



Universidade do Porto
Faculdade de Engenharia

FEUP

Informações sobre as Universidades Portuguesas em Linguagem Natural

Relatório Intercalar

Inteligência Artificial

3º ano do Mestrado Integrado em Engenharia Informática e Computação

Elementos do Grupo:

José Ferreira up200702566 josemcf89@gmail.com

29 de Maio de 2014

Objetivo

Este trabalho está a ser realizado no âmbito da cadeira de inteligência artificial lecionada no 2º semestre do mestrado Integrado em Engenharia Informática e computação da faculdade de Engenharia da Universidade do Porto.

O objetivo deste trabalho é a implementação de um programa que permita a obtenção de informações relativas a universidades Portuguesas usando linguagem natural.

Descrição

Especificação

O programa a desenvolver consiste num conjunto de módulos que em conjunto permite responder a perguntas ou indagar a veracidade de afirmações sobre as universidades portuguesas.

Foi projetado que farão parte do programa os seguintes módulos:

- **Modulo de interface** ; permite ao utilizador o input de uma frase e trata esta de forma a poder ser processada a seguir;
- **Modulo gramatical** ; consiste na definição de uma gramática que pode ser vista como uma simplificação da gramática portuguesa, este modulo é nuclear ao programa uma vez que tem o objetivo de validar sintaticamente as frases recebidas;
- **Modulo lexical**; consiste na definição de todas as palavras reconhecidas pela linguagem e sua caracterização semântica assim como propriedades de género e numero;
- **Modulo Base de conhecimento**; este modulo representa o conhecimento do programa sobre as universidades portuguesas;
- **Modulo de inferência** ; conjunto de predicados que podem ser associados a ações , permite a consulta da base de conhecimento
- **Modulo de testes** ;

Trabalho efetuado

Código produzido até á escrita deste relatório

```
code{
```

modulo de interface

```
read_routine(L):-
```

```
read_line_to_codes(user_input, Input),
```

```
atom_codes(Atom, Input),
```

```
atomic_list_concat(L, ' ', Atom),
```

```
print(L).
```

Modulo gramatical

```
s --> np(Suj,_,Q), vp(Suj,Obj,Q).
```

```
np(Suj,Obj,Q) --> det(S,Q), n(Suj,S,Obj,Q).
```

```
np(Suj,Obj,Q) --> n(Suj,S,Obj,Q).
```

```
vp(Suj,Obj,C) --> v(Suj,Obj,C), np(_,Obj,_),{write('aqui')}.
```

```
det(S,Q) --> [Word],{lexico(Word,'det',S,Q)}.
```

```
n(Word,S,Word,Q) --> [Word],{lexico(Word,'n',S,Q)}.
```

```
v(Suj,Obj,C) --> [Word],{lexico(Word,Accao,'v',C),Pred=..[Accao,Suj,Obj],Pred}.
```

Modulo de Inferencia

```
frequentar(Nome_aluno,Nome_cadeira):-aluno(Nome_aluno),cadeira(Nome_cadeira,_,frequentar').
```

```
lecionar(Nome_professor,Nome_cadeira):-
```

```
professor(Nome_professor),cadeira(Nome_cadeira,Nome_professor,'leciona').
```

Modulo Base de Conhecimento

```
aluno('aluno').
```

```
aluno('alunos').
```

```
aluno('Ze').
```

```
professor('professor').
```

```
professor('Henrique').
```

```

professor('Ana').

cadeira('Iart','Henrique','leciona').

cadeira('Iart','Ze','frequenta').

curso(X,'Informatica'):-faculdade(_,X).

curso('Engenharia','Mecanica').

curso('Engenharia','Informatica').

curso('Desporto','Alta Competicao').

faculdade(A,'Engenharia'):-universidade(A).

faculdade(A,'Psicologia'):-universidade(A).

faculdade(A,'Desporto'):-universidade(A).

universidade('Porto').

universidade('Lisboa').

universidade('Coimbra').

```

Modulo Lexical

```

%%%lexico

%%% determinantes

lexico('o','det','m','s').

lexico('a','det','f','s').

lexico('os','det','m','p').

lexico('as','det','f','p').

%%%nomes

lexico('aluno','n','m','s').

lexico('alunos','n','m','p').

lexico('Ze','n','m','s').

lexico('professor','n','m','s').

lexico('Henrique','n','m','s').

lexico('Ana','n','f','s').

lexico('Engenharia','n','f','s').

lexico('Universidade','n','f','s').

lexico('cadeira','n','f','s').

```

```
lexico('cadeiras','n','f','p').
lexico('iart','n','m','s').
lexico('café','n','m','s').
lexico('frequenta','frequentar','v','s').
lexico('frequentam','frequentar','v','p').
lexico('leciona','lecionar','v','s').
lexico('lecionam','lecionar','v','p').
}
```

Resultados esperados e forma de avaliação

É esperado que o programa cumpra os seguintes requisitos:

- Aceitação de frases afirmativas
- Aceitação de frases interrogativas
- Verificação da sintaxe da frase
- Verificação da semântica da frase
- Inclusão de um ou mais comandos. A inclusão de mais que um comando numa mesma frase é traduzida pelo uso da partícula "e" nessa frase
- Aceitação de frases iniciadas pela partícula "e", devendo ser guardado o contexto da frase anterior para resposta a esta frase
- Uso de adjetivos

Modulo de testes

%%%%testa concordância de género entre det e nome%%%%

teste1:- s(['o' , 'Henrique' , 'leciona' , 'iart'], []).

teste2:- s(['o' , 'Ana' , 'leciona' , 'iart'], []).

%%%%testa concordância semântica entre sujeito e verbo%%%%

teste3:- s(['o' , 'Ze' , 'frequenta' , 'iart'], []).

teste4:- s(['o' , 'Henrique' , 'frequenta' , 'iart'], []).

%%%%testa concordância semântica entre verbo e complemento%%%

teste5:- s(['o' , 'aluno' , 'frequenta' , 'o' , 'cafe'], []).

teste6:- s(['o' , 'Henrique' , 'leciona' , 'iart'], []).

%%%%testa concordância em numero entre determinante e nome%%%

teste7:- s(['os' , 'alunos' , 'frequenta' , 'iart'], []).

teste8:- s(['os' , 'Ze' , 'frequenta' , 'iart'], []).

%%%%testa concordância em numero entre sujeito e verbo%%%

teste9:- s(['os' , 'alunos' , 'frequenta' , 'iart'], []).

teste10:- s(['os' , 'alunos' , 'frequentam' , 'iart'], []).

Conclusões

Ao escrever este relatório e pelo trabalho realizado até hoje chego à conclusão que ainda tenho bastante trabalho pela frente e que a minha eficiência em termos de produção de código pode beneficiar de uma abordagem mais direcionada para a escrita de testes somando incrementos aos módulos já criados.

Recursos

Software:

[SWI-PROLOG](#)

[Sublime](#)

Bibliografia:

["Prolog and Natural-Language Analysis "](#)

["Natural Language Processing in Prolog "](#)

"Artificial Intelligence: A Modern Approach";

“The Art of Prolog”

