# 第7次3：语义理解

## 实验目的：

1. 熟悉robot\_audio功能包中的语音服务；
2. 学习如何启动aiui语义理解服务；
3. 编写客户端程序让机器人对语音做出回答。

## 实验原理：

1. 需要打开的节点：

voice\_aiui\_node：语音服务节点，该节点包括“/voice\_iat”、“/voice\_aiui”、“/voice\_tts”、/voice\_up\_sync等服务。

1. 调用的服务：

“voice\_aiui”服务

功能：

语义理解，并分析语义。可区分语音技能，分别为**系统技能（system）**和**用户自定义技能（user）**。**系统自定义技能**可以回答一些日常对话。**用户自定义技能**可以识别语音意图，包括**导航意图（robot\_nav）**和**控制意图（robot\_control）**。**slots\_name**参数可以提取**用户自定义技能**中附加的**导航意图**和**控制意图**中的。

类型：

robot\_audio/robot\_semanteme.srv

int32 mode

string textorpath

---

string tech

string iat

string anwser

string intent

string[] slots\_name

string[] slots\_value

其中：

mode：1为文字输入，2为音频输入；

textorpath：mode为1时，此处为要语义识别的文字，mode为2时，此处为音频所在的路径；

---

tech：技能，system为系统技能，user为用户自定义技能；

iat：听写内容；

anwser：回答；

intent：自定义意图，会响应robot\_nav、robot\_control参数；

slots\_name：自定义意图附加的语义实体；

slots\_valur：slots\_name对应的具体值。

图示

描述已自动生成

## 实验步骤：

1. 创建aiui.cpp文件，终端输入：

$ cd ~/ros\_workspace/src/bobac3\_audio/src

$ touch semanteme.cpp

1. semanteme.cpp代码如下：

#include <ros/ros.h>

#include <robot\_audio/robot\_semanteme.h>

#include <string>

int main(int argc, char\*\* argv)

{

ros::init(argc, argv, "semanteme");

ros::NodeHandle n;

//定义“audio\_semanteme”客户端

ros::ServiceClient client = n.serviceClient<robot\_audio::robot\_semanteme>("voice\_aiui");

//等待服务开启

ros::service::waitForService("voice\_aiui");

//创建服务实例

robot\_audio::robot\_semanteme srv;

srv.request.mode = 1;

// srv.request.textorpath = "今天天气怎么样";

srv.request.textorpath = "带我去卧室";

// srv.request.textorpath = "前进0.2米";

std::cout<<"问题："<< srv.request.textorpath<<std::endl;

client.call(srv);

//系统技能

if(srv.response.tech == "system"){

std::cout<<"回答："<< srv.response.anwser<<std::endl;

//用户自定义技能

}else if(srv.response.tech == "user"){

//导航意图

if(srv.response.intent == "robot\_nav"){

//打印消息

std::cout<<srv.response.slots\_name[0]<<":"<<srv.response.slots\_value[0]

<<"\n"<<srv.response.slots\_name[1]<<":"<<srv.response.slots\_value[1]<<std::endl;

//控制意图

}else if(srv.response.intent == "robot\_control"){

//打印消息

std::cout<<srv.response.slots\_name[0]<<":"<<srv.response.slots\_value[0]

<<"\n"<<srv.response.slots\_name[1]<<":"<<srv.response.slots\_value[1]

<<"\n"<<srv.response.slots\_name[2]<<":"<<srv.response.slots\_value[2]<<std::endl;

}

std::cout<<"回答："<< srv.response.anwser<<std::endl;

}

return 0;

}

修改~/ros\_workspace/src/bobac3\_audio/CMakeLists.txt文件，添加如下：

图片包含 文本

描述已自动生成

1. 编译功能包，终端输入：

$ cd ~/ros\_workspace

$ catkin\_make

1. 创建voice\_semanteme.launch启动文件，终端输入:

$ cd ~/ros\_workspace/src/bobac3\_audio/launch

$ touch voice\_semanteme.launch

1. voice\_semanteme.launch为本实验的启动文件，：

<launch>

<!-- 实验节点 -->

<node pkg="bobac3\_audio" type="semanteme\_node" name="semanteme" output="screen"/>

<!-- 开启语音服务 -->

<node pkg="robot\_audio" type="voice\_aiui\_node" name="voice\_aiui\_node"/>

<node name="voice\_collect" pkg="robot\_audio" type="voice\_collect\_node" output="screen">

<!-- 音频文件目录 -->

<param name="audio\_file" type="string" value="./AIUI/audio/audio.wav"/>

</node>

</launch>

1. 打开终端，运行代码：

$ roslaunch bobac3\_audio voice\_semanteme.launch

1. 运行终端，修改程序当问题为：“今天天气怎么样”时：

文本

描述已自动生成

1. 运行终端，问题为：“带我去卧室”时：

文本

描述已自动生成

1. 运行终端，问题为：“前进0.2米”时：

文本

描述已自动生成