

1. 概述	4
2. 需求分析	5
2.1 目标.....	5
2.2 要求.....	5
3. 核心设计	6
3.1. 预处理模块.....	7
3.2. 意图识别模块.....	7
3.3. 状态转移模块.....	7
3.4. 自然语言生成模块.....	8
4. 数据库设计	9
4.1. 关键词表 (KEYWORD)	9
4.1.1. 意义.....	9
4.1.2. 属性说明.....	9
4.1.3. 表结构.....	9
4.2. 菜单表 (FOOD)	9
4.2.1. 作用.....	9
4.2.2. 属性说明.....	9
4.2.3. 表结构.....	10
4.3. 状态图—弧表 (GRAPH_ARC)	11
4.3.1. 作用.....	11
4.3.2. 属性说明.....	11
4.3.3. 表结构.....	11
4.4. 状态图—状态表.....	12
4.4.1. 作用.....	12
4.4.2. 属性说明.....	12
4.4.3. 表结构.....	12
4.5. 回答表 (REPLY)	13
4.5.1. 作用.....	13
4.5.2. 属性说明.....	13
4.5.3. 表结构.....	13
4.6. IDF 向量表 (IDF_VECTOR)	13
4.6.1. 作用.....	13
4.6.2. 属性说明.....	13
4.6.3. 表结构.....	14
4.7. 句式表 (SENTENCE_VECTOR)	14
4.7.1. 作用.....	14
4.7.2. 属性说明.....	14
4.7.3. 表结构.....	14
5. 功能拓展	15
5.1 句式拓展.....	15
5.2 节点拓展.....	15

6. 测试	15
6.1 单元测试.....	15
6.2 整体测试.....	16

Megrez 设计文档

V0.1

张帅

2017/10/13

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
0.1	Zhang Shuai	2017.10.12			

1. 概述

对话系统 (Spoken Dialogue Systems, SDSs) 是人机交互的核心, 对交互的应用至关重要。目前对话系统已经出现在很多复杂的环境中, 智能环境

(Intelligent Environments, Heinroth & Minker, 2013), 车载系统 (Geutner et al., 2002), 个人助手 (Janarthanam et al., 2013), 智能家居 (Kreber et al., 2004), 及人机交互 (Foster et al., 2014) 等都已经出现相应的应用。

当前的对话系统主要包括 5 个模块: 1) 语音识别 (ASR) 模块, 2) 口语理解 (SLU) 模块, 3) 对话管理 (DM) 模块, 4) 自然语言生成 (NLG) 模块, 5) 文本生成语音 (TTS)。本系统 (Megrez) 为对话系统引擎主要针对对话系统中 SLU、DM、NLG 三个模块进行开发封装。本文旨在介绍 Megrez 系统的主要设计思路及设计内容, 如下图所示。

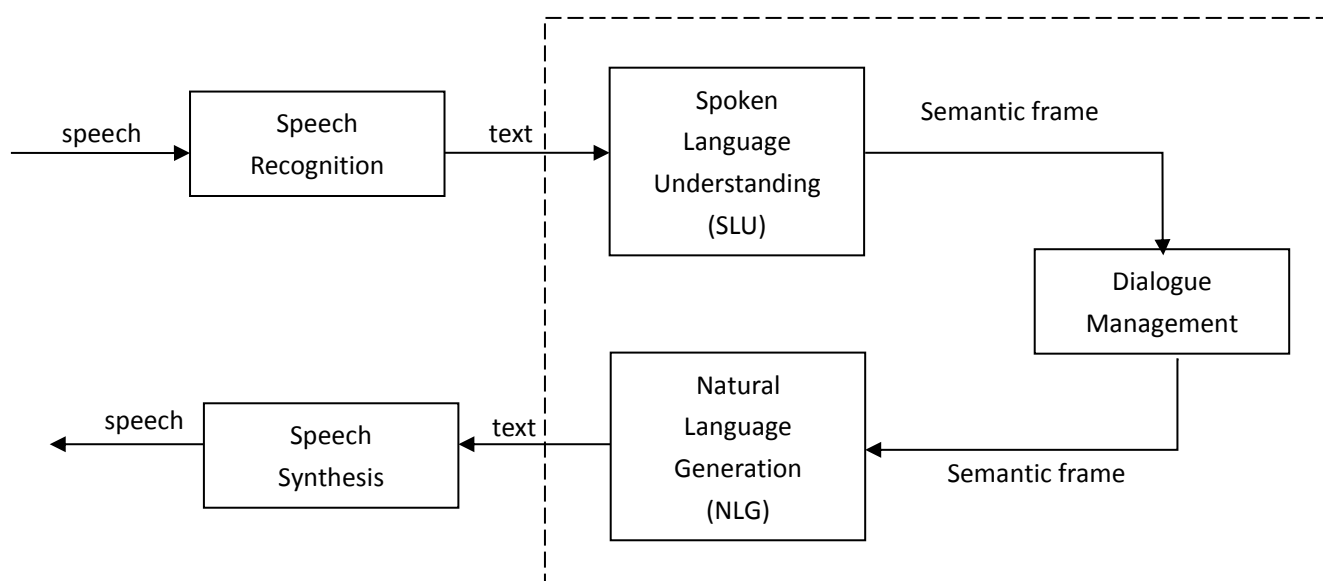


图 1 Spoken Dialogue Systems, SDSs

2. 需求分析

2.1 目标

实现 ASR 的输入接口和 TTS 的输出接口。

2.2 要求

对 ASR 转换的句子进行意图分析，从而做出正确的回应，达到完成订餐流程和获取正确订餐数据的目的。

1. 完成对句子意图的理解。
2. 完成对句式的匹配。
3. 完成订单的获取和自然语言的生成。

3. 核心设计

本系统主要包括四个模块，如图所示：

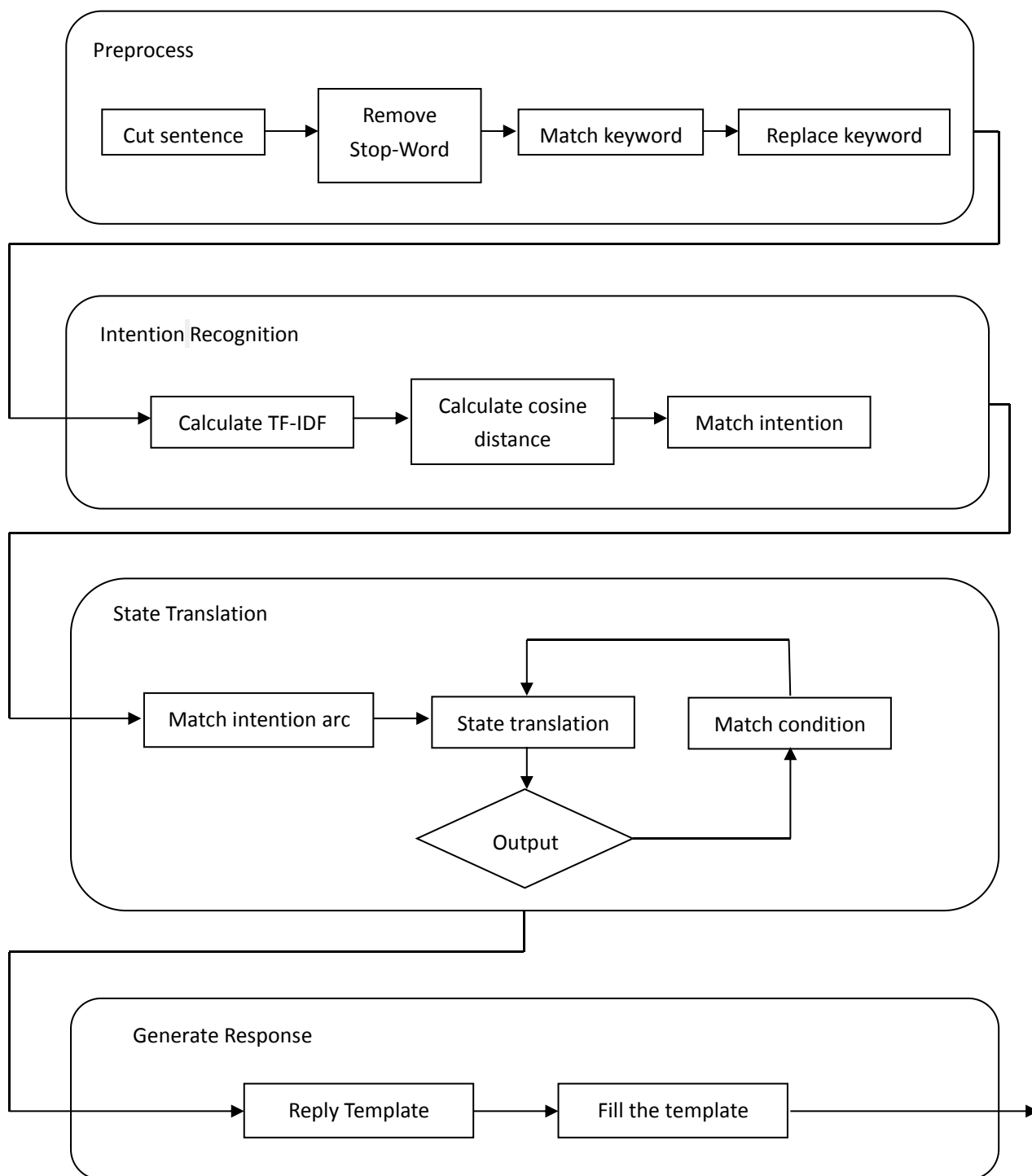


图 2 System Design

3.1. 预处理模块

用户的语音信号经过语言识别模块后转化为文字。本模块将对语句进行分词、去除停用词，匹配关键词，替代关键词等操作。经过这个步骤获得关键词对应及替换关键词后的抽象化语式。

eg. Input: “我想吃西红柿炒鸡蛋的菜”

=> 我 想 吃 西红柿炒鸡蛋 的 菜

=> 我 想 吃 西红柿炒鸡蛋 菜

=> 西红柿炒鸡蛋 <==> dish_name

=> 我 想 吃 <dish_name>

3.2. 意图识别模块

根据预处理阶段后的输出获得抽象化句式，通过 TF-IDF 进行计算，对比数据库中句式的 cosine 距离，距离越大越接近 1 即匹配度越高，从而获得数据库中句式对应的意图。

3.3. 状态转移模块

为了是系统更具有可扩展性，Megrez 设计定义了一套状态转移图，根据设计规范可以自定义对话流程。状态转移图主要有 State 和 Arc 组成，其具体属性参见数据库表。部分流程图如下图所示。

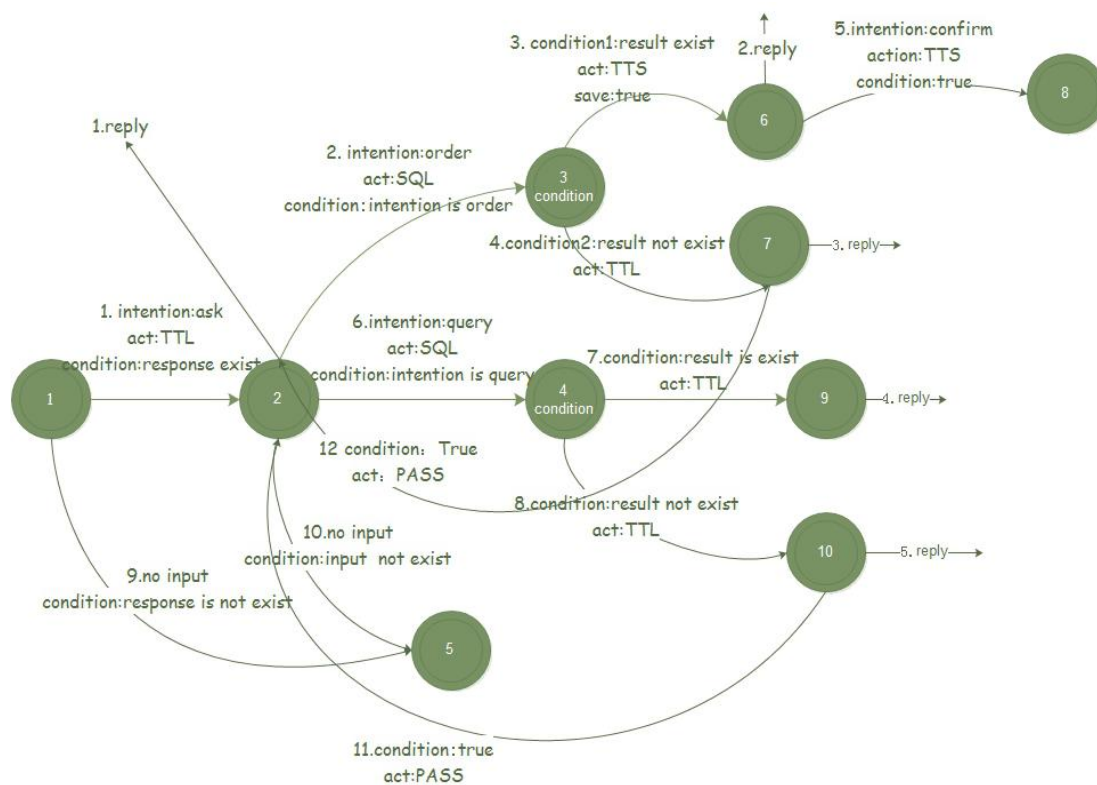


图 3 State Translation

3.4. 自然语言生成模块

根据状态转移图中分析出的应答模板进行填槽，生成应答语句。

eg. Reply Template 好的，已经下单<dish_name>

=> 好的，已经下单西红柿炒鸡蛋

4. 数据库设计

4.1. 关键词表（keyword）

4.1.1. 意义

用于替换语句中的关键词，如菜名、配菜名、口味等。

4.1.2. 属性说明

列名	属性	说明
name	string	关键词内容
type	string	关键词类型

4.1.3. 表结构

```
{
    "name": "炒鸡蛋",
    "type": "dish_name"
}
```

4.2. 菜单表（food）

4.2.1. 作用

用于记录菜品相关信息。

4.2.2. 属性说明

列名	属性	说明
dish_name	string	菜名
style	List<string>	菜品特性：荤菜、海鲜
method	string	做法
taste	List<string>	菜品口味：咸鲜，蚝香

material	List<string>	菜品所用材料
price	int	价格
chef	string	厨师
unit	string	单位：份 瓶 碗
flavour	List<string>	调料
heat	string	火候
type	string	菜品类型：菜、主食、汤

4.2.3. 表结构

```

{
    "style": [
        "荤菜",
        "海鲜"
    ],
    "method": "爆",
    "taste": [
        "蚝香",
        "酱香",
        "咸鲜",
        "辣"
    ],
    "material": [
        "内酯豆腐",
        "大海米",
        "虾仁",
        "乌鱼花",
        "鲜冻扇贝丁",
        "香菇",
        "蘑菇",
        "海参",
        "肉片",
        "青红椒",
        "蒜末",
        "蒜片"
    ],
    "price": "45",
    "chef": "温安时",
    "unit": "份",
    "flavour": [
        "蚝油",
        "生抽",

```

```

        "料酒",
        "老抽",
        "胡椒粉"
    ],
    "heat": "大火",
    "dish_name": "八珍豆腐",
    "type": "菜"
}

```

4.3. 状态图—弧表 (graph_arc)

4.3.1. 作用

存储状态转移图中的 arc

4.3.2. 属性说明

列名	属性	说明
id	int	编号
intention	string	意图内容: 点餐、菜品询问
input_state	List<string>	输入状态编号
output_state	List<string>	输出状态编号
option	List<map>	操作
act	string	操作类型: SQL 查询, TTS 输出
save	string	是否需要保存: true, false
reply	int	回答 id

4.3.3. 表结构

```

{
    "intention": "order",
    "input_state": [
        2.0
    ],
    "output_state": [
        3.0
    ],
    "option": {
        "act": "SQL",

```

```

        "save": "False"
    },
    "id": 2.0
}

```

4.4. 状态图—状态表

4.4.1. 作用

存储状态转移图中的状态结构。

4.4.2. 属性说明

列名	属性	说明
id	int	编号
reply	int	回答 id
input	List<int>	输入状态编号
output	List<int>	输出状态编号
conditions	List<map>	条件

4.4.3. 表结构

```

{
    "id": 1.0,
    "input": [],
    "output": [
        2.0,
        5.0
    ],
    "conditions": [
        {
            "response == None": 5.0
        },
        {
            "response == not None": 2.0
        }
    ],
    "reply": ""
}

```

```
}

```

4.5. 回答表 (reply)

4.5.1. 作用

存储应答模板。

4.5.2. 属性说明

列名	属性	说明
id	int	编号
type	string	应答类型： list 列表类型； signal 单一会话； simple 简单回答
content	string	回答内容模板

4.5.3. 表结构

```
{
  "id": 1.0,
  "type": "simple",
  "content": "小店欢迎您,您想点点儿什么呢"
}
```

4.6. idf 向量表 (idf_vector)

4.6.1. 作用

用于存储 idf 数据。

4.6.2. 属性说明

列名	属性	说明
keyword	string	idf 存储词
value	string	idf 值

4.6.3. 表结构

```
{
  "keyword": "请问",
  "value": "6.12686918411418"
}
```

4.7. 句式表 (`sentence_vector`)

4.7.1. 作用

存储训练过程中的句式向量。

4.7.2. 属性说明

列名	属性	说明
<code>intention</code>	<code>string</code>	意图
<code>sentence_vector</code>	<code>string</code>	句式向量

4.7.3. 表结构

```
{
  "intention": "order",
  "sentence_vector": "0.0000000000000000e+00
0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00
0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00
0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00
0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00
0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00
0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00
0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00
0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00
0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00
0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00
0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00
0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00
0.0000000000000000e+00 0.0000000000000000e+00 0.0000000000000000e+00
```

```

0.000000000000000000e+00 0.000000000000000000e+00 0.000000000000000000e+00
0.000000000000000000e+00 0.000000000000000000e+00 0.000000000000000000e+00
0.000000000000000000e+00 0.000000000000000000e+00 0.000000000000000000e+00
0.000000000000000000e+00 0.000000000000000000e+00 0.000000000000000000e+00
0.000000000000000000e+00 0.000000000000000000e+00 0.000000000000000000e+00
0.000000000000000000e+00 0.000000000000000000e+00 0.000000000000000000e+00
0.000000000000000000e+00 0.000000000000000000e+00 0.000000000000000000e+00
0.000000000000000000e+00 0.000000000000000000e+00 0.000000000000000000e+00
0.000000000000000000e+00 9.793747073007318393e-01 0.000000000000000000e+00
0.000000000000000000e+00 0.000000000000000000e+00 0.000000000000000000e+00
0.000000000000000000e+00 0.000000000000000000e+00 0.000000000000000000e+00
0.000000000000000000e+00 0.000000000000000000e+00 0.000000000000000000e+00
0.000000000000000000e+00 0.000000000000000000e+00 0.000000000000000000e+00
0.000000000000000000e+00 0.000000000000000000e+00 0.000000000000000000e+00
0.000000000000000000e+00"
    }

```

5. 功能拓展

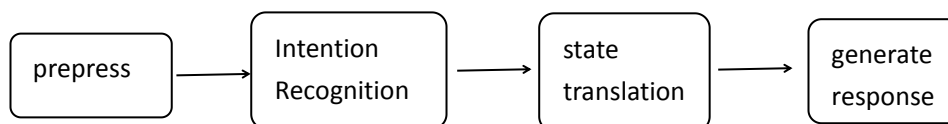
5.1 句式拓展

句式拓展可以通过 `tf-idf` 算法计算出新的 `sentence_vector` 并添加到句式表中。

5.2 节点拓展

节点拓展可通过 `megrez_tool` 中对图形进行节点的添加，从而添加数据库节点和弧的内容。

6. 测试



6.1 单元测试

对于一个句子在内部的 `reply` 生成的结果是否正确进行测试

6.2 整体测试

模仿用户请求对整个架构进行测试