面向 App 评论响应的语义检索和生成框架

范国栋[1)] 陈世展[1)] 肖建茂[2)] 吴洪越[1)] 张 璐[1)]
薛 霄[1)] 王忠杰[3)] 冯志勇[1)]

[1)](天津大学智能与计算学部 天津 300350)
[2)](江西师范大学软件学院 南昌 330022)
[3)](哈尔滨工业大学计算学部 哈尔滨 150001)

摘 要 在应用程序维护过程中,移动应用(Mobile Application,App)评论的响应为应用程序开发者提供了用户反馈机制,对应用的评级产生积极影响.为了减轻响应大量用户评论的工作负担,开发者通常采用自动化的机制回复评分或跟进用户问题.当前流行使用序列到序列(Sequence to Sequence,Seq2seq)的深度生成模型或融合信息检索的方法来生成用户评论的响应.然而,现有检索方法没有考虑句子的语义相似性,生成模型没有考虑用户评论与检索到评论之间的差异,导致模型对知识的利用不佳,降低了响应质量.为了解决这些问题,本文提出了一种面向 App 评论响应的语义检索和生成框架(A Semantic Retrieval and Generation Framework,SRGen).首先,基于响应相似但评论不一定相似的现象,通过自监督学习方法对 Sentence-BERT(SBERT)模型微调.然后,利用 SBERT 获得名称、评分、评论信息的向量表示,检索知识库中 Top-k 最相似的评论-响应对.最后,根据检索到的评论与待响应评论的差异和相应响应内容,生成评论的响应.实验表明,与现有的基线工作相比,SRGen 在 BLEU 指标下提升了 12.4%,在 ROUGE 指标下提升了 9.4%.

关键词 软件维护;用户评论;App 评论响应;语义检索;Seq2seq
中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2022.02528

A Semantic Retrieval and Generation Framework for App Review Response

FAN Guo-Dong[1)] CHEN Shi-Zhan[1)] XIAO Jian-Mao[2)] WU Hong-Yue[1)] ZHANG Lu[1)]
XUE Xiao[1)] WANG Zhong-Jie[3)] FENG Zhi-Yong[1)]

[1)](*College of Intelligence and Computing, Tianjin University, Tianjin* 300350)
[2)](*School of Software, Jiangxi Normal University, Nanchang* 330022)
[3)](*Faculty of Computing, Harbin Institute of Technology, Harbin* 150001)

Abstract In the process of application maintenance, the response of mobile application (App) reviews provides developers with a user feedback mechanism, which has a positive impact on the rating of the application. To reduce the workload of responding to a large number of user comments, developers usually adopt an automated mechanism to respond to users' ratings or follow-up user questions. Currently, it is popular to use Sequence to Sequence (Seq2seq) generation models or fusion information retrieval methods to generate responses to user reviews. However, the existing

收稿日期:2022-01-20;在线发布日期:2022-09-30.本课题得到国家自然科学基金重点基金(61832014,62032016)、国家自然科学基金(61972276,62102281)、江西省教育厅科技攻关项目(GJJ210338)资助.**范国栋**,博士研究生,主要研究方向为服务计算、认知服务.E-mail:guodongfan@tju.edu.cn.**陈世展**,博士,副教授,中国计算机学会(CCF)会员,主要研究领域为服务计算、软件生态系统挖掘与分析.**肖建茂**(通信作者),博士,讲师,中国计算机学会(CCF)会员,主要研究领域为服务计算、智能化软件工程.E-mail:jm_xiao@jxnu.edu.cn.**吴洪越**(通信作者),博士,副教授,中国计算机学会(CCF)会员,主要研究领域为服务计算、边缘计算.E-mail:hongyue.wu@tju.edu.cn.**张 璐**,博士研究生,主要研究方向为服务计算、认知服务.**薛 霄**,博士,教授,中国计算机学会(CCF)会员,主要研究领域为服务计算、群体智能、计算实验.**王忠杰**,博士,教授,中国计算机学会(CCF)杰出会员,主要研究领域为服务计算、服务工程.**冯志勇**,博士,教授,中国计算机学会(CCF)杰出会员,主要研究领域为服务计算、知识工程、计算机认知.

retrieval methods do not consider the semantics of the sentences, and the generative models do not consider the difference between user reviews and retrieved reviews, which leads to poor utilization of knowledge by the model and reduces the quality of the response. To solve these problems, in this paper, we propose a Semantic Retrieval and Generation Framework named SRGen, which consists of a retrieval model and a generation model. Firstly, the Sentence-BERT (SBERT) model is fine-tuned in a self-supervised learning manner according to different reviews that may have similar responses especially in the same application. Then the Top-k most similar review-response pairs are retrieved in the knowledge base using the joint vector representation of the name, rating, and review information obtained by SBERT. Finally, according to the difference between the retrieved comment and the original comment and the corresponding response content, the response of the comment is generated by a generative model. Experimental results show that the SRGen improves by 12.4% under the metric of BLEU, and 9.4% under the metric of ROUGE.

Keywords　software maintenance; user review; app review response; semantic retrieval; Seq2seq

1　引　言

智能手机订阅数持续增长[1]，为移动应用(Mobile Application，App)带来了大量的用户，并为开发者带来了繁杂的响应工作. App 通常通过应用程序商店分发. 它允许用户通过评分或评论对下载的 App 提供反馈，然后开发者对评分或评论进行响应，而这种交互式反馈可作为用户和开发者之间的重要沟通渠道. App 的响应流程如图 1 所示. 用户下载一个 App 后，可根据用户体验对其评价. 也可以编辑、删除已提交的评价，但不能重复对一个 App 评价. 此外，用户可以赞同和不赞同他人的评价. 开发者可以查看所有的评价、按星级和版本信息过滤评价、对评价排序或公开响应，但不能删除评价. 每个评价只能有一个响应，如果开发者响应了一个评价，将会通过电子邮件发送给用户[2].

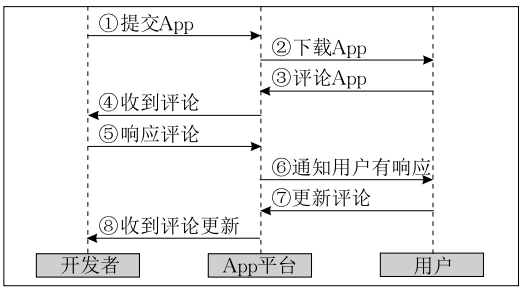


图 1　App 的响应流程

App 评论响应的目的是进一步了解用户的需求或解决用户的问题，让用户通过评论和评分来提供反馈、提出问题，公开表达他们对应用的满意度. 正向的评论和评级是获取和留住用户的重要因素. 开发者可以通过响应用户反馈来维护应用程序，以换取更好的评论和评级. 有研究证实[3]响应最多的类型是表扬、特征请求和错误报告，其中表扬应用程序通常伴随着的特征请求或小问题. 据统计[4]，65%的用户在有不好的体验后，会给 App 留下打分或者评论，只有 15%的用户会考虑下载只有 2 星的 App. 用户收到响应后比不响应有 6 倍的概率提高他的打分，34%的问题只需响应即可解决，无需升级 App. 因此，在有大量同质应用的应用商店中，保证用户体验、提升 App 的星级对开发者来说是非常重要的.

然而，收集并人工响应大量的评论，对于开发者来说即耗时又耗力，迫切需要一些自动化工具来辅助完成评论响应工作. 为了方便开发人员更好的进行 App 的开发与维护，一些研究针对评论进行了自动的分类和总结[5-7]. Srisopha 等人[4]研究发现 iOS App Store 中，评分、评论长度、发布时间、情感、书写方式都对是否需要响应评论有重要的影响. Hassan 等人[8]分析评论、响应机制的动态性质，研究了 Google Play 商店中 450 万条评论、2328 种免费下载应用的 126 686 条响应，一个主要发现是评论是非静态的，并确定了指导开发人员响应的四种模式. 这些工作为评论响应提供了策略指导，提高了用户的满意度，增进了用户和开发者之间的沟通，但对于评论的响应工作，开发者仍需要深度参与.

为了解决海量评论的信息过载问题，观点分析、情感分析、命名实体识别、主题抽取等[5,9-10]方法相

继被提出，挖掘出规则的结构化数据以供检索或推荐.然而，多个独立的任务会累计误差[11]，导致检索精度降低.没有固定结构的自然语言的语义检索难点在于文本的语义和句法属性的捕获[12]，及预训练模型中词向量的各向异性问题[13].而面向 App 评论信息中，用户对 App 中概念理解不同，甚至是应用情景不同，这样即使是相同的响应解决方案，评论的差异也可能较大.因此，需要引入更多的特征信息，例如，App 名，评论与回复内容等.监督学习可以提升相似度度量效果[14]，但如何构建监督信号，指导语义相似度计算也是一难点.与一般的会话系统不同，App 评论响应生成通常只有一轮，其响应内容通常与特定领域甚至特定 App 相关，其性能也依赖于领域知识的建立[15].其次 App 的评论响应数据呈现长尾分布，导致部分评论响应难以训练.针对生成特定 App 评论响应的问题，Farooq 等人[16]提出 AARSYNTH，使用特定 App 的信息来增强序列到序列(Sequence to Sequence，Seq2seq)模型，在内容相关性上取得了到更好的效果.针对自动生成特定领域评论响应问题，基于机器翻译(Neural Machine Translation，NMT)，Gao 等人[17]提出一个基于 Seq2seq 的生成式模型 RRGen，将 keywords 和 App 的属性融入其中，获得评论响应的序列.但是，该模型高度依赖于外部工具，例如使用 SURF 确定预定义关键字，使用 SentiStrength 估计评论的情感，外部工具的性能对模型结果产生影响很大且不可控.针对该问题，提出 CoRe[15]，通过输入 App 的描述信息、检索到的响应和用户的评论，将序列到序列的生成任务转换为摘要提取任务.然而，回复内容相同或相似的评论之间的差异可能很大，因此无法仅对评论内容进行精确的相似度检索.本文将评论响应的相似度作为监督信号来训练检索模型，以建立查询评论与知识库中评论-响应等信息的语义关联.同时，针对 App 评论响应的领域相关性，将 App 名称和评分信息也建模到检索模型中，综合判断与知识库中内容的相似度.

现有研究方法，没有考虑评论检索的语义信息，以及待响应评论与检索到评论之间的差异性.本文提出一种面向 App 评论响应的语义检索和生成框架(A Semantic Retrieval and Generation Framework，SRGen).首先，构建正负样本以微调 Sentence-BERT(SBERT)[18]模型.然后，通过该模型，综合 App 名称、评分和评论得到句子表示向量.进一步，从语义层面检索到 Top-k 相似评论-响应对.最后，对比待响应评论与检索到评论内容上的差异，以及检索到的评论响应，生成最终评论的响应信息.贡献如下：

(1)针对检索不准确的问题，本文提出一种自监督的 App 评论向量表示方法.基于响应的相似性，结合名称、评分、评论、响应等内容，采用 TF-IDF 算法获取正样本，随机生成负样本，微调 SBERT.然后得到评论句子表示向量，并检索相似的响应-评论对.

(2)为更好的利用检索到的信息，本文构建基于语义检索的生成模型.首先，使用 App 评论的向量表示，基于余弦相似度检索知识库，得到 Top-k 条最相似的评论-相应对.然后，利用待响应评论与检索到评论的差异和相应响应，生成评论的响应内容.

(3)与现有的基线工作对比，SRGen 在 BLEU 指标下提升了 12.4%，在 ROUGH 指标下提升了 9.4%.同时验证了构建正负样本的有效性、在检索方面的优势以及语义检索对头尾部数据生成结果的影响.此外，代码已开源到了 GitHub① 供其他相关人员进一步深入研究 .

本文第 2 节介绍背景知识；第 3 节介绍 SRGen 的框架主要包括检索过程和生成过程；第 4 节介绍实验设计及结果分析；第 5 节介绍本文的相关工作；第 6 节对全文进行总结，并简要介绍下一步工作.

2 背景知识

本节介绍相关知识，包括研究动机、相关定义，及文本的生成与检索任务当中常用的技术，如 Seq2seq 模型、Attention 机制和 SBERT.

2.1 研究动机

如图 2 所示，有两条用户的评论及响应，从响应的内容看，回复是 App 或叫做领域相关的，如果结合检索到相应的领域知识，能更好的辅助评论响应的生成.例如，图中的“the following steps：…”，这些内容是通用的领域知识.

从检索的角度看，响应内容一致，两条评论理应相似.但直接对比评论内容，却很难确定它们的相似程度.第一条报告了一个 bug 并请求修复，第 2 条是用户不知如何使用某项功能.将此称之为响应相似但评论不一定相似的现象，即响应相似性.基于响应

① https://www.github.com/guodongfan/SRGen

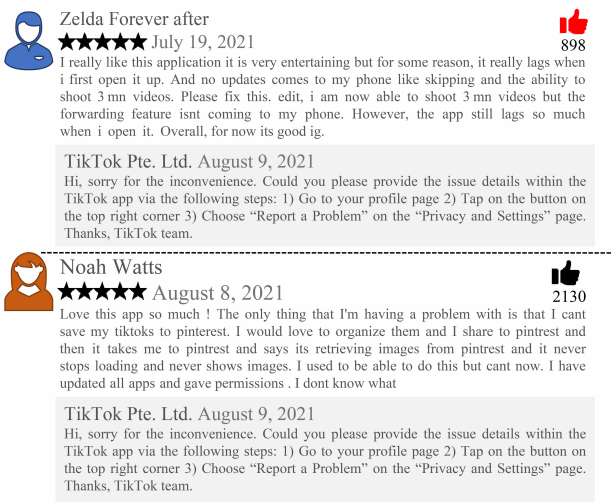


图 2　Motivation 例子

相似性，构造度量评论内容语义的监督信号，训练语义查询模型，查询语义相似度. 最终，模型的生成由查询到的知识指导.

综上，受响应相似性的启发，得出以下结论：知识库的构建可辅助评论生成，同时可利用相似响应为知识检索模型的训练提供监督信号.

2.2　相关定义

定义 1.　Knowledge Base(KB)知识库. 应用商店中，已有的评论-回复内容构成 KB. 其中每条数据 P 包含评论、响应、评分、App 名称信息，表示为四元组(Review，Response，Rating，Name).

定义 2.　语义相似性检索. 设 Q 为查询，即想要被响应的评论，包含(Review，Rating，Name). 语义相似度检索为查询 Q 检索知识库 KB，得到 Top-k 最相似的数据$\{P_i\}$，其中 $0\leq i\leq k$.

定义 3.　评论响应的生成. 设用户评论 Review 序列为 X，开发者响应 Response 序列 Y. 给定训练数据集 D，由若干评论-响应对序列(X,Y)组成，每对包含长度为 n 的评论输入序列 $X=\{x_1,x_2,\cdots,x_n\}$ 和对应长度为 m 的响应输出序列. $Y=\{y_1,y_2,\cdots,y_m\}$. 目标是学习一个概率映射函数 $g(\cdot)$，它允许为一个新的评论序列 X' 及其 k 个相似的检索数据$\{P_i\}$，其中 $0\leq i\leq k$，预测一个可能的目标序列 Y'.

2.3　Seq2seq 模型

Seq2seq[19]模型由编码器(Encoder)和解码器(Decoder)组成，最开始用于机器翻译中，给定输入序列时能够有效预测输出序列的概率分布.

给定输入序列 $X=(x_1,x_2,\cdots,x_t,\cdots,x_m)$，其中 x_t 表示输入序列中的第 t 个词. 模型输出序列 $Y=(y_1,y_2,\cdots,y_t,\cdots,y_n)$，其中，$y_t$ 表示输出序列中第 t 个词. m 和 n 分别为序列 X 和 Y 的长度. Seq2seq 模型需要通过学习编码 X 和解码 Y 来建模条件概率分布 $P(Y|X)$. 与基于 RNN 的语言模型类似，可以使用自回归的方式对 $P(Y|X)$进行求解.

$$P(Y|X)=\prod_{t=1}^{n}p(y_t|Y_{<t},X) \tag{1}$$

Seq2seq 模型一般采用两个 RNN 结构分别作为 Encoder 和 Decoder. 它们每个时间步 t，编码状态和解码状态都由 RNN 的隐状态向量表示：

$$\boldsymbol{h}_t=Encoder(h_{t-1},e(x_t)) \tag{2}$$

$$\boldsymbol{s}_t=Decoder(\boldsymbol{s}_{t-1},\boldsymbol{e}(y_{t-1}),\boldsymbol{h}_m) \tag{3}$$

其中，$\boldsymbol{h}_t$ 表示编码器在时间步 t 的编码状态向量，$\boldsymbol{s}_t$ 表示解码器在时间步 t 的解码状态向量. $\boldsymbol{h}_m$ 是编码器编码完输入 X 后最后的隐状态向量，可看成对 X 中所有输入信息的编码总结，解码器的起始状态用 $\boldsymbol{h}_m$ 进行初始化. $\boldsymbol{e}(y)$ 指词 y 对应的编码向量. 每得到一个时间步的解码状态后，模型通过一个多层感知机计算在整个词典上的概率分布，以此得到对应位置的输出.

$$P(y_t|Y_{<t},X)=\mathrm{Softmax}(mlp(\boldsymbol{s}_t)) \tag{4}$$

2.4　Attention 机制

注意力机制使用广泛[20]，使用序列到序列模型做生成任务时，X 中越靠后的词对解码生成序列的影响越大，而靠前的词容易被忽略. 因为最终的编码状态 $\boldsymbol{h}_m$ 更容易记住靠后词的信息. 注意力机制可以解决这一问题，Bahdanau 等人[21]改变了解码器的计算方式.

$$\boldsymbol{s}_t=Decoder(\boldsymbol{s}_{t-1},\boldsymbol{e}(y_{t-1}),c_t) \tag{5}$$

$$c_t=\sum_{i=1}^{m}\alpha_{it}\boldsymbol{h}_i \tag{6}$$

$$\alpha_{it}=\mathrm{Softmax}(e_{it})=\frac{\exp(e_{it})}{\sum_{k=1}^{m}\exp(e_{kt})} \tag{7}$$

$$e_{it}=V_a\cdot\tanh(W_aS_{t-1}+U_a\boldsymbol{h}_i) \tag{8}$$

其中，$V_a\in\mathbb{R}^d$，$W_a\in\mathbb{R}^{d\times d}$，$U_a\in\mathbb{R}^{d\times d}$ 是网络的可训练参数. e_{it} 表示 t 时刻解码状态向量 $\boldsymbol{s}_{t-1}$ 受到编码状态向量 $\boldsymbol{h}_i$ 的影响程度.

2.5　SBERT

Sentence-BERT(SBERT)[18]，一种基于 BERT 网络构建的预训练模型，它使用孪生网络和三元组网络结构来推导语义上有意义的句子表示，可使用余弦相似度进行文本相似度比较，其结构如图 3 所

示.给定一个锚定的句子 a,一个正样本句子 p,一个负向本句子 n,三元组损失调整网络使得句子 a 与 p 的距离拉近,而使得句子 a 与 n 拉远.

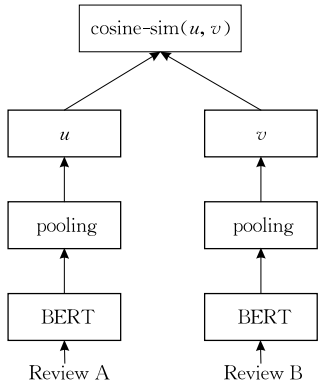


图 3 Sentence-BERT

3 SRGen 框架实现

如图 4 所示,模型的实现分为检索阶段和生成阶段,其中检索阶段又分为数据准备如图 4(a)、自监督训练如图 4(b)和向量检索如图 4(c)、生成阶段如图 4(d)检索阶段.

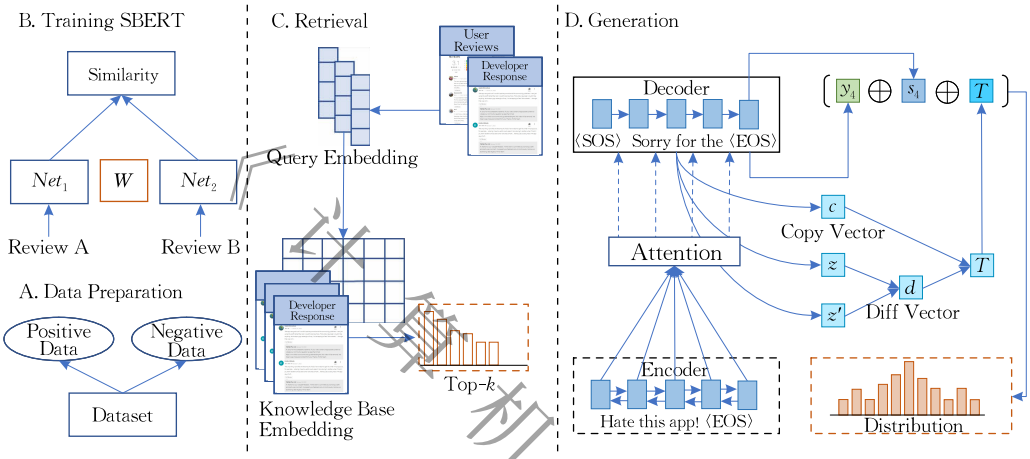


图 4 SRGen 架构图

3.1 检索阶段

本节分为三部分,数据准备、自监督训练和向量检索.首先构建评论响应的正负样本,然后通过该样本自监督学习微调 SBERT.最后通过 SBERT 得到评论数据的表示向量,进而检索到 Top-k 条相似的评论-响应对.

3.1.1 数据准备

基于响应相似性的动机,使用 TF-IDF 算法,对知识库 KB 中的响应内容建模,即具有相似响应内容的数据作为检索模型的正样本 Ground Truth.负样本通过随机的方式获取.正样本很多都是 App 相关的,但尾部的一些 App 的评论响应数据很少,此时可利用其他 App 评论-响应的数据作为补充.但仅仅使用完全随机的方式会导致,模型认为只要相同 App 的数据就是相似的,所以还随机将相同 App 下的评论-响应作为负样本.训练集通过知识库中的数据(除去与自身),响应内容之间比对得到,测试集和验证集通过查询样本与知识库中的响应数据比对得到.总体来讲包含:(1)通过 TF-IDF 计算响应内容的相似度,确定正样本;(2)通过完全随机的方式取得负样本;(3)在相同 App 下随机得到另一条负样本.具体见算法 1.

算法 1. 生成数据集.

输入:kb: Knowledge Base(知识库)

qd: Query Data(查询数据)

输出:spl: Sampling Dataset(正负样本集合)

$spl \leftarrow \varnothing$ //初始化

FOR each $(i, n_i, r_i, rw_i, rp_i \in qd)$ DO //遍历查询

  $q \leftarrow (n_i, r_i, rw_i)$ //q 包含 App 名称、评分及评论

  $\text{top}k \leftarrow \text{tfidf}(kb, rp_i)$ //响应之间的相似数据

  FOR each$(j, n_j, r_j, rw_j, rp_j \in \text{top}k)$ DO //遍历 top$k$

    $base \leftarrow (n_j, r_j, rw_j, rp_j)$ //拼接数据

    IF $(i \neq j)$ THEN //不是同一数据时

      $s_+ \leftarrow (q, base, 1.0)$ //正样本对

      $spl_{\text{rand}} \leftarrow \text{rand}(kb)$ //随机负样本

      $s_- \leftarrow (q, spl_{\text{rand}}, J(rp_i, rp_{\text{rand}}))$ //负样本对

      $spl_{\text{app}} \leftarrow \text{rand}(n_i, kb)$ //随机相同 App 下样本

      $s_- \leftarrow (q, spl_{\text{app}}, J(rp_i, rp_{\text{app}}))$ //负样本对

13.　　$spl \leftarrow spl \cup (s_+, s_-, s_-)$ //样本存入集合中
14.　　END
15.　END
16. END

算法 1 根据查询输入的不同分别生成训练、测试或验证集. 输入参数中 *KnowledgeBase*(简写为 *kb*),见定义 1,表示知识库的内容,这里也指训练集的数据. *QueryData*(简写为 *qd*)为查询输入集数据,输出 *Sampling*(简写为 *spl*),输入输出可为训练、测试或验证集的数据. 遍历需要查询的集合(Line 2～3),使用 TF-IDF 查询 Top-*k* 最相近的评论响应(Line 4). 遍历查询到的 Top-*k* 条样本每个为 *base*(Line 5～6),*q* 与知识库中查询出来的对应的内容 *base* 构成了正样本 Ground Truth(Line 8). 随机从知识库中取一条数据为负样本(Line 9～10),随机从知识库中取一条与查询同名 n_i 的数据为第二个负样本(Line 11～12). 最后把样本都加入到要输出的集合中(Line 13). 由于评论的响应很多是 App 相关的,在进行负采样的时候只是完全随机采集一条数据,会导致模型认为只要是相同 App 的就相似,所以还增加了相同 App 的随机数据的负采样本. 此外,随机过程中难免会随机到非负的样本,SRGen 对负样本加入了相似度分值,即通过 Jaccard 相似度,如式(9),计算两个响应之间共同单词占全部单词的比例,从而可以防止错误的负采样.

$$J(P_1, P_2) = \frac{|P_1 \cap P_2|}{|P_1 \cup P_2|} \tag{9}$$

其中 P_1 为查询评论的响应,P_2 为知识库中评论的响应,计算结果为两个响应句子词汇的交集除以两个响应句子词汇的并集的大小.

3.1.2　自监督训练

训练的过程,使用算法 1 所取得的训练集和测试集对 SBERT 微调. 模型每次输入一条知识库的数据 *A*、一条检索数据 *B* 以及它们之间的相似度,范围为[0,1]. 如图 5 所示,SRGen 将知识库中的被检索数据以 Name is App name [SEP] Rating is score [SEP] Review sentence [SEP] Response sentence 排列,查询数据以 Name is App name [SEP] Rating is score [SEP] Review sentence 的形式组织. 这样可以使得模型自动学习到名称、评分、评论内容等构成的句子向量表示,其中 App name 使用“.”分割应用的包名,取第一位之后的词按顺序排列以空格排列,例如,com.picsart.studio 分割为 picsart studio.

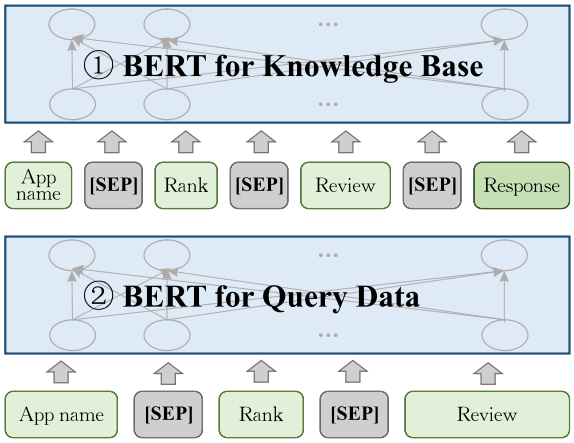


图 5　自监督学习中句子的组织

3.1.3　向量检索

如算法 2 所示,将用户的评论连同已知的应用名称、评分输入到 SBERT 中编码为低维向量表示(Line 2～3),然后分别计算得到的向量表示与知识库中每个数据的向量表示的余弦距离,如式(10),排序后得到 Top-*k* 最相似的评论-响应对(Line 6～8).

$$\mathrm{Cos}(X, Y) = \frac{X \cdot Y}{\|X\|_2 \cdot \|Y\|_2} \tag{10}$$

通过将长序列数据转为低维稠密向量,降低了计算的空间复杂度,并利用 Tensor 的并行运算技术,并行计算多个查询,减少了检索时间.

算法 2.　相似评论检索.

输入:*querys*: Query Reviews(需要响应的评论)
　　kb: Knowledge Base(知识库)
　　ssize: Split Size(查询 Batch 的大小)
输出:Top-*k* Similar Reviews(top*k* 评论相似数据)

1. top*k* ← ∅ //初始化
2. *kb_emb* = encode(*kb*) //编码知识库数据
3. *q_emb* = encode(*querys*) //编码查询数据
4. *split_emb* = split(*q_emb*, *ssize*) //将查询数据拆分
5. FOR each(*emb* ∈ *split_emb*) DO//遍历查询数据
6.　　*sims* = cos_sim(*kb_emb*, *emb*) //相似度计算
7.　　*sorted* = argsort(*sims*) //根据相似度排序
8.　　top*k* ← top*k* ∪ get_top*k*(*sorted*) //相似数据
9. END

3.2　生成阶段

生成阶段模型如图 6 所示,由编码器、解码器、特征融合及概率分布等组成. 使用用户评论和知识库中检索到的评论-响应对,生成响应内容.

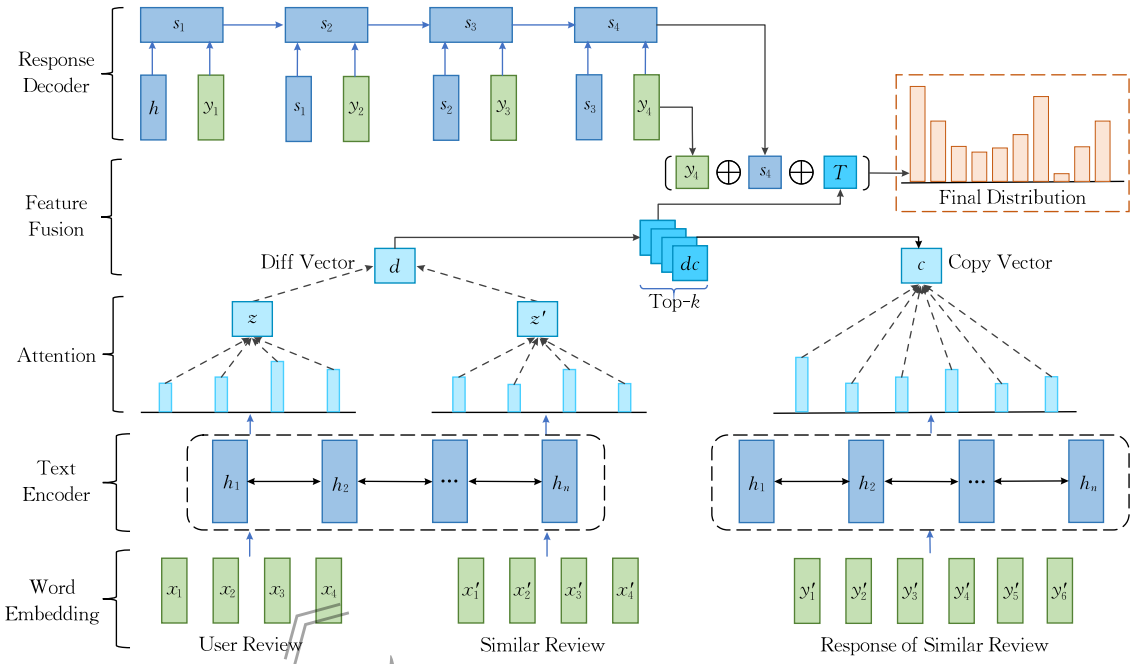


图 6 生成模型结构

3.2.1 编码解码

设 $X=(x_1,x_2,\cdots,x_m)$ 为评论序列，$Y=(y_1,y_2,\cdots,y_m)$ 为响应序列. 利用评论序列 X 检索到 K 个相似的评论-响应对，分别为 $X'_k=(x'_{1k},x'_{2k},\cdots,x'_{mk})$ 和 $Y'_k=(y'_{1k},y'_{2k},\cdots,y'_{mk})$，其中 $1\leq k\leq K$. 得到每个词的向量表示后，采用双向 GRU 对序列的嵌入向量进行编码，如式(11)～(13).

$$e^{(X)},h^{(X)}=biGRU(X) \tag{11}$$

$$e^{(X'_k)},h^{(X'_k)}=biGRU(X'_k) \tag{12}$$

$$e^{(Y'_k)},h^{(Y'_k)}=biGRU(Y'_k) \tag{13}$$

解码器使用单向 GRU，产生一个状态向量 $\boldsymbol{s}_i$，如式(14).

$$\boldsymbol{s}_i=GRU(\boldsymbol{s}_{i-1},\boldsymbol{e}(y_{i-1})) \tag{14}$$

其中，$\boldsymbol{s}_{i-1}$ 为解码器上个状态向量，$\boldsymbol{e}(y_{i-1})$ 表示词 y_{i-1} 的向量表示，后简写为 y_{i-1}.

3.2.2 特征融合

使用解码器得到的状态向量 $\boldsymbol{s}_i$ 后，将其作为查询向量，与经过双向 GRU 得到用户评论序列的隐状态向量 $\boldsymbol{h}_i^x$，$0\leq i\leq \text{len}(X)$ 计算注意力得到向量 $\boldsymbol{z}$，如式(15). 使用查询向量 $\boldsymbol{q}$，如式(16)，与检索得到的评论序列隐状态 $\boldsymbol{h}_i^{x'}$，计算注意力得到向量 $\boldsymbol{z}'$，如式(17). 计算检索到的评论响应序列隐状态 $\boldsymbol{h}_i^{y'}$ 的注意力，得到向量 $\boldsymbol{c}$，称之为拷贝向量，表示生成模型用来直接拷贝检索到内容的向量，如式(18).

$$\boldsymbol{z}=Attention(\boldsymbol{h}^x,\boldsymbol{s}_i) \tag{15}$$

$$\boldsymbol{q}=W_q\boldsymbol{h}_n^{x'} \tag{16}$$

$$\boldsymbol{z}'=Attention(\boldsymbol{h}^{x'},\boldsymbol{q}) \tag{17}$$

$$\boldsymbol{c}=Attention(\boldsymbol{h}^{y'},\boldsymbol{s}_i) \tag{18}$$

通过用户实际评论向量 $\boldsymbol{z}$ 与检索到的评论向量 $\boldsymbol{z}'$，融合生成差异向量 $\boldsymbol{diff}$，表示评论与检索到评论之间的差异，如式(19). 融合差异向量 $\boldsymbol{diff}$ 与拷贝向量 $\boldsymbol{c}$ 做非线性融合，得到差异拷贝向量 $\boldsymbol{dc}$，表示根据差异决定拷贝检索到的响应内容，如式(20). 进一步叠加 Top-k 个差异拷贝向量得到向量 $\boldsymbol{T}$，如式(21).

$$\boldsymbol{diff}=g(\boldsymbol{z},\boldsymbol{z}',\boldsymbol{z}-\boldsymbol{z}') \tag{19}$$

$$\boldsymbol{dc}=\boldsymbol{g}(\boldsymbol{d},\boldsymbol{c}) \tag{20}$$

$$\boldsymbol{T}=\{dc_i\}_{i=1}^K \tag{21}$$

3.2.3 概率生成

在解码器步骤 i，结合输入词的向量 $\boldsymbol{y}_i$、状态向量 $\boldsymbol{s}_i$、Top-k 个差异拷贝向量 $\boldsymbol{T}$，来计算整体分布. 生成词 y_{i+1} 的概率分布 $P_{y_{i+1}}$，如式(22).

$$P_{y_{i+1}}=\text{Softmax}(mlp(y_i,\boldsymbol{s}_i,\boldsymbol{T})) \tag{22}$$

其中，mlp 为全连接层，输出层的大小为整个词典的大小.

3.2.4 损失函数

训练中，整个序列的损失计算如式(23)所示，是每个时间步的平均负对数似然损失.

$$L(\theta)=-\frac{1}{|y|}\sum_{t=1}^{|y|}\log(p_t(y_t|y_{<t},x,\{x'_k\}_1^K,\{y'_k\}_1^K)) \tag{23}$$

其中 θ 为模型参数，K 为检索评论-响应对的个数.

4　实验设计与分析

本节介绍实验设计与分析，包括数据集、环境配置、参数设置、实验结果及分析讨论.

4.1　数据集、环境及参数设置

本文使用文献[15]的数据集来验证模型的有效性，数据呈现的长尾分布可视化如图 7 所示，前两个 App 的数据占到了 1/3 的数据，大部分的 App 数据在尾部. 该模型在配备 Intel Xeon Processor(Skylake, IBRS) 16 核、内存 64GB、Tesla V100 16GB 显卡的设备上进行训练.

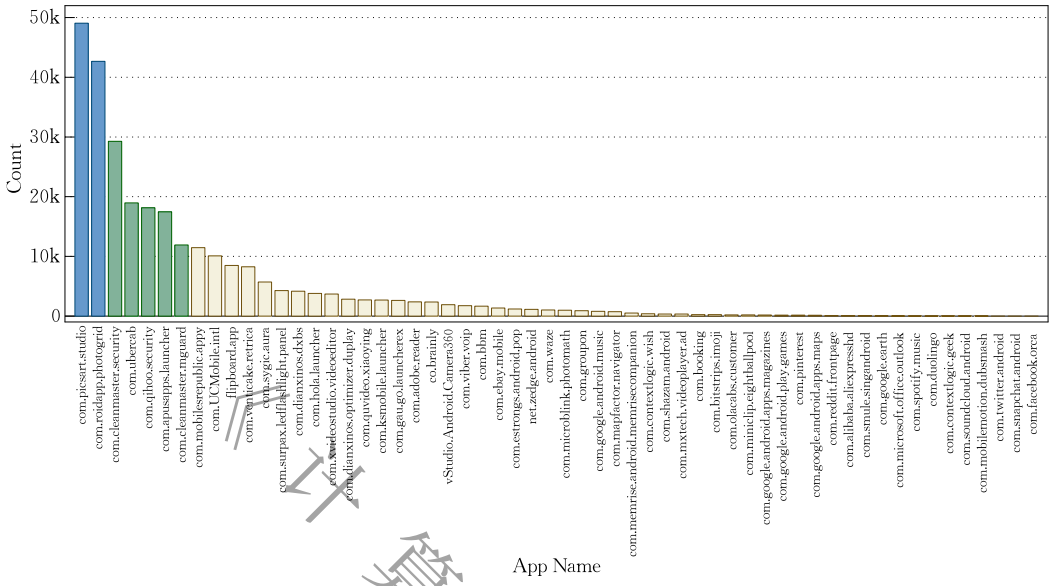


图 7　数据集的数据分布

根据链接①，综合模型精度、速度以及序列长度，选择 all-MiniLM-L6-v2 作为 SBERT 的预训练模型. 另外，序列长度不能超过 Transformer 模型支持的最大长度，BERT 模型输入 WordPiece 最多为 512，对应约 300～400 个英语单词，较长的文本被截断. 模型参数见表 1.

表 1　模型参数

参数名	参数值
Loss Function	ADAM
Learning Rate	0.001
Weight Decay	1e-5
Hidden Size	200
Type of RNN	GRU
Number of Layers	1
Word Embedding Vector	GLOVE
Word Embedding Dimension	200
Top-k	5
Epochs	3
Batch Size	100
Dropout	0.1
Max Vocabulary Size	10000
Review Length	100
Response Length	100

实验设计受以下 5 个问题指导.

RQ1：和基线方法相比，SRGen 是否有优势？

RQ2：不同参数下 SRGen 的精度如何？

RQ3：SRGen 在检索方面比 BERT 和 TF-IDF 等方法相比是否有优势？

RQ4：头部数据和尾部数据训练效果的差异，以及语义检索在其中的作用是什么？

RQ5：生成结果可以用于开发者的实际评论响应吗？

4.2　评测工具

4.2.1　BLEU 评测

BLEU[22]度量一组机器产生的翻译句子集合与一组人工翻译句子集合的相似程度. BLEU 的计算如式(24).

$$BLEU\text{-}N(\hat{y},y)=B(\hat{y},y)\exp\Big(\sum_{n=1}^{N}w_n\log P_n(\hat{y},y)\Big) \tag{24}$$

式中，n 为当前匹配 n-gram 的长度，最大长度 N 设置为 4，w_n 是 n-gram 匹配权重，$B(\hat{y},y)$ 是用来对过短候选翻译的惩罚系数，$P_n(\hat{y},y)$ 是使用当前 n-gram 匹配对候选机器翻译的评估，P_n 计算如式(25).

$$P_n(\hat{y},y)=\frac{\sum\limits_{C\in\{cdts\}}\sum\limits_{gram_n\in C}Count_{\text{clip}}(gram_n)}{\sum\limits_{C'\in\{cdts\}}\sum\limits_{gram'_n\in C'}Count(gram'_n)} \tag{25}$$

① https://sbert.net/docs/pretrained_models.html

式中，$Count_{clip}(gram_n)$为 n-gram 同时在 candidate ($cdts$)和给定标准 reference 中出现次数最大值，$Count(gram_n')$为所有 candidate 中 n-gram 的次数.

4.2.2 ROUGH 评测

ROUGH 评测[23]用来检测机器自动产生的文本摘要和一组标准文本摘要的相似度，如式(26). 使用ROUGE-1、ROUGE-2 和 ROUGE-L 的 $F1$ 分数，分别测量在参考摘要和被评估的摘要之间的单词重叠、二元重叠和最长公共序列.

$$ROUGE\text{-}N=\frac{\sum_{S\in\{Ref\}}\sum_{gram_n\in S}Count_{match}(gram_n)}{\sum_{S'\in\{Ref\}}\sum_{gram_n'\in S'}Count(gram_n')} \tag{26}$$

式中，$ROUGE\text{-}N$ 是 n-gram 召回率，即在标准和自动文本摘要中均出现的 n-gram 占标准文本摘要中 n-gram 的比例. $Count_{match}$ 为 n-gram 同时在一组标准文本摘要和自动文本摘要中出现的次数的最大值，$Count(gram_n)$为 n-gram 在标准文本摘要集合中出现的次数.

4.3 结果分析

4.3.1 方法效果分析(RQ1)

如表 2 和表 3 所示，本文对比了 SRGen 与经典的 NMT 算法、RRGen 算法[17]、CoRe[15]算法、CoRe＋SBERT 算法、SRGen-Similar 算法在 BLEU 和 ROUGE 指标下的效果. CoRe＋SBERT 表示 CoRe 的检索结果替换为 SRGen 的检索结果，SRGen-Similar 表示 SRGen 去掉检索到的相似评论，即不考虑待生成响应评论与检索到评论之间的差异.

表 2 BLEU 实验结果

模型	BLEU-4	P_1	P_2	P_3	P_4
NMT	19.43	37.34	18.47	14.94	13.84
RRGen	33.82	50.18	32.90	28.91	27.37
CoRe	40.18	56.62	39.93	35.76	33.91
CoRe＋SBERT	44.16	59.89	43.59	39.20	37.17
SRGen-Similar	44.12	60.54	43.94	39.48	37.47
SRGen	45.17	60.80	44.65	40.20	38.15

表 3 ROUGE 实验结果

模型	ROUGE-1	ROUGE-2	ROUGE-L
NMT	37.00	18.36	36.02
RRGen	47.78	30.65	47.11
CoRe	52.60	36.66	51.59
CoRe＋SBERT	56.59	40.74	55.76
SRGen- Similar	56.67	40.52	55.70
SRGen	57.28	41.54	56.48

可以发现，在 BLEU 指标下，SRGen 比 CoRe 模型提升了 12.4%，在 ROUGH 指标下，SRGen 比 CoRe 模型提升了 8.9%. 句子语义检索对模型的性能提升最为显著. 将 CoRe 模型的检索部分替换为微调后的 SBERT 后，性能有显著提升从 40.18 提升到 44.16. 当没有考虑用户评论与检索到评论的差异时，模型 SRGen-Similar 的 BLEU-4 为 44.12，考虑后的 BLEU-4 为 45.17，指标提升 1.05，证明利用检索到的评论与待响应评论之间的差异信息有助于提高生成质量.

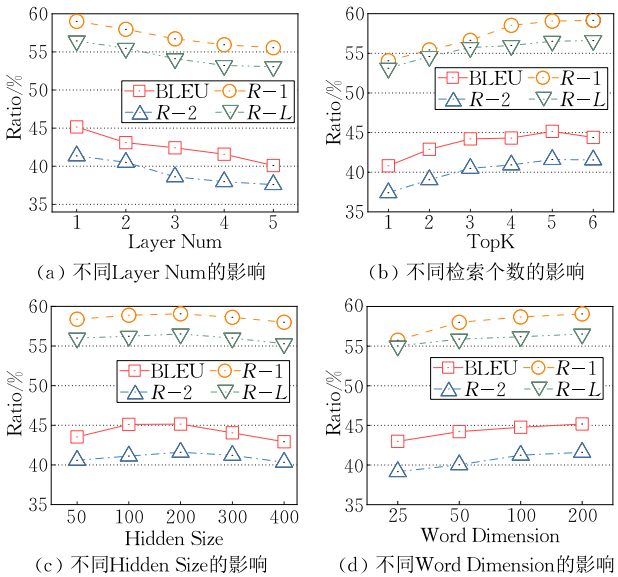
4.3.2 不同参数的影响(RQ2)

本文还比较了在不同参数设置下 SRGen 的准确性. 分析四个参数，即 Layer Num 循环神经网络的层数、检索个数、Hidden 向量的维度、Word Embedding 的维度.

Layer Num. 如图 8(a)所示，设置 Layer Num 为 1 到 5. 可见 Layer Num 为 1 时，BLEU 取得最高值，ROUGE-L 值 Layer Num 为 1 也取得了最高值. 随着 Layer Num 的增多，会引入更多的参数，网络性能呈现下降趋势，表明更多的层数并不能带来更好的性能. 由于 Layer Num 为 1 时候参数量较少，实验效果更好，本文将 Layer Num 的值设置为 1.

检索个数. 如图 8(b)所示，设置 Top-k 从 1 到 6. 实验效果随着检索数的增多而增多，到达 5 时达到峰值，随之下降. 这表明，多个检索的内容利于生成更好的响应内容，但也会导致网络在融合 Top-k 条检索信息时参数增加，使性能降低. 因此本文将检索个数设置为 5.

Hidden 维度. 如图 8(c)所示，实验设置 Hidden Size 为 50 到 400. 当 Hidden Size 较小时候如 50 和 100，BLEU、ROUGHE-1、ROUGE-2 和 ROUGE-L



(a) 不同Layer Num的影响 (b) 不同检索个数的影响

(c) 不同Hidden Size的影响 (d) 不同Word Dimension的影响

图 8 不同参数设置下模型的性能

的精度都不高. 到达 200 时候到达峰值，随后下降. 达到 400 时，精度降到最小值. Hidden Size 过大时，模型复杂度增高，会导致过拟合现象的出现，使模型性能下降. 而 Hidden Size 过小时，模型的表达能力会不足. 因此本文将 Hidden Size 的值设置为 200.

Word Embedding 维度. 如图 8(d)所示，设置 Word Embedding 维度分别为 25、50、100、200. 随着 Word Embedding 维度的增加，实验效果更好. 说明维度高的预训练 Word Embedding 有利于网络的训练. 因此本文将其维度设置为 200，使用的预训练文件为 glove. twitter. 27B. 200d. txt.

4.3.3　语义检索优势分析(RQ3)

为了证明语义检索方面的优势，我们利用不同检索方法得到的数据，辅助生成评论响应. 对比了直接使用 BERT、GLOVE、TF-IDF 和 Ground Truth 的结果，其中 Ground Truth 通过 3.1.1 节方法得到. 实验中 Epoch 为 3，Top-k 为 1，Hidden Size 为 100，Layer Num 为 1，Word Embedding 维度为 100. SBERT 的训练效果，如图 9 所示，随着 Epoch 的增大，Cosine Pearson 和 Cosine Spearman 指标都有显著提升，当达到 3 个 Epoch 时，增加速度变慢.

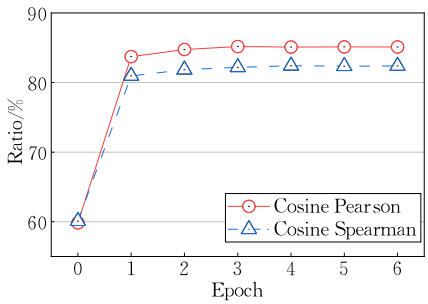


图 9　SBERT 的微调结果

语义检索实验结果如表 4，与其他检索方法相比，SRGen 语义检索方法取得了更好的效果. 文献[10]的 TF-IDF 算法获得了比 GLOVE 和 BERT 更好的效果. Ground Truth 的结果为 92.61%，可以证明在 3.1 节中，SRGen 得到的 Ground Truth 是准确的，同时可以证明响应相似的发现是正确的.

表 4　检索对比

方法	BLUE-4
GLOVE	36.10
BERT	36.63
TF-IDF	37.93
SRGen	41.60
Ground Truth	92.61

为了验证知识库中不同数据对检索结果的影响，本文还进行了消融实验. 其中，Epoch 设置为 3，+Name 表示增加 App 名，+Rating 表示增加评分信息，以此类推. 如表 5 所示，在检索中增加 Category 信息，Cosine Person 指标减小了 0.09，Cosine Spearman 增大了 0.16，可见类型信息不能提升检索效果. 同时 App 名、Rating、Response 的增加也都对检索结果有提升. 特别的，只有响应内容比只有评论内容的结果要好，如 Only Response 的 Cosine Pearson 为 70.32 而 Only Review 为 66.20，这表明语义检索中增加响应内容的必要性.

表 5　知识库中不同信息对检索的作用

方法	Cosine Pearson	Cosine Spearman
+Name+Rating +Category+Response	85.01	82.17
+Name+Rating +Response	85.10	82.01
+Name+Response	84.24	81.46
+Response	78.40	74.65
Only Response	70.32	69.27
Only Review	66.20	65.07

本文还进行了检索性能开销实验，如图 10 所示. 分别在编码阶段和 Top-k 相似度计算阶段进行了比较. 如图中折线所示，编码阶段，TF-IDF 比 SBERT 消耗的时间要少得多，SBERT 编码每条评论数据消耗 2.17 ms，TF-IDF 要 0.21 ms. 如图中柱状图所示，在 Top-k 相似度计算阶段，借助于 GPU 的并行运算，Batch Size 等于 100 时，每条评论检索耗时 8.19 ms，TF-IDF 耗时 432.64 ms. 综合看，本文方法的执行效率更高.

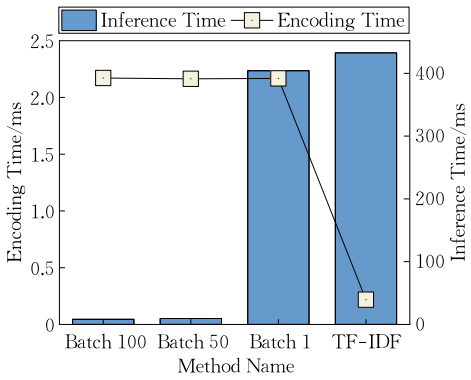


图 10　性能分析

4.3.4　数据分布影响分析(RQ4)

为了探究长尾分布对生成结果的影响，本文将数据分为三段：Head 头部数据、Median 中部数据及 Tail 尾部数据，各占 1/3. 进行了三组实验，第一组仅

使用 NMT 模型,第二组使用 CoRe 模型,第三组使用 SRGen 模型. 实验中,Top-k 设置为 1,Hidden Size 设置为 200,Layer Num 设置为 1,Word Embedding 维度设置为 100,Epoch 设置为 5,CoRe 和 SRGen 结合一条检索到的评论-响应对.

第一组实验如图 11 所示,发现头部数据在仅使用 NMT 的情况下,BLUE-4 即可达到 0.4,中部数据也可以达到 0.33,而尾部数据的 BLEU-4 仅在 0.1 左右. 随着 Epoch 的增加,头部数据的 BLUE-4 下降,中部和尾部的数据有明显的提升,总体上呈现先上升后下降的趋势,Epoch 等于 3 时 BLEU-4 到达峰值. 第二组实验如图 12 所示,发现 CoRe 算法随着 Epoch 的增加仅有很少的提升或没有提升,可见 CoRe 模型,没有对生数据成进行充分的学习,更倾向于直接复制检索到的数据. 第三组实验如图 13 所示,SRGen 不同于 CoRe,除头部的数据外,其他部分都随着 Epoch 的增加有明显的提升. 头部数据经过补充检索的信息能达到 0.5 的 BLEU-4,中部的数据在检索的加持下甚至能超过头部数据的表现,而尾部数据也有所提升达到 2.6,但依然距离头部和中部数据的表现差距很大. 综上,SRGen 能更好地利用检索的信息进行拷贝与生成,在不同分布的数据下有更好的表现.

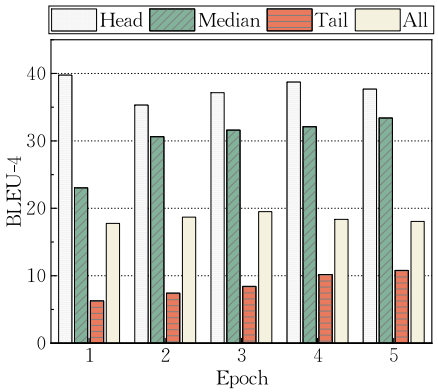


图 11 NMT 的长尾实验

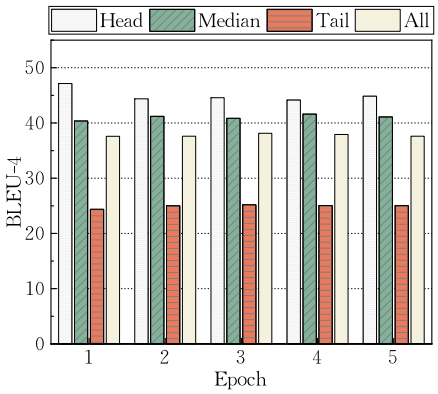


图 12 CoRe 的长尾实验

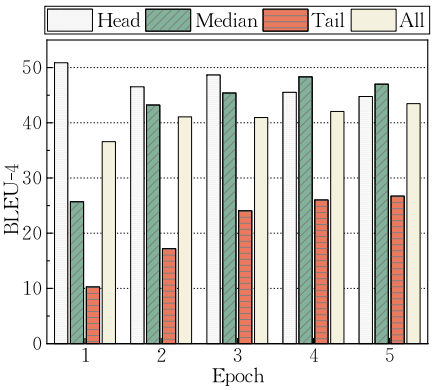


图 13 SRGen 的长尾实验

由此,可以得出以下结论,头部数据不用进行检索就能取得比较好的效果,通过检索的数据亦能提高其生成质量. 尾部数据由于训练样本不足,检索的数据可以来补充一定的生成概率,但依然改变不了效果差的局面. 因此,丰富训练样本和知识库,将是评论响应工作的重要内容.

4.3.5 人工验证分析(RQ5)

除了自动评价的方法,本文还设计了调查问卷,使用人工评价的方法对生成的响应进行评估. 人工评估通过问卷进行,共邀请了 8 名参与者,其中包括 2 名硕士研究生、2 名博士研究生和 4 名学士,均为计算机相关专业. 在参与者中,8 人均具有至少半年的软件开发行业经验. 邀请每位参与者阅读 30 条用户评论,并判断 SRGen、CoRe 生成的和应用程序开发人员人工的响应质量.

为了减轻人工检查的偏差,本文随机选择了头尾部各 30 条评论-响应对,并将它们平均分配到两份调查问卷中,每份问卷由头尾部各 15 条评论-响应对组成. 确保每份问卷由四个不同的参与者评价. 根据文献[10,24],对评论响应结果的评估从语法流畅度(Grammaticality Fluency)、相关性(Relevance)、准确性(Accuracy)和综合评分(Overall Score)四个方面考虑. 语法流畅度衡量文本是否易于理解,相关性度量用户评论和回复之间主题相关程度,准确性估计响应本身的准确度,综合评分是用户对评论响应的整体打分. 调查问卷样例,如图 14 所示.

表 6 和表 7 描述了头部和尾部抽样数据的人工评估结果. 可以看出,在所有三个响应中,开发人员的响应在所有指标方面都里获得了最高分. 在语法流畅度方面,SRGen 生成的响应和开发者的响应的平均分数相当接近,分别为 3.92 和 3.96、4.17 和 4.15. 在相关性方面,SRGen 生成的响应被评为比 CoRe 输出的响应略好. 在准确性指标方面,本文发

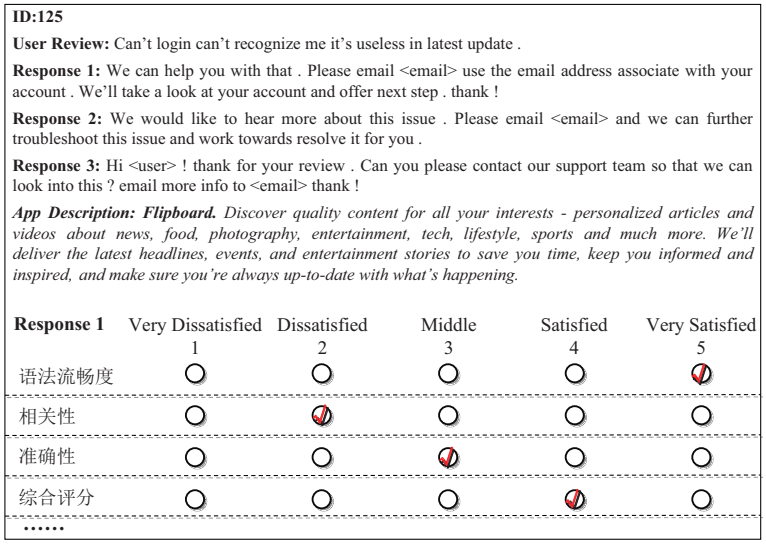


图 14　问卷调查样例

现 SRGen 输出的响应的平均分数与开发者的响应非常接近，在尾部数据中甚至更好一点. 在综合打分方面，SRGen 与开发人员的响应也非常相近. 头部数据中，参与者对 CoRe 生成的响应给出综合星级与 SRGen 相近. 在尾部对 SRGen 给出的综合评分更高. 这表明与 SRGen 可以用于开发者的实际评论响应中.

表 6　头部数据人工验证结果

模型	Grammaticality Fluency	Relevance	Accuracy	Overall Score
CoRe	3.86	4.21	4.26	3.97
SRGen	3.92	4.26	4.33	4.05
Developer	3.96	4.43	4.46	4.20

表 7　尾部数据人工验证结果

模型	Grammaticality Fluency	Relevance	Accuracy	Overall Score
CoRe	3.93	3.90	3.60	3.63
SRGen	4.17	3.92	3.90	3.96
Developer	4.15	3.96	3.81	3.93

图 15 和图 16 比较了 SRGen 和其他模型的响应样例对比. 可见 SRGen 可以更好地响应用户的情感，使用虚线标注，如“Previous is better”，SRGen 响应为“Thank for your feedback”；“Hate this app”，SRGen 响应为“Apologize”而 CoRe 响应为“Thanks a lot”. 相比之下，SRGen 的响应结果更符合用户表达的情感. 在评论-响应之间的知识关系方面，使用实线标注，用户回复为“you can disable our app”，RRGen 和 SRGen 都提到“you can disable it”，而 CoRe 回复地很通用，且与具体问题不相关. 可见，SRGen 生成的内容比 RRGen 和 CoRe 更接近开发者响应的内容. 这表明 SRGen 能够为生成提供更准确的知识，有助于在给定评论的情况下生成相关且准确的响应.

User Review: Previous ui is better. This is worst.
Response: Thank for your feedback. ES team is trying to develop a handier app for our user . can you tell me what do you most dislike? we will try to improve it.
RRGen: Hi pal. haha I know you like this simple flashlight! And could I ask for a <digit> star rat? You know your support is the best award for this little flashlight thank.
Core: Sorry for the confusion . We have fixed this in version <digit>. please try it and update review . The toast message shows how much time it cost to start es. We enable this in the debug version to try to reduce start up time and the reason why you see that be we mistakenly forget to disable it in the public release. Sorry again for inconvenience.
SRGen: Thank for your feedback. ES team is trying to develop a handy app for our user. can you tell me what do you think the dislike? we will try to improve it.

图 15　用户评论生成的响应样例 1

User Review: Hate this app but come with the phone!
Response: Hi there sorry for your poor experience. HTC has partnered with news republic to provide news content on their device so our app is preloaded on your phone and can't be deleted. You can disable our app in your device setting menu apps news republic disable. Hope this helps news republic.
RRGen: Hi <user> thank for your review. HTC has partnered with news republic to provide news content on their device so our app is preloaded on your phone and can't be deleted. You can disable it however in your device setting menu apps news republic disable. Hope this helps news republic.
Core: Hi there thanks a lot for your helpful feedback and support! Enjoy your read! News republic.
SRGen: Hi there thank for the review and we apologize you didn't enjoy the experience. HTC has partnered with news republic to provide news on their device so our app be preloaded on your phone and can't be deleted. You can disable it however in your device setting menu apps news republic disable. hope this helps!

图 16　用户评论生成的响应样例 2

4.4　有效性讨论

本文方法实施过程中，对有效性存在威胁的因素主要体现在如下三个方面.

第一,对于真值集的构造,SRGen 依赖于响应相似的发现,如果开发者的评论响应比较随意,不符合论文的发现时,则上述的方法的有效性会降低;第二,同样因为响应的相似性,例如用户明明给了 5 分评价,但开发人员没有仔细阅读用户的评论或使用通用的响应模板,导致响应内容还会请求用户给 5 分好评,这会导致响应内容和评分之间存在逻辑上的不一致;第三,没有进行异常样本的检测,实际应用中,模型上线后得到的数据无法完全控制,模型接收的数据可能是异常样本.此时检索模型查找到的评论-响应对的相似度较低,如果生成模型不能有效泛化样本,会导致效果不佳.除了异常样本外,还有一些评论在实际应用中不需要响应.

对于第一和第二方面来说,由于响应评论的开发者人数远小于评论用户人数,不像用户评论那样多样化,因此真值集构造问题风险较小.虽然可能有数据明明给了 5 分好评,开发者还是请求 5 分的响应,但这种数据占比并不多,而且神经网络学习具有泛化能力,所以这个问题影响较小.对于异常样本来说,实际应用中应该考虑进行异常样本检测,如果检测正常再进入 SRGen 模型中进行生成任务,否则不响应或人工响应并补充知识库数据.虽然 SRGen 提高了评论响应生成的性能,但 SRGen 不能判断是否要响应.可以参考文献[4],该工作总结了哪些评论特征会激发开发者的响应.本文更关注开发人员的后续行为,即评论响应的自动生成.

5 相关工作

端到端的评论响应与对话系统比较相似并且牵扯到信息检索的内容,因此本文从传统的对话系统、信息检索的方法和基于深度学习的对话系统来介绍相关工作.

5.1 传统的对话系统

对话理解部分主要包括三个子任务[25]:领域识别、意图识别和槽填充.基于规则的方法广泛用于对话管理建模[26-27],该方法将对话过程建模为有限状态之间的跳转过程,设计人员依据对话任务设计合理的对话状态及转移关系,这种模型结构简单、可控性强,但难以应对复杂的用户变化,用户体验不够好.槽填充是一种基于规则的方法,指从用户对话中抽取与任务相关的关键信息,它的性能对整个对话系统的质量有非常重要的影响.用序列标注模型是解决槽填充问题的主要方法,具体实现方法是利用序列标注模型为句子中的每个单元(字或词)打一个槽标签,根据打标的结果生成最后的槽填充结果.序列标注模型可以利用整个待标注序列的信息.很多序列标注模型都被用于槽填充问题,最通常使用的模型是条件随机场模型[28-29].随着深度神经网络在自然语言处理领域的应用成功,有学者将其用于槽填充任务,Yao 等人[30]采用循环神经网络模型在 ATIS 数据上得到了比条件随机场模型更好的结果,并通过加入词性特征和命名实体特征进一步提升了槽填充性能.

5.2 基于信息检索的方法

检索式的方法,一般从文本匹配和排序的角度看待,利用信息检索技术和机器学习排序技术来实现答案的检索和排序,具有返回流畅和信息丰富的响应的优点[31].给定一个问题 q,通过信息检索,得到信息检索文档集,并将信息检索得到的结果视为候选答案集$\{a_1,a_2,\cdots,a_s\}$.系统的目标是将候选集中的答案,按照相关度从高到低进行排序,相关度越高,位置越靠前.基于检索的方法依赖于信息检索或最近邻技术,选择响应的过程通常有以下两个标准,一是与所选答案相对应的历史输入应该与输入的对话相似;二是所选择的答案应该在语义上与输入对话的历史相关.文本相似度计算,如向量空间模型或 TF-IDF,以及 PageRank 重要性排名算法[32]或个性化技术,都可被组合到检索的排序中.基于检索的模型的优点为,相对于生成的模型,检索模型易实现,并且由于响应是从训练集中直接复制的,响应的语句在语法上总是正确的;可通过制定排序函数,从而使开发人员能够相对较好的控制对话系统中生成用户想看到的响应.其缺点为缺乏处理自然语言多样性的灵活性.也缺乏处理重要语言特征的能力,及辨别不同输入上下文之间细微的语义差异能力[33].自然语言的多样性使得用基于规则的方法来确定语义相似性度量变得困难,其中文本的词汇特征易于捕获,但文本的语义和句法属性较难捕获[12].各向异性问题是基于 BERT 的句子表示的关键瓶颈,阻碍了模型充分利用底层语义特征.Su 等人[13]发现传统机器学习中的 Whitening 操作同样可以增强句子表示的各向同性并取得有竞争力的结果.此外,Whitening 还能够降低句子表示的维度.

5.3 基于深度学习模型的方法

对于对话系统,深度学习可以使用大量数据来学习有意义的特征表示和响应生成策略.同时需要最少的手工制作[34],能有效规避流水线式任务型对

话系统误差传播等缺陷而成为研究热点.端到端的任务型对话系统主要是基于 Seq2Seq 架构,即用大量任务型对话语料训练对话编码器和解码器,由对话解码器自动生成响应.常见的编码器和解码器是循环神经网络(RNN),可以处理对话句子等序列数据.为了更好适应特定任务的对话,基于记忆网络(MemNN)[35]的方法由于其更好的长期记忆能力,可以有效存储领域知识,逐渐呈现出其优越性.Zhu 等人[36]提出一种检索增强的对抗性训练方法,该方法与现有方法不同,在对抗训练中利用编码器-解码器框架,同时利用基于检索的系统中的 N 条最佳响应候选来构建鉴别器.

在 App 自动评论响应任务中,Gao 等人[17]提出一个 Seq2seq 生成式的模型 RRGen,将 keywords 和 App 的属性融入其中,最终得到评论响应的序列.然而上述模型高度依赖于外部工具,例如 SURF 用于确定预定义关键字,SentiStrength 用于估计评论情感.外部工具的不准确估计会降低模型的准确性,削弱模型的灵活性和通用性.其次,RRGen 与基于 NMT 的方法有类似问题,即倾向于生成语料库中的高频词,并且生成的响应通常是通用的,提供一些没有用的信息.随后又提出 CoRe[15],通过输入 App 的描述信息、检索到相关评论的响应,将序列到序列的生成问题转换为摘要提取的工作.Li 等人[37]在生成代码摘要任务上,提出了基于检索和编辑的框架 EDITSUM,动机是检索到的原型为后期生成提供了一个很好的起点,因为相似代码片段的摘要通常具有相同的模式.

6　结束语

本文提出了一种面向 App 评论响应的语义检索和生成框架 SRGen,可为开发者创造更好的用户体验并提高应用程序的评级.针对使用字符级别的检索方法没有考虑句子的语义,且在生成响应时没有考虑用户评论与检索到评论差异,导致知识运用不充分,生成响应质量降低的问题,本文首先在评论数据集上构建包含正负样本的训练和测试集,并使用获得的训练和测试集对 Sentence-BERT(SBERT)进行微调训练.通过 SBERT 获得评论句子向量表示后,通过余弦距离获得 Top-k 条语义相近评论-响应对.最后利用检索到的评论与待生成响应评论的差异等信息,构建融合检索信息的生成模型,生成用户评论的响应.实验表明,在 BLEU 和 ROUGH 评测标准下 SRGen 生成的响应更加准确.

当前,App 评论响应的生成和检索过程,是两个独立的过程,未来我们希望构建生成与检索联合模型来训练提高生成质量.此外,还会对评论内容进行超出范围检测,判断是否存在产生评论响应所需的知识,从而不断完善知识库,提高响应质量.

参　考　文　献

[1] O'Dea S. Number of smartphone users from 2016 to 2021. https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/2021,8,6

[2] Ratings, reviews, and responses in app store. https://developer.apple.com/app-store/ratings-and-reviews/

[3] McIlroy S, Shang W, Ali N, et al. Is it worth responding to reviews? Studying the top free apps in Google play. IEEE Software, 2015, 34(3): 64-71

[4] Srisopha K, Link D, Swami D, et al. Learning features that predict developer responses for iOS App store reviews//Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). Bari, Italy, 2020: 1-11

[5] Xiao Jian-Mao, Chen Shi-Zhan, Feng Zhi-Yong, et al. An automatic analysis of user reviews method for APP evolution and maintenance. Chinese Journal of Computers, 2020, 43(11): 2184-2202(in Chinese)
(肖建茂,陈世展,冯志勇等.一种基于用户评论自动分析的 APP 维护和演化方法.计算机学报,2020,43(11):2184-2202)

[6] Martin W, Sarro F, Jia Y, et al. A survey of app store analysis for software engineering. IEEE Transactions on Software Engineering, 2016, 43(9): 817-847

[7] Gao C, Zeng J, Lyu M R, et al. Online app review analysis for identifying emerging issues//Proceedings of the 40th International Conference on Software Engineering. Gothenburg, Sweden, 2018: 48-58

[8] Hassan S, Tantithamthavorn C, Bezemer C P, et al. Studying the dialogue between users and developers of free apps in the Google play store. Empirical Software Engineering, 2018, 23(3): 1275-1312

[9] Xu H, Liu B, Shu L, et al. BERT post-training for review reading comprehension and aspect-based sentiment analysis//Proceedings of the NAACL-HLT. Minneapolis, USA, 2019: 2324-2335

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

[11] E Hai-Hong, Zhang Wen-Jing, Xiao Si-Qi, et al. Survey of entity relationship extraction based on deep learning. Journal

of Software, 2019, 30(6): 1793-1818(in Chinese)
(鄂海红, 张文静, 肖思琪等. 深度学习实体关系抽取研究综述. 软件学报, 2019, 30(6): 1793-1818)

[12] Chandrasekaran D, Mago V. Evolution of semantic similarity—A survey. ACM Computing Surveys(CSUR), 2021, 54(2): 1-37

[13] Su J, Cao J, Liu W, et al. Whitening sentence representations for better semantics and faster retrieval. arXiv preprint arXiv:2103.15316, 2021

[14] Hu X, Guo Y, Lu J, et al. Lighting up supervised learning in user review-based code localization: Dataset and benchmark //Proceedings of the 30th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. Singapore, 2022

[15] Gao C, Zhou W, Xia X, et al. Automating app review response generation based on contextual knowledge. ACM Transactions on Software Engineering and Methodology (TOSEM), 2021, 31(1): 1-3

[16] Farooq U, Siddique A B, Jamour F, et al. App-aware response synthesis for user reviews//Proceedings of the 2020 IEEE International Conference on Big Data (Big Data). Atlanta, USA, 2020: 699-708

[17] Gao C, Zeng J, Xia X, et al. Automating app review response generation//Proceedings of the 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE). San Diego, USA, 2019: 163-175

[18] Reimers N, Gurevych I, Reimers N, et al. Sentence-BERT: Sentence embeddings using siamese BERT-networks//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. Hong Kong, China, 2019: 3980-3990

[19] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks//Advances in Neural Information Processing Systems. Montreal, Canada, 2014: 3104-3112

[20] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need//Advances in Neural Information Processing Systems. Long Beach, USA, 2017: 5998-6008

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

[22] Papineni K, Roukos S, Ward T, et al. BLEU: A method for automatic evaluation of machine translation//Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. Philadelphia, USA, 2002: 311-318

[23] Lin C Y. Looking for a few good metrics: Automatic summarization evaluation-how many samples are enough?//Proceedings of the 4th Workshop on Research in Information Access Technologies Information Retrieval, Question Answering and Summarization(NTCIR-4). Tokyo, Japan, 2004

[24] Du X, Cardie C. Harvesting paragraph-level question-answer pairs from Wikipedia//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia, 2018: 1907-1917

[25] He Y, Young S. A data-driven spoken language understanding system//Proceedings of the 2003 IEEE Workshop on Automatic Speech Recognition and Understanding(IEEE Cat. No. 03EX721). St Thomas, USA, 2003: 583-588

[26] Walker M, Whittaker S. Mixed initiative in dialogue: An investigation into discourse segmentation. arXiv preprint cmp-lg/9504007, 1995

[27] Seneff S. Response planning and generation in the MERCURY flight reservation system. Computer Speech & Language, 2002, 16(3&4): 283-312

[28] Lafferty J D, Mccallum A K, Pereira F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data//Proceedings of the 18th International Conference on Machine Learning. Williamstown, USA, 2001: 282-289

[29] Wang Y Y, Acero A, Mahajan M, et al. Combining statistical and knowledge-based spoken language understanding in conditional models//Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions. Sydney, Australia. 2006: 882-889

[30] Yao K, Zweig G, Hwang M Y, et al. Recurrent neural networks for language understanding//Proceedings of the Interspeech. Lyon, France, 2013: 2524-2528

[31] Yang L, Qiu M, Qu C, et al. Response ranking with deep matching networks and external knowledge in information-seeking conversation systems//Proceedings of the 41st International ACM Sigir Conference on Research & Development in Information Retrieval. Ann Arbor, USA, 2018: 245-254

[32] Page L, Brin S, Motwani R, et al. The PageRank citation ranking: Bringing order to the web. Stanford InfoLab, SIDL-WP-1999-0120, 1999

[33] Li J. Teaching machines to converse. Stanford University, 2017. https://purl.stanford.edu/rm355rg0253

[34] Chen H, Liu X, Yin D, et al. A survey on dialogue systems: Recent advances and new frontiers. ACM SIGKDD Explorations Newsletter, 2017, 19(2): 25-35

[35] Bordes A, Boureau Y L, Weston J. Learning end-to-end goal-oriented dialog. arXiv preprint arXiv:1605.07683, 2016

[36] Zhu Q, Cui L, Zhang W, et al. Retrieval-enhanced adversarial training for neural response generation//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy, 2019: 3763-3773

[37] Li J, Li Y, Li G, et al. EDITSUM: A retrieve-and-edit framework for source code summary//Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering. Melbourne, Australia, 2021: 155-166

FAN Guo-Dong, Ph. D. candidate. His main research interests include service computing and cognitive services.

CHEN Shi-Zhan, Ph.D., associate professor. His research interests include service computing and service ecosystem mining and analysis.

XIAO Jian-Mao, Ph. D., lecturer. His main research interests include service computing, intelligent software engineering.

WU Hong-Yue, Ph. D., associate professor. His main research interests include service computing, and edge computing.

ZHANG Lu, Ph. D. candidate. Her main research interests include service computing and cognitive services.

XUE Xiao, Ph. D., professor. His research interests include service computing, swarm intelligence and computational experiment.

WANG Zhong-Jie, Ph. D., professor. His main research interests include service computing, and service engineering.

FENG Zhi-Yong, Ph. D., professor. His research interests include knowledge engineering, service computing, and computer cognition.

Background

The research work in this paper is about mobile service maintenance and evolution. The responses to mobile application reviews provide application developers with a user feedback mechanism, which has a positive impact on the rating of the application. In order to reduce the workload of responding to a large number of user reviews, developers usually adopt an automated mechanism to respond to user reviews, such as scoring praise or follow-up user questions and other types of responses. We have proposed some efficient automated user review mining and analysis methods, which are of great significance for developers to maintain the mobile application. This method can effectively help the developers to respond to the reviews submitted by the user in the app store, which can improve the rating of a mobile application.

In order to facilitate developers to better develop and maintain apps, some previous work mainly focused on automatic classification and summary of reviews. These researches study which aspects of the iOS App Store are related to responding to user reviews. Also, finding that ratings, comment length, as well as posting time, emotion, and writing style all have an important impact on comment responses. Hassan et al. began to analyze the dynamics of comment and response mechanisms. They study 4.5 million reviews and 126 686 responses from 2328 top free downloadable apps in the Google Play Store, a key finding is that reviews are non-static, and identified four patterns that guide developers' responses. In terms of generating responses to reviews, the previous method of responding to reviews is to use encoder-decoder-based methods to combine keywords or retrieved knowledge to generate responses. They only use character-level matching when retrieving, and do not consider the differences of reviews when generating. Therefore, this paper proposes a framework for semantic retrieval and generation.

We believe that our research methods can enhance the quality of the response no matter in terms of metrics or in terms of manual evaluation. At the practical application level, this method can help the developers to respond to the user reviews precisely, so as to continuously meet the developer's requirements for a higher rating. Furthermore, this paper aligns with the topics, Software Evolution and Cognitive Services, from two National Natural Science Key Foundation. What's more, our work can be verified in the elderly care service platform fytlun.com, the National Key R&D Program of China.

This work is supported by the National Natural Science Key Foundation of China under Grant Nos. 61832014 and 62032016, the National Natural Science Foundation of China under Grant Nos. 61972276 and 62102281, and the Science and Technology Research Project of Jiangxi Provincial Department of Education under Grant No. GJJ210338.