密级：_____

浙江大学

# 硕 士 学 位 论 文


论文题目 用于客服辅助的对话模型研究

作者姓名 梁建增

指导教师 王东辉 副教授

学科(专业) 计算机科学与技术

所在学院 计算机学院

提交日期 2017-01

A Dissertation Submitted to Zhejiang University for the Degree of Master of Engineering


# TITLE: Research of Conversation Model for Assistance of Custom Service

Author: Jianzeng Liang

Supervisor: A.P.Donghui Wang

Subject: Computer Science

College: College of Computer Science

Submitted Date: 01/2017

# 摘要

随着互联网经济的不断发展，提供在线商品和服务选购的电商平台的规模和成交量也在日益增大。这种改变的潮流对在线客服的服务质量和服务效率提出更高的要求。因此如何通过计算机技术来辅助人工客服提升其工作效率和工作质量是个值得研究的问题。在此基础上，本文围绕两种客服辅助技术展开：知识库查询服务和客服回复推荐服务，对相关的对话模型进行研究和探索。

针对客服需要参考相关专业知识来完成高质量服务的需求，本文设计了一种知识库查询服务。该服务接受用户的自然语言问句作为输入，通过 AIML 模板匹配技术从输入中提取关键词和待查询属性，并返回知识库中对应的信息条目。传统匹配方式受制于自然语言表达的多样性，存在关键词匹配失效的问题。本文针对这个问题，提出了一种多轮迭代的同义词匹配算法，该算法提升了同义词的检出数量和准确度。

针对如何提升人工客服工作效率的问题，本文提出了用于客服回复推荐的深度对话模型。本文从检索式深度对话模型和产生式对话模型两个方向来解决该问题。在检索式的深度对话模型中，本文设计了一种带上下文建模的对话模型，通过实验对比，其比不带上下文的对话模型有较大性能改善，在此基础上本文使用用户咨询的意图信息对模型进行了改善，获得了部分性能提升。在产生式对话模型中，本文设计了一种使用完整上下文用于预测客服对话的产生式对话模型，并在客服咨询数据集上同传统的 Seq2Seq 模型进行对比，该模型产生的回复效果更好。在产生式模型中，本文先实现了基础的 Seq2Seq 模型，用来作为对照，并根据本文提出的上下文编码方式提出并实现了对应的产生式对话模型。通过实验分析，我们发现本文提出的上下文建模方法对回复的推荐有提升效果。

**关键词：** 客服辅助，对话模型，回复推荐，深度学习

# Abstract

With the development of the Internet economy, electronic business platforms develop fast both on the scale and volume. This trend of developing requires higher efficiency and better quality for online customer service. Therefore, how to assist custom service staffs improve service efficiency and quality with computer technology is a problem worthy of study. On this basis, we focused on two kinds of customer service support technology: query service on knowledge base and custom service response recommendation service, researched and explored the related dialogue models.

We designed a query service on knowledge base, which provides quick access of professional knowledge during working and help custom service staff achieve high-quality service. The service accepts natural language question as input, then extracts the keywords and attributes by using the AIML template matching technique, at last returns the corresponding knowledge. The matching method suffers from a problem of keyword matching failure. We analyzed and altered the semantic-based synonym matching algorithm, proposed a multi-round iterative synonym matching algorithm, which improves the performance of synonyms detection.

To improve the efficiency of the customer service, we use deep conversation model for custom service response recommendation. We tried to solve this problem in two ways: retrieve-based deep conversation model and generation-based conversation model.

On the retrieve-based model, we designed a conversation model with context modeling, which has better performance on response recommendation than non-contextual conversation model. Later we use intention information improved this model, and obtained some performance gains.

On the generation-based model, we implemented a Seq2Seq model, use it as a baseline model. Then we designed and implemented our contextual generation-based conversation model. Through the experimental analysis, we found that our model performs better than baseline model in generation quality.

**Keywords：** Customer Service Assistance, Conversation Model, Response Recommendation, Deep Learning

# 目录

摘要 ..... i
Abstract ..... ii
目录 ..... I
图目录 ..... III
表目录 ..... IV
第 1 章 绪论 ..... 1
1.1 课题背景与意义 ..... 1
1.2 本文的主要工作 ..... 2
1.3 本文的组织结构 ..... 3
1.4 本章小结 ..... 4
第 2 章 相关工作综述 ..... 2
2.1 主要概念及相关研究 ..... 2
2.1.1 聊天机器人 ..... 2
2.1.2 客服辅助系统 ..... 5
2.2 基于模板的对话模型 ..... 8
2.2.1 AIML ..... 8
2.2.2 新词发现 ..... 10
2.3 基于检索匹配的对话模型 ..... 12
2.4 基于深度学习的对话模型 ..... 13
2.4.1 词嵌入 ..... 13
2.4.2 句子的向量表示 ..... 15
2.4.3 基于产生式的深度对话模型 ..... 17
2.4.4 基于检索的深度对话模型 ..... 21
2.5 本章小结 ..... 24
第 3 章 基于模板的客服辅助技术 ..... 26
3.1 应用场景和面临的问题 ..... 26
3.2 领域关键词扩充算法 ..... 29
3.2.1 数据预处理 ..... 29
3.2.2 传统扩充算法 ..... 32
3.2.3 多轮扩充算法 ..... 34
3.2.4 算法性能评估 ..... 36
3.3 知识检索服务设计 ..... 37
3.4 本章小结 ..... 39
第 4 章 深度对话模型用于客服回复推荐 ..... 41
4.1 基于检索的深度对话模型研究 ..... 42
4.1.1 无上下文的检索对话模型 ..... 42

4.1.2 带上下文的检索对话模型 ..... 44
4.1.3 使用意图信息的改进 ..... 48
4.1.4 实验结果和分析 ..... 49
4.2 基于产生式的深度对话模型研究 ..... 55
4.2.1 无上下文的产生式会话模型 ..... 55
4.2.2 带上下文的产生式会话模型 ..... 56
4.2.3 实验结果和分析 ..... 57
4.3 本章小结 ..... 60
第 5 章 总结和展望 ..... 61
5.1 总结 ..... 61
5.2 展望 ..... 62
参考文献 ..... 64
攻读硕士学位期间主要的研究成果 ..... 67
致谢 ..... 68

# 图目录

图 2.1 基于知识库匹配的客服支持系统基本框架..... 7
图 2.2 基于匹配的基本实现示意图..... 7
图 2.3 Skip-gram 模型示意图..... 14
图 2.4 普通的神经网络应用于句子生成向量表达..... 15
图 2.5 循环神经网络示意图..... 16
图 2.6 循环神经网络按时间片展开..... 16
图 2.7 Encoder-Decoder 框架示意图 ..... 18
图 2.8 层次化的 RNN 模型用来建模的对话中的上下文[21]..... 21
图 2.9 深度排序模型的抽象框架..... 23
图 3.1 数据预处理流程..... 30
图 3.2 同义词的聚集效应示意图..... 35
图 3.3 手工模板示例..... 38
图 4.1 QA-LSTM 模型..... 44
图 4.2 Utterance Encoder 结构示意图..... 47
图 4.3 Context Encoder 的编码过程示意图..... 48
图 4.4 QA-LSTM 验证集测试记录..... 51
图 4.5 HRR 验证集测试记录 ..... 51
图 4.6 HRR+Intention 验证集测试记录..... 52
图 4.7 Seq2Seq 模型结构示意图..... 56
图 4.8 带上下文的产生式会话模型结构..... 57

# 表目录

表 2.1 AIML<category>标签实例 ..... 8
表 2.2 AIML<set>标签使用实例 ..... 9
表 2.3 一条上下文敏感的规则 ..... 10
表 2.4 新词发现处理流程 ..... 12
表 2.5 One-hot Representation 示例 ..... 13
表 3.1 新词发现算法阈值设定 ..... 31
表 3.2 算法流程 ..... 36
表 3.3 同义词数量对比 ..... 37
表 3.4 多轮扩充算法示例 ..... 37
表 3.5 目前支持查询的属性 ..... 39
表 3.6 部分查询示例 ..... 39
表 4.1 对话示例 ..... 45
表 4.2 检索式对话模型实验结果 ..... 50
表 4.3 回复推荐测试实例-1 ..... 53
表 4.4 回复推荐测试实例-2 ..... 54
表 4.5 产生式对话模型实验结果 ..... 57
表 4.6 回复推荐测试实例-3 ..... 58
表 4.7 回复推荐测试实例-4 ..... 59

# 第1章 绪论

## 1.1 课题背景与意义

随着互联网经济的不断发展，在线购买商品和服务已经逐渐成为人们的生活习惯，这种潮流为许多行业及工作带来改变，其中包括在线客服。电商平台的成交量日益增多，对在线客服的服务质量和服务效率不断提出新的要求。与此同时部分传统行业的互联网化，产生了一些对专业知识要求较高的客服场景，如在线健康咨询、在线法律咨询，这些场景的客服人工成本较高。这些因素使得如何在现有基础上提升客服的服务质量和服务效率成为一个迫切需要解决的问题。目前已有的解决思路有两种，一种是使用智能客服机器人来完全替换人工客服，另一种是使用计算机来辅助人工客服提升其工作效率和质量。

智能客服机器人通常以大量手工构建的知识库作为核心[1]。其不足之处主要有：1）机器人通常只能处理常见的标准问题，对于一些个性化较强或出现频次较低的问题则较难处理；2）复杂知识库的建立需要极大成本；3）用户对机器人的接受程度不如人工客服。

客服辅助系统（使用计算机辅助人工客服的系统），可以完成一些不直接和用户交互的任务如：标准回复推荐、常用问答对推荐。现有的客服辅助系统主要针对所需专业知识相对较少的场景设计，如交易纠纷、商品信息咨询等，但是对于一些专业性要求较高的客服场景如法律咨询、健康咨询则很少涉及。

因为现有的客服机器人和客服辅助系统存在缺点和不足，本文对实现客服辅助系统所需的对话模型进行了研究。提出了两种客服辅助服务：知识库查询和客服回复推荐，分别用以提升人工客服的服务质量和服务效率，且均可用于对专业知识要求较高的客服场景。

其中知识库查询服务，提供一种改进的自然语言查询知识库的服务，通过给客服人员在工作过程中便捷地提供专业知识作为参考，提升人工客服的服务质量。

回复推荐服务，根据历史对话自动学习一个模型，能够在客服工作过程中，

根据对话历史推荐客服可能需要的输入，从而减少客服输入量，提升工作效率。该模型不同于通常使用的对话模型的地方是，其无需预先构建高质量的知识库或者 FAQ 数据集，而是直接根据客服的服务历史对话记录来训练模型。

## 1.2 本文的主要工作

本文根据客服需求的不同主要研究两种客服辅助服务：对于客服需要参考专业知识来提供高质量服务的需求，本文提出了一种改进的知识库查询服务；对于提升客服人员工作效率的需求，本文提出了一种回复推荐服务。

在本文提出的知识库查询服务中，系统接收用户的自然语言提问作为输入，将用户的自然语言提问转换为{知识库实体关键词，待查询属性}二元组，并通过检索知识库中对应的条目信息，将答案返回给客服或用户作为参考。由于自然语言表达中存在较强的不确定性，如何从用户提问中解析出正确的实体名是该任务的难点。传统的思路包括：通过字符串相似进行匹配、使用语义向量进行匹配以及通过其他手工特征进行匹配[9]，文本考虑了向量空间中同义词的聚类特性，提出一种迭代的构建同义词集的方法，用来查找和构建知识库中高频实体的同义词集，进而提升知识库检索效果。

在回复推荐服务中，系统读入客服和用户的对话历史，并在每次用户输入之后，预测客服下一句可能的回复。我们就深度会话模型展开研究，分别尝试使用检索式对话模型和产生式对话模型来完成回复推荐。在检索式模型中，我们对比了三种上下文建模方式的效果：1）前人提出的无上下文的建模方式；2）本文提出的考虑上下文的建模方式；3）本文提出的综合考虑上下文和用户咨询意图的建模方式。在产生式模型中，本文对比了两种模型的效果：1）前人提出的不考虑上下文的标准 Seq2Seq 模型；2）本文提出的考虑上下文的产生式模型。

现将本文的主要工作归纳如下：

1）在知识库检索服务中，针对使用自然语言查找知识库实体过程中出现的匹配失效问题，本文提出了一种同义词多轮扩充算法。该算法能够提升知识库中高频实体同义词的检出数量，从而帮助提升使用自然语言检索知识库的检索效果。

2）在回复推荐服务中，本文提出了一种对对话历史上下文进行建模的方法，将其用于产生式回复推荐模型和检索式回复推荐模型中。并通过实验对其推荐效果进行验证评估。通过分析实验结果，本文针对性的对该模型进行了改进，改进之后的方法通过综合用户的咨询意图和对话历史信息来推荐回复。

## 1.3 本文的组织结构

本文组织结构如下：

第一章介绍了本课题的研究背景、研究意义以及本文的主要工作。简要介绍了客服辅助服务的应用背景和发展趋势，以及现有模型的不足，在此基础上提出本文的研究目标和预期应用场景。进而简要介绍了本文提出的两个客服辅助服务的基本框架。

第二章介绍了用于客服辅助的会话模型的相关工作综述。从会话模型和客服辅助的概念开始，进而围绕会话模型展开。首先介绍了基于模板匹配的会话模型中用到的一些技术，包括：AIML 匹配技术、新词发现，其次介绍了通过检索方法实现会话模型的基本思路，最后对基于深度学习的会话模型及相关技术进行了介绍分析。

第三章介绍了本文在知识库检索服务方面的工作。先介绍了本文提出的知识库检索服务的应用场景和相关概念，其后对本文提出的同义词扩充算法进行了介绍，包括传统实现思路和本文提出的多轮扩充算法。最后介绍了知识库检索服务的整体实现框架。

第四章介绍了本文在回复推荐方面所做的工作。先介绍了回复推荐任务的定义。之后介绍本文用来实现回复推荐的两种思路：检索式对话模型和产生式对话模型。在两种思路中，我们围绕对话模型的上下文建模部分展开研究，实现了前人提出的无上下文的对话模型，之后设计并实现了考虑上下文的会话模型，在对实验结果进行分析的基础上，对考虑上下文的检索是对话模型进了改进。

第五章对本文的工作进行了总结，并分析了用于回复推荐的会话模型当前存在的问题和不足，以及未来可能改进的地方。

## 1.4 本章小结

本章介绍用于提升客服工作效率和质量的知识库检索服务和客服回复推荐服务的研究背景和研究意义。同时对本文的主要研究内容、主要贡献和后续章节的组织进行了介绍。

# 第2章 相关工作综述

随着互联网的不断发展，人们的生活正在被逐渐改变已成为共识，但被改变的不仅仅是人们的生活习惯，还包括一些相关的行业和工作，如在线客服。互联网的兴起对在线客服的服务质量和服务效率提出了更高的要求。一个切实可行的解决方案是通过计算机来辅助人工客服，以此来提升客服的服务质量和服务效率，并减少人力成本。

客服辅助系统的实现通常使用聊天机器人中的对话技术，且聊天机器人的应用场景同客服辅助系统存在诸多相似之处，可以相互借鉴。目前随着深度学习在自然语言领域的推广应用，对话模型也开始得到研究者越来越多的关注。

本章从聊天机器人和客服辅助系统的基础概念和相关技术开始谈起。详细介绍了传统会话模型的特点和不足，并综述了当前前沿相关研究的现状。

## 2.1 主要概念及相关研究

### 2.1.1 聊天机器人

聊天机器人[2](Chatterbot，对话机器人)是通过语音和文本的方式同人类进行交互的一类计算机程序。其发展历史悠久，早期通常被设计用来模拟人类在对话过程中的行为，以尝试通过图灵测试。图灵测试[3]是由 Alan Turing 于 1950 年提出的用来判断机器是否具有智能的测试，其形式为由一个人类作为裁判，通过文字的形式分别与待测试机器和一个正常人类进行交流，如果裁判无法通过文字交流区别出机器和人类，那么可以认定该机器通过了图灵测试。

最早的聊天机器人可以追溯到 1966 年，Joseph Weizenbaum 发布的 ELIZA[4]，作为聊天机器人的开山鼻祖，该系统使用正则表达式对用户的输入进行匹配，如果能够匹配到合适的规则，则对匹配到的结果进行一系列基于规则的改写，并将最终结果以文本的形式返回给用户。因为是基于正则匹配的实现，ELIZA 其实并不具备复杂的知识系统，但是在条件较为宽松的测试条件下，会让用户觉得其拥

有智能。

ELIZA 的设计虽然简单，出现时间也早，但是其基于模式匹配的设计思想，对后来的聊天机器人设计发展有着深刻的影响。

后来，随着对话机器人渐渐吸引了人们的关注，又有一些比较经典的对话机器人出现，例如 PARRY[5]（1972 年）、JABBERWACKY[6]（1988 年）、DR.SBAITSO[7]（1992 年）等，它们均在前人的基础上有一些发展，比如尝试增加语音控制和输出。

直到 1995 年，一款经典的对话机器人 A.L.I.C.E[8](Artificial Linguistic Internet Computer Entity)被发布。其嵌入了 AIML (Artificial Intelligence Markup Language)语法规则，并结合一系列启发式的规则重写了后台引擎，使得对话质量相比之前的对话机器人得到大幅改善。A.L.I.C.E Bot 一举夺得了 2001、2002 和 2004 年的 Loebner Prize，获得了大量关注。对 AIML 语法的支持，使得人们可以轻易编写自己的对话规则，降低了创建对话机器人的门槛和成本。但是 A.L.I.C.E 本身仍然是一个基于模板匹配的系统，其表现非常依赖于模板的数量和质量，和人类交谈的过程当中也非常容易露出破绽，距离图灵测试仍然很遥远。

由于对话机器人的构建需要大量的人力工作，产生的回复质量不高，同时也并没有一个理想的应用场景，之后十年对话机器人并没有得到太多的关注[2]，虽然期间也有优秀的框架出现如 ChatScript，但总体处于相对沉默期。

直到最近几年，随着深度学习横扫人工智能研究领域，在自然语言处理的诸多任务上带来明显的性能提升，同时互联网行业巨头的投资和关注，使得聊天机器人得到了巨大的关注。

**开放域和封闭域聊天机器人**

从应用的场景来看，对话机器人可以分类开放域（Open-Domain）和封闭域（Closed-Doman）两大类。

开放域聊天机器人，顾名思义对聊天机器人交互的主题和内容并不做限制，用户可以在指定形式下（通常是文本或者语音）同聊天机器人进行任何内容的自由对话。开放域的聊天机器人难度很大，构建的复杂度通常也更高，需要准备大

量的知识库作为基础，模型通常也较为复杂。但和原始的图灵测试问题更为接近，都是为了模拟对话过程中的人类角色。从实际应用的角度来看，开放域聊天机器人一般用于聊天、虚拟角色等娱乐领域，主要目的在于吸引用户的关注，比如我们熟悉的小黄鸡、微软小冰等。一个优秀的开放域聊天机器人容易传播，可以快速吸引到大量用户。开放域的聊天机器人经常面临对话目的性不强、内容无意义、对话质量低等问题，且其应用场景有限，故商业价值较低。

封闭域聊天机器人通常有明确的目标和受限的主题或知识范围。封闭域聊天机器人所面对的输入相比开放域聊天机器人较为有限。期待应对的问题空间大幅缩小，使得其可以在相对较小的主题上构建更为稠密和准确的对话规则和知识库，从而获得更高质量的回复。与此同时，封闭域的对话机器人可以应用到各个垂直领域，应用场景具有明确的目的性，因此用户对封闭域对话机器人能否回复通用类问题的期望也不高。封闭域聊天机器人具有更高的商业价值，因此对话的可控程度要求更高，对对话中出现的错误的容忍度更低，所以封闭域的聊天机器人往往采用知识图谱作为其知识核心。

**单轮和多轮聊天器人**

从同用户交互过程中聊天机器人是否具有上下文的概念进行分类，聊天机器人可以分为单轮对话和多轮对话。

单轮对话的聊天机器人，没有上下文概念，其表现形式类似于传统的自然语言处理中的问答（Question-Answering）任务。通常通过预先定义好的业务知识或者整理得到的 FAQ（Frequently Asked Question），在运行过程中对用户的提问的进行解析或者匹配，找到合适的回复，对其进行处理后返回给用户。

多轮对话的聊天机器人有更大的发挥空间，其不仅可以利用上下文内容对用户的状态和需求进行整合，甚至可以在用户信息缺失的情况下，选取合适的问题主动向用户提问来尝试补全缺失信息。相比单轮对话，多轮对话面临更多技术挑战，如：意图识别、属性提取、上下文控制等。

**聊天机器人实现技术分类**

从实现聊天机器人所用技术的角度进行分类，聊天机器人可以分为基于模板

匹配、基于检索式、基于产生式三种类型。

基于模板匹配的实现，如 A.L.I.C.E，ChatScript 等，通过正则表达式或特定语法规则如 AIML（Artificial Intelligence Markup Language）来匹配用户的输入，如果用户的输入能够命中某条模板或者规则，则将对应的回复返回给用户。

基于检索式的实现，通常使用分词、关键词提取、倒排索引的方式将机器人可以回答的问题及其标准答复存储起来。在运行过程中，根据用户历史信息和提问使用关键词或属性命中等方式检索所有相似的问答对，并使用排序算法对命中的结果进行排序，并将最终选定的答案做适当处理后返回给用户。

基于产生式的实现，不依赖于任何特定的知识库或答案库，利用产生式模型，从大量语料中学习人类对话的模式。模型可以对用户的任意问题，都产生出一个自然语言回复。但是回复内容的质量和其合理性则较难控制，是目前研究当中的难点。

### 2.1.2 客服辅助系统

随着电商平台的不断拓展，其对在线客服的服务质量和服务效率的要求也越来越高。根据提供服务的来源不同，本文将客服服务分为人工客服、客服机器人和客服辅助系统三种类型。

人工客服,通过对从业人员进行业务培训使，使之成为合格客服人员。人工客服按照工作安排，在线实时地接待用户的咨询，并完成相关处理。

人工客服的优点在于：1）人工客服能处理非常复杂的咨询问题;2）真实人类带给用户的体验会比机器人好。其缺点也非常明显:1）非常高的人工成本和培训时间成本,对于专业性较高的领域的客服来讲尤甚;2）人工客服的服务承接能力的可拓展性非常差，承接数量的提升依赖于客服数量和客服经验的增长。

客服机器人，通过预先定义好的知识库或者规则系统作为引擎完成客服工作的一类机器人。其通过与用户进行若干次特定形式的交互，收集用户的基本信息，通过这些信息将用户的咨询匹配到合适的知识或者规则，进而根据该规则完成咨询服务。

优点在于：1）用机器人替代人工的成本很低；2）咨询的流程和服务质量高度可控；3）使用机器人咨询承接能力的拓展性好。其缺点也比较明显：1）大规模知识库的生成需要专业人员参与整理，成本较高；2）通常知识库和规则只能覆盖一些常见的高频问题，对于长尾的咨询和个性化的情况则不能很好的处理；3）用户对于机器人的接受程度不如人工客服。

客服辅助系统，是在人工客服和客服机器人之间的一种折衷办法。其通过计算机为人工客服在服务过程中提供帮助。常见的功能包括收集并展示用户的基本信息、对特定类型的问题提供标准化的回复、为客服推荐回复以减少客服的输入量等。

客服辅助系统和客服机器人之间并没有一个明确的界限。相反经常可以在客服机器人产品中看到全自动模式（无客服人员参与）与半自动模式（推荐回复内容，由人工客服决定是使用），两种模式使用的实现技术相同，给出的回复内容并无太大区别。因为本文主要关注于如何给客服人员提供专业知识参考和推荐合适回复内容，以提高其工作效率和质量，减少其工作量。因此在本文的后续章节中，我们在介绍和阐述相关技术时均围绕客服辅助系统进行。

目前客服支持系统主要有两种实现思路。

一种是通过问句识别、属性提取、语义约束等自然语言处理方法从用户对话中抽取相关属性，构建一个表示用户咨询意图的查询语句，在预定义的知识库中查找和匹配对应知识，图 2.1 所示为基于的知识库匹配的客服支持系统的基本实现框架。

另一种是通过数据挖掘的方法构建一批常见问答对（frequent asked questions, FAQ），通过机器学习的方法学习一个匹配模型，当用户有新的提问进入时，检索 FAQ 数据中是否有相似的问答对，如果存在合适的匹配，则将对应的 QA-Pair 返回给用户作为参考。

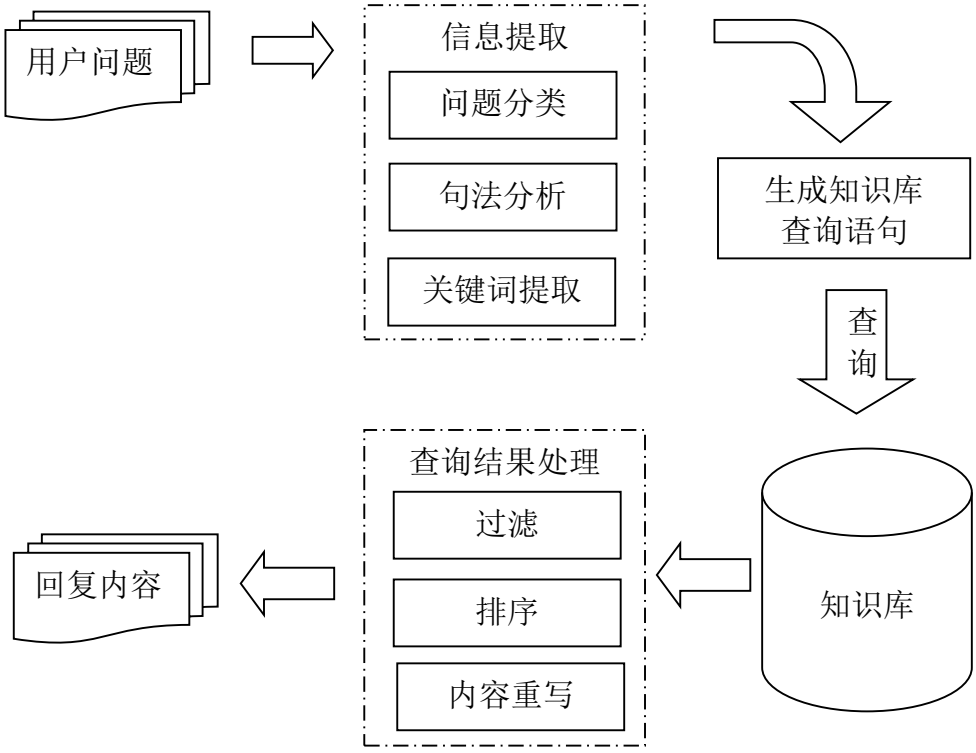



图 2.1 基于知识库匹配的客服支持系统基本框架

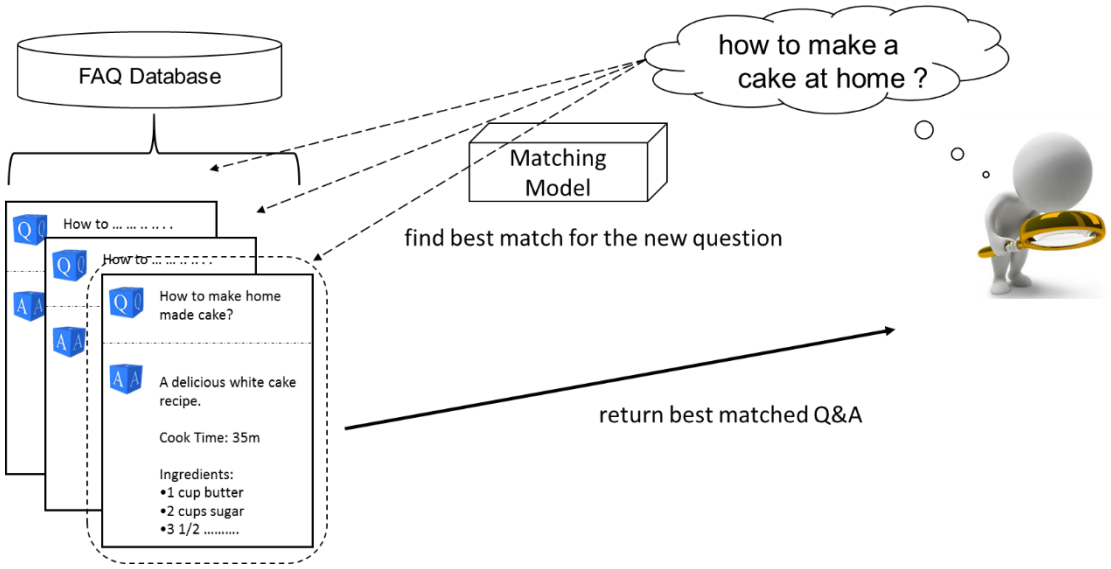



图 2.2 基于匹配的基本实现示意图

## 2.2 基于模板的对话模型

### 2.2.1 AIML

AIML[10](Artificial Intelligence Markup Language)，是一种用来编写聊天机器人对话规则的 XML 语法。AIML 是基于“刺激-反应”原理，用户所编写的每条规则或知识都可以视为一个“刺激-反应”的实例。每当用户同聊天机器人进行一次对话，均可视为对其的一次刺激，机器人会在规则库中查找是否有命中的规则，如果有则将对应结果返回给用户。

虽然 AIML 支持用户对上下文进行处理，但其处理过程较为复杂，往往需要用户额外编写代码逻辑才能实现对上下文信息的个性化处理。因此 AIML 的主要应用场景还是在不考虑上下文的对话情景。

AIML 中最常见的基本单元是<category>标签，每个<category>标签均是一个规则实例。<category>标签中一般会包含<pattern>和<template>标签。<pattern>标签描述可以匹配的用户输入，其中允许使用 AIML 通配符来扩大匹配的范围。<template>标签描述了当命中这条规则时，机器人应该如何回复用户。

表 2.1 中所示为，一条简单的 AIML 规则实例，当用户提问“你好！最近怎么样”时，聊天机器人会根据这条规则回复“还不错啊，你呢？”。

表 2.1 AIML<category>标签实例

```
<category>
    <pattern> 你好！最近怎么样 </pattern>
    <template> 还不错啊，你呢？ </template>
</category>
```

为了提高模板的匹配效率，AIML 支持一种集合词标签<set>，用户可以在模板外部自定义一些词集，并在<pattern>标签中引用该词集，当用户的输入命中词集中的任何一个词时，都可以视为对模板中该<set>标签的一次命中。这种语法使

得模板的通用性大大提升，尤其在垂直领域，开发者甚至可以直接利用该语法成简单的关键词提取功能。

表 2.2 中所示为<set>标签的使用示例，用户在外部定义一个词的集合“drug”，当用户提到其中的任何一个词，即可视为命中了该模板中的<set>drug</set>部分。

表 2.2 AIML<set>标签使用实例

```
<category>
    <pattern> 你知道 <set>drug</set> 该怎么用吗？</pattern>
    <template> 这种问题还是找医生咨询一下比较保险 </template>
</category>
```

AIML 中对上下文处理的支持主要通过三种标签来实现：<topic> 、<that>、<set>（与前面的<set>标签重名，但功能不同）。

<topic>标签允许开发者对聊天机器人的对话规则进行分组，将相同主题的规则置于一个<topic>标签下，在匹配时通过指定 topic 参数，实现只使用特定主题的规则进行匹配回复，使得回复内容和上下文的主题一致。

<that>标签用来指代机器人最近一次说的话，机器人可以使用该标签来实现对同样的提问产生不同回复。

<set>标签可以用来设置一些自定义变量，这些变量的作用域为整个对话过程，并且允许 AIML 标签或者后台代码对其进行操作，通过该语法可以实现一定程度的上下文操作。

如表 2.3 所示为，一条对上下文敏感的规则。在这条规则中，当机器人的上一句话是“你喜欢《星际穿越》吗？”，且用户的回答为“喜欢”时，这条规则才会被匹配到。接着机器人会向用户回复“我也是，诺兰的电影我都喜欢”，同时将变量 like 的值设为 true，从而在后续的交互过程和处理过程中，可以根据需要获取用户是否喜欢《星际穿越》的信息。

AIML 中支持一种递归标签<srai>，其功能是在匹配过程中，可以将该标签中的内容作为用户的输入送给匹配模块，使其对新的输入再走一次“刺激-反应”的

流程，相当于一次输入，多次匹配。用<srai>标签可以实现一些有有趣的功能：

- 将复杂的匹配表达式拆分成多个简单表达式的组合
- 对不同的用户表达进行归一化，给出相同的回复
- 常见的拼写错误、语法错误的纠错
- 分段信息提取，从用户的输入中提取属性

值得一提的是，AIML 作为一种语法规范，其不仅仅在 A.L.I.C.E 中有实现，还在多个语言中，如 Java、Python、C++等均有对应实现，对构建对话机器人开发者来说使用非常方便。

表 2.3 一条上下文敏感的规则

```
<category>
    <pattern> 喜欢 </pattern>
    <that> 你喜欢《星际穿越》吗？ </that>
    <template>
        <set name='like'> true </set>
        我也是，诺兰的电影我都喜欢
    </template>
</category>
```

### 2.2.2 新词发现

在进行中文自然语言处理任务时，一个非常基础且重要的步骤是进行中文分词，虽然目前有许多开源的优秀分词算法可以使用，但是未登录词（新词）的存在使得分词算法在一些分词情景中的表现不太好，从而对后续的任务如：属性提取，关键词检测等都会造成负面影响。

未登录词的产生原因主要由三种。

1）人们在表达过程中使用的词汇并不是一成不变的，相反爆炸性新闻的出现，流行表达的传播都会不断制造新的流行词汇。

2）人们经常在网络上通常通过输入文本进行交流，在这个过程中，也产生了

大量错误拼写和同义表达，这些错别字或词一般没收录在词库里，对于一些出现频次比较高的错误的表达，所以也需要进行区分。

3）垂直领域的专业词汇，由于日常生活中使用较少，分词器词库也很少有收录。

在待处理的语料上进行新词发现，能够帮助我们进行更为精确的文本语义处理和分析。目前新词发现任务中经常使用凝固度、自由度、新词频次三个标准来判断一个候选表达是否是一个新词。

凝固度的基本思想是：对于一个正常的词，其在语料中出现的概率，要远远大于其各个组成元素出现概率的乘积。也就是给出了一个词的前半部分或者后半部分，我们很容易预测出这个词完整的内容是什么。例如,在语料中

“电视”出现的概率为$P_1$

“机”在语料中出现的概率为$P_2$

“电视机”在语料中出现的概率为$P_3$

则有$P_3 \gg P_1 \times P_2$

自由度则是指如果一个文本片段可以作为一个词，则它应该可以出现在各种不同的上下文中，而非有限的几个组合。即正常的词应当具有非常丰富的左邻字和右邻字。在计算过程中，一般使用候选表达的左邻集和右邻集对应的信息熵作为评价标准，信息熵越大，说明该候选词在使用过程中灵活度越大，越有可能是一个词。

新词频次是指候选表达在语料库中出现的频次，出现次数越高表明人们对这个词的接受程度越高。

对语料库进行新词发现的常用流程如表 2.4 所示。因为新词发现的算法性能在实际使用中效果不错，在简单的阈值筛选之后再加一次人工过滤，便可快速获得该语料库中的新词。通过调用对用分词的 API，便可将获得的新词加入分词器词库中，使得在后续分词过程获得更为准确的分词结果。

表 2.4 新词发现处理流程

| |
|---|
| 1. 设定候选词最长长度：L |
| 2. 枚举语料库中的所有长度为$L+1$后缀串，按字典序对候选串排序 |
| 3. 对于每个候选串用滑动窗口选出长度$2-L$子串（最左侧的字不选取），作为候选词 |
| 4. 计算所有候选词的正向自由度（左邻集信息熵） |
| 5. 倒序排列整个语料库，重新运行 2~4 步，计算逆向自由度（右邻集信息熵） |
| 6. 计算所有候选词的凝固度 |
| 7. 统计候选词磁盘 |
| 8. 通过对正向自由度、逆向自由度、凝固度和词频选取合适的阈值进行初选 |
| 9. 人工筛选过滤，选取并添加新词 |

## 2.3 基于检索匹配的对话模型

检索和匹配是实现聊天机器人主要方法之一。相比基于模板匹配的实现方式，基于检索的实现方式不需要手工从头构建大量模板或知识库，成本更低。就返回的内容质量而言，检索式模型比产生式模型更好，因为检索的语料一般由垂直领域从业人员或普通人的聊天记录中整理得到。

但基于检索和匹配的实现不足之处在于，对话的可控程度相比使用模板实现的机器人较低。我们只能通过检索模型本身的性能预估，机器人回复的内容同用户的提问存在一定相关性，但具体回复什么样的内容则需要通过模型测试之后才能给出结论。所以基于检索匹配的实现方式可靠性不如基于模板的实现方式。

构建检索式的聊天机器人，首先需要一批问题-答案对语料作为构建基础。对于开放域的聊天机器人，这些问答对可以来自于社交软件或开放网络社区里的聊天记录、影视字幕、剧本等。开放域的对话机器人并不要求语料中保持一问一答的形式。而封闭域的对话机器人则相反，因其主要目的在于辅助或替代人工来服务用户，所以一般使用特别整理出的 FAQ、垂直问答社区的问答记录作为语料。

在构建过程中，需要对语料进行分词，提取关键词，并对问题和答案建立索引，在实际运行时，将用户的提问或输入同样进行分词和关键词提取处理，然后从索引中查找相似的问题，并用排序算法对这些问题按照相关度进行排序，最终将最相关的问题所对应的答案作为聊天机器人回复，返回给用户。

## 2.4 基于深度学习的对话模型

随着深度学习相关理论和技术在自然语言处理领域得到大量应用，许多传统任务的性能都得到了提升，对话模型也不例外。本节主要介绍深度学习应用于对话模型中时涉及的一些相关模型和技术。按照模型和技术的应用层次从下至上分别为：词的向量表示，句子的向量表示，产生式深度对话模型和检索式深度对话模型。

### 2.4.1 词嵌入

词嵌入（Word Embedding）将离散的文本符号映射到连续的向量空间中，通常用作机器学习算法的输入。优秀的 Word Embedding 具有良好的语义特性，可以大幅提高相关任务的得分。

在很多 NLP（自然语言处理）任务中，第一步都需要将文本符号转换为机器学习模型可以接受的数学符号，也就是向量或数值作为后续算法和处理的输入。传统最为直观和简单的数学表达是 One-hot Representation，这种方法的基本思想是将每个词均表示成为一个字典长度的向量，该向量绝大部分分量均为 0，只有该单词所对应的一个分量为 1。表 2.5 所示，为 One-hot Representation 的一个简单示例。

表 2.5 One-hot Representation 示例

| |
|---|
| 假定字典中总有 5 个词：{a, b, c, d, e} |
| ‘a’可以表示为:[1, 0, 0, 0, 0] |
| ‘d’可以表示为:[0, 0, 0, 1, 0] |

这种表示方法存在诸多缺点：向量长度过长，训练中会造成维数灾难，向量本身不包含语义知识。于是从上世纪 90 年代起，一些基于连续向量空间的词的表示方法被提出，包括：LSA（Latent Semantic Analysis：隐含语义分析），LDA（Latent Dirichlet Allocation：隐狄利克雷分布）等。Bengio 等人 2001 年[11]在提出使用一个三层的神经网络来构建语言模型的方法，发表在当年的 NIPS。该方法尝试在通过模型训练语言模型的同时学习单词分布式表达。

真正让 Word Embedding 流行起来成为 NLP（自然语言处理）任务中必备组件的，还是 Mikolov 等人在 2013 年发表通过无监督的方式训练的 word2vec[12]。word2vec 包含两个模型，CBOW 和 Skip-gram，在本文的相关实验中，我们使用通过 Skip-gram 训练的词向量，故在这里只介绍 Skip-gram 模型。Skip-gram 的主要思想是：意思相近的词具有相似的上下文。例如：“私家车”同“越野车”的语义相近，同“火箭”的语义相差较大，所以经常出现“私家车”上下文中的单词类似于“驾驶”、“排量”等也经常会出现在“越野车”的上下文中，但是不会出现在“火箭”的上下文中。

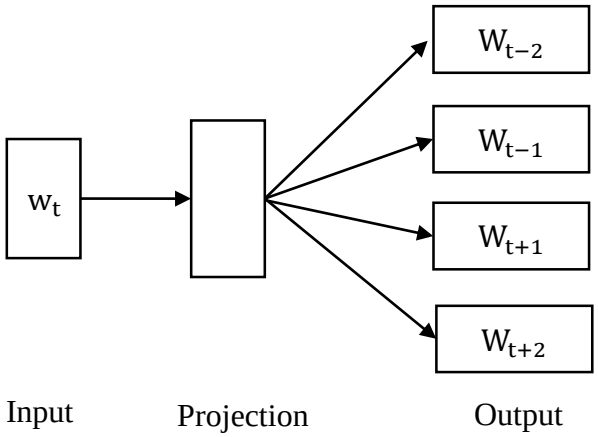



图 2.3 Skip-gram 模型示意图

图 2.3 所示为 skip-gram 模型的示意图，其优化目标函数如公式(2.1)和公式(2.2)：

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T}\sum_{-1\le j\le n,\neq 0}\log p(w_{t+j}|w_t) \qquad \text{公式(2.1)}$$

$$P(w_O|w_I) = \frac{\exp(V_{w_t}^T V_{w_{t+j}})}{\sum_{w=1}^{V} \exp(V_w^T V_{w_t})} \quad \text{公式(2.2)}$$

输入和输出分别为对应单词的词向量，通过单词的 one-hot 编码来抽取词向量矩阵中的对应列作为该单词的词向量。模型通过使用 softmax 函数来约束中心词的词向量使其与上下文中出现的单词的词向量尽可能相近。

### 2.4.2 句子的向量表示

聊天机器人接受的输入通常是用户输入的自然语言文本，其有一个特点：输入的长度不确定，即句子的长度不一定。如图 2.4 中所示，普通神经网络只能接受固定长度的输入，并不适合用做句子的建模。

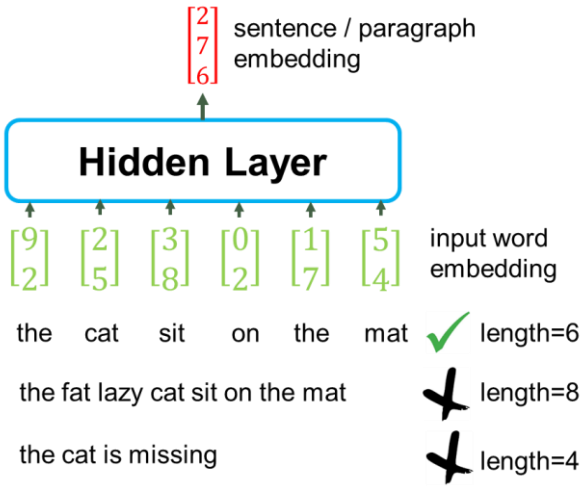



图 2.4 普通的神经网络应用于句子生成向量表达

因此 NLP 任务中经常使用循环神经网络对句子进行建模。循环神经网络是一种特殊的神经网络结构，在每一时刻读入一个新输入，并将其和上一时刻的输出整合成为完整输入，将最后时刻的输出作为整个神经网络的输出。

如图 2.5 中所示，我们可以将其应用至生成自然语言的向量化表达当中，如图中所示，循环神经网络t时刻的输入由两部分组成，一部分是t时刻新输入的词的词向量，另外一部分是$t-1$时刻网络内部记忆节点保存的（或者输出的）向量表达，神经网络在最后时刻的状态或输出作为网络的最终输出。图中的神经网络是由一个循环神经网络和附加的全连接层组成。

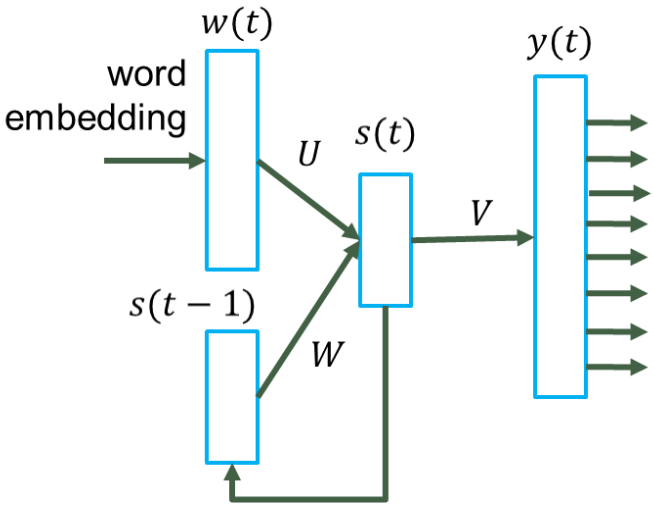



图 2.5 循环神经网络示意图

在训练过程中，通常将循环神经网络按照时间片段展开，生成一个便于计算的结构，如图 2.6 所示

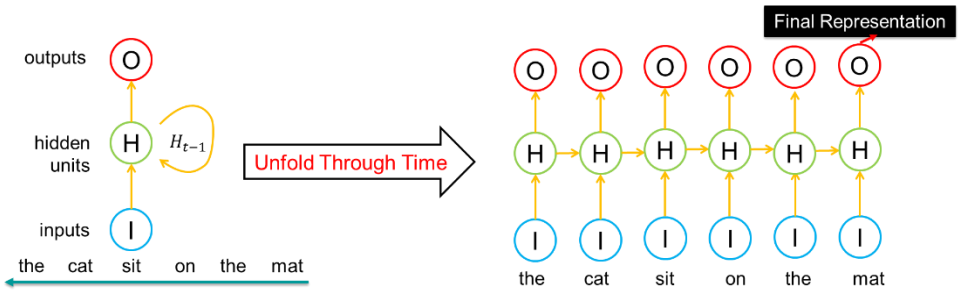



图 2.6 循环神经网络按时间片展开

循环神经网络的训练也通常使用反向传播算法来实现，在这种特殊的网络结构中的反向传播过程如图所示。图中各层神经网络共享同一组参数，灰色的节点代表在反向传递过程中这部分没有误差传递下来。由于各层共享同一组参数，所以在参数更新的时候，需要先将各层对参数的修改进行累加，等待所有计算完成之后，在该组参数上做一次参数更新。

单纯使用 RNN 做句子的建模，在反向传递时，每个时间刻都会之后都会有梯度的衰减或放大，因此在训练过程中非常容易出现梯度爆炸和梯度消失的问题，导致模型难以训练。一个解决这个问题的办法是在 RNN 中使用特殊的隐层单元，如 LSTM，GRU。

使用 LSTM 单元[13]作为循环神经网络的基础单元。LSTM 具有一些较为良好的特性，能够捕捉词序列中的长期和短期依赖关系，从而能够根据句子生成较好的向量表达，其性能超过标准结构的循环神经网络。

LSTM 更新方式如公式(2.3)至公式(2.8)所示。

$$\mathrm{f_t} = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{公式(2.3)}$$

$$\mathrm{i_t} = \delta(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{公式(2.4)}$$

$$\mathrm{o_t} = \delta(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{公式(2.5)}$$

$$\widetilde{C_t} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad \text{公式(2.6)}$$

$$\mathrm{C_t} = f_t * C_t + i_t * \widetilde{C_t} \quad \text{公式(2.7)}$$

$$\mathrm{h_t} = o_t * \tanh(C_t) \quad \text{公式(2.8)}$$

公式中的$*$代表逐点相乘，从$\mathrm{f_t}$，$\mathrm{i_t}$，$\mathrm{o_t}$分别代表遗忘门、输入门和输出门，LSTM 单元通过这三个门结构来控制哪些信息需要保留，哪些信息需要遗忘和输出。不同于标准的 RNN，LSTM 通过逐点相乘和门结构的使用避免了多级 sigmoid 等激活函数嵌套导致在求导过程中对梯度的连续缩小或放大的问题，从而一定程度上避免了梯度爆炸和梯度消失问题。

通过前面对词向量、RNN、LSTM 的介绍，我们已经知道可以使用 RNN 可以将一个变长的自然语言序列转换为定长的向量表达，在此基础上可以完成一些分类和回归的任务，但是 RNN 的功能不止局限于此。

### 2.4.3 基于产生式的深度对话模型

Seq2Seq 技术，是最近兴起一类深度学习技术（模型），其全称为 Sequence to Sequence，该技术在许多产生式的任务中拥有较好的表现，比如：机器翻译、机器作曲、机器写诗以及对话机器人等。

Seq2Seq 技术（框架）最早被称为 Encoder-Decoder 框架，可以追溯到 Oriol Vinyals 等人 2014 年在 NIPS 发表的文章[14]中提出使用 Encoder-Decoder 框架来完成机器翻译任务，同期 Yoshua Bengio 等人也通过发表在 EMNLP[15]的文章提出了类似的解决思路。

这两篇论文所提出的 Encoder-Decoder 架构，为机器翻译方向的研究者的提供了新的研究方向，其主要思想如图 2.7 所示

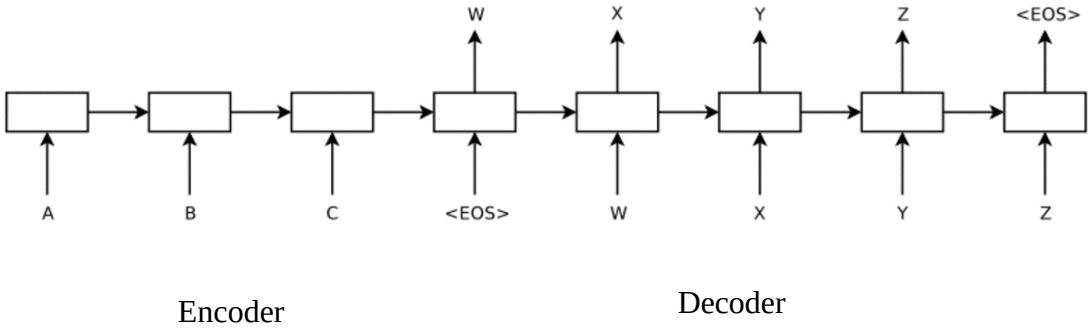



图 2.7 Encoder-Decoder 框架示意图

以机器翻译为例，以源语言的单词序列作为一个 RNN 的输入，并取出其最后时刻的输出，从而将输入序列编码成为一个定长向量V，这个将句子转换为定长向量的过程称为编码，期间使用的 RNN 称为 Encoder。之后使用另外一个 RNN 来完成从定长向量V到目标语言单词序列的转换过程，具体过程如下：

1）将V设定为该 RNN 的初始状态

2）在0时刻输入一个特殊符号EOS(End of Sentence)，并输出一个目标语言的单词

3）之后每个时刻都使用上一时刻的输出作为输入，并输出一个目标语言的单词

4）直到某一时刻输出的单词为EOS

在解码过程中使用的 RNN 称为解码器。编码和解码可以通过数学符号进行表示，令$\mathrm{X}=\{\mathrm{x}_1,\ldots,\mathrm{x}_{\mathrm{Ts}}\}$为源语言序列，$\mathrm{Y}=\{\mathrm{y}_1,\ldots,\mathrm{y}_{\mathrm{Tt}}\}$为 Encoder-Decoder 预测的序列，预测过程如公式(2.9)至公式(2.12)所示。

$$\mathrm{h}_\mathrm{t}=f(x_t,h_{t-1}) \quad \text{公式(2.9)}$$

$$c=h_{Ts} \quad \text{公式(2.10)}$$

$$\mathrm{s}_\mathrm{t}=f(y_{t-1},s_{t-1},c) \quad \text{公式(2.11)}$$

$$p(y_t|y_{<t}, X) = g(y_{t-1}, s_t, c) \quad \text{公式(2.12)}$$

其中c为 Encoder 对原始句子的编码结果，$p(y_t|y_{<t}, X)$为解码器综合历史信息计算当前时刻输出能量分布的函数。

Seq2Seq 技术可以广泛应用在各种产生式任务中，其中包括智能对话模型。基于 Seq2Seq 的对话模型可以建模成为如下形式：

- 输入部分
  - $C = \{c_0, \dots, c_{t-1}\}$ ： Context 双方的对话历史，t为当前的时间刻下标
  - $Q = \{q_0, \dots, q_M\}$ ：Query 用户的最后一句话，由多个单词组成，其中 $q_i \in D$，D为字典
- 输出部分
  - $R = \{r_0, \dots, r_N\}$：Response：机器人的回复

结合 Seq2Seq 的相关表达，单个样本的目标函数可以定义为公式(2.13)：

$$\ell = \sum_{n=1}^{N} \log P(r_n|r_0{:}r_{n-1}, Q, C) \quad \text{公式(2.13)}$$

通过最大化 response 中每个单词的最大似然来训练模型，使之产生的回复尽可能的同预期的回复相同。

根据 Seq2Seq 模型中是否考虑对话的上下文，可以将模型分为两类：无上下文的模型和考虑上下文的模型。

#### 2.4.3.1 无上下文的 Seq2Seq 模型

早期研究提出的 Seq2Seq 会话模型大多是没有上下文概念的模型。模型的目标是根据用户的最后一句输入，来产生尽可能自然的回复。相关的研究工作包括：O Vinyals 等人在 2015 年[16]首先提出使用 Seq2Seq 来训练自动问答机器人，并在两个数据集上进行了测试。类似的工作也在我国的微博数据集上得以应用，同样在 2015 年华为团队的 Lifeng Shang 等人[17]，在中文数据集上对一些 Seq2Seq 模型进行了比较，并提出一种带注意力机制的改进。

只使用用户的上一句话来生成回复，存在诸多不足，包括：

- 容易产生无意义的回复，如“I don’t know”等，使得对话难以继续

- 对话缺乏一致性，对于相同意思的提问比如“你多大了”和“你今年多少岁了”可能会产生不同的回复，这点对于普通用户来讲是难以理解的
- 回复的内容和对话的主题无关

**无意义回复的解决方法**

对于模型容易产生大量无意义回复的问题，主要的解决方法有两种。

一种是通过在 Seq2Seq 中增加 Attention 机制，从一些论文的实验结果中可以看出使用了 Attention 机制的模型产生的回复的多样性会明显优于不带 Attention 机制的对照模型。

另外一种思路是修改目标函数中的约束，来减少无意义回复出现的概率[18]，原始 Seq2Seq 模型的优化目标函数中只有P(Response|Query)相关的损失，且目标函数有一个严格的对齐约束，即如果模型预测出的结果同预期的Response不是完全一样的话，那么预测出完全不相关的一句话的损失和预测出同义但顺序不同的表达的损失是一样的。这使得模型在不知道如何回复时，预测一些高频句子的准确率得分期望会更高一点。而进行改进的方式是：首先，预训练一个根据Response生成Query的 Seq2Seq 模型A，我们通过公式$\mathrm{p_{qr}} = -\log(\prod_{i=1}^{m} P(q_i|q_0{:}\, q_{i-1}, R))$来估计给定某个Response，其上文为Query的概率值。再训练由Query预测Response的模型B时，将$\mathrm{p_{qr}}$加入损失函数，可以避免模型B在测试过程中出现过多无意义的回复。

**回复一致性问题的解决方法**

对于模型的前后回复缺乏一致性和回复和主题相关性不高的问题，现有的模型大多通过引入外部信息来约束模型的模型生成的内容来实现。Galley M 等人在 2016 年发表于 ACL 的文章[19]中，提出通过在 Decoder 中添加一个额外的人物属性向量，来解决机器人前后对话信息不一致的问题。Xing Chen 等人[20]提出一种使用 LDA 算法产生的主题词作为额外信息的改进 Seq2Seq 模型，在新浪微博语料集上比对照的模型拥有更好的上下文相关性。

#### 2.4.3.2 无上下文的 Seq2Seq 模型

相比无上下文的 Seq2Seq 模型，考虑上下文的模型则得到关注不久，相关研

究较少。Serban 等人在发表于 2016 年 AAAI 的文章[21]中提出一种层次化的 RNN 模型来对上下文建模。如图 2.8 中所示，该模型在编码过程中使用了两种级别的 RNN。底层 RNN 用来编码单句对话，将其转换成为定长的向量表示。高层的 RNN 以各个句子的向量表示作为的输入，其在每个时间刻输出对当前历史信息的编码给 Decoder。Decoder 以历史信息编码作为初始状态，预测下一句对话。

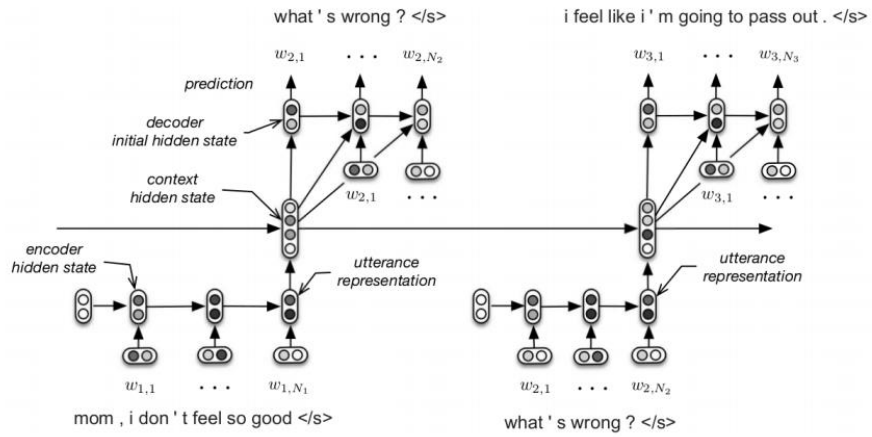



图 2.8 层次化的 RNN 模型用来建模的对话中的上下文[21]

作者后来这个模型的基础上提出了改进 [22]，通过在上下文编码过程中引入具有一定随机性的隐变量，从而获得了更好的产生效果，在人工评价指标中超出的原始模型。

带有上下文建模的 Seq2Seq 模型同普通的 Seq2Seq 模型一样，面临着容易产生无意义的回复、产生回复质量低等问题，且由于依赖序列过长，导致模型增大，训练较慢。

### 2.4.4 基于检索的深度对话模型

相比使用产生式模型存在的各种问题，基于检索的会话模型则在回复内容上具有更高的质量，这是因为两方面的原因，一是基于检索的模型的目标函数相比 Seq2Seq 的模型其具有一定容错能力，二是检索式的模型给出的回复必然在语料库中，所以本身出现语法错误的概率比较小。当然也有不足之处：基于排序的模型只能回复语料库里的句子，而在一些高度个性化的场景下，语料库中可能并不

存在一个合适的句子用来回复。

#### 2.4.4.1 基本框架

基于检索的深度对话模型定义如下：给定语料集$C = \{c_0, \dots, c_m\}$，其中的任意一个元素$c_i$均是一个会话（Session），包含两个或两个以上句子（utterance），$c_i = \{u_0, \dots, u_n\}, n \geq 2$，在每个$c_i$中，均有一个或多个$u_j$为一个正样本和多个负样本组成，称之为候选集，其中正样本为原始会话中该位置的句子，负样本为通过特定方法采样得到内容不同的句子。对话模型的训练目标就是，学习一个模型，能够根据会话（Session）中上一句对话或者全部的对话历史信息，从候选集中选出正样本。

本文对基于检索的深度对话模型进行如下总结，基于检索的深度对话模型的基本框架可以分为三个部分，如图 2.9 中所示。

包括三个部分：上下文编码模块、候选句子编码模块以及相似性计算部分。上下文编码模块和候选句子编码模块的目标都是将一个或多个句子的通过深度模型编码成为定长的向量表达。相似性计算模块的目标则是在给定上下文的向量编码和一个候选句子的编码，判断候选句子是否是当前这个会话的正确回复，相似性计算模块通过输出一个浮点数来表示当前候选句子的得分。最后对话模型通过比较所有候选句子的得分，来选择一个得分最优的句子作为最终输出。

同样，基于深度排序的对话模型根据其是否使用出了最近一句对话之外的上下文信息可以分为无上下文的对话模型和带上下文的对话模型。两者之间的主要区别在于上下文编码模块使用的信息和模型不同，相关编码的基本原理和方法类似于上一小节在 Seq2Seq 技术中介绍的相关模型，此处不再做详细介绍。

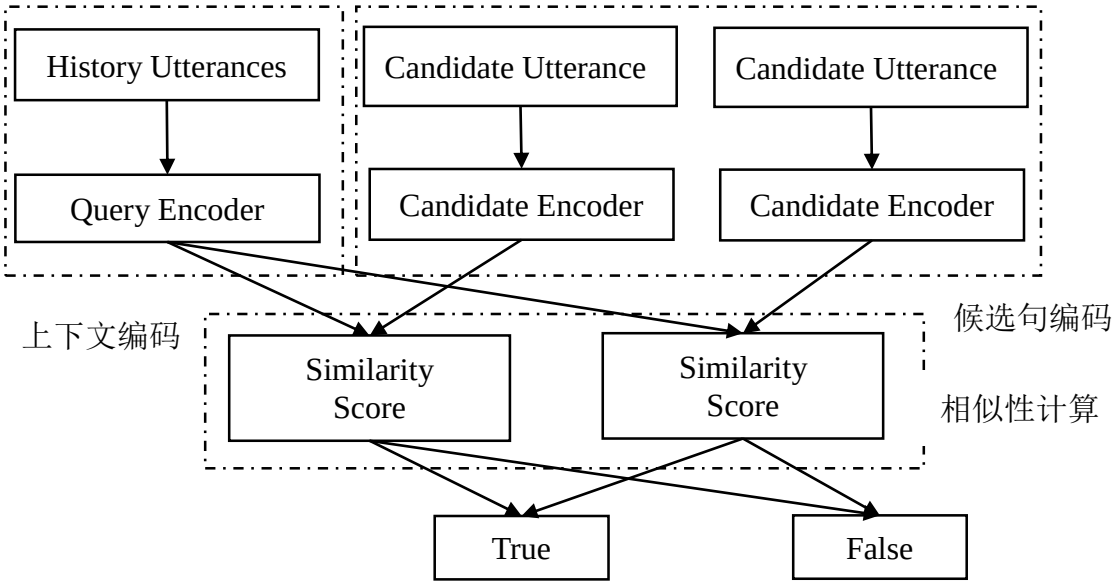



图 2.9 深度排序模型的抽象框架

目前在上下文编码方面的研究还比较有限，已有的模型主要包括：

Inaba M 等人[23]在 2016 年 ACL 的文章中提出一种使用 RNN 对完整上下文（包括候选回复）进行编码的排序对话模型，并在其中根据其训练和实际应用的人机智能问答对话场景引入了一种假设：完整的对话可以分割若干个的不相交问答对，每个问答对均包含一个人类用户提问和与之相邻的一条机器人回复。

Yan R,等人[24]在 2016 年发表于 SIGIR 的文章中提出了一种将 RNN 网络和 CNN 网络进行结合的对话模型，并使用对原始的对话内容进行改写以提升模型的准确度。

Xu Z 等人[25]在 2016 年提出使用额外的弱结构化知识库来提升产生对话的性能。该模型通过对上下文的进行预处理，并检索出与之相关的关键词作为附加“知识”。通过一种经过修改的 LSTM 单元，将额外的知识处理为定长向量，在 Encoder 运行的每个时间片输入给 Encoder，从而实现使用附加的弱结构 “知识”来约束模型，使之产生更好的回复内容。

#### 2.4.4.2 优化目标的选取

在优化目标函数的选取上，基于检索的深度对话模型目前主要使用两种方法。

分别对应于 Learning to rank[26]中的 Pointwise 和 Pairwise 两种方法，基于 Listwise 的方法则在目前此领域的研究当中比较少见。

Pointwise 方法，处理的对象是单个上下文和单个候选回复的组合，对于给定的上下文和候选回复{Context, Response}，令其所对应的向量表达分别为$\mathrm{V}_C$和$\mathrm{V}_\mathrm{R}$（向量表达通过模型中的对应编码器得到），然后使用一个组合函数将两个定长向量进行组合，成为一个统一的向量表达$\mathrm{V}_\mathrm{u} = \mathrm{f}(\mathrm{V}_\mathrm{c}, \mathrm{V}_\mathrm{R})$，最后以$\mathrm{V}_\mathrm{u}$作为输入，以候选回复和该上下文的匹配程度作为类标签作为预期输出,如{good, normal, bad}，训练一个分类器。在测试过程中，使用该分类器的打分函数作为对各个候选回复排序的依据，并选出得分最优的候选回复作为会话模型的输出。在这个过程中，上下文向量和回复向量的组合可以通过多种方式实现，包括简单的向量拼接、多层感知机等等。分类器训练也可以选取各种成熟的技术，包括：神经网络、最大熵、支持向量机等。

Pairwise 方法，是目前比较流行的方法，其考虑的重点是文档之间的相对顺序。在会话模型中，这类方法输入为：Context, True Response, False Response构成的三元组。Context同之前一样，是对话的上下文，True Response是和该上下文相匹配的一条回复，False Response则是和该上下文不匹配的回复。Pairwise 方法的处理流程如下：先分别计算 Context 和两个 Response 的向量表达$\mathrm{V}_\mathrm{C}, \mathrm{V}_\mathrm{T}, \mathrm{V}_\mathrm{F}$；其次通过一个回归模型或者相似性度量函数如 Cosine Similarity 等，分别以$\{\mathrm{V}_\mathrm{C}, V_T\}$和$\{\mathrm{V}_\mathrm{C}, \mathrm{V}_\mathrm{F}\}$作为输入，计算出上下文同两个答案的相似性得分$\{\mathrm{S}_\mathrm{T}, S_F\}$，最后使用一个 hinge loss 的变体函数作为优化目标函数：$\ell(\mathrm{V}_\mathrm{c}, \mathrm{V}_\mathrm{T}, \mathrm{V}_\mathrm{F}) = \max(0, \mathrm{margin} - \mathrm{S}_\mathrm{T} + \mathrm{S}_\mathrm{F})$，其中margin是根据经验设定的正负样本之间的得分差距。

## 2.5 本章小结

本章围绕对聊天机器人和客服支持系统的相关概念入手，阐述了二者的主要应用场景和发展历史，并论述了聊天机器人的对话技术模型在客服支持系统中的重要作用及研究意义。之后围绕聊天机器人的相关技术，以相关技术的发展顺序为主，分别详细介绍了基于模板的对话技术和模型、基于传统检索的对话技术和

最新的基于深度学习的对话模型，并对这些技术和模型中使用和依赖的一些基础技术进行了简要介绍。为本文后续章节的内容展开进行铺垫。

# 第3章 基于模板的客服辅助技术

本文利用现有的 AIML 模板匹配技术设计实现了一种知识库查询服务，为人工客服提供便捷的专业知识查询方法，提升其服务质量。

但现有模板匹配方法经常遇到关键词匹配失效问题。比如：用户输入了关键词“白癜疯”，实际上是想咨询“白癜风”相关的信息，但现有的模板匹配方法不能直接将“白癜疯”的相关咨询映射到“白癜风”的相关模板上，导致模板匹配失效。

针对这个问题本文提出了一种同义词多扩充算法，该算法通过预先构建垂直领域关键词的同义表达集（包含同义词和常见的错误表达）来减少关键词匹配失效问题出现的次数，进而达到提升模板匹配的效果。

本章的内容按照如下顺序组织：

1）应用背景和面临的问题。介绍了：知识库查询服务作为客服辅助技术的必要性；使用 AIML 模板匹配技术实现该服务的主要思路；当前模板匹配方法中存在的主要问题。

2）领域关键词扩充算法。详细介绍了本文提出的关键词扩充算法。

3）知识库检索服务设计。详细介绍了知识库查询服务的设计思路和部分实现细节。

## 3.1 应用场景和面临的问题

**知识库查询服务及实现思路**

由于客服在工作过程中需要使用的信息和知识较多，且人工客服本身的业务水平参差不齐，所以在服务过程中，为人工客服提供相关专业知识作为参考，是提升服务质量的重要手段。

作为参考的知识主要有两个方面来源：1）客服平台在运营过程中，通过业务运营沉淀出的知识库，类似于商品的属性、常用问答、标准解决方案等内容；2）

垂直领域的专业性较高的知识库，比如医疗健康领域的疾病百科、药品知识库等。

对于类似于商品属性、疾病百科这样的知识，具有这样的特点：个性化程度相对较低；由一系列类别相同实体属性及属性值组成；实体的属性通常是少量可枚举；用户可能的提问存在明显模式；而且模式数量相对较少。例如：对一个电商平台来讲，其可能拥有数以万计的商品，但是在知识库中这些商品的属性则相对统一且有限，如价格、优惠、产地等，因此查询这些知识自然语言问句也有规律和模式可循：咨询商品A价格使用的提问，和咨询商品B价格使用的提问，二者直接除了商品名称之外，基本一致。

我们已经在相关工作中介绍过，基于模板匹配技术是对话机器人中一种比较成熟的技术。相比其他实现方法，其优点在于通过模板匹配产生的回复内容高度可控，缺点在于不同句式的模板通常需要人工参与才能生成，成本比较高。

不难看出，使用模板匹配技术非常适合用来为前面提到的个性化较小的知识库提供一个使用学习成本低、容易使用的知识库自然语言查询接口，从而为客服提供辅助。

本文使用 AIML 模板匹配技术实现该服务，主要思路如下：1）根据知识库的有限属性，手工编写一些模板，这些模板能够匹配合适的用户输入，并从其中解析出关键词和待查询属性；2）根据解析出的关键词和待查询属性，返回对应的知识条目。

**实验数据的选取和收集**

随着互联网的发展和人们生活方式的变迁，在线问诊和在线健康咨询平台在不断涌现。这些问诊和咨询平台招募了大量医疗从业人员以在线的方式为用户提供咨询服务。在这个过程中，医疗从业人员扮演的角色同在线客服存在相似之处，不同的是，在线问诊对服务提供者的专业素养要求比较高，同时医疗行业对服务的质量高度敏感，稍有差池便会引起纠纷，使用纯自动的机器人技术不成熟也难以被大众接受。从另外一个方面讲，因为健康医疗专业在生产生活教学中的特殊地位，一些医疗健康的专业知识已经通过各种途径被电子化，可以直接在互联网上获取。还有一些健康医疗社区的语料相对容易获取，这些是其他专业领域所不

具备的优点。因为在线健康咨询的这些特点，我们使用其作为应用和实验的场景。

本文使用从医学百科网站上爬到的疾病百科和药品说明书作为实验知识库。在爬取数据之前，我们对比了寻医问药、A+医学百科和医学百科等网站的数据，发现寻医问药的信息较为丰富、结构化程度较高，故选择寻医问药的数据作为实验知识库，考虑到本文的目的是为客服提供专业知识作为参考，而非直接返回给咨询用户作为参考，故可以接受知识库中少量的错误，这些错误可以通过人工客服在使用过程中进行修正。本文在此知识库的基础上设计并实现了一种通过自然语言查询知识库的客服辅助服务，来为专业客服提供知识检索，该服务通过 AIML 模板匹配技术实现。

**现有方法存在的主要问题**

该客服辅助服务的核心是模板匹配技术，同其他检索技术一样，模板匹配也面临两个较难处理的问题，包括：

1）用户表达的不规范问题

网络环境下，用户在表达过程的用词变化较多，经常会出现错别字，输入不完整等表达方式，与标准的书面表达不一致。例如：六味地黄丸/六位地黄丸/六味帝皇丸/地黄丸，在不规范输入的情况下可能指同一种药品“六味地黄丸”。这种不规则的描述会使得模板匹配失效，从而不能为用户返回预期的内容。同时因为一般的咨询过程都存在多轮问答，在有上下文的环境中，用户的表达中通常会出现大量句法结构不完整的短句，因此传统的句法解析工具也无法有效工作，进而增加了处理难度。

2）实体或概念的别名问题

即使在正常的输入情况下，对于同一个实体或概念，也有可能会存在多个的别称。例如：“阿尔茨海默病”又称“老年性痴呆症”，在实际使用过程中，又经常简写为“老年痴呆症”；一般药品会有多个名称，包括通用名、商品名以及别名。因为知识库的质量、完整程度和覆盖率问题，这些别称和简写通常在知识库中覆盖不够全面。因此在匹配过程中，如何将一些知识库中未登记的实体和概念别名映射到正确的知识条目上，也是模板匹配中重要问题。

本文提出的知识库检索工具在使用过程中，其完整处理流程如下：1）分词，2)词归一化处理，3)匹配模板，4）产生回复。其中模板部分通过手工编写，分词和词归一化的部分使用关键词扩充算法产出的同义词表作为辅助。

## 3.2 领域关键词扩充算法

自然语言表达中存在的表达不规范和实体别名问题，会对模板匹配及其他自然语言处理任务造成困难，因此如何将自然语言文本中的不规范表达和实体别名映射到正确的实体或单词，是一个必须解决的问题。

一个可行的思路是：关键词扩充。即使用算法从语料中挖掘知识库中专业领域关键词的同义表达，并建立一个同义词表。通过使用这份同义词表，我们可以将用户输入中一些出现频率比较高的不规范词映射到一个标准的、在知识库中有收录的领域关键词。从而实现定位到正确的知识条目，提高模板匹配性能。

按照这种思路，本文提出了一种基于词向量的多轮同义词扩充算法来完成领域关键词扩充。该算法使用已有的关键词作为种子，通过人工辅助的多次迭代来完成同义词词典的扩充。

其处理过程的包括：1）数据预处理，2）基于字典的种子词扩充，3）基于词向量的多轮扩充

下面我们对将该算法和与其功能相似的传统算法进行介绍。

### 3.2.1 数据预处理

整个数据预处理的流程如图 3.1 所示，分为：爬取数据、数据整理、新词发现、分词、词向量训练等 5 个部分。


图 3.1 数据预处理流程

我们基于 python 的开源软件 Beautiful Soup[27]实现了一些爬虫，并使用这些爬虫从 Ask39[28]、寻医问药[29]两个健康医疗社区网站，获得了一些健康医疗问答数据，其中从 Ask39 网站获得数据约 79 万条，从寻医问药获取的数据约 20 万条。每条数据均包含一个用户的提问、一条或多条医疗从业人员的回复以及一些元数据。同时从寻医问药网站爬到了全部的疾病百科知识和药品说明书作为垂直领域的知识库。

通过对获取的问答数据进行整理，删除其中多余格式以及附加元数据，只保留提问和回答的文本内容，我们将其整理成为一个健康医疗的语料库用来训练词向量。同时对疾病百科和药品说明书的数据进行整理，删除异常字符，将文本信息转换为统一编码，并设计合适的数据库存储格式，将这部分数据存储到数据库中，作为后续任务中的知识库使用。

同大多中文自然语言处理任务一样，想要对关键词进行扩充，首选要对文本语料进行分词。但是考虑到我们的目标是能够找出语料中常见用户的不规范输入，且使用了医疗行业的语料，因此直接使用分词器进行分词会难以得到我们期望的结果，例如我们想要将用户输入的“六味帝皇丸”映射到标准的药品名称“六味地黄丸”，就需要先把“六味帝皇丸”这个词从句子中单独切出来，但直接使用分词工具进行分词会得到“六味/帝皇/丸”的结果，这种将目标词切分为多个词的情况，无疑为处理过程增加了难度，且不能被预选设计的算法所处理。

因此，我们在分词之前，我们在上一步整理的健康医疗数据集上进行了未登

录词挖掘，尝试从中找出一些分词器未收录的词汇。在这个过程中，我们使用了能够计算候选词汇的“凝固度”、“正向信息熵”、“反向信息熵”和“频次”四个特征的新词发现算法，算法及使用的特征在第二章中已经有过介绍，这里不再解释。将完整的未分词语料作为输入，得到所有候选词汇及其在四个特征上的得分。通过选取适当的阈值，筛选得到纯度较高的候选词。本文实验过程中使用的阈值如表 3.1 所示。

表 3.1 新词发现算法阈值设定

| | |
|---|---|
| 凝固度阈值 | 0.0003 |
| 正向自由度阈值 | 0.5 |
| 反向自由度阈值 | 0.5 |
| 词频阈值 | 4 |

通过人工对该筛选结果进行过滤，删除其中明显不是单个词的项目，得到质量比较高的未登录词结果。最后将这些词添加进分词器的词典中，用于后续处理。

接下来对语料进行分词处理。目前开源的分词软件较多，常用的工具有：Stanford Tokenizer[30]、Ansj[31]、结巴分词[32]等等，我们在分词处理中使用了 Stanford Tokenizer 作为分词工具。

之后本文在分好词的语料基础上进行了词向量训练。训练的方法包括 Word2Vec[12]、LSI[33]和 GloVe[34]，GloVe 是斯坦福提出的一种词向量训练模型。词向量的维度设定为 300 维，其中 Word2Vec 的训练使用 Skip-Gram 模型，上下文窗口大小设定为5，单词频次阈值设定为 3，其余参数采用默认值。对训练好的三种词向量进行简单评估，分别使用三种词向量来查找同样本关键词最相近的词语，通过人工对比查找结果发现，通过 LSI 和 GloVe 词向量找到的单词存在较多共现词，即经常与样本词一起相邻使用的词，但二者的语义相关性并不高。而 Word2Vec 词向量找到的词则同样本词语义相关性相对较强。我们这个现象进行分析，推测可能是垂直领域的语料（健康医疗问答数据）数量太少，和通用领域使用的语料相比（如维基百科）要少 1-2 个数量级，所以导致 GloVe 的词向量性能非常差。

故本文最终选择使用 Word2Vec 训练出的词向量。

### 3.2.2 传统扩充算法

如前文所述，如何进行关键词扩充是检索任务中一个比较常见的问题，如搜索引擎调优等任务中也有类似的需求。两类常用的方法是：

- 字符串相似度匹配算法
- 单词的语义相似度匹配算法

**基于字符串相似度的匹配算法**

关键词的标准表达和用户非标准表达之间通常存在一定字符串相似性，比如拼写错误的情况，正确书写和错误书写的字符序列中，大部分字是一样的，且大部分字与字之间的位置关系保持一致。不一致的情况可以分为下面几种：

- 关键词中的若干个个字拼写成为另外一些字
- 非标准表达相比表达表达少了若干字
- 非标准表达相比标准表达多出了若干个字

对于这类型的非标准表达，我们可以通过使用合适的字符串相似性度量函数，来将可能的非标准表达筛选出来。最小编辑距离（Levenshtein distance）就是一个这样的算法，其基本思想是，通过计算需要多少次原子操作能将字符串A转换为字符串B，来度量两个字符串的相似程度，最小编辑距离中允许的原子操作总共有三种：增加一个字符；删除一个字符；将一个字符替换为另外一个字符。但是原始最短编辑距离只考虑了两个字符串之间不相同字符的绝对数量，并没有考虑字符串原始长度的影响。在实际使用的过程中，通常会根据语料的特点，在编辑距离的基础上添加字符串长度进行标准化处理，比如公式(3.1)为莱文斯坦。

$$\text{ratio} = 1 - \frac{LD(w_1,w_2)}{len(w_1)+len(w_2)} \qquad \text{公式(3.1)}$$

其中$\mathrm{LD}(\mathrm{w}_1,\mathrm{w}_2)$最小编辑距离计算函数。

**单词语义相似度算法**

如相关工作中的介绍，词的向量表示已经有较长发展历史，一些模型和算法能够将词汇之间的语义相关性进行建模，从而使得在这些向量表达上度量词汇的语

义相关性成为可能。

经典的带语义的向量表达方法有：隐含语义索引（LSI，Latent Semantic Index）[33]，Word2Vec[12]等方法。

LSI 通过对单词-文档矩阵进行奇异值分解（SVD，Singular Value Decomposition），从而分别获得可以表示单词和文档语义属性的两个矩阵。其基本思想是：如果两个单词具有很强的关联性，那么当一个单词出现在某篇文档中时，另外一个单词的应该也会出现。这里的关联性包含：同义词、主题相关等标准。LSI 在单词和文档层面进行建模，这使得要想获得比较好的表达结果，往往需要大量文档级别的语料库作为输入，同时过于庞大的单词-文档矩阵在做 SVD 分解时计算速度也比较慢。在实际使用过程中，我们发现使用较小语料库训练出的 LSI 单词矩阵作为词向量时，距离某个词最近的一些词，大多是经常与该词相邻出现的一些词汇，而非该词的同义词。

Word2Vec 的基本原理已经在第二章中有过介绍，这里不再赘述。通过 Word2Vec 也可以获得带有语义的词的向量表达，相比 LSI 等基于矩阵分解的方法，Word2Vec 优点在于其在单词和单词的小范围上下文层面上进行建模，使用神经网络进行训练，其参数数量大小确定可知，训练速度非常快，一个优化的单机版 word2vec 程序，每天可以训练上亿词，最重要的是 word2vec 在建模过程中，关注范围更为集中，其训练出的词向量相比传统的 LSI 和潜在狄立克雷分配[35]（LDA，Latent Dirichlet Allocation）相比，具有更丰富的语义信息，所以词向量质量更高一些。

基于语义的相似度算法，通过计算不同词汇词向量之间的相似度得分来实现，通常相似度得分通过余弦距离计算得到：

$$\mathrm{Sim}(\mathrm{w_a}, \mathrm{w_b}) = \frac{V_{wa} \cdot V_{wb}}{|V_{wa}|_2 |V_{wB}|_2} \qquad \text{公式(3.2)}$$

通过遍历整个字典中的所有词，并计算它们与给定单词相似度得分，从中筛选出得分最高的单词即为算法选出语义最相关的单词。

### 3.2.3 多轮扩充算法

在上一小节中介绍的两类关键词扩充算法，分别通过字符层面和语义层面的相似特征来寻找同义单词，相对而言基于语义的相似度更加稳定，且其性能理论上可以通过训练更好的词的向量表达来得到提升。

但是在实际使用过程中受制于垂直领域语料库的大小和词嵌入模型的性能，往往并不能直接获得质量非常高的词向量，故而直接使用语义相似性匹配得到的同义词的准确率不高，检索到的同义表达数量也有限，需要人工在检索结果的基础上进行筛选。

针对这些问题，本文提出了一种基于词向量的同义词多轮扩充算法，能够在少量人工介入的情况下，快速根据给定关键词从词典中找出其同义词。

其主要思想如下。在词嵌入质量较低的情况下，同义词之间的距离并一定是最近的。使用单个词向量进行匹配的精度有限，因为其基本假设是在向量空间的分布中，具有某个单词最近的若干个单词中很有可能包含其同义词。但是这种方法十分依赖于词向量的质量，好的词向量训练需要大量语料支持，在垂直领域往往语料有限，训练出的词向量质量较差，导致选取查找同义词的准确度和召回率不足。

在本文提出的算法中，我们使用了新的假设：语义相同的一组词，如图 3.2 所示{痤疮、粉刺、青春痘、痘痘}构成一个同义词集，在向量空间中同义词集中的各个单词的向量分布存在聚类效应，即各个单词的向量分布在真实语义向量（图中 C1）周围。每个同义词集的真实语义可用集合内各个单词的向量平均来表达，如图中C1所代表的向量即为 4 个单词向量的平均。在查找同义词时，我们可以通过增加查询词数量，来更准确定位该概念的真实向量，从而减少或避免因为训练不足或随机扰动导致的错误。如图 3.2 中所示，如果单独使用{痘痘}进行查询，容易匹配到一些和该词经常一起出现的单词，如{痘印，色斑}等，但如果同时使用{痘痘，痤疮}进行查询，则单词{痤疮}可以对匹配结果进行约束，使得最佳匹配的单词为{粉刺}，而非{痘印}。

在这种假设上，我们设计了一种多轮同义词扩充算法，算法的输入为一个领域关键词，输出则为可以查找到的同义词列表。

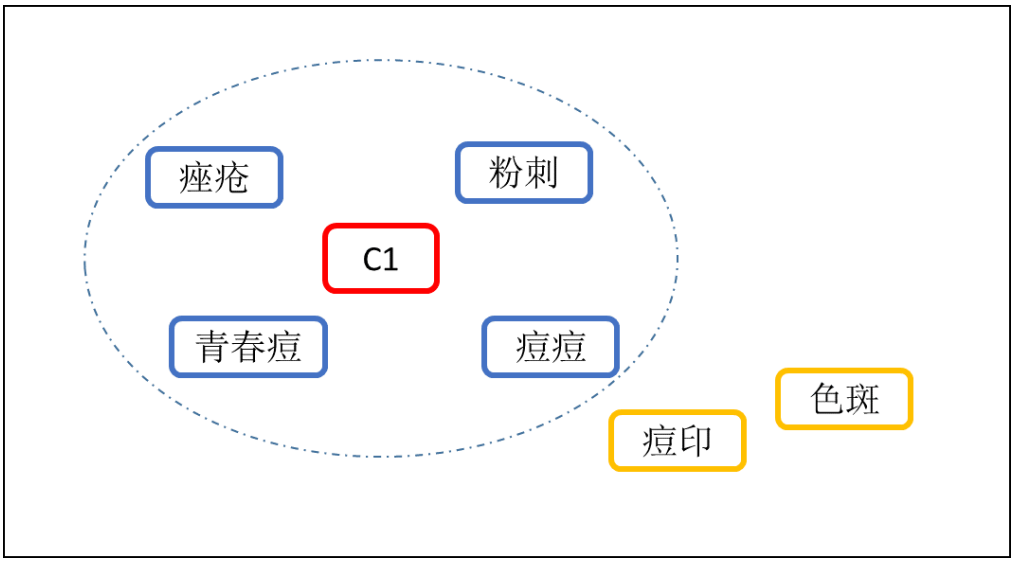



图 3.2 同义词的聚集效应示意图

算法的处理过程分为两步：第一步建立种子词集合；第二步循环使用语义相似度对种子词进行扩种。

如前面所述，增加查询词的数量，可以提升同义词匹配结果的质量。因此在算法的第一步，我们需要在保证准确性的情况下，获取查询词所对应的同义词集，称之为种子词集。

为了保证种子词集准确性，我们使用知识库中的相关属性作为种子词集。对于药品名，使用{商品名，通用名}作为种子词集。对于疾病名，我们使用疾病百科中的别名属性值作为种子词集。

在获得查询词的种子同义词集合之后，我们通过一种循环迭代的方式来不断扩充和查找同义词。单轮扩充过程可以描述为：

给定同义词集合$S = \{s_1, .., s_n\}$，其所对应的词向量集合为$V = \{v_1, \dots, v_n\}$。对于任意一个候选单词w，其词向量为$v_w$，其与该同义词集的相似度计算方式为公式(3.3)所示：

$$Sim(S, w) = \frac{1}{n}\sum_{i=1,n;s_i \in S} \frac{v_i \cdot v_w}{|v_i|_2 \cdot |v_w|_2} \qquad \text{公式(3.3)}$$

通过遍历字典中单词，计算其与当前同义词集的相似度，之后以其作为依据进行排序，通过设定阈值或者人工筛选的方式从排名靠前的单词中，选出新的同义词，并将其加入同义词集。

完整处理流程如表 3.2 所示。

表 3.2 算法流程

| |
|---|
| 1. 输入查询词q |
| 2. 根据q找出知识库中已有的同义词，同q组成同义词集Q |
| 3. 利用公式(3.3)计算字典中每个词w与同义词集Q的相似度$S_{wQ}$ |
| 4. 通过阈值限定或人工筛选获取新增同义词集$Q_{new}$ |
| 5. 同原始同义词集进行合并$Q = Q \cup Q_{new}$ |
| 6. 若迭代次数未超出阈值且$Q_{new}$不为空,则回到步骤 3 |
| 7. 返回同义词集合Q |

### 3.2.4 算法性能评估

本节对上一小节中提出的算法性能进行了评估。本文先根据疾病百科的知识条目构建了一个疾病名词集合，之后在社区问答语料库上进行统计各个疾病的提及频次，并按照出现次数从高到低对所有疾病名词进行排序。从出现次数较多的前 1000 个疾病名词中随机抽样 50 个作为的测试集。通过人工评判的方法对算法的能力进行测试。

为了尽可能的方便对比，我们选取的 Baseline 算法为单纯通过计算 Word2Vec 词向量相似性得分来选取同义词。

测试中限定每次的只从得分最高的五个单词中挑选同义词，多轮算法的最大迭代次数限定为 3 次。评价的指标包括：

- Hit Count:算法检索出同义词的数量
- P@1:得分最高的词为同义词的比例

测试结果如表 3.3 中所示，从中我们可以看出对比 baseline，本文提出的方法能够明显增加同义词的查找数量。但其不足之处在于，在第一轮迭代中准确度较

低。

表 3.3 同义词数量对比

| 算法及迭代次数 | P@1 | Hit Count |
|---|---|---|
| Baseline | 0.32 | 0.73 |
| Ours@iter-1 | 0.67 | 0.86 |
| Ours@iter-2 | 0.72 | 0.59 |
| Ours@iter-3 | 0.45 | 0.21 |

表 3.4 为多轮扩充算法在几个出现频次较高的疾病名词的扩充效果。表中空白部分代表该部分没有同义词被引入，通过该表可以看出，多轮扩充算法对高频关键词同义词扩充有较好的效果。

表 3.4 多轮扩充算法示例

| 核心词 | 种子词 | Iter-1 | Iter-2 | Iter-3 |
|---|---|---|---|---|
| 痤疮 | 青春痘、粉刺 | 痘痘、座疮、长痘痘 | 起痘痘、起痘 | 青春豆 |
| 阳痿 | | 阳萎、阳痿、不坚、勃起功能障碍 | 不举、勃而不坚 | 勃不起 |
| 痔疮 | | 内痔、外痔 | 痔 | |

## 3.3 知识检索服务设计

本文结合新提出的领域关键词拓展算法和传统的 AIML 模板匹配技术，针对我们构建的健康医疗知识库设计并实现了一个知识检索服务。该服务接受用户的自然语言咨询作为输入，使用手工构造的 AIML 模板，将用户查询映射为关键词和查询属性的组合，并返回对应知识。

本文将该服务的构建和使用过程分为三个部分进行说明：领域关键词扩充、模板构建、知识检索。

**领域关键词扩充**

领域关键词扩充模块实现了本文提出的多轮扩充算法。其通过对知识库和语料的交叉对比，将知识库中的领域关键词（药品名和疾病名）根据人工设定的阈值，划分为高频关键词和低频关键词。对于高频关键词使用多轮扩充算法进行扩充，建立这些高频关键词的同义词表。低频关键词直接使用知识库中的别名属性作为其同义词集。在多轮扩充算法中，用户可以选择在扩充过程中或扩充过程结束之后对同义词集进行人工筛选，以获得最终可以使用的同义词集。

**模板构建**

本文针对前面整理的医疗知识库的属性和用户可能的提问方式，手工构建了一批 AIML 模板。这些模板可以实现两个功能：1）关键词提取；2）用户查询属性提取。如图 3.3 所示为一个手工模板的例子，该模板对应于查询疾病百科中的治疗方法信息。例如，当用户输入“长痘痘了应该吃什么药”时，可以先通过前面扩充得到的同义词集映射，将“长痘痘”改写为“痤疮”，之后通过模板匹配，命中此模板，提取出疾病关键词“痤疮”待查询属性名“treatment”。

```
<category>
    <pattern> ^ <set>disease</set> ^ 吃什么药 ^ </pattern>
    <template>
        <set name="detail"><get name="detail"/>|<star index="2"/>;treament;disease</set>
    </template>
</category>
```

图 3.3 手工模板示例

**知识检索**

本文以 Web API 的形式实现了知识检索服务。该服务接受用户的自然语言查询服务作为输入，通过分词、同义词改写、属性提取、返回对应知识这几个步骤来完成服务。其中属性提取即为前面提到的模板匹配过程。

该服务已经应用于阿里健康轻问诊平台的客服系统中，在实际使用过程中，该服务实时的根据用户和客服（医疗从业人员）的对话，为客服提供其可能需要参考的专业知识。

目前针对疾病百科和药品说明书知识库，该服务可以提供以下属性的自然语

言查询，如表 3.5 所示。

表 3.5 目前支持查询的属性

| |
|---|
| ● 疾病症状 |
| ● 疾病传播方式 |
| ● 疾病易感人群 |
| ● 疾病确诊需做检查 |
| ● 疾病治疗常用药物 |
| ● 药品功能主治 |
| ● 药品使用方法剂量 |
| ● 药品使用禁忌和注意事项 |

表 3.6 中展示了本文设计的知识检索服务部分使用示例。

表 3.6 部分查询示例

| |
|---|
| 脸上长痘痘应该吃什么药？<br>【痤疮的常用药品】头孢氨苄胶囊 甲硝唑乳膏 |
| 通脉颗粒能治心绞痛吗？<br>【通脉颗粒的功能主治】本品活血通脉。用于缺血性心脑血管疾病，动脉硬化，脑血栓，脑缺血，冠心病，心绞痛。 |
| 得舒特有什么副作用吗？<br>【得舒特的禁忌和注意事项】不良反应:1、极少数人中观察到轻微的胃肠不适 。2、极个别人出现皮疹样过敏反应。 |
| 足光散怎么用？<br>【足光散用法用量】外用，取药粉 40 克加沸水 1000～1500 毫升，或取药粉 20 克加沸水 500～750 毫升，搅拌，溶解，放温，趁热浸泡患处 20～30 分钟，一日 1 次，连续三日为一疗程。 |

## 3.4 本章小结

本章围绕模板匹配用于客服辅助的相关技术和研究展开。详细介绍了利用模板匹配技术实现的客服辅助服务的功能和主要实现框架。并针对当前技术所面临的关键词匹配失效的问题，进行了仔细分析，并提出了一种的多轮同义词扩充算

法。并通过实验与对照算法进行对比，验证了算法的有效性。

虽然多轮扩种算法对比 Baseline 方法具有更高的同义词检出数量，但是在获得准确可用的同义词集的过程中，仍然需要人工参与，这是算法有待改进的地方。

我们将该算法封装成为一个可以运行使用的模块，并针对示例的垂直领域知识库，实现了一个客服辅助服务，以 Web API 的形式提供调用，展示了模板技术用于客服辅助问题的有效性。

# 第4章 深度对话模型用于客服回复推荐

目前深度对话模型是相关领域的研究热点，相比于传统基于手工特征的自然语言处理方法，深度对话模型的优势在于其端到端的训练方式，模型可以自动学习所需特征，不需或极少需要人工设计特征。

本文将深度对话模型应用于客服回复推荐任务中。设计了用于客服辅助场景的深度模型，其能够根据当前咨询对话历史，实时的为客服推荐其可能需要输入的回复。本文希望通过这种预测回复的方式减少客服输入量，提升客服工作效率。

客服回复推荐任务的背景是：客服和用户的自然语言对话，对话中包含两个角色：$\{\mathrm{User}, \mathrm{Host}\}$，User 为发起对话的咨询用户，Host 代表提供专业咨询服务的人工客服。为了简化问题，我们将对话过程建模为：User 和 Host 严格交替输入的形式，相邻的两句对话必须是不同角色的输出，且所有对话都有用户提问或描述开始。回复推荐问题可以进行如下建模：

- 输入：从开始到最近一次用户输入的所有对话内容：$\mathrm{C_j} = (\mathrm{U_0}, \mathrm{H_0}, \mathrm{U_1}, \mathrm{H_1}, \dots, \mathrm{U_j})$
- 输出：在当前上下文下，Host 最可能的回复：$\mathrm{H}'_j = \mathrm{F}(\mathrm{C_j})$

其中：

- $\mathrm{U_n}$为当前对话中用户说的第$\mathrm{n}$句话
- $\mathrm{H_n}$为当前对话中客服说的第$\mathrm{n}$句话
- $\mathrm{F}$为根据上下文预测当前回复的方法

本文尝试通过两种不同的思路来完成客服回复推荐任务：基于产生式的对话模型和基于检索式的对话模型。在两种思路上，均分别设计实现了用于客服回复的带上下文的对话模型，同时也实现了对应的不带上下文的会话模型作为对比。最后通过实验对这些对模型进行了测试。

## 4.1 基于检索的深度对话模型研究

基于检索的深度会话模型，其基本思路是：从历史的对话记录中学习一个模型M，其能够计算给定一句话作为特定提问或上下文有效回复的合适程度。在预测过程中，预先构建一个客服回复内容的候选集合$R = \{r_1, \ldots, r_n\}$，然后根据当前对话的提问或上下文内容，遍历R中所有候选回复，计算各个候选回复的匹配得分，最终选取得分最优的回复返回。

根据模型M在计算过程使用历史对话信息的多少，基于检索的对话模型，可以分为两类

1）无上下文的检索模型。只根据用户的最后一句对话进行预测，不利用完整的对话历史信息。其优点在于模型简单，缺点在于没有利用完整的上下文信息，即使让人类来判断也难度很高。

2）带上下文的检索模型，匹配过程中使用全部上下文的信息进行预测。其优点在于使用完整上下文信息，理论上可以完成更为精确的匹配，缺点在于上下文中包含非常多的冗余信息，目前缺乏强有力的上下文建模手段，且模型训练难度增加。

本小结后续会按照：无上下文的检索模型、带上下文的检索模型的顺序进行展开。详细介绍我们在实验过程中使用的模型和其原理。

### 4.1.1 无上下文的检索对话模型

无上下文的检索模型只使用用户最近一句话作为输入，来预测客服下一句回复，故而对于模型来讲，一个包含多次用户-客服交互的对话，可以分解为的多个$\{u, h\}$二元组的集合，其中 u 为用户的一句输入，h 为紧跟其后的客服输入。这些二元组都是真实人类对话产生的内容，因此我们认为在这些二元组中，H就是用户输入为U时，所对应客服的正确回复。

从给定数据集中根据用户的输入检索合适的回复有两种思路。

一种是使用查询语句，即用户输入$u_{new}$，同语料库中的所有用户输入U进行匹配，找到同$u_{new}$最相近的$u_i$，并取出其所对应的回复$h_i$作为最佳匹配回复，即

$u_{new} \rightarrow u_i \rightarrow h_i$。这种思路利用了用户和客服问答之间的隐性映射关系，但是由于匹配$u_{new}$和$u_i$缺乏合适的标注数据集，难以将其设计成为一个有监督的学习任务，因此一般使用无监督的方法实现。

另外一种思路是，直接通过用户的输入$u_{new}$去匹配语料库里所有的客服回复H，从中选出得分最优的回复。即$u_{new} \rightarrow h_i$。这种方法的优点在于：历史对话记录无需太多处理，就是一个天然带标注的语料库，可以用有监督的方式训练模型，同时模型的评价也相对比较容易。模型在测试和实际使用过程中，可以人为的增加或修改回复，可以用来进行回复标准化，语法错误清除等工作。其预测过程可以描述为：

- 给定数据集:$D = \{(u_i, h_i)\}, u_i \in U, h_i \in H$
- 输入：用户的上句对话$u_{new}$
- 输出: $h_{cand} = \arg\max_{H} Sim(u_{new}, h_i)$

本文使用$u_{new} \rightarrow h_i$直接匹配的思路实现匹配模型。这种思路也经常用于 cQA（社区问答，Community Question Answering）问题的求解和建模当中。因此我们使用了 cQA 中一个比较流行的基于 RNN 的问答匹配模型：QA-LSTM 作为我们的无上下文的检索式对话模型。

该模型分为三个部分：1.线性映射，2.RNN 编码，3.相似性计算。如图 4 .1 中所示。

1）模型分别读取用户和客服的对话（经过分词处理）作为为输入

2）通过查询词向量矩阵，将句子中所有单词转换为其所对应的词向量，从而将每个原始的文本序列表示为词向量数组（H 和 U）

3）接着使用一个线性变换层（HL）对词向量数组中每个的词向量进线性变换

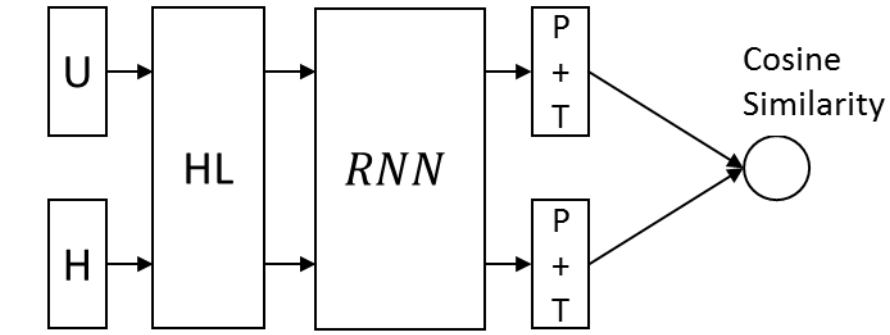
4）使用 Bi-LSTM 分别对两个句子进行编码

5）通过 Mean Pooling 和非线性层 Tanh（P+T）将句子编码压缩为定长向量

6）计算上一层两个向量的 Cosine 距离，并将其作为相似性得分

模型的训练使用

其中 Bi-LSTM 对比普通 LSTM，不同之处在于其通过使用两个 LSTM 单元以不同顺序对句子进行编码，一个以正常顺序读入句子，另一个将句子进行反转之后读入。因此，每个单词所对应的时间刻，都可以得到正向 LSTM 的输出和反向 LSTM 的输出两个向量，二者分别包含以该单词为界的前半部分句子和后半部分句子的信息，将两个向量拼接，就可以捕获输入完整信息。




。

图 4.1 QA-LSTM 模型

QA-LSTM 模型训练使用我们在绪论中提过的 Pairwise 方法。训练过程中对每个 Sample，都选取一个对话二元组$\{u, h^+\}$，其中u为某时刻用户的输入，$h^+$为其所对应的客服回复，作为正样本，同时随机选取一条不同的客服回复$h^-$作为负样本。损失函数如公式(4.1)所示。

$$\mathcal{L} = \sum_{(u_i,h_i^+)\in D} \sum_{h_j^-\in H, h_j^-\neq h_i^+} \max(0, \text{margin} - S(u_i, h_i^+) + S(u_i, h_j^-)) \quad \text{公式(4.1)}$$

## 4.1.2 带上下文的检索对话模型

在上一小节中，我们介绍了使用不考虑上下文的检索式对话模型用于客服回复推荐任务的基本思路。但是在人类正常的对话中，存在诸多表达不规范，信息省略的情况。如表 4.1 所示，有时用户的单句对话可能包含信息极少，如表中第 3 句“没”，单靠这一个字，并不能预测下一句应当回复什么内容，需要将“没”同完整对话历史进行对照，才能获得足够信息，最终预测可能的回复。

本小节，我们尝试将客服服务中的对话上下文纳入检索式对话模型建模中，以获得更好的回复推荐效果。

表 4.1 对话示例

| |
|---|
| 1. A:最近好像有点感冒，还流鼻涕 |
| 2. B:发烧吗? |
| 3. A:没 |
| 4. B:试试感冒灵 |

同上小节的模型相同，本节提出的会话模型通过给定上下文直接匹配答案，即上下文 → 回复，而非上下文 → 上下文 → 回复。考虑了上下文之后，匹配问题可以描述为：

给定数据集：$D = \{(u_{0,0}, h_{0,0}, \dots, u_{I,1}, h_{I,1}, \dots), \dots\}$，在其上学习一个匹配函数 F，使之根据对话样本在特定时刻的上下文状态：$(u_{i,0}, h_{i,0}, \dots, u_{i,j})$，从回复候选集 $a_{ij} = (a_{ij}^0, \dots, a_{ij}^m)$中挑选出最合适的回复内容：

1）在训练过程中，最佳的回复内容为$h_{i,j}$，即为客服的原始回复内容；

2）测试时则通过遍历给定候选回复集，分别计算每条回复同该上下文的相关性，并选取得分最优的一条作为最优回复。

在设计会话模型时，我们考虑到下面这些因素：

1）客服对话的目的都在于了解用户的需求并提供解决方案，其中存在较多提问，问题和其回复一般位置相邻，具有较好的局域性，但问题的提出并不限定于特定角色，用户和客服都有较大可能性提出问题；

2）对于同样一句话，来自用户和来自客服有可能会代表不同的意思，因此在编码过程中需要对用户和客服的身份进行区分；

3） RNN 在计算过程每一时刻的输入都依赖于前一时刻输出，因此非常容易出现依赖序列过长的问题，导致单次运行时间较长，出于实际使用考虑，需要将模型预测部分的依赖设计得尽可能简短，以提升预测的执行速度。

我们设计了一种带上下文的检索对话模型，简称为 HRR（Hierarchical RNN Ranker）,并将其用于客服回复任务中。我们设计的模型主体由两种编码器和一个相似度估量度量函数组成。编码器分别为：Utterance Encoder（单句编码器）和

Context Encoder（上下文编码器）。每个 Encoder 的功能都是通过 RNN 将输入的带有语义的变长向量数组，编码成为同样带有语义的定长向量。

Utterance Encoder 用来编码单条对话，将其转换为定长向量。其编码内容包括：用户的对话，客服的对话以及推荐的对话内容。

Utterance Encoder 输入为单个分好词的句子$w = (w_1, \dots, w_n)$，其中$w_i$为该位置单词所对应的 one-hot 编码。通过使用一个词向量索引层将这些单词的 one-hot 编码转换为对应的词向量表达。我们使用 Skip-gram 模型在爬取的垂直领域问答语料库上训练了 word embedding，并将其用于模型中的词向量查询层中作为初始权重，并在训练过程中保持对词向量索引层的更新。

Utterance Encoder 需要分别对不同角色对话进行编码，为了实现这些角色进行区分，我们在编码不同角色对话时，在词向量中添加了额外的标记。具体而言，给定单词的词向量$v$，和角色标记$r$，对二者进行级联获取带角色标记的词向量：$v' = [v; r]$。当 Encoder 编码内容是用户的输入时，$r = 0.0$，是客服的输入或者推荐回复时$r = 1.0$。

训练和预测过程中 Utterance Encoder 使用 RNN 对各个句子分别进行编码，并使用最后时刻的输出作为各个句子的表达。在训练过程中，我们使用了双层 GRU 网络作为 RNN 编码器。GRU 单元的表达性能同 LSTM 单元相近，但模型结构较为精简，运行速度更快。

令 Utterance Encoder 对某个对话历史$(u_0, h_0, \dots, u_j)$的编码结果为：$(v_0^u, v_0^h, \dots, v_j^u)$，对候选回复集合$(a_0, \dots, a_m)$的编码结果为:$(v_0^a, \dots, v_n^a)$。

Utterance Encoder 的结构如图 4-3 所示。

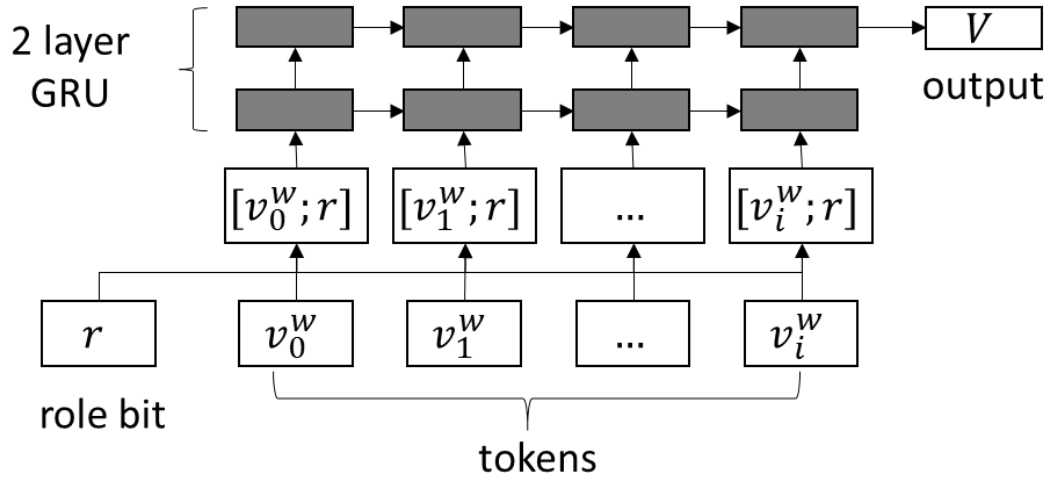



图 4.2 Utterance Encoder 结构示意图

Context Encoder 用来生成对当前上下文的向量表达。我们考虑到在正常对话中：完整的上下文中包含对话全部的背景信息，但是其中也存在非常多的冗余和失效信息，建模相对困难；最近一句话内容相对较少，有较大可能性包含回复所需内容，但所含信息不全。所以我们在建模时，将两种信息进行整合，共同组成最终上下文表达。

我们对 Utterance Encoder 的输出进行重组，将相邻两句对话的向量表达级联在一起，形成 Context Encoder 的输入：$([0; \mathrm{v}_0^{\mathrm{u}}], [\mathrm{v}_0^{\mathrm{u}}, \mathrm{v}_0^{\mathrm{h}}], [\mathrm{v}_0^{\mathrm{h}}, \mathrm{v}_1^{\mathrm{u}}], \dots, [\mathrm{v}_{\mathrm{i}}^{\mathrm{h}}, \mathrm{v}_{\mathrm{i}}^{\mathrm{u}}])$，其中0代表一个全零向量。将这些输入喂给一个独立的 GRU 网络，该网络会在用户和客服的每条对话所对应时刻，都输出一个向量代表该时间点全部上下文的表达：$(\mathrm{v}_0^{\mathrm{c}}, \dots, \mathrm{v}_{\mathrm{i}}^{\mathrm{c}})$。

将上面 GRU 网络的输出和对应时刻用户输出句子的表达做级联之后，得到对话历史的一个中间表达为：$([\mathrm{v}_0^{\mathrm{c}}; 0], [\mathrm{v}_1^{\mathrm{c}}; \mathrm{v}_0^{\mathrm{u}}], [\mathrm{v}_2^{\mathrm{c}}; \mathrm{v}_0^{\mathrm{h}}], \dots)$。之后使用一个线性映射层，对该中间表达进行维度变换，使其与候选回复的向量维度相同，方便进行相似度计算。因为本文的任务是预测客服回复，故我们在所有用户说话的时间点利用上下文表达预测可能回复。图 4.3 为 Context Encoder 的结构示意图。

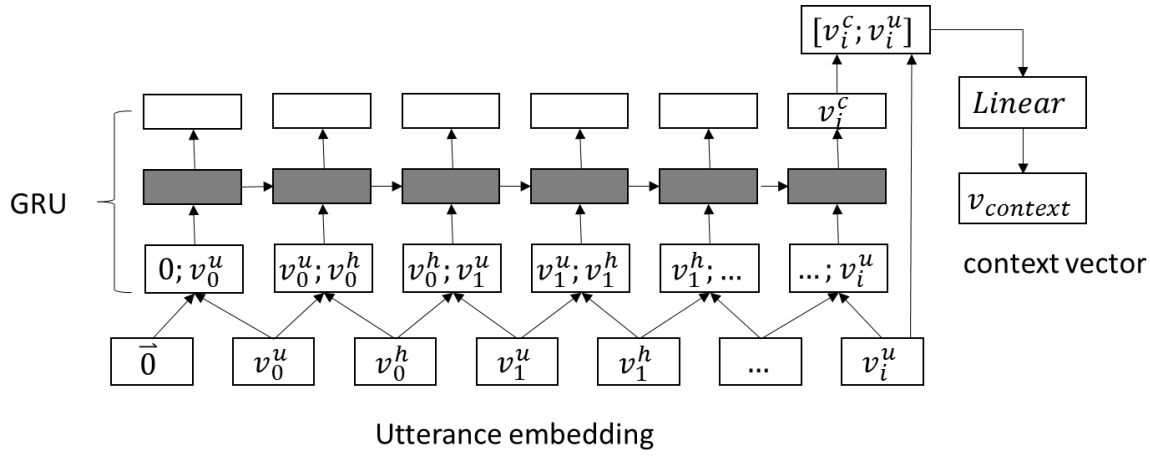



图 4.3 Context Encoder 的编码过程示意图

在获得上下文和候选回复的向量表达之后，我们使用 Cosine 距离计算各个候选回复同该上下问的匹配程度。使用 Pairwise 的方法对模型进行训练，训练过程中单个时间点的预测训练目标为最小化目标函数 hinge 损耗，如公式(4.2)所示。

$$\text{minimize } \max(0, \text{margin} - \text{Sim}(\text{c}, \text{a}^+) + \text{Sim}(\text{c}, \text{a}^-) \qquad \text{公式(4.2)}$$

其中c为模型对该时间点上下文的向量表示，$\text{a}^+, a^-$分别为正确和错误的回复内容的向量表示，Sim()函数为 cosine 距离函数。

本节提出的模型具有一个特点，在同一个对话中，Utterance Encoder 和 Context Encoder 的计算中间结果具有可复用性，即在同一个对话中进行多次预测时，时间点靠后的预测可以使用时间点靠前的预测的计算结果。基于这个特点考虑，模型单个 Sample（一个完整对话）的损失为该 Sample 中各个预测的损失之和。

### 4.1.3 使用意图信息的改进

在实验过程中，我们发现上下文排序模型给出的最优回复，存在一个出现次数较多的问题：模型容易给出和用户咨询意图无关的回复。经过分析，发现这种情况通常发生在当用户对问题描述不够标准的情况下，如拼写错误。我们认为其产生原因有两个方面：1）上下文信息过多，模型从冗余信息中找出用户意图的能力不足；2）用以训练的对话语料不足。

针对上面的问题，本文在之前提出的模型上进行了改进，得到了一个新模型，用 HRR+Intention 来表示。使用附加模型，将用户在对话中对其基本咨询意图描述（例如遇到了什么问题，需要什么帮助等简短描述）的内容部分进行编码，得到一个包含用户意图分布的向量。通过将该意图向量加入上下文的编码过程中，来完成用户的意图信息来约束模型应答选取什么样的回复。

在客服回复推荐实验中，我们使用的对话语料是来自在线健康咨询和问诊的语料，因此为了建立合适的意图模型，我们使用了 Ask39 社区问答数据集进行训练，该数据集中的每个提问都有一组疾病标签信息，标签为和该提问相关的疾病名字，数量为一个或多个。语料库中总共包含了超过 6759 种标签，但存在明显的长尾效应，我们在选取了出现次数最多且同对话语料相关的前 300 个标签，并删除了其他标签，和处理后无标签存在的问题。

我们将建立用户咨询意图的过程建模为一个多分类模型。给定输入一个自然语言描述，输出其在各个类上的能量分布，即描述用户想要咨询哪种病的可能性分布。我们使用一个双向 GRU 将描述语句编码为定长向量，之后用一个线性层和 Sigmoid 非线性层将其映射为长度为 300 维的向量，该向量表示各个标签的能量分布。使用交叉熵作为训练过程中的损失函数。

因为本文使用的对话语料数据集本身的特点，我们在实验过程中，通过适当的数据预处理和筛选，使得用户的第一句话为对其想要咨询问题的一个基本描述，其中包含用户基本的咨询意图信息。我们利用前面描述的分类模型对数据集进行了二次处理：每个样本将用户的第一句话作为输入，计算出分类模型 Sigmoid 层的输出，作为该对话中的意图向量。

检索模型的改动如下：在 Context Encoder 每次读取输入（Utterance Encoder 的输出组合）时，将原始输入和意图向量进行级联，形成新的输入，模型其余结构维持不变。

### 4.1.4 实验结果和分析

本文基于 Tensorflow 框架对提到的三种检索式对话模型进行了实现。为了分

析测试本节提出和使用的检索式对话模型的性能，本文构建了一个客服回复推荐数据集。该数据集包含 30,588 条皮肤病相关的在线健康咨询记录。每条咨询记录均包含用户和医疗从业人员（后续称为：医生）的一次完整对话。在数据预处理过程中，我们进行了如下操作

- 数据脱敏，清除了数据中用户和医疗人员的姓名、ID、通讯信息、咨询时间等敏感信息。
- 将原始语料中的一些标准化回复进行了删除和替换，减少了这部分回复在语料中所占比例
- 对同一角色连续发言的情况进行合并，使处理后的对话为：用户开始，两人交替发言的格式

将数据按$\{70\%, 10\%, 20\%\}$的比例划分为训练集，验证集和测试集，其中验证集和测试集中每条医生的回复均有 1 个原始回复和 99 个随机选取的医生回复构成的候选集。训练过程中取验证集上得分最高的模型进行测试。

我们使用两个指标对模型进行评估：P@1，Hit@3，MRR。其中 P@1 表示排名第一的推荐内容是正确回复的比例。Hit@3 为排名在前三的正确答案的比例。MRR 正确答案排名的倒数平均。三个指标都是值越高代表模型性能越好。

实验结果如表 4.2 所示。

表 4.2 检索式对话模型实验结果

| 模型 | P@1 | Hit@3 | MRR |
|---|---|---|---|
| QA-LSTM | 14.59% | 32.36% | 29.31% |
| HRR | 20.69% | 38.73% | 35.41% |
| HRR+ Intention | 21.72% | 40.40% | 36.23% |

在训练过程中使用参数配置如下。词向量层初始化均使用在医疗社区问答语料上用 Skip-Gram 模型训练的 300 维词向量。QA-LSTM 模型中：隐层权重的形状为$300 \times 300$ LSTM 单元的维度为 256 维；margin 设定为 0.009，学习率为 0.1，l2 正则项权重 0.0001。HRR 模型中：Utterance Encoder 的 GRU 单元维度 256，

Context Encoder 的 GRU 单元维度 256，margin 为 0.3，学习率初始值 0.5，l2 正则项权重 0.000005。HRR + Intention 模型中，预训练的多分类模型中 GRU 单元维度 256，其余参数同 HRR 模型一致。所有模型均使用 mini batch 梯度下降方法进行训练，训练过程中 batch size 设置为 64。

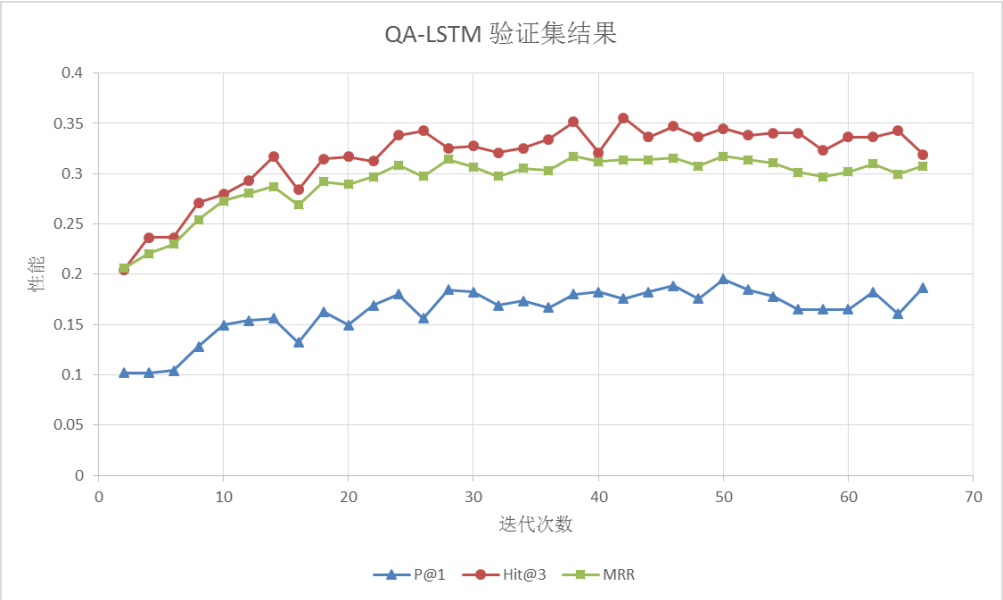



图 4.4 QA-LSTM 验证集测试记录

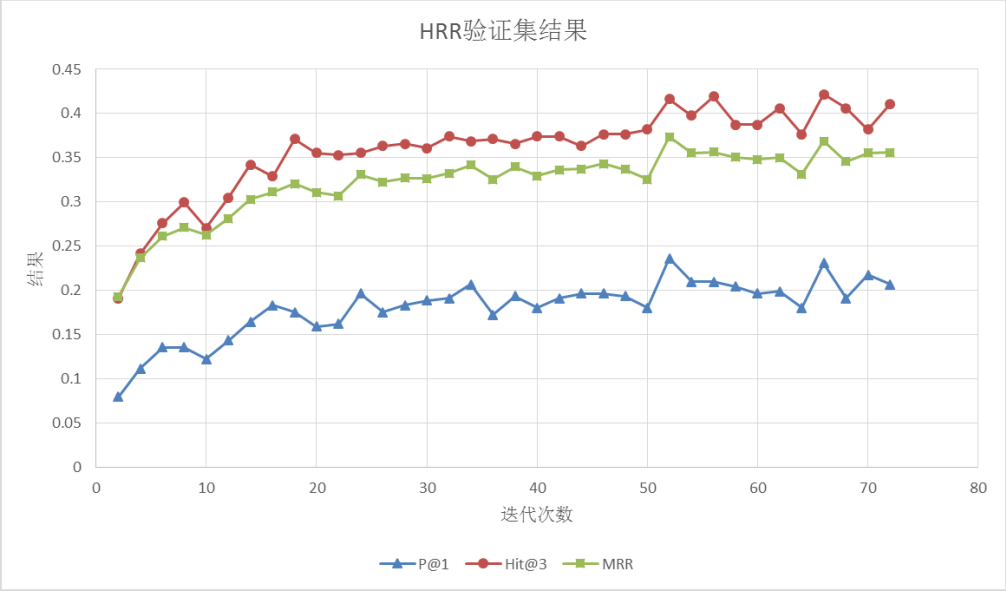



图 4.5 HRR 验证集测试记录

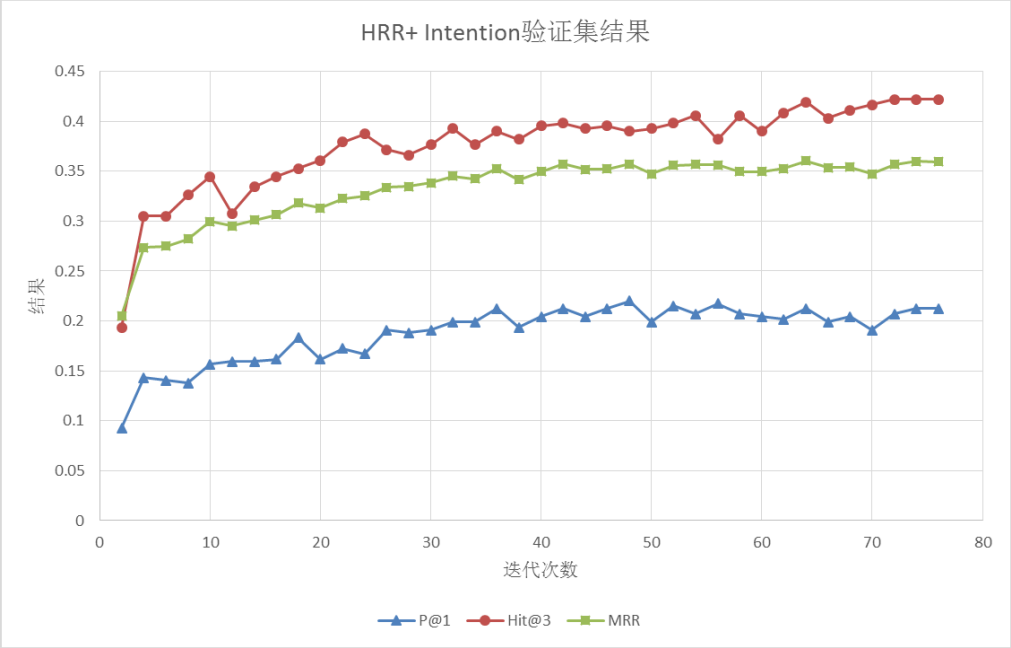



图 4.6 HRR+Intention 验证集测试记录

图 4.4 到图 4.6 分别为本文提到的三种检索式对话模型训练过程中，对验证集上的测试结果记录。

从实验结果中我们可以看出，HRR 模型相比 QA-LSTM 模型的性能提升比较明显，说明上下文是客服回复推荐过程中的重要信息，对上下文建模是正确的思路。通过对比 HRR 和 HRR+Intention 模型，可以发现 HRR+Intention 模型的性能有小幅度提升，说明对用户原始咨询的意图进行建模有效果。但性能提升幅度较小，我们分析认为，当前分类模型使用的语料库质量较差，后续可以使用更好的标注语料来实现性能提升。

**实例分析**

本文对改进后模型的测试结果进行了分析。

表 4.3 为测试集上一个的例子，其中“User”代表用户所说的话，“Host”代表人工客服所说的话，“Suggest”代表模型从候选集中选出的推荐回复。从这个例子中我们可以看出，虽然模型预测客服对话的准确性存在较大的可提升反省，但是其推荐回复基本同用户的咨询相关：用户询问痤疮的相关问题，推荐回复中出现了

维 A 酸，其为治疗痤疮的常用药品之一。但是在这个例子中，模型未能准确分别出客服的推荐意图：不做任何推荐。

表 4.3 回复推荐测试实例-1

| Role | Content |
|---|---|
| User | 脸上长痘 |
| Host | 好的 |
| Suggest | 好的 |
| | |
| User | 脸上有很多小痘痘有的看不到但是摸得到，挤出来是白色的 小颗粒 |
| Host | 尽量别用手挤 |
| Suggest | 痤疮建议使用外用软膏 |
| | |
| User | 有时候那个痘痘还会痛，不挤的话就会形成毛孔一样的 |
| Host | 那尽量顺其自然吧 ， 你您这个年龄长痘痘也是正常的 |
| Suggest | 吃点清热排毒的痤疮药吧 |
| | |
| User | 就看起来有好多毛孔，一挤还是能挤出来 |
| Host | 痘痘不能挤的，会留下疤痕 |
| Suggest | 痘痘不能挤的，会留下疤痕 |
| | |
| User | 可是痘痘多满脸红红的不好看 |
| Host | 那也没办法，先忍着 |
| Suggest | 平时可以补充维生素 A，局部用点润肤霜，可以用点去角质 的药物，比如维 A 酸、水杨酸软膏可以改善 |

表 4.4 为改进后模型测试过程中的另外一个例子。在这个例子中，模型尝试推荐“工作环境好吗？住宿环境等”，其意图同客服后来实际说的“您是做什么工作

的 ？”意思相近，意图均为判断用户的症状是否由其最近生活环境所引起。虽然出现先后次序不同，但是这个例子说明模型学习到了一定的会话策略。但是模型最后推荐的“湿疹不会蜕皮”，说明其对上下文建模部分还存在改进之处。

表 4.4 回复推荐测试实例-2

| Role | Content |
|---|---|
| User | #图片链接# 湿疹 ？ |
| Host | 看到了 |
| Suggest | 多久了？用药了吗？ |
| | |
| User | 湿疹吧 ？ |
| Host | 就一只手吗 ？ |
| Suggest | 痒吗 |
| | |
| User | 双手 手背 别的地方没有 |
| Host | 恩恩 |
| Suggest | 有水泡 ？是全身吗 ？ |
| | |
| User | 这两天出汗就有了 |
| Host | 痘痘不能挤的，会留下疤痕 |
| Suggest | 工作环境好吗？住宿环境等 |
| | |
| User | 是不是湿疹？ |
| Host | 您是做什么工作的 ？ |
| Suggest | 哦哦，应该不是，湿疹一般不会蜕皮 |

## 4.2 基于产生式的深度对话模型研究

用产生式深度会话模型来进行的回复推荐的基本思想是，先将会话的背景信息进行编码，再直接根据这个编码生成回复内容。同检索式模型相同是：二者都存在一个编码过程，将对话背景信息进行编码。在大多数情况下，模型的编码过程可以通用。二者的不同之处在于：检索模型的通过对已有的候选回复进行排序得到最优回复，而产生式模型使用一个 RNN 来动态生成回复内容。

根据是否只使用最近一句话作为背景信息，我们可以将会话模型的分为不考虑上下文的产生式会话模型和考虑上下文的产生式会话模型两种。两类模型的优缺点同之前介绍相同，此处不再赘述。

在本文中，我们在上一小节提出的检索式模型的基础上，利用其上下文编码的部分，重新构造了对应的产生式对话模型，并同一个产生式模型 baseline 进了对比。

### 4.2.1 无上下文的产生式会话模型

无上下文的产生式会话模型只使用最近一句话作为背景信息，对背景信息进行建模之后，使用 RNN 动态产生回复内容。这种模型通常应用于开放域对话场景中。在本文中，我们尝试使用这种模型来完成客服回复推荐任务。

回复推荐过程可以表述为：给定数据集$d = \{(u, h)\}$，其中u为用户在某个时刻的输入，h为客服对应的回复，我们尝试在这些数据上学习一个模型，使之读入u后，生成尽可能和h相似的内容。

我们使用 Vinyals O[16]在相似背景下使用的 Seq2Seq 模型作为用来生成回复的产生式会话模型。图 4.7 为其结构示意图。其中 Encoder 将单词序列转换为词向量，并通过一个单层 LSTM 将其编码为向量形式，作为 Decoder LSTM 的初始状态。Decoder 在初始时刻读入一个特殊标记的符号EOS，作为占位符，每个时间片，Decoder LSTM 的输出被转换为一个确定的单词符号，并作为下一时间刻的输入，直到 Decoder LSTM 的输出被映射为 EOS 符号，或输出序列长度超出阈值。

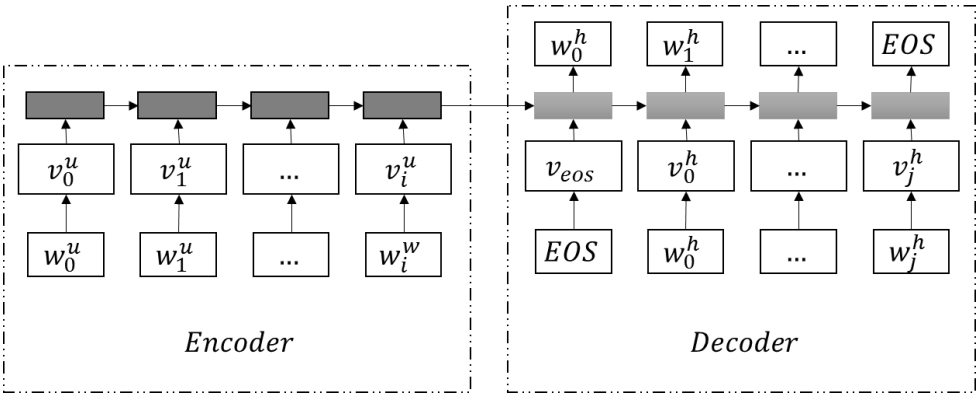



图 4.7 Seq2Seq 模型结构示意图

在回复推荐任务中，用户和客服的语言相同，故 Seq2Seq 模型中的 Encoder 和 Decoder 共享符号集合和词向量。训练过程中使用 Perplexity 作为损失函数，单个样本的损失函数计算为：

$$\mathcal{L} = 2^{-l};\ l = \frac{1}{M}\sum_{i=1}^{m} log\ p(h_i) \qquad \text{公式(4.3)}$$

其中m为输出序列的长度，$p(h_i)$为在预测时正确单词获得的能量值。

## 4.2.2 带上下文的产生式会话模型

参考 4.1.2 节中提出的 HRR 模型的编码部分和 Serban 在《Building end-to-end dialogue systems using generative hierarchical neural network models》[11]中提出的 HRED（Hierarchical Encoder Decoder RNN），我们设计了一个用于的客服回复推荐任务的产生式会话模型，其能够利用单个对话中的全部历史信息，生成客服可能需要的回复内容。

该模型的框架如图 4.8 所示。其中 Utterance Encoder（单句编码器）和 Context Encoder（上下文编码器）结构同 HRR 模型中对应部分相同。Context Encoder 的输出与用户最后一句话的编码结果做级联，并通过线性层映射为指定长度，作为 RNN 解码器的初始状态值。在用户的每个输入时刻，均进行一次回复生成。因为本模型的上下文编码和单句编码信息可以在不同时刻的预测中可以复用，故在训练时，以一个完整的对话作为一个 Sample，其损失函数为各次预测回复时计算的

Perplexity 损失之和。

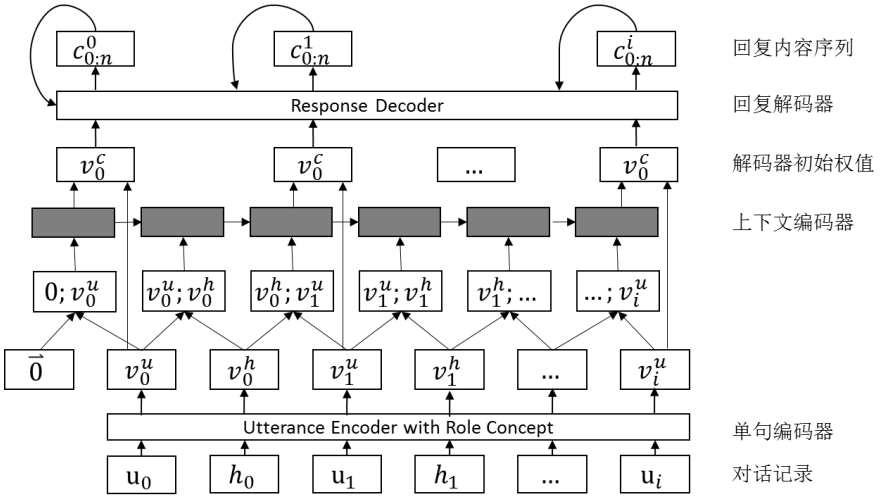



图 4.8 带上下文的产生式会话模型结构

因为模型复杂度高且训练困难，我们在计算损失函数时使用带采样的 softmax 函数替代原始的 softmax 函数，以提升训练速度。

## 4.2.3 实验结果和分析

为了测试模型的效果和性能，我们基于 Tensorflow 框架对本节的两个模型进行了实现，并在之前构造的在线健康咨询对话数据集上进行了实验。数据仍然按照约{90%, 9%，1%}的比例划分为训练集、验证集和测试集（300 条）。评测通过人工对模型产生的内容进行判断，并给出回复内容{适合，不适合}当前上下文的标签。测试结果如表 4.5 所示。

表 4.5 产生式对话模型实验结果

| 模型 | 合适比例 |
|---|---|
| Seq2Seq | 15.21% |
| Seqs2Seqs | 19.35% |

实验中使用医疗垂直语料库上通过 Skip-gram 模型训练的 300 维词向量对所

有模型的词向量层进行初始化，同时在训练过程中也保持词向量更新。Seq2Seq 模型的 encoder 和 decoder 均使用单层 LSTM 作为 RNN 隐层更新单元，其隐层维度为 256。Seqs2Seqs 模型中使用双层 GRU 作为各个 RNN 的隐层更新单元，维度均为的 256。使用 mini batch 梯度下降进行训练，batch size 设为 64。其余参数采用了默认设置。

表 4.6 回复推荐测试实例-3

| Role | Content |
|---|---|
| User | 脸上有痘 |
| Host | 哦 |
| Suggest | 克林霉素磷酸酯凝胶。就这个药就可以 |
| | |
| User | 用什么药 |
| Host | 多久了？ |
| Suggest | 您好。您的年龄是？ |
| | |
| User | 一年 |
| Host | 治疗过没有？ |
| Suggest | 哦 |
| | |
| User | 没有 |
| Host | 嗯。我给亲推荐一些药物吧 |
| Suggest | 可以的 |
| | |
| User | 好的。[表情] |
| Host | 夫西地酸乳膏 |
| Suggest | 可以的 |

表 4.6 为 Seq2Seq 模型在测试集上的一个例子。不难看出该模型非常容易生成一些出现频次较高但无实际意义的回复，如： “哦”、“好的”、 “可以的”。且生成的回复内容和用户的原始提问无太大相关性，所以性能较差。

表 4.7 回复推荐测试实例-4

| Role | Content |
|---|---|
| User | 头上一块一块的头皮屑，粘在头发上 |
| Host | 有没有小丘疹， |
| Suggest | 你好。有没有掉发？ |
| | |
| User | 好几年了吧。没有 |
| Host | 使用过药物吗 |
| Suggest | 哦。六味地黄丸可以吃的。和天麻首乌一起吃 |
| | |
| User | 就是头皮屑。没有 |
| Host | 可以使用复方酮康唑洗剂。这个去头屑的效果比较好 |
| Suggest | 用复方酮康唑洗剂 |
| | |
| User | 是洗发露吗 |
| Host | 是的，。一天洗一次就可以 |
| Suggest | 可以的 |
| | |
| User | 就跟正常的洗发露一样使用？ |
| Host | 是的，去头屑的效果很好 |
| Suggest | 可以的 |

表 4.7 为本文提出的带上下文的产生式模型在测试集上的一条结果。从例子中可以看出，虽然该模型仍然会产生一些高频但无意义的回复，但相比 Seq2Seq

模型，其产生回复的质量有一定提升。

从实验结果中我们可以发现，产生式模型虽然可通过引入上下文信息获得更好的产生效果，但总体性能仍然较差，提升有限。不论是否引入上下文信息，产生式模型都受困于容易产生高频无意义回复的问题，我们认为其问题在与产生式模型函数存在缺陷，目前的目标中只考虑了字与字的严格对应约束，但缺乏语义相关性的约束。

## 4.3 本章小结

本章围绕使用深度对话模型来完成客服回复推荐任务展开，详细介绍了深度对话模型的不同实现思路和分类。并重点在检索式的对话模型上进行了探索研究，设计、实现、改进并测试了对上下文信息进行建模的检索式对话模型。之后在其基础上设计了对应的产生式对话模型，并对这些模型和 baseline 模型在新构造的在线健康咨询数据集上进行了实验和分析。

# 第5章 总结和展望

## 5.1 总结

随着互联网经济的不断发展，在线购买商品和服务已经逐渐成为人们的生活习惯，这种潮流为许多行业和工作带来改变，其中包括在线客服。电商平台的成交量日益增多，对在线客服的服务质量和服务效率不断提出更高的要求，与此同时随着经济的发展，人力成本也在不断上升。这些因素使得如何在现有基础上提升客服的服务质量和速度的成为一个迫切的需求。

在此基础上，本文对现有的对话模型展开研究，通过对对话模型进行改进和调整，并将其应用于客服辅助服务中。

本文的主要贡献在于：

1）客服在服务过程中需要查询或参考相关的专业知识，来完成更高质量的服务。本文设计了一种知识库查询服务。该服务接受用户的自然语言提问作为输入，通过分词和词的归一化操作，将该自然语言提问转换为{知识库实体关键词，待查询属性}二元组，之后使用该二元组从知识库中检索对应的知识条目返回给用户。在该过程中，本文从网络上爬取了垂直领域的社区问答语料和专业知识，构建了实验用的专业知识库，并根据知识库手工设计了查询模板，将查询服务实现为Web API。针对模板匹配过程中出现的关键词匹配实现问题，本文通过对常用算法进行分析，提出一种多轮同义词扩充算法，其可用于专业领域中的同义词挖掘，相比原始算法具有更好的性能。

2）针对如何提升人工客服工作效率的问题，本文提出了使用基于深度学习的对话模型来完成客服回复推荐任务。本文通过分析客服回复推荐任务的特点和对话模型的部分最新模型展开探索。本文在实验过程中，分别对基于检索式的深度对话模型应用于客服回复推荐和基于产生式的深度对话模型应用于客服回复推荐进行研究。在检索式的模型中，本文先实现了一个 Baseline 模型 QA-LSTM，之后设计了一种适用于客服回复推荐的上下文编码方式，设计实现了对应的检索

式深度对话模型。并根据实验过程中的 case 分析，提出了一种模型，使用用户咨询意图信息对推荐效果进行改善。在产生式模型中，本文先实现了基础的 Seq2Seq 模型，用来作为对照，并根据本文提出的上下文编码方式提出并实现了对应的产生式对话模型。通过实验分析，我们发现本文提出的上下文建模方法对回复的推荐有提升效果。

## 5.2 展望

本文的目标在通过对对话模型的研究和改进，从而实现对专业客服的辅助支持，提升其工作效率和服务质量。针对如何提高客服服务质量的问题，本文提出一种基于模板匹配的知识库查询服务。针对如何提升客服服务效率的问题，本文提出了两种方法来使用深度对话模型完成客服回复推荐。由于自然语言本身的复杂性和相关模型技术还不够成熟，本文认为还有下面几个方面可以值得研究，以期有所突破。

1）训练高质量的垂直领域词向量

词向量已经成为 NLP 任务中重要的基础，对相关任务性能有着明显影响。目前已有的词向量训练方法均需要使用大规模语料库作为基础，如维基百科，Google New 等。但是在垂直领域内， 相关语料的获取并不容易，难以得到大规模的语料库来训练词向量，因此往往导致词向量的性能较低。探索如何使用小语料库的情况下也能训练出高质量词向量是一个需要去尝试的方向。

2）自动产生模板

本文提出的知识库查询服务主要针对个性化较小的知识库设计，因为其中涉及的属性较少，所以采用了手工设计模板进行匹配。但是这种方法并不适合于 DBpedia 等内容较丰富的知识，因为其中实体和属性都十分丰富，个性化程度较高，通过手工模板的方法进行匹配成本较高。因此如何根据语料自动生成对应的问模板也是一个值得研究的方向。

3）提升对话模型训练效率

随着对深度对话模型研究的深入，将上下文的信息纳入模型成为研究的一种

趋势。对上下文进行建模理论上可以使模型获得更好的表现效果。但是同时也使得模型的复杂度大大增加。输入输出之间的依赖关系较长，训练时收敛较慢。因此如何通过合理的设计，来减小模型复杂度，使得模型在训练过程中容易收敛是一个重要的问题。对模型实际环境进行应用影响较大。

4）产生式对话模型的优化

目前产生对话模型存在的主要问题是：容易产生常见但是无意义的回复。本文通过分析认为，这是可能使用 Perplexity 作为损失函数的结果，这种损失函数只考虑生成内容和预期内容字符串之间的严格对应关系，而没有考虑二者之间的语义相似性也应当作为评价产生回复是否合理的标准。但是苦于缺乏合适的评价语义相似性的方法和标准，目前还没有能完全解决该问题的损失函数提出。本文认为设计更为合理的损失函数是产生式对话模型未来优化的方向之一。

# 参考文献

[1] 朱旺南. 基于本体的自动问答客服系统研究[D]. 青岛理工大学, 2012.

[2] https://en.wikipedia.org/wiki/Chatterbot

[3] https://en.wikipedia.org/wiki/Turing_test

[4] https://en.wikipedia.org/wiki/ELIZA

[5] https://en.wikipedia.org/wiki/PARRY

[6] https://en.wikipedia.org/wiki/Jabberwacky

[7] https://en.wikipedia.org/wiki/Dr._Sbaitso

[8] Wallace R S. The anatomy of ALICE[M]//Parsing the Turing Test. Springer Netherlands, 2009: 181-210.

[9] Shekarpour S, Höffner K, Lehmann J, et al. Keyword query expansion on linked data using linguistic and semantic features[C]//Semantic Computing (ICSC), 2013 IEEE Seventh International Conference on. IEEE, 2013: 191-197.

[10] Zhang H, Kishore R, Sharman R, et al. Agile Integration Modeling Language (AIML): A conceptual modeling grammar for agile integrative business information systems[J]. Decision Support Systems, 2007, 44(1): 266-284.

[11] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J]. journal of machine learning research, 2003, 3(Feb): 1137-1155.

[12] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[13] Gers F A, Schmidhuber E. LSTM recurrent networks learn simple context-free and context-sensitive languages[J]. IEEE Transactions on Neural Networks, 2001, 12(6): 1333-1340.

[14] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]//Advances in neural information processing systems. 2014: 3104-3112.

[15] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations

using RNN encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.

[16] Vinyals O, Le Q. A neural conversational model[J]. arXiv preprint arXiv:1506.05869, 2015.

[17] Shang L, Lu Z, Li H. Neural responding machine for short-text conversation[J]. arXiv preprint arXiv:1503.02364, 2015.

[18] Li J, Monroe W, Ritter A, et al. Deep Reinforcement Learning for Dialogue Generation[J]. arXiv preprint arXiv:1606.01541, 2016.

[19] Li J, Galley M, Brockett C, et al. A Persona-Based Neural Conversation Model[J]. arXiv preprint arXiv:1603.06155, 2016.

[20] Xing C, Wu W, Wu Y, et al. Topic Aware Neural Response Generation[J]. arXiv preprint arXiv:1606.08340, 2016.

[21] Serban I V, Sordoni A, Bengio Y, et al. Building end-to-end dialogue systems using generative hierarchical neural network models[C]//Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16). 2016.

[22] Serban I V, Sordoni A, Lowe R, et al. A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues[J]. arXiv preprint arXiv:1605.06069, 2016.

[23] Inaba M, Takahashi K. Neural Utterance Ranking Model for Conversational Dialogue Systems[C]//17th Annual Meeting of the Special Interest Group on Discourse and Dialogue. 2016: 393.

[24] Yan R, Song Y, Wu H. Learning to Respond with Deep Neural Networks for Retrieval-Based Human-Computer Conversation System[C]//Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM, 2016: 55-64.

[25] Xu Z, Liu B, Wang B, et al. Incorporating Loose-Structured Knowledge into LSTM with Recall Gate for Conversation Modeling[J]. arXiv preprint arXiv:1605.05110, 2016.

[26] Trotman A. Learning to rank[J]. Information Retrieval, 2005, 8(3): 359-381.

[27] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[28] http://ask.39.net/

[29] http://www.xywy.com/

[30] Manning C D, Surdeanu M, Bauer J, et al. The Stanford CoreNLP Natural Language Processing Toolkit[C]//ACL (System Demonstrations). 2014: 55-60.

[31] https://github.com/NLPchina/ansj_seg

[32] https://github.com/fxsjy/jieba/

[33] Deerwester S, Dumais S T, Furnas G W, et al. Indexing by latent semantic analysis[J]. Journal of the American society for information science, 1990, 41(6): 391.

[34] Pennington J, Socher R, Manning C D. Glove: Global Vectors for Word Representation[C]//EMNLP. 2014, 14: 1532-43.

[35] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of machine Learning research, 2003, 3(Jan): 993-1022.

# 攻读硕士学位期间主要的研究成果

**专利申请**

[1] 王东辉，**梁建增**，李亚楠，基于深度学习的客服回复推荐方法（申请号：201710100757.1）

[1] 王东辉，熊奎，李亚楠，蔺越檀，孙欢，黄鹏程，洪高峰，徐灿，梁建增，庄越挺，一种基于本体内容的 SPARQL 查询语句生成的系统及方法（申请号：201410852126.1）

[2] 王东辉，孙欢，李亚楠，蔺越檀，熊奎，黄鹏程，洪高峰，徐灿，**梁建增**，庄越挺，一种基于任务难度与标注能者能力的众包标注数据整合方法（申请号：201410850691.4）

# 致谢

时光匆匆，如白驹过隙，在浙大读研的两年半已经快要结束。从初入浙大时对人生略感迷茫，走到今天对未来的规划已渐渐明确。期间的变化离不开陪伴我的师长、同窗和亲友的关心和帮助，在这里我想衷心的感谢和祝福他们。

我的导师王东辉教授，用他严谨的科研态度和实事求是的治学思路，为我树立了榜样。王老师培养了我的科研能力，带领我进入机器学习的领域，在学术研究过程中，给我了许多宝贵的意见和建议。在生活方面，王老师用他乐观豁达的人生态度影响我的成长，在许多问题上独特理解对我思考问题的方法有很大启发和影响。

同时还要感谢实验室的师兄师姐师弟师妹们，大家共同营造了一个和谐温馨的实验室环境。总是分给我们各种好吃零食的李亚楠师姐、未来成为同事的熊奎师兄、已经成为人生赢家的孙欢师姐、喜欢带着我们这帮学弟健身的蔺越檀师兄、老是吵着要到我们去留食的洪高峰师兄、跑去北京吸雾霾的黄鹏程师兄、羽毛球很厉害的徐灿、喜欢自称 pzy 大人的庞章阳、前端高手夏子琪、以及同样喜欢看动漫的胡焕行和王纪东。感谢 608，感谢 DCD，感谢浙大，该给我这段美好的经历。

感谢家人和朋友，感谢他们关爱和支持。和他们的沟通，让我一点点勾勒出自己未来的发展规划，让我在困难的时候坚持下去。

最后感谢，我所遇到的所有优秀的人，感谢他们让我看到更大的世界。

梁建增

2017-01 于求是园