

# Relatório de Inteligência Artificial

## Trabalho 2 - Question Answering

Flavio Cruz, Hugo Carneiro, Mariana Werneck, Thiago Ururay

<sup>1</sup>Instituto de Computação – Universidade Federal Fluminense (UFF)

{flaviocruz, hugocarneiro, marianawerneck, taraujo}@id.uff.br

**Resumo.** *O objetivo desse relatório é fazer a implementação, análise e comparação entre dois algoritmos de questionário, utilizando como base de conhecimento dos algoritmos o artigo da Wikipedia sobre a biografia de Henrique VIII, um famoso monarca inglês. Os algoritmos a serem comparados são forward chaining utilizando um grafo de conhecimento e uma máquina de inferência em Prolog.*

### 1. Introdução

Iniciamos o nosso trabalho com a escolha do texto, a biografia de Henrique VIII, retirado do artigo sobre o mesmo na Wikipedia. A escolha foi feita baseada na complexidade das relações políticas daquela época, especialmente com relação a um dos personagens mais icônicos da monarquia inglesa. Sendo responsável pelo rompimento com a igreja católica e criação da igreja anglicana, uma série de casamentos em busca de um herdeiro homem e incontáveis traições, a vida de Henrique VIII é marcada por intrigas políticas dentro e fora da corte inglesa.

Com a decisão do tema do trabalho, nos propomos a construir um grafo baseado no texto selecionado descrevendo as relações entre os vários personagens da corte inglesa e Henrique VIII. Para análise do artigo nos baseamos na utilização da base de conhecimento DBpedia em conjunto com a ferramenta LODmilla para visualização dos grafos, que serão explicados em mais detalhes na próxima sessão.

### 2. Metodologia

Antes de falar sobre a metodologia propriamente dita, vamos discorrer um pouco sobre as ferramentas utilizadas. Para a obtenção da nossa base de conhecimento, utilizamos a DBpedia, um projeto iniciado na Universidade de Leipzig para extração de conteúdo de páginas da Wikipédia. Enquanto a maioria do conteúdo disponível uma página comum da Wikipedia seja considerado informação pública, DBpedia também trata estruturas pertencentes à página, tais como índices, links, tabelas, imagens, etc [Auer et al. 2007]. Logicamente, a facilidade de interação com a plataforma foi o motivo principal para a utilização da DBpedia.

Entre os aplicativos que são disponibilizados para os usuários da ferramenta encontra-se o LODmilla, um serviço de visualização de grafos LOD (Link Open Data) desenvolvido pela MTA SZTAKI (Hungarian Academy of Sciences Institute for Computer Science and Control) para ser utilizado em conjunto com a DBpedia [Micsik et al. 2013].

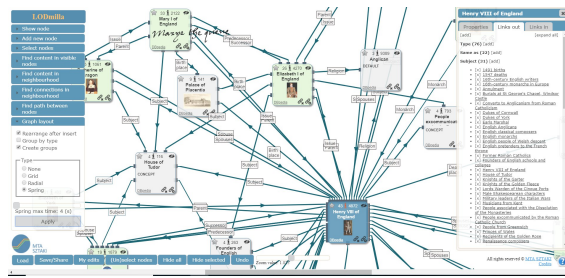


Figure 1. LODmilla utilizado em conjunto com DBpedia

Sendo a LODmilla criada para gerar grafos através da informação recebida pela DBpedia e a DBpedia feita para extrair dados diretamente da Wikipedia, não tivemos nenhum problema em utilizar diretamente as informações retiradas do artigo sobre Henrique VIII.

A partir do artigo da Wikipedia, LODmilla cria um grafo baseado nas relações existentes entre os objetos descritos no artigo. As arestas desses grafos descrevem a natureza da relação entre dois vértices quaisquer, de forma a gerar a representação sujeito-predicado que será explorada para o desenvolvimento do algoritmo. Baseados na estrutura supracitada nos dividimos entre a implementação de dois métodos de questionário, que serão avaliados mais tarde: a utilização do algoritmo de Forward Chaining em conjunto com o grafo de conhecimento e, utilizando o paradigma da programação lógica, implementar uma máquina de inferência em Prolog utilizando como fatos as relações sujeito-predicado descritas no grafo.

## 2.1. Prolog

Prolog é a mais famosa das linguagens de programação que utilizam o paradigma de programação lógica[Clocksin and Mellish 2012]. Ao contrário das linguagens procedurais, que são baseadas em comandos sequenciais, programação lógica se baseia em fatos lógicos e regras de inferência. Com relação ao nosso trabalho, esses fatos são baseados nas relações do grafo de conhecimento utilizado.

```

henriqueviii_wiki.pl
File Edit Browse Compile Prolog Pce Help
henriqueviii_wiki.pl
pai(henrique_viii,eduardo_i).
mae(isabel_blount,henrique_fitzRoy).
mae(ana_boleyn,isabel_i).
mae(catarina_de_aragao,maria_i).
mae(joana_seymour,eduardo_i).
mae(isabel_de_york,henrique_viii).
homem(eduardo_i).
homem(henrique_fitzRoy).
homem(henrique_viii).
homem(henrique_viii).
mulher(maria_i).
mulher(isabel_i).
relacionamento_amoroso(henrique_viii,isabel_blount).
relacionamento_amoroso(henrique_viii,catarina_de_aragao).
relacionamento_amoroso(henrique_viii,ana_boleyn).
relacionamento_amoroso(henrique_viii,joana_seymour).
relacionamento_amoroso(henrique_viii,ana_de_cleves).
relacionamento_amoroso(henrique_viii,catarina_howard).
relacionamento_amoroso(henrique_viii,catarina_pars).
relacionamento_amoroso(henrique_viii,maria_boleyn).

bastardo(X) :- pai(henrique_viii,X), mae(Y,X), \+ casou(henrique_viii,Y).
avo(X,Y) :- pai(X,Z),pai(Z,Y).
asistente(X,Y) :- relacionamento_amoroso(Y,X), \+ casou(Y,X).
direito_ao_trono(X) :- homem(X),pai(henrique_viii,X),mae(Y,X),casou(henrique_viii,Y).

user:relacionamento_amoroso/2: (loaded) static, 8 clauses, number_of_rules(0), last_modified_generation(3050), defined
Line: 31

```

Figure 2. Implementação do algoritmo em prolog

As regras de inferência são utilizadas para que a máquina possa tirar conclusões lógicas a partir dos fatos, no caso do trabalho isso significa fornecer o contexto do texto no

qual os fatos foram baseados. A única outra ferramenta que o Prolog fornece é a consulta, que utilizaremos para fazer os questionamentos do trabalho.

## 2.2. Forward Chaining

Forward Chaining é uma estratégia de implementação altamente utilizada em máquinas de inferência. O oposto do Backward Chaining, a ideia do algoritmo pode ser interpretada como uma série de iterações utilizando a regra de *Modus Ponens*, isto é, se utilizar de atribuições já dadas como verdadeiras no sistema para inferir verdades mais complexas[Salvat 1998].

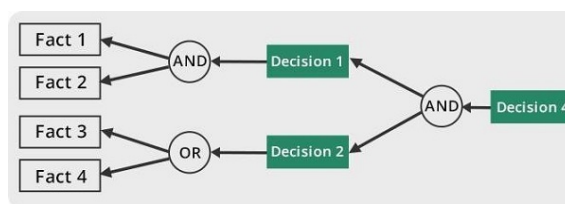


Figure 3. Descrição visual do Forward Chaining

Nós escolhemos fazer a implementação do Forward Chaining em Python3, e conforme as regras do trabalho nos utilizamos somente dos fatos lógicos dados pelo grafo. Infelizmente, devido a um erro do LODmilla o grafo precisou ser implementado à mão, e utilizamos a estrutura de dicionário como uma substituição para uma lista de adjacência para determiná-lo. Para que as relações ficassem mais claras, cada nó chave possui uma lista de valores que indicam o significado de cada aresta no grafo original, com sua própria lista de nós que completam a relação.

```

graph = {
  'henryVIII': {
    'spouse': [
      'catherineparr', 'janeseymour', 'catherineofaragon', 'anneboleyn', 'catherinehoward', 'ar
    ],
    'sex': 'male',
    'relationships': [
      'catherineparr', 'janeseymour', 'catherineofaragon', 'anneboleyn', 'catherinehoward', 'ar
      'isabelblount'
    ],
    'parent': [
      'henryfitzroy', 'elizabethI', 'maryI', 'edwardVI'
    ],
    'successor': [
      'edwardVI'
    ],
    'issue': [
      'henryfitzroy', 'elizabethI', 'maryI', 'edwardVI'
    ]
  },
  'henryVII': {
    'spouse': [
      'elizabethofyork'
    ],
    'sex': 'male',
    'parent': [
      'henryVIII'
    ],
  },
}

```

Figure 4. Grafo representado como uma lista de adjacência

## 3. Resultados

Tendo montado o grafo de conhecimento, cada membro do grupo escolheu uma pergunta para ser feita para os programas:

- *Quem herdou o trono de Henrique VIII?*
- *Quem foi filho bastardo de Henrique VIII?*
- *Quem foi amante de Henrique VIII?*

- *Quem foi avô de Henrique VII?*

Para poder responder essas perguntas em Prolog, o grupo implementou regras de inferência para que a máquina pudesse obter contexto sobre as perguntas. Dessa forma, por exemplo, foi possível determinar que um herdeiro ao trono teria de ser um filho legítimo e homem. Após essa fase, o programa não teve dificuldade em responder as perguntas.

Em Python, nos utilizamos da estrutura de dicionário para apurar as cláusulas necessárias para o algoritmo poder realizar as inferências lógicas. Além disso, como o Python não é projetado para consultas como Prolog, precisamos tratar as entradas de forma a evitar erros ou confusões que seriam possíveis em linguagem natural. De fato, as perguntas foram implementadas dentro do programa para evitar confusão entre os membros do grupo.

#### **4. Conclusão**

Prolog se revelou uma linguagem bem adaptada para a consulta de questões lógicas, possuindo uma máquina de inferência excepcionalmente robusta, além de não exigir tratamentos de entrada como Python. Mesmo com o Forward Chaining, a implementação em Prolog se revelou o melhor algoritmo.

#### **References**

- Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.
- Clocksin, W. F. and Mellish, C. S. (2012). *Programming in Prolog: Using the ISO standard*. Springer Science & Business Media.
- Micsik, A., Tóth, Z., and Turbucz, S. (2013). Lodmilla: Shared visualization of linked open data. In *International Conference on Theory and Practice of Digital Libraries*, pages 89–100. Springer.
- Salvat, E. (1998). Theorem proving using graph operations in the conceptual graph formalism. In *ECAI*, pages 356–360.