



Escola de Engenharia da Universidade do Minho
Departamento de Informática
Mestrado Integrado em Engenharia Informática
Gramáticas na Compreensão de Software

Sistema de Perguntas e Respostas

João Gomes, A74033
Tiago Fraga, A74092

14 de Janeiro de 2019

Conteúdo

1	Introdução	2
2	Contextualização	3
2.1	Etapas de Desenvolvimento	3
2.2	Escolhas de Desenvolvimento	3
2.2.1	Base de Conhecimento	3
2.2.2	Perguntas	5
2.3	Como funciona o nosso sistema	5
2.4	Resultados	6
3	Conclusões	8

Capítulo 1

Introdução

Neste trabalho prático foi nos pedidos que simulássemos um sistema de Perguntas e Respostas (*Q&A system*), isto é, um sistema que responde de forma automática a questões efetuadas por utilizadores sobre um determinado tema.

Foi pedido que esta simulação fosse realizada através da criação de uma **DSL** (*Domain-specific language*).

Como o tema era livre de escolha, optamos por criar um sistema que respondesse a perguntas sobre carros, simulando assim um stand virtual, ou seja, responde a perguntas sobre características básicas, preços, modelos, segmentos, etc.

Ao longo deste relatório vamos apresentar:

- Etapas de desenvolvimento; 2.1
- As nossas escolhas durante o seu desenvolvimento; 2.2
- Como funciona o nosso sistema; 2.3
- Resultados 2.4
- Conclusões e trabalho futuro. 3

Capítulo 2

Contextualização

Tal como foi dito anteriormente o nosso sistema, pretende simular um stand virtual respondendo a questões *tipo* sobre carros.

2.1 Etapas de Desenvolvimento

De modo a simularmos o sistema pretendido resolvemos estruturar a nossa base de conhecimento, isto é, que informação ia ser carregada inicialmente no sistema. Depois de estruturada partimos para a definição da mesma através de uma gramática desenvolvida em *ANTLR*.

Com a nossa base de conhecimento definida, partimos para a definição de uma gramática capaz de reconhecer as perguntas. Posto isto acrescentamos atributos a nossa gramática de modo a guardar a base de conhecimento em estruturas de dados, reconhecer as perguntas e gerar uma resposta-

2.2 Escolhas de Desenvolvimento

2.2.1 Base de Conhecimento

De modo a ter uma base de conhecimento abrangente e sólida definimos a seguinte estrutura: (segmento,marca,modelo,cilindrada,combustível,preço,ano,potência,características)

Dentro do **Segmento** podemos ter: Pequeno Citadino, Citadino, Utilitário, Sedan, Carrinha, Monovolume, Suv, Cabrio, Coupé e Comercial.

De modo a não termos uma base de conhecimento muito grande limitamos as marcas permitidas. As **Marcas** definidas são Renault e Dacia.

Os **Modelos** na nossa base de conhecimento dos carros Renault são: Captur, Clio, Espace, Kangoo, Laguna, Megane, Talisman, Twingo, Zoe. Os modelos dos carros Dacia são: Duster e

Sandero. Como ter uma base de conhecimento muito grande ou pequena não iria influenciar o objectivo final deste trabalho resolvemos limitar as marcas e os modelos.

As **Cilindradas** são: 0.9 TCE, 1.2 TCE, 1.5 DCI, 1.6 DCI, RS 200.

No tipo de **Combustível** temos: gasolina, gasóleo, elétrico, híbrido.

Depois temos o **Preço** que é um número definido de forma aleatória para a simulação.

No **Ano** podemos ter desde 2000 até 2018.

No tipo de **Potência** tivemos em conta os valores originais dos carros inseridos.

As **Características** são *Cor*, tipo de *Caixa*, número de *Portas* e tipo de *Tração*.

De modo a guardarmos esta informação para mais tarde darmos resposta às perguntas efetuadas, definimos duas classes *Java*. Sendo elas *Caracteristica* e *Carro*.

- **Caracteristica:**

Esta classe tem as seguintes variáveis e apresenta métodos básicos de classes *Java* tais como: *get*, *set*, *equals*, *toString* e *clone*.

```
1      class Caracteristica{
2          String cor;
3          String caixa;
4          String portas;
5          String traccao;
6      }
```

- **Carro:**

Esta classe contém a anterior.

```
1      class Carro{
2          String segmento;
3          String marca;
4          String modelo;
5          String versao;
6          int cilindrada;
7          String combustivel;
8          int preco;
9          int ano;
10         int potencia;
11         Caracteristica caracteristica;
12     }
```

2.2.2 Perguntas

Depois de definida a nossa base de conhecimento definimos como iriam ser realizadas as perguntas e optamos por definir as perguntas como um tuplo (Operador,Palavra Chave) sendo feita uma divisão entre perguntas no plural ou no singular, de modo a que o sistema responda corretamente.

Com a gramática definida, definimos os seus atributos. Estes atributos permitem guardar a informação da base de conhecimento e impõe condições relativamente as perguntas além de responder as mesmas.

2.3 Como funciona o nosso sistema

Nesta secção apresentamos como se pode interagir com o nosso sistema. A nossa gramática está definida como:

```
1      sistema: baseConhecimento '---' perguntas
2      ;
```

Por este motivo o ficheiro de *input* tem de ter a seguinte forma como por exemplo:

```
1 (pequeno citadino , renault , twingo , 0.9 TCE, 900, gasolina , 10000, 2018, 90,
   branco , manual, 4-5, dianteira);
2 (pequeno citadino , renault , twingo , 0.9 TCE, 900, gasolina , 9500, 2017, 90,
   azul , manual, 4-5, dianteira);
3 (pequeno citadino , renault , twingo , 0.9 TCE, 900, gasolina , 10000, 2018, 90,
   vermelho , manual, 4-5, dianteira);
4 (pequeno citadino , renault , twingo , 0.9 TCE, 900, gasolina , 10000, 2018, 90,
   preto , manual, 4-5, dianteira);
5 (pequeno citadino , renault , twingo , 0.9 TCE, 900, gasolina , 9000, 2017, 75,
   cinza escuro , manual, 4-5, dianteira);
6 ———
7 (Qual ,(segmento ,pequeno citadino ))?
```

Quando introduzido o ficheiro, o nosso sistema vai ler a base de conhecimento e guardar as informações em estruturas de dados, depois lê as perguntas uma a uma e fornece uma resposta com base no tipo de pergunta.

As perguntas devem ter a formulação especifica bem como usar, os operadores e as palavras chave definidas para tal criamos condições de contexto para responder em caso de erro na formulação da pergunta e na inserção na base de conhecimento.

Na base de conhecimento não permitimos qualquer tipo de veículo que não tenha aquela estrutura e os tipos apresentados anteriormente, ou seja, se tivermos um segmento, marca ou outra característica diferente das que permitimos, o sistema vai apresentar erro na inserção.

Tal como na base de conhecimento temos algumas regras nas perguntas, tais como:

- Se o operador for plural o sistema espera "(Operador,[(Palavra Chave)])?" se for singular espera (Operador,(Palavra Chave))?
- Os operadores singulares válidos são: Qual, Quanto, Que
- Os operadores plurais válidos são: Quais, Quantos
- Com base na palavra chave o sistema espera diferentes argumentos por exemplo se for preço o sistema espera receber um conjunto de opções sendo elas:

```

1      opcoes
2      : '(' '>' ' ', ' NUM ' ) '
3      | '(' '<' ' ', ' NUM ' ) '
4      | '(' '-+' ' ', ' NUM' ) '
5      | '(' '=' ' ', ' NUM ' ) '
6      ;

```

2.4 Resultados

Os resultados obtidos pelo sistema são os pretendidos, com os atributos implementados na gramática e com as condições de contexto conseguimos dar resposta as perguntas efetuadas com base na informação carregada pela Base de Conhecimento.

Exemplo de Output:

- **Carregamento da Base de Conhecimento:**

```

1      ***** Base de Dados *****
2      1->Carro:
3      Segmento: pequeno citadino
4      Marca: renault
5      Modelo: twingo
6      Versao: 0.9 TCE
7      Cilindrada: 900
8      Combust vel: gasolina
9      Pre o: 10000

```

10	Ano: 2018
11	Potência: 90
12	Característica:
13	Cor: branco
14	Caixa: manual
15	Numero de Portas: 4-5
16	Tracção: dianteira

• Pergunta e Resposta:

1	
2	*** Pergunta: ***
3	Qual o carro cujo segmento pequeno citadino ?
4	*** Resposta: ***
5	
6	Carro:
7	Segmento: pequeno citadino
8	Marca: renault
9	Modelo: twingo
10	Versão: 0.9 TCE
11	Cilindrada: 900
12	Combustível: gasolina
13	Preço: 10000
14	Ano: 2018
15	Potência: 90
16	Característica:
17	Cor: branco
18	Caixa: manual
19	Numero de Portas: 4-5
20	Tracção: dianteira

Capítulo 3

Conclusões

Concluimos que o objetivo do sistema foi cumprido na integra, ou seja o sistema reage bem as perguntas respondendo corretamente, dentro de certos limites e consoante uma estrutura especifica para input.

Como trabalho futuro poderíamos aumentar a base de Conhecimento de modo a possuir todos os carros existentes com todas as suas características. E utilizar técnicas de linguagem natural permitindo que a pergunta fosse realizada textualmente e com essas técnicas realizássemos a extração do que interessa na pergunta, para fornecer uma resposta correta.

Com estas alterações no nosso sistema e com a criação de uma interface, poderíamos construir facilmente um sistema *Q&A* completo e atrativo para os utilizadores.

Em suma, achamos que o nosso sistema permite uma grande variedade de perguntas, obtendo sempre uma resposta correta, além disso achamos que possui uma boa base de conhecimento, achamos que a estrutura está completamente funcional, permitindo um fácil melhoramento da mesma, de modo a construir um bom produto.

Bibliografia