

基于融合模型与语义网络的 App 用户意图识别研究

陈瀚^{1,2}, 赵春蕾^{1,2*}, 蒋昊达^{1,2}, 王春东^{1,2}

(1. 天津理工大学教育部计算机视觉与系统省部共建重点实验室, 天津 300384;

2. 天津市智能计算与软件新技术重点实验室, 天津 300384)

摘要: 随着手机应用程序的流行, 应用市场上出现了大量非结构化的中文用户评论。基于用户评论识别 App 用户意图, 可以帮助开发人员对 App 软件进行有针对性的维护和改善。为了从中准确识别用户意图, 提出一种基于融合模型和语义网络的 App 用户意图识别方法 FSAUIR。使用百度工具 Senta 判断评论的情感倾向, 构建基于 RoBERTa 的融合意图分类模型 RBMS, 通过 RoBERTa 模型将用户评论转化为语义特征表示, 并将其输入到双向门控循环单元中, 以提取评论的全局上下文语义信息, 同时利用多头自注意力机制和 SoftPool 获取关键的特征信息, 保留主要特征, 通过 Softmax 进行归一化处理, 得到意图分类结果。在意图分类的基础上, 引入 PositionRank 模型提取各意图类别下评论的关键词, 计算关键词之间的共现关系, 构建关键词语义网络, 从而更细粒度地识别用户意图。实验结果表明, 相比 BERT、RoBERTa、RoBERTa-CNN 等模型, RBMS 模型在人工标注数据集上具有较优的分类性能, 准确率、精确率、召回率、F1 值分别为 87.75%、88.09%、87.80%、87.88%。此外, 在意图分类的结果集中, FSAUIR 构建的语义网络可以高效地挖掘出用户评论中有价值的信息。

关键词: 意图识别; 意图分类; RoBERTa 模型; 双向循环门控单元; PositionRank 模型; 多头自注意力机制

中图分类号: TP391.1

文献标志码: A

DOI: 10.19678/j.issn.1000-3428.0068206

Research on App User Intent Recognition Based on Fusion Model and Semantic Network

CHEN Han^{1,2}, ZHAO Chunlei^{1,2*}, JIANG Haoda^{1,2}, WANG Chundong^{1,2}

(1. Key Laboratory of Computer Vision and System of Ministry of Education, Tianjin University of Technology, Tianjin 300384, China;

2. Tianjin Key Laboratory of Intelligent Computing and Novel Software Technology, Tianjin 300384, China)

【Abstract】 With the popularity of mobile Applications (Apps), a large number of unstructured Chinese user reviews have appeared in the application market. Identifying App user intent based on these reviews helps developers make targeted maintenance and improvement of App software. To accurately recognize user intent, this study proposes an App user intent recognition method based on fusion model and semantic network, named FSAUIR. First, FSAUIR uses the Baidu tool Senta to determine the emotional tendency of the reviews. It then introduces Robustly optimized Bidirectional Encoder Representation from Transformers approach (RoBERTa)-based fusion intent classification model, RoBERTa-BiGRU-Multiple Self-Attention+SoftPool (RBMS), which transforms user reviews into semantic feature representations through the RoBERTa model. These representations are input into a Bidirectional Gated Recurrent Unit (BiGRU) to extract the global contextual semantic information of the reviews. Simultaneously, the multiple self-attention and SoftPool mechanisms obtain more critical feature information, retaining the main features. Finally, the Softmax normalizes the features to obtain the intent classification results. Subsequently, FSAUIR employs the PositionRank model to extract keywords from reviews under each intent category, calculate the co-occurrence relationship between keywords, and construct a keywords semantic network to recognize user intent with finer granularity. Experimental results show that compared to BERT, RoBERTa, RoBERTa-CNN, and other models, the RBMS model exhibits superior classification performance on the manually labeled dataset. The model achieves accuracy, precision, recall, and F1 value of 87.75%, 88.09%, 87.80%, and 87.88%, respectively. Additionally, the semantic network constructed by FSAUIR efficiently mines valuable information from user reviews in the intent classification result set.

【Key words】 intent recognition; intent classification; RoBERTa model; Bidirectional Gated Recurrent Unit (BiGRU); PositionRank model; multihead self-attention mechanism

收稿日期: 2023-08-10 修回日期: 2023-11-08

基金项目: 国家重点研发计划“科技助力经济 2020”重点专项 (SQ2020YFF0413781, SQ2020YFF0401503)。

通信作者 E-mail: * zcltjut@126.com

0 引言

移动智能终端的应用程序分发平台允许用户以评论的形式提供其对所下载 App(Application 的简称,指移动智能终端的第三方应用软件)的反馈。用户评论包含许多重要信息,例如 App 软件相关的错误、新功能的请求、已有功能的改进建议等^[1-3]。对于开发人员而言,用户评论可以指导其维护和改善 App 软件。然而,在短时间内从中文用户评论中提取有价值的信息存在众多挑战。首先,应用商店每天都会收到大量的评论,人工难以对庞大数量的评论进行高效分析,例如 App Store 的微信 App 仅在 2022 年就收到了 48 657 条评论。其次,用户评论是用非结构化、嘈杂和模棱两可的自然语言编写的^[4],具有文本篇幅较短、不遵循相应语法规则等特点,且由于中文用户评论的特殊性,还需要采用不同的语言规则和工作流程来处理中文文本^[5]。最后,用户评论的有用信息比例偏低,即每条用户评论中仅包含少量错误报告、功能请求、功能评估以及和用户体验相关的信息,导致没有足够的信息来识别用户意图。因此,对中文用户评论进行自动化分析,识别 App 用户细粒度意图,从而对软件的维护和改善做出指导,具有重要的研究意义与实用价值。

App 评论的自动化分析主要以预定义的主题(错误报告、功能请求等)对评论进行分类^[5-10]。文献[6]使用自然语言处理(NLP)结合情感分析的技术将评论分为信息提供、信息搜索、特征请求等类别。但是到目前为止,App 评论分析的相关研究主要集中在英文 App 评论,只有文献[1]提出的 RASL 方法和文献[5]提出的 KEFE 方法结合文本分类、关键词提取等方法对中文 App 用户评论进行分析。但上述 2 个方法均无法满足日益增长的软件工程需求,原因如下:1)RASL 及 KEFE 使用的支持向量机(SVM)和 BERT 对文本的语义结构和上下文理解浅薄,存在无法灵活捕获特征等问题;2)RASL 及 KEFE 方法无法从更细粒度识别用户意图,帮助开发人员对 App 软件进行有针对性的维护和改善。针对以上问题,本文提出一种基于融合模型和语义网络的 App 用户意图识别方法 FSAUIR,对中文 App 评论进行分析。首先使用百度工具 Senta 判断情感倾向,并提出 RBMS 融合意图分类模型自动对用户评论进行意图分类;在意图分类的基础上,引入 PositionRank 模型提取评论关键词,搭建关键词语义网络,从而指导开发人员维护

和改善 App 软件。本文的主要工作如下:

1)提出基于 RoBERTa^[11]的融合意图分类模型 RBMS。该模型首先使用 RoBERTa 模型进行预训练,有效捕捉用户评论中的语义特征;其次通过双向门控循环单元(BiGRU)对评论的上下文特征进行提取;然后引入多头自注意力机制和 SoftPool,优化用户意图分类任务中关键特征的提取机制,进一步提高用户评论意图分类的准确性。

2)引入基于 PositionRank 模型的语义网络,从更细粒度识别用户意图,全方位挖掘用户真实意图,弥补 RASL 等方法单一性的不足。

3)选取 App Store 几款流行 App 的原生中文评论进行实验,与现有的意图分类模型进行对比,验证 RBMS 融合模型的性能和 FSAUIR 的有效性。

1 相关工作

近年来,App 评论信息挖掘相关研究主要以机器学习和深度学习算法的自动化分类为主。文献[12]提出 AR-Miner(App Review Miner)框架,采用无监督的朴素贝叶斯(NB)模型将应用评论分为有用和无用信息 2 类,通过优先排序,将最具信息量的应用评论提供给开发人员。文献[13]同样采用逻辑回归、NB 等机器学习算法将评论分为性能、可靠性、可用性、安全性和可维护性等类别,对软件的非功能需求进行分析。文献[14]使用机器学习算法(NB、 k 近邻)手动生成了一种分类方法,对 Uber 应用的评论进行分类。文献[15]提出了一种基于框架语义的 App 评论分类方法,采用框架语义将评论变换为框架元素,同时利用 NB 和 SVM 对变换后的 App 评论进行分类。随着硬件性能的提升,大规模预训练模型被广泛应用于 NLP 的中下游任务^[16-18]。在 App 评论分类领域,文献[19]提出一种基于预训练模型 BERT 的多语言 App 评论分类方法,旨在对不同语言的 App 评论进行分类。文献[20]提出了 Review-BERT 模型,利用预训练语言模型提取包含上下文语义信息的词嵌入向量来表征 App 评论。此外,文献[21]探究了预训练模型在 App 评论文本分类任务中的性能,涵盖 XLNET、BERT、ALBERT 和 RoBERTa 这 4 种预训练模型。上述工作能够在一定程度上减少人工分析 App 评论的工作量,但无法识别用户细粒度意图,因此难以指导开发人员维护和改善 App 软件。

为分析用户的真实意图,App 评论信息挖掘的研究引入了意图挖掘的概念,并成为当前 App 评论挖掘研究的主流。文献[22]根据主题和用户意图将

评论分为功能请求、意见查询、问题发现、解决方案建议、信息搜索和信息提供等 6 种意图。文献[23]同样也采用 NLP、文本挖掘、情感倾向判断等相关技术将评论分为信息提供、信息搜索、功能请求、解决方案、问题发现等意图。文献[24]结合 NLP、情感倾向判断等技术对评论进行分类,将 App 评论分为错误报告、功能请求、用户体验、评级等 4 类。文献[25]利用文献[24]提供的分类信息,使用卷积神经网络(CNN)对 App 评论进行自动分类。文献[6]在文献[22]的基础上,提出 App 评论自动分析方法 ARICA,该方法结合 BERT 模型、SentiStrength 情感分析方法以及 DBSCAN 聚类算法,将评论分为信息提供、信息搜索、特征请求、问题发现、其他等 5 种意图进行分析,获得了较好的意图挖掘效果。

综上所述,当前 App 评论信息挖掘相关研究主要通过对 App 评论进行分类并辅以情感分析、主题提取、关键词提取等进行分析。然而,无论是基于预

定义类别分类还是基于用户意图分类对评论进行分析,均存在分类模型无法针对性提取用户评论的意图特征、无法全方位挖掘用户关注点等问题。本文在上述工作的基础上,提出 RBMS 融合意图分类模型对中文 App 评论进行分类,提高评论意图分类的准确率。同时,采用语义网络可视化的方式,识别用户细粒度意图,指导开发人员维护和改善 App 软件。

2 FSAUIR 整体框架

本文提出一种基于融合模型和语义网络的 App 用户意图识别方法 FSAUIR,该方法通过对各种模型创新性融合来提高意图分类的准确性,同时引入 PositionRank 模型提取关键词,构建语义网络,以便从更细粒度识别用户意图。FSAUIR 框架结构如图 1 所示(彩色效果见《计算机工程》官网 HTML 版,下同)。

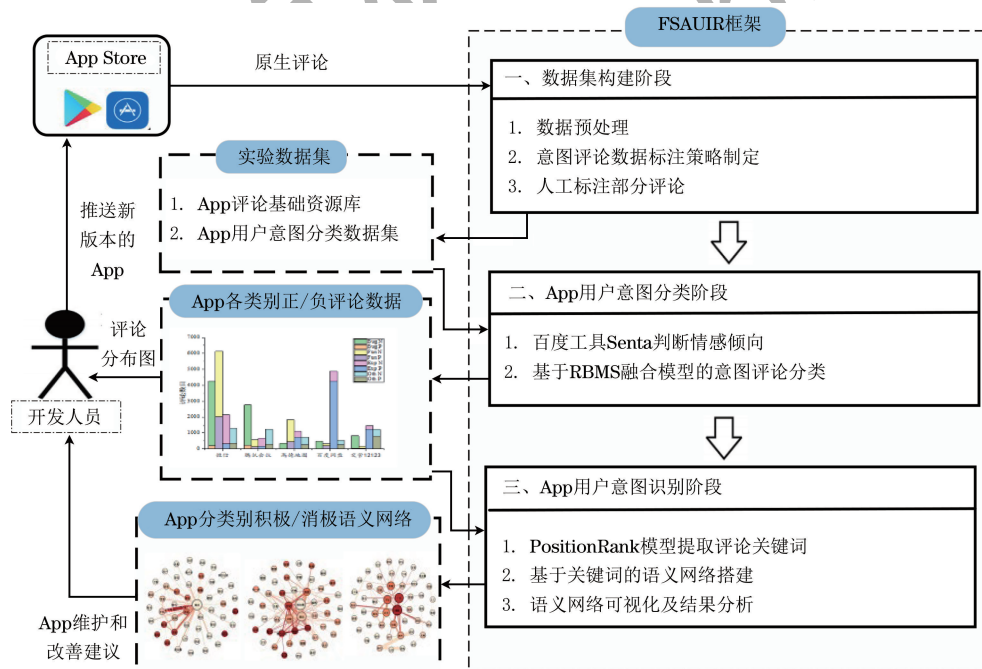


图 1 FSAUIR 整体框架

Fig.1 FSAUIR overall framework

FSAUIR 通过数据集构建、App 用户意图分类、App 用户意图识别 3 个阶段来识别用户细粒度意图。

1)数据集构建阶段:从 App Store 爬取流行 App 的原生中文评论,进行数据预处理、App 评论意图类别划分及数据标注相关工作,构建去噪的评论基础资源库以及用于模型训练的用户意图分类实验数据集。

2)用户意图分类阶段:使用百度工具 Senta 判断评论的情感倾向,提出 RBMS 融合意图分类模

型,对 App 用户评论进行分类。

3)App 用户意图识别阶段:基于 PositionRank 模型提取各意图类别下的评论关键词,构建关键词语义网络识别用户细粒度意图。

2.1 数据集构建阶段

2.1.1 数据预处理

App 用户通过手机应用软件提交的中文评论存在许多干扰数据,如邮箱账号、emoji 表情符号、非法字符等,此类数据会对用户意图分类与识别的结果造成影响,所以需要获取的用户评论数

据进行预处理,构建去噪的 App 评论基础资源库。本文采用 NLP 相关技术对用户评论数据进行预处理,采用正则表达式过滤网络链接、电话号码等噪声信息,同时删除非中文词汇、emoji 表情符号、非法字符等。此外,由于一部分用户发表了大量与软件开发和维护无关的评论,如“垃圾”“突突突”等,这些评论通常由 2~3 个汉字组成,无法为开发人员提供有效建议,同时会对用户意图分类和识别造成不良的影响。因此,本文方法在预处理的最后对长度小于 4 的评论进行了过滤,并对评

论进行了去重。

2.1.2 评论意图类别划分与数据标注

意图类别划分的合理性直接决定意图分类和识别的准确性和意图识别的结果,因此,本文基于文献[6-7, 24-25]的意图分类方式,给出适用于 App Store 中文评论的意图类别划分。如表 1 所示,本文将 App 用户意图分为错误报告、功能相关(功能请求、功能评估)、用户体验、其他等 4 类。这些意图不仅能够呈现用户对 App 的真正需求,还便于开发人员分析。

表 1 用户评论意图类别划分

Table 1 User reviews intend category classification

类别	描述	数据样例
错误报告	报告异常行为或问题的句子	朋友圈发不了视频是怎么回事
功能请求	描述用户对新功能的请求以及对现有功能的改进的句子	弄个已读功能吧拜托了
功能评估	表达用户对现有功能的评价的句子	折叠功能浏览朋友圈完全无感
用户体验	用户对 App 整体体验的句子	广告多成这样非常影响使用体验
其他	不属于上述任何类别的句子	微信里没有大帅哥和我聊天

2.2 App 用户意图分类阶段

2.2.1 Senta 情感倾向判断

用户在发表评论时,会将其对 App 软件的喜欢或讨厌的情感倾向表达出来,包括积极、消极和中性情感。评论的情感倾向对于开发人员通过用户意图判断 App 的优势和劣势至关重要。因此,对评论进行情感倾向判断是必要的。然而,在之前 App 评论分析的相关研究中,并无适用于中文 App 评论的情感分析工具。为减少人力分析成本,本文使用百度工具 Senta 对 App 评论进行情感倾向判断。Senta 采用深度学习训练,可以学习深层次的语义特征,具备强泛化能力,在相对长的句子上仍能保持高准确率。Senta 工具使用示例如图 2 所示,其中, text 为 App 评论, sentiment 为情感极性分类结果(0 为积极,1 为中立,2 为消极), confidence 为分类的置信度, positive_prob 表示 App 评论为积极的概率, negative_prob 表示 App 评论为消极的概率。

```

“text”: “点赞了别人朋友圈后面有人再点赞评论给我提醒个没必要的玩意能不能去掉烦死了”
“items”: [
  {
    “sentiment”: 2,
    “confidence”: 0.703 628,
    “positive_prob”: 0.133 367,
    “negative_prob”: 0.866 633,
  }
]

```

图 2 Senta 情感分析使用示例

Fig.2 Sentiment analysis usage example of Senta

2.2.2 RBMS 融合意图分类模型

针对用户意图分类中用户意图提取困难等问题,本文提出了 RBMS 融合意图分类模型。首先引入大规模预训练语言模型 RoBERTa 感知用户评论的语义特征;其次通过 BiGRU 捕获句子间的依赖关系;最后创新性地结合多头自注意力机制与 SoftPool 提取文本中深层次的关键意图特征。RBMS 模型包括输入层、RoBERTa 层、BiGRU 层、SoftPool 层、多头自注意力机制层、全连接层和输出层,其结构如图 3 所示。各层的具体过程下文将分别描述。

1) 输入层与文本表示层。

为解决 Word2Vec^[26] (Word 2 Vector) 和 GloVe^[27] (Global Vectors for Word Representation) 词向量模型在处理中文 App 评论时存在的一词多义等问题,本文引入 RoBERTa 模型进行文本表示,通过建立词嵌入向量之间的双向关联,使模型学习到各个词嵌入向量在不同语句环境下的语义。

在模型的输入层,首先将分类值映射为整数值,使用二进制向量表示每一个整数值;然后通过位置编码保存各个词在整个评论中的相对或绝对位置,学习各个词之间的位置关系;最后将词向量与位置编码相加生成词嵌入向量作为 RoBERTa 层的输入,表示为 $E = (E_1^d, E_2^d, \dots, E_n^d)$, E 的维度大小为 $n \times d$ 。输入层的结构如图 4 所示,图中 [CLS] 代表输入评论的起始位置, [SEP] 代表输入评论的结束位置。

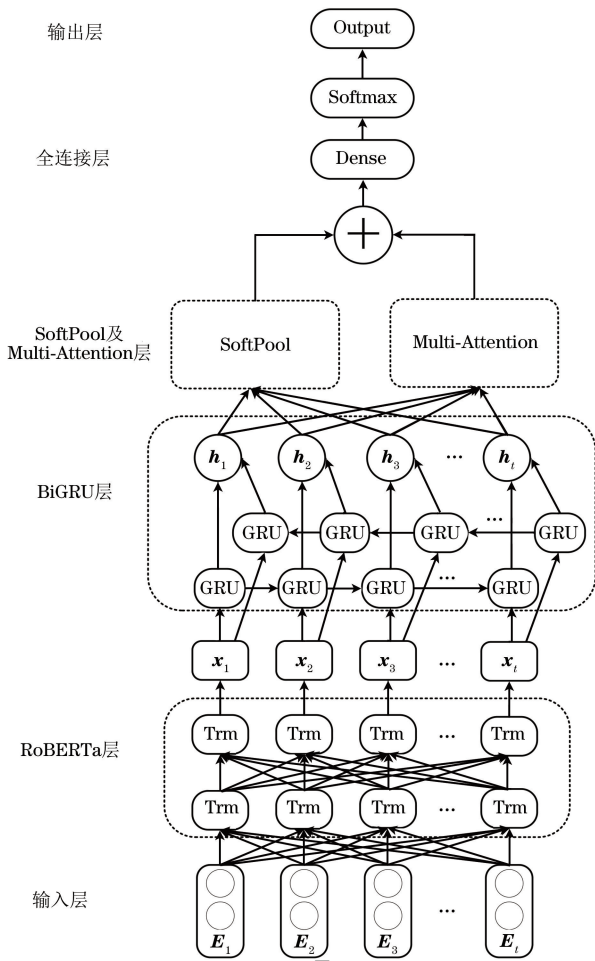


图 3 RBMS 融合意图分类模型

Fig.3 RBMS fusion intent classification model

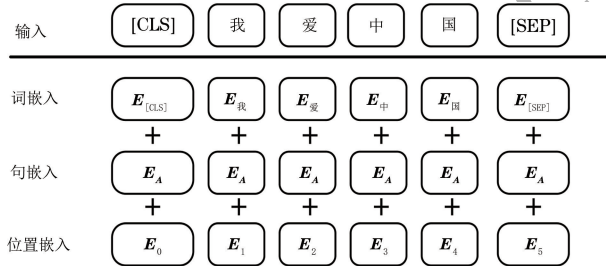


图 4 输入层逻辑结构

Fig.4 Input layer logic structure

在 RoBERTa 模型的文本表示层,使用 12 层的 Transformer 双向编码器作为用户评论的意图特征提取器。RoBERTa 模型结构如图 5 所示。

RoBERTa 采用动态 masking 机制对词嵌入向量进行预训练,采用 2 个预训练任务来训练词嵌入向量 E 。在首个任务中, RoBERTa 将词嵌入向量复制 10 份,并从每份中随机选取 15% 的 tokens 进行动态 mask,通过预测被遮盖标记的上下文来学习不同的掩码策略,同时对语义表征进行捕捉;在第 2 个任务中, RoBERTa 通过 full-sentences 训练方式,多方位学习来自输入层的词嵌入向量,充分提取语

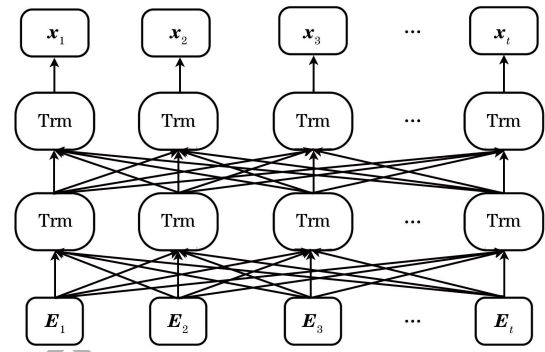


图 5 RoBERTa 模型结构

Fig.5 RoBERTa model structure

义信息,同时将当前输出输入到前馈神经网络层,提高收敛速度。词向量嵌入 E 经过 RoBERTa 层编码后得到该层的输出序列为 $X = [x_1, x_2, \dots, x_t]$ 。

2) BiGRU 层。

长短时记忆(LSTM)网络和 GRU 均为循环神经网络(RNN)的改进模型,解决了 RNN 模型存在的梯度弥散等缺陷。但 LSTM 网络训练时间长,容易过拟合,因此,本文选择 GRU 捕获评论的上下文特征。GRU 对 LSTM 网络进行了优化,模型结构如图 6 所示。

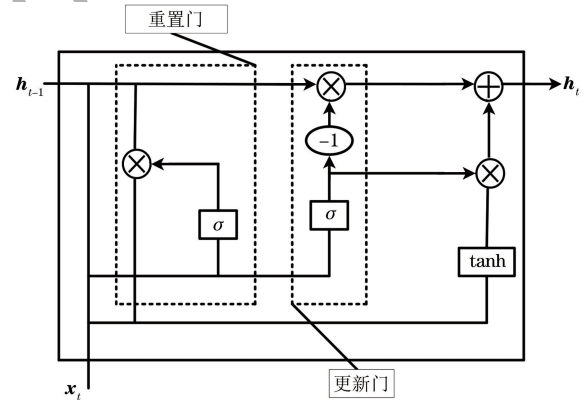


图 6 GRU 神经网络基本单元

Fig.6 Basic unit of GRU neural network

如图 6 所示,更新门用于计算上一时刻隐藏层状态对当前层的影响,更新门的值与上一时刻隐藏层状态对当前层的影响大小成正比,其计算方式如式(1)所示:

$$r_t = \sigma(W_r x_t + W_r h_{t-1} + b_r) \quad (1)$$

重置门用于剔除上一时刻隐藏层状态中的不相关信息,重置门的值越小,剔除的不相关信息就越多。重置门的计算公式如式(2)所示:

$$z_t = \sigma(W_z x_t + W_z h_{t-1} + b_z) \quad (2)$$

当前状态的计算方式如式(3)和式(4)所示:

$$\tilde{h}_t = \tanh(W_h x_t + W_h (r_t \times h_{t-1}) + b_h) \quad (3)$$

$$h_t = z_t \times \tilde{h}_t + h_{t-1} (1 - z_t) \quad (4)$$

式中： h_t 、 h_{t-1} 分别表示当前时刻和上一时刻的隐藏层状态； \tilde{h}_t 为候选激活状态； x_t 为当前时刻的输入； r_t 与 z_t 分别是更新门和重置门的计算结果； W 是权重矩阵； b 为偏置向量。

然而单向的 GRU 只提取了评论中上文对下文的影响，无法表征下文对上文的影响。为充分提取评论的上下文关系，本文采用 BiGRU 网络，其模型结构如图 7 所示。

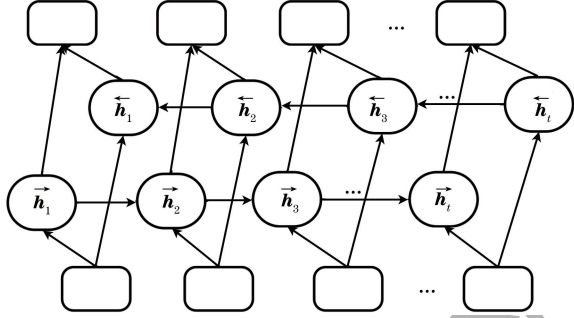


图 7 BiGRU 结构

Fig. 7 BiGRU structure

如图 7 所示，BiGRU 每步输出由当前状态的正向状态和反向状态组成，更新公式如式(5)所示：

$$h_t = \vec{W}_t \vec{h}_t + \overleftarrow{W}_t \overleftarrow{h}_t + b_t \quad (5)$$

式中： \vec{W}_t 与 \overleftarrow{W}_t 分别表示正向传播和反向传播的权重矩阵； b_t 为偏置向量。最终，经综合输入矩阵 X ，获得 BiGRU 层的输出序列为 $H = [h_1, h_2, \dots, h_t]$ 。

3) 多头自注意力层。

为过滤用户评论中存在的冗余信息和噪声，本文采用多头自注意力机制^[28]对 BiGRU 层输出序列 H 的不同位置进行加权关注，以各个时刻的注意力权重表征不同位置对分类结果的贡献程度，避免了自注意力机制过度地将注意力集中于自身的问题，从而提取出评论的关键意图特征。

首先，通过线性变换将 BiGRU 层的输出 H 变换为查询向量矩阵 Q 、键向量矩阵 K 、值向量矩阵 V ，变换过程如式(6)所示：

$$\begin{cases} Q = W_Q H \\ K = W_K H \\ V = W_V H \end{cases} \quad (6)$$

式中： W_Q 、 W_K 、 W_V 为对应的转换矩阵。

接着将 Q 、 K 、 V 投影到 n 个子空间中，投影过程如式(7)所示：

$$\begin{cases} [Q_1, Q_2, \dots, Q_n] = [QW_{Q,1}, QW_{Q,2}, \dots, QW_{Q,n}] \\ [K_1, K_2, \dots, K_n] = [KW_{K,1}, KW_{K,2}, \dots, KW_{K,n}] \\ [V_1, V_2, \dots, V_n] = [VW_{V,1}, VW_{V,2}, \dots, VW_{V,n}] \end{cases} \quad (7)$$

式中： $W_{Q,i}$ 、 $W_{K,i}$ 、 $W_{V,i}$ 分别为 Q 、 K 、 V 的第 i 个转换矩阵。

随后通过缩放点积对 Q_i 和 K_i 进行内积运算，并使用 Softmax 函数归一化后与 V_i 相乘得到单头的注意力值，并行地计算 n 个子空间的注意力。计算方式如式(8)所示：

$$o_{\text{head},i} = \text{Softmax}\left(\frac{Q_i \cdot K_i^T}{\sqrt{d}}\right) V_i \quad (8)$$

式中： d 为缩放因子，用于将 Q 、 K 的内积缩放成标准的正态分布。

最后，合并所有子空间的注意力值，得到结果 A 。计算方式如式(9)所示：

$$A(Q, K, V) = \text{Concat}(o_{\text{head},1}, o_{\text{head},2}, \dots, o_{\text{head},n}) W_0 \quad (9)$$

式中： W_0 为转换矩阵。

4) SoftPool 层。

本文模型引入 SoftPool 层来提取通过 BiGRU 层学习到的词向量 H 中的综合信息。SoftPool 的核心部分在于使用 Softmax 指数加权的方式对区域 R 中各个权重的特征值非线性激活，权重较高的特征值具有更大的激活值，在整个区域中占主导地位，保证特征的表达性且操作是可微的^[29]。SoftPool 层计算过程如式(10)~式(11)所示：

$$w_i = \frac{e^{a_i}}{\sum_{j \in R} e^{a_j}} \quad (10)$$

$$\tilde{a} = \sum_{i \in R} w_i \times a_i \quad (11)$$

首先，计算出每一个激活区域 a_i 的自然指数，将其除以所有激活区域的自然指数之和，得到每个激活区域的比值 w_i ，提取用户评论中关键的语义特征；其次，区域 R 内的每个特征值在反向传播的时候都会有一个最小梯度，在取得不一样的权重 w_i 后，对加权区域 R 内的特征值进行加权，就能够得到输出 \tilde{a} 。

与最大池化计算区域内的最大值和平均池化计算区域内的平均值不同，SoftPool 采用指数加权的方式，最大限度地保留了评论的主要特征，这不仅可以在下采样阶段保留更多的评论语义信息，同时也能够增强评论中重要的语义特征，避免最大值池化和平均池化容易丢失细节特征的问题。最终 BiGRU 输出序列经 SoftPool 后得到输出序列 S 。

5) 全连接层与输出层。

本文将 SoftPool 层的输出序列 S 和多头自注意力层的输出序列 A 进行拼接，作为全连接层的输入。全连接层对 A 与 S 调整后得到 M ，计算方式如

式(12)所示:

$$\mathbf{M} = f(\mathbf{S} \oplus \mathbf{A}) \quad (12)$$

输出层将全连接层的输出信息 \mathbf{M} 进行 Softmax 归一化处理,得到 App 用户评论在各类别的概率分布矩阵,选择具有最高概率的结果作为用户评论的类别。计算方式如式(13)所示:

$$\hat{y} = \text{Softmax}(\mathbf{W}_d \mathbf{M} + \mathbf{b}_d) \quad (13)$$

式中: \mathbf{W}_d 为权重矩阵; \mathbf{b}_d 为偏移量。

由于模型中的参数是连续的,因此采用反向传播机制训练和更新,同时使用交叉熵损失函数优化,避免在梯度下降计算的时候出现梯度弥散,弥补 Sigmoid 型函数的导数形式易发生饱和的缺陷。计算方式如式(14)所示:

$$L_{\text{Loss}} = - \sum_{i=1}^m y_i \ln \hat{y}_i \quad (14)$$

式中: y_i 为真实的概率分布; \hat{y}_i 为模型预测的每个意图类别对应的概率; m 为用户意图类别总数。

2.3 App 用户意图识别阶段

为解决先前相关研究(关键词提取、主题提取等)识别用户细粒度意图效率低的问题,本文提出使用语义网络更细粒度地识别用户意图。通过 PositionRank 模型提取评论关键词、关键词共现三元组后,利用 Gephi 构建语义网络,可视化用户意图。

2.3.1 App 评论关键词提取

关键词是 App 用户评论中具有代表性、可以反映主题思想、发挥关键作用的词语。在无监督学习的相关研究中,关键词提取被描述为一个排序问题,基于图的排序算法被认为是最佳的求解算法,比较经典的有 PageRank^[30] 算法等。本文使用 PositionRank^[31] 模型提取 App 各意图类别评论的关键词,文献[31]已通过大量实验证明该算法效果优于 PageRank 等算法。PositionRank 提取关键词的过程如图 8 所示。

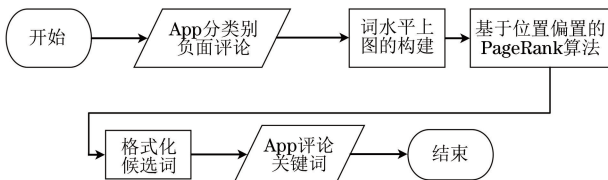


图 8 PositionRank 关键词提取过程

Fig. 8 PositionRank keywords extraction process

首先,为评论构建一个词图 $G = (V, E)$,评论中保留的词均在图 G 中作为 1 个节点且出现 1 次。如果在评论中这些节点同时出现在 1 个窗口中,那么 v_i 、 v_j 这 2 个节点通过 1 条边 $(v_i, v_j) \in E$ 连

接,边的权重由 2 个词的共现次数决定。

其次,使用基于位置偏置的 PageRank 算法计算各个词的重要性得分,计算方式如式(15)所示:

$$S_{\text{score}}(v_i) = (1 - \alpha) \times \tilde{p}_i + \alpha \times \sum_{v_j \in e_{\text{adj}}(v_i)} \frac{w_{ji}}{O(v_j)} S_{\text{score}}(v_j) \quad (15)$$

式中: S_{score} 代表每个节点的得分; α 表示阻尼因子; e_{adj} 表示形容词词性; w_{ji} 表示图中边的权重; $O(v_j)$ 表示词所有出向边的权重和; \tilde{p}_i 表示词的初始得分归一化的值(向量 $\tilde{\mathbf{p}}$ 中的节点 v_i 的表示)。 $\tilde{\mathbf{p}}$ 计算公式如式(16)所示:

$$\tilde{\mathbf{p}} = \left[\begin{array}{c} \frac{p_1}{p_1 + p_2 + \dots + p_{|V|}} \\ \frac{p_2}{p_1 + p_2 + \dots + p_{|V|}} \\ \dots \\ \frac{p_{|V|}}{p_1 + p_2 + \dots + p_{|V|}} \end{array} \right] \quad (16)$$

最后,将评论中具有连续位置的候选词连接成短语,并以正则表达式来匹配短语,对组合后的短语进行单个关键词求和再排名,根据重要性得分提取评论中的关键词。

2.3.2 语义网络的构建

语义网络 $G = \{V, E, W(E)\}$ 是用来呈现 App 评论关键词及其共现关系的数据结构图模型,能够准确地提炼用户各方面细粒度意图。其中: $V = \{V_i \mid i = 1, 2, \dots, n\}$ 表示所有评论关键词; $E = \{e_{ij} \mid i, j = 1, 2, \dots, n\}$ 表示关键词节点间共现关系所构成的边; $W(E)$ 表示 e_{ij} 的权重(关键词节点和节点的共现次数)。本文对 PositionRank 模型提取的关键词及其共现关系采用 Gephi 构建语义网络,全方位展示用户意图。

3 实验设计

本文采用 NLP 等技术对 App 评论进行意图分类、情感倾向判断以及用户意图识别,目的在于为开发人员提供用户评论中有价值的信息,指导开发人员维护和改善 App 软件。本文的实验设计主要解决以下 3 个问题:

RQ1: 预训练模型相较于传统的深度学习模型在评论意图分类任务中是否具有优势?

RQ2: 模型融合是否能够提高评论意图分类的准确率、F1 值,本文所提出的融合模型是否具有优势?

RQ3: FSAUIR 能否通过语义网络识别用户细粒度意图,帮助开发人员对 App 软件进行有针对性的维护和改善?

3.1 数据准备

本文基于 App Store 的原生用户评论来评估 FSAUIR 方法。在 App 软件选择方面,本文遵循以下 3 个原则:1)App 是 App Store 中国市场上流行的应用软件,开发人员会定期进行更新;2)所选择的 App 软件须涵盖不同类别;3)App 软件中应包含大量的用户评论,能够验证用户意图识别的效果。基于上述原则,在一定程度上可以保证 FSAUIR 方法的泛化性。最后,本文选择了 5 个 App 软件,具体情况如表 2 所示。表 2 列出了 App 软件的名称、所属类别、评论数量以及对应的 App 版本。总体来看,本文在 5 个 App 下获得了 38 898 条评论。

表 2 原生评论数据集

Table 2 Native reviews dataset

App 名称	类别	评论数/条	App 版本
微信	社交	15 158	8
腾讯会议	办公	6 287	21
高德地图	导航	4 629	15
百度网盘	工具	8 266	12
交管 12123	生活	4 558	10

3.2 RQ1、RQ2 分析方法

为回答 RQ1 和 RQ2,本文对数据集进行预处理后,以人工标注的方式创建意图分类实验数据集。在相关参数设置下,使用文本分类任务中常用的评价指标,衡量意图分类模型优劣。

3.2.1 意图分类实验数据集的构建

为了满足深度学习模型训练、验证和测试的要求,并且降低人力成本,本文从各个 App 中采用分层随机抽样策略选取 3 293 条评论(占预处理后总评论的 10%),根据设定的意图类别进行人工标注,标注结果如表 3 所示。本文以 8:1:1 的比例将实验数据集划分为训练集、验证集、测试集进行实验。其中,功能相关和用户体验占比分别为 30.00%和 29.09%,错误报告和用户体验占比分别为 24.77%和 16.12%。由于其他类别的评论都是一些与 App 本身无关的评论,因此本文方法在意图分类之后,对于其他类别的评论将不再进行关键词提取与语义网络搭建。

表 3 用户评论意图分类实验数据集

Table 3 Experimental dataset for intent classification of user reviews

类别	评论数量/条	所占比例/%
错误报告	816	24.77
功能相关	988	30.00
用户体验	958	29.09
其他	531	16.12

3.2.2 实验环境及参数设置

实验环境配置如表 4 所示。

表 4 实验环境

Table 4 Experimental environment

实验环境	配置
操作系统	Windows 10
显卡	NVIDIA GeForce GTX1070
内存/GB	16
编程语言	Python 3.8
深度学习框架	PyTorch 1.12.1

本文的实验参数是在多次对比实验的基础上,根据实验结果对初始参数进行调优确定的,并且以候选参数中的最佳参数进行其余所有实验。实验参数选取设置如表 5 所示。

表 5 实验参数选取设置

Table 5 Experimental parameter selection and setting

可选参数	待取值	取值
学习率	$1 \times 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}$	1×10^{-5}
BiGRU 隐藏层维度	64,128,256	256
批大小	32,64,128	32
丢弃率	0.1,0.3,0.5	0.1
注意力机制头数	4,6,8	4
最大文本长度	128,150,256	256
CNN 卷积核大小	(2,3,4),(3,4,5)	(2,3,4)
模型优化器	RMSProp, Adam	Adam

除上述参数外,通过预训练模型提取到的词向量维度为 768,训练迭代参数(Epoch)设置为 10(Epoch 为 10 时已经充分收敛)。

3.2.3 评价指标

本文使用准确率、精确率、召回率以及 F1 值作为实验的评价指标。

准确率是分类正确的样本占样本总数的比例,计算公式如下:

$$A = \frac{N_{TP} + N_{TN}}{N_{TP} + N_{TN} + N_{FP} + N_{FN}} \times 100\% \quad (17)$$

精确率是分类正确的正样本占所有预测为正例的样本数的比例,计算公式如下:

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}} \times 100\% \quad (18)$$

召回率是分类正确的样本数占所有真实为正例的样本数的比例,计算公式如下:

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}} \times 100\% \quad (19)$$

式中: N_{TP} 表示预测为正样本且分类正确的样本数; N_{FP} 表示实际为负而分类错误的样本数; N_{FN} 表示实际为正而分类错误的样本数。

由于精确率和召回率是相互对立的,因此本文在实验结果对比中采用两者的调和平均 F1 值配合准确率进行评估,计算公式如下:

$$F_1 = \frac{2 \times P \times R}{P + R} \times 100\% \quad (20)$$

3.3 RQ3 分析方法

为回答 RQ3,本文对来自 App Store 的原生中文评论数据进行预处理、意图分类以及关键词提取后,使用 Gephi 生成语义网络,更细粒度地识别用户意图。同时,对识别出的用户意图进行分析,判断 FSAUIR 能否帮助开发人员对 App 软件进行有针对性的维护和改善。

4 实验结果分析

4.1 针对 RQ1 的实验结果分析

高效率的意图分类是开发人员准确识别 App 用户意图的基础。为了探究预训练模型在用户意图分类任务中的高效性,本文设计了多种分类模型进行比较,包括 4 种使用传统 Word2Vec 模型作为词嵌入层的经典文本分类模型与 4 种预训练模型。对比模型如下:

1)TextCNN:由 3 个不同大小的卷积核构成,能够提取评论的局部关键特征。

2)BiGRU:由正反向 2 个 GRU 构成,能够捕获评论的上下文信息。

3)BiGRU-ATT:在 BiGRU 模型的基础上,使用注意力机制对 BiGRU 各时刻的输出进行加权,提取关键特征。

4)深层金字塔网络(DPCNN):基于词级别的网络,能够提取长距离依赖关系。

5)BERT:Google 提出的预训练模型,采用多层 Transformer 架构,在大规模语料库上进行无监督预训练,通过学习用户评论的上下文词向量进行意图分类。

6)ALBERT:BERT 的改进模型,减少了模型参数,提升了训练速度,同时移除了 Next Sentence Prediction 任务,采用句子连贯性预测(SOP)。

7)ERNIE:BERT 的改进模型,将 mask 字改为 mask 词,同时采用更多的中文语料进行预训练。

8)RoBERTa:BERT 的改进模型,采用动态 masking 机制,拥有更大的 mini-batch,使用了更多的训练数据。

模型 1)~4)采用 Word2Vec 模型作为词嵌入层。各分类模型结果如表 6 所示,其中,加粗数据表示最优值,下同。

表 6 RQ1 实验结果对比

模型	准确率	精确率	召回率	F1 值
TextCNN	76.60	77.01	76.47	75.96
BiGRU	76.29	76.59	75.13	75.63
BiGRU-ATT	78.12	77.06	77.28	77.60
DPCNN	77.51	77.80	76.94	76.85
BERT	82.98	82.34	81.75	81.89
ALBERT	80.85	81.29	79.92	80.09
ERNIE	83.44	83.94	81.83	82.24
RoBERTa	84.06	84.47	83.45	83.70

从表 6 中结果可以看出,预训练模型中效果最差的 ALBERT 模型比传统深度学习模型中效果最佳的 BiGRU-ATT 准确率和 F1 值分别提高了 2.73 和 2.49 个百分点,说明预训练模型相较于传统深度学习模型更具优势,这是因为传统深度学习在采用 Word2Vec 获取 App 评论的词向量表示时,丢弃了大量的位置信息和 App 评论的语义表示,所以分类准确率和 F1 值偏低。在传统深度学习模型中:BiGRU 模型准确率、F1 值均最低,这是因为 App 评论中短句偏多,造成 BiGRU 模型难以提取到句子的转折关系,最后在分类器累加输出导致分类错误;而 TextCNN 模型能够捕获短句中的局部特征,所以准确率和 F1 值比 BiGRU 模型稍高;DPCNN 模型使用等长卷积、近连接,能够提取更多的意图特征,所以效果比 TextCNN 模型更佳;BiGRU-ATT 模型在 BiGRU 模型上增加了注意力机制,弥补了 BiGRU 无法获取局部特征的缺陷,在传统深度学习模型中取得最佳效果。在预训练模型中:由于 ALBERT 模型对 BERT 模型的参数进行了削减,因此准确率、F1 值比 BERT 较低;ERNIE 升级了 BERT 的 mask 的方式,将 mask 分为字符、字体、短语 3 个级别,效果进一步提升;RoBERTa 对 BERT 进行了改进,采用动态 masking 机制,取得预训练模型中的最佳效果,准确率和 F1 值分别达到 84.06%和 83.70%。

4.2 针对 RQ2 的实验结果分析

本文设计了以下 5 种融合模型与 RQ1 中取得最佳效果的基线模型 RoBERTa 进行比较。

1)RB-CNN:在 RoBERTa 模型的基础上,通过添加 CNN 更深层次地提取评论中的局部关键特征。

2)RB-BiGRU:在 RoBERTa 模型的基础上,通过添加 BiGRU 进一步提取评论的全局特征。

3)RB-CNN-BiGRU:在 RoBERTa 模型的基础

上,先通过 CNN 获取局部重要特征,再通过 BiGRU 获取评论全局特征。

4) RB-BiGRU-ATT: 在 RB-BiGRU 模型的基础上,增加多头自注意力机制提取关键意图特征。

5) RBMS: 本文模型,在 RB-BiGRU-ATT 模型的基础上,增加 SoftPool 进一步提取关键信息。

各分类模型实验结果如表 7 所示。

表 7 RQ2 实验结果对比

模型	准确率	精确率	召回率	F1 值
RoBERTa	84.06	84.47	83.45	83.70
RB-CNN	85.35	85.12	84.16	84.30
RB-BiGRU	84.63	85.15	83.86	84.19
RB-CNN-BiGRU	85.48	85.53	85.46	85.43
RB-BiGRU-ATT	85.76	85.92	85.77	85.78
RBMS(本文模型)	87.75	88.09	87.80	87.88

表 7 结果表明:相比于 RoBERTa 原生模型,以 RoBERTa 为词嵌入,然后接入神经网络的 RB-CNN 和 RB-BiGRU 模型有效地优化了输入文本的向量表达,提升了训练效果;CNN 在 App 短评论中可以提取到更多的特征,故在经过 RoBERTa 词向量优化之后,RB-CNN 的准确率和 F1 值会高于 RB-BiGRU;由于 BiGRU 网络会重复提取文本特征,丢失主要特征,因此本文使用了 RB-CNN-BiGRU、RB-BiGRU-ATT、RBMS(本文模型)这 3 种模型来弥补 BiGRU 网络的缺陷,准确率分别提高 0.85、1.13、3.12 个百分点,F1 值分别提高 1.24、1.59、3.69 个百分点;RB-CNN-BiGRU 模型结合了 CNN 提取局部特征和 BiGRU 提取上下文语义的优势,但 CNN 卷积池化时会导致某些特征信息的丢失,对后续 BiGRU 提取上下文语义造成影响,故其准确率和 F1 值提高得最少;RBMS 模型(本文模型)在 BiGRU 网络之后,使用 SoftPool 和多头自注意力机制相结合的方式,能够获取评论更多的意图特征,故其准确率、F1 值提高得最多,相比于 RoBERTa 模型,RBMS 模型在准确率和 F1 值上分别提升 4.69 和 4.08 个百分点,取得最佳效果,证明 RBMS 模型能够有效地应用于用户意图分类任务。

同时,为了探究 RBMS 模型中多头自注意力机制与 SoftPool 的作用,本文对 RBMS 模型进行了消融实验,对 RB-BiGRU、RB-BiGRU-ATT 以及去除 RBMS 模型中多头自注意力机制的 RB-BiGRU-SoftPool 进行对比。各模型实验结果如表 8 所示。

表 8 消融实验结果

模型	准确率	精确率	召回率	F1 值
RB-BiGRU	84.63	85.15	83.86	84.19
RB-BiGRU-ATT	85.76	85.92	85.77	85.78
RB-BiGRU-SoftPool	85.20	85.52	85.10	85.19
RBMS(本文模型)	87.75	88.09	87.80	87.88

表 8 结果表明:RB-BiGRU-ATT 和 RB-BiGRU-SoftPool 模型相较于 RB-BiGRU,准确率分别提高 1.13 和 0.57 个百分点,F1 值分别提高 1.59 和 1.00 个百分点,说明多头自注意力机制和 SoftPool 能够有效提取文本的关键特征,实现全局特征和局部特征的互补,从而全面地提取意图评论特征信息;RBMS 模型相较于 RB-BiGRU-ATT 和 RB-BiGRU-SoftPool 模型准确率分别提高 1.99 和 2.55 个百分点,F1 值分别提高 2.10 和 2.69 个百分点,说明 RBMS 融合模型中多头自注意力机制与 SoftPool 相结合的网络能够实现特征间关系的学习,使得特征充分交互,进一步提高分类效果。

4.3 针对 RQ3 的实验结果分析

识别用户意图的目的是为了帮助开发人员分析用户反馈,因此,本文通过对 3.1 节中的大量原生评论进行意图分类、情感倾向判断以及关键词提取,构建语义网络来验证 FSAUIR 的有效性。

1) 情感倾向判断及意图分类。

对 3.1 节中的原生评论数据进行去噪后,先通过百度工具 Senta 对评论进行情感倾向判断,再利用训练好的 RBMS 融合模型对评论进行意图分类,得到的结果如图 9 所示。其中:Bug P、Fun P、Exp P、Oth P 分别代表错误报告、功能相关、用户体验、其他的积极评论;Bug N、Fun N、Exp N、Oth N 分别代表错误报告、功能相关、用户体验、其他的消极评论。由于中立评论极少,且同样包含 App 的问题,因此本文将中立评论归于消极评论。实验结果表明,不同 App 用户的关注点是不同的:微信和高德地图关注点在功能相关;百度网盘和交管 12123 关注点在用户体验;而腾讯会议关注点则在错误报告。此外,各 App 中不同类别的积极和消极评论的占比存在巨大差距。

2) 关键词提取与共词网络搭建。

对于意图分类后的 App 正/负评论,本文过滤了与 App 维护和改善无关的其他类别的评论,利用 PositionRank 模型计算每个词语的关键性得分,经统计排序确定每个类别 App 评论的关键词。为搭建内容明晰、规模适中的语义网络,本文仅提取各意图类

5 结束语

本文提出一种基于融合模型和语义网络的 App 用户意图识别方法 FSAUIR,旨在通过识别用户细粒度意图,指导开发人员维护和改善 App。在意图分类阶段,提出 RBMS 融合意图分类模型,利用预训练模型 RoBERTa 捕获意图评论中的语义特征,随后通过 BiGRU 学习评论的正反向语义信息,捕获句子间的依赖关系,以此对意图评论上下文进行深层次的特征提取,最后使用多头自注意力机制与 SoftPool 相结合的方式提取评论关键特征,突出评论中隐含的重要意图信息。一系列的对比实验和消融实验,验证了 RBMS 模型相对于其他模型在识别准确率和 F1 值上的优势。在意图识别阶段,利用 PositionRank 模型计算关键词重要性得分,并以关键词为节点、以关键词共现次数为边构建语义网络,可视化用户细粒度意图。本文以微信 App 消极评论为例,通过实证分析表明 FSAUIR 方法识别出的用户意图可以帮助开发人员对 App 软件进行有针对性的维护和改善。

由于 App 评论中仍会存在评论注水、评论中无效信息略多等问题,下一步工作将考虑对 App 评论中的注水评论进行识别,剔除评论中的无效信息。同时,重点考虑开发人员的回复,从开发人员的角度出发,为 App 的维护和改善提供更具价值的参考。

参考文献

- [1] 陈琪,张莉,蒋竞,等.一种基于支持向量机和主题模型的评论分析方法[J].软件学报,2019,30(5):1547-1560.
CHEN Q, ZHANG L, JIANG J, et al. Review analysis method based on support vector machine and latent Dirichlet allocation[J]. Journal of Software, 2019, 30(5): 1547-1560. (in Chinese)
- [2] MALGAONKAR S, LICORISH S A, SAVARIMUTHU B T R. Prioritizing user concerns in App reviews—a study of requests for new features, enhancements and bug fixes [J]. Information and Software Technology, 2022, 144: 106798.
- [3] de LIMA V M A, de ARAÚJO A F, RICARDO M M. Temporal dynamics of requirements engineering from mobile App reviews[J]. PeerJ Computer Science, 2022, 8: e874.
- [4] JIANG H, ZHANG J X, LI X C, et al. Recommending new features from mobile App descriptions [J]. ACM Transactions on Software Engineering and Methodology, 28(4): 22.
- [5] WU H Y, DENG W J, NIU X T, et al. Identifying key features from App user reviews[C]//Proceedings of the 43rd International Conference on Software Engineering. Washington D. C., USA: IEEE Press, 2021: 922-932.
- [6] 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[J].计算机学报,2020,43(11):2184-2202.
XIAO J M, CHEN S Z, FENG Z Y, et al. An automatic analysis of user reviews method for APP evolution and maintenance[J]. Chinese Journal of Computers, 2020, 43(11): 2184-2202. (in Chinese)
- [7] ALI S F, SIRTS K, PFAHL D. Using App reviews for competitive analysis: tool support[C]//Proceedings of the 3rd ACM SIGSOFT International Workshop on App Market Analytics. New York, USA: ACM Press, 2019: 40-46.
- [8] GUO H, SINGH M P. Caspar: extracting and synthesizing user stories of problems from App reviews[C]//Proceedings of the 42nd International Conference on Software Engineering. New York, USA: ACM Press, 2020: 628-640.
- [9] MALGAONKAR S, LICORISH S A, SAVARIMUTHU B T R. Automatically generating taxonomy for grouping App reviews—a study of three Apps [J]. Software Quality Journal, 2022, 30(2): 483-512.
- [10] CIURUMELEA A, PANICHELLA S, GALL H C. Automated user reviews analyser [C]//Proceedings of the 40th International Conference on Software Engineering. New York, USA: ACM Press, 2018: 317-318.
- [11] LIU Y, OTT M, GOYAL N, et al. RoBERTa: a robustly optimized BERT pretraining approach[EB/OL]. [2023-07-10]. <https://arxiv.org/abs/1907.11692>.
- [12] CHEN N, LIN J L, HOLS C H, et al. AR-miner: mining informative reviews for developers from mobile App marketplace [C] // Proceedings of the 36th International Conference on Software Engineering. New York, USA: ACM Press, 2014: 767-778.
- [13] 贾一获,刘璘.中文非功能需求描述的识别与分类方法研究[J].软件学报,2019,30(10):3115-3126.
JIA Y D, LIU L. Recognition and classification of non-functional requirements in Chinese[J]. Journal of Software, 2019, 30(10): 3115-3126. (in Chinese)
- [14] SHARMA M, AGGARWAL D, PAHUJA D. Categorization and classification of Uber reviews [M] // SHARMA H, GOVINDAN K, POONIA R C, et al. Advances in computing and intelligent systems. Singapore: Springer Singapore, 2020: 347-355.
- [15] JHA N, MAHMOUD A. Mining user requirements from application store reviews using frame semantics [C] // Proceedings of International Working Conference on Requirements Engineering: Foundation for Software Quality. Berlin, Germany: Springer, 2017: 273-287.
- [16] DABROWSKI J, LETIER E, PERINI A, et al. Analysing App reviews for software engineering: a systematic literature review[J]. Empirical Software Engineering, 2022, 27(2): 43.
- [17] 王曙燕,原柯.基于RoBERTa-WWM的大学生论坛情感分析模型[J].计算机工程,2022,48(8):292-298,305.
WANG S Y, YUAN K. Sentiment analysis model of college student forum based on RoBERTa-WWM [J]. Computer Engineering, 2022, 48(8): 292-298, 305. (in Chinese)
- [18] 李军怀,陈苗苗,王怀军,等.基于ALBERT-BGRU-CRF的中文命名实体识别方法[J].计算机工程,2022,48(6):89-94,106.
LI J H, CHEN M M, WANG H J, et al. Chinese named entity recognition method based on ALBERT-BGRU-CRF [J]. Computer Engineering, 2022, 48(6): 89-94, 106. (in Chinese)
- [19] HENAO P R, FISCHBACH J, SPIES D, et al. Transfer learning for mining feature requests and bug reports from Tweets and App store reviews[C]//Proceedings of the 29th International Requirements Engineering Conference. Washington D. C., USA: IEEE Press, 2021: 80-86.
- [20] de ARAÚJO A F, MARCACINI R M. RE-BERT: automatic extraction of software requirements from App reviews using BERT language model[C]//Proceedings of the 36th Annual ACM Symposium on Applied Computing. New York, USA: ACM Press, 2021: 1321-1327.

- [21] HADIM A, FARD F H. Evaluating pre-trained models for user feedback analysis in software engineering: a study on classification of APP-reviews [J]. *Empirical Software Engineering*, 2023, 28(4): 88.
- [22] di SORBO A, PANICHELLA S, VISAGGIO C A, et al. Development emails content analyzer: intention mining in developer discussions (T) [C]//*Proceedings of the 30th International Conference on Automated Software Engineering*. Washington D. C., USA: IEEE Press, 2015: 12-23.
- [23] PANICHELLA S, di SORBO A, GUZMAN E, et al. How can I improve my APP? Classifying user reviews for software maintenance and evolution[C]//*Proceedings of 2015 IEEE International Conference on Software Maintenance and Evolution*. Washington D. C., USA: IEEE Press, 2015: 281-290.
- [24] MAALEJ W, KURTANOVIC Z, NABIL H, et al. On the automatic classification of App reviews[J]. *Requirements Engineering*, 2016, 21(3): 311-331.
- [25] NAILA A, YOUSUF R W, XIA K W, et al. Convolutional neural network based classification of App reviews[J]. *IEEE Access*, 2020, 8: 185619-185628.
- [26] MIKOI O T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[EB/OL]. [2023-07-10]. <https://arxiv.org/pdf/1301.3781v1>.
- [27] PENNINGTON J, SOCHER R, MANNING C D. GloVe: global vectors for word representation[C]//*Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Doha, Qatar: [s. n.], 2014: 1532-1543.
- [28] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//*Proceedings of the 31st International Conference on Neural Information Processing Systems*. New York, USA: ACM Press, 2017: 6000-6010.
- [29] STERGIOU A, POPPE R, KALLIATAKIS G. Refining activation downsampling with SoftPool[C]//*Proceedings of the IEEE/CVF International Conference on Computer Vision*. Washington D. C., USA: IEEE Press, 2021: 10357-10366.
- [30] BRINS, PAGE L. The anatomy of a large-scale hypertextual web search engine [J]. *Computer Networks and ISDN Systems*, 1998, 30(1/2/3/4/5/6/7): 107-117.
- [31] FLORESCU C, CARAGEA C. PositionRank: an unsupervised approach to keyphrase extraction from scholarly documents[C]//*Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Stroudsburg, USA: Association for Computational Linguistics, 2017: 1105-1115.

编辑 金胡考