

Conteúdo

Inteligência Artificial

Luís A. Alexandre

UBI

Ano lectivo 2018-19

Representação do conhecimento

Introdução

Lógica de Primeira Ordem

Regras

Redes semânticas

Ontologias

Semantic Web

BC versus BD

Raciocínio

Leitura recomendada

Conteúdo

Representação do conhecimento

Introdução

Lógica de Primeira Ordem

Regras

Redes semânticas

Ontologias

Semantic Web

BC versus BD

Raciocínio

Leitura recomendada

Introdução

- ▶ Nesta aula iremos estudar como efetuar a representação de conhecimento (RC).
- ▶ Queremos representar conhecimento para que depois o agente possa raciocinar a partir desse conhecimento.
- ▶ As abordagens de RC estão normalmente associadas a um motor de raciocínio ou inferência que usa o conhecimento armazenado para deduzir novo conhecimento.

Formalismos para RC

- ▶ Exemplos de formalismos para RC:
 - ▶ lógica de primeira ordem (LPO)
 - ▶ regras
 - ▶ redes semânticas (RS)
 - ▶ ontologias

Conteúdo

Representação do conhecimento

Introdução

Lógica de Primeira Ordem

Regras

Redes semânticas

Ontologias

Semantic Web

BC versus BD

Raciocínio

Leitura recomendada

LPO

- ▶ Uma dada RC tem associado um equilíbrio que é preciso gerir: mais **expressividade** implica normalmente maior **complexidade**.
- ▶ O formalismo mais poderoso que iremos estudar (com maior expressividade) é a lógica de primeira ordem (LPO).
- ▶ Problemas com LPO:
 - ▶ difícil de usar
 - ▶ pouco prático de implementar

Lógica proposicional

- ▶ Antes de chegarmos à LPO, temos a LP.
- ▶ A LP é um ramo da Lógica que trata o estudo das proposições.
- ▶ Uma proposição pode ser construída a partir de outras com o auxílio de **conetivas lógicas**: negação, conjunção, disjunção, implicação, etc..
- ▶ Tem origem no século 3 A.C.
- ▶ Exemplo:
 - Premissa 1: Se está a chover então está nublado.
 - Premissa 2: Está a chover.
 - Conclusão: Está nublado.

Lógica proposicional

- ▶ Tanto as premissas como a conclusão são exemplos de proposições.
- ▶ A conclusão é obtida usando uma **regra de inferência** (nesta caso, modus ponens).
- ▶ Podemos representar as proposições por letras, P ="Está a chover", Q ="Está nublado" e reescrever o exemplo assim:
 Premissa 1: $P \Rightarrow Q$
 Premissa 2: P
 Conclusão: Q

Lógica proposicional: sintaxe

- ▶ A gramática formal da lógica proposicional no formato BNF (Backus-Naur Form):

| | | |
|--------------------|---|---|
| Proposição | → | ProposiçãoAtômica ProposiçãoComplexa |
| ProposiçãoAtômica | → | True False Símbolo |
| Símbolo | → | P Q R ... |
| ProposiçãoComplexa | → | \neg Proposição $(\text{Proposição} \wedge \text{Proposição})$ $(\text{Proposição} \vee \text{Proposição})$ $(\text{Proposição} \Rightarrow \text{Proposição})$ $(\text{Proposição} \Leftrightarrow \text{Proposição})$ |

LPO

- ▶ A LPO amplia as capacidades da LP com a inclusão de quantificadores, funções, predicados, variáveis, etc. Comparar gramáticas.
- ▶ Enquanto que a LP trabalha apenas com factos, a LPO acrescenta objetos e suas relações.
- ▶ Exemplo da definição duma função:

$$\text{Bipede}(a) \Rightarrow \exists p_1, p_2, c : \text{Perna}(p_1) \wedge \text{Perna}(p_2) \wedge \text{Corpo}(c) \wedge$$

$$\text{ParteDe}(p_1, a) \wedge \text{ParteDe}(p_2, a) \wedge \text{ParteDe}(c, a) \wedge$$

$$\text{Ligado}(p_1, c) \wedge \text{Ligado}(p_2, c) \wedge$$

$$p_1 \neq p_2 \wedge [\forall p_3 \text{Perna}(p_3) \wedge \text{ParteDe}(p_3, a) \Rightarrow (p_3 = p_1 \vee p_3 = p_2)]$$

Lógica de primeira ordem: sintaxe

- ▶ A gramática formal da lógica de primeira ordem no formato BNF:

| | | |
|-------------------|---|--|
| Proposição | → | ProposiçãoAtômica $(\text{Proposição} \text{ Conectiva } \text{Proposição})$ Quantificador Variável, ... Proposição \neg Proposição |
| ProposiçãoAtômica | → | Predicado(Termo, ...) Termo = Termo |
| Termo | → | Função(Termo, ...) Constante Variável |
| Conectiva | → | \Rightarrow \wedge \vee \Leftrightarrow |
| Quantificador | → | \forall \exists |
| Constante | → | A X_1 Ana ... |
| Variável | → | a x s ... |
| Predicado | → | Antes TemCor Chover ... |
| Função | → | Bipede PaiDe ... |

LPO: categorias

- ▶ O conceito de **categoria** é fundamental para a representação de conhecimento. Permite agrupar vários objetos dentro de uma só entidade (a categoria).
- ▶ Na LPO temos 2 formas de representar categorias: **predicados** e **objetos**.
- ▶ Exemplo usando predicados: *BolasBasket(BB9)*
- ▶ Exemplo usando objetos: **reificar** (tornar num objeto) a categoria, *BolasBasket*, e depois dizemos *Membro(BB9, BolasBasket)*.

LPO: categorias

- ▶ Podemos representar a função membro com o símbolo \in , resultando o exemplo anterior em $BB9 \in BolasBasket$
- ▶ Para dizer que uma categoria é sub-classe de outra: $Subconjunto(BolasBasket, Bolas)$, ou $BolasBasket \subset Bolas$
- ▶ Os membros duma categoria possuem uma propriedade: $x \in Bolas \Rightarrow Esferico(x)$
- ▶ Reconhecer membros duma categoria a partir das suas propriedades: $Laranja(x) \wedge Esferico(x) \wedge Diametro(x) = 25 \Rightarrow x \in BolasBasket$

Conteúdo

Representação do conhecimento

Introdução

Lógica de Primeira Ordem

Regras

Redes semânticas

Ontologias

Semantic Web

BC versus BD

Raciocínio

Leitura recomendada

Regras

- ▶ Uma outra forma de representar conhecimento é a partir de regras do tipo SE-ENTÃO.
- ▶ Exemplo: SE $x > 37$ ENTÃO y
onde x poderia representar uma medição de temperatura e y uma ação como tomar um anti-pirético.
- ▶ As regras são tipicamente usadas nos sistemas periciais.

Sistemas periciais

- ▶ Usados em domínios muito específicos onde existem normalmente peritos humanos.
- ▶ As regras usadas são do tipo: SE x ENTÃO y
- ▶ O x é a **condição** e o y o **consequente**.
- ▶ Uma regra dispara se a condição for verdadeira.
- ▶ Estão divididos em dois sub-sistemas: a base de conhecimento (BC) e o motor de inferência.
- ▶ A BC guarda factos e regras.
- ▶ O SP pode fazer **dedução**: cada consequente é um novo facto.
- ▶ Ou fazer **reação**: cada consequente é uma ação.

Sistemas periciais: exemplo

- ▶ R1: SE x anda na estrada ENTÃO x é um transporte
- ▶ R2: SE x anda na estrada E x tem 2 rodas E x tem motor ENTÃO x é uma moto
- ▶ R3: SE x anda na estrada E x tem 2 rodas E x não tem motor ENTÃO x é uma bicicleta
- ▶ R4: SE x anda na estrada E x tem 4 rodas E x tem 5 lugares ENTÃO x é um carro de passageiros
- ▶ R5: SE x anda na estrada E x tem mais de 4 rodas ENTÃO x é um pesado

Sistemas periciais

- ▶ Vantagens dos SP:
 - ▶ O formato das regras é intuitivo e permite explicar as conclusões obtidas.
 - ▶ Fácil de manter: não é necessário escrever código para alterar as regras.
 - ▶ Fácil e rápido fazer protótipos.
- ▶ Desvantagens:
 - ▶ Obtenção de conhecimento: o tempo dos peritos é valioso.
 - ▶ Desempenho: corriam originalmente em sistemas interpretados (lisp).

Sistemas periciais

- ▶ Os SPs devem ser usados quando:
 - ▶ faça sentido economicamente;
 - ▶ peritos humanos não estejam sempre disponíveis;
 - ▶ o problema requeira raciocínio simbólico.

Conteúdo

Representação do conhecimento

Introdução

Lógica de Primeira Ordem

Regras

Redes semânticas

Ontologias

Semantic Web

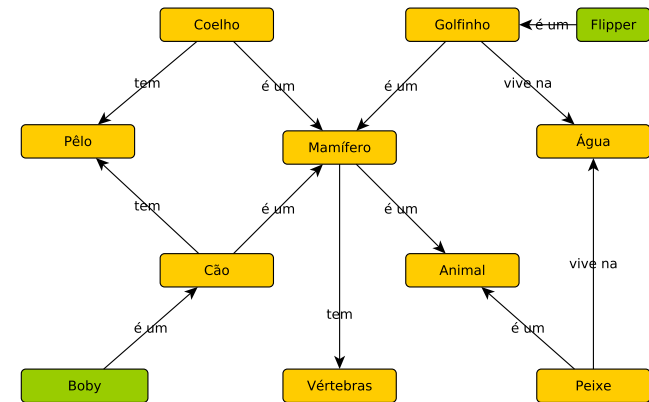
BC versus BD

Raciocínio

Leitura recomendada

Redes semânticas

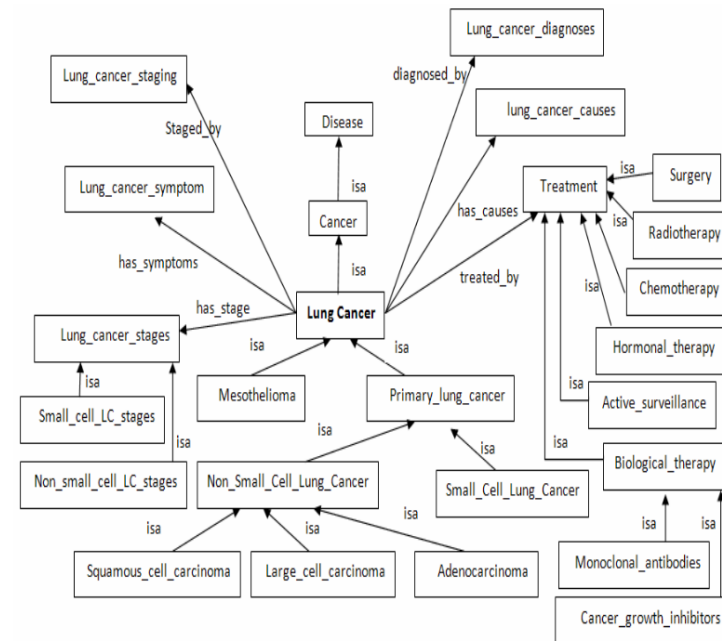
- Uma RS é um **grafo** que representa relações semânticas entre conceitos.



Redes semânticas

- A ideia original duma rede semântica apareceu em 1906 por Charles S. Peirce, com o nome de “grafo existencial”.
- Existem muitas variantes de RS mas todas permitem representar **objetos**, **categorias** de objetos e as **relações** entre eles.
- As RS tipicamente não têm todo o poder expressivo da LPO. Mas têm a vantagem de serem simples e o processo de inferência transparente.

Redes semânticas: exemplo

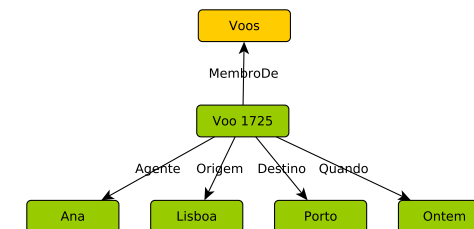


Redes semânticas

- ▶ Exemplo de inferência:
 - ▶ “Como se diagnostica um Small_Cell_lung_cancer?”
 - ▶ Encontramos esse nodo da RS e procuramos uma aresta relativa a diagnóstico: não tem. Logo vamos seguir a aresta is_a
 - ▶ Chegamos a Primary_lung_cancer que também não tem aresta relativa a diagnóstico. Seguimos aresta is_a.
 - ▶ Chegamos a Lung_cancer que tem aresta sobre diagnóstico: usa-se um Lung_cancer_diagnoses.

Redes semânticas

- ▶ As **relações** nas RS são binárias apenas, pois estão codificadas em arestas.
- ▶ Conseguimos representar relações entre 2 elementos facilmente, mas como fazer quando as relações envolvem mais que 2 elementos?
- ▶ Para resolver este problema o que se faz é **reificar a proposição**.
- ▶ Exemplo: como representar a seguinte proposição numa RS: “A Ana tomou o voo 1725 ontem de Lisboa para o Porto”?



Redes semânticas

- ▶ Alguns problemas com as RS:
 - ▶ são pesadas do ponto de vista computacional pois para responder a questões é necessário fazer a travessia da rede.
 - ▶ falta alguma capacidade de expressão: quantificadores, negação, entre outras.
- ▶ A Google tem uma RS chamada “Knowledge Graph” que em 2012 continha 570 milhões de objetos e 18 mil milhões de factos e relações entre esses objetos.
- ▶ Está disponível uma API que permite extrair informação desta RS (<https://developers.google.com/knowledge-graph>).

Conteúdo

Representação do conhecimento

Introdução
Lógica de Primeira Ordem
Regras
Redes semânticas

Ontologias

Semantic Web
BC versus BD
Raciocínio
Leitura recomendada

Ontologia

- ▶ Uma **ontologia** é uma representação de entidades e das suas relações.
- ▶ As ontologias são escritas usando linguagens próprias para descrição de ontologias.
- ▶ Uma ontologia pode ser **visualizada** num grafo.

Ontologia



Componentes duma ontologia

- ▶ Indivíduos: instâncias ou objetos
- ▶ Classes: conjuntos, coleções conceitos, tuplos de objetos ou de coisas.
- ▶ Atributos: aspetos, características, propriedades ou parâmetros de objetos ou classes
- ▶ Relações: formas segundo as quais indivíduos e classes se relacionam.
- ▶ Restrições: descrições formais que devem ser verdade para que uma dada afirmação seja aceite.
- ▶ Regras: afirmações com a forma SE-ENTÃO que descrevem uma inferência lógica.
- ▶ Axiomas: afirmações (inclui regras) que contêm toda a teoria descrita pela ontologia.
- ▶ Eventos: alterações de atributos ou relações.

Ontologia versus rede semântica

- ▶ Qual a diferença entre uma ontologia e uma rede semântica?
 - ▶ A RS é uma **notação gráfica** usada para representar conhecimento com os nodos e arestas de um grafo.
 - ▶ Uma ontologia é a representação de conceitos dentro de um domínio e das suas relações, de forma explícita e formal, que **pode ser visualizada como um grafo**, mas que existe sem qualquer relação com um grafo.

Conteúdo

Representação do conhecimento

Introdução

Lógica de Primeira Ordem

Regras

Redes semânticas

Ontologias

Semantic Web

BC versus BD

Raciocínio

Leitura recomendada

Semantic Web

- ▶ A Resource Description Framework (RDF) permite representar conhecimento.
- ▶ A Web Ontology Language (OWL) adiciona semântica e permite o uso de sistemas de raciocínio automáticos, como os classificadores.
- ▶ Enquanto que:
 - ▶ o HTML descreve documentos e ligações entre documentos,
 - ▶ a RDF + OWL + XML conseguem **descrever entidades arbitrárias** como pessoas, componentes de automóveis ou uma festa de aniversário, e as suas **relações**.

Semantic Web

- ▶ Mais recentemente, a SW surge integrando RC e raciocínio recorrendo a linguagens baseadas em XML.
- ▶ A SW é uma **extensão da web** que possibilita a partilha de dados entre aplicações, empresas e comunidades.
- ▶ Isto é conseguido **adicionando meta-dados** às páginas web, em formatos que são legíveis por máquinas.
- ▶ Isto permite que agentes / motores de pesquisa, consigam um acesso aos dados mais fácil e mais rico, resultando num maior número de tarefas realizáveis.

Semantic Web

- ▶ A ideia é adicionar uma camada de significado (semântica) sobre a internet.
- ▶ A abordagem básica usa palavras contidas nas páginas para construir índices para depois permitir a pesquisa com os motores de busca.
- ▶ Com a SW, são criadas **ontologias** de conceitos.

RDF

- ▶ A RDF é uma **família de especificações** para modelar dados.
- ▶ A ideia é semelhante à entidade-relacionamento das BDs ou aos diagramas de classes do UML.
- ▶ Na RDF fazem-se afirmações sob a forma de expressões sujeito-predicado-objeto.
- ▶ Estas expressões são chamadas **triplos**.
- ▶ O sujeito representa um recurso; o predicado representa a relação entre o sujeito e o objeto.
- ▶ Exemplo: representar “o céu tem cor azul”, usando RDF.
 - ▶ o sujeito é “o céu”
 - ▶ o predicado é “tem a cor”
 - ▶ o objeto é “azul”
- ▶ Na prática o RDF é um modelo abstrato que possui vários formatos de serialização e assim a forma de escrever o triplo varia com o formato.

RDF

- ▶ Exemplo http://www.w3schools.com/XML/xml_rdf.asp

| Title | Artist | Country | Company | Price | Year |
|------------------|--------------|---------|-------------|-------|------|
| Empire Burlesque | Bob Dylan | USA | Columbia | 10.90 | 1985 |
| Hide your heart | Bonnie Tyler | UK | CBS Records | 9.90 | 1988 |

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cd="http://www.recshop.fake/cd#">
```

```
<rdf:Description
  rdf:about="http://www.recshop.fake/cd/Empire_Burlesque">
  <cd:artist>Bob Dylan</cd:artist>
  <cd:country>USA</cd:country>
  <cd:company>Columbia</cd:company>
  <cd:price>10.90</cd:price>
  <cd:year>1985</cd:year>
</rdf:Description>
```

- ▶ Identifique os sujeitos, predicados e objetos deste exemplo.

```
<rdf:Description
  rdf:about="http://www.recshop.fake/cd/Hide_your_heart">
  <cd:artist>Bonnie Tyler</cd:artist>
  <cd:country>UK</cd:country>
  <cd:company>CBS Records</cd:company>
  <cd:price>9.90</cd:price>
  <cd:year>1988</cd:year>
</rdf:Description>
...
</rdf:RDF>
```

RDF

- ▶ Existem múltiplos formatos de serialização, entre eles:
 - ▶ Turtle
 - ▶ N-Triples
 - ▶ N-Quads
 - ▶ JSON-LD
 - ▶ Notation3 (N3)
 - ▶ RDF/XML (o primeiro formato standard para RDF)
- ▶ Note-se que o RDF não é suposto ser lido por humanos, apenas por máquinas.

Web Ontology Language

- ▶ A OWL (e não WOL!) é uma linguagem que permite **escrever ontologias**.
- ▶ A versão mais recente é de 2009 e chama-se OWL 2.
- ▶ Exemplo da representação da classe limonada em OWL2 XML (sem preâmbulos nem definições de prefixos):

```
<Ontology ontologyIRI="http://example.com/limonada.owl" ... >
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#" />
  <Declaration>
    <Class IRI="Limonada" />
  </Declaration>
</Ontology>
```

BC versus BD

- ▶ Porque não usar uma base de dados (BD) para armazenar o conhecimento para um agente de IA?
 - ▶ Uma BD é muito boa para guardar informação **quando sabemos quais as características** (atributos) que queremos guardar relativas aos dados.
 - ▶ Em IA, **não sabemos** muitas vezes todas as possíveis características que irão ser medidas: conforme o agente vai recolhendo conhecimento do mundo este tem que ser guardado.

Tipos de Raciocínio

- ▶ **Inferência**: chamamos inferência ao processo de derivação de novo conhecimento a partir de conhecimento já existente. Na IA o componente do sistema que faz inferência é o **motor de inferência**.
- ▶ **Dedutivo**: nova informação é deduzida a partir de informação com relação lógica. Ex.: Todos os homens são mortais. Sócrates é homem. Deduzimos: Sócrates é mortal.
- ▶ **Indutivo**: partir de um conjunto de observações e generalizar. Ex.: Todos os corvos que vi até hoje são pretos. Induzo: Todos os corvos são pretos.
- ▶ **Abduativo**: é uma forma de dedução que permite inferência plausível. Ex.: Ela leva guarda-chuva se chover. Hoje ela leva guarda-chuva. Conclusão: Hoje está a chover.
- ▶ **Analogia**: fazer analogias entre duas situações. Ex.: guiar um camião é semelhante a guiar um carro, mas claro que tem algumas diferenças.

Tipos de Raciocínio

- ▶ **Senso-comum**: raciocínio informal que usa regras aprendidas pela experiência (heurísticas).
- ▶ **Não-monótono**: usado quando os factos podem mudar.
 - ▶ Ex.: SE o vento sopra ENTÃO as cortinas abanam.
 - ▶ Quando o vento parar as cortinas deixam de abanar. Se usarmos o raciocínio monótono, uma vez que o vento soprasse e concluíssemos que as cortinas abanam, esse facto seria retido mesmo que o vento parasse de soprar.
 - ▶ No raciocínio não-monótono existe um mecanismo que verifica se o que levou um facto a ser verdade se mantém verdade. Se passar a falso, então os factos concluídos a partir desse facto são removidos.

Leitura recomendada

- ▶ https://en.wikipedia.org/wiki/Knowledge_representation_and_reasoning
- ▶ http://www.zeepedia.com/read.php?knowledge_representation_and_reasoning_artificial_intelligence
- ▶ Introdução ao RDF: <https://www.w3.org/TR/rdf-syntax>
- ▶ Introdução à OWL 2: <https://www.w3.org/TR/owl2-overview>
- ▶ Russell e Norvig, sec. 12.5.