、模型基本参数设置和实验具体进行的对比和消融实验。最后展示出对比和消融实验的结果，并对其进行进一步具体分析，讨论模型中的各种因素对于实验结果的影响。

4.1 数据集

本章主要在三个公开的对话数据集上进行对话匹配模型的训练、验证和测试。训练数据包括一个英文数据集 Ubuntu 语料库和两个中文数据集豆瓣（Douban）、电子商务（E-commerce）语料库。表 4.1 展示了三个数据集的统计分布。

表 4.1 Ubuntu、Douban 和 E-commerce 数据集的数据统计

语料统计	Ubuntu			Douban			E-commerce		
	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
数据量	1M	500K	500K	1M	50K	50K	1M	10K	10K
样本候选回复数量	2	10	10	2	2	10	2	2	10
样本平均对话轮数	10.13	10.11	10.11	6.69	6.75	6.45	5.51	5.48	5.64
每轮话语平均单词数	11.35	11.34	11.37	18.56	18.50	20.74	7.02	6.99	7.11

Ubuntu 数据集是从 Ubuntu 论坛中收集的关于技术支持的英文多回合对话语料，其中包含 50 万的对话样例数，通过在每个对话样例中插入正确与错误回复形成 100 万的训练样例，而验证和测试样例数量均为 50 万。Ubuntu 对话语料中的样本平均对话轮数较长，大于 10 轮。

豆瓣（Douban）数据集是从豆瓣小组中收集的中文多回合对话语料，豆瓣小组是中国网民用于社交服务的网络平台。豆瓣对话语料包含两人之间多于 6 轮的对话，

并且每轮话语中的平均单词数量较多，超过 18 个。训练样例、验证样例和测试样例分为 1 百万、5 万和 5 万。

电商（E-commerce）数据集是由电商平台上客服与客户之间真实的多回合对话组成，语料中包含关于各种商品的各种类型的对话，例如商品咨询、物流快递、推荐和闲聊等。数据集包含 1 百万的训练样例和 1 万的验证与测试样例。

经过本文统计，Ubuntu、Douban 和 E-commerce 数据集中的对话轮数范围分别是（2，20）、（4，91）和（2，120）。其中 Ubuntu 和 Douban 数据集的轮数范围步长为 1，而 E-commerce 的步长为 2。具体的三种数据集的对话轮数的统计分布如图 4.1 所示，由于 Ubuntu 数据集的最大轮数为 20，且另外两个数据集的轮数超过 20 的占比较少，因此本文利用图中的轮数 21 来代表所有轮数超过 20 轮的对话样例。从图中可以看出，Ubuntu 数据集对话轮数为 20 的样例所占比例最多，并且其他轮数的样例占比较为均衡，而 Douban 和 E-commerce 数据集对话轮数为 4 的样例所占比例最多，且大多数的样例的轮数集中在前 10 轮。总的来说，Ubuntu 数据集整体相比另外两个数据集较为均衡，因此，在接下来的消融实验中就单独采用 Ubuntu 数据集来对比实验效果。

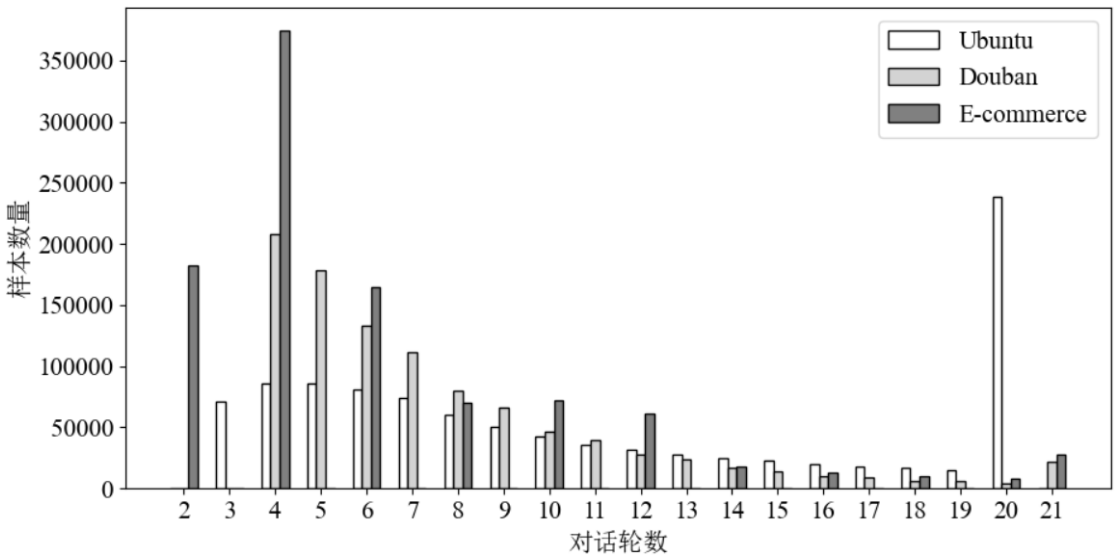

图 4.1 三种数据集中对话轮数的分布统计图

4.2 评价指标

对于检索式对话模型性能的评价，与以往研究的评价指标相同，本章实验主要使用 n 个候选回复中的前 k 召回（$R_n@k$）作为评价指标。对于豆瓣中文语料，除了使用$R_n@k$之外，还使用了倒数排名均值（Mean Reciprocal Rank, MRR）和平均准确率均值（Mean Average Precision, MAP）作为对话模型的评价指标。因为豆瓣语料库中的一些对话不止一个正确的候选答案。

$R_n@k$指标主要用于检索系统上，用来评估系统的查全率。当模型计算出所有候选回复的匹配分数并将其排序后，$R_n@k$通过计算在 n 个候选回复中前 k 个位置里正确回复的数量与所有正确回复的比例，计算方式如下：

$$R_n@k = \frac{c}{T} \tag{4.1}$$

其中 c 为在 n 个候选回复中前 k 个位置里正确回复的数量，T 为所有正确回复的数量。$P@k$指标用来评估检索系统的查准率，计算方式与$R_n@k$类似：

$$\mathrm{P@k} = \frac{c}{k} \tag{4.2}$$

MAP 指标为平均准确率均值，主要计算过程是先计算测试集中所有对话的平均准确率，再将这些值取平均。通过计算出候选项中每个正确项被检索出来的准确率，再将这些值取平均即可得到每一个对话的平均准确率。其计算方式如下：

$$\begin{aligned} AP_i &= \frac{1}{Q_i}\sum_{j=1}^{Q_i}\frac{c}{P(j)} \\ MAP &= \frac{1}{N}\sum_{i=1}^{N} AP_i \end{aligned} \tag{4.3}$$

其中，AP_i对应于第 i 个对话样本的平均准确率，Q_i代表样本中正确回复的数量，$P(j)$代表第 j 个正确回复在候选回复中的位置，N代表测试集中的样本数量。MAP衡量了检索式对话系统的整体性能。

MRR 指标为倒数排名均值，主要计算过程是将正确回复在给定排好序的候选回复中的位置取倒数从而得到精准度，再将全部样例的精准度取平均从而得到计算结果。从计算过程来看，正确回复在候选回复中的排名越高，得分也就越高，表示最后

的检索结果越好。其计算方式如下：

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \tag{4.4}$$

其中$rank_i$为第 i 个对话样本中第一个正确回复在候选回复中的排序位置。

4.3 基线模型

本章将多级匹配检索式对话模型与多个检索式对话系统中的经典模型进行比较，这些基线模型主要分为两种：交互式方法和基于预训练语言模型的方法。由于本文是基于预训练语言模型的方法，因此这里将使用在基于交互的方法中表现最好的模型和目前大多数最新的基于预训练语言模型的方法中的基线模型。

基于交互的方法：最经典的基于交互式的方法为顺序匹配模型[20]，该方法将可选回复与对话历史的话语在多个表示上进行交互，并通过深度神经网络提取匹配信息。接着多个模型相继提出，包括深度话语聚合模型[21] 和深度注意力匹配模型[23]等，而在预训练语言模型被提出之前，在回复选择任务上表现最好的模型为多跳选择器网络模型。多跳选择器网络（Multi-hop Selector Network, MSN）模型[27]将上下文中的最后一句话作为关键，利用多跳选择器选择出与其在单词和句子层面相关的上下文话语，然后融合这些选定的上下文话语并将其与候选回复匹配。

基于预训练语言模型的方法包括：BERT 模型[35]通过特殊符号将上下文中的话语和回复连接起来并作为输入，最后使用特殊符号[CLS]对应的输出表示作为分类结果来进行建模；BERT-SS-DA 模型[36]在输入序列中区分了不同说话者来充分利用说话者的特征，并在对话样例不同时间点的真实对话中进行切断来扩充训练语料库；SA-BERT 模型[37]可以感知说话者的变换信息，并提供了一种对话解缠策略来根据说话者的信息选择少量重要的话语作为上下文；BERT-UMS 模型[38]通过引入插入、删除和查询三种话语操作策略来充分提取话语之间的顺序和时间依赖关系；BERT-SL 模型[39]引入四个自监督任务，并以多任务的形式联合训练基于预训练语言模型的对话模型；Bert-FP 模型[40]通过一种新的细粒度的后训练方法来帮助对话模型充分学习每个上下文话语回复对中的语义信息，并通过一个新的训练目标-话语关联分类来帮助

模型理解话语之间的语义关联和连贯性。

4.4 模型训练

由于本文方法是基于预训练语言模型的，因此对于模型训练的实验环境较为苛刻。接下来将具体介绍实验环境和模型训练的参数设置。

4.4.1 实验环境

本章的实验是在本校的高性能计算公共服务平台上进行的，操作系统镜像选择的是 Jupyter-cuda 11.2，开发语言为 Python，版本为 3.6，深度学习框架为 Pytorch1.4.0。预训练语言模型 BERT 根据数据集语言的不同分别选用 base-uncased 和 base-chinese 版本。实验环境具体配置如表 4.2 所示。

表 4.2 实验环境

名称	内容
CPU	Intel Xeon Gold 6230R
内存	128GB
GPU	NVIDIA Tesla V100s
显存	32GB
操作系统镜像	Jupyterlab-cuda 11.2
深度学习框架	Pytorch 1.4.0
Python 版本	3.6
BERT	base-uncased, base-chinese

4.4.2 参数设置

本文对话模型是基于 Pytorch 实现的，预训练语言模型选择$\text{BERT}_{\text{base}}$，在实际训练和微调中遵循与 BERT 参数相同设置。分词和词嵌入学习选用 BERT 的分词器完成。模型参数的更新选用 Adam 梯度下降算法，训练最大轮数为 3，模型初始学习率为$3e^{-5}$，选用 warmup 技术并设置比率为 0.1 保证网络训练的速度和收敛性，训练和评估批次大小分别为 32 和 16。在话语级别对话匹配模型中使用了三层卷积神经网络来提取匹配特征，每层的卷积核尺寸都为（3，3），步长为 1。第一层的最大池化尺寸为（2，3），步长为（2，3）；第二层最大池化尺寸为（3，4），步长为（3，4）；第

三层最大池化尺寸为（6，5），步长为（6，5）。这么设置的原因是为了使输入维度为（322，768）的匹配矩阵转换为（8，12）的特征矩阵。卷积核为循环神经网络 GRU 的隐藏状态维度设置成 768，与 BERT 的隐藏状态维度大小相同。单词级别对话匹配模型设置与上大致相同。

模型进行训练时的几个关键参数也会对整体模型有一定的影响。这些关键参数主要有两个：话语级别对话匹配模型中上下文的话语轮数和单词级别对话匹配模型中上下文的单词数量。因此，本节分析这两种参数的不同设置对于模型性能的影响，其中上下文中的话语轮数范围设置为（1, 10），步长为 1，上下文的序列长度设置为（50, 500），步长为 50。

图 4.2（a）展示了上下文中不同话语轮数对于话语级别匹配模型性能影响的实验结果。实验结果表明，当话语轮数小于等于 6 时，轮数增加可以显著提升模型的性能，当轮数大于 6 时，模型性能的提升趋于稳定。这意味着多轮对话中早期的话语可以为回复选择提供非常重要的匹配信息。同理，图 4.2（b）展示了上下文序列的长度即单词数量对于单词级别匹配模型性能影响的实验结果。结果表明，当上下文序列长度小于 150 时，序列长度的增加可以显著提升模型的性能，当长度大于 150 时，模型性能的提升趋于稳定。这意味着当上下文序列长度增加到一定程度时，上下文中的每个单词学习到的上下文表示信息更加丰富，对模型性能的提升也就越来越好。

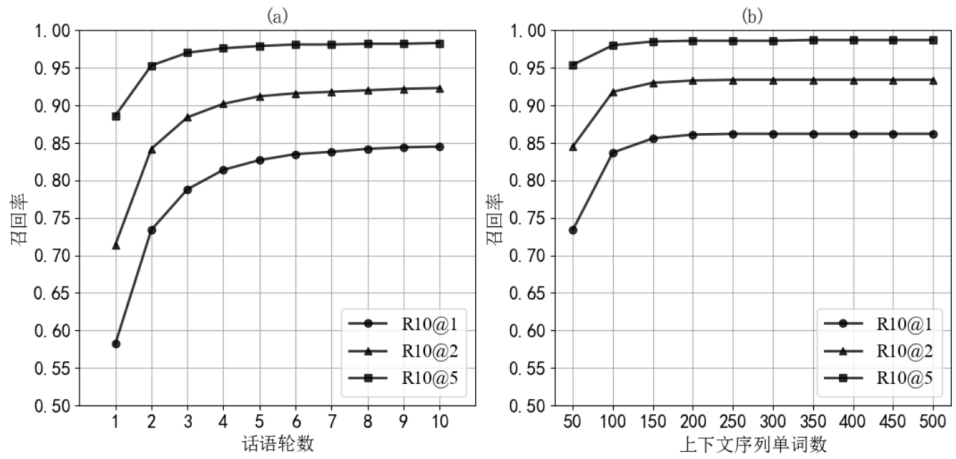

图 4.2 关键参数对于模型性能的影响

（a）上下文中不同话语轮数对于话语级别匹配模型性能的影响;（b）上下文不同序列单词数对于单词级别匹配模型性能的影响

4.5 实验结果及分析

本节首先在三个数据集 Ubuntu、Douban 和 E-commerce 上对模型进行了对比实验，通过与基线模型的比较来说明本文方法的有效性。在此基础上，选择在数据较为均衡的 Ubuntu 数据集上进行消融实验来分析模型不同设置对于整体性能的影响。

4.5.1 对比实验

本小节首先将多级匹配检索式对话方法与现有的基线模型进行的性能对比并给出具体的实验结果，其次将话语级与单词级匹配模型单独训练，分析两种级别的匹配信息对于融合模型的影响，最后分析外部知识引入对于对话模型性能的影响。

（1） 基线模型对比结果

表 4.3、4.4 和 4.5 展示了本文的多级匹配检索式对话模型（BERT-MM）在三个数据集上与基准模型的对比实验结果。结果表明，在基于预训练语言模型的方法中$\text{BERT}_{\text{base}}$的效果要优于基于交互式的方法，这意味着利用预训练语言模型对上下文中的话语和回复进行编码和提取信息是非常有效的。

本文提出的多级匹配检索式对话方法相比较于$\text{BERT}_{\text{base}}$在 Ubuntu、Douban 和 E-commerce 数据集上核心指标$R_{10}@1$分别得到了 5.9%、2.5%和 11.3%的提升，这表明多级匹配对话模型结构可以很有效地提高回复选择任务的表现。相较于之后的 BERT-SS-DA 和 SA-BERT，本文方法仍在大多数指标上领先，尤其在 Ubuntu 和 E-commerce 数据集上的指标$R_{10}@1$仍得到了 1.2%和 1.9%的性能提升。

BERT-UMS 和 BERT-SL 以自监督任务的方法来改进对话模型，而 BERT-FP 以细粒度对比学习的方式来改进。从表中数据来看，三种模型带来的效果提升非常好，特别是 BERT-FP，相较于$\text{BERT}_{\text{base}}$在三个数据集的$R_{10}@1$带来了 10.3%、4.4%和 26.0%的性能提升。考虑到本文方法与这三种方法是不同方向上的改进，因此本节将多级匹配对话模型结构与效果最好的 BERT-FP 模型相结合形成 BERT-FP-MM，并在数据集上进行了实验。实验结果和预想的一样，在三个数据集上核心指标 $R_{10}@1$分别得到了 0.5%、1.0%和 0.4%的提升。这表明，本文提出的方法可以有效结合自监督任务的方法进一步提高对话模型的性能。

表 4.3 Ubuntu 数据集上的对比实验结果

模型	Ubuntu		
	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
基于交互的方法			
MSN[27]	0.800	0.899	0.978
基于预训练语言模型的方法			
$\text{BERT}_{\text{base}}$[35]	0.808	0.897	0.975
BERT-SS-DA[36]	0.813	0.901	0.977
SA-BERT[37]	0.855	0.928	0.983
BERT-MM（本文）	0.867	0.936	0.987
BERT-UMS[38]	0.875	0.942	0.988
BERT-SL[39]	0.884	0.946	0.990
BERT-FP[40]	0.911	0.962	**0.994**
BERT-FP-MM（本文）	**0.916**	**0.967**	0.993

由于 Douban 数据集中一些对话不止一个正确候选回复，而 Ubuntu 和 E-commerce 数据集中只有一个，因此 Douban 数据集额外使用 MAP 和 MRR 两个评价指标。

表 4.4 Douban 数据集上的对比实验结果

模型	Douban				
	MAP	MRR	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
基于交互的方法					
MSN[27]	0.587	0.632	0.295	0.452	0.788
基于预训练语言模型的方法					
$\text{BERT}_{\text{base}}$[35]	0.591	0.633	0.280	0.470	0.828
BERT-SS-DA[36]	0.602	0.643	0.280	0.491	0.843
SA-BERT[37]	0.619	0.659	0.313	0.481	0.847
BERT-MM（本文）	0.616	0.655	0.304	0.503	0.850
BERT-UMS[38]	0.625	0.664	0.318	0.482	0.858
BERT-FP[40]	0.644	0.680	0.324	0.542	0.870
BERT-FP-MM（本文）	**0.651**	**0.693**	**0.334**	**0.550**	**0.877**

虽然本文的方法与现有的 BERT-FP 相结合，模型在大多数指标上可以超过基线模型，但是仍在几个指标上有所下降，如 Ubuntu 的$R_{10}@5$和 E-commerce 的$R_{10}@2$。其原因为：细粒度对比学习使得预训练语言模型从数据集中学习到了丰富的上下文

语义信息，本文的方法虽然可以进一步提取匹配信息，但两者还是在一定程度上重复了，导致在局部造成了过拟合，从而模型的性能有所下降。

表 4.5 E-commerce 数据集上的对比实验结果

模型	E-commerce		
	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
基于交互的方法			
MSN[27]	0.606	0.770	0.937
基于预训练语言模型的方法			
$\text{BERT}_{\text{base}}$[35]	0.610	0.814	0.973
BERT-SS-DA[36]	0.648	0.843	0.980
SA-BERT[37]	0.704	0.879	0.985
BERT-MM（本文）	0.723	0.882	0.985
BERT-UMS[38]	0.762	0.905	0.986
BERT-SL[39]	0.776	0.919	0.991
BERT-FP[40]	0.870	**0.956**	0.993
BERT-FP-MM（本文）	**0.874**	0.952	**0.994**

（2） 话语和单词级别建模方式对比结果

为比较在话语和单词级别两种建模方式下模型的性能和融合之后的多级匹配模型的性能，本文分别对话语级别和单词级别的对话匹配模型进行了训练和测试，表 4.6 展示了三种模型在 Ubuntu 数据集上的对比实验结果。

表 4.6 三种建模方式在 Ubuntu 数据集上的实验结果

模型	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
话语级别模型	0.844	0.922	0.982
单词级别模型	0.855	0.929	0.986
多级匹配模型	**0.867**	**0.936**	**0.987**

由表 4.6 可知，话语级别和单词级别的对话匹配模型都展现了不俗的性能，都优于以往的大多数基线模型，并且两种级别的匹配模型性能大致相同，意味着两种级别的匹配信息对于多轮回复选择任务来说是同样重要的。由于两个级别的匹配模型关注不同的信息：话语级别关注上下文中每个话语和候选回复之间的局部匹配信息而单词级别关注上下文序列中的全局匹配信息，因此，融合了全局和局部的匹配信息的多级匹配模型性能得到了再次提升，这表明话语级别和单词级别的对话匹配

在匹配不同的互补信息时非常有效。

（3） 知识驱动模型对比实验结果

为分析引入外部知识对于检索式对话系统的影响，本文在三个数据集上简单地构建了知识图谱，表 4.7 展示了知识图谱的相关数据统计。其中，Ubuntu 数据集中的实体、关系和三元组数量为 1541、349 和 6125；Douban 数据集中实体、关系和三元组数量为 1307、331 和 5747；E-commerce 数据集中的实体、关系和三元组数量为 1102、292 和 4854。此外，本节还统计了三个数据集中的每个三元组的平均词元数量，分别为 31.2、29.7 和 26.4。

表 4.7 三种数据集对应知识图谱的数据统计

数据集	Ubuntu	Douban	E-commerce
实体	1541	1307	1102
关系	349	331	292
三元组	6125	5747	4854
三元组平均词元数	31.2	29.7	26.4

在抽取出知识三元组之后，本文通过一个键值存储模块将其引入检索式对话系统以提高其模型表现。本章考虑在$\text{BERT}_{\text{base}}$和多级匹配模型中引入外部知识，由于 BERT-FP 数据集经过特殊处理的原因，因此不考虑在 BERT-FP 中引入外部知识。实验结果如表 4.8、4.9 和 4.10 所示，两个模型的性能在 Ubuntu 数据集的核心指标上得到了 0.9%和 0.5%的提升，其中+KG 表示引入外部知识。

表 4.8 Ubuntu 数据集上知识引入的实验结果

模型	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
$\text{BERT}_{\text{base}}$	0.808	0.897	0.975
$\text{BERT}_{\text{base}}$+KG	**0.817**	**0.903**	**0.976**
多级匹配模型	0.862	0.934	0.987
多级匹配模型+KG	**0.867**	**0.936**	0.987

两个模型的性能在 Douban 数据集的核心指标上得到了 0.5%和 0.3%的提升。

表 4.9 Douban 数据集上知识引入的实验结果

模型	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
$\text{BERT}_{\text{base}}$	0.280	0.470	0.828
$\text{BERT}_{\text{base}}$+KG	**0.285**	**0.473**	**0.829**
多级匹配模型	0.301	0.501	0.849
多级匹配模型+KG	**0.304**	**0.503**	**0.850**

两个模型的性能在 E-commerce 数据集的核心指标上得到了 0.4%和 0.2%的提升。

表 4.10 E-commerce 数据集上知识引入的实验结果

模型	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
$\text{BERT}_{\text{base}}$	0.610	0.814	0.973
$\text{BERT}_{\text{base}}$+KG	**0.614**	**0.817**	**0.974**
多级匹配模型	0.721	0.881	0.985
多级匹配模型+KG	**0.723**	**0.882**	0.985

由上表的实验结果可知，外部知识的引入可以有效提高检索式对话模型的性能。但是一方面，相对于预训练语言模型这种深度模型，键值存储模块的网络结构还是相对较浅。另一方面，数据集中包含的实体与关系数量较少，因此对话模型的性能提升相对较弱。所以未来的研究工作考虑引入 XLORE 通用领域知识图谱来扩充本文构建的知识图谱，并进一步对比效果。

4.5.2 消融实验

本小节首先分析预训练语言模型不同后训练方式对于对话模型性能的影响，其次通过修改话语和单词级别的对话模型结构来探讨不同结构设置对于对话模型性能的影响。

（1） 预训练任务对模型的影响

为分析本文提出的改进预训练任务-对话匹配的有效性，本章在预训练语言模型基础上训练了三个模型：① $\text{BERT}_{\text{base}}$：直接使用谷歌公开的$\text{BERT}_{\text{base}}$模型参数进行

对话模型的构建；② BERT-DPT：基于 Ubuntu 对话语料库进行领域后训练（Domain Post-Training, DPT）的$\text{BERT}_{\text{base}}$模型，其使用的两个预训练任务为 2.3.1 节提到的 MLM 和 NSP 任务；③ BERT-DM（本文）：基于 Ubuntu 对话语料使用 2.3.2 节的对话匹配（Dialogue Matching, DM）训练任务训练$\text{BERT}_{\text{base}}$模型。

表 4.11 在 Ubuntu 数据集上不同方式训练$\text{BERT}_{\text{base}}$的实验结果

模型	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
$\text{BERT}_{\text{base}}$	0.808	0.897	0.975
BERT-DPT	0.851	0.924	0.984
BERT-DM（本文）	**0.855**	**0.929**	**0.985**

由表 4.11 可知，在使用 Ubuntu 对话语料再次训练之后的$\text{BERT}_{\text{base}}$模型其表现要优于无训练的$\text{BERT}_{\text{base}}$模型，在指标$R_{10}@1$、$R_{10}@2$和$R_{10}@5$上分别提升了 4.3%、2.7%和 0.9%个点，这表明虽然$\text{BERT}_{\text{base}}$在大规模通用语料上训练之后已经得到非常强大的表示和理解能力，但是利用小规模的特定领域的对话语料对其再次训练可以进一步提高其表现能力。而当使用了改进后的对话匹配预训练任务之后，模型效果得到再次提升，比原有的$\text{BERT}_{\text{base}}$模型指标提升了 4.7%、3.2%和 1.0%。这表明，本文提出的对话匹配预训练任务可以更有效的利用多轮对话的特征来提升模型的性能。

（2） 不同模块对模型的影响

为分析检索式对话模型中输入表示层、话语回复匹配层、特征聚合层和的不同设置对于模型的影响，本章将话语和单词级别对话匹配模型中的模块进行替换以测试出具有最优的模型设置。数据集选用 Ubuntu 对话语料，预训练语言模型选用$\text{BERT}_{\text{base}}$模型，通过如下组合形成多个测试模型：①模型I：输入方式为上下文话语和回复分段单独输入，采用 CNN 进行话语回复匹配，然后使用 RNN 进行匹配信息的累积，最后使用 RNN 的最后一层隐藏状态h_n进行计算匹配分数；②模型II：输入方式为上下文和回复连接成一个序列，并直接使用特殊符号[CLS]对应的上下文表示h_0进行计算匹配分数；③模型III：输入方式同②，采用点积注意力进行话语回复的匹配交互，然后使用 RNN 进行匹配信息的聚合，最后利用 RNN 的最后隐藏状态h_n计算匹配分数；④模型IV：与③不同的是利用[CLS]对应的上下文表示h_0和 RNN 的最后隐藏状

态h_n同时计算匹配分数；⑤模型V：与④不同的是使用多头注意力机制进行上下文话语和回复之间的交互。

表 4.12 不同模型设置在 Ubuntu 数据集上的实验结果

模型	输入方式	话语回复匹配	特征聚合	预测	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$
I	分段	CNN	RNN	h_n	0.784	0.892	0.972
II	序列	无	无	h_0	0.808	0.897	0.975
III	序列	点积	RNN	h_n	0.815	0.901	0.976
IV	序列	点积	RNN	h_0, h_n	0.823	0.904	0.976
V	序列	多头	RNN	h_0, h_n	**0.824**	**0.905**	**0.978**

从表 4.12 中的模型I和模型II的实验结果可知，通过将上下文中话语与候选回复拼接成序列的输入方式对模型提升的效果要明显优于分段输入的方式，这表明以序列的方式输入至预训练语言模型编码器可以学习到更多的上下文表示；从模型II和模型III的实验结果可知，通过深度神经网络对编码器输出的上下文表示进行匹配信息的再次提取可以有效的提高模型的表现，这表明经过编码之后的上下文话语和回复表示之间仍然具有丰富的匹配信息待提取；从模型III和模型IV的实验结果可知，同时使用[CLS]对应的上下文表示h_0和 RNN 的最后隐藏状态h_n来计算匹配分数的效果要优于单独使用两者计算，这表明h_0和h_n提取了话语回复中不同维度的匹配信息，这些信息可以通过叠加的方式来提升模型效果；从模型IV和模型V的实验结果可知，点积和多头两种注意力机制对于模型的效果大致相同，多头注意力要稍好一些，因为多头注意力可以基于相同的缩放点积注意力学习到输入中的不同信息，这些信息被组合起来形成最终的匹配信息。

4.6 本章小结

本章首先介绍了实验所需的数据集和对话模型的评价指标，其次逐个介绍需要对比的基线模型和实验所需的硬件环境与模型的训练参数设置，分析了不同话语轮数和序列单词数量对于模型的影响。最后列出多组对比和消融，并对实验结果进行了具体分析。实验结果充分证明了本文提出的模型和方法的有效性。

5 总结与展望

5.1 本文总结

由于检索式对话方法可以将一个复杂的对话问题抽象为简单的搜索问题，因此大量的对话系统工作都是采用检索式方法实现的。检索式对话方法的实现方式大致分为三种，前两种方法通过深度神经网络对上下文话语和候选回复进行编码、匹配和信息的累积以计算出匹配分数，模型提升的性能有限，并且随着对话场景的日益复杂和对话历史长度的增加，这些方法面临非常大的挑战。第三种为基于预训练语言模型的方法，由于预训练语言模型拥有强大的理解和表示能力，可以很好地提取出上下文话语和回复中丰富的语义信息。因此，本文主要研究的内容是将经过领域后训练得到的预训练语言模型与检索式对话方法相结合，并在此基础上设计出合适的匹配网络框架以更好地提取话语回复中的匹配信息。此外，考虑到对话模型中外部知识的缺失，本文在三个数据集上构建了知识图谱来为模型引入外部知识。具体地，本文主要的工作可分为以下四个方面：

（1） 考虑到预训练语言模型是在大量文本数据上训练得到的，对于特定领域的对话语料适应性不足，并且预训练任务对于回复选择的帮助有限。因此，本文提出了一个新的对话匹配预训练任务，该任务目的是预测给定样本中的回复是否为上下文的下一句，样本的正例与负例都取自一段对话。通过利用该任务对预训练语言模型进行对话领域后训练，可以有效提高预训练语言模型的对话领域适应性，从而提高对话模型的性能。

（2） 提出了一个多级匹配检索式对话方法，该方法结合预训练语言模型从话语级别和单词级别同时建模，分别提取不同级别的匹配信息，再将这两种信息融合在一起计算出最终的匹配分数。多级匹配方法不仅考虑了上下文话语和候选回复中的局部匹配信息，还考虑了整个对话历史中的全局匹配信息，解决了单种输入方式所带来的信息提取能力不足的问题

（3） 在三个数据集上构建了知识图谱，并将其中包含的知识三元组信息引入检索式对话系统，进一步提升了模型的性能。知识图谱的构建采用管道流水线的方法，

结合预训练语言模型对文本进行实体和关系提取，然后通过一个键值存储模块计算出三元组的分值向量并将其融入对话模型中，以提高模型的表现。

（4） 在 Ubuntu、Douban 和 E-commerce 数据集上开展了对比实验来测试多级匹配检索式对话模型的性能，并使用多个评价指标$R_{10}@1$、$R_{10}@2$、$R_{10}@5$、MAP 和 MRR 对模型的表现进行评价。此外，本文还进行了多个消融实验来说明对话匹配预训练任务和不同模块的有效性。实验结果表明，多级匹配方法结合预训练语言模型提高了对话模型的信息提取能力和表现，并且对话匹配预训练任务和知识的引入有效地提高了模型的性能。

5.2 研究展望

本文结合预训练语言模型的多级匹配检索式对话方法在数据集上取得了一定的性能提升，但该方法与现有工业界已落地的方法仍有一定差距和很大的优化空间，主要可分为以下几个方面：

（1） 对话模型是否可以很好地理解对话历史。无论是何种对话系统的建模方法，多轮对话的建模和理解都起着非常重要的作用。本文发现对话模型倾向于选择出语义上与语境更有关系的话语，而忽略了对话的一致性。因此，可以尝试提出相关的自监督任务的方法来帮助模型理解和保持对话的上下文一致性。

（2） 对话领域的迁移性问题。为了更好地比较模型，现有的检索式对话方法通常都侧重于单一领域的对话语料，例如 Ubuntu、豆瓣和电子商务等，用于模型的学习和测试。但是用户的对话内容会随着社会发展和语言演变而变化，这种变化导致用户实际的输入和实际训练中的数据存在较大的差异，从而导致对话模型无法很好地理解用户输入，最终生成低质量的回复。此外，目前可用的数据集涵盖的对话领域有限，远未达到开放域对话涉及的内容。因此，未来的工作一方面需要不断扩充对话语料，丰富不同领域的对话信息，另一方面需要建立一个可持续“进化”的对话系统，可以根据不断更新的对话语料学习和更新自身，从而保证模型的质量。

致 谢

依稀还记得两年前那个刚刚踏入华科对一切非常好奇的自己，时光像手中的流沙一样转瞬而过。这两年经过华科浓厚的学术氛围的熏陶，我在学业上获得了很大的进步，同时也认识了很多的朋友，提高了自己的人际交往能力。在这里，我想一一感谢在读研道路上默默助力我前行的老师、朋友和家人。

初见胡迎松老师的时候，他严肃认真的样子让我心里觉得这是一位知识渊博、治学严谨的好导师。学业上，老师对我的要求严格且细致；生活上，老师也给予我无微不至的关心。为了帮助我完成毕业设计，从开始的选题、文献阅读到后来的开题、大纲撰写、实验等，老师都给了非常多的宝贵意见，让我可以顺利的完成这篇毕业论文。在每周一次的组会上，老师会督促我的研究进度，并指正我在研究中的一些错误，这让我受益匪浅。此外，实验室的李丹老师在平时的学业和生活上也给予了我莫大的支持与帮助，她温柔地对待每一个人，细致严谨地完成每一份工作，让我从心底里对她敬佩不已。在这里我向两位老师表达最衷心的感谢，感谢你们对我的莫大帮助。

在读研的这两年里，我认识了很多知心的好朋友。我要感谢实验室里的熊耀武、刘进龙和贺炜，是他们在这条学术道路上一直陪着我，在数不清个日夜里一起奋斗至深夜，解决了一个又一个难题。我也要感谢宿舍里的张亮、杨海猛和庄潮坚，在每次失意的时候，你们都会默默鼓励安慰我，让我可以坚定地走下去。我们相互鼓励、支持，终是走完了这条路，希望你们在接下来的人生之路越来越好，天下没有不散的宴席，我们后会有期！

家永远是最温暖的港湾，读研的道路少不了他们的支持与陪伴。在我沮丧失意的时候，他们总是会不厌其烦地聆听着我的牢骚，也给予了我精神和物质上最大的支持。没有你们，就没有我的今天，感谢你们一直以来无私的爱。

最后，感谢所有给予我帮助与支持的你们，谢谢！这里也向此次参与答辩的老师们致以最高的敬意！

参考文献

[1] 曹均阔, 陈国莲. 人机对话系统. 北京: 电子工业出版社, 2017

[2] J. Weizenbaum. Eliza-A computer program for the study of natural language communication between man and machine. Communications of the ACM, 1983, 26(1): 23-28

[3] R. S. Wallance. The anatomy of Alice. Parsing the Turing Test, Springer, 2009: 181-210

[4] B. F. Green, A. K. Wolf, C. Chomsky, K. Laughery. Baseball: An automatic question-answer. In: Proceedings of IRE-AIEE-ACM Computer Conference, Los Angeles, California, May 9-11, 1961, Association for Computing Machinery, 1961: 219-224

[5] D. Ferrucci, A. Levas, S. Bagchi, D. Gondek, E. T. Mueller. Watson: beyond jeopardy. Artificial Intelligence, 2013, 199: 93-105

[6] 赵阳洋, 王振宇, 王佩, 杨添, 张睿, 尹凯. 任务型对话系统研究综述. 计算机学报, 2020, 43(10): 1862-1896

[7] T. H. Wen, D. Vandyke, N. Mrksic, M. Gasic, L. M. Barahona, P. H. Su, et al. A network-based end-to-end trainable task-oriented dialogue system. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, April 3-7, 2017, Association for Computational Linguistics, 2017: 438-449

[8] M. Eric, C. D. Manning. Key-value retrieval networks for task-oriented dialogue. In: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrucken, Germany, August 15-17, 2017, Association for Computational Linguistics, 2017: 37-49

[9] J. D. Williams, K. Asadi, G. Zweig. Hybrid Code Networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, July 30 – August 4, 2017, Association for Computational Linguistics, 2017: 665-677

[10] S. Lee, Q. Zhu, R. Takanobu, Z. Zhang, Y. Zhang, X. Li, et al. ConvLab: Multi-

Domain end-to-end dialog system platform. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Florence, Italy, Jul 28 - August 2, 2019, Association for Computational Linguistics, 2019: 64-69

[11] W. Lei, X. Jin, M. Kan, Z. Ren, X. He, D. Yin. Sequicity: Simplifying task-oriented dialogue systems with single sequence to sequence architectures. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne, Australia, July 15-20, 2018, Association for Computational Linguistics, 2018: 1437-1447

[12] D. Bahdanau, K. Cho, Y. Bengio. Neural machine translation by jointly learning to align and translate. In: International Conference on Learning Representations, San Diego, United States, 2015: 1-15

[13] S. Wiseman, A. M. Rush. Sequence-to-sequence learning as beam-search optimization. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, November 1-5, 2016, Association for Computational Linguistics, 2016:1296-1306

[14] Y. Zhang, S. Sun, M. Galley, Y. Chen, C. Brockett, X. Gao, et al. DialoGPT: Large-scale generative pre-training for conversational response selection. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, July 5-10, 2020, Association for Computational Linguistics, 2020: 270-278

[15] H. Zhou, C. Zheng, K. Huang, M. Huang, X. Zhu. KdConv: A Chinese multi-domain dialogue dataset towards multi-turn knowledge-driven conversation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, July 5-10, 2020, Association for Computational Linguistics, 2020: 7098-7108

[16] C. Tao, J. Feng, R. Yan, W. Wu, D. Jiang. A survey on response selection for retrieval-based dialogues. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence Survey Track, 2021: 4619-4626

[17] R. Lowe, N. Pow, I. Serban, J. Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Prague, Czech Republic, September 2-4, 2015, Association for Computational Linguistics,

2015: 285-294

[18] M. Inaba, K. Takahashi. Neural utterance ranking model for conversational dialogue system. In: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Los Angeles, USA, September 13-15, 2016, Association for Computational Linguistics, 2016: 393-403

[19] X. Zhou, D. Dong, H. Wu, S. Zhao, D. Yu, H. Tian, et al. Multi-view response selection for human-computer conversation. In: Proceedings of the 2016 Conference on Empirical Methods in natural language processing, Austin, Texas, November 1-5, 2016, Association for Computational Linguistics, 2016: 372-381

[20] Y. Wu, W. Wu, C. Xing, M. Zhou, Z. Li. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, July 30 – August 4, 2017, Association for Computational Linguistics, 2017: 496-505

[21] Z. Zhang, J. Li, P. Zhu, H. Zhao, G. Liu. Modeling multi-turn conversation with deep utterance aggregation. In: Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, New Mexico, USA, August 20-26, 2018, Association for Computational Linguistics, 2018: 3740-3752

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 2017: 6000-6010

[23] X. Zhou, L. Li, D. Dong, Y. Liu, W. Zhao, D. Yu, et al. Multi-turn response selection for chatbots with deep attention matching network. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, July 15-20, 2018, Association for Computational Linguistics, 2018: 1118-1127

[24] J. Gu, Z. Ling, Q. Liu. Interactive matching network for multi-turn response selection in retrieval-based chatbots. In: Proceeding of the 28th ACM International Conference on Information and Knowledge Management, BeiJing, China, November 3-7, 2019, Association for Computing Machinery, 2019: 2321-2324

[25] C. Tao, W. Wu, C. Xu, W. Hu, D. Zhao, R. Yan. One time of interaction may not be enough: Go deep with an interaction-over-interaction network for response selection

in dialogues. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, July 28 – August 2, 2019, Association for Computational Linguistics, 2019: 1-11

[26] C. Tao, W. Wu, C. Xu, W. Hu, D. Zhao, R. Yan. Multi-representation fusion network for multi-turn response selection in retrieval-based chatbots. In: Proceedings of the 12th ACM International Conference on Web Search and Data Mining, Melbourne, VIC, Australia, February 11-15, 2019, Association for Computing Machinery, 2019: 267-275

[27] C. Yuan, W. Zhou, M. Li, S. Lv, F. Zhu, J. Han, et al. Multi-hop selector network for multi-turn response selection in retrieval-based chatbots. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, November 3-7, 2019, Association for Computational Linguistics, 2019: 111-120

[28] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang. Pre-trained models for natural language processing: A survey. Science China Technological Sciences, 2020, 63: 1872-1897

[29] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. lee, et al. Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, Louisiana, January 1-6, 2018, Association for Computational Linguistics, 2018: 2227-2237

[30] J. Devlin, M. Chang, K. Lee, K. toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota, January 2-7, 2019, Association for Computational Linguistics, 2019: 4171-4186

[31] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le. XLNet: Generalized autoregressive pretraining for language understanding. In: 33rd Conference on Neural Information Processing Systems, Vancouver, Canada, 2019: 5754-5764

[32] P. Budzianowski, I. Vulić. Hello, it’s GPT-2-How can I help you? Towards the use of

pre-trained language models for task-oriented dialogue systems. In: Proceedings of the 3rd Workshop on Neural Generation and Translation, Hong Kong, China, November 4, 2019, Association for Computational Linguistics, 2019: 15-22

[33] K. Clark, M. T. Luong, Q. V. Le, C. D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In: International Conference on Learning Representations, 2020: 1-18

[34] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In: International conference on Learning Representations, 2020: 1-17

[35] T. Whang, D. Lee, C. Lee, K. Yang, D. Oh, H. Lim. An effective domain adaptive post-training method for BERT in response selection. In: Proceedings of the Annual Conference of the International Speech Communication Association, Shanghai, China, October 25-29, 2020, INTERSPEECH, 2020:1585-1589

[36] J. Lu, X. Ren, Y. Ren, A. Liu, Z. Xu. Improving Contextual language models for response retrieval in multi-turn conversation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China, July 25-30, 2020, Association for Computing Machinery, 2020: 1805-1808

[37] J. Gu, T. Li, Q. Liu, Z. Ling, Z. Su, S. Wei, et al. Speaker-aware BERT for multi-turn response selection in retrieval-based chatbots. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020, Association for Computing Machinery, 2020: 2041-2044

[38] T. Whang, D. Lee, D. Oh, C. Lee, K. Han, D. Lee, et al. Do response selection models really know what's next? Utterance manipulation strategies for multi-turn response selection. In: Proceedings of the Thirty-fifth AAAI Conference on Artificial Intelligence, Palo Alto, California USA, February 2-9, 2021, Association for the Advancement of Artificial Intelligence, 2021: 14041-14049

[39] R. Xu, C. Tao, D. Jiang, X. Zhao, D. Zhao, R. Yan. Learning an effective context-response matching model with self-supervised tasks for retrieval dialogues. In: Proceedings of the Thirty-fifth AAAI Conference on Artificial Intelligence, Palo Alto,

California USA, February 2-9, 2021, Association for the Advancement of Artificial Intelligence, 2021: 14158-14166

[40] J. Han, T. Hong, B. Kim, Y. Ko, J. Seo. Fine-grained post-training for improving retrieval-based dialogues systems. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics, June 6-11, 2021, Association for Computational Linguistics, 2021: 1549-1558

[41] 谭真. 面向非结构化数据的知识图谱构建与表示技术研究[博士学位论文]. 长沙: 国防科技大学, 2018

[42] J. Li, A. Sun, J. Han, C. Li. A survey on deep learning for named entity recognition. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(1): 50-70

[43] 鄂海红, 张文静, 肖思琪, 程瑞, 胡莺夕, 周筱松等. 深度学习实体关系抽取研究综述. 软件学报, 2019, 30(6): 1793-1818

[44] S. Wu, Y. He. Enriching pre-training language model with entity information for relation classification. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019: 2361-2364

[45] A. H. Miller, A. Fisch, J. Dodge, A. Karimi, A. Bordes, J. Weston. Key-value memory networks for directly reading documents. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, November 1-5, 2016, Association for Computing Linguistics, 2016: 1400-1409