



**人工智能实验报告**

**课程名称\_\_\_\_\_人工智能实验 \_**

**学生学院\_\_ 计算机学院\_\_\_\_\_\_**

**专业班级\_\_\_ 16级计科3班**

**学 号\_\_\_\_ 3116004560 \_\_\_**

**学生姓名\_\_\_\_ 刘付彩文 \_ \_**

**2018 年 11 月 17 日**

**动物识别系统**

1. **实验目的**

动物识别系统，利用产生式的规则方法，根据动物的相关特征来识别海燕、老虎、金钱豹、长颈鹿、斑马、鸵鸟、企鹅等几种动物。

1. **实验环境**
   1. **硬件环境：微型计算机**
   2. **软件环境：Windows操作系统，Java语言，eclipse开发工具**
2. **实验内容与分析**

产生式规则：

（1）若某动物有奶，则它是哺乳动物。

（2）若某动物有毛发，则它是哺乳动物。

（3）若某动物有羽毛，则它是鸟。

（4）若某动物会飞且生蛋，则它是鸟。

1. 若某动物是哺乳动物且有爪且有犬齿且目盯前方，则它是食肉动物。

（6）若某动物是哺乳动物且吃肉，则它是食肉动物。

（7）若某动物是哺乳动物且有蹄，则它是有蹄动物。

（8）若某动物是哺乳动物且反刍食物，则它是有蹄动物。

（9）若某动物是食肉动物且黄褐色且有黑色条纹，则它是老虎。

（10）若某动物是食肉动物且黄褐色且有黑色斑点，则它是金钱豹。

（11）若某动物是有蹄动物且长腿且长脖子且黄褐色且有暗斑点，则它是长颈鹿。

（12）若某动物是有蹄动物且白色且有黑色条纹，则它是斑马。

（13）若某动物是鸟且不会飞且长腿且长脖子且黑白色，则它是驼鸟。

（14）若某动物是鸟且不会飞且会游泳且黑白色，则它是企鹅。

（15）若某动物是鸟且善飞，则它是海燕。

设计思路：

用户界面：采用问答形式；

知识库（规则库）：存放产生式规则，推理时用到的一般知识和领域知识，比如动物的特征，动物的分类标准，从哺乳动物、食肉动物来分，再具体地添加一些附加特征得到具体动物；建立知识库的同时也建立了事实库。事实库是一个动态链表，一个事实是链表的一个结点。知识库通过事实号与事实库发生联系。

数据库：用来存放用户回答的问题，存放初始状态，中间推理结果，最终结果；

推理机：采用正向推理，推理机是动物识别的逻辑控制器，它控制、协调系统的推理，并利用知识库中的规则对综合数据库中的数据进行逻辑操作。推理机担负两项基本任务：一是检查已有的事实和规则，并在可能的情况下增加新的事实；二是决定推理的方式和推理顺序。将推理机制同规则对象封装在一起，事实对象记录了当前的状态，规则对象首先拿出前提条件的断言（只有这些前提都有符合时才会做这条规则的结论），询问事实对象集，如事实对象集不知道，则询问用户，如所有前提条件都被证实为真则结论为真，否则系统不知道结论真假。

程序流程图：

以老虎，金钱豹，长颈鹿为例画出程序流程图如下：

哺乳动物

有毛发

有奶

吃肉

有爪

有犬齿

目盯前方

有蹄

长腿

长脖子

有暗斑点

黄褐色

有黑色条纹

食肉动物

有黑色斑点

有蹄动物

老虎

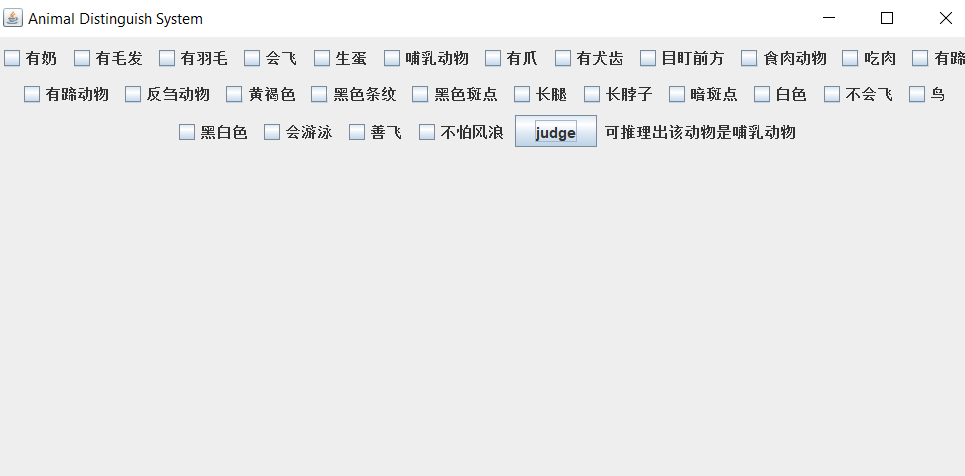
金钱豹

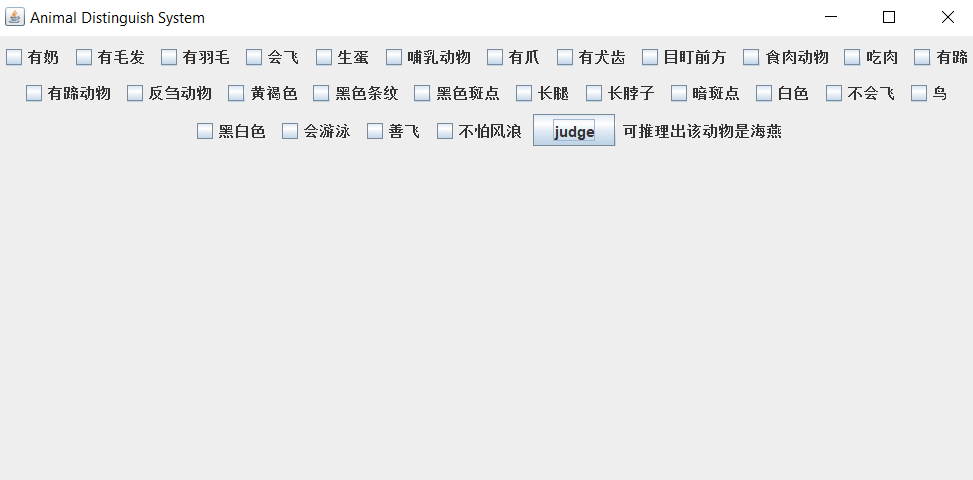
长颈鹿

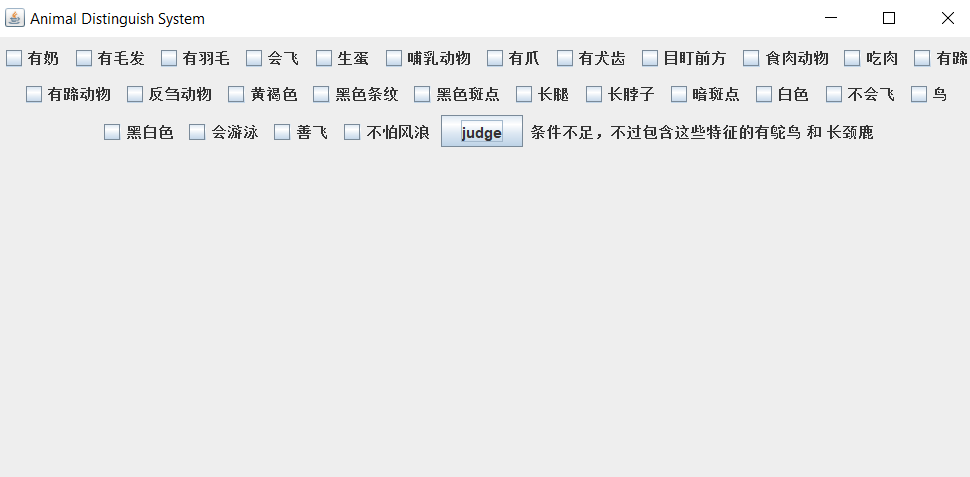
# 采用的技术方法

设计三个txt文件分别存放动物特征、规则库以及目标动物。根据输入的动物特征，结合规则库里的规则，进行推理产生新的规则或者结果。

1. **设计实现**
   1. **io：这个包为了读取三个文件的内容，将规则与知识库的内容取出**
   2. **model：这个包为实体类，对应着取出的数据**
   3. **AnimalBoard：建立系统界面**
   4. **GOAnimal：启动动物识别**
   5. **StartAnimal：开启程序**
   6. **DistinguishSystem：产生式系统算范**
2. **运行与测试**







1. **心得体会**
   1. **完成该实验过程中，按一定策略从规则库中选择规则与数据库的已知事实进行匹配，从而推理出相应的结果，于是更加加深我对产生式系统结构的理解，利用这种产生式系统结构求解问题的过程和人类求解问题时的思维很相似，因而可以用来模拟人类求解问题的思维过程。**
2. **源代码**

package wen.animal;

import java.awt.EventQueue;

public class StartAnimal {

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

@Override

public void run() {

GOAnimal go = new GOAnimal();

go.GO();

}

});

}

}

package wen.animal;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

import javax.swing.JButton;

import javax.swing.JCheckBox;

import javax.swing.JLabel;

import javax.swing.JPanel;

public class GOAnimal {

public AnimalBoard animal;

public JCheckBox checkBox[];

public ArrayList<String> numList;

public DistinguishSystem ds;

public GOAnimal() {

ds = new DistinguishSystem();

}

public void GO() {

animal = new AnimalBoard();

initPanel();

}

public void initPanel() {

JPanel northPanel = new JPanel();

JPanel southPanel = new JPanel();

JButton checkButton = new JButton("judge");

JLabel resultLabel = new JLabel("result");

checkBox = new JCheckBox[27];

checkBox[0] = new JCheckBox("有奶");

checkBox[1] = new JCheckBox("有毛发");

checkBox[2] = new JCheckBox("有羽毛");

checkBox[3] = new JCheckBox("会飞");

checkBox[4] = new JCheckBox("生蛋");

checkBox[5] = new JCheckBox("哺乳动物");

checkBox[6] = new JCheckBox("有爪");

checkBox[7] = new JCheckBox("有犬齿");

checkBox[8] = new JCheckBox("目盯前方");

checkBox[9] = new JCheckBox("食肉动物");

checkBox[10] = new JCheckBox("吃肉");

checkBox[11] = new JCheckBox("有蹄");

checkBox[12] = new JCheckBox("有蹄动物");

checkBox[13] = new JCheckBox("反刍动物");

checkBox[14] = new JCheckBox("黄褐色");

checkBox[15] = new JCheckBox("黑色条纹");

checkBox[16] = new JCheckBox("黑色斑点");

checkBox[17] = new JCheckBox("长腿");

checkBox[18] = new JCheckBox("长脖子");

checkBox[19] = new JCheckBox("暗斑点");

checkBox[20] = new JCheckBox("白色");

checkBox[21] = new JCheckBox("不会飞");

checkBox[22] = new JCheckBox("鸟");

checkBox[23] = new JCheckBox("黑白色");

checkBox[24] = new JCheckBox("会游泳");

checkBox[25] = new JCheckBox("善飞");

checkBox[26] = new JCheckBox("不怕风浪");

for(int i = 0; i < 27; i++) {

northPanel.add(checkBox[i]);

}

northPanel.add(checkButton);

northPanel.add(resultLabel);

checkButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

numList = new ArrayList<>();

for(int i = 0; i < 27; i++) {

if(checkBox[i].isSelected()) {

numList.add(String.valueOf(i));

}

}

for(int j = 0; j < 27; j++) {

checkBox[j].setSelected(false);

}

String info;

if(numList.size() == 0) {

info = "请选择特征";

}else {

info = ds.distinguish(numList);

}

resultLabel.setText(info);

System.out.println(info);

}

});

animal.setContentPane(northPanel);

animal.setVisible(true);

}

}

package wen.animal;

import javax.swing.JCheckBox;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class AnimalBoard extends JFrame {

public AnimalBoard() {

setSize(800,400);

setTitle("Animal Distinguish System");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setVisible(true);

}

}

package wen.animal;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.Stack;

import wen.animal.io.SourceDataReader;

import wen.animal.model.Data;

import wen.animal.model.Goal;

import wen.animal.model.Rules;

public class DistinguishSystem {

private ArrayList<Rules> rules = new ArrayList<>();

private ArrayList<Data> data = new ArrayList<>();

private ArrayList<Goal> goal = new ArrayList<>();

public String distinguish(ArrayList<String> numList){

String info = "";

StringBuilder sb = new StringBuilder();

for(String e : numList) {

System.out.print(e + " ");

}

System.out.println();

SourceDataReader sdr = new SourceDataReader();

rules = sdr.getRules();

data = sdr.getData();

goal = sdr.getGoal();

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

Stack<String> result = new Stack<>();

int maxWeight = 0;

for(Rules r : rules) {

int currentWeight = r.getWeigth(numList);

if (0 == currentWeight) {

continue;

}

weigth.put(String.valueOf(r.getId()), currentWeight);

if (currentWeight > maxWeight) {

result.clear();

result.push(r.getId());

maxWeight = currentWeight;

} else if (currentWeight == maxWeight) {

result.push(r.getId());

}

}

int hasFeatureCount = 0;

sb.append("条件不足，不过包含这些特征的有");

int index = 0;

while (index < result.size()) {

if(getRepeat(result, result.get(index)) > 1) {

result.remove(index);

}

index++;

}

while (!result.isEmpty()) {

String cur = result.pop();

for(Rules e : rules) {

if(e.getId().equals(cur)) {

if(weigth.get(e.getId()) == numList.size()) {

info = "可推理出该动物是" + getAnimal(e.getResult());

break;

} else {

if (hasFeatureCount > 0) {

sb.append(" 和 ");

}

++hasFeatureCount;

sb.append(getAnimal(e.getResult()));

}

}

}

}

if (hasFeatureCount > 0) {

info = sb.toString();

}

return info;

}

public String getAnimal(String id) {

for(Data d : data) {

if(d.getId().equals(id)) {

return d.getPoint();

}

}

for(Goal g : goal) {

if(g.getId().equals(id)) {

return g.getName();

}

}

return "";

}

public int getRepeat(Stack<String> result, String id) {

int num = 0;

for(String e : result) {

if(rules.get(Integer.parseInt(id)).getResult().equals(rules.get(Integer.parseInt(e)).getResult())) {

num++;

}

}

return num;

}

}

package wen.animal.io;

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

import java.util.ArrayList;

import wen.animal.model.Data;

import wen.animal.model.Goal;

import wen.animal.model.Rules;

import wen.animal.tools.Directory;

public class SourceDataReader {

private static final String PATH\_RULES = "/lib/rules.txt";

private static final String PATH\_DATA = "/lib/data.txt";

private static final String PATH\_GOAL = "/lib/goal.txt";

private ArrayList<Rules> rules = new ArrayList<>();

private ArrayList<Data> data = new ArrayList<>();

private ArrayList<Goal> goal = new ArrayList<>();

public ArrayList<Rules> getRules(){

try {

BufferedReader in = new BufferedReader(

new FileReader(Directory.getPath(PATH\_RULES))

);

String s;

// StringBuilder sb = new StringBuilder();

int index = 0;

while((s = in.readLine()) != null) {

rules.add(rulesSplit(s, String.valueOf(index)));

index++;

}

in.close();

System.out.println(rules.toString());

}catch(IOException e) {

e.printStackTrace();

}

return rules;

}

public ArrayList<Data> getData() {

try {

BufferedReader in = new BufferedReader(

new FileReader(Directory.getPath(PATH\_DATA))

);

String s;

while((s = in.readLine()) != null) {

data.add(dataSplit(s));

}

in.close();

System.out.println(data.toString());

}catch (IOException e) {

e.printStackTrace();

}

return data;

}

public ArrayList<Goal> getGoal() {

try {

BufferedReader in = new BufferedReader(

new FileReader(Directory.getPath(PATH\_GOAL))

);

String s;

while((s = in.readLine()) != null) {

goal.add(goalSplit(s));

}

in.close();

System.out.println(goal.toString());

}catch(IOException e) {

e.printStackTrace();

}

return goal;

}

public Data dataSplit(String s1) {

Data d = new Data();

String[] s2 = s1.split("\\.");

d.setId(s2[0]);

d.setPoint(s2[1]);

return d;

}

public Goal goalSplit(String s1) {

Goal g = new Goal();

String[] s2 = s1.split("\\.");

g.setId(s2[0]);

g.setName(s2[1]);

return g;

}

public Rules rulesSplit(String s1, String index) {

Rules rules = new Rules();

ArrayList<String> rules1 = new ArrayList<>();

String[] s2 = s1.split("->");

String[] s3 = s2[0].split("\\\*");

for(int i = 0; i < s3.length; i++) {

rules1.add(s3[i]);

}

rules.setResult(s2[1]);

rules.setRulesList(rules1);

rules.setId(index);

return rules;

}

}

package wen.animal.model;

public class Data {

private String id;

private String point;

public void setId(String id) {

this.id = id;

}

public String getId() {

return id;

}

public void setPoint(String point) {

this.point = point;

}

public String getPoint() {

return point;

}

public String toString() {

return id + "+" + point + " ";

}

}

package wen.animal.model;

public class Goal {

private String id;

private String name;

public void setId(String id) {

this.id = id;

}

public String getId() {

return id;

}

public void setName(String name) {

this.name = name;

}

public String getName() {

return name;

}

public String toString() {

return id + "+" + name + " ";

}

}

package wen.animal.model;

import java.util.ArrayList;

public class Rules {

private ArrayList<String> rulesList;

private String result;

private String id;

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public void setRulesList(ArrayList<String> rulesList) {

this.rulesList = rulesList;

}

public ArrayList<String> getRulesList() {

return rulesList;

}

public void setResult(String result) {

this.result = result;

}

public String getResult() {

return result;

}

public int getWeigth(ArrayList<String> numList) {

int index = 0;

for(String n : numList) {

for(String r : rulesList) {

if(n.equals(r)) {

index++;

break;

}

}

}

return index;

}

public String toString() {

StringBuilder result = new StringBuilder();

for (String s : rulesList) {

result.append(s + "+");

}

result.append(this.result);

return result.toString();

}

}

package wen.animal.tools;

import java.io.File;

import java.io.IOException;

public class Directory {

public static String getPath(String path) throws IOException {

File root = new File("");

return root.getCanonicalFile() + path;

}

}