EZInterviewer：借助模拟面试生成器提升求职面试表现

Mingzhe Li*†北京大学 li_
mingzhe@pku.edu.cn

Xiuying Chen* CBRC，阿卜杜拉国王科技大学 CEMSE，阿卜杜拉国王科技大学
xiuying.chen@kaust.edu.sa

Weiheng Liao Made by DATA
Liaoweiheng@gmail.com

Yang Song BOSS直聘自然语言处理中心
songyang@kanzhun.com

Tao Zhang Boss直聘
kylen.zhang@kanzhun.com

Dongyan Zhao 北京大学
zhaody@pku.edu.cn

Rui Yan‡高瓴人工智能学院 中国人民大学
ruiyan@ruc.edu.cn

arXiv:2301.00972v1 [cs.CL] 3 Jan 2023

摘要

面试一直被视为招聘过程中最关键的环节之一。为了充分准备与招聘人员的面试，求职者通常会彼此进行模拟面试练习。然而，与同伴进行的模拟面试通常与真实的面试经验相去甚远：模拟面试官无法保证具备专业素养，也不太可能像真正的面试官那样行事。近年来，随着在线招聘的快速发展，招聘人员倾向于进行线上面试，这使得从真实面试官那里收集真实面试数据成为可能。本文提出了一款名为 EZInterviewer 的新型应用，旨在从线上面试数据中学习，并为求职者提供模拟面试服务。这项任务面临两方面的挑战：（1）虽然如今可以获取面试数据，但数据仍然稀缺；（2）生成有意义且相关的面试对话，需要深入理解简历和职位描述。为应对低资源挑战，EZInterviewer 使用一小批面试对话进行训练。其核心思路是将知识选择器与对话生成器解耦，从而减少依赖面试对话的参数数量，使大多数参数能够利用无依据对话以及并不稀缺的简历数据进行训练。具体而言，为了确保对话符合专业面试的要求，我们预训练了一个知识选择器模块，在职位与简历匹配过程中提取简历信息。我们还使用无依据对话预训练了一个对话生成器，使其学会生成流畅的回复。随后，对解码管理器进行微调，以结合两个预训练模块中的信息来生成面试问题。在真实的求职面试对话数据集上的评估结果表明，我们在生成模拟面试方面取得了令人鼓舞的成果。借助 EZInterviewer，我们希望让求职者更轻松地练习模拟面试。

CCS 概念

• **计算方法 → 自然语言生成**。

关键词

EZInterviewer，模拟面试生成，知识增强对话，在线招聘，低资源深度学习

ACM 参考格式：

Mingzhe Li、Xiuying Chen、Weiheng Liao、Yang Song、Tao Zhang、Dongyan Zhao、Rui Yan。2023年。EZInterviewer：借助模拟面试生成器提升求职面试表现。载于第十六届 *ACM* 网络搜索与数据挖掘国际会议论文集（*WSDM '23*），*2023*年*2*月*27*日至*3*月*3*日，新加坡，新加坡。ACM，纽约州纽约，美国，9页。https://doi.org/10.1145/3539597.3570476

1 引言

为了做好准备，求职者会进行模拟面试，以预测面试问题，并为真正面试时可能遇到的问题做好准备。然而，这种方法的效果并不理想，因为这些“模拟面试官”自身并没有面试经验，也不了解真实招聘人员会关注什么。模拟面试生成（MIG）为解决这一问题提供了一种可行方案。模拟面试生成器不仅能让面试更具成本效益，而且似乎切实可行，因为我们可以从求职者的简历中了解很多关于他们的信息，也可以从职位描述（JD）中了解职位本身。图1展示了 MIG 任务的示意图。

这项任务主要面临两大挑战。其一，收集有依据的面试数据极其耗时且成本高昂。如果没有足够的训练数据，

---

* 两位作者对本研究的贡献相同。
† 本研究工作完成于在 BOSS Zhipin 实习期间。
‡ 通讯作者：Rui Yan (ruiyan@ruc.edu.cn)。

允许免费出于个人或课堂用途制作本作品的全部或部分数字副本或纸质副本，前提是不得为营利或商业利益制作或分发副本，且副本须包含本声明及首页上的完整引文。必须尊重本作品中归其他作者所有的组成部分的版权。注明出处后，允许摘录。以其他方式复制、再版、发布到服务器或重新分发至列表，须事先获得明确许可和/或支付费用。请向 permissions@acm.org 申请许可。
WSDM '23，*2023*年*2*月*27*日至*3*月*3*日，新加坡，新加坡© 2023年 版权所有者/作者所有。出版权授权给 ACM。ACM ISBN 978-1-4503-9407-9/23/02...$15.00 https://doi.org/10.1145/3539597.3570476

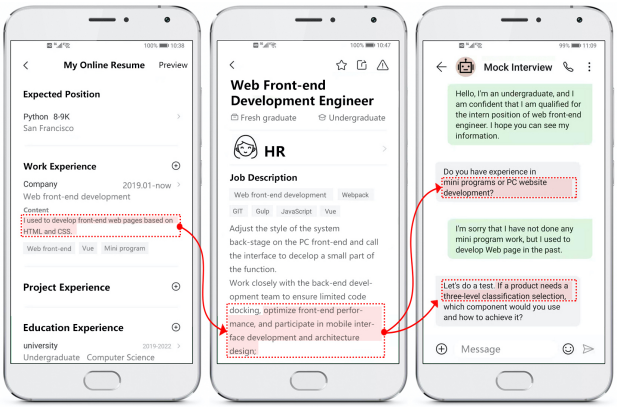

图1：模拟面试生成任务示例。系统根据候选人的工作经验，以及当前关于网页开发经验的对话，生成面试问题："如果产品需要进行三级分类选择，你会使用哪个组件？如何实现？"

此类对话生成模型的性能会大幅下降 [37]。第二个挑战是使知识支撑对话与候选人简历、职位描述以及先前的对话话语相关。这使得 MIG 成为一项复杂的任务，涉及文本理解、知识选择和对话生成。

本文提出了一种新颖的模拟面试生成器 EZInterviewer，旨在让面试准备更轻松。其核心思想是在低资源设置下训练 EZInterviewer：模型首先在大规模无依据对话和简历数据上进行预训练，然后在极少量以简历为依据的面试对话上微调。具体而言，**知识选择器**由一个用于编码简历的简历编码器 以及一个采用掩码自注意力机制的键值记忆网络 组成，负责从简历中选择相关信息并聚焦于这些信息，以帮助生成下一条面试话语。**对话生成器** 也包含两个组件：用于编码当前对话上下文的上下文编码器 ，以及响应解码器，后者负责在不使用简历信息的情况下生成下一条对话话语。这个不依赖知识的对话生成器通过**解码管理器** 与知识选择器协同工作，解码管理器会动态决定由哪个组件负责生成话语。

值得注意的是，解码管理器中的参数数量可以很少，因此只需要少量以简历为依据的面试对话。在真实世界的面试数据集上进行的大量实验表明，我们的模型是有效的。

总而言之，我们的贡献有三点：

•我们提出了一项新颖的模拟面试生成任务，这是对智能在线招聘的一项探索性研究，具有潜在的商业价值。

• 为应对低资源挑战，我们提出将知识选择器与对话生成器解耦，从而减少依赖面试对话的参数数量，使大多数参数能够利用大规模、无依据的对话和简历数据进行训练。

• 我们提出一种新颖的模型，可联合处理对话上下文、候选人简历和职位描述，并生成高度相关且具备知识感知能力的面试对话。

2 相关工作

多轮响应生成旨在根据前几轮中的话语，生成自然且与整体上下文相关的回复。 [36] 将多个话语拼接成一个句子，并利用RNN编码器或Transformer对长序列进行编码，从而将多轮对话简化为单轮对话。为了更好地建模多轮话语之间的关系，[4, 10] 在对每个话语进行编码后，引入了话语之间的交互。

由于人类对话几乎总是以外部知识为依据，缺乏知识落地已成为当前开放域对话系统与真实人类对话之间的主要差距之一 [8, 24, 35]。一系列研究 [20, 29]着重于根据上下文与非结构化文档知识之间的交互生成回复，而另一些研究 [22, 33] 则将知识图谱引入对话中。然而，这些模型通常在低资源设置下表现不佳。

为了解决低资源问题， [16] 提出在预训练单语 BERT 表示的基础上增强上下文相关的跨语言映射。 [28]扩展了元学习算法，利用从高资源领域学到的知识，提升低资源无监督神经机器翻译的性能。与上述方法不同， [37]提出了一种解耦响应解码器，以便将依赖于知识增强对话的参数与整个生成模型隔离开来。我们的模型更进一步，考虑了多轮对话场景中对知识的注意力变化。

3 模型

3.1 问题定义

对于求职者与面试官之间的输入多轮对话上下文 $U = \{u_1, u_2, \ldots, u_m\}$，其中 u_i 表示第 i 个话语，我们假设存在真实标注文本面试问题 $Y = \{y_1, y_2, \ldots, y_n\}$。 m是对话上下文中的话语编号，n是问题 Y中的词语总数。在第 i个话语中， $u_i = \{x_1^i, x_2^i, \ldots, x_{T_u^i}^i\}$。与此同时，面试中的候选人对应一份候选人简历 $R = \{(k_1, v_1), (k_2, v_2) \ldots, (k_{T_r}, v_{T_r})\}$，其中包含 T_r 个键值对，每个键值对代表简历中的一项属性。对于职位与简历匹配预训练任务，存在一份外部职位描述 $J = \{j_1, j_2, \ldots, j_{T_j}\}$，其中包含 T_j 个词语。我们的目标是生成一个面试问题 Y'，该问题不仅与对话上下文 U连贯，也与求职者的简历 R相关。

3.2 系统概述

本节提出我们的低资源模拟面试生成器（EZInterviewer）模型，如图 2 所示，该模型分为三个部分：

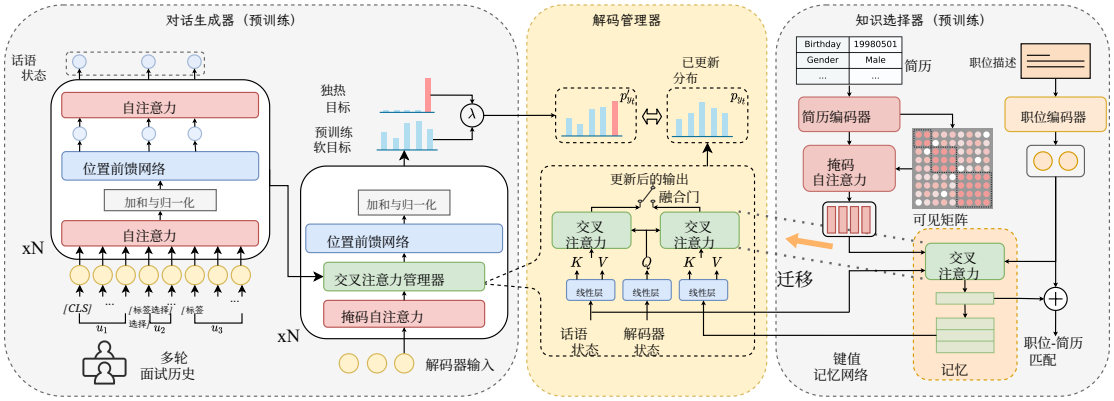

图 2：EZInterviewer 概览，由三个部分组成：(1) *Knowledge Selector* 从候选人简历中选择显著的知识信息；(2) *Dialog Generator* 在不使用简历知识的情况下预测下一个词；(3)*DecodingManager* 协调知识选择器和对话生成器的输出，以生成面试问题。

• 对话生成器根据前序子序列预测回复中的下一个词。在我们的模型中，我们使用大规模无依据对话对其进行预训练。

•*Knowledge Selector* 从候选人简历中选择显著的知识信息，用于面试问题生成。在我们的模型中，我们通过让知识选择器执行职位与简历匹配任务来增强其能力。

• *Decoding Manager* 协调知识选择器和对话生成器的输出，以预测面试问题。

值得注意的是，训练 EZInterviewer 模型时采用了两种预训练技术。首先，我们在职位匹配任务中对知识选择器进行预训练。这是因为，仅凭简历本身很难关注到恰当的内容，而在职位与简历匹配任务中则可以识别简历中的显著信息 [13, 34]。其次，使用大规模无依据对话对对话生成器的上下文编码器和响应解码器进行预训练，使其能够根据前序子序列预测回复中的下一个词。最后，解码管理器只依赖少量参数，负责协调这两个组件，生成以知识为依据的面试话语。

3.3 对话生成器

上下文编码器。 我们没有将对话上下文作为扁平序列进行处理，而是采用层次编码器 [3] 来捕捉话语内及话语间关系，该编码器由局部句子编码器和全局上下文编码器组成。对于句子编码器，为了对话上下文的语义进行建模，我们通过由 BERT [5]初始化的自注意力机制 (SAM) 学习每个话语 u_i 的表示：

$$h_j^i = \mathrm{SAM_u}(e(x_j^i), h_*^i). \tag{1}$$

我们提取“[cls]”位置的状态来表示话语状态，简称为 h^i。除了在每个话语中进行局部信息交换外，我们还让信息在多轮上下文中流动：

$$h_t^c = \mathrm{SAM_c}(h^t, h_*^c), \tag{2}$$

其中，h_t^c 表示 SAMc 中第 t 个话语的隐藏状态。

响应解码器。 响应解码器负责理解之前的对话上下文，并在不了解简历信息的情况下生成回复 [19]。我们的解码器同样遵循 Transformer 的风格。

具体而言，我们首先对掩码解码器输入应用自注意力，得到 d_t。基于 d_t，我们计算先前话语上的交叉注意力分数：

$$\alpha_t^c = \mathrm{ReLU}([d_t W_d (h_i^c W_h)^T]). \tag{3}$$

随后使用注意力权重 α_t^c获取上下文向量，形式为 $c_t = \sum_{i=1}^m \alpha_t^c h_i^c$。上下文向量 c_t被视为来自不同来源的重要内容，与解码器隐藏状态d_t 拼接后，用于计算目标词汇表上的分布：

$$P_v^w = \mathrm{Softmax}\left(W_o\left[d_t; c_t\right]\right). \tag{4}$$

预训练过程。虽然面试对话难以获取，但互联网上的在线对话十分丰富，也很容易收集。因此，我们在无依据的对话上预训练对话生成器。具体而言，在预训练过程中，我们使用上下文编码器首先对多轮历史对话上下文进行编码。然后，在第 t 个解码步，我们使用响应解码器预测回复中的第 t 个词。我们将损失设为目标词 y_t 的负对数似然：

$$Loss_g = -\frac{1}{n}\sum_{t=1}^n \log P_v^w(y_t). \tag{5}$$

3.4 知识选择器

简历编码器。 如图 2 所示，一份简历包含若干键值对（k_i，v_i）。大多数键和值字段由单个词语或短语组成，例如“技能”或“性别”，我们可以通过嵌入矩阵获得特征表示。具体而言，对于由单个词语或短语组成的每个键或值字段，我们会建立一个与先前矩阵不同的对应简历嵌入矩阵 e_r^i。然后，我们使用简历嵌入矩阵将每个字段词语 k_i或 v_i映射到高维向量空间，分别记为 $e_r^i(k_i)$ 或 $e_r^i(v_i)$。对于包含多个词语的字段，例如“工作经验”或“我曾经……”，我们将其记为$v_i = (v_i^1, \ldots v_i^{l_i})$，其中 l_i 表示当前字段的词语数量

字段。我们首先使用先前的词嵌入矩阵 e对其进行处理，然后通过一个 SAMR（与第 3.3 节中的 SAMu 类似）来建模词语之间的时间交互：

$$h_t^{r_i} = \mathrm{SAM_R}(e(v_i^j), h_{t-1}^{r_i}). \tag{6}$$

我们使用 SAMR 的最后一个隐藏状态，即 $h_{l_i}^{r_i}$来表示字段 v_i 的整体表示。

为简洁起见，在以下各节中，我们用 h_i^k和 h_i^v表示简历中的编码键值对（k_i，v_i）。

掩码自注意力。传统自注意力可以用于更新简历中各项的表示，因为它能够以不受距离限制的方式灵活地关联两个元素 [17]。然而，如 [21], 所示，过多地融入知识可能会使表示偏离其正确含义，这一问题称为知识噪声（KN）问题。在我们的场景中，简历信息分为若干部分，即 基本个人信息、工作经历和扩展工作经历，每部分包含数量不定的项目。每个部分中的项目彼此紧密关联，而不同部分可视为不同领域，它们之间的交互可能引入一定程度的噪声。为了解决这一问题，我们引入可见矩阵：属于同一部分的项目彼此可见，而项目之间的可见程度由语义表示的余弦相似度决定，即 $C_{i,\ j} = \cos_\mathrm{sim}(h_i^v,\ h_j^v)$。随后，缩放点积掩码自注意力定义如下：

$$\alpha_{i,j} = \frac{\exp\left((h_i^k W_q) C_{i,j} (h_j^k W_k)^T\right)}{\sum_{n=1}^{T_r} \exp\left((h_i^k W_q) C_{i,n} (h_j^k W_k)^T\right)}, \tag{7}$$

$$\hat{h}_i^v = \sum\nolimits_{j=1}^{T_r} \frac{\alpha_{i,j} h_j^v}{\sqrt{d}}, \tag{8}$$

其中，d 表示隐藏维度，C 是可见矩阵。随后，利用$\hat{h}_i^v$作为已更新的简历值表示。

键值记忆网络。键匹配的目标是计算简历中的每个属性与先前对话上下文之间的相关性。给定对话上下文 h^i，对于第 j 个属性对（k_j，v_j），我们计算 h^i 在 k_j 上的概率，即 $P(k_j|h^i)$，并将其作为匹配分数$\beta_{i,\ j}$。为此，我们利用上下文表示 h^i 来计算匹配分数：

$$\beta_{i,j} = \frac{\exp\left(h^i W_a h_j^k\right)}{\sum_{n=1}^{T_r} \exp\left(h^i W_a h_n^k\right)}. \tag{9}$$

由于上下文表示 h^i和简历键表示 h_j^k不在同一语义空间中，我们使用可训练键匹配参数W_a将这些表示转换到同一空间中。

作为上下文 h^i 与简历表格（k_j、v_j）中各个配对之间的相关性，匹配分数 $\beta_{i,\ j}$有助于找出最相关的配对，以生成正确的问题。因此，如公式 10 所示，知识选择器通过对存储的数值求和，从 KVMN 中读取信息 M_i，并引导后续的响应生成，因此我们有：

$$M_i = \sum\nolimits_{j=1}^{T_r} \beta_{i,j} \hat{h}_j^v, \tag{10}$$

^其中 h_j^v是数值 v_j的表示，$\beta_{i,\ j}$是对话上下文 h^i与键 k_j 之间的匹配分数。

预训练过程。实际上，简历知识包含各种专业和先进的科学概念，例如“Web前端”、“HTML”和“CSS”。对于不熟悉特定领域的人来说，这些术语很难理解，更不用说无法访问大规模简历知识支撑对话数据集的模型了。因此，知识选择器很难理解简历内容以及先前关于简历的上下文，也就难以选择接下来要关注的简历属性对。

另一方面，我们注意到，在职位与简历匹配任务中，捕捉简历中的决定性信息对于实现良好的匹配至关重要。例如，在背景相似的多位候选人中，招聘人员可能倾向于聘用具有特定经验的候选人 [34]。直观而言，对职位与简历匹配至关重要的键值对，也是求职面试中需要考虑的关键因素。因此，如果我们能让模型通过在大规模职位-简历数据上执行职位与简历匹配任务，学习简历中的显著信息，那么也有助于在面试问题生成过程中筛选显著信息。

具体而言，我们使用职位描述关注简历，以执行职位与简历匹配任务，作为知识选择器模块的预训练过程。如图 2 所示，职位编码器通过 SAMjd 对职位描述进行编码：

$$h_i^{jd} = \mathrm{SAM_{jd}}(e(j_i), h_{i-1}^{jd}), \tag{11}$$

其中，j_i表示职位描述中的第 i 个词，e（j_i）由先前的嵌入矩阵 e 映射得到。我们将 SAMjd 的最终隐藏状态 即 $h_{T_j}^{jd}$用作职位描述的整体表示，记作 h^{jd}。h^{jd}的作用与上下文表示 h^i 类似：它先关注简历中的键，然后用于对简历中的数值进行“加权”读取。我们用 m_{jd} 表示加权读取结果。

在训练过程中，我们首先通过职位与简历匹配任务对知识选择器进行预训练，该任务可以表述为一个分类问题 [26]。目标是提高正样本的分数，同时降低负样本的分数。具体而言，我们将 h^{jd}和m_{jd}拼接起来，因为已知向量拼接有利于匹配 [27]。随后，将拼接后的向量输入多层全连接前馈神经网络，得到职位与简历匹配分数s_{jr}，如下所示：

$$s_{jr} = \sigma\left(F_s([h^{jd}; m_{jd}])\right), \tag{12}$$

其中 [;] 表示拼接操作，输出为成功匹配的概率。我们将面试中的职位-简历对用作正样本，再将没有面试的职位-简历对用作负样本。

预训练后，职位描述被上下文表示取代，而键匹配和值组合过程保持不变。我们使用知识记忆 M存储选择结果，其中每个槽位存储公式 10 中的值组合结果 M_i。

3.5 解码管理器

解码管理器需要根据知识记忆和响应解码器生成恰当的词语。我们的想法源于对面试对话本质的一项观察：尽管对话以简历为基础，但其中的词语和话语并不总是与简历相关。因此，我们认为，回复的生成可以分解为两个互不相关的操作：（1）根据上下文选择词语，使对话连贯（对应对话生成器）；（2）根据额外的知识记忆选择词语，为对话提供依据（对应知识选择器）。这两个操作可以独立进行，这正是大型简历与职位匹配数据集和无依据对话数据集尽管看似与面试对话无关，却能在 MIG 任务中发挥重要作用的关键原因。

请注意，在第 §3.4 节中，我们将选定的知识 M_i 存储在知识记忆 M中。为了基于这些知识选择一个词，与响应解码器类似，我们使用 d_t 关注知识记忆中的每个槽位，从而获得知识上下文向量 g_t^k和输出解码器状态 d_t^{ko}。

响应解码器和知识选择器由解码管理器控制，解码管理器通过“融合门”决定在面试问题预测的每个步骤中应重点关注来自两侧的多少信息。

$$\gamma_f = \sigma\left(F_m(d_t)\right), \tag{13}$$

其中， d_t 是第 t 个解码器隐藏状态。然后，预测词语 y_t 的概率可表示为：

$$d_t^o = \gamma_f d_t^{wo} + (1-\gamma_f) d_t^{ko}, \tag{14}$$

$$P_v = \text{softmax}\left(W_v d_t^o + b_v\right). \tag{15}$$

关于优化目标，采用独热分布作为优化目标的生成模型总会面临过度自信问题，导致生成多样性较差 [32]。因此，除了真实标注独热标签 P外，我们还提出了一个软目标标签 P_v^w（见公式 4），该标签借鉴自第 3.3 节中的预训练对话生成器。强制解码管理器模拟预训练解码器，有助于其学习面试对话的上下文。我们通过编辑门 λ将独热标签与软标签相结合，如图 2 所示。具体而言，我们提出了一个平滑目标分布 P，来替代硬目标分布 P，如下所示：

$$P' = \lambda P + (1-\lambda) P_v^w. \tag{16}$$

其中， $\lambda \in [0,1]$ 是一个适应因子， P_v^w由公式 4 得出，而 P是作为独热分布的硬目标，为目标词语 y_t 赋予概率 1，并为其他词语赋予概率 0。

4 实验设置

4.1 数据集

本文使用中国最大的在线招聘平台“Boss直聘”1 提供的真实数据集开展实验。为保护候选人的隐私，所有用户记录均已匿名化，并删除了所有个人身份信息。该

[1]https://www.直聘网

表1：实验所用数据集的统计数据。

统计数据	数值
面试对话数据集	
简历总数	12,666
对话话语总数	49,214
每个对话上下文的平均轮数	4.47
每条话语的平均词语数	13.18
职位-简历数据集	
每份简历的键值对数	22
简历中每段工作经验的平均词数	72.80
简历中每段自我描述的平均词数	51.13
每份职位描述的平均词数	74.26
无依据对话数据集	
上下文-响应对总数	2,995,000
每个对话上下文的平均轮数	4
每条话语的平均词数	15.15

数据集包含 12,666 份简历、8,032 条职位描述和 49,214 条面试对话话语。数据集的统计数据汇总于表1。随后，我们使用基准中文分词工具包“结巴” [2]将每个句子切分为词语。

为了预训练知识选择器模块，我们再次使用来自“Boss直聘”的职位-简历匹配数据集 [34], 。训练集和验证集分别包含 355,000 对和 1,006 对职位-简历对。为了预训练对话生成器，我们选择了微博数据集 [2], , 该数据集包含从“微博”[3]收集的大量多轮对话。数据包含 2,990,000 对用于训练的上下文-响应对和 5,000 对用于验证的上下文-响应对。详细信息也汇总在表1中。

4.2 比较

我们将提出的模型与传统的知识无关对话生成基线以及知识相关对话生成基线进行比较。

- 知识无关对话生成基线：

Transformer [30]：仅基于注意力机制。**BERT** [5]：以 BERT 作为编码器来初始化 Transformer。**Di-aloGPT** [36]：提出了一种大型、可调节的神经对话响应生成模型，该模型在更接近真实对话的交流数据上进行训练。**T5-CLAPS** [14]：分别通过添加小幅和大幅扰动，为对比学习生成样本。

- 知识感知对话生成基线：

TMN [6]：基于 Transformer架构构建，并配备承载知识的外部记忆。**ITDD** [20]：渐进式编码多轮对话和知识，并通过审议技术解码回复。**DiffKS** [38]：利用多轮对话中已选择知识之间的差异信息进行知识选择。**DRD** [37]：采用基于无依据对话和文档的预训练技术，应对低资源挑战。**DDMN** [31]：动态跟踪多轮交互中的对话上下文，并将知识库知识融入生成过程

[2]https://github.com/fxsjy/jieba [3] https://www.weibo.com

表 2：全量数据集上的模型性能比较：自动评估指标。

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	极值	平均值	贪心	Dist-1	Dist-2	实体 F1	Cor
知识无关对话生成											
Transformer [30]	0.5339	0.3811	0.2836	0.2530	0.4859	0.7673	0.6803	0.0928	0.3157	0.3606	0.2711
BERT [5]	0.5671	0.3864	0.2735	0.2583	0.4861	0.7669	0.6792	0.0947	0.3558	0.3711	0.2894
DialoGPT [36]	0.5722	0.4015	0.3004	0.2697	0.4858	0.7670	0.6814	0.1001	0.3620	0.3843	0.3002
T5-CLAPS [14]	0.5846	0.4126	0.3020	0.2783	0.4837	0.7851	0.6674	0.0970	0.3702	0.3549	0.2870
知识感知对话生成											
TMN [6]	0.5437	0.3891	0.2963	0.2630	0.4841	0.7655	0.6811	0.0996	0.3299	0.3830	0.2652
ITDD [20]	0.5484	0.4009	0.2929	0.2656	0.4833	0.7650	0.6859	0.1055	0.3703	0.3661	0.2715
DiffKS [38]	0.5617	0.3898	0.2776	0.2441	0.4826	0.7830	0.6752	0.0937	0.3612	0.3672	0.2750
DRD [37]	0.5711	0.4001	0.2914	0.2548	0.4824	0.7813	0.6783	0.0867	0.3661	0.3825	0.2883
DDMN [31]	0.5693	0.4065	0.2968	0.2694	0.4831	0.7655	0.6811	0.0944	0.3640	0.3754	0.2869
Persona [9]	0.5532	0.3829	0.2715	0.2377	0.4823	0.7822	0.6783	0.0911	0.3598	0.3833	0.2928
EZInterviewer	**0.6106**	**0.4320**	**0.3284**	**0.2917**	**0.4893**	**0.7884**	**0.6886**	**0.1071**	**0.3747**	**0.3927**	**0.3145**
无预训练	0.5738	0.4029	0.2929	0.2599	0.4846	0.7833	0.6831	0.0981	0.3673	0.3819	0.3007
无知识记忆	0.5795	0.4127	0.3069	0.2754	0.4847	0.7841	0.6762	0.0979	0.3685	0.3803	0.3010
无知识选择	0.5775	0.4122	0.3067	0.2746	0.4781	0.7668	0.6787	0.1003	0.3691	0.3848	0.2994
无标签选择	0.6007	0.4232	0.3176	0.2821	0.4869	0.7863	0.6832	0.0969	0.3664	0.3902	0.3127

。**Persona** [9]：将个人记忆引入知识选择，以解决个性化问题。

4.3 实现细节

我们使用 TensorFlow [1] 在 NVIDIA GTX 1080 Ti GPU 上进行实验。对于我们的模型和所有基线，我们均采用下述相同设置。我们将输入对话截断至 100 个词语，每个话语包含 20 个词语；这是因为我们发现，将输入长度从 100 个词元增加到 200 个词元并未带来显著提升。最小解码步为 10，最大解码步为 20。词嵌入维度设为 128，隐藏单元数为 256。实验的批次大小为 256，词汇表由频率最高的 5 万个词语组成。我们使用 Adam优化器 [12] 作为优化算法。我们根据验证集上的表现选取了 5 个最佳检查点，并报告测试集上的平均结果。请注意，为了获得更好的性能，我们的模型基于 BERT 构建，解码过程与 Transformer [30]相同。最后，由于时间和记忆的限制，预训练基线采用了较小的设置。

4.4 评估指标

为评估 EZInterviewer 相较于基线的表现，我们采用了现有研究中广泛使用的以下指标。

基于重叠的指标。根据 [18], 我们使用 BLEU分数[25] 来衡量真实值与生成回复之间的 n元语法重叠。此外，我们采用相关性（Cor）计算生成问题与职位描述之间的词语重叠，以衡量生成的问题与招聘意图的契合程度。

嵌入指标。 我们计算生成结果与参考文本的词袋（BOW）嵌入之间的相似度，以衡量它们的语义匹配程度 [11]。具体而言，我们采用三种指标：1）贪心，即 根据余弦相似度，贪心地匹配两段话语中的词语；2）平均值，计算两段话语中平均词嵌入之间的余弦相似度 [23]；3）极值，计算两段话语中词嵌入的最大极值之间的余弦相似度 [7]。

表 3：人工评估结果：可读性（Read）、信息丰富度（Info）、意义性（Mean）、实用性（Use）、相关性（Rel）和连贯性（Coh）。

模型	对话级		访谈级			
	Read	Info	Mean	Use	Rel	Coh
DiffKS	1.79	2.01	1.87	2.03	1.99	2.10
DDMN	1.97	1.83	1.63	2.12	2.14	1.91
DRD	2.05	2.11	2.09	2.08	2.17	2.02
EZInterviewer	**2.42**▲	**2.51**▲	**2.39**▲	**2.46**▲	**2.57**▲	**2.38**▲

多样性。多样性得分 [15] 通过计算生成的回复中不同一元组和二元组的比例，衡量词语层面的多样性。

实体 F1。 实体 F1 通过对整个系统回复集合中的知识库实体的精确率和召回率进行微平均计算得出，用于评估模型能否根据提供的知识库 [31]生成与特定任务相关的实体。我们使用的实体提取自"Boss直聘"提供的实体词汇表。

人工评估指标。 除了自动评估之外，我们还采用人工评估。我们聘请了三位来自不同专业、受过良好教育的标注者，以双盲方式评估生成回复的质量。每位标注者都会对每个模型生成的 100 条随机抽样回复进行评分，分别从对话层面和面试层面进行评估。我们采用可读性 （回复的语法是否正确？）和信息丰富度 （回复是否包含信息丰富的词语？）来判断生成

回复在对话层面的质量。在面试层面，我们采用意义性（生成的问题是否有意义？）、实用性（求职者是否值得提前准备这个问题？）、相关性（问题是否与简历相关？）和连贯性（生成的文本与上下文是否连贯？）来评估模型的整体性能和用户体验质量。每项指标的评分范围为 1 到 3（1 = 差，2 = 平均，3 = 好）。

5 实验结果

5.1 整体性能

自动评估。表 2 列出了 EZInterviewer 与最先进的生成式基线模型之间的比较。

我们注意到，知识感知对话生成模型优于传统对话模型，这表明利用外部知识有助于生成相关回复。我们还发现，基于预训练的模型 DRD 优于其他基线模型，说明在大规模数据上进行预训练以初始化参数能够显著提升性能。值得注意的是，一些模型的实体 F1 分数较高，但 BLEU 分数较低；这表明这些模型倾向于从知识中复制必要的实体词语，却无法恰当地使用它们。

EZInterviewer 在所有自动评估指标上均优于基线。首先，与 DRD 相比，我们的模型将 BLEU-1 提高了 6.92%。在 Dist-1 多样性指标上，我们的模型比 DialoGPT 高出 6.99%，这表明生成的面试问题丰富多样，并能根据不同候选人的简历进行个性化定制。此外，我们的模型在实体 F1 上取得了 0.3927 的良好分数，该指标用于评估生成的问题以多大程度上以知识库为依据。最后，0.3145 的 Cor 分数表明 EZInterviewer 生成的问题符合职位描述，因此能够反映招聘人员的意图。总体而言，这些指标表明，我们的模型成功学会了面试官关注简历中哪些要点，并能恰当地将这些知识融入面试问题。

人工评估。表 3 列出了所有模型的人工评估结果。EZInterviewer 在所有指标上均表现最佳。具体而言，在可读性方面，我们的模型比 DiffKS 高出 35.20%，表明 EZInterviewer 能够减少语法错误并提升生成回复的可读性。在信息丰富度方面，我们的模型得分比 DDMN 高 0.68。这表明 EZInterviewer 能够捕捉简历中的显著信息。在面试层面，EZInterviewer 的实用性得分比 DRD 高出 18.27%，展现了其帮助求职者选择合适问题进行准备的能力。在相关性指标上，我们的模型大幅领先所有基线，表明生成的问题与面试流程密切相关。此外，我们的模型在意义性和连贯性指标上的表现也优于其他基线，说明其整体质量更高。

上述结果表明，EZInterviewer 能够生成有意义且实用的面试问题，同时保持面试对话流畅进行，就像人类招聘人员一样。请注意，人工评估的平均卡帕统计量在对话层面和面试层面分别为 0.51 和 0.48，

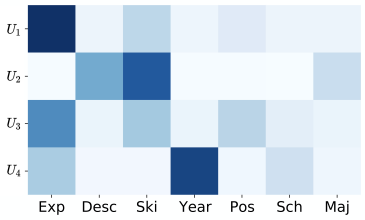

图 3：对话上下文与选定简历键之间键匹配的可视化，即，工作经历（Exp）、自我描述（Desc）、技能（Ski）、工作年限（Year）、期望职位（Pos）、学校（Sch）和专业（Maj）。U_i 表示第 i 个话语。

这表明标注者之间具有中等程度的一致性。为验证这些结果的显著性，我们还对模型与 DRD（带阴影背景的行）进行了双尾配对学生t检验。观察到的差异的统计显著性使用 ▲（或 ▼）表示，分别代表 $\alpha = 0.01$的强显著性（或弱显著性）。此外，在两个层面上，我们分别获得了 5×10^{-6} 和 3×10^{-4} 的平均p值。

5.2 消融研究

我们进行了消融研究，以评估模型中各个组件的贡献。结果如表 2 所示。

为了验证知识记忆的有效性，我们省略了对话上下文历史的知识选择，直接使用最后一个话语表示来选择知识。结果（见“无 KM”行）证实，利用历史对话的每一轮来选择知识并将其存入记忆，有助于生成更好的回复。为了确认知识选择是否有助于响应生成过程，我们将其从模型中移除，随后仅将每个话语的表示与所有简历内容相加，并将其存入记忆。这导致 BLEU-1 下降了 5.42%（见“无 KS”行），表明选择简历知识有益于响应生成。

5.3 KnowledgeSelector 分析

在第 § 3.4 节中，我们介绍了知识选择器的选择机制，并在公式 9 中得到了最终的注意力（匹配）分数。为研究知识选择器关注的具体信息，以及所选信息是否适合用于提出下一个面试问题，我们开展了一项案例研究，将知识选择器生成的匹配分数可视化，如表4和图3所示。历史记录中的第一条话语是“你以前从事过前端开发工作吗？”，知识选择器据此学习到，这条话语关注的是简历中的工作经验。因此，第四条话语“我有十多年的工作经验。”对工作年限和工作经验的关注度高于简历中的其他项目。这表明，知识选择器能够学会在生成每条话语时应关注简历中的哪一项。因此，当我们想请候选人“介绍一个与 React 相关的项目”时，知识选择器会关注简历中的工作经验，并生成模拟面试问题。

表4：基线和 EZInterviewer 生成的翻译后面试问题示例。表示知识选择器提取的信息和生成的词语，对话生成器生成的词语。

简历		面试
性别	Male	职位描述： 这项工作的主要内容包括基于 React 进行设计和开发 前端框架。要求能够高效完成前端 开发工作，并为客户提供优质服务。
Age	28	
教育	本科生	
专业	计算机科学	
工作年限	10	背景： U1：你之前从事过前端开发工作吗? U2：是的，我擅长 Vue、Node.js 以及其他一些技能。 U3：好的，那么你有 React 相关的经验吗? U4：有，我有超过 10 年的工作经验。
期望职位	前端工程师	
最低薪资	5	
最高薪资	6	
技能	Vue、Node.js、Java	
工	我从事前端设计 责基于 React 前端框架开展项目开发 基于 R 参与了系统架构设计过程。	真实标注：那么，你能介绍一个你做过的 React 相关项目吗? DDMN：你还会使用哪些前端框架? DRD：你好，能说说你之前的工作经历吗? EZInterviewe

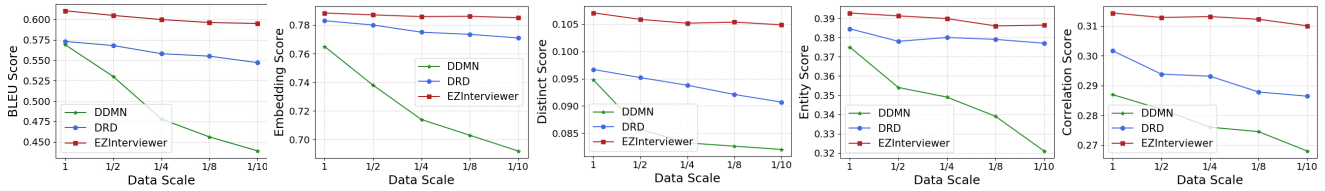

图4：自动 不同规模训练数据上 DDMN、DRD 和 EZInterviewer 的评估指标 .

5.4 训练数据规模的影响

为了解我们的模型和基线模型在低资源场景下的表现，我们先在完整训练数据集上对它们进行评估，再在训练数据集的较小子集上进行评估。图4分别展示了 DDMN、DRD 和 EZInterviewer 在完整训练数据集以及训练数据集的 1/2、1/4、1/8 和 1/10（数据规模）上的表现。可以观察到，随着训练数据集规模缩小，DDMN 在所有指标上都大幅下降，而基于预训练的模型即，DRD 和 EZInterviewer 的分数则相对稳定。这表明预训练是应对低资源挑战的有效策略。此外，我们的模型在所有数据规模下均优于 DRD，证明了我们模型的优越性。图4显示，EZInterviewer 最终在所有指标上都取得了最佳表现；即使只使用**1/10** 的训练数据，其表现也略胜于所有使用完整训练数据集训练的最先进的基线模型。

5.5 案例研究

表4展示了 EZInterviewer 和基线模型生成的一个翻译示例。我们观察到，EZInterviewer提出的问题不仅抓住了上下文，还借助恰当的知识拓展了对话。颜色标注突显了这一点：粉色词语 即"经验"和"React 框架"，是知识选择器从简历知识中提取的；蓝色词语 即 "那么，您能介绍一下……"和"基于"，与上下文紧密衔接，由对话生成器生成。相比之下，基线模型提出的问题虽能回应对话，却未能与简历知识建立联系。

6 结论

本文针对智能在线招聘这一新颖应用开展了一项试点研究，提出了 EZInterviewer，旨在为求职者提供模拟面试官。生成模拟面试时，系统会充分理解候选人的简历、职位要求、上下文中的先前话语，以及为有依据的面试选取的知识。为应对低资源挑战，EZInterviewer 使用一小批面试对话进行训练。其核心思路是将知识选择器与对话生成器解耦，以减少依赖面试对话的参数数量，从而使大多数参数能够利用无依据对话以及并非低资源的简历数据进行训练。我们通过大量实验验证了所提出的 EZInterviewer 方案的有效性。在使用完整训练数据及其小规模子集时，我们的模型在 BLEU、基于嵌入的相似度、多样性等多项指标以及人工评估方面均取得了最佳结果。尤其是，人工评估表明，我们提出的 EZInterviewer 能够提供令人满意的模拟面试，帮助求职者为真实面试做准备，使面试准备过程更加轻松。

致谢

我们谨向匿名审稿人致谢，感谢他们提出的建设性意见。本研究得到国家自然科学基金（NSFC项目编号：62122089）资助。Rui Yan得到北京人工智能研究院（BAAI）资助。

参考文献

[1] Martín Abadi、Paul Barham、Jianmin Chen、Zhifeng Chen、Andy Davis、Jeffrey Dean、Matthieu Devin、Sanjay Ghemawat、Geoffrey Irving、Michael Isard、Manjunath Kudlur、Josh Levenberg、Rajat Monga、Sherry Moore、Derek Gordon Murray、Benoit Steiner、Paul A. Tucker、Vijay Vasudevan、Pete Warden、Martin Wicke、Yuan Yu 和 Xiaoqiang Zhang。2016。TensorFlow：一种用于大规模机器学习的系统。载于*OSDI*。[2] Zhangming Chan、Juntao Li、Xiaopeng Yang、Xiuying Chen、Wenpeng Hu、Dongyan Zhao 和 Rui Yan。2019。通过增强型Wasserstein自编码器在连续空间中建模个性化，以实现响应生成。载于*2019*年自然语言处理实证方法会议暨第九届自然语言处理国际联合会议（*EMNLP-IJCNLP*）论文集。1931–1940。[3] Xiuying Chen、Hind Alamro、Mingzhe Li、Shen Gao、Rui Yan、Xin Gao 和 Xiangliang Zhang。2022。通过对比学习实现目标感知的抽象式相关工作生成。*arXiv*预印本 *arXiv:2205.13339*（2022）。[4] Xiuying Chen、Zhi Cui、Jiayi Zhang、Chen Wei、Jianwei Cui、Bin Wang、Dongyan Zhao 和 Rui Yan。2020。对话中的推理：通过上下文阅读理解改进响应生成。*arXiv*预印本 *arXiv:2012.07410*（2020）。[5] J. Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。2019。BERT：用于语言理解的深度双向Transformer预训练。载于*NAACL-HLT*。[6] Emily Dinan、Stephen Roller、Kurt Shuster、Angela Fan、Michael Auli 和 Jason Weston。2019。维基百科向导：知识驱动的对话代理。*ICLR*（2019）。[7] Gabriel Forgues、Joelle Pineau、Jean-Marie Larchevêque 和 Réal Tremblay。2014。利用词嵌入引导对话系统。载于*Nips，workshop*，第2卷。[8] Chenpeng Fu、Zhixu Li、Qiang Yang、Zhigang Chen、Junhua Fang、Pengpeng Zhao 和 Jiajie Xu。2019。面向开放世界知识图谱补全的多重交互注意力模型。载于*Web*信息系统工程——*WISE 2019*：第*20*届国际会议，香港，中国，*2020*年*1*月*19*日至*22*日，论文集。630–644。[9] Tingchen Fu、Xueliang Zhao、Chongyang Tao、Ji-Rong Wen 和 Rui Yan。2022。千人眼中有千个哈姆雷特：利用个人记忆增强知识支撑对话。*ACL*（2022）。[10] Shen Gao、Xiuying Chen、Chang Liu、Li Liu、Dongyan Zhao 和 Rui Yan。2020。学习用贴纸进行回复：一种统一多模态与多轮对话的框架。载于*2020*年万维网大会论文集。1138–1148。[11] Xiaodong Gu、Kyunghyun Cho、Jung-Woo Ha 和 Sunghun Kim。2019。Dialog-WAE：基于条件Wasserstein自编码器的多模态响应生成。载于国际学习表征会议。https://openreview.net/forum?id=BkgBvsC9FQ [12] Diederik P Kingma 和 Jimmy Ba。2015。Adam：一种随机优化方法。*ICLR*（2015）。[13] Ran Le、Wenpeng Hu、Yang Song、Tao Zhang、Dongyan Zhao 和 Rui Yan。2019。迈向有效且可解释的人岗匹配。载于第*28*届*ACM*信息与知识管理国际会议论文集。1883–1892。[14] Seanie Lee、Dong Bok Lee 和 Sung Ju Hwang。2021。结合对抗扰动的对比学习，用于条件文本生成。载于第九届国际学习表征会议，*ICLR 2021*，线上活动，奥地利，*2021*年*5*月*3*日至*7*日。OpenReview.net。[15] J. Li、Michel Galley、Chris Brockett、Jianfeng Gao 和 W. Dolan。2016。促进神经对话模型多样性的目标函数。*NAACL*（2016）。[16] Juntao Li、Chang Liu、Jian Wang、Lidong Bing、Hongsong Li、Xiaozhong Liu、Dongyan Zhao 和 Rui Yan。2020。面向全球电子商务的跨语言低资源集合到描述检索。*AAAI*（2020）。[17] Mingzhe Li、Xiuying Chen、Shen Gao、Zhangming Chan、Dongyan Zhao 和 Rui Yan。2020。VMSMO：学习为视频新闻文章生成多模态摘要。载于 *2020* 年自然语言处理实证方法会议（*EMNLP*）论文集。9360–9369。[18] Mingzhe Li、Xiuying Chen、Min Yang、Shen Gao、Dongyan Zhao 和 Rui Yan。2021。吸引力的风格与内容二元性：通过解耦学习撰写吸睛标题。*AAAI*（2021）。[19] Mingzhe Li、Xiexiong Lin、Xiuying Chen、Jinxiong Chang、Qishen Zhang、Feng Wang、Taifeng Wang、Zhongyi Liu、Wei Chu、Dongyan Zhao 等。2022。关键词与实例：一种统一文本生成混合粒度的层次对比学习框架。载于计算语言学协会第*60*届年会论文集（第*1*卷：长篇论文）。4432–4441。[20] Zekang Li、Cheng Niu、Fandong Meng、Yang Feng、Qian Li 和 Jie Zhou。2019。用于文档知识支撑对话的带审议解码器的增量式 Transformer。载于计算语言学协会第*57*届年会论文集。12–21。[21] Weijie Liu、Peng Zhou、Zhe Zhao、Zhiruo Wang、Qi Ju、Haotang Deng 和 Ping Wang。2020。K-BERT：利用知识图谱实现语言表示。载于人工智能促进协会年会论文集，第34卷。2901–2908。[22] Zeming Liu、Haifeng Wang、Zheng-Yu Niu、Hua Wu、Wanxiang Che 和 Ting Liu。2020。面向多类型对话的对话式推荐。*arXiv*预印本 *arXiv:2005.03954*（2020）。[23] Jeff Mitchell 和 Mirella Lapata。2008。语义组合的基于向量的模型。*NAACL-HLT*（2008），236–244。[24] Lei Niu、Chenpeng Fu、Qiang Yang、Zhixu Li、Zhigang Chen、Qingsheng Liu 和 Kai Zheng。2021。利用多重交互注意力进行开放世界知识图谱补全。万维网 24，1（2021），419–439。[25] Kishore Papineni、Salim Roukos、Todd Ward 和 Wei-Jing Zhu。2002。BLEU：一种机器翻译自动评估方法。载于*ACL*。ACL，311–318。[26] Chuan Qin、Hengshu Zhu、Tong Xu、Chen Zhu、Liang Jiang、Enhong Chen 和 Hui Xiong。2018。提升人才招聘中的人岗匹配：一种基于能力感知神经网络的方法。载于第*41*届 *ACM SIGIR* 信息检索研究与发展大会。25–34。[27] Aliaksei Severyn 和 Alessandro Moschitti。2015。利用卷积深度神经网络学习短文本对的排序。载于第*38*届 *ACM SIGIR* 信息检索研究与发展国际会议论文集。373–382。[28] Yunwon Tae、Cheonbok Park、Taehee Kim、Soyoung Yang、Mohammad Azam Khan、Eunjeong Park、Tao Qin 和 Jaegul Choo。2020。面向低资源无监督神经机器翻译的元学习。*arXiv*预印本 *arXiv:2010.09046*（2020）。[29] Zhiliang Tian、Wei Bi、Dongkyu Lee、Lanqing Xue、Yiping Song、Xiaojiang Liu 和 Nevin L Zhang。2020。用于响应生成中按需知识整合的响应预期记忆。*arXiv*预印本 *arXiv:2005.06128*（2020）。[30] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。2017。注意力机制即全部所需。载于神经信息处理系统进展。5998–6008。[31] Jian Wang、Junhao Liu、Wei Bi、Xiaojiang Liu、Kejing He、Ruifeng Xu 和 Min Yang。2020。用于端到端多轮任务型对话系统的双动态记忆网络。载于第*28*届国际计算语言学会议论文集。4100–4110。[32] Yida Wang、Yinhe Zheng、Yong Jiang 和 Minlie Huang。2021。通过自适应标签平滑实现多样化对话生成。*arXiv*预印本 *arXiv:2105.14556*（2021）。[33] Hongcai Xu、Junpeng Bao 和 Junqing Wang。2020。采用改进的元学习进行基于知识图谱的主动对话生成。*arXiv*预印本 *arXiv:2004.08798*（2020）。[34] Rui Yan、Ran Le、Yang Song、Tao Zhang、Xiangliang Zhang 和 Dongyan Zhao。2019。面试选择揭示你对市场的偏好：利用画像记忆提升求职简历匹配。载于第*25*届 *ACM SIGKDD* 知识发现与数据挖掘国际会议论文集。[35] Xiangliang Zhang、Qiang Yang、Somayah Albaradei、Xiaoting Lyu、Hind Alamro、Adil Salhi、Changsheng Ma、Manal Alshehri、Inji Ibrahim Jaber、Faroug Tifratene 等。2021。COVID-19疫情期间全球对话的兴衰与情绪转变。人文与社会科学通讯 8，1（2021），1–10。[36] Yizhe Zhang、Siqi Sun、Michel Galley、Yen-Chun Chen、Chris Brockett、Xiang Gao、Jianfeng Gao、Jingjing Liu 和 Bill Dolan。2019。DialoGPT：用于对话响应生成的大规模生成式预训练。*arXiv*预印本 *arXiv:1911.00536*（2019）。[37] Xueliang Zhao、Wei Wu、Chongyang Tao、Can Xu、Dongyan Zhao 和 Rui Yan。2020。低资源知识支撑对话生成。*ICLR*（2020）。[38] Chujie Zheng、Yunbo Cao、Daxin Jiang 和 Minlie Huang。2020。面向知识支撑对话生成的差异感知知识选择。载于计算语言学协会论文集：*EMNLP 2020*。115–125。