



UNIVERSITY OF
LIVERPOOL

Aprendizado Exato de Ontologias em Lógica Descritiva

16 de junho de 2016

OUTLINE

Introdução

Modelo de Comunicação

Aprendizado Exato

Nossos Resultados

Conclusão

INTRODUÇÃO

- ▶ Ontologias em Lógica Descritiva (LD)
 - ▶ Formalismo para representação de conhecimento baseado em lógica.
 - ▶ Usadas para descrever um domínio em termos de **conceitos** (predicado unário) e **papéis** (predicado binário).

INTRODUÇÃO

Exemplo:

- ▶ Vocabulário:
 - ▶ Conceitos: Acadêmico
 - ▶ Papéis: temOrientador
- ▶ Ontologia em Lógica Descritiva:
 - ▶ Acadêmico $\sqsubseteq \exists$ temOrientador.Acadêmico
- ▶ Tradução:

$$\forall x(\text{Acadêmico}(x) \rightarrow \exists y(\text{temOrientador}(x, y) \wedge \text{Acadêmico}(y)))$$

EXEMPLOS DE ONTOLOGIAS

- ▶ Nomenclatura Sistematizada de Medicina, Termos Clínicos (SNOMED CT): 500.000 conceitos [Ruch et al., 2008]
- ▶ Enciclopédia do Instituto Nacional de Câncer (NCI) dos Estados Unidos: 50.000 conceitos [Golbeck et al., 2011]
- ▶ Genética: 25.000 conceitos [Consortium et al., 2013]

SNOMED CT

Exemplo de um axioma definindo o termo Tuberculose Pulmonar em SNOMED CT (sintaxe LD):

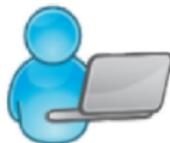
$$\begin{aligned} \textit{PulmonaryTuberculosis} \quad \equiv \quad & \textit{Disease} \sqcap \\ & \exists \textit{causativeAgent}.\textit{MycobacteriumTuberculosisComplex} \sqcap \\ & \exists \textit{findingSite}.\textit{LungStructure} \end{aligned}$$

Tradução para Lógica de Primeira Ordem:

$$\forall(x)(\textit{PulmonaryTuberculosis}(x) \leftrightarrow (\textit{Disease}(x) \wedge \exists y_1(\textit{causativeAgent}(x, y_1) \wedge \textit{MycobacteriumTuberculosisComplex}(y_1)) \wedge \exists y_2(\textit{findingSite}(x, y_2) \wedge \textit{LungStructure}(y_2))))))$$

CONSTRUÇÃO DE ONTOLOGIAS

- ▶ O processo de construção de ontologias de grande porte é altamente complexo.
 - ▶ Engenheiro de Ontologia - Não conhece o domínio
 - ▶ Especialista no Domínio - Não conhece as linguagens usadas para construção de ontologias



Ontology
Engineer

Communication



Domain
Expert

CONSTRUÇÃO DE ONTOLOGIAS

- ▶ Neste trabalho, consideramos um modelo simples do processo de comunicação e analisamos, dentro do modelo, a complexidade computacional de construir uma ontologia.



Ontology
Engineer

Communication



Domain
Expert

OUTLINE

Introdução

Modelo de Comunicação

Aprendizado Exato

Nossos Resultados

Conclusão

MODELO DE COMUNICAÇÃO

► Premissas:

- o especialista no domínio conhece o domínio e seu vocabulário sem saber como formalizá-lo como uma ontologia;
- o especialista no domínio informa ao engenheiro de ontologia os termos relevantes do domínio, o vocabulário.



Ontology
Engineer

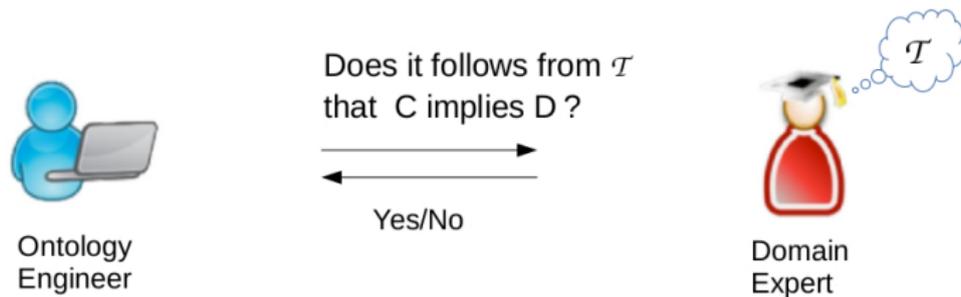
Communication



Domain
Expert

MODELO DE COMUNICAÇÃO

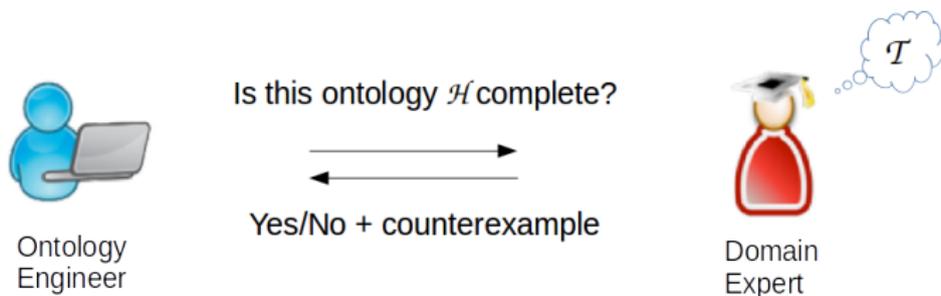
- ▶ O engenheiro de ontologia pode perguntar se um conhecimento conceitual é verdadeiro no domínio.



Em símbolos: $\mathcal{T} \models C \sqsubseteq D?$

MODELO DE COMUNICAÇÃO

- ▶ O engenheiro de ontologia também precisa saber se a ontologia construída está completa e, se não estiver, o engenheiro pede um contraexemplo.



Em símbolos: $\mathcal{H} \equiv \mathcal{T}$?

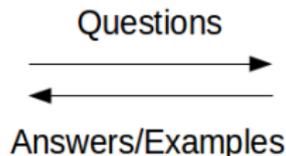
ENUNCIADO DO PROBLEMA

Dado esse modelo, onde se assume que o vocabulário e a linguagem da lógica descritiva são conhecidos:

- ▶ Pode o engenheiro de ontologia construir uma ontologia que reflita o conhecimento do especialista no domínio?
- ▶ Quantas perguntas são necessárias para isso?



Ontology
Engineer



Domain
Expert

OUTLINE

Introdução

Modelo de Comunicação

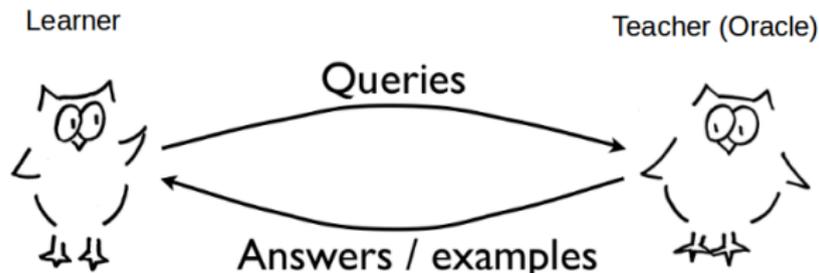
Aprendizado Exato

Nossos Resultados

Conclusão

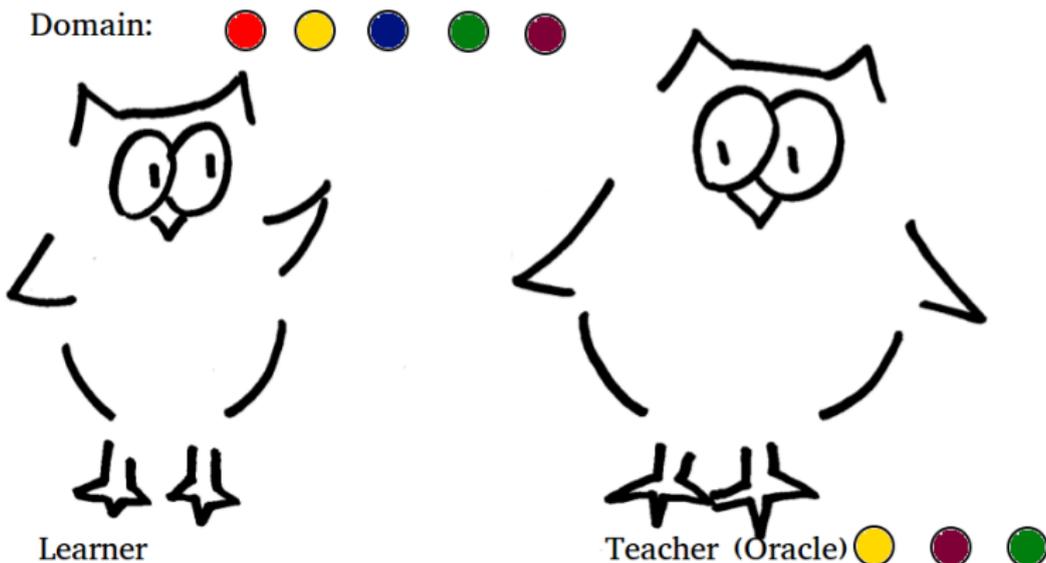
APRENDIZADO EXATO (ANGLUIN)

- ▶ Um algoritmo aprende exatamente um conjunto alvo L_* se sempre termina e retorna uma hipótese L_h tal que $L_h = L_*$.
 - ▶ Pergunta de Pertinência: $x \in L_*$? Sim/Não
 - ▶ Pergunta de Equivalência: $L_h = L_*$? Sim/Não e $x \in L_h \oplus L_*$



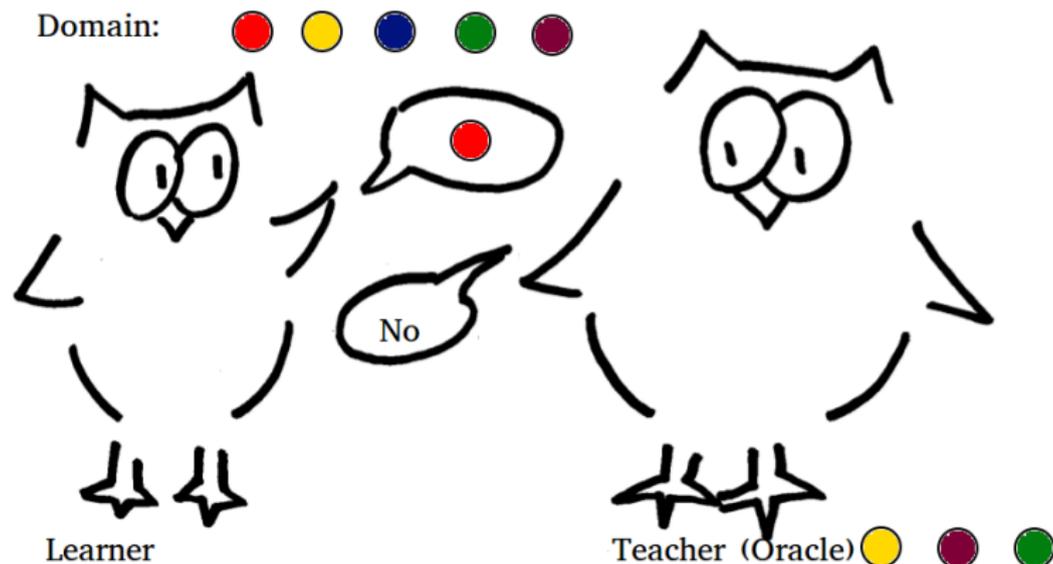
APRENDIZADO EXATO

- ▶ Pergunta de Pertinência: $x \in L_*$? Sim/Não
- ▶ Pergunta de Equivalência: $L_h = L_*$? Sim/Não e $x \in L_h \oplus L_*$



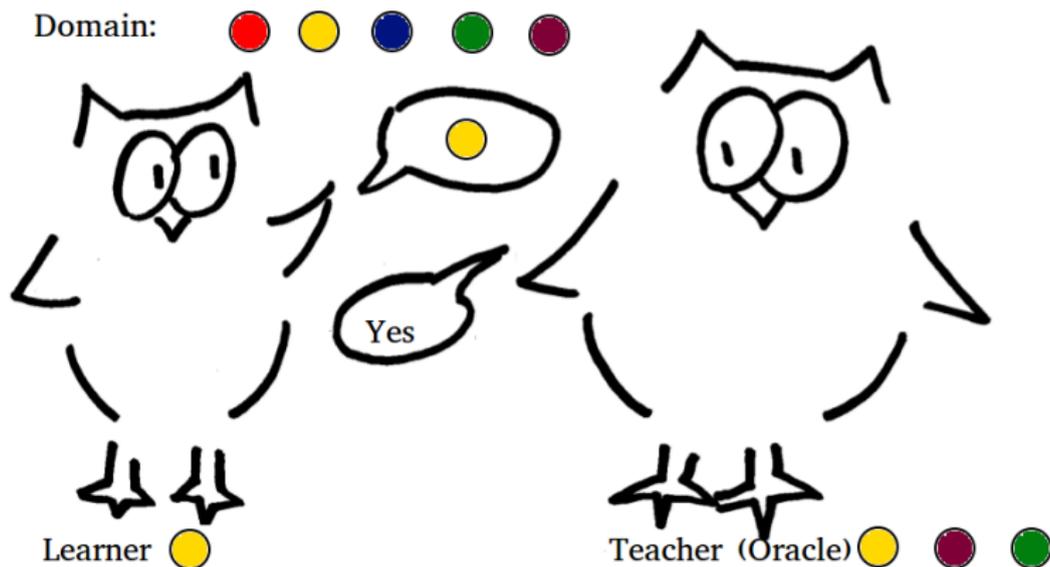
APRENDIZADO EXATO

- ▶ Pergunta de Pertinência: $x \in L_*$? Sim/Não
- ▶ Pergunta de Equivalência: $L_h = L_*$? Sim/Não e $x \in L_h \oplus L_*$



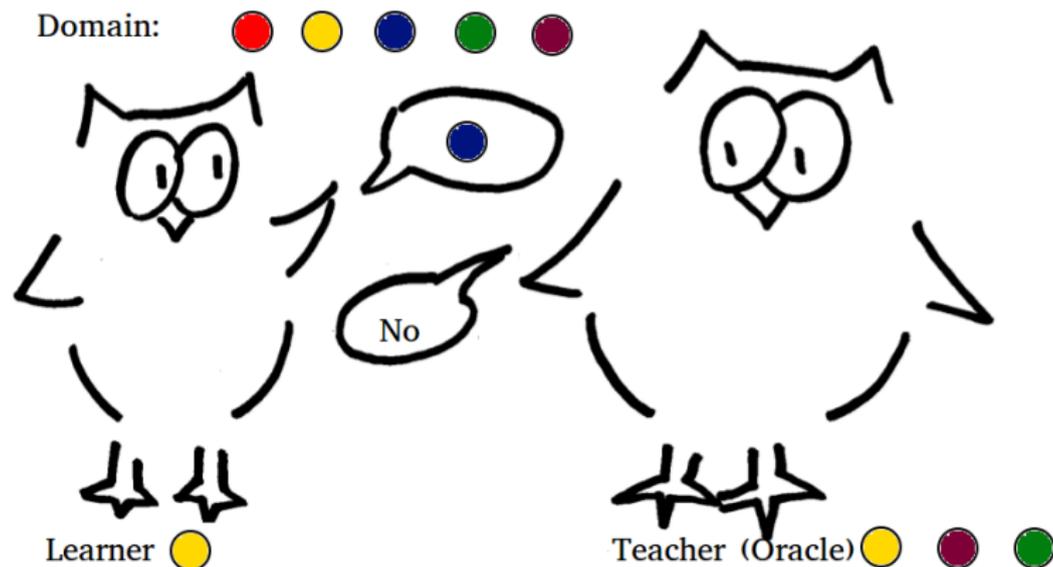
APRENDIZADO EXATO

- ▶ Pergunta de Pertinência: $x \in L_*$? Sim/Não
- ▶ Pergunta de Equivalência: $L_h = L_*$? Sim/Não e $x \in L_h \oplus L_*$



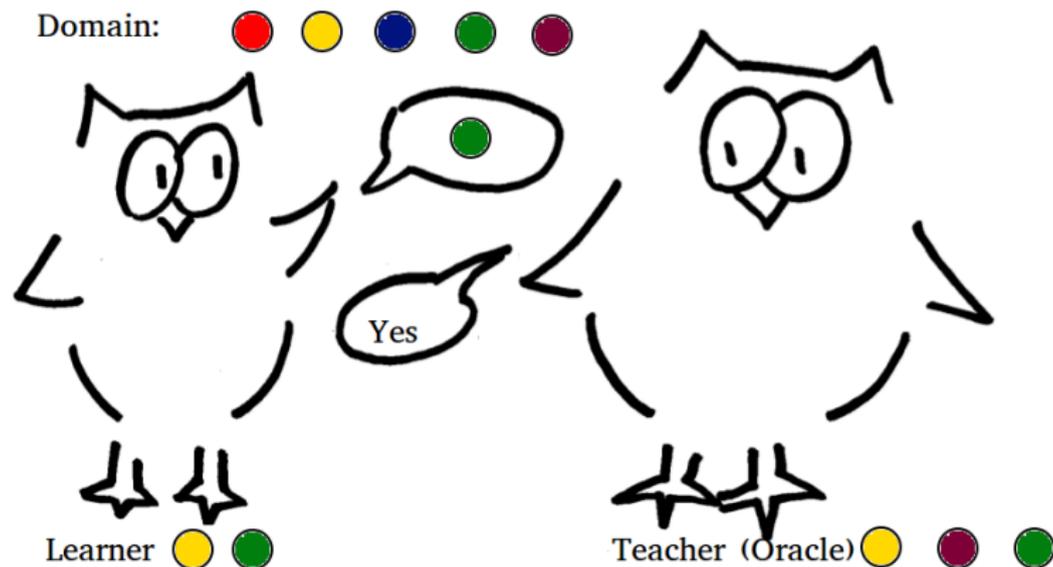
APRENDIZADO EXATO

- ▶ Pergunta de Pertinência: $x \in L_*$? Sim/Não
- ▶ Pergunta de Equivalência: $L_h = L_*$? Sim/Não e $x \in L_h \oplus L_*$



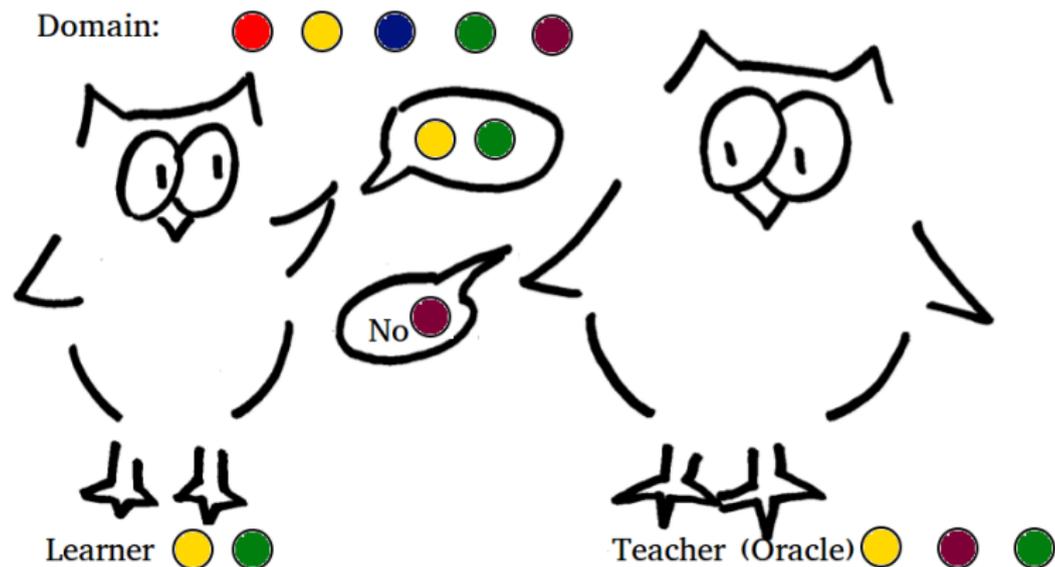
APRENDIZADO EXATO

- ▶ Pergunta de Pertinência: $x \in L_*$? Sim/Não
- ▶ Pergunta de Equivalência: $L_h = L_*$? Sim/Não e $x \in L_h \oplus L_*$



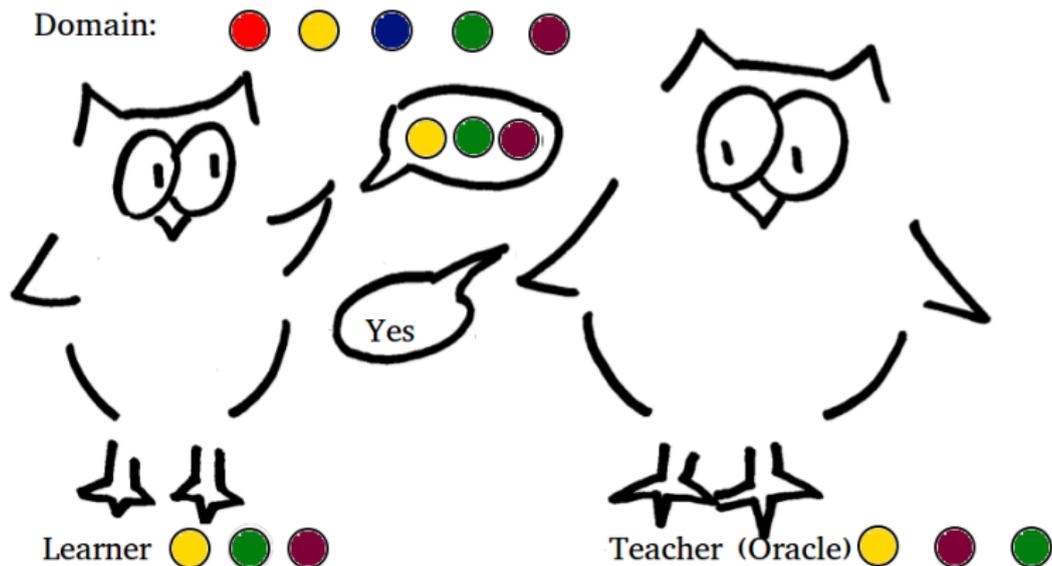
APRENDIZADO EXATO

- ▶ Pergunta de Pertinência: $x \in L_*$? Sim/Não
- ▶ Pergunta de Equivalência: $L_h = L_*$? Sim/Não e $x \in L_h \oplus L_*$

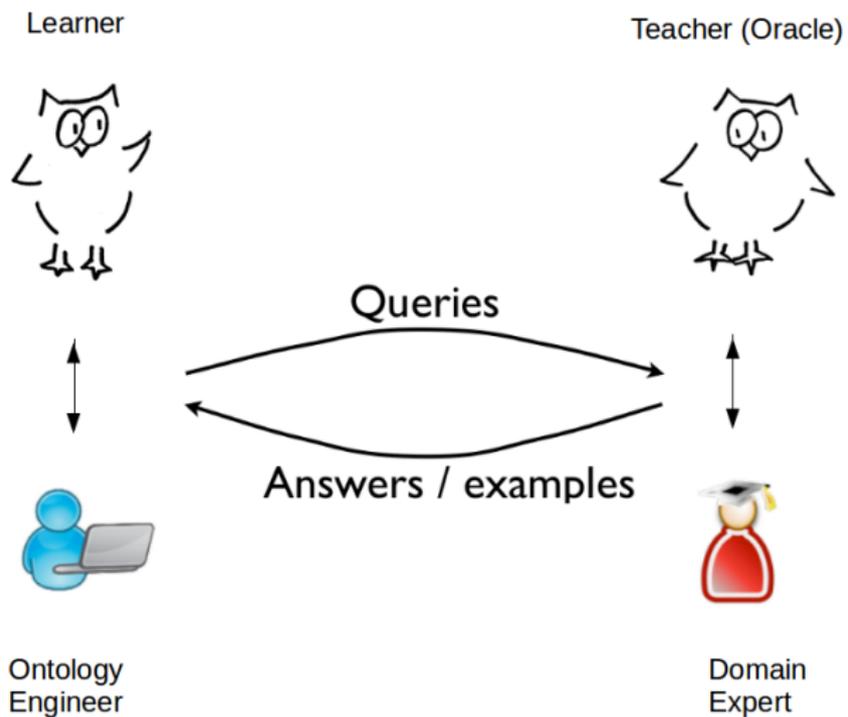


EXACT LEARNING

- ▶ Membership query: $x \in L_*$? Yes/No
- ▶ Equivalence query: $L_h = L_*$? Yes/No and $x \in L_h \oplus L_*$



APRENDIZADO EXATO



APRENDIZADO EXATO DE ONTOLOGIAS EM LD

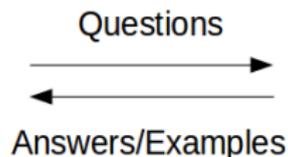
- ▶ Elementos de conjuntos
 - ▶ Pergunta de Pertinência: $x \in L_*$? Sim/Não
 - ▶ Pergunta de Equivalência: $L_h = L_*$? Sim/Não e $x \in L_h \oplus L_*$
- ▶ Inclusão de conceitos
 - ▶ Pergunta de Pertinência: $\mathcal{T} \models C \sqsubseteq D$? Sim/Não
 - ▶ Pergunta de Equivalência: $\mathcal{H} \equiv \mathcal{T}$? Sim/Não e o oráculo retorna $C \sqsubseteq D$ tal que $\mathcal{T} \models C \sqsubseteq D$ e $\mathcal{H} \not\models C \sqsubseteq D$ (um *contraexemplo positivo*), ou vice versa (um *contraexemplo negativo*).

APRENDIZADO EXATO

- ▶ Aprendizado em tempo polinomial:
 - ▶ Polinomial no tamanho do alvo de aprendizado e do maior contraexemplo visto até o momento.



Ontology
Engineer



Domain
Expert

APRENDIZADO EXATO

- ▶ Aprendizado em tempo polinomial:
 - ▶ Linguagens Regulares [Angluin, 1987],
 - ▶ Horn Proposicional [Angluin et al., 1992],
 - ▶ Horn de Primeira Ordem Não-Recursivo [Reddy and Tadepalli, 1999],
 - ▶ Mapeamentos de Schemas de Banco de Dados [ten Cate et al., 2012].

- ▶ Em aberto:
 - ▶ FNC e FND Booleano,
 - ▶ Horn de Primeira Ordem (brecha entre ‘lower bound’ e melhor algoritmo conhecido).

OUTLINE

Introdução

Modelo de Comunicação

Aprendizado Exato

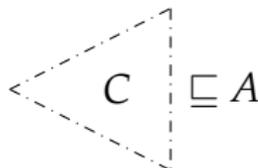
Nossos Resultados

Conclusão

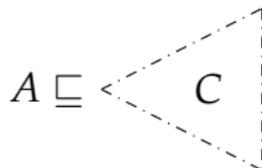
LINGUAGENS DA LÓGICA DESCRITIVA

 \mathcal{EL}


$$\exists r.A \sqsubseteq \exists s.A'$$

 $\mathcal{EL}_{\text{lhs}}$


$$\exists r.A' \sqsubseteq A$$

 $\mathcal{EL}_{\text{rhs}}$


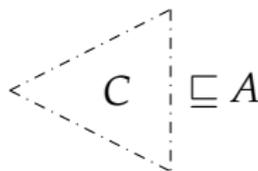
$$A \sqsubseteq \exists r.A'$$

- ▶ Expressões de conceitos em \mathcal{EL} :
 - ▶ $C, D := A \mid \top \mid C \sqcap D \mid \exists r.C$
- ▶ Problema SAT é trivial.
- ▶ Problema $\mathcal{T} \models C \sqsubseteq D$ é polinomial.

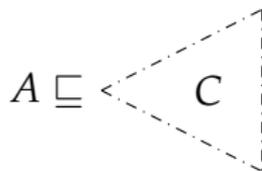
PRIMEIRAS OBSERVAÇÕES DO PROBLEMA

 \mathcal{EL}


$$\exists r.A \sqsubseteq \exists s.A'$$

 $\mathcal{EL}_{\text{lhs}}$


$$\exists r.A' \sqsubseteq A$$

 $\mathcal{EL}_{\text{rhs}}$


$$A \sqsubseteq \exists r.A'$$

► Expressões de conceitos em \mathcal{EL} :

$$\triangleright C, D := A \mid \top \mid C \sqcap D \mid \exists r.C$$

1. Com um mesmo vocabulário pode-se criar expressões de conceitos de tamanho arbitrário.
2. Seria inviável perguntar todos os axiomas possíveis em uma das linguagens, mesmo conhecendo o vocabulário.

PRIMEIRAS OBSERVAÇÕES DO PROBLEMA

- ▶ Suponha que estejamos tentando aprender a seguinte ontologia formulada em $\mathcal{EL}_{\text{lhs}}$.
- ▶ $\mathcal{T} = \{\exists r.A \sqsubseteq A\}$

O seguinte algoritmo poderia não terminar.

inicio

$\mathcal{H} = \emptyset$

enquanto $\mathcal{H} \not\equiv \mathcal{T}$ **faça**

 Seja $C \sqsubseteq A$ o contraexemplo (positivo)

$\mathcal{H} := \mathcal{H} \cup \{C \sqsubseteq A\}$

fim

fin

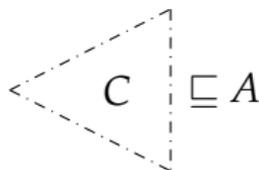
retorna \mathcal{H}

O oráculo pode retornar uma sequência infinita de contraexemplos $\exists r^n.A \sqsubseteq A$, sendo n um número primo.

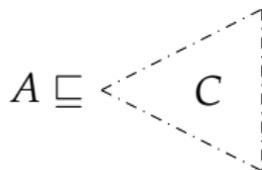
NOSSOS RESULTADOS

 \mathcal{EL}


$$\exists r. A \subseteq \exists s. A'$$

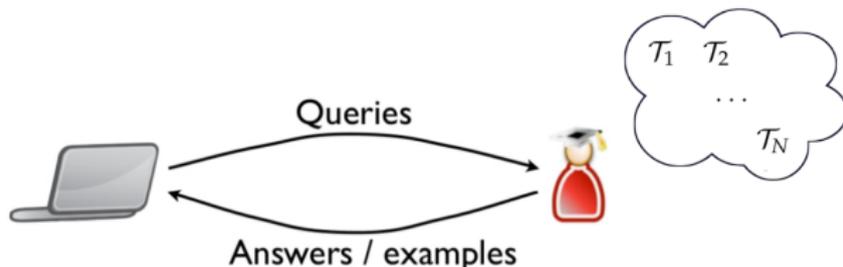
 $\mathcal{EL}_{\text{lhs}}$


$$\exists r. A' \subseteq A$$

 $\mathcal{EL}_{\text{rhs}}$


$$A \subseteq \exists r. A'$$

- $\mathcal{EL}_{\text{lhs}}$ e $\mathcal{EL}_{\text{rhs}}$: é possível aprender em tempo polinomial.



NOSSOS RESULTADOS

- ▶ $\mathcal{EL}_{\text{lhs}}$ e $\mathcal{EL}_{\text{rhs}}$: é possível aprender em tempo polinomial.

inicio

| $\mathcal{H} = \emptyset$

| **enquanto** $\mathcal{H} \neq \mathcal{T}$ **faça**

| | Refina o contraexemplo $C \sqsubseteq D$ com perguntas

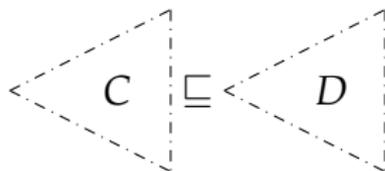
| | Atualize \mathcal{H} com o contraexemplo refinado

| **fim**

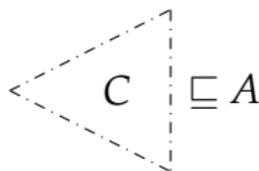
fin

retorna \mathcal{H}

NOSSOS RESULTADOS

 \mathcal{EL}


$$\exists r. A \subseteq \exists s. A'$$

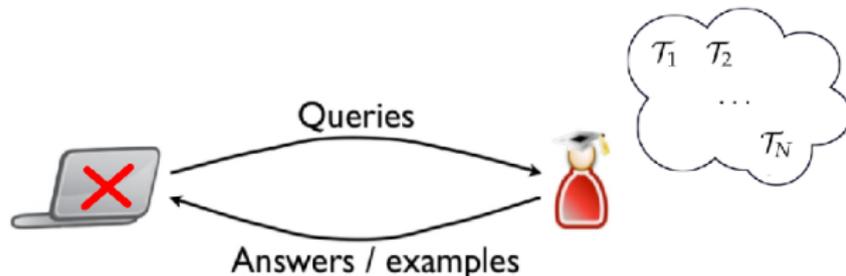
 $\mathcal{EL}_{\text{lhs}}$


$$\exists r. A' \subseteq A$$

 $\mathcal{EL}_{\text{rhs}}$


$$A \subseteq \exists r. A'$$

- \mathcal{EL} : não é possível aprender em tempo polinomial.



OUTLINE

Introdução

Modelo de Comunicação

Aprendizado Exato

Nossos Resultados

Conclusão

CONCLUSÃO

- ▶ Neste trabalho, tratamos o problem de construir uma ontologia como um problema de aprendizado.
- ▶ Investigamos a complexidade de aprender ontologias em diversas linguagens da lógica descritiva.
- ▶ Classificação:
 - ▶ $\mathcal{EL}_{\text{lhs}}$: ✓
 - ▶ $\mathcal{EL}_{\text{rhs}}$: ✓
 - ▶ \mathcal{EL} : ✗

TRABALHOS PUBLICADOS

Tese de Doutorado:

- ▶ Exact Learning of Lightweight Description Logic Ontologies - KR 2014;
- ▶ Exact Learning Description Logic Ontologies from Data Retrieval Examples - DL Workshop 2015;
- ▶ A Model for Learning Description Logic Ontologies Based on Exact Learning - AAI 2016.

Outros Projetos:

- ▶ Exact Learning of Multivalued Dependencies - ALT 2015;
- ▶ On Metric Temporal Description Logics - ECAI 2016.

OUTROS PROJETOS

Em andamento:

- ▶ Metric Temporal Logic Translations;
- ▶ Rewritability in Modal Logic.

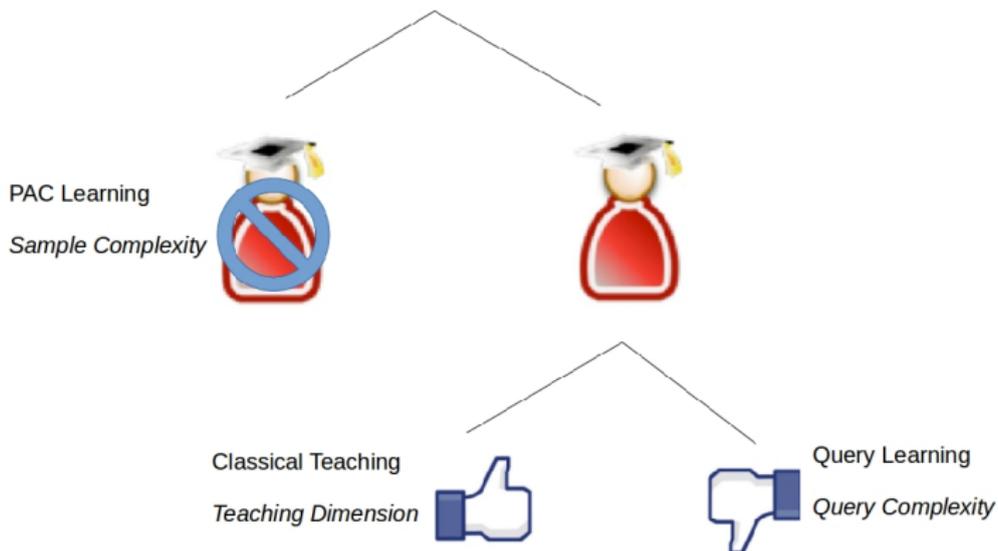
Futuros:

- ▶ Projeto Posdoc: Dimensão Vapnik-Chervonenkis de Ontologias em Lógica Descritiva;
- ▶ ?

PERGUNTAS

Obrigada!

QUAL É A 'COMPLEXIDADE' DE APRENDER ONTOLOGIAS EM LÓGICA DESCRITIVA?



-  Angluin, D. (1987).
Learning regular sets from queries and counterexamples.
Inf. Comput., 75(2):87–106.
-  Angluin, D., Frazier, M., and Pitt, L. (1992).
Learning conjunctions of Horn clauses.
Machine Learning, 9:147–164.
-  Consortium, G. O. et al. (2013).
Gene ontology annotations and resources.
Nucleic acids research, 41(D1):D530–D535.
-  Golbeck, J., Frago, G., Hartel, F., Hendler, J., Oberthaler, J., and Parsia, B. (2011).
The national cancer institute's thesaurus and ontology.
Web Semantics: Science, Services and Agents on the World Wide Web, 1(1).
-  Reddy, C. and Tadepalli, P. (1999).

Learning Horn definitions: Theory and an application to planning.

New Generation Comput., 17(1):77–98.



Ruch, P., Gobeill, J., Lovis, C., and Geissbühler, A. (2008). Automatic medical encoding with snomed categories. *BMC medical informatics and decision making*, 8(1):1.



ten Cate, B., Dalmau, V., and Kolaitis, P. G. (2012). Learning schema mappings. In *ICDT*, pages 182–195.