

Artificial Intelligence

CM3331-3

Development of an intelligent system

Anne Holzapfel, hq923380

Table of Contents

Table of Contents.....	2
Introduction.....	3
Purpose and aims of the system.....	3
Purpose.....	3
Aim.....	3
Domain knowledge.....	3
Target user.....	3
Structure.....	3
Output.....	4
Knowledge representation.....	4
Reasons.....	4
Advantages.....	6
Limitations.....	6
Implementation.....	6
User Interface.....	11
Test runs.....	12
Conclusion.....	15

Introduction

The assignment of the lecture Artificial Intelligence was handed out on the 9th of October and should be handed in until the 28th of November 2003.

The task was to select a suitable domain for developing an intelligent system. The domain of this system described in the following report deals with the classification of celebrities after their profession.

Purpose and aims of the system

Purpose

The purpose of the system is that it is able to decide, to which category a celebrity belongs when giving some information about that person.

The user is asked to answer a number of questions concerning the person he has in mind and the system tries to find out, to which category the person belongs. If the description doesn't fit to a known celebrity, the program should find similar persons in the same or the superior category.

The user also should have the possibility to add a person to the system, if the person isn't already classified.

Aim

The aim of the system is to give the user the option to classify a celebrity with its knowledge about that person. Furthermore it provides the possibility to find out which celebrity has been described by the user if that person is registered in the system.

There are many possibilities to classify celebrities and this system mirrors one way to classify them and should be able to do that for the user.

Domain knowledge

The system only stores information about birth date, date of death, gender and if the person is famous for its physical or psychical abilities. It is not always easy to categorize celebrities after that less information but a way of doing that is provided by the system. The domain does not include all categories only a cut-out. Providing all categories would exceed the limit of this assignment.

Target user

The user to whom the system is targeted should be interested in celebrities and different ways of classifying them. This system provides to this intended user the ability to do that in one special way and after given criteria without expecting detailed knowledge from the user. He/she only should know the basic facts of a celebrity like its birth date.

Structure

This classification of celebrities is- as mentioned before -only a cut-out of a classification hierarchy. The system structures the celebrities after their profession and additionally stores information about dates of birth, death and gender. The user is first asked to give information on the gender of the person. Then he is asked, if the person is still alive and afterwards he has to decide whether the person is famous for its physical (for example a sportsman) or psychical work (for example a scientist). After collecting that information the system asked for the work of the person specifically.

Output

The system put out the name of the celebrity which has been described if this is classified or the category, to which the celebrity belongs. If there is no match to the given criteria the system will give back a list of celebrities with nearly the same criteria.

If the celebrity is not yet classified the user has the possibility to add the person by answering some questions about that celebrity.

Knowledge representation

It was decided to use semantic networks to represent the system.

Semantic networks consist of entities and relations between these entities. They can be represented as a graph (see picture 1). The nodes of the graph correspond to entities and the links between nodes to the relationship between entities labeled by the name of the relations.

“Is a” is the name of relations between objects. It denotes a relation between a class and its superclass. In semantic nets knowledge is organized using explicit links between objects.

Reasons

The reasons for choosing semantic networks as knowledge representation lie in the structure.

The celebrity system is a hierarchical model.

Semantic nets allow defining relations between object including class relations (X isa Y).

Inference is supported by inheritance.

Semantic nets are easy to read and to understand another reason for using this knowledge representation.

To ensure consistency in the knowledge base all members of a class inherit the properties of that class.

The reasoning is efficient for that kind of system including mainly factual knowledge.

Inference is done in an acceptable time.

Advantages

Semantic nets can capture inheritance information in a modular way. Their simplicity makes them easy to understand and implement.

They are efficient because information does not have to be entered sentence-by-sentence, but simply by drawing links to connect nodes. Inference is done simply by following links.

Limitations

Semantic nets have some limitations given by the structure. They do not distinguish between the class of an object and a particular object. Furthermore there is the risk that the presentation and structure for complex systems could become unmanageable when adding more and more facts to the network.

Additional it is very difficult to handle exceptions because every class inherits the attributes of the superclass.

Implementation

The “isa” relation is used to link a class and its superclasses:

```
isa(artist,celebrity).
isa(leader,celebrity).
isa(scientist,celebrity).
isa(sportsman,celebrity).
isa(musician,artist).
isa(painter,artist).
isa(filmstar,artist).
isa(writer,artist).
isa(politican,leader).
isa(king,leader).
isa(dictator,leader).
isa(physicist,scientist).
isa(chemist,scientist).
isa(biologist,scientist).
isa(tennisplayer,sportsman).
isa(footballer,sportsman).
isa(figureskater,sportsman).
isa(singer,musician).
isa(componist,musician).
isa(violinplayer,musician).
isa(poet,writer).
isa(novelwriter,writer).
isa(impressionist,painter).
isa(renaissancepainter,painter).
isa(surrealist,painter).
```

And the “isa” relation is also used to link an instance with its class.

```
isa(luciano_pavarotti,singer).
isa(vanessa_mae,violinplayer).
isa(wolfgang_amadeus_mozart,componist).
isa(vincent_van_gogh,impressionist).
isa(leonardo_da_vinci,renaissancepainter).
isa(salvatore_dalí,surrealist).
isa(julia_roberts,filmstar).
isa(john_f_kennedy,politican).
isa(louisXIV,king).
isa(stalin,dictator).
isa(albert_einstein,physicist).
isa(charles_darwin,biologist).
```

```
isa(marie_curie,chemist).
isa(pete_sampras,tennisplayer).
isa(pelé,footballer).
isa(katharina_witt,figureskater).
isa(william_shakespeare,poet).
isa(arthur_conan_doyle,novelwriter).
```

Properties of classes and instances are defined using the “hasproperty” fact.

```
hasproperty(leader,work,brain).
hasproperty(scientist,work,brain).
hasproperty(sportsman,work,body).
hasproperty(singer,work,body).
hasproperty(painter,work,body).
hasproperty(filmstar,work,body).
hasproperty(writer,work,brain).
hasproperty(componist,work,brain).
hasproperty(violinplayer,work,body).
hasproperty(julia_roberts,is,female).
hasproperty(julia_roberts,born,'28.10.1967').
hasproperty(julia_roberts,die,na).
hasproperty(luciano_pavarotti,is,male).
hasproperty(luciano_pavarotti,born,'12.10.1935').
hasproperty(luciano_pavarotti,die,na).
hasproperty(wolfgang_amadeus_mozart,is,male).
hasproperty(wolfgang_amadeus_mozart,born,'27.01.1756').
hasproperty(wolfgang_amadeus_mozart,die,'05.12.1791').
hasproperty(vanessa_mae,is,female).
hasproperty(vanessa_mae,born,'27.10.1978').
hasproperty(vanessa_mae,die,na).
hasproperty(arthur_conan_doyle,is,male).
hasproperty(arthur_conan_doyle,born,'22.05.1859').
hasproperty(arthur_conan_doyle,die,'07.07.1930').
hasproperty(william_shakespeare,is,male).
hasproperty(william_shakespeare,born,'26.04.1564').
hasproperty(william_shakespeare,die,'23.04.1616').
hasproperty(vincent_van_gogh,is,male).
hasproperty(vincent_van_gogh,born,'30.03.1853').
hasproperty(vincent_van_gogh,die,'29.07.1890').
hasproperty(leonardo_da_vinci,is,male).
hasproperty(leonardo_da_vinci,born,'15.04.1452').
hasproperty(leonardo_da_vinci,die,'02.05.1519').
hasproperty(salvatore_dalí,is,male).
hasproperty(salvatore_dalí,born,'11.05.1904').
hasproperty(salvatore_dalí,die,'23.01.1989').
hasproperty(john_f_kennedy,reign,usa).
hasproperty(john_f_kennedy,is,male).
hasproperty(john_f_kennedy,born,'11.05.1904').
hasproperty(john_f_kennedy,die,'23.01.1989').
hasproperty(louisXIV,is,male).
hasproperty(louisXIV,born,'05.09.1638').
hasproperty(louisXIV,die,'01.09.1715').
hasproperty(stalin,is,male).
hasproperty(stalin,born,'21.12.1879').
hasproperty(stalin,die,'05.03.1953').
hasproperty(albert_einstein,is,male).
hasproperty(albert_einstein,born,'14.03.1879').
hasproperty(albert_einstein,die,'18.04.1955').
hasproperty(charles_darwin,is,male).
hasproperty(charles_darwin,born,'12.02.1809').
hasproperty(charles_darwin,die,'19.04.1882').
```

```
hasproperty(marie_curie,is,female).
hasproperty(marie_curie,born,'07.11.1867').
hasproperty(marie_curie,die,'04.07.1934').
hasproperty(pete_sampras,is,male).
hasproperty(pete_sampras,born,'12.08.1971').
hasproperty(pete_sampras,die,na).
hasproperty(pelé,is,male).
hasproperty(pelé,born,'23.10.1940').
hasproperty(pelé,die,na).
hasproperty(katharina_witt,is,female).
hasproperty(katharina_witt,born,'03.12.1965').
hasproperty(katharina_witt,die,na).
```

Properties are inherited along “isa” paths, therefore you need recursive rules.

```
inherit(Concept, Prop, Attribute):
    hasproperty(Concept,Prop,Attribute).
inherit(Concept, Prop, Attribute):-
    isa(Concept, Superclass),
    inherit(Superclass,Prop,Attribute).

is_celebrity(Class,Subclass):-
    isa(Class,Subclass).
is_celebrity(Class,Subclass):-
    isa(Class,Subsubclass),
    is_celebrity(Subsubclass,Subclass).
```

The program is divided into two parts. The first and main part is about the classification of the celebrities. The implementation of the functionality will be given directly in the source code. To make it more concise the comments will be marked in bold letters.

```

/*The run method is the method which is called at the program start.*/
run:- intro,nl, /*the intro only shows a welcome message*/
      head, /*head includes the functionality*/
      nl,end. /*end shows a leave message*/
/*there are two possibilities to use the system, the first is to*/ /
/*identify the person the user has in mind and the second is to add a*/ /
/*person to the classification*/
head:- write('Please select one and press return'),nl,nl,
      tab(4),write('(1) Let the system identify one person and its
      category'),nl,
      tab(4),write('(2) Let the system categorize a new person and
      add it to the system'),nl,nl,
      write('Don't forget to put the "." behind your input!'),nl,
      read(Choice),nl,
      mainpart(Choice)./*input is read in and decided which method*/
/*should be called to go one*/

/*possibility to list all celebrities categorized*/
mainpart(1):- write('Do you first want to have a look, which persons
      are in the system? y/n'),nl,
      read(Choice2),nl,
      choice2(Choice2)./*check input*/
mainpart(2):- two.
mainpart(y):- two.
mainpart(n):- nl.
      /*list all celebrities and proceed as afterwards*/
      choice2(y):findall(Person,hasproperty(Person,born,_),L1),
      write(L1),nl,one.
      /*proceed without listing*/
      choice2(n):-one.
/*start of identification of the celebrity*/
/*collect general information*/

one:- nl, write('Is the person you think up male or female?'),nl,
      read(Gender),nl,
      write('Is the person still alive? y/n '),nl,
      read(Alive),nl,
      write('Is the person famous for its'),nl,
      tab(4),write('(1)wit'),nl,
      tab(4),write('(2)manual labour'),nl,
      read(Work),nl,
      write('What attribute fits best to the persons
      profession?'),nl,
      tab(4),write('(1)political'),nl,
      tab(4),write('(2)artistical'),nl,
      tab(4),write('(3)sporty'),nl,
      tab(4),write('(4)scientific'),nl,
      read(Profession),nl,
      profession(Profession,Work,Gender,Alive).

is_alive(Person,Gender,y):- hasproperty(Person,is,Gender),
      hasproperty(Person,die,na).
is_alive(Person,Gender,n):- hasproperty(Person,is,Gender),
      hasproperty(Person,die,Date),Date\=na.
/*find persons fitting the given attributes*/
is_some(Person,Gender,Alive,Part,Prof):- is_alive(Person,Gender,Alive).
      inherit(Person,work,Part),
      is_celebrity(Person,Prof).
/*find persons with similar attributes*/
is_some_related(Person,Part,Prof):- inherit(Person,work,Part),
      is_celebrity(Person,Prof),
      hasproperty(Person,born,_).
/*translate input of user to categories of the system*/

```

The second part is about adding a celebrity to the system.

```
/*the user should answer several questions*/
two:-    nl, write('What is the name of the person?'),nl,
        read(Name),nl,
        two_zero(Name).

/*IF celebrety is allready categorited THEN ask to proceed with*/
/*other ELSE proceed asking questions about celebrity*/
two_zero(Name):-is_celebrety(Name,celebrety),
                two_one(Name);
                two_two(Name).

two_one(Name):- write(Name),
                write(' is allready categorized'),nl,
                write('Do you want to proceed? y/n'),nl,
                read(Proceed),
                proceed(Proceed),nl,nl.
                proceed(y):-head.
                proceed(n):-nl.

two_two(Name):- nl,write('What is the persons gender?
                    male/female'),nl,
                read(Gender),nl,
                write('What is the persons birthdate?'),nl,
                write('Put the date in quotation marks'),nl,
                read(Birth),nl,
                write('Is the person still alive? y/n'),nl,
                read(Dead),nl,
                dead(Dead,Name,Birth,Gender).

                dead(y,Name,Birth,Gender):-
                    Diedate=na,two_two_one
(Name,Birth,Diedate,Gender).
                dead(n,Name,Birth,Gender):-
                    write('So please give me the date of death'),nl,
                    read(Diedate),nl,
                    two_two_one(Name,Birth,Diedate,Gender).
/*ask for the category*/
two_two_one(Name,Birth,Diedate,Gender):-
                write('And now choose the category'),nl,
                bagof(Category,isa(Category,celebrety),List),
                write(List),nl,
                read(Cat1),
                help(Name,Birth,Diedate,Cat1,Gender).
/*IF subcategories exist THEN ask again ELSE*/
/*call method for assert statements*/
help(Name,Birth,Diedate,Cat1,Gender):-
                isa(Cat2,Cat1),hasproperty(Cat2,born,_),
                two_two_one_one(Name,Birth,Diedate,Cat1,Gender);
                two_two_two(Name,Birth,Diedate,Cat1,Gender).

/*assert(clause)*/
two_two_one_one(Name,Birth,Diedate,Category,Gender):-
                assert(hasproperty(Name,born,Birth)),
                assert(hasproperty(Name,die,Diedate)),
                assert(hasproperty(Name,is,Gender)),
                assert(isa(Name,Category)).

/*go one category deeper*/
two_two_two(Name,Birth,Diedate,Cat,Gender):-
```

```
bagof(Category, isa(Category, Cat), List2),
write(List2), nl,
read(Cat2),
help(Name, Birth, Diedate, Cat2, Gender).
```

This second part doesn't work caused by an exception created by SWI-prolog.

```
ERROR: No permission to modify static_procedure `hasproperty/3'
^ Exception: (21) assert(hasproperty(cindy_lauper, born, '12.08.1962')) ?
```

It is planned and implemented but not working although all parameters of the "assert" predicate are correct.

User Interface

The user interface looks like follows . The user has to be careful with his input. Type mistakes and other errors aren't caught by the system. If there is something special about the input like quotation marks the user get a short info about. There are two types of input, a selection list, where the user just has to type in the number and text answers.

To run the program you only have to enter

run.

The following screen will appear:

```
*****
*                Welcome to the celebrity system                *
*****

Please select one and press return

(1) Let the system identify one person and its category
(2) Let the system categorize a new person and add it to the system

Don't forget to put the "." behind your input!
|: 1.

Do you first want to have a look, which persons are in the system? y/n
|: n.

Is the person you think up male or female?
|: male.

Is the person still alive? y/n
|: n.

Is the person famous for its
(1)wit
(2)manual labour
|: 1.

What attribute fits best to the persons profession?
(1)political
(2)artistical
(3)sporty
(4)scientific
|: 4.
```

What is the persons exact profession?
[physicist, chemist, biologist]
|: **biologist.**

The person you think of is [charles_darwin] and belongs to the category biologist

```
*****  
*                               *  
*           Thanks for using this system           *  
*                               *  
*****
```

The names of the celebrities are stored in upper letters and with an underscore between first name and surname.

Test runs

There are three main cases when running the program. The first is, that the user think of a person, describe the person and the person is in the system.

```
*****  
*                               *  
*           Welcome to the celebrity system           *  
*                               *  
*****
```

Please select one and press return

- (1) Let the system identify one person and its category
- (2) Let the system categorize a new person and add it to the system

Don't forget to put the "." behind your input!

| 1.

Do you first want to have a look, which persons are in the system? y/n

| y.

[julia_roberts, luciano_pavarotti, wolfgang_amadeus_mozart, vanessa_mae,
arthur_conan_doyle, william_shakespeare, vincent_van_gogh, leonardo_da_vinci,
salvatore_dalí, john_f_kennedy, louisXIV, stalin, albert_einstein, charles_darwin,
marie_curie, pete_sampras, pelé, katharina_witt]

Is the person you think up male or female?

| female.

Is the person still alive? y/n

| y.

Is the person famous for its

- (1)wit
- (2)manual labour

| 2.

What attribute fits best to the persons profession?

- (1)political
- (2)artistical
- (3)sporty

(4)scientific

| 2.

What is the persons exact profession?

[musician, painter, filmstar, writer]

| filmstar.

The person you think of is [julia_roberts] and belongs to the category filmstar

```
*****
*                               *
*           Thanks for using this system           *
*                               *
*****
```

The second is that the user think of a person, describes its characteristics and the person is not in the system.

```
*****
*                               *
*           Welcome to the celebrity system           *
*                               *
*****
```

Please select one and press return

- (1) Let the system identify one person and its category
- (2) Let the system categorize a new person and add it to the system

Don't forget to put the "." behind your input!

|: 1.

Do you first want to have a look, which persons are in the system? y/n

|: y.

[julia_roberts, luciano_pavarotti, wolfgang_amadeus_mozart, vanessa_mae,
arthur_conan_doyle, william_shakespeare, vincent_van_gogh, leonardo_da_vinci,
salvatore_dalí, john_f_kennedy, louisXIV, stalin, albert_einstein, charles_darwin,
marie_curie, pete_sampras, pelé, katharina_witt]

Is the person you think up male or female?

|: male.

Is the person still alive? y/n

|: n.

Is the person famous for its

- (1)wit
- (2)manual labour

|: 2.

What attribute fits best to the persons profession?

- (1)political
- (2)artistical
- (3)sporty
- (4)scientific

|: 2.

What is the persons exact profession?

[musician, painter, filmstar, writer]

|: musician.

Which part of its body needs the person most?

(1)voice

(2)hands

(3)brain

|: 1.

There is no celebrity in the system matching the given criteria.

Related celebrities are: [luciano_pavarotti]

But do you want to add your person to the system? y/n

|: n.

```
*****
*                               *
*           Thanks for using this system           *
*                               *
*****
```

And the third is that the user wants to add a celebrity.

```
*****
*           Welcome to the celebrity system           *
*                               *
*****
```

Please select one and press return

(1) Let the system identify one person and its category

(2) Let the system categorize a new person and add it to the system

Don't forget to put the "." behind your input!

| 2.

What is the name of the person?

| george_clooney.

What is the persons gender? male/female

| male.

What is the persons birthdate?

Put the date in quotation marks

| '06.05.1961'.

Is the person still alive? y/n

| y.

And now choose the category

[artist, leader, scientist, sportsman]

| artist.

[musician, painter, filmstar, writer]

| filmstar.

ERROR: No permission to modify static_procedure `hasproperty/3'

^ Exception: (20) assert(hasproperty(george_clooney, born, '06.05.1961')) ?

Conclusion

The system is limited in its knowledge caused by the complexness of the categories. Only a few are implemented and there is also only one instance for every category in the system.

But for this selection of categories the system does what it was intended to do.

It is expandable and if the assert predicate would work it would have been possible to expand the knowledge of the system at runtime.

All in all the system is running. Additionally a method could have been implemented to catch wrong input to prevent the system to stop at an earlier point before receiving a result .