

SHANGHAIUNIVERSITY

毕业设计（论文）

**UNDERGRADUATEPROJECT(THESIS)**

**题目:聊天机器人**

|  |  |
| --- | --- |
| **学院** | **计算机工程与科学学院** |
| **专业** | **计算机科学与技术** |
| **学号** | **14122151** |
| **学生姓名** | **王晨悦** |
| **指导教师** | **戴佳筑** |
| **起讫日期** | **2018.01.29 – 2018.06.08** |

目录

[摘要 III](#_Toc514835253)

[ABSTRACT IV](#_Toc514835254)

[第1章 绪论 1](#_Toc514835255)

[§1.1 聊天机器人制作的背景及意义 1](#_Toc514835256)

[§1.2 聊天机器人技术发展现状 1](#_Toc514835257)

[§1.2.1 机器学习算法 1](#_Toc514835258)

[§1.2.2 聊天机器人技术发展现状 2](#_Toc514835259)

[§1.3 聊天机器人的使用 2](#_Toc514835260)

[§1.4 本文组织结构 2](#_Toc514835261)

[第2章 单词的向量表示 4](#_Toc514835262)

[§2.1 SKIP-GRAM概述 4](#_Toc514835263)

[§2.2 代码及训练 6](#_Toc514835264)

[§2.2.1 代码 6](#_Toc514835265)

[§2.2.2 训练过程 6](#_Toc514835266)

[§2.2.3 训练结果 7](#_Toc514835267)

[§2.3 本章小结 9](#_Toc514835268)

[第3章 基于LSTM构建机器人 10](#_Toc514835269)

[§3.1 RNN及LSTM概述 10](#_Toc514835270)

[§3.2 代码及训练 10](#_Toc514835271)

[§3.2.1 代码 11](#_Toc514835272)

[§3.2.2 训练过程 11](#_Toc514835273)

[§3.3 本章小节 12](#_Toc514835274)

[第4章 图形界面搭建 17](#_Toc514835275)

[§4.1 网站整体设计 17](#_Toc514835276)

[§4.2 开发环境简介 18](#_Toc514835277)

[§4.3 代码实现 22](#_Toc514835278)

[§4.4 本章小结 24](#_Toc514835279)

[第5章 总结与展望 26](#_Toc514835280)

[§5.1 本文总结 26](#_Toc514835281)

[§5.1.1 本文的主要工作 26](#_Toc514835282)

[§5.1.2 本文的主要创新点 26](#_Toc514835283)

[§5.2 展望 26](#_Toc514835284)

[致谢 28](#_Toc514835285)

[参考文献 29](#_Toc514835286)

[附录：部分源程序清单 31](#_Toc514835287)

聊天机器人

# 摘要

XXX。

关键词：XXXX，XXX，XXX，XXX

Chat Robot

# ABSTRACT

In recent years, with the rapid development of Micro-blog, the need that users gain the access to information is also a linear growth momentum. The amount of Sina Micro-blog registered users has reached 503 million by 2012. However, the daily flood of Micro-blogs has a serious impact on the quality of information users receive. Thus, how to find content that they are interested in quickly and accurately? Or can we push the information according to the user's interest actively? That is what this paper concerns.

User model is a formal description of users' interests. To establish a precise user model for Micro-blog users, in order to recommend the information they concerned about and contents they are interested in, is the development trend of personalized recommendation. The achievements of this paper can be used for e-commerce, public opinion monitoring, advertising and other areas.

In this paper, Micro-blog user model construction method based on the ontology technology is discussed. First of all, according to each user's micro-blog content analysis, extract the keywords which represent the content information of each Micro-blog; then create user's eigenvectors in order to calculate the user's interest degree; finally match these keywords with the ontology library to create Micro-blog user interest model. The achievements of this paper will lay the foundation of personalized service based on Micro-blogs.

**Keywords:** Ontology, Micro-blog, User model, Interest Degree

# 绪论

本章主要描述了聊天机器人构建的背景、意义等，研究了国内外在机器学习领域科学进展现状，进而调查分析国内外基于机器学习在聊天机器人领域的发展现状，并在此基础上，依托tensorflow框架，运用rnn算法搭建了一个聊天机器人。

## 聊天机器人制作的背景及意义

聊天机器人在本文中是指一个可以根据用户的输入来提供输出的程序，可以根据用户的提问作出一些智能的回答，或者跟用户进行简单的聊天交流，基本概念是模拟人类的对话。聊天机器人的概念诞生于二十世纪四十年代，经过长时间的研究和发展。在几十年的发展过程中，人类对自然语言处理的算法也不断发展，这给聊天机器人的发展提供了必备的基础，也顺理成章地推动了聊天机器人的智能程度变高。但是由于自然语言经过更漫长的时间进化，很难总结出固有的规律，语法也繁杂难以列举，更有很多的习惯用法，使得计算机程序模拟人类对话的难度变得很高。而近年来机器学习算法越来越火热，也给了聊天机器人的发展提供了一条更宽敞的路。机器学习的存在，使得聊天机器人的对话产生并不需要人为总结所有的语法规律，而是可以通过不断地训练，让程序本身就有应有的智能。结合机器学习算法，让计算机程序具备自己的智能，实现模拟人类对话，这就是本文的理论支撑点。

而在应用方面，聊天机器人已经在很多领域创造出了自己的价值。在通讯平台、教育等方面起到了很重要的作用，很多公司也开始使用较成熟的聊天机器人实现简单的客服等工作，而一些大公司，则结合更多的类似语音识别、图像识别技术，创造出能接收更多种形式的输入的机器人，帮助很多需要的人也创造了很大的价值。

## 聊天机器人技术发展现状

从聊天机器人的概念诞生以来，就有大量学者不断地进行研究钻研，近年来，随着机器学习算法变得更成熟和便于应用，聊天机器人的发展也开始走向一个高潮。接下来先具体阐述本文将应用的机器学习算法相关基础，然后阐述聊天机器人的发展历程和现状以及实际各种方面的应用。

### 机器学习算法

机器学习是人工智能的一个分支（1000字）

### 聊天机器人技术发展现状

聊天机器人的发展我们从1966年的ELIZA说起，由一位MIT的教授joseph所发明。最初创造ELIZA的目的是想让它作为一个陪伴人们聊天的工具，本身ELIZA的功能也只是简单的陪伴。但是事实证明，ELIZA的效果非常好，有聊天记录显示，曾经有来访者跟ELIZA聊天以后发出这样的感慨“你在某些方面挺想我父亲的”，这就是ELIZA的厉害之处。ELIZA的实现并没有涉及很复杂的技术，甚至没有最简单的语义理解，只是简单的将输入和数据库进行匹配，处理的内容也只是文本。ELIZA的工作原理是这样的，它将输入去与关键词匹配，大概提供了两百多个模板，每次匹配到一个或多个关键词就根据相应的模板返回内容。如果没有匹配到关键词，则简单地把输入中的“我”改成“你”返回，这样的规则虽然简单但是非常有效。ELIZA的成功在我看来主要得益于它的规则，一是非常有效的模板和关键词设置，第二是对于没有匹配到关键词的输入的输出规则。而这样的规则，就成了很多现在的产品的基石，比如ALICE、MITSUKU、机器人小冰等，区别就是会增加或者更改更好的模板，以满足更多的功能。但是这样做的缺点很明显，模板数量是有限的，而语言的创造力是无限的，在不理解语义的情况下，仅仅根据模板去回答问题，会非常受限于模板和性能。

之后的机器人就开始对语义进行分析，比如SIRI、MBI的WASTON等等。

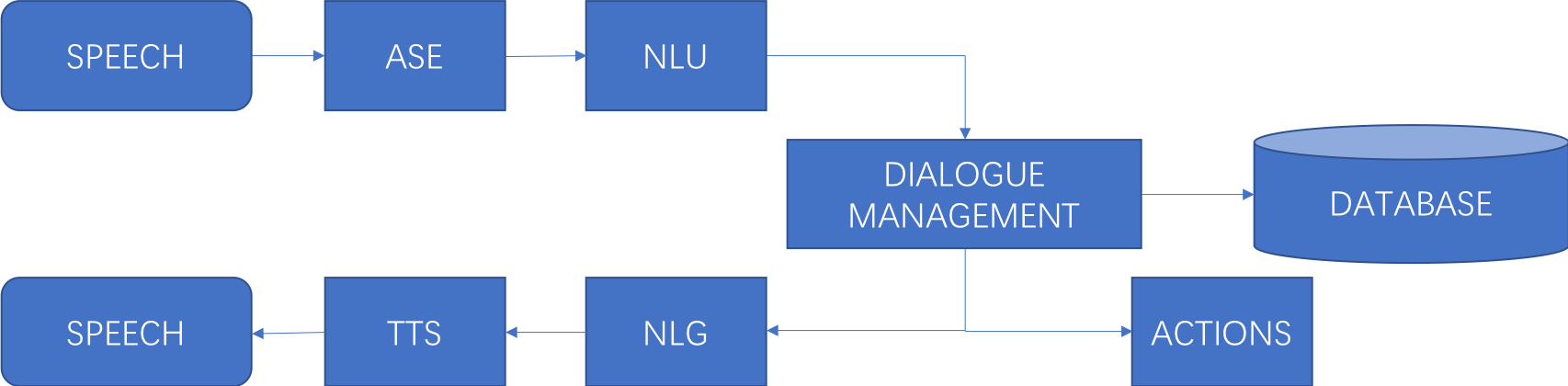
而随着科学发展，我们能收集到的数据量已经大量增加，而处理数据的能力也大大增强，在此基础上，机器人不需要限制在固定的模板上。

总结上面的发展，我们可以看到机器人的架构发生了如下的变化：

最古老的一种，只能处理文本，处理过程是简单的匹配，如图：



下一个阶段出现更多的是这样的架构，它可以在一定程度上理解自然语言，能执行一些比较复杂的动作，之后根据对话管理系统生成一些回答并返回，如图：



而现代的机器人框架则更多结合机器学习算法



旨在构建一个足够强大的神经网络，通过大量的训练来实现智能回答。

## 聊天机器人的使用

我们很容易关注到今年来，很多公司都开始使用聊天机器人，这代表了一种发展趋势，使用更多智能产品来代替人类的工作，解放生产力的同时，提高某些方面的用户体验以及更好地为人类服务。在国内，微信在2016年就开始使用聊天机器人，虽然是处于实验状态，并在中途曾关闭了机器人账号，但随着技术的发展，现状的机器人已经很成熟了。QQ小冰也是另一个例子，不过很显然也是一个实验的状态。百度在人工智能方面投入很大的成本，刚刚上线的APP“简单搜索”也运用了部分聊天机器人的功能，来实现语音的输入到搜索结果的产生的流程优化，提高用户体验。而在国外，聊天机器人的运用则更为领先，谷歌公司一直在人工智能、深度学习等方面有着很好的研究。这几年，无人车技术也开始成为大头公司争夺的高地之一，在无人车上也运用了很多聊天机器人相关技术，用以方便用户的使用。在教育方面，聊天机器人也开始发力，对学习语言的同学来说肯定不陌生翻译器，各种翻译程序在聊天机器人发展迅速的今天，效率和准确度大大提升，而更多的产品走进某些领域的简单应用。因为聊天机器人不需要像人类一样休息，同时在具备大量的训练样本的条件下，可以根据训练样本训练成为一个某领域的“专家”，那么如果在经过样本的提炼和安排之后，可以说各个领域，只要涉及信息的输入和查询的地方，都可以运用聊天机器人来代替人类的繁复工作。

## 本文组织结构

整篇论文分为五章。

第一章介绍了聊天机器人的研究背景、研究意义，引出了聊天机器人实现背后的技术点，并介绍了相关技术的概念和发展。

第二章主要介绍了聊天机器人实现的第一步，对训练样本的选取和处理的过程，包括处理技术的介绍和核心代码部分，总结了需要提高的点。

第三章主要介绍了将第二章提到的训练样本集训练的响亮结果，存储为二进制文件，并传入LSTM算法进行训练的过程，这一步是整个聊天机器人训练和生成的核心步骤，在第三章的训练之后，聊天机器人就具备了自己的智能，可以接受训练集意外的的句子，并通过计算得到合理的输出。

第四章是为聊天机器人搭建了一个图形界面，介绍了主要的框架和服务器环境等等，通过网页的形式展现给用户。便于用户操作和测试。

第五章总结全文，阐述了本文的主要工作难点、重点和创新点，也总结了需要进一步研究和改进的点。

# 技术综述

## 单词的向量表示

算得到（3000字）

## 循环神经网络

算得到（3000字）

## 网站框架描述

算得到（1000字）

# 单词的向量表示

本章具体描述了聊天机器人实现的第一步，核心是将训练集中的中文文本训练成为向量，以作为下一步骤循环神经网络训练的输入。另外，涉及到训练之前的文本预处理，也会涉及部分包括中文分词之类的描述。

## 文本预处理及SKIP-GRAM概述

（1500字）微博用户模型的构建主要包括特征词提取、兴趣度计算以及用户模型的生成。以下将分为三个小节来详细描述构建方法。

首先，我们将获取微博用户信息，理想的用户建模方法应该无需用户主动提供任何信息。自动用户建模就是根据用户在微博平台上注册留下的信息及用户发布在微博平台上的微博信息来构建用户模型。我们可以通过微博平台提供的API，获取到用户的昵称、密码、编号、姓名、关注数、粉丝数、发表微博的数量等个人基本信息以及每个用户在平台上发布过的所有微博信息内容来构建用户模型。

在中文信息处理领域，对中文自动分词的研究已较为成熟，经典的分词方法主要有最大向前匹配法、逐词遍历匹配法、最小向前匹配法等。本文未对分词方法进行深入研究，而是选择直接使用开源的中国科学院计算技术研究所研制的汉语词法分析系统ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System)[31]，该分词系统分词准确率达到97%以上。本文实现的本体构建系统中使用ICTCLAS汉语分词系统官网提供的32位Windows操作系统下的JAVA版本的中文分词和词性标注。

计算机不具有人类的智能，不能像人类一样阅读微博后根据自身的知识和理解能力对微博内容产生理解。因此，在进行微博语义扩展之前首先要将微博转换成易被计算机理解和识别的结构形式。微博的表示要求能够准确有效的表达微博内容，并且还要易于计算机处理。

目前，典型的文本表示方法主要有：布尔模型（Boolean Model）、向量空间模型（VSM）、语言模型（Language Model）、潜在语义索引（LSI, Latent Semantic Indexing）[32]和概率检索模型（Probability Model）[33][34]。这些模型从不同的角度出发，使用不同的方法标注特征词权重和相似度计算等问题。

向量空间模型由Salton等人于20世纪70年代提出，并成功地应用于著名的SMART文本检索系统。VSM被广泛应用于文本分类、文本聚类、信息检索等领域。近年来，在文本挖掘领域向量空间模型已经成为最常用的文本表示方法。VSM是基于这样一个关键假设下提出的，即文档中各词条出现的先后顺序是无关紧要的，每个特征词对应特征空间的一维，他们每一维对于判定文档所属的类别所起的作用是相互独立的。因此，可以把一篇文档看成是一系列无序词条的集合，从文档中选取出n个特征词来表示文本就是形成一个n维向量空间。例如一篇文档中选取三个特征词t1、t2、t3，那么这篇文档就表示为。但是对于整个文档来说，每个特征词对文本的重要程度不同，因此，需要对每个特征词赋予一定的权重。一篇具有n个特征词的文档利用VSM表示方法就可以表示为公式（1）。

 （1）

其中，di是第i篇文档，tij表示第i篇文档的第j个关键词，wij是第i篇文档的第j个关键词权重。关于权重的计算方法有很多种，将在3.3.2兴趣度计算步骤中再作介绍。本文中微博文本表示采用VSM方法。

经过文本分词处理后，需要抽取一定数量的特征词作为向量的各维表示文本。然而，文本中每个特征词对文本主题内容的贡献度不一样，即每个特征词的权重不同，如何准确有效地计算特征词权重成为重要的研究点。在研究最初特征词的权重只有0或者1，如果该特征词在文本中出现过它的权重就设为1，否则设为0。这种方法完全没有体现出在文本中出现的特征词之间对文本主题内容贡献度的差异性，所以这种权重计算方法慢慢被更精确的基于词频统计的方法替代。

常用的权重计算方法有布尔函数、特征词频平方根、WIDF函数及TF-IDF法等。目前使用最为广泛也是本文中所用到的方法就是TF-IDF法，计算方法如公式（2）所示。

将每个特征词在某个用户发表的所有微博中出现的权重相加，就得到了该用户对于该特征词的兴趣度值。

本文中利用基于维基百科的中文本体自动向上扩展建立一个树状层次结构，其中最主要的关系为上下位关系，下一层的同义词集是其父节点的下位关系，反之亦然。上下层的关系也是包含与被包含的关系，下一层的节点包含于其父节点。我们手动建立了一个基于维基百科分类下的分类特征词本体库，共输入有4757条记录，分为财经、IT、健康、体育、旅游、教育、招聘、文化、军事九个大类。

当之前分析出的用户兴趣主题与我们所建立的本体库中的某个词匹配时，说明用户也对该兴趣主题的直接父节点以及祖先节点感兴趣，只不过对它们的感兴趣程度有所差别。基于这个思想，用户模型构建中会自动查找并更新兴趣主题的父亲节点，直至根节点。

通过以上步骤就可以构建出微博用户模型了，包括用户兴趣树和兴趣度两大模块。在用户模型中，用户的兴趣表示为一棵用户兴趣的本体子树，如**错误!未找到引用源。**所示。这棵本体子树包含的信息主要有：

（1）用户的兴趣主题

（2）用户不同的兴趣主题对应的兴趣度

（3）兴趣主题之间的层次结构关系

图3‑2用户模型中一个用户的兴趣树

某个用户对于其所有微博中的每个特征词的兴趣度值已经过计算得到，在用户模型构建中，也将这部分内容在界面上显示出来，如图‑3‑3所示。

本章主要介绍了用户模型的基本概念以及常用用户模型和本体用户模型的表示方法。另外，详细地描述了微博用户模型构建方法的三大步骤，也是本文的最主要内容和贡献。

常用的三种用户模型表示方法缺乏语义和统一的标准。而基于本体的用户模型表示方法不仅能充分描述用户兴趣的语义，还具有兼容性和可扩展性，很好地解决了常用用户模型表示方法存在的问题，是用户模型表示方法的发展方向，因此本文将实现基于本体的用户模型构建。

本文采用基于词频统计的分词方法进行微博文本内容的特征词提取；基于TF-IDF方法计算出微博特征词的权重，并相加得到其兴趣度；最后生成用户模型，显示出用户兴趣树以及用户对每个特征词的兴趣度

## 代码及训练

根据以上的描述开始写我们的代码。代码量不是很大，更关键的部分在训练以及训练过程中的参数调整。

### 代码

（200字）

。

### 训练过程

（1000字）

### 训练结果

（500字）效果图以及训练结果描述。

## 本章小结

本章主要具体阐述了训练样本的选择、文本的预处理等。详细阐述了文本预处理过程中，中文分词的难点和关键技术。在预处理之后，将训练集运用skip-gram训练成向量表示。详细阐述了skip-gram的原理，并将其与其他文本向量表示算法比较。详细描述了训练的过程，以及对最后训练结果的存储。

# 基于LSTM构建机器人

本章是全文的重点章节，全面阐述了本文的工作内容。首先简单介绍了用户模型的概念及其表示方法，随后将本文的构建过程完整的进行了描述，并将部分系统功能运行的结果作了展示。

## RNN及LSTM概述

（1000字）

。

## 代码及训练

（100）

### 代码

（200）。

### 训练过程

（2000）。

## 本章小节

。

# 图形界面搭建

本章主要介绍图形界面的搭建过程。描述了Django网站框架、界面设计过程、网站搭建过程，最后是界面最后成型以及功能的展示。

## 网站整体设计

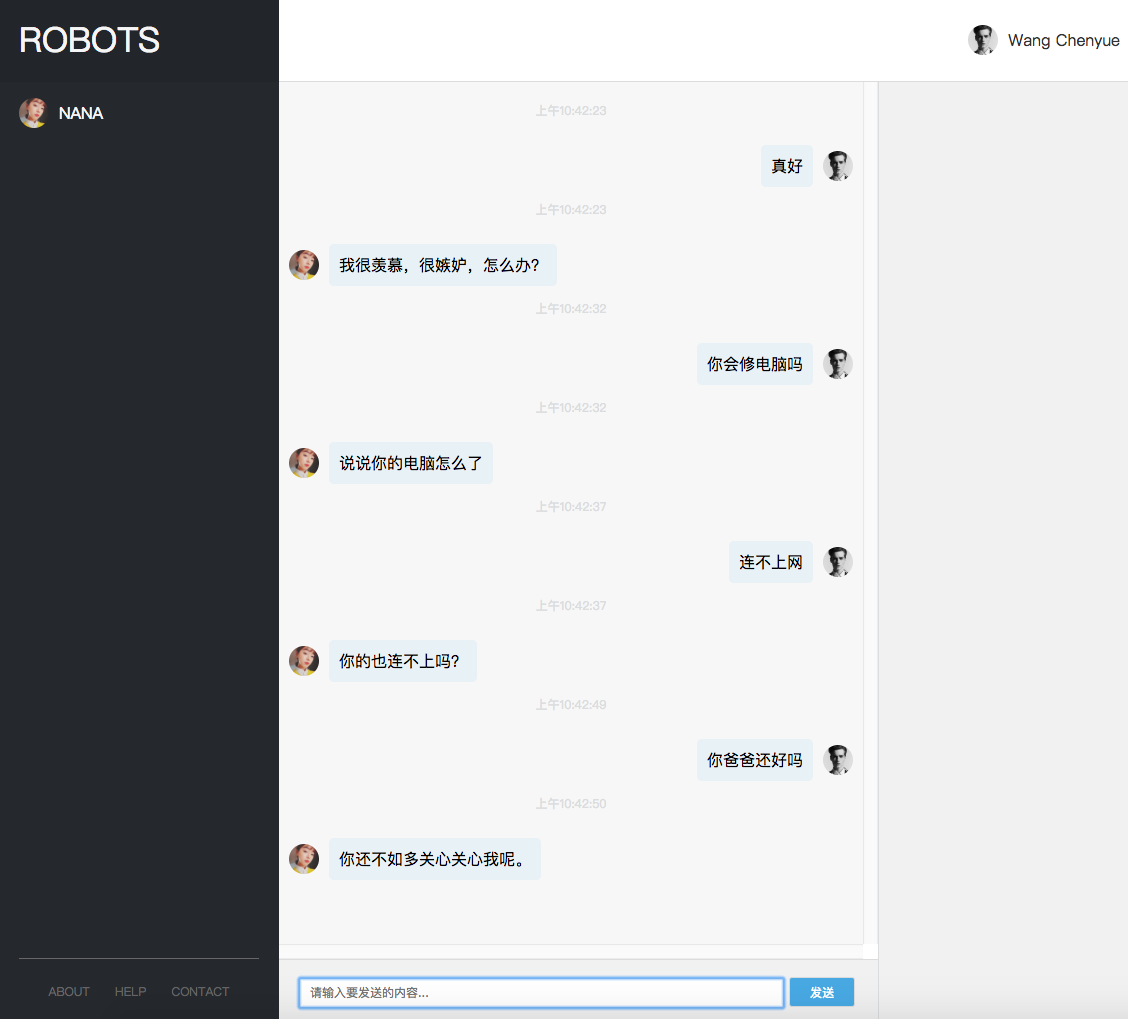
网站的主要功能就是给用户提供一个使用界面，可以跟聊天机器人聊天，所及基本上可以归类为网页聊天室的设计。

网站后端为训练好的模型，用户在前端输入框输入文字，点击“发送”或者按回车发送消息到后端模型，然后前端接受后端传回的数据，展示在输出窗口。

考虑到这个网站可以制作成为聊天机器人平台，可以添加多个不同类型的机器人，比如根据不同的训练样本，可以制作出用于在线诊疗的辅助医疗机器人、用于心理学治疗的陪聊机器人，或者在某个领域的专家，比如可以通过训练大量的计算机修理方面的专业内容，训练出可以用于帮忙日常诊断电脑是否出问题以及维修的机器人。为了实现这个功能，在网站左侧设计了机器人列表，以后在训练出更多的机器人之后可以逐个添加。

出于为用户提供个性化服务考虑，可以添加用户登录注册功能，所以在网页上方设置了一个现实用户头像和姓名的链接，设计点击可以跳转到个人主页，这样，这部分设计暂时未实现，不过不影响核心功能——与机器人聊天的展示。（600字）

界面展示如图：



## 开发环境简介

开发环境是macos+python3.6+django2.0，没有涉及数据库内容，代码版本同步工具是git，最后发布环境是云服务器，服务器环境为nginx。（200字）

## 代码实现

（200字）

## 本章小结

本章主要展示了最后的结果，将机器人的核心代码赋予一个ui界面，并展示了ui界面的设计过程以及最后的实现结果。

如果只是作为一个机器人的展示窗口的话那么基本功能已经完成了，不过如果作为一个产品来看待则还有更多的可以改进的地方。首先是可以增加用户系统，可以根据用户的需求来定制具备某个领域知识的机器人，还有移动端的适配还没有做好。

# 总结与展望

本章对全文的主要工作和创新点作了总结，并提出需要进一步研究和改进之处。

## 本文总结

（200）。

### 本文的主要工作

本文的主要工作是基于机器学习构造一个聊天机器人，主要内容包括以下几个方面：

（1）单词的向量表示：。

（2）利用循环神经网络对词向量进行训练：。

（3）训练结果测试：。

（4）机器人个性化和天气查询功能实现：。

（5）用户图形界面设计与实现：。

### 本文的主要创新点

本文基于机器学习实现的聊天机器人的创新点主要有以下两项：

（1）实现了机器人的个性化。

（2）结合模块化的聊天机器人架构和机器学习，在实现以较高准确率与用户聊天的基础上，可以通过模块的加载实现天气的查询，并且具有一定的拓展性，可以通过模块的加载实现更多的查询以及推荐功能。模型方法

## 展望

本文基于lstm的机器学习算法训练出了一个性能较好的模型，可以在一定程度上个性化机器人，并根据一些模块的添加，可以实现天气查询的功能。但是机器人的功能仅限于此，还有很多可以提升的地方。理想的机器人应该更智能，比如可以根据用户的需求推荐附近餐厅等。简单的推荐可以在用户输入之后提取关键词匹配，并执行对应代码来实现，但是会存在的问题是这样的推荐结果是固 定的，或者说是根据某些搜索引擎的api变化的，而非针对用户的，那么怎么让机器人来给用户个性化推荐就是一个改进的方向。将机器人本身个性化并非难事，如何让机器人识别跟自己聊天的人是谁，并知道跟自己聊天的人的个人信息，这就是值得思考的问题。另一个方面，基于机器学习制作的机器人永远受到训练集的影响。在实验的过程中，发现在100M的样本训练下，机器人的性能是非常差的，准确度极低，上文提到本实验的训练集为5G以上，但这并不就是最优的结果。训练集带 来的思考有两方面，一方面是如何更有效率地增加训练集，如何去改进训练方式可以训练更多的样本。另一个方面是如何挑选训练样本。重复的10G“你好”构成的训练集并不会产生好的结果，所以训练集的挑选是一个很重要的方面，但本实验对于这方面的研究几乎没有，对于这方面的研究所得的资料也很少。

# 致谢

XXX。

# 参考文献

1. 林鸿飞,杨元生.用户兴趣模型的表示和更新机制.计算机研究与发展，200239(7):838-842.
2. 胡学联,潘金贵,李俊,张灵玲. 一个个性化的信息搜集Agent的设计与实现.软件报,2001,12(7):1074-1079.
3. 应晓敏,面向Internet个性化服务的用户建模技术研究. 2003,国防科技大学:长沙
4. 徐振宁,张维明,陈文伟. 基于Ontology的智能信息检索. 计算机科学,20.28(6):21-26.
5. 李勇.智能信息检索中基于本体的个性化用户建模技术及应用. 2002,国防科技学:长沙.
6. FragoudisD.User Modeling in Information Discovery:An overview.InProceedings of Advanced Course on Artificial Intelligence,1999(ACAI99), July,1999, Greece.
7. Balabanovie M, Shoharn Y.Learning Information Retrieval Agents:Experiments with Auto-mated Web BrowsingIn:Proceedings of the AAAISpring Symposium Series on Information Gathering from Heterogeneous, Distributed Environments, March, 1995:13-18.
8. Lieberman H.Letizia. An Agent that Assists Web Browsing.In:Proceedings ofthe International Joint Conference on Artificial Intelligence, Montreal, August, 1995:924-929.
9. Chan P.K.A Non-Invasive Learning Approach to Building Web User Profiles in KDD-99Workshop on Web Usage Analysis and User Profiling, 1999, New York:ACMpress.
10. Schwab L,Kobsa A,and Koychev I. Learning about User from Observation[J].in AAAISpring Symposium on Adaptive User Interface, 2000, Standord,California:AAAI Press.
11. Adomavicius G and Tuzhilin A.Using Data Mining Methods to Build CustomerProfiles.IEEE Computer.Feb 2001:74-82.
12. 余伟. 基于本体的微博客用户行为模型研究[J]. 广东技术师范学院学报, 2010, 31(006): 27-30.
13. 赵岩露, 王晶, and 沈奇威. "基于特征分析的微博用户兴趣发现算法." 电信工程技术与标准化 25.11: 79-83.
14. Uschold M. and Gruninger M., Ontologies: Principle, Methods and Application. Knowledge Engineering Review, 1996, 11(2): p.93-155.
15. Neches R, et al. Enabling technology for knowledge sharing.AI Magazine, 1991, 12(3): p.36-56.
16. Swartout W. Ontologies, 1999. Intelligent Systems and their Applications, IEEE. Issue:1, p.18-19.
17. Gruber T.R.,A Translation Approach to Portable Ontology Specifications .knowledge acquisition, 1993, 5(2): p.199-221.
18. Borst W.N. Construction of Engineering Ontologies for Knowledge Sharing and Reuse. 1997, University of Twente: Enschede.
19. Studer R, Benjamins VR, and Fensel D. Knowledge Engineering: Principles and Methods. Data and Knowledge Engineering, 1998, 25(1-2): p.161-197.
20. 朱晓冰，寇雅楠基于维基技术的本体构建方法探讨图书馆学研究2009.1: p.55-58.
21. 于江生，俞士汝.中文概念词典的结构.中文信息学报.2002年第4期: p.64-68.
22. 由丽凭，杨翠.汉语框架语义知识库概述电脑开发与应用.2007年06期: p.72-74.
23. 张晶，姚建民，赵铁军，李生.基于WordNet和HowNet建设双语语义词典.高技术通讯.2001.12(2-3): p.43-46.
24. 侯汉清，薛春香.用于中文信息自动分类的《中图法》知识库的构建.中国图书馆学报.2005年第5期: p.67-70.
25. 罗志成，马费成，吴晓东，宋倩倩.从维基分类系统构建中文语义词典研究.信息系统学报.2008年02期: p.35-39.
26. 杨玲贤.基于ontology的教学资源知识库构建.计算机与现代化.2009年第11期: p.27-32.
27. Fromkin V, Rodman R. Introduction to Language. London: Holt, Rinehart and Winston, Inc., 1988, 12(1): p.10–16.
28. Brachman R, What IS-A is and isn't: An analysis of taxonomic links in semantic networks [J]. IEEE Computer, 1983, 16(10): p.30–36.
29. Fischer G. User modeling in human–computer interaction[J]. User modeling and user-adapted interaction, 2001, 11(1-2): 65-86.
30. 王金花. 一种利用本体关联度改进的TF-IDF特征词提取方法[D]. 河北大学, 2011.
31. 中国科学院计算技术研究所[EB/0L]. 中文自然语言处理开放平台. http://www.nlp.org.en/project/ project.php?proj\_id=6, 2005-2-2.
32. Scott Deerwester, Susan T Dumais, Furnas W George, et al. Indexing by latent semantic analysis[J]. Journal of the American Society for Information Science, Vol 41, 1990.
33. 王娟琴. 三种检索模型的比较分析研究—布尔模型、向量模型、概率模型[J]. 情报科学, 1998, 16(3): 225-231.
34. 景玉峰等. 概率检索模型.现代图书情报技术, 1987(1): 29-31.
35. 冀胜利, 李波. 基于SVM的中文文本分类算法[J]. 重庆工学院学报(自然科学), 2008, Vol.22 No.7: 84-87.
36. 崔争艳. 中文短文本分类的相关技术研究[D].
37. 埃克尔著, 陈昊鹏译. Java编程思想[M]. 第4版. 北京: 机械工业出版社, 2007.
38. 李刚. 疯狂Java讲义[M]. 第2版. 北京: 电子工业出版社, 2008.
39. 李刚. 轻量级Java EE企业应用实战:Struts2＋Spring3＋Hibernate整合开发[M]. 第3版. 北京: 电子工业出版社, 2011.
40. Dou Shen, Jian-Tao Sun, Qiang Yang, et al. Latent Friend Mining from Blog Data[C]. Sixth International Conference on Data Mining, 2006: 552-561.
41. Katarzyna Musiał, Przemysław Kazienko, and Tomasz Kajdanowicz. Social Recommendations within the Multimedia Sharing Systems[C]. First World Summit on the Knowledge Society, 2008: 364-372.

# 附录：部分源程序清单

//1. 分词、特征词提取

**publicvoid** Feaword()**throws** Exception

{

List<S\_user> userlist =findusers();

Map<String,Integer> globalmap= **new**HashMap<String,Integer>();//存放全部微博特征词

**int** weibo\_count=0;

**for**(**int** u=0;u<userlist.size();u++)

{

String uid=userlist.get(u).getUid();

List<S\_weibo\_content> weibolist=findByUid(uid);

**if**(weibolist==**null**||weibolist.size()==0)

{

}

**else**

{

weibo\_count=weibo\_count+weibolist.size();

**for**(**int** w=0;w<weibolist.size();w++)

{

S\_weibo\_content sweibo=weibolist.get(w);

//单条微博开始

Map<String,Integer> currentmap = **new** HashMap<String,Integer>();//存放单条微博特征词

SplitWord splitWord = SplitWord.*getInstance*();

splitWord.init();

**int** count=0;

String str=sweibo.getContent();

str=str.replaceAll(" ", "");

String result=**null**;

result=splitWord.split(str);

//System.out.println(result);

String[] rr=result.split(" ");

**for**(**int** i=0;i<rr.length;i++)

{

String[] rr1=rr[i].split("/");

**if**(rr1[1].equals("n"))

{

String noun=rr1[0];

count++;

**if**(currentmap.get(noun)==**null**)

{

currentmap.put(noun, **new** Integer(1));

}

**else**

{

Integer number = currentmap.get(noun);

number++;

currentmap.put(noun, **new**Integer(number));

}

}

}

Iterator iter = currentmap.entrySet().iterator();

**while** (iter.hasNext())

{

Map.Entry<String,Integer> entry = (Map.Entry<String,Integer>) iter.next();

String key=entry.getKey();

**if**(globalmap.get(key)==**null**)

{

globalmap.put(key, **new** Integer(1));

}

**else**

{

Integer number=globalmap.get(key);

number++;

globalmap.put(key, **new** Integer(number));

}

}

}//单条结束

}

System.*out*.println("用户"+uid+"的微博分词结束");

}

System.*out*.println("总微博数weibo\_count="+weibo\_count);

//2. 特征词权重计算

**for**(**int** u=0;u<userlist.size();u++)

{

String uid=userlist.get(u).getUid();

List<S\_weibo\_content> weibolist=findByUid(uid);

**if**(weibolist==**null**||weibolist.size()==0)

{

}

**else**

{

**for**(**int** w=0;w<weibolist.size();w++)

{

S\_weibo\_content sweibo=weibolist.get(w);

Map<String,Integer> currentmap = **new** HashMap<String,Integer>();

SplitWord splitWord = SplitWord.*getInstance*();

splitWord.init();

**int** count=0;

String str=sweibo.getContent();

str=str.replaceAll(" ", "");

String result=**null**;

result=splitWord.split(str);

System.*out*.println(result);

String[] rr=result.split(" ");

**for**(**int** i=0;i<rr.length;i++)

{

String[] rr1=rr[i].split("/");

**if**(rr1[1].equals("n"))

{

String noun=rr1[0];

count++;

**if**(currentmap.get(noun)==**null**)

{

currentmap.put(noun, **new** Integer(1));

}

**else**

{

Integer number = currentmap.get(noun);

number++;

currentmap.put(noun, **new** Integer(number));

}

}

}

Iterator iter2 = currentmap.entrySet().iterator();

**while** (iter2.hasNext())

{

Map.Entry<String,Integer> entry2 = (Map.Entry<String,Integer>) iter2.next();

String key=entry2.getKey();

**double** tf=(**double**)currentmap.get(key)/(**double**)count;

System.*out*.println("count="+count);

System.*out*.println("tf="+tf);

**double** idf=(**double**)weibo\_count/(**double**)globalmap.get(key);

idf=Math.*log10*(idf);

System.*out*.println("globalmap.get(key)="+globalmap.get(key));

System.*out*.println("idf="+idf);

**double** weight=tf\*idf;

S\_feaword ff=**new** S\_feaword();

ff.setSweibo(sweibo);

ff.setFeaword(key);

ff.setWeight(weight);

saveFeaword(ff);

}

}

}

}

//3.用户模型兴趣树的显示

<ul class="easyui-tree">

<s:iterator value="#request['menus']" id="c2">

<s:if test="#c2.parent\_id == 0">

<li data-options="state:'closed'">

<span><s:property value="#c2.name" /></span>

<ul>

<s:iterator value="#request['menus']" id="c3">

<s:if test="#c3.parent\_id == #c2.ids">

<li>

<span><s:property value="#c3.name" /></span>

<ul>

<s:iterator value="#request['menus']" id="c4">

<s:if test="#c4.parent\_id == #c3.ids">

<li>

<span><s:property value="#c4.name" /></span>

<ul>

<s:iterator value="#request['menus']" id="c5">

<s:if test="#c5.parent\_id == #c4.ids">

<li>

<s:property value="#c5.name" />

</li>

</s:if>

</s:iterator>

</ul>

</li>

</s:if>

</s:iterator>

</ul>

</li>

</s:if>

</s:iterator>

</ul>

</li>

</s:if>

</s:iterator>

</ul>

//4. 用户兴趣度计算

public void contentinterest() throws Exception

{

List<S\_user> userlist = findByuid("283358587");

//读取所有用户

if(userlist==null||userlist.size()==0)

{

}

else

{

//读取所有本体类别

List<S\_category> categorylist=ms\_categoryDao.getS\_categoryname();

if(categorylist==null||categorylist.size()==0)

{

}

else

{

for(int u=0;u<userlist.size();u++)

{

//读取某一个用户的所有微博

String uid=userlist.get(u).getUid();

List<S\_weibo\_content> weibolist=findByUid(uid);

if(weibolist==null||weibolist.size()==0)

{

}

else

{

for(int c=0;c<categorylist.size();c++)

{

//判断任意一条微博在每一个分类上的权重，如有则类别兴趣累计

Double wc=(double) 0;

Double ws=(double) 0;

System.out.println(weibolist.get(w).getWeibo\_id()); feaword=findBywid(s\_weibo\_content.getWeibo\_id());//找到微博w的所有特征词记录

List<S\_feaword> feaword=findBywfid(categorylist.get(c).getName(),uid);

if(feaword==null||feaword.size()==0)

{

}

else

{

for(int f=0;f<feaword.size();f++)

{

wc=wc+feaword.get(f).getWeight();

ws=ws+categorylist.get(c).getScd();

}

}

if(wc>0)

{

S\_s\_user\_profile user\_profile= new S\_s\_user\_profile();

user\_profile.setUser(userlist.get(u));

user\_profile.setCategory(categorylist.get(c));

user\_profile.setContentweight(wc);

user\_profile.setSemanticweight(ws);

try

{

s\_UserProfileDao.save(user\_profile);

}catch(Exception e)

{

e.printStackTrace();

}

}

}

}

}

}

}

System.out.println("用户完");

}

//5. 用户兴趣度显示

<table id=*"tt"*class=*"easyui-datagrid"*style="width:*auto*;">

<thead>

<tr>

<thdata-options=*"field:'uid',width:150"*align=*"center"*>uid</th>

<thdata-options=*"field:'s\_category.ids',width:100"*align=*"center"*>category\_id</th>

<thdata-options=*"field:'s\_category.name',width:150"*align=*"center"*>category\_name</th>

<thdata-options=*"field:'contentweight',width:200"*align=*"center"*>content weight</th>

</tr>

</thead>

<s:iteratorvalue=*"#request['plansearch']"*id=*"c2"*>

<tr>

<td><s:propertyvalue=*"#c2.uid"*/></td>

<td><s:propertyvalue=*"#c2.category\_id"*/></td>

<td><s:propertyvalue=*"#c2.category\_name"*/></td>

<td><s:propertyvalue=*"#c2.contentweight"*/></td>

</tr>

</s:iterator>

</table>