项目编号：

密 级：

JustChat详细设计

第23组

易嘉祯 201922080909

刘守池 201922080634

王思颖 201952080635

闵齐星 201922080637

牟泽培 201922081202

2019年12月13日

**变更记录**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 变更  编号 | 版本号 | 日期 | 章节/段落/行  或图/表号 | 变更  状态 | 变更简单描述 | 审核人 | 批准人 |
| 1 | v1 | 2019/11/29 |  | A |  |  |  |
| 2 | v2 | 2019/12/5 |  | M |  |  |  |
| 3 | V3 | 2019/12/13 |  | M |  |  |  |
| 4 |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |
| 17 |  |  |  |  |  |  |  |
| 18 |  |  |  |  |  |  |  |

变更状态：**A** – 增加 **M** – 修改 **D** – 删除

**目 录**

[1绪言及目标 1](#_Toc27149136)

[2范围 1](#_Toc27149137)

[3定义及缩写 1](#_Toc27149138)

[4引用 1](#_Toc27149139)

[5角色及职责 1](#_Toc27149140)

[6工具及环境 2](#_Toc27149141)

[6.1研发语言及编译器 2](#_Toc27149142)

[6.2软件支持工具 2](#_Toc27149143)

[6.3第三方工具 2](#_Toc27149144)

[7高层分析 2](#_Toc27149145)

[7.1数据模式视图 2](#_Toc27149146)

[7.2数据表定义 3](#_Toc27149147)

[8详细设计 3](#_Toc27149148)

[8.1接口设计 3](#_Toc27149149)

[8.1.1通讯模块 3](#_Toc27149150)

[8.1.2日志模块 5](#_Toc27149151)

[8.1.3读写json模块 6](#_Toc27149152)

[9情感分析功能设计 6](#_Toc27149153)

[9.1介绍BERT 6](#_Toc27149154)

[9.2预训练BERT 8](#_Toc27149155)

[9.3微调BERT 8](#_Toc27149156)

[9.4基于BERT的情感分析 8](#_Toc27149157)

# 1绪言及目标

项目名称：JustChat局域网P2P聊天软件

项目提出单位：

项目开发者：电子科技大学数据库新技术23组

项目使用单位：

软件的详细设计阶段是给出软件模块的内部过程描述即是模块内部的算法设计。

本文档为“JustChat局域网P2P聊天软件详细设计说明书”，主要用于为实现系统的功能而进行的系统详细设计说明，详细描述了系统各软件组成模块的实现流程、功能、接口、编译、测试要点等内容，便于对系统的编码进行指导和约束。

本文档供项目组全体成员及项目组领导，单元测试人员阅读。

# 2范围

本文档用于软件设计阶段的详细设计，该详细设计的范围是：系统各组成部分的构成、每个软件模块（或类）的属性、方法、事件，各软件模块的外部接口说明等内容。

# 3定义及缩写

|  |  |
| --- | --- |
| **缩写** | **定义** |
| BERT | 基于Transformer的双向编码表达 |

# 4引用

|  |  |
| --- | --- |
| **文档名** | **文档标题** |
| C++ GUI Qt 4编程 | Qt基础教程 |
| NAACL | BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding |

# 5角色及职责

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **姓名** | **角色** | **职责** |
| 1 | 易嘉祯 | 程序员，UI设计员 | 负责代码整合，界面设计，整体软件把握，编写程序代码。 |
| 2 | 牟泽培 | 程序员 | 负责通讯模块，实现点对点、组播聊天。 |
| 3 | 刘守池 | 程序员，测试员 | 负责日志模块，利用单例模式创建一个单线程写日志。 |
| 4 | 王思颖 | 程序员，测试员 | 负责编写json文件的读写接口。 |
| 5 | 闵齐星 | 程序员 | 负责情感分析模块的实现。 |

# 6工具及环境

# 6.1研发语言及编译器

C++, Qt5.8, Python, Tensorflow, 客户端，某一局域网

# 6.2软件支持工具

开发使用Visual Studio2015等软件工具。

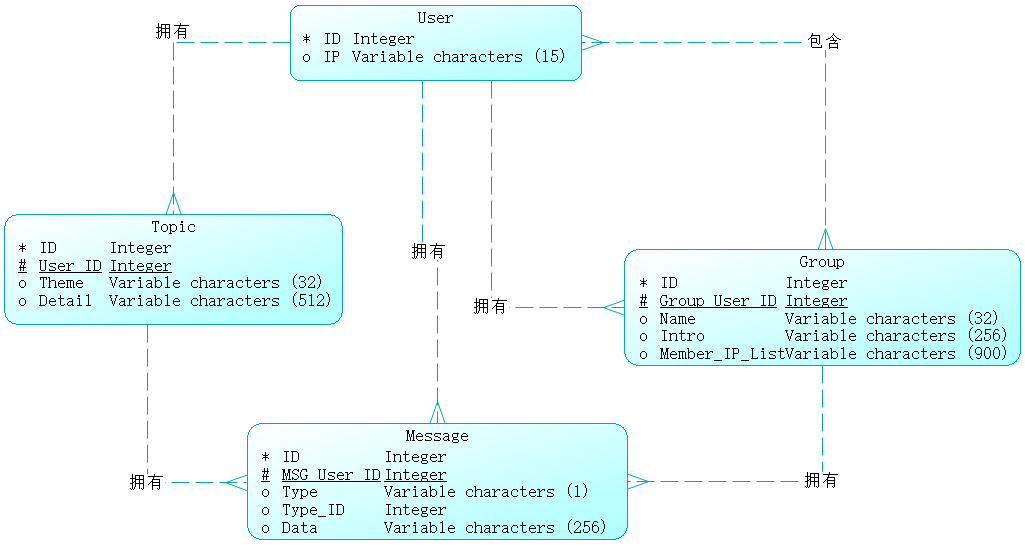
# 6.3第三方工具

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 程序包、工具 | 版本 | 用途 |
| 1 | Tensorflow | 1.11.0 | 训练和测试BERT模型 |
| 2 | Qt | 5.8.0 | 软件实现 |
| 3 | Qtdesigner | 4 | 软件界面设计 |

# 7高层分析

# 7.1数据模式视图

系统数据视图如下：



# 7.2数据表定义

|  |  |
| --- | --- |
| **表名** | **描述** |
| User | 用户信息表，用于在某一局域网内唯一确定每个用户。 |
| Group | 组聊信息表，通过组聊创建者ID和组聊ID唯一确定组聊，保存组聊名称、简介信息，并记录组员的IP。 |
| Topic | 话题信息表，通过话题创建者ID和话题ID唯一确定话题，保存话题名称和详情信息，使用所有在线用户的IP。 |
| Message | 聊天消息信息表，通过用户ID确定二者之一（组聊、话题）下具体某一个组聊或话题的消息具体内容。 |

# 8详细设计

# 8.1接口设计

## 8.1.1通讯模块

主要用于底层数据包的传输，不考虑数据包的内容，提供UDP（P2P，广播）、TCP（点对点）的发送和接收功能。其中前者主要处理较短文本字符串的传输，后者主要用于文件的传输。

对外接口（供上层调用）：

1、/\*

功能说明：Udp点对点发送

传入参数：data - 需要传输的数据包、receiverIp - 传输目标Ip地址

\*/

void UdpSendP2P(const QByteArray &data, const QString &receiverIp);

2、/\*

功能说明：Udp局域网内广播发送

传入参数：data - 需要传输的数据包

\*/

void UdpSendBroadcast(const QByteArray &data);

3、/\*

功能说明：Tcp点对点发送

传入参数：data - 需要传输的数据包、receiverIp - 传输目标Ip地址

\*/

void TcpSendP2P(const QByteArray &data, const QString &receiverIp);

对外信号（连接后触发外部事件，Qt信号槽机制）：

1、/\*

功能说明：Udp有数据到来时发出相应信号

传出参数：data - 接收的数据包、senderIp - 传输来源Ip地址

\*/

void UdpReceive(QByteArray\* data, QString\* senderIp);

2、/\*

功能说明：Tcp有数据到来时发出相应信号

传出参数：data - 接收的数据包

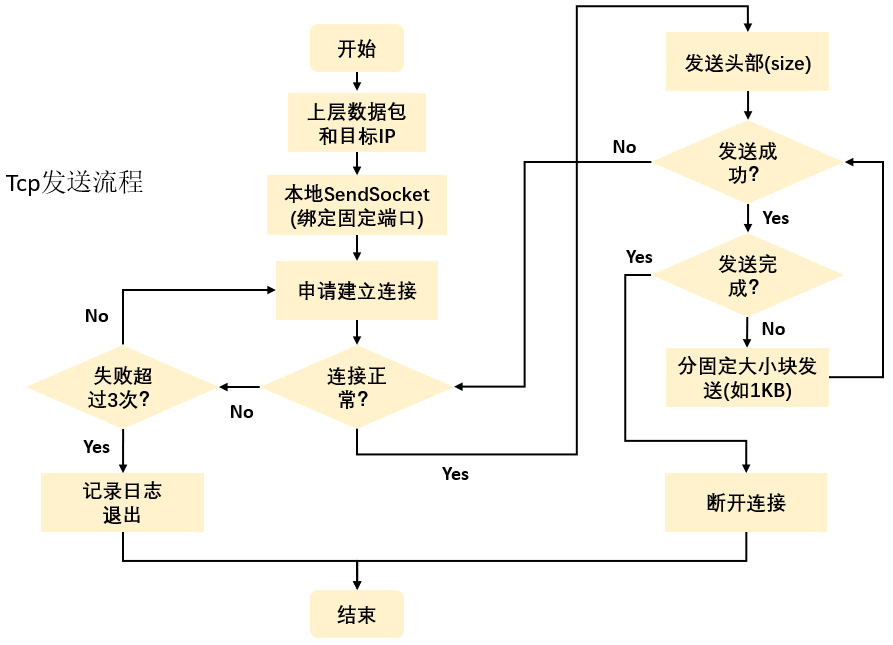
\*/

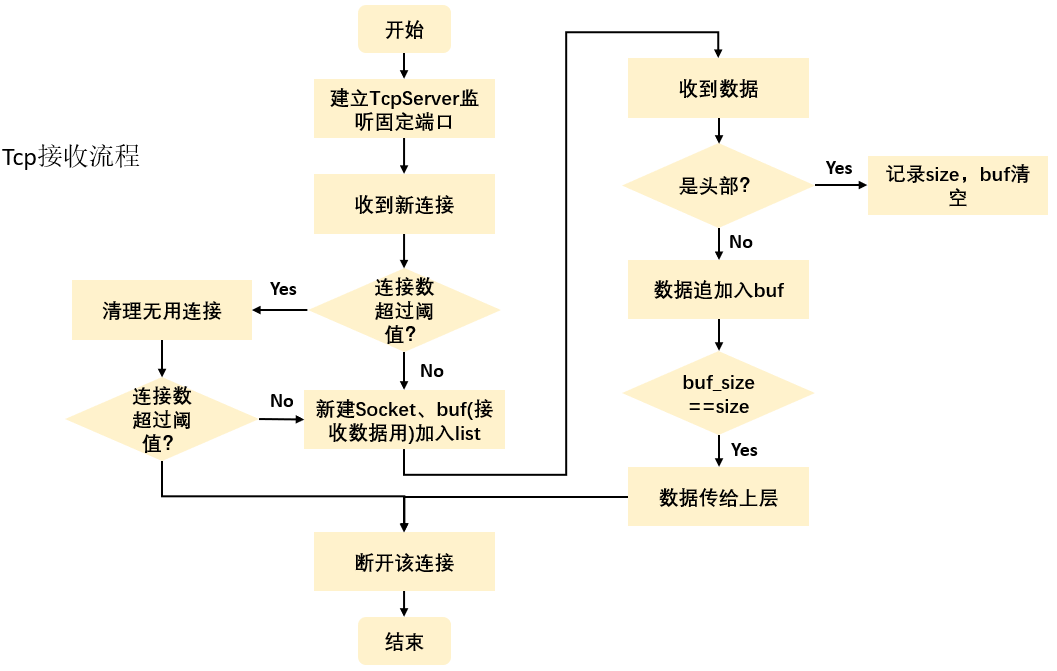
void TcpReceive(QByteArray\* data);

实施细节：

1、Udp通信，逻辑比较简单，分为专门处理接收的套接字（SendSocket）和专门处理发送的套接字（RecvSocket），分别绑定不同的固定端口号，当有相应事件来临时直接调用

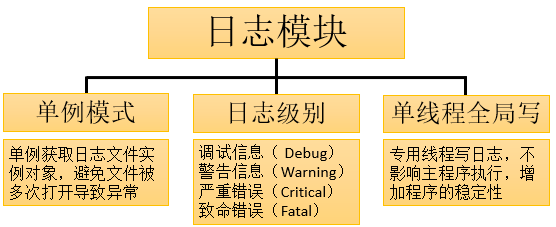
2、Tcp通信流程图：





## 8.1.2日志模块

以单例模式创建一个单线程，按消息队列形式全局写日志，其中日志级别用0~3来进行记录调试信息、警告信息、严重错误和致命错误。



日志写入接口：

输入某类日志的具体日志信息

void Loger::outputMessage(QtMsgType type, const QMessageLogContext &context, const QString &msg)

LogWrite::LogWrite(QObject \*parent) : QObject(parent)

{

m\_pThread = new QThread; // 新建一个线程

this->moveToThread(m\_pThread);

m\_pThread->start(); // 线程开始执行

}

LogWrite::~LogWrite()

{

m\_pThread->terminate(); // 析构函数，用于终结线程

m\_pThread->deleteLater();

m\_pThread = nullptr;

}

void LogWrite::slot\_writeInfo(QString info) // 写信息

{

QFile file("log.txt"); //日志文件

file.open(QIODevice::WriteOnly | QIODevice::Append); //打开方式

QTextStream text\_stream(&file);

text\_stream << info << "\r\n";

file.flush();

file.close();

}

## 8.1.3读写json模块

// 在线广播消息

{

"type": ONLINE\_MSG,

"user\_id": "Bd23J23Iueh4342dsa", // 18位随机码[0-9a-zA-Z]

"user\_ip": "222.222.222.222"

};

// 离线广播消息

{

"type": OFFLINE\_MSG,

"user\_id": "Bd23J23Iueh4342dsa",

"user\_ip": "222.222.222.222"

};

// 广场聊天消息

{

"type": SQUARE\_MSG,

"msg\_id": "23Jerk34FG4fdGV", // 15位随机码[0-9a-zA-Z]

"user\_id": "Bd23J23Iueh4342dsa",

"type\_id": "", // 广场聊天则空

"data": "这是广场聊天的消息"

};

// 群组聊天消息

{

"type": GROUP\_MSG,

"msg\_id": "sfd3rk34FG434fs",

"user\_id": "Bd23J23Iueh4342dsa",

"group\_id": "fi23J9Sfe98", // 群组聊天则为10位随机码[0-9a-zA-Z]

"data": "这是群组聊天的消息"

};

// 话题评论消息

{

"type": TOPIC\_MSG,

"msg\_id": "df32fwg5SfFFF3a",

"user\_id": "Bd23J23Iueh4342dsa",

"topic\_id": "I23222jfeuk", // 群组聊天则为10位随机码[0-9a-zA-Z]

"data": "这是话题评论的消息"

};

// 建立主题消息

{

"type": NEW\_TOPIC\_MSG,

"topic\_id": "fffwer31Wd2",

"user\_id": "Bd23J23Iueh4342dsa",

"theme": "电影推荐",

"detail": "大家有没有好看的电影推荐啊,动作类,科幻类的都可以呀!"

};

// 建立群组消息

{

"type": NEW\_GROUP\_MSG,

"group\_id": "FW2Srr42dff",

"user\_id": "Bd23J23Iueh4342dsa",

"name": "户外运动小组",

"intro": "专注于户外运动50年,欢迎大家加入组聊哦!",

"member\_list":

[

{

"user\_id": "d33Wkfjie324Hi7dda", // 18位随机码[0-9a-zA-Z]

"user\_ip": "222.222.222.220"

},

{

"user\_id": "3jeu23hJHHHH948Gsd", // 18位随机码[0-9a-zA-Z]

"user\_ip": "222.222.222.221"

},

{

"user\_id": "Bd23J23Iueh4342dsa", // 18位随机码[0-9a-zA-Z]

"user\_ip": "222.222.222.222"

}

]

};

# 9情感分析功能设计

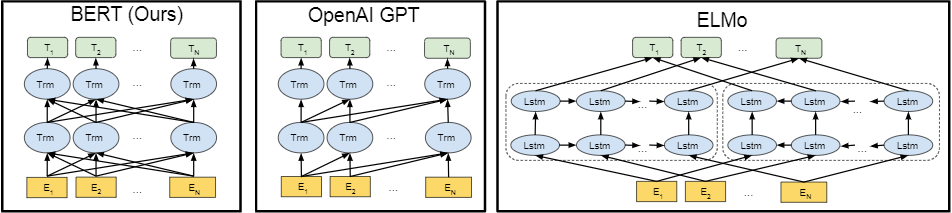
## 9.1介绍BERT

结合本P2P局域网聊天软件的特点，除了基本的通信聊天功能，我们将情感分析这一自然语言领域的分类任务用于软件之中，统计某一话题赞同和不赞同的人数，使得话题讨论更具意义。

2018年提出的BERT（Bidirectional Encoder Representations from Transformers）模型，使用多头注意力机制和位置嵌入，替换不易并行的循环神经网络，一举打破自然语言处理领域11个不同问题的最好记录。本文将采用BERT模型对本软件的话题聊天内容做中文文本分类。

BERT使用双向Transformer，在无标签文本上预训练深度双向表达，在每一层都受到左右文本的限制；OpenAI GPT使用从左到右的Transformer；ELMo使用独立训练地从左到右和从右到左的LSTM的连接来生成下游任务的特征。三者除了架构的不同，BERT和OpenAI GPT是基于微调的方法，然而ELMo是基于特征的方法。

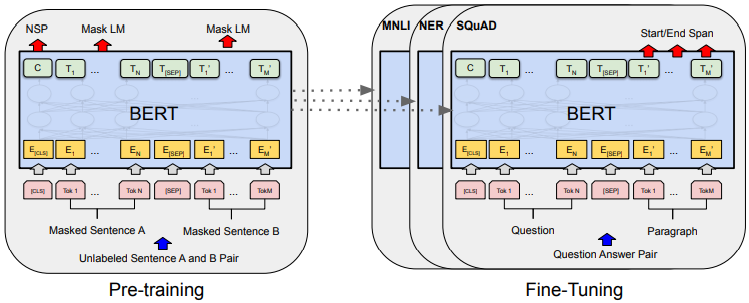
预训练模型的不同：



BERT训练分两个阶段：第一阶段采用双层双向Transformer模型，通过MLM（masked language model）和NSP（next sentence prediction）两种策略进行预训练；第二阶段采用Fine-Tuning的模式应用到下游任务。

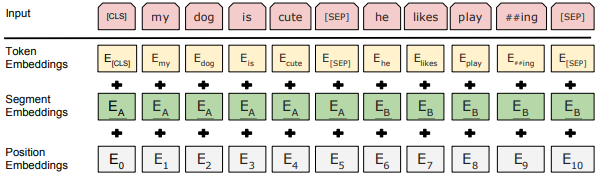
“完形填空”的学习模式迫使模型更多依赖上下文信息预测单词，赋予了模型一定的纠错能力；Transformer模型相比LSTM模型没有长度限制问题，具备更好的能力捕获上下文信息特征；相比单向训练模式，双向训练模型捕获上下文信息会更加全面。BERT一个独特的优点：对于下游不同的任务，都提供了一个统一的结构，即预训练网络结构和下游任务的网络结构差别很小。

BERT的预训练和微调过程：



预训练和微调网络结构仅在输出层不同，所以对于不同的下游任务即使用相同的预训练网络参数。微调时，所有参数都要微调。[CLS]是每个输入样本最前面的符号，[SEP]是特殊的分隔token（比如分隔问题和答案）。

BERT输入表达：



词嵌入由token嵌入、分隔嵌入、位置嵌入相加得到，从而BERT能够知道“句子”这个概念。Tocken嵌入是词向量，第一个单词是CLS标志，可以用于之后的分类任务；分隔嵌入用来区分两种句子，因为预训练包括了以两个句子为输入的分类任务；位置嵌入使得BERT能够生成查询向量和键向量。

## 9.2预训练BERT

BERT预训练中将传统的语言模型建模的任务划分为两个无监督的任务：完形填空（Masked LM）和句对预测（Next Sentence Prediction）NSP。

MLM:随机掩盖一定概率的输入tokens，然后预测这些masked tokens，即只预测这些掩盖的词语，而不是重构整个输入。

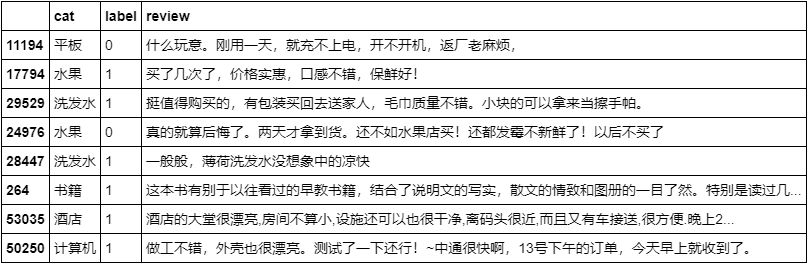
NSP: 许多重要的下游任务，如问答(QA)和自然语言推理(NLI)，都是建立在理解两个句子之间的关系的基础上的，而语言建模并不能直接捕捉这些关系。为了训练一个理解句子关系的模型，BERT预先训练了一个二值化的下一个句子预测任务。

## 9.3微调BERT

对于每个任务，只需将特定任务的输入和输出输入BERT并对所有参数进行端到端的finetune。

## 9.4基于BERT的情感分析

训练数据：从网上下载的网上购物情感极性分类数online\_shopping\_10\_cats（来自各电商平台），共有10个类别、6万多条评论数据，分别关于书籍、平板、手机、水果、洗发水、热水器、蒙牛、衣服、计算机、酒店，正、负向评论各约3万条。



如之前章节介绍，使用BERT的预训练模型，并在新数据集上进行微调可以得到很好的任务结果。本软件的情感分析即是中文文本的二分类问题，而在中文上BERT提供了预训练模型，所以我们能很快的利用预训练模型在本数据集上进行微调，得到本软件的话题情感分析模型，对某一话题下的评价进行正向和反向的情感统计。