Universidade do Minho

Mestrado em Engenharia Informática

Processamento de Linguagens e Conhecimento

Scripting no Processamento de Linguagem Natural

João Pimentel A80874

4 de Maio de 2020

Resumo

O presente projeto consistiu no desenvolvimento de métodos para analisar as interações entre personagens de um livro, utilizando uma ontologia. Para tal, foram utilizadas as linguagens de programação *Python* e as bibliotecas *Owlready2* e *NLTK*. Numa fase de análise de resultados, observou-se que a solução implementada cumpriu todos os objetivos propostos, apesar de a biblioteca *Owlready2* ter algumas falhas na definição de condições de inferência de novas classes.

Conteúdo

Lista de Figuras

Lista de Extratos

1 Introdução

O presente relatório é o resultado da resolução do segundo trabalho prático da unidade curricular de *Scripting* no Processamento de Linguagem Natural, do perfil de Processamento de Linguagens e Conhecimento. O foco deste trabalho passou por utilizar uma biblioteca previamente atribuída num *script* em contexto de processamento de linguagem natural. Para tal, teve-se por base a utilização das funcionalidades fornecidas pela linguagem de programação *Python*, bem como a biblioteca *OwlReady2*.

A linha de pensamento base da resolução do projeto passou por encontrar encontrar um problema dentro do contexto pedido, que permitisse uma resolução com base em ontologias. Deste modo, utilizando o problema associado ao primeiro projeto da unidade curricular, ou seja, a contagem de relações entre personagens de um livro, apenas foi necessário pensar na estruturação da resolução. Assim, tendo em conta que uma personagem pode ser caraterizada por vários nomes, a utilização de uma ontologia permite lidar com estes casos de forma simples e rápida.

O grande objetivo deste projeto consistiu no aumento da experiência dos alunos no âmbito do desenvolvimento de código *Python* para processamento de textos de linguagem natural. Além disso, o projeto visava a consolidação da utilização de bibliotecas de linguagem natural e ontologias, realçando a utilidade e simplicidade destas para a resolução deste tipo de problemas.

Nos capítulos seguintes serão demonstrados os problemas, formas de resolução, métodos de desenvolvimento e testes efetuados de modo a obter os resultados ideais para o problema proposto.

2 Análise e Especificação

2.1 Descrição do Projeto

O projeto em questão consistiu no desenvolvimento de um *script* no contexto de processamento de linguagem natural que tirasse proveito da biblioteca *OwlReady2*, permitindo contabilizar as relações entre as várias personagens principais de um livro. No que toca a este, foi utilizado um dos livros da saga *Harry Potter* como ferramenta do caso de estudo.

2.2 Especificação de Requisitos

O principal objetivo deste projeto foi encontrar as personagens mais importantes num livro, sendo esta importância definida pelo número de interações que cada personagem efetua com as restantes. Assim, serão apresentados, em seguida, os requisitos associados ao desenvolvimento de forma detalhada:

1. Deve ser utilizada a biblioteca *OwlReady2*, bem como a linguagem de programação *Python*.
2. Devem ser contabilizadas as interações que as personagens efetuam com as restantes personagens.
3. Uma personagem deve ser considerada personagem principal se interagir com mais de 5 personagens.
4. Uma interação é definida como a existência dos nomes de duas ou mais personagens numa mesma frase.
5. A interação entre duas personagens deve ser definida por um valor de ocorrência.
6. Devem ser contabilizadas interações mesmo que o nome da personagem numa determinada frase seja um *alias* do seu nome mais comum.

3 Concepção da Resolução

Ao longo deste capítulo serão apresentados os métodos associados a cada tarefa do projeto desenvolvido, bem como a forma de funcionamento destes, problemas encontrados e forma de resolver os mesmos.

3.1 Definição da Ontologia

O primeiro passo para a resolução do problema foi idealizar a ontologia que seria a base de todo o processamento dos dados. Tendo em conta que se pretendia calcular relações entre personagens, sendo estas relações definidas por um valor numérico, existem duas classes a implementar: Personagem e Interação. Caso não fosse necessário contabilizar as interações, esta classe não seria necessária, passando a existir na forma de relação simétrica entre duas personagens. Além disso, como se pode observar pela Figura 1, existe uma classe específica para as personagens principais, sendo que uma personagem principal é, também, uma personagem. É importante realçar que uma personagem além do seu nome possui mais nomes pelos quais é conhecido, sendo estes utilizados para obter um valor mais realista do número de interações ao longo do livro.

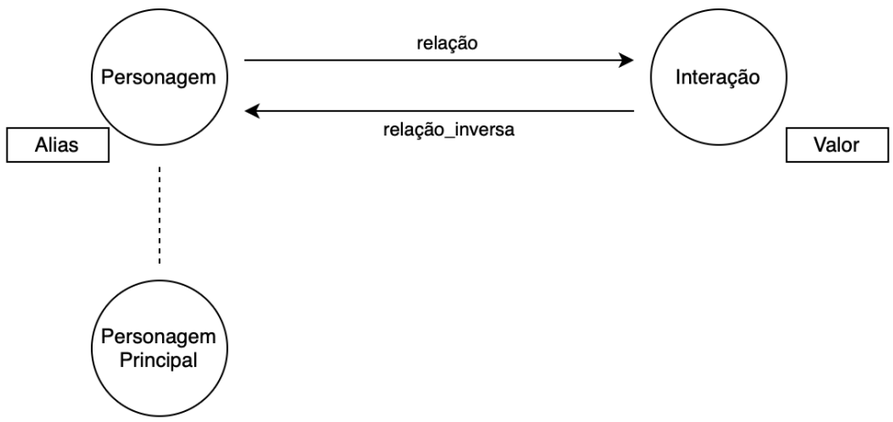

Figura 1 - Esquema da ontologia.

No que toca à codificação da ontologia idealizada, existem 3 passos cruciais, sendo estes a definição de classes, *data properties* e *object properties*.

No que toca às classes, veja-se o Extrato 1, onde a classe personagem é definida como parte de *Thing*, estando no *namespace* da ontologia definida para o problema.

```
class Personagem(Thing):
    namespace = onto
```

Extrato 1 - Código referente à classe Personagem.

Já no caso das *data properties*, pela análise do Extrato 2 é possível observar que esta se trata de uma lista de *strings* e está no universo dos atributos na ontologia.

```
class alias(DataProperty):
    namespace = onto
    range = [str]
```

Extrato 2 - Código referente à *data property alias*.

Por fim, no que toca às relações entre classes, ou seja, as *object properties*, veja-se o Extrato 3. Neste é visível que se trata de propriedade com domínio na classe Interação e subdomínio em Personagem. Além disso, possui uma propriedade inversa em *relação*, ou seja, esta vai de *Personagem* para *Interação*.

```
class relacao_inverse(Interacao >> Personagem):
    namespace = onto
    inverse_property = relacao
```

Extrato 3 - Código referente à *object property relacao_inverse*.

3.2 Processamento do Texto

Tendo a ontologia definida, o passo seguinte é processar o texto, sendo este passo crucial em qualquer aplicação de linguagem natural. Desta forma, após a leitura completa do ficheiro, é aplicada uma função do módulo *NLTK*, de modo a serem obtidas todas as *named entities*, ou seja, possíveis personagens do livro em estudo. Como se trata de uma área de estudo em que os resultados não são 100% corretos, é necessária uma análise extra destes valores. Assim, foi definida uma função de *clustering*, em que recebendo um nome, uma lista de entidades e um dicionário para armazenar o resultado, é efetuada uma pesquisa de verosimilhança na lista de entidades pelo nome. Para isso, cada entidade presente na lista foi dividida por espaços, permitindo uma análise palavra a palavra. Em seguida, com auxílio do módulo *difflib*, foi aplicada a função *get_close_matches* que permite obter uma lista de palavras semelhantes à palavra em estudo. No caso desta lista não ser vazia, é assumido que aquela entidade se trata de um *alias* do nome em estudo, sendo adicionada à lista de *alias* da personagem no dicionário. Por fim, tendo o dicionário completo, este é convertido para o formato ontológico, permitindo a análise das interações.

Tendo a ontologia povoada, a partir das entidades previamente obtidas, é necessário dividir o texto em frases. Tendo estas no formato de lista, cada frase tem que ser analisada de forma a serem obtidas as *named entities* lá presentes. Por cada par de entidades presentes, é efetuada uma pesquisa na ontologia de modo a verificar a qual instância pertence o *alias* em análise. Caso ambas as personagens existam na ontologia, é testado se existe já uma relação entre as mesmas. No caso positivo, o valor associado à interação é incrementado. Caso contrário, é criada uma nova instância de interação, associada ao par de personagens em questão, com valor de 1.

3.3 Classificação de Personagens

O último passo para a resolução do projeto era o cálculo das personagens principais a partir dos dados presentes na ontologia. Assim, observando a definição da classe personagem principal presente no Extrato 4, uma personagem principal é uma sub-classe de Personagem. Além disso, tem que possuir, no mínimo, 6 interações. É interessante realçar que as classes definidas com auxílio da biblioteca *Owlready2* permitem a execução de funções próprias de cada classe, como é o caso da função *info*, onde são impressos no ecrã o nome da personagem, os seus *alias* e com quem interagiu.

```
class PersonagemPrincipal(onto.Personagem):
    equivalent_to = [
        Personagem
        & relacao.min(6, Interacao)
    ]

    def info(self):
        print(f"{self.name} Also known as: {self.alias}")
        for r in self.relacao:
            print(f"{set(r.relacao_inverse)} Interactions: {r.value}")
```

Extrato 4 - Código referente à classe *PersonagemPrincipal*.

Note-se que, de modo a calcular estas personagens, é necessário sincronizar o *reasoner* do módulo, já que as instâncias da classe não foram explicitamente declaradas. Assim, após a chamada deste *reasoner*, observando a Figura 2, é visível que tendo em conta a condição definida, as personagens principais são Harry Potter e Hagrid. Note-se que, caso a condição fosse "cada personagem deve possuir pelo menos uma interação com valor de N", os resultados seriam diferentes e talvez mais realistas. No entanto, não foi encontrada forma de codificar uma condição deste género, sendo esta uma funcionalidade em falha no módulo.

```
Hagrid Also known as: ['Hagrid', 'Hagrid Hagrid']
{book.Harry, book.Hagrid} Interactions: 57
{book.Dumbledore, book.Hagrid} Interactions: 5
{book.Voldmort, book.Hagrid} Interactions: 1
{book.Ron, book.Hagrid} Interactions: 3
{book.Hermione, book.Hagrid} Interactions: 4
{book.Snape, book.Hagrid} Interactions: 2
Harry Also known as: ['Harry', 'Mr. Harry Potter', 'Should Harry', 'Mr. Harry',
'Harry Potter', 'Daily Prophet Harry']
{book.Harry, book.Dumbledore} Interactions: 70
{book.Harry, book.Hagrid} Interactions: 57
{book.Harry, book.Ron} Interactions: 74
{book.Harry, book.Hermione} Interactions: 50
{book.Harry, book.Snape} Interactions: 23
{book.Harry, book.Draco} Interactions: 4
{book.Voldmort, book.Harry} Interactions: 5
```

Figura 2 - Resultado do cálculo das personagens principais.

4 Conclusões e Trabalho Futuro

Ao longo do presente relatório encontra-se representado o resultado do segundo trabalho prático da unidade curricular de *Scripting* no Processamento de Linguagem Natural. Neste contexto foram abordados os passos referentes ao desenvolvimento e implementação de vários métodos que permitissem a utilização de uma ontologia para cálculo de interações entre personagens de um livro. Foi, desta forma, necessário idealizar e codificar uma ontologia, tendo-se como ferramenta o módulo *Owlready2*.

O desenvolvimento do projeto permitiu aprimorar os conhecimentos previamente adquiridos e postos em prática no decorrer das aulas da Unidade Curricular. Desta forma, foi possível melhorar o primeiro trabalho prático, bem como confirmar a facilidade de utilização de ontologias no contexto de processamento de linguagem natural. Além disso, o módulo em estudo mostrou-se capaz de fornecer informação no formato ontológico de forma simples e clara, ligando-se à linguagem de programação *Python* no contexto de programação orientada a objetos, na forma de classes.

Em suma, a realização deste trabalho exigiu a aplicação de todos os conhecimentos lecionados em contexto de aula, permitindo que o grupo cumprisse todos os objetivos propostos no enunciado, conciliando assim a teoria e a prática.