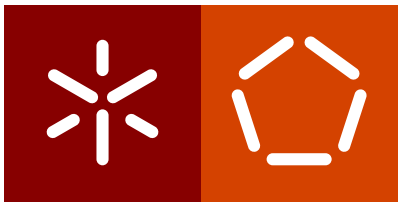
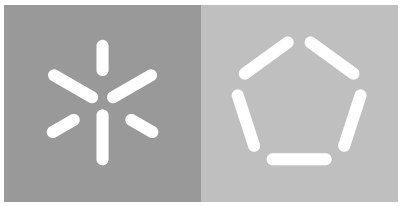


Universidade do Minho
Escola de Engenharia
Departamento de Informática

Shahzod Yusupov

Uma Ontologia para a Descrição de Conteúdos de Testamentos

Maio 2022



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Shahzod Yusupov

Uma Ontologia para a Descrição de Conteúdos de Testamentos

Dissertação de Mestrado
Mestrado Integrado em Engenharia Informática

Dissertação supervisionada por
Professor Doutor Orlando Manuel de Oliveira Belo
Professora Doutora Anabela Leal de Barros

Maio 2022

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico.

Eu confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

RESUMO

Cada vez é mais notória a importância que as ontologias têm vindo a ganhar no que toca ao desenvolvimento de sistemas baseados em conhecimento. Para além de ainda haver alguma dificuldade em compreender o seu modo de implementação, a sua construção manual é muito dispendiosa tanto a nível de recursos como de tempo e, após a construção, é necessário manter a ontologia atualizada consoante os novos requisitos que poderão surgir. Nesta dissertação apresentamos, numa primeira parte, a definição de ontologia, a sua utilidade e algumas das metodologias que podem ser utilizadas na sua construção manual, analisando a sua evolução ao longo do tempo. Após esta introdução, apresentamos algumas técnicas de construção (semi-)automática de ontologias a partir de textos e abordamos o conceito de *ontology learning*, bem como tudo aquilo que este processo envolve. Além disso, enunciaremos alguns dos sistemas que fazem uso dessas mesmas técnicas. Por fim, apresentamos o trabalho desenvolvido na extração de uma ontologia a partir de um conjunto de textos relativos a testamentos antigos, que foram editados por Barros e Alves (2019) em *O Livro dos Testamentos – Picote, 1780-1803*, detalhando o processo de extração realizado para a ontologia pretendida, bem como apresentando as técnicas e ferramentas utlizadas. Neste processo, queremos relevar a importância da utilização de padrões léxico-sintáticos e o *dependency parsing*, que contribuíram de forma efetiva para a obtenção dos resultados que alcançámos.

PALAVRAS-CHAVE Ontologias, Extração de Ontologias, Processamento de linguagem natural, *Ontology Learning*, Padrões léxico-sintáticos, *Dependency parsing*, Textos não estruturados, Base de dados orientada a grafos, Neo4J, Spacy

ABSTRACT

The importance that ontologies have gained in terms of knowledge-based systems development is increasingly evident. In addition to the difficulties that still exist in understanding how to build ontologies, their manual construction is very costly not only in terms of resources but also in time and, after their construction, it is necessary to keep them updated according to new requirements that may arise. In this dissertation we'll present, in the first part, the definition of ontology, its usefulness, and some methodologies for its manual construction as well as the possibility of its evolution. After this introduction to the concept of ontology some techniques for its (semi-) automatic construction from texts will be presented in which the concept of ontology learning will be introduced as well as everything that this process involves and some systems that make use of these techniques. Finally, will be presented the work developed in the extraction of an ontology from old testaments, which were edited by Barros and Alves (2019), O Livro dos Testamentos – Picote, detailing the process carried out to extract the ontology intended as well as presenting the techniques and tools used in this process. It is important to highlight, in this process, the importance of using lexical-syntactic patterns and the dependency parsing that effectively contributed to the achievement of the obtained results.

KEYWORDS Ontology, Ontology Extraction, Ontology Learning, Natural Language Processing, Lexico-syntactic patterns, Dependency parsing, Unstructured texts, Graph database, Neo4J, Spacy

CONTEÚDO

LISTA DE FIGURAS

LISTA DE TABELAS

1

INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Nos últimos anos, o aumento no tamanho e complexidade dos sistemas de computação, bases de conhecimento e principalmente da internet e toda a informação que nela circula criou a necessidade de um mecanismo que facilite a comunicação entre componentes heterogéneos. O aspeto fundamental de troca de informação entre sistemas, aplicações e serviços é o desenvolvimento de um modelo consistente e compreensivo para a representação do conhecimento do domínio (Khattak et al., 2013). As ontologias têm vindo a ganhar cada vez mais popularidade e reconhecimento e têm um papel fundamental nos sistemas modernos baseados em conhecimento pois constituem uma ferramenta poderosa para o suporte de processamento de linguagem natural, filtragem e recuperação de informação e a acesso de dados (Drumond and Girardi, 2008). Ontologias podem ser consideradas como sendo um conjunto de termos num determinado domínio e os relacionamentos entre esses termos. Em outras palavras ontologias são esquemas de metadados que fornecem um vocabulário controlado de conceitos, cada um com uma semântica explicitamente definida e processável por máquinas (Hazman et al., 2008). Uma das maiores aplicações das ontologias é a Web Semântica, uma geração da Web na qual a semântica dos documentos, na maioria das vezes, atualmente expressa apenas em linguagem natural, seria expressa através de ontologias. Dessa forma, a Web Semântica pode ser vista como uma abordagem para aumentar a eficácia do acesso à informação na web (Drumond and Girardi, 2008). No entanto, para além de ainda haver certa dificuldade em compreender o método de construção das ontologias, a construção manual destas é muito dispendiosa tanto a nível de recursos como de tempo e, após a construção, é necessário manter a ontologia atualizada consoante os novos requisitos que poderão surgir. De modo a ultrapassar estes obstáculos, outros métodos abordam o processo de extração de ontologias através de técnicas não supervisionadas, baseadas em métodos estatísticos e ferramentas linguísticas básicas ou extração de informação não supervisionada (Drymonas et al., 2010). Estes tipos de metodologias fazem parte de uma área designada de *Ontology Learning*, que consiste no suporte automático ou semiautomático para a construção de ontologias e tem como objetivo a descoberta de conhecimento a partir

de diferentes fontes de dados e a sua posterior representação numa estrutura ontológica. O principal desafio deste tipo de abordagens está em alcançar uma cobertura satisfatória do domínio em termos de conceitos e relações conceituais, reduzindo o esforço humano ao mínimo absoluto (Drymonas et al., 2010).

1.2 Motivação e Objetivos

Dentro do contexto referido na secção anterior, o trabalho apresentado nesta dissertação teve como principal objetivo a implementação e desenvolvimento de um sistema semiautomático capaz de extrair a informação presente em testamentos antigos, séc. XVIII, mais especificamente os editados por Barros e Alves (2019), *O Livro dos Testamentos – Picote 1780-1803*, e representá-la numa estrutura ontológica. *O Livro dos Testamentos de Picote* (1780-1803) é um códice encadernado a pergaminho, de 30,8 por 20 centímetros, composto por cem fólios, dos quais se chegaram a preencher apenas setenta e seis. Durante a segunda metade do século XX permaneceu no Museu da Terra de Miranda, encontrando-se desde 2013 no Arquivo Municipal de Miranda do Douro. Composto por dois cadernos, singelamente cosidos a meio com cordel fino, foi previamente preparado para utilização como Livro das Notas dos Testamentos de Picote, redigidas pelo escrivão Manuel Domingues (Alves and Barros, 2019). O conteúdo destes documentos baseia-se nos testamentos que os habitantes de Picote, uma freguesia de Miranda do Douro, mandavam fazer, normalmente, quando se encontravam débeis e acamados. Esses testamentos eram então redigidos pelo escrivão Manuel Domingues, na presença de algumas testemunhas que, normalmente, eram também habitantes de Picote. Estando estes documentos escritos em Português do séc. XVIII, sendo alguns bastante longos, a extração manual de informação torna-se uma tarefa árdua e bastante demorada. De modo a tentar resolver este problema, com o desenvolvimento deste sistema ontológico, pretendia-se criar uma ferramenta que fosse capaz de servir como auxílio na análise do conteúdo dos testamentos, fornecendo uma interface gráfica simples e intuitiva, que possibilitasse a aquisição de conhecimento no domínio dos testamentos.

1.3 Estrutura da Dissertação

Para além do presente capítulo, esta dissertação integra mais 4 capítulos, nomeadamente:

- Capítulo 2 – Sistemas Ontológicos, em que é introduzido o conceito de ontologia, apresentando-se algumas das definições usadas na literatura e os vários contextos nos quais podemos utilizar as ontologias. Após esta introdução são apresentados alguns exemplos e tipos de ontologias existentes e, posteriormente, é explicado o processo de desenvolvimento de uma ontologia. Por fim, apresentam-se algumas ferramentas que

podem ser utilizadas no desenvolvimento de ontologias, explicando-se o seu processo de aplicação.

- Capítulo 3 – Extração de uma Ontologia de Testamentos -, no qual se apresenta o processo de extração automática de ontologias, introduzindo-se o conceito de *Ontology Learning* e apresentando-se algumas técnicas e sistemas para este domínio. Este capítulo termina com a apresentação de algumas abordagens e ferramentas para a avaliação de ontologias.

- Capítulo 4 – Extração de uma Ontologia de Testamentos -, neste capítulo é apresentado e explicado o trabalho desenvolvido na extração de uma ontologia a partir de textos de testamento antigos, abordando-se as tecnologias, frameworks e técnicas utlizadas, até à sua posterior representação e exploração num sistema de grafos. Além disso, apresentam-se os testes que foram realizados e as métricas utilizadas na análise dos resultados obtidos.

- Capítulo 5 – Conclusões e Trabalho Futuro – em que se apresentam e analisam os resultados obtidos e se expõem algumas linhas de atuação para trabalho futuro.

1.4 TRABALHO REALIZADO

De acordo com a motivação apresentada e os objetivos definidos, nesta dissertação idealizámos, desenvolvemos e implementámos um sistema de processamento de textos capaz de extrair a informação presente nos testamentos editados por Barros e Alves (2019), O Livro dos Testamentos – Picote, 1780-1803, e construir de forma semiautomática uma ontologia com o conhecimento contido nesses textos. Dada a natureza dos textos (testamentos) a processar, considerámos que a informação que seria importante extrair estaria relacionada com o indivíduo que escreveu o testamento, aquele que o mandou fazer, a data e o local onde o testamento foi elaborado, os herdeiros e legatários do testador, as testemunhas presentes e, por fim, mas não menos importante, a herança e os deveres deixados pelo testador. Após a extração da informação para uma estrutura de triplos semânticos (Sujeito, Predicado, Objeto), esta foi posteriormente representada num sistema de grafos, de forma a facilitar a sua exploração. Concluído o trabalho de extração e representação, foram efetuados vários testes e utilizadas algumas métricas de forma a avaliar os resultados obtidos e, para além disso, foi feita uma comparação destes resultados com o resultado obtido da aplicação do sistema desenvolvido em testamentos antigos de outras localidades. É importante mencionar que, inicialmente, antes da aplicação de qualquer técnica, foi feito um estudo aprofundado sobre a área das ontologias e, consequentemente, sobre os processos de extração destas a partir de textos.

2

SISTEMAS ONTOLÓGICOS

2.1 DEFINIÇÃO E UTILIZAÇÃO

O termo ontologia tem origem no grego ontos ('ser') + logos ('conhecimento') + -ia. Este termo foi introduzido na filosofia, no século XIX, pelo filósofo alemão Rudolf Gockel, no seu "Lexicon Philosophicum", para distinguir o estudo do ser do estudo dos vários tipos de seres nas ciências naturais. (Breitman et al., 2007). A definição de ontologia que mais frequentemente é citada na literatura da Web Semântica é a que foi apresentada por (Gruber, 1993):

Uma ontologia é uma especificação formal e explícita de uma conceptualização compartilhada.

Nesta definição, a conceptualização representa um modelo abstrato de algum fenómeno no mundo que identifica conceitos relevantes daquele fenómeno; explícito, que significa que os elementos devem ser claramente definidos, e formal, que indica que a especificação deve ser processável por máquina. Indo um pouco mais além, na visão de Gruber, uma ontologia é a representação do conhecimento de um domínio, na qual um conjunto de objetos e seus relacionamentos são descritos por um vocabulário (Breitman et al., 2007). Em termos gerais, uma ontologia, juntamente com um conjunto de instâncias individuais de classes, constitui uma base de conhecimento (Noy and McGuinness, 2001). São várias as áreas de aplicação das ontologias. Em (Öhgren, 2004) podemos encontrar algumas dessas aplicações, que foram propostas por diversos autores, de entre os quais Obitko. Segundo Obitko (2001), as ontologias podem ser usadas num leque muito diversificado de áreas, para suporte à realização de várias tarefas, como, por exemplo:

- Expressar termos de domínio geral numa estrutura de alto nível;
- Partilhar e reutilizar conhecimento;
- Suportar a comunicação entre sistemas multiagente;
- Compreender a linguagem natural;

- Facilitar a pesquisa de documentos.

Uschold e Gruninger (1996) especificaram três diferentes categorias de aplicação nas quais se podem utilizar ontologias, nomeadamente:

1. Comunicação – em que as ontologias podem ser usadas para facilitar e aumentar a comunicação entre pessoas, criando uma rede de relacionamentos, para acompanhar o que está vinculado e usar isso para navegar e explorar;
2. Interoperabilidade – na qual as ontologias podem servir como um ambiente de integração para diferentes ferramentas de *software*;
3. Engenharia de Sistemas - área em que as ontologias desempenham um papel importante no design e desenvolvimento de sistemas de *software*. Como tal, podem ajudar a identificar os requisitos de um sistema e definir explicitamente os relacionamentos que possam existir entre as suas componentes.

Além de Uschold e Gruninger (1996), McGuinness (2005) também identificou muitas outras áreas de aplicação para ontologias. De referir as seguintes:

- Navegação, browsing e suporte de pesquisa;
- Verificação de consistência;
- Fornecimento de suporte de configuração;
- Validação de suporte;
- Teste de verificação de dados.

2.2 ONTOLOGIAS – TIPOS, EXEMPLOS E APLICAÇÕES

Na literatura podemos encontrar diversas classificações de ontologias, propostas por diversos autores, nem sempre coincidindo nos termos e nas definições usadas. Em (Öhgren, 2004) estão especificados alguns estudos feitos sobre esta matéria, realizados por diversos autores, como Obitko (2001). Este autor defende que as ontologias podem ser classificadas como:

- Ontologias de Local de Trabalho, que nos permitem especificar as condições de fronteira que caracterizam e justificam o comportamento de resolução de problemas no local de trabalho.
- Ontologias de Tarefas, que consistem em vocabulários para a descrição de estruturas de resolução de problemas de todas as tarefas existentes, independentes do domínio; o conhecimento da tarefa atribui funções a cada objeto e às relações entre estes.

- Ontologias de Domínio, que modelam um domínio específico, representando os significados dos termos aplicados ao domínio em questão; este tipo de ontologia pode ser dividida em:
  - Tarefas dependentes, que contêm algum conhecimento específico do domínio para ser capaz de resolver um problema;
  - Tarefas Independentes, que cobrem a estrutura ou comportamento de um objeto ou teorias e princípios que governam um domínio, para mencionar alguns;
- Ontologias Gerais, que cobrem objetos gerais ou comuns, tais como coisas, eventos, tempo, espaço, etc.

Em (Breitman et al., 2007) podem-se encontrar outras classificações feitas por outros autores como (Gómez-Pérez et al. 2004), os quais propõem uma classificação baseada no tipo de informação representada pela ontologia, nomeadamente:

- Ontologias de Representação de Conhecimento, que fornecem elementos primitivos de modelação de modelos de representação de conhecimento e oferecem as construções de modelação usada em representações baseadas em frames, como classes, subclasses, valores, atributos (slots) e axiomas.
- Ontologias de uso geral e comum, que representam o conhecimento de senso comum que pode ser usado em diferentes domínios. Normalmente, estas ontologias incluem um vocabulário que relaciona classes, eventos, espaço, casualidade e comportamento, entre outros conceitos.
- Ontologias Superiores — com base num conjunto de diferenças acerca de como cada ontologia superior trata os conceitos, o IEEE propôs a criação de um grupo de trabalho cujo foco era a criação de uma ontologia superior padrão conhecida como SUO.
- Ontologias de domínio, que oferecem conceitos que podem ser reutilizados dentro de um domínio específico (médico, jurídico, entre outros). A fronteira entre uma ontologia superior e uma ontologia de domínio deve ser clara.
- Ontologias de Tarefa, que descrevem o vocabulário relacionado com uma tarefa ou atividade.
- Ontologias de Método, que fornecem definições para conceitos e relacionamentos relevantes para um processo;
- Ontologias de Aplicação, que contêm todos os conceitos necessários para modelar a aplicação em questão. Este tipo de ontologia é usado para especializar e alargar ontologias de domínio ou tarefa para uma aplicação específica.

Com o objetivo de analisar a estrutura de uma ontologia, de seguida iremos apresentar um exemplo de uma ontologia no domínio do cinema, que foi construída utilizando um editor open-source de ontologias designado por Protégé (Mason, 2015).

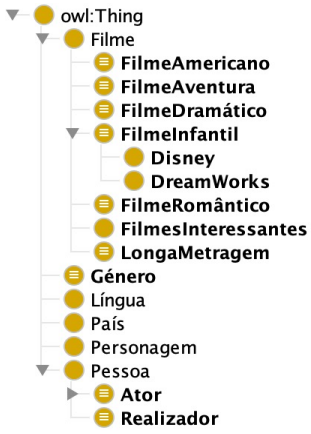


Figura 1: Uma Ontologia sobre cinema.

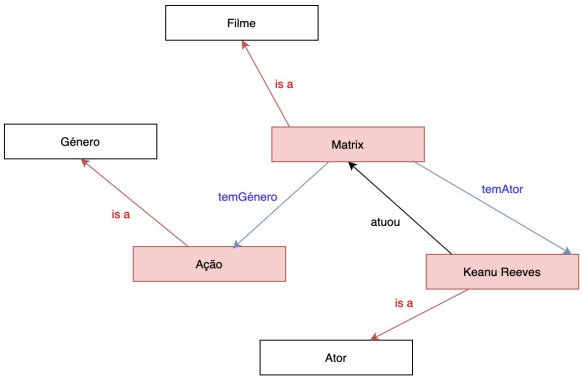


Figura 2: Representação gráfica de parte da ontologia sobre cinema.

No contexto cinematográfico, temos a classe principal "Filme" que pode ser dividida em várias subclasses, consoante o tipo de filme em questão (ex: "filme romântico", "filme infantil", etc.). Cada filme tem um género específico, uma língua, um país onde foi produzido, os atores que fazem parte deste, as personagens que os atores representam nele e ainda o seu realizador.

Para representar as características do filme, são criadas as classes específicas "Género", "Língua", "País", etc., e as relações "temGénero", "temLíngua", "temPaís", etc., para fazer os relacionamentos entre as classes "Filme" e as classes que acabamos de referir. Assim, por exemplo, podemos ter que 'John Wick', instância da classe "Filme", "temPaís"

'Estados Unidos da América', instância da classe "País" ou 'John Wick' "temGénero" 'Ação'. Posteriormente, são criadas instâncias de cada classe.

Para acolher a definição dos atores e realizadores de cada filme foi criada a classe "Pessoa", que tem como subclasses "Ator" e "Realizador". A classe "Ator" é ainda dividida em "AtorMasculino" e "AtorFeminino", consoante o género do ator. Já para representar as personagens que cada ator representa no filme foi criada a classe "Personagem" e a relação "temPersonagem" entre as classes "Ator" e "Personagem". As classes, para além de se relacionarem entre si, possuem ainda propriedades intrínsecas.

Utilizando como exemplo a classe "Ator", esta encontra-se associada a um nome, uma idade, uma morada, entre outros. Mais tarde, se o desejarmos, esta ontologia poderá ser importada por um sistema de bases de dados (ex: Neo4J ou GraphDB). Para além disso, pode ser construído um website sobre filmes, no qual a persistência de dados poderá ser atingida utilizando um dos sistemas de bases de dados referidos.

2.3 O PROCESSO DE DESENVOLVIMENTO DE UMA ONTOLOGIA – METODOLOGIAS

Existem várias metodologias que podem ser utilizadas na construção de ontologias. Todas elas compreendem um conjunto de princípios, práticas, métodos e atividades que são usados para projetar, construir, avaliar e fazer o *deploy* das ontologias. Porém, não existe uma metodologia ideal, já que não existe uma maneira correta de modelar um dado domínio de conhecimento. Além disso, o desenvolvimento de ontologias é necessariamente um processo iterativo (Gasevic et al., 2006).

2.3.1 *A Metodologia de Noy e McGuiness*

A primeira metodologia que iremos apresentar é a sugerida por Noy e McGuiness (2001). Esta metodologia impõe a realização de um conjunto específico de tarefas, nomeadamente:

1. **Determinar o domínio e o propósito da ontologia**, isto é, responder a várias questões básicas, designadamente:
   - Qual é o domínio que a ontologia vai cobrir?
   - Para que é que se vai utilizar a ontologia?
   - Para que tipo de questões é que a ontologia deve ter respostas?
   - Quem vai usar e manter a ontologia?
2. **Considerar a reutilização de ontologias existentes**, o que significa que se deve partir das ontologias já existentes e verificar se podemos refinar e alargar as fontes existentes,

em particular para o nosso domínio e tarefa, ou reutilizar ontologias existentes. Este último caso pode ser um requisito a cumprir, caso o nosso sistema precise de interagir com outras aplicações que já se comprometeram com ontologias específicas ou vocabulários controlados. Além disso, muitas ontologias já se encontram disponíveis em formato eletrónico e podem ser importadas para um ambiente de desenvolvimento de ontologia que podemos estar a usar. Mesmo que um sistema de representação de conhecimento não possa trabalhar com um formalismo particular, a tarefa de traduzir uma ontologia de um formalismo para outro geralmente não é difícil. Por fim, existem várias bibliotecas reutilizáveis na Web e na literature, um dos exemplos é o Ontolingua (Farquhar, 1997).

3. **Enumerar termos importantes na ontologia**, no momento em que se começa a construção da terminologia. É útil escrever uma lista de todos os termos acerca dos quais gostaríamos de fazer declarações ou explicar a um utilizador. Neste ponto, importa saber:
   - Quais são os termos das quais gostaríamos de falar?
   - Que prioridades têm esses termos?
   - O que gostaríamos de dizer sobre esses termos?

4. Por exemplo, termos importantes relacionados com o cinema incluem filmes, atores, género de filme, país de origem, duração do filme, personagens do filme, data de lançamento, entre outros.

5. **Definir as classes e a hierarquia das mesmas**. Existem várias possibilidades para a definição das classes e a sua hierarquia, nomeadamente:
   - O método **top-down** começa por definir os conceitos mais gerais no respetivo domínio e só depois passa para a especialização desses conceitos. Por exemplo, na ontologia "Cinema", apresentada anteriormente, podemos começar por definir a classe "Filme", especializando depois esta classe e criando subclasses como "FilmeAventura", "FilmeInfantil" ou "LongaMetragem". Posteriormente, podemos ainda categorizar, por exemplo, a classe "FilmeInfantil" em "Disney" e "DreamWorks", e assim por adiante;
   - Podemos ainda combinar os dois métodos anteriores, criando um novo método designado por middle-out em que, seguindo o exemplo anterior, podemos numa primeira fase criar as classes "Disney" e "Dreamworks" e agrupá-las numa superclasse Filme. Posteriormente criamos uma subclasse de "Filme" designada como "FilmeInfantil", que se tornará superclasse das classes "Disney" e "DreamWorks".

   É importante salientar que nenhum dos métodos é melhor que outro. A sua escolha depende muito da visão pessoal do domínio e da preferência de cada um.

6. **Definir as propriedades das classes**. Após a definição das classes é necessário descrever a estrutura interna dos conceitos, uma vez que as classes, por si só, não são suficientes para responder às questões enumeradas no passo 1. Esta descrição é feita através da criação de propriedades para cada classe. Isto é, escolhendo a classe "Ator" como exemplo, este pode ter nome, idade, altura, morada, nacionalidade, entre outros. Este processo é feito para cada classe existente, sendo importante destacar que as subclasses herdam as propriedades da sua superclasse. Ou seja, tendo a classe "Filme" a propriedade data de lançamento, todas as suas subclasses terão também essa propriedade.

7. **Definir as características dos slots**. Podemos definir estas características como sendo o tipo de valor do slot, ou seja, os valores permitidos (domínio e intervalo), o número de valores (cardinalidade) e outras características dos valores que o *slot* pode assumir.

8. **Criar instâncias**. A última etapa consiste na criação de instâncias de cada classe da hierarquia. Definir instâncias de cada classe requer a escolha de uma classe; a criação de uma instância dessa classe e, por fim, o preenchimento dos valores dos slots. Seguindo o exemplo da ontologia apresentada inicialmente, podemos criar uma instância 'Matrix' que representa um "Filme" específico.

2.3.2 *A Metonthology*

Na prática, o método apresentado por Noy e McGuinness (2001) é complicado, uma vez que requer a consideração de muitas questões que geram conflitos e implicam uma série de detalhes minuciosos. Um exemplo de uma metodologia mais compreensiva é a Metonthology, que foi desenvolvida por Férnandez et al. (1997). O ponto de partida da *Metonthology* é que a engenharia ontológica requer a definição e padronização de todo o ciclo de vida da ontologia – desde a especificação de requisitos até à manutenção – bem como metodologias e técnicas que conduzem o desenvolvimento da ontologia ao longo do ciclo de vida. A *Methontology* considera a identificação do processo de desenvolvimento, um ciclo de vida baseado em protótipos em evolução, a própria metodologia, que especifica as etapas a realizar em cada etapa, as técnicas usadas, os produtos de cada atividade e um procedimento de avaliação da ontologia. Quanto ao processo de desenvolvimento de ontologias em *Metonthology*, este compreende as seguintes fases de trabalho (Gasevic et al., 2006):

- **Especificação**, na qual se faz a identificação da terminologia da ontologia, objetivo principal, finalidade e nível de granularidade;

- **Conceptualização**, em que se organiza e se estrutura de forma semiformal o conhecimento adquirido durante a fase da especificação, usando um conjunto de represen-

tações intermédias que tanto especialistas de domínio quanto ontologistas podem entender;

- **Implementação**, que, usando um ambiente de desenvolvimento de ontologias para representar formalmente e implementar o resultado das duas fases anteriores, nomeadamente conceitos, hierarquias, relações e modelos.

Juntamente com os processos que ocorrem nas três etapas descritas, a Metonthology cobre processos que funcionam em paralelo ao longo do ciclo de vida da ontologia, em particular: garantia de qualidade, integração, avaliação, manutenção, documentação e gestão de configuração. Também identifica as interdependências entre o ciclo de vida da ontologia que está a ser desenvolvida e os ciclos de vida de outras ontologias relacionadas.

Além disso, a Methontology especifica em detalhe as técnicas utilizadas em cada atividade, o resultado de cada atividade e como devem ser avaliados. *Methontologies* também reconhecem a importância da aquisição de conhecimento, sendo esta um processo de trabalho com especialistas no domínio e as suas atividades estão interligadas com as atividades das fases de especificação e conceptualização. Na Figura 3 é possível observar a representação dos estados e atividades que envolvem o processo de *Metonthology*, bem como a ligação entre estes. É importante salientar que as Methontologies são adequadas para construir ontologias a partir do zero ou através da reutilização de ontologias já existentes (Gasevic et al., 2006).

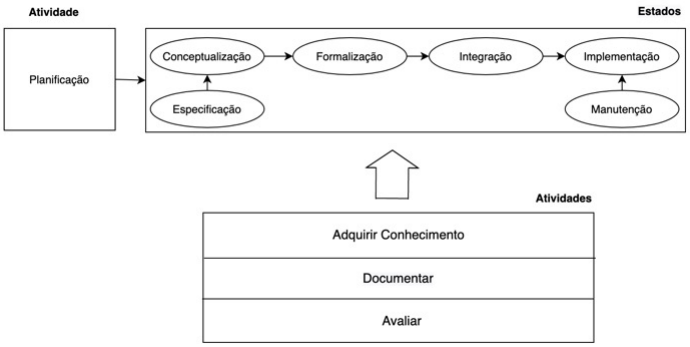


Figura 3: Estados e atividades – figura adaptada de Baccigalupo e Plaza (2007).

2.3.3 *A Metodologia de Uschold e King*

O processo de construção proposto por Uschold e King (1995) é composto por quatro etapas distintas:

- ***Identificação do propósito***. É importante esclarecer o motivo da construção da ontologia e que uso se pretende dar-lhe. Uma ontologia pode ser projetada com a intenção de compartilhar conhecimento, reutilizar conhecimento ou como parte de uma

base de conhecimento existente (Breitman et al., 2007). Também será útil identificar e caracterizar os pressupostos utilizadores.

- **Construção**. Nesta segunda etapa desenvolvem-se as seguintes tarefas:
  - Capturar, para fazer a identificação dos conceitos chave e das relações no domínio de interesse e das produções precisas e inequívocas de definições de texto para tais conceitos e relações. Além disso, dever-se-á também fazer a identificação de termos para se referir a tais conceitos e relações.
  - Codificar, para fazer a representação explícita da conceptualização capturada, no passo anterior, numa linguagem formal.
  - Integrar, para questionar a possibilidade de reutilização de outras ontologias existentes, podendo este processo ser feito em paralelo com os anteriores.
- **Avaliação**. Usar critérios técnicos para verificar a conceptualização, usando questões de competência e validações do mundo real (Breitman et al., 2007).
- **Documentação**. Descrição do processo de construção da ontologia. O formato final poderá variar de acordo com o tipo de ontologia em questão. Os utilizadores podem definir as suas próprias convenções, como representar os nomes das classes por letras maiúsculas e relações em itálico.

Este método tem sido criticado por oferecer pouco suporte na identificação das classes e relações das ontologias. Uma representação intermédia deve ser proposta, juntamente com heurísticas para ajudar os utilizadores a decidir quais os conceitos que devem ser incluídos na ontologia e como devem ser classificados.

2.4 Ferramentas para Desenvolvimento de Ontologias

Atualmente, acha-se disponível um leque diversificado de ferramentas com capacidade para construir e editar ontologias, que podem ser bastante úteis no processo de desenvolvimento de uma ontologia desde o início ou na reutilização de ontologias previamente construídas. Usualmente, estas ferramentas disponibilizam serviços de edição, navegação, documentação, visualização, exportação e importação de ontologias. Vejamos algumas dessas ferramentas.

Protégé

O Protégé é um editor de ontologias e de bases de conhecimento produzido pela Universidade de Stanford que permite a construção de ontologias de domínio, definição de classes, hierarquias de classes, variáveis, restrições de valor de variável e as relações entre classes e as propriedades dessas mesmas relações. É uma ferramenta *open-source* que pode ser

encontrada em (Musen, 2015) e que tem sido utilizada por especialistas em vários domínios, como a medicina, e para a construção de sistemas de base de conhecimento. O Protégé disponibiliza serviços de visualização como, por exemplo, o OntoViz, que ajudam o utilizador a visualizar ontologias com o auxílio de diagramas (Bhaskar and Savita, 2010). As ontologias em Protégé podem ser exportadas para uma variedade de formatos, incluindo RDF(S), OWL e XML(Schema). O Protégé pode ainda ser usado para aceder a motores de raciocínio; para editar e usar queries e regras; para comparar versões de ontologias e para visualizar relações entre conceitos. É uma ferramenta que é instalada localmente no computador e não permite a edição colaborativa de ontologias por grupos de utilizadores (Escórcio and Cardozo, 2007).

OntoEdit

O OntoEdit foi desenvolvido pelo Grupo de Gestão de Conhecimento do instituto AIFB, na Universidade de Karlsruhe. Tipicamente, é um ambiente de engenharia de ontologias que suporta o desenvolvimento colaborativo de ontologias (Sure et al., 2009). Os resultados são arquivados através de uma arquitetura cliente/servidor, na qual as ontologias são geridas num servidor central e vários clientes podem aceder e modificar essas ontologias. O OntoEdit foi desenvolvido tendo em mente dois objetivos principais. Por um lado, o editor foi projetado para ser, tanto quanto possível, independente e neutro quanto a uma linguagem de representação concreta. Por outro lado, foi planeado para fornecer uma poderosa interface gráfica do utilizador para representar hierarquias de conceitos, relações, domínios, intervalos, instâncias e axiomas. Esta ferramenta suporta *F-Logic, RDF Schema* e *OIL*, e é multilingue, podendo cada conceito ou nome de relação ser especificados em várias línguas (Escórcio and Cardozo, 2007).

Apollo

O Apollo é uma aplicação *user-friendly* de modelação de conhecimento. A modelação é baseada em primitivas básicas como classes, instâncias, funções, relações, etc. O sistema de classes de Apollo é modelado de acordo com o OKBC (*Open Knowledge Base Connectivity*) e a base de conhecimento acolhe ontologias que são organizadas hierarquicamente. A ontologia pode herdar outras ontologias e, de seguida, usar classes de ontologias herdadas como próprias. Cada ontologia herda pelo menos uma ontologia – uma ontologia padrão que contém todas as classes primitivas: inteiro, boolean, float, string, lista, etc (Bhaskar and Savita, 2010).

Swoop

Swoop é um editor e browser de ontologias OWL baseado na Web. Contém validação OWL e oferece várias visualizações de sintaxe de apresentação OWL. Tem suporte de raciocínio e

fornece um ambiente de múltiplas ontologias, em que estas podem ser comparadas, editadas e fundidas. Diferentes ontologias podem ser comparadas através das suas definições de descrição baseadas na lógica, propriedades associadas e instâncias. A interface do Swoop possui hiperligações para que a navegação seja fácil e simples. Esta ferramenta não segue nenhuma metodologia para a construção de ontologias. Os utilizadores podem reutilizar dados ontológicos externos, através da ligação à entidade exterior ou através da importação total da ontologia externa, porém não possibilita importações parciais de OWL (Escórcio and Cardozo, 2007).

2.5 Evolução de uma Ontologia

São vários os autores que apresentam uma definição de evolução de uma ontologia. Porém, todas elas assentam numa mesma ideia principal: uma pequena mudança numa ontologia pode alterar ou corromper outras partes da própria ontologia, outras ontologias que são dependentes desta ou de aplicações que a usem (Öhgren, 2004). Nesta dissertação iremos adotar a definição usada em Stojanovic (2004) e Stojanovic, Stojanovic e Handschuh (2002)como sendo a adaptação oportuna de uma ontologia às mudanças surgidas e à propagação ou manutenção consistente dessas mudanças para serviços dependentes.

As ontologias são dinâmicas e devem ser capazes de evoluir ao longo do tempo, por várias razões: por uma mudança de domínio (novos conceitos, novas regras de negócio, etc.), pela mudança da conceptualização partilhada ou dos requisitos do utilizador (Öhgren, 2004). Portanto, se uma ontologia pretende ser útil, é essencial que seja capaz de se adaptar às mudanças que, inevitavelmente, ocorrerão ao longo do tempo no seu domínio de aplicação (Stojanovic, 2004).

Para além disso, o número de ontologias em uso e os custos associados à sua adaptação é cada vez maior. O desenvolvimento de ontologias é caro, mas o processo de evolução ainda o é mais (Stojanovic, 2004).

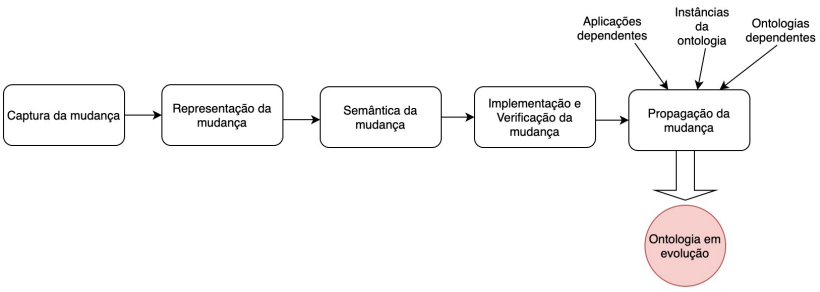


Figura 4: Fases do processo de evolução de uma ontologia

O processo de evolução de uma ontologia (representado na Figura 4) pode ser organizado em 5 fases distintas, nomeadamente:

- ***Captura da mudança*** (Merelli and Luck, 2004) - em que mudanças a serem aplicadas na ontologia são identificadas. Três tipos de mudanças podem ser identificadas, baseadas:
  - no uso, isto é, derivada do comportamento do utilizador;
  - nos dados, provocada pelas mudanças nas instâncias da ontologia;
  - na estrutura, em que as mudanças são derivadas da análise da estrutura da ontologia.
- **Representação da mudança** (Khattak et al., 2013) – em que todas as mudanças necessárias são representadas usando um formato de representação formal.
- **Semântica da mudança** (Khattak et al., 2013) - em que os efeitos das mudanças necessárias são testados na ontologia quanto à sua consistência e, se necessário, algumas mudanças deduzidas também são incluídas na solicitação de mudança, para evitar conflitos. Este processo está relacionado principalmente com a área de *debugging* de ontologias.
- **Implementação e verificação da mudança** (Merelli and Luck, 2004) - em que o pedido de mudança completo é executado na ontologia, verificando-se se as alterações realizadas levaram a um resultado válido (ou desejável) e permitindo que o utilizador desfaça tais alterações, se o resultado for inválido.
- **Propagação da mudança** (Stojanovic et al., 2002) - potencialmente, uma mudança de ontologia pode corromper as instâncias, ontologias dependentes, bem como programas em execução que possam estar a usar a ontologia. A tarefa da fase de propagação da mudança é trazer automaticamente todos os elementos dependentes para um estado consistente após ter sido realizada uma atualização da ontologia.

3

EXTRAÇÃO AUTOMÁTICA DE ONTOLOGIAS

3.1 O PROCESSO DE *ontology learning*

A designação *Ontology learning* refere-se ao suporte automático ou semiautomático para a construção de ontologias e tem como objetivo a descoberta de conhecimento a partir de diferentes fontes de dados e a sua posterior representação numa estrutura ontológica. Técnicas de diversas áreas, como processamento de linguagem natural, *machine learning* ou recuperação de dados, têm sido fundamentais no desenvolvimento de sistemas de *ontology learning* (Belhoucine and Mourchid, 2020). Este processo requer o fornecimento de dados como input a partir dos quais são aprendidos os conceitos relevantes para qualquer domínio e suas definições, bem como as relações entre estes (Belhoucine and Mourchid, 2020). Estes dados podem ser divididos em três tipos:

- **Dados estruturados** – que seguem uma estrutura de um esquema definido, como esquemas de base de dados, ontologias existentes e bases de conhecimento.
- **Dados semiestruturados** – que incluem dados estruturados mistos com texto livre, por exemplo, dicionários, como o WordNet, documentos html ou documentos xml.
- **Dados não estruturados** - que incluem textos em linguagem natural, como documentos Word, PDF ou páginas web.

Usualmente, a ontology learning é usada sobre textos não estruturados, que são o tipo de *input* mais acessível para o processo de ontology learning. No entanto, estes textos também são considerados os mais difíceis, porque a maior parte do conhecimento que pode estar contido neles está implícito e permite diferentes interpretações por diferentes pessoas, mesmo quando se usam as mesmas palavras (Belhoucine and Mourchid, 2020).

No capítulo anterior, uma ontologia foi definida como sendo um meio formal e estrutural de representar conceitos e relações entre estes num determinado domínio (Gruber, 1993). Antes de estabelecer as tarefas no processo de ontology learning, deve-se definir as etapas para o desenvolvimento da ontologia, quer este seja manual ou automático. Embora não

exista um padrão no que diz respeito a este processo de desenvolvimento, Cimiano (2016) descreve as tarefas envolvidas no processo de ontology learning como um conjunto de camadas, usualmente designado por "*ontology learning layer cake*" (Figura 5), que estão ordenadas ascendentemente, por termos, sinónimos, conceitos, taxonomias, relações e, por fim, axiomas e regras (Belhoucine and Mourchid, 2020).

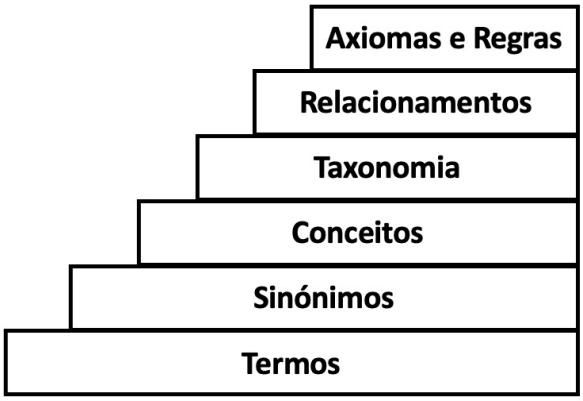


Figura 5: *Ontology learning cake* - figura adaptada de Cimiano et al. (2016).

As diferentes camadas do Ontology learning cake são as seguintes:

- **Termos**, que são o bloco de construção mais básico. Podem ser simples (uma única palavra) ou complexos (várias palavras) e são considerados realizações linguísticas de conceitos específicos de domínio.
- **Sinónimos**, que integram as variantes de termos semânticos no mesmo idioma e entre os idiomas.
- **Conceitos**, que podem ser concretos ou abstratos, reais ou fictícios. Estes devem incluir:
  - Intenção, a definição formal do conjunto de objetos que este conceito descreve.
  - Extensão, um conjunto de objetos que a definição deste conceito descreve.
  - Realizações lexicais, um conjunto de realizações linguísticas, termos (multilingues) para este conceito.
- **Relações**, que são propriedades cujos valores são referência para outros objetos da ontologia. O tipo mais comum é a relação "is-a", uma relação de inclusão, que define que objetos fazem parte de cada classe de objetos.
- **Axiomas**, que são um conjunto de regras que servem para modelar restrições inerentes às instâncias derivadas das classes.

Assim, com a identificação das diferentes etapas de *ontology learning*, torna-se claro quais os passos a seguir de modo a obter a ontologia pretendida.

3.2 TÉCNICAS DE *ontology learning*

São muitos os estudos feitos na área de *Ontology Learning*, realizados por diversos autores. No entanto, as técnicas que irão ser apresentadas neste capítulo são baseadas, essencialmente, nos trabalhos de Asim et al. (2018), por estes estarem mais atualizados. Como foi dito anteriormente, algumas das técnicas utlizadas em áreas como o processamento de linguagem natural, machine learning, ou a recuperação de dados têm sido fundamentais no desenvolvimento de sistemas de ontology learning (Belhoucine and Mourchid, 2020). Num processo de *ontology learning*, em primeiro lugar, tipicamente existe um pré-processamento do texto, usando técnicas linguísticas como POS (*part of speech tagging*), *parsing* ou lematização. Após este pré-processamento, faz-se a extração de termos e conceitos relevantes do domínio, usando várias técnicas de processamento de linguagem natural, nomeadamente *syntatic parsing*, *subcategorization frames* ou *seed words extraction*. Adicionalmente podem ser usadas técnicas de domínio estatístico, tais como a C/NC value, a *contrastive analysis*, a *latent semantic analysis* (LSA) ou o *clustering*.

Para além da obtenção dos clusters de conceitos, é necessário, também, extrair as relações taxonómicas e não-taxonómicas que possam ser estabelecidas entre estes, usando, mais uma vez, técnicas de processamento de linguagem natural e técnicas de domínio estatístico. Estas irão ser explicadas nas secções 3.2.2 e 3.2.3. Por fim, são formados os axiomas usando ILP (*inductive logic programming*). De modo a avaliar a integridade da ontologia desenvolvida, são utilizadas algumas técnicas próprias para a avaliação de ontologias.

3.2.1 *Pré-Processamento*

Em *ontology learning*, após a análise do texto, o pré-processamento corresponde à etapa inicial deste processo, existindo para isso várias técnicas passíveis de serem aplicadas, tais como:

- **POS** (***part of speech tagging***), que é o processo de marcação de uma palavra num texto como correspondendo a uma classe gramatical específica, com base na sua definição e no seu contexto, ou seja, na sua relação com palavras adjacentes e relacionadas numa frase ou parágrafo.
- ***Parsing***, que é um tipo de análise sintática que encontra várias dependências entre as palavras numa frase e as representa na forma de uma estrutura de dados designada por *parsing tree*.

- ***Lematização***, que é outra técnica linguística que tem como objetivo reduzir uma palavra à sua forma base e agrupar diferentes formas da mesma palavra. Por exemplo o lema de "saltar" e "salto" é "saltar".

A importância desta etapa pode ser observada em Jiang e Tan (2013), concluindo-se que, para obter uma maior precisão na tarefa de *ontology learning*, é necessário um pré-processamento eficiente de dados, utilizando boas técnicas linguísticas (Asim et al., 2018).

3.2.2 *Extração de Termos/Conceitos*

Técnicas Linguísticas

Em *ontology learning* são também usadas técnicas linguísticas na extração de termos, conceitos e relações. Para se extraírem termos e conceitos utilizando estruturas sintáticas é necessário fazer a marcação do texto com classes gramaticais (POS). Essas estruturas são utilizadas para encontrar termos, analisando as palavras e os modificadores presentes. Por exemplo, *em ontology learning*, a estrutura sintática da frase nominal pode ser usada para extrair potenciais termos candidatos do texto (Asim et al., 2018). Outra técnica que pode ser utilizada neste processo designa-se por *Subcategorization frame*. A *subcategorization frame* de uma palavra é o número de palavras de uma determinada categoria que ela seleciona ao aparecer numa frase. Por exemplo, na frase "A Ana atirou o dado", o verbo "atirou" escolhe "Ana" e "dado" como palavras vizinhas, por isso *subcategorization frame* de "atirou" consiste nessas duas palavras. Em outras palavras, uma restrição de seleção agora é feita para o verbo 'atirar', em que este irá selecionar as suas palavras vizinhas das classes de 'Pessoa' e 'Objeto'. Quando usado em conjunto com técnicas de clustering, essa restrição é capaz de descobrir conceitos (Asim et al., 2018). Além das técnicas referidas existe também a *Seed words*, que é uma metodologia utilizada no processo de *ontology learning*. Basicamente, nesta técnica identificam-se palavras específicas do domínio que fornecem uma base para outros algoritmos para extrair termos e conceitos específicos de domínios semelhantes, garantindo que apenas os termos mais relevantes e semanticamente mais próximos das *seed words* sejam extraídos.

Técnicas Estatísticas

Para além de técnicas linguísticas, é também possível usar técnicas estatísticas na extração de termos e conceitos. Nesta categoria podemos encontrar as seguintes técnicas:

- ***Contrastive analysis***. Após a realização da extração de termos verifica-se que existem palavras que não são relevantes para o domínio do texto. Esta técnica permite fazer a filtragem desses termos. No domínio de *ontology learning*, Navigli et al. (2003) introduziram duas novas medidas para este método, nomeadamente *relevância de domínio* e *consenso de domínio*. Estes investigadores utilizaram dois tipos de *corpora*: o *corpus* relevante (domínio alvo) e o *corpus* não relevante (domínio contrastivo). A filtragem garante que permaneçam os termos que são mais relevantes para o domínio de destino. A relevância do domínio é usada para medir a especificidade de um termo em relação ao domínio de destino, atribuindo pontuações com base na sua relevância no domínio de destino e na sua irrelevância nos domínios contrastivos (Asim et al., 2018). A relevância dos termos para o domínio é calculada usando uma série de fórmulas que envolvem combinação linear e probabilidades, fórmulas essas que são explicadas detalhadamente em Asim et al. (2018).

- ***Co-occurrence analysis***: técnica de extração de conceitos que localiza as unidades lexicais que ocorrem juntas com o objetivo de encontrar as associações implícitas entre vários termos e conceitos, bem como extrair termos relacionados.

- **LSA**: algoritmo matemático baseado na ideia de que os termos que ocorrem juntos terão significados próximos. O primeiro passo neste algoritmo é interpretar o texto que é recebido numa matriz em que cada linha representa uma única palavra e cada coluna representa uma parte do texto, como, por exemplo, uma frase. De seguida, as entradas de cada célula são submetidas a uma transformação, na qual cada frequência da célula é ponderada por uma função que expressa a importância da palavra naquela passagem de texto e pelo grau em que o tipo de palavra carrega informação no domínio do discurso em geral. O segundo passo consiste na aplicação da decomposição de valores singulares (SVD) na matriz, de modo a reduzir o número de linhas, preservando a estrutura de similaridade entre as colunas. Os documentos são então comparados a partir do cosseno do ângulo entre os dois vetores, formados por quaisquer duas colunas, em que valores próximos de 1 indicam semelhança, enquanto valores próximos de 0 indicam que os documentos são muito diferentes (Landauer et al., 1998).

- ***Clustering***: é um algoritmo não supervisionado de *machine learning* que permite o agrupamento de dados consoante o seu grau de semelhança. Existem vários

trabalhos em extração de ontologias que utilizam este algoritmo. De referir os de Faure e Nedellec (1998) e Cimiano, Hotho e Staab (2016).

3.2.3 *Extração de relações*

Técnicas Linguísticas

A análise de dependências ajuda a descobrir relações entre termos, usando informações de dependências presentes nas árvores de parsing. O **padrão léxico-sintático** é uma abordagem baseada nas relações semânticas estabelecidas entre as palavras e desempenha um papel fundamental nas fases de extração de relações taxonómicas e não taxonómicas de *ontology learning*. Por exemplo "NP como NP, NP, ... NP", é uma regra que vai extrair padrões do tipo "animais como cão, gato, cavalo e tigre". Estes tipos de regras são bastante úteis para extrair relações do tipo "is-a " - is-a (gato, animal), que representam uma relação de hiperonímia. Por outro lado, padrões léxico-sintáticos como "NP é parte de NP" podem ser usados para extrair relações não taxonómicas.

Dependency parsing. Esta técnica permite realizar um outro tipo de abordagem, que, basicamente, consiste na análise de dependências entre palavras, em linguagem natural. Uma relação de dependência é uma relação binária assimétrica estabelecida entre uma palavra designada por "cabeça" e uma palavra designada por "modificador" ou "dependente" (Hazman et al., 2008). Quando se fala em dependências entre palavras, quer-se referir as dependências sintáticas. Utilizando como exemplo a frase "A Ana comeu um bolo" (Figura 6):

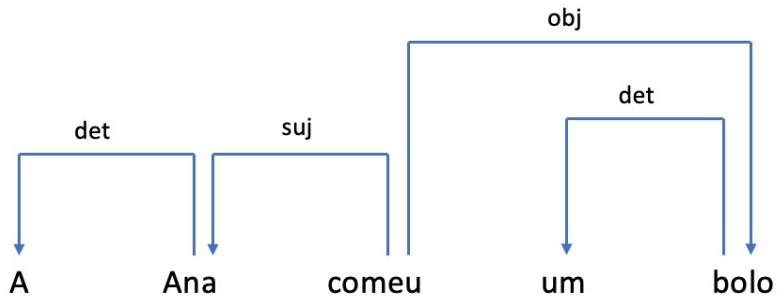


Figura 6: Representação das dependências sintáticas na frase "A Ana comeu um bolo".

Através da Figura 6, podemos ver que, na frase apresentada, "Ana" é considerado o sujeito ligado ao verbo "comer" e "um" é considerado determinante ligado à palavra "bolo".

Padrões léxico-sintáticos

Segundo Gruber (1993), uma relação é um conjunto de n-uplas que representam um relacionamento entre objetos no universo do discurso. Cada n-upla é uma sequência finita e ordenada de objetos. Assim sendo, a n-upla pode ser representada pela expressão (nome da relação, arg1, arg2, ... argn), em que arg1 é um objeto na n-upla. Um exemplo de representação de uma relação binária pode ser (arg1, nome da relação, arg2), podendo haver alterações na ordem dos elementos da n-upla (Machado and Strube de Lima, 2015). Na Tabela 1 apresentam-se alguns tipos de relações semânticas.

Tabela 1: Exemplos de relações semânticas

Tipo de relação	**Arg1**	**Arg2**
Sinonímia	rápido	veloz
Antonímia	alto	baixo
Hiperonímia	animal	cavalo
Hiponímia	cavalo	animal
Holonímia	livro	páginas
Meronímia	páginas	livro

Entre as relações acima apresentadas, as relações de hierarquia representadas por hiperonímia e hiponímia são as que irão ser abordadas nesta dissertação. A hiperonímia expressa uma relação de significado geral, capaz de abranger vários hipónimos, enquanto a hiponímia é o oposto, pois o seu significado é hierarquicamente mais específico e remete para um ou mais hiperónimos. Existem vários estudos realizados nesta área, nomeadamente o trabalho desenvolvido por (Hearst, 1992), que propõe um método de extração de relações hiponímicas entre sintagmas nominais para a língua inglesa, com base em seis padrões que podem ser encontrados com grande frequência em textos, e que estão representados na tabela seguinte:

Tabela 2: Padrões de Hearst (1992)

1.	NP such as {NP1, NP2 . . . , (and | or)} NPn
2.	such NP as {NP , }* {(or | and)} NP
3.	NP { , NP}* { , } or other NP
4.	NP { , NP}* { , } and other NP
5.	NP { , } including {NP , }* { or | and } NP
6.	NP { , } especially {NP , }* { or | and } NP

Conforme se pode observar na tabela, NP representa um sintagma nominal. Usando um exemplo específico da autora, "... works by such authors as Herrick, Goldsmith, and Shakespeare", e aplicando o padrão (2) "such NP as {NP, }* {(and \| or)} NP", é possível extrair as seguintes relações:

- Hiponímia ("Herrick", "author")
- Hiponímia ("Goldsmith", "author")
- Hiponímia ("Shakespeare", "author")

A autora aplicou estes padrões em corpora enciclopédicos e jornalísticos, obtendo 63% de relações de boa qualidade. É fundamental realçar a importância do trabalho realizado por Hearst, pois este foi um dos pioneiros no que toca à utilização de padrões lexicais na extração de relações semânticas.

No que toca à língua portuguesa, autores como Freitas e Quental (2007), Taba e De Medeiros Caseli (2014) e Baségio (2007) realizaram também trabalhos nesta área, baseando-se no trabalho desenvolvido por Hearst, sendo o último direcionado para a língua portuguesa do Brasil.

O trabalho desenvolvido por Taba De Medeiros Caseli (2014) teve como fonte de dados dois corpora: o CETENFolha, um corpus jornalístico, e outro de uma revista de divulgação científica, composto por 646 artigos de Pesquisa FAPESP. Ambos os documentos foram morfologicamente anotados pelo *parser* PALAVRAS. Neste estudo as autoras pretendiam investigar a extração automática de relações semânticas do tipo (*is-a, part-of, location-of, effect-of, made-of e used-of*) a partir de textos em língua portuguesa, através do uso de duas técnicas distintas: padrões textuais e algoritmos de *machine learning*, nomeadamente árvores de decisão C4.5 e máquinas de vetores de suporte (Machado and Strube de Lima, 2015). Relativamente às relações do tipo is-a, essas autoras basearam-se nos padrões de Hearst (1992) e de Taba e De

Medeiros Caseli (2014), introduzindo ainda novos padrões manualmente definidos, que podem ser observados na tabela seguinte:

Tabela 3: Padrões para a relação *is-a* (Taba and De Medeiros Caseli, 2014)

1.	T1 (tais como | como) T2 {, T3}* (e | ou) TN
2.	T2 {, T3}* ,? (e | ou) outros T1
3.	tipos de T1: T2 {, T3}* (e | ou) TN
4.	T1 chamad(o | a | os | as) de? T2
5.	T2 {, T3}* ,? (e | ou) (qualquer | quaisquer) outro{s}? T1
6.	T2 é (o | a | um | uma) T1
7.	T2 sãoT1

Na tabela, T1 representa o primeiro termo na relação e T2 o segundo. Os termos T3, T4, ... TN, se presentes, são sempre relacionados hierarquicamente com T1. Após vários ensaios, Taba e De Medeiros Caseli (2014) obtiveram resultados e fizeram comparações com outros trabalhos, no que toca a este tipo de relação (*is-a*). Enquanto Hearst (1992) obteve uma precisão de 63% e Freitas e Quental (2007) conseguiram uma precisão de 73.4%, o primeiro ensaio, que consistiu unicamente na aplicação de padrões textuais, resultou numa precisão média de 61%. Após a aplicação das árvores de decisão e dos classificadores SVM, obtiveram-se precisões de 76.9% e 78.2%, respetivamente. Estes últimos resultados, apesar de não ser possível compará-los com resultados obtidos por outros autores, devido às diferenças nos corpora e métodos utilizados, revelam um bom nível de eficiência.

Técnicas estatísticas

Também podem ser utilizadas técnicas estatísticas para extrair relações taxonómicas e não taxonómicas do texto. Para a indução de hierarquias taxonómicas podem ser utilizadas técnicas de ***term subsumption*** e de ***clustering***. Além disso, podemos utilizar ARM (***association rule method***) para fazer a extração de relações não taxonómicas (Asim et al., 2018). Vejamos com um pouco mais de detalhe cada uma dessas técnicas:

- ***Term subsumption*** é uma técnica que permite encontrar relações hierárquicas entre os termos usados usando a probabilidade condicional desses termos em documentos subjacentes. Este algoritmo afirma que o termo t é mais geral do que o termo x se $P(t \mid x) > P(x \mid t)$, em que:

$$P(t|x) = \frac{\#\ documentos\ com\ os\ termos\ t\ e\ x}{\#\ documentos\ com\ o\ termo\ x},$$

$$P(t|x) = \frac{\#\ documentos\ com\ os\ termos\ t\ e\ x}{\#\ documentos\ com\ o\ termo\ t}$$

As equações apresentadas referem que, se o termo x ocorrer nos documentos, que são um subconjunto dos documentos que contêm o termo t, então t é mais geral do que x (Asim et al., 2018).

- ***Clustering hierárquico***, que é utilizado maioritariamente para encontrar relações taxonómicas entre os dados, através da aplicação de medidas de similaridade para agrupar os termos e construir hierarquias. Uma avaliação dos métodos de clustering para ontology learning a partir de textos não estruturados pode ser encontrado em Drymonas et al. (2010).
- **ARM** (***association rule method***), que é uma técnica bastante popular de data mining que permite mostrar a correlação entre conjuntos de itens numa série de dados ou transações. As regras de associação são um tipo de regra "IF antecedente THEN consequente" que garante, com uma certa probabilidade (limite de confiança), que sempre que o antecedente acontece, o consequente virá (Ferraz and Garcia, 2013). Em *ontology learning* esta técnica é usada, maioritariamente, para fazer a extração de relações não taxonómicas.

3.3 SISTEMAS DE ONTOLOGY LEARNING

Ao longo dos últimos anos foram desenvolvidos vários sistemas de *ontology learning* que usam um ou mais dos algoritmos que foram descritos anteriormente, com o objetivo de reduzir o esforço humano necessário para o desenvolvimento de ontologias. De seguida são apresentados alguns desses sistemas:

- **HASTI** (Shamfard and Barforoush, 2002), que é um sistema que usa como input dados não estruturados em formato de linguagem natural, em persa. Este sistema não usa nenhum conhecimento prévio. Desenvolve as ontologias do "zero". Uma ontologia em HASTI é um pequeno kernel no início. O HASTI aprende conceitos, relações taxonómicas e não taxonómicas e axiomas, para construir ontologias sobre o *kernel* existente.

A abordagem de aprendizagem do HASTI é uma abordagem simbólica híbrida, uma combinação de métodos de análise linguística, lógica, baseada em modelos e semântica. Além disso, o sistema realiza *clustering*, online e offline, para organizar a ontologia que cria (Drumond and Girardi, 2008).

- **Text2Onto** (Cimiano and Völker, 2002), um sucessor do sistema Text-to-Onto, é um sistema que combina abordagens de machine learning com técnicas básicas de processamento de linguagens como a tokenização, a lematização ou a *shallow parsing*. O processamento linguístico em Text2Onto começa com a tokenização e a divisão da frase. O conjunto de anotações resultante serve como entrada para um POS *tagger* que atribui categorias sintáticas apropriadas a todos os tokens. Finalmente, a lematização é conduzida através de um analisador morfológico e de um lematizador, respetivamente. O processo de aprendizagem começa com base em *machine learning* e heurísticas linguísticas para identificar conceitos e relações.

  O Text2Onto inclui ainda várias medidas para avaliar a relevância de determinado termo em relação ao texto em questão. Além disso, utiliza textos não estruturados, semiestruturados, dicionários e bases de dados como input. O *output* do processo de extração é uma ontologia de domínio que contém conceitos específicos e conceitos independentes do domínio. Os conceitos independentes do domínio são removidos para um melhor ajuste do vocabulário da ontologia de domínio. Portanto, o resultado do processo é uma ontologia de domínio que contém apenas os conceitos de domínio aprendidos dos dados de *input* (Park et al., 2010).

- **OntoLearn** (Velardi et al., 2005), que é um sistema (semi-)automático de ontology learning a partir de textos. O sistema OntoLearn utiliza técnicas de mineração de texto e recursos linguísticos existentes, como o WordNet (base de dados lexical com relações semânticas entre as palavras) e o SemCor (conjunto de textos semanticamente anotados) para aprender, a partir de repositórios de documentos que estejam disponíveis ou de *websites* dedicados, conceitos de domínio e relações taxonómicas que estejam definidas entre eles (Liu et al., 2011).

3.4 AVALIAÇÃO DE ONTOLOGIAS

Existe uma grande variedade de abordagens para fazer a avaliação de ontologias. Dependendo do tipo de ontologia e do propósito da avaliação, essas abordagens podem ser agrupadas nas seguintes categorias:

- ***Gold Standard-based evaluation***, que compara a ontologia aprendida com uma ontologia padrão predefinida que represente um resultado idealizado do algoritmo de aprendizagem. No entanto, ter uma ontologia adequada pode ser um desafio, uma vez que esta deve ser criada em condições semelhantes, com objetivos semelhantes aos da ontologia aprendida (Belhoucine and Mourchid, 2020).
- ***Task-based evaluation***, que permite examinar a forma como os resultados da aplicação baseada em ontologia são afetados pelo uso da mesma. Por exemplo, no caso de uma ontologia projetada para melhorar o desempenho da recuperação de documentos, os utilizadores podem recolher algumas consultas de amostra e determinar se os documentos recuperados são mais relevantes quando a ontologia é usada (Belhoucine and Mourchid, 2020).
- ***Criteria-based evaluation***, que avalia em que medida uma ontologia obedece a certos critérios desejáveis. Podemos distinguir entre medidas relacionadas com a estrutura de uma ontologia e medidas mais sofisticadas (Belhoucine and Mourchid, 2020).
- ***Corpus-based evaluation***, que avalia o grau de aptidão da cobertura da ontologia para um determinado domínio. Este tipo de abordagem compara a ontologia aprendida com o conteúdo de um conjunto de textos que cobre significativamente o domínio correspondente. Técnicas de processamento de linguagem natural ou extração de informação são utilizadas para analisar o conteúdo do conjunto (Belhoucine and Mourchid, 2020).

Várias ferramentas para avaliação de ontologias foram desenvolvidas, diferindo consoante o contexto da avaliação. Em Belhoucine e Mourchid (2020) podemos encontrar alguns exemplos dessas ferramentas, nomeadamente:

- ***OntoQA***, que é uma ferramenta que mede a qualidade da ontologia do ponto de vista do consumidor, usando métricas de esquema e instância. Esta ferramenta recebe como entrada uma ontologia povoada e rastreada ou um conjunto de termos de pesquisa fornecidos pelo utilizador, classificando-os de acordo com as métricas relacionadas com vários aspetos de uma ontologia.

- ***OntoKhoj***, que é um mecanismo de pesquisa de ontologias. Este é baseado em algoritmos usados para pesquisa, agregação, ranking e classificação de ontologias na Web Semântica. Para além de permitir a recuperação de conhecimento fidedigno por parte de especialistas e engenheiros de ontologias, agiliza o processo de engenharia de ontologias através da reutilização extensiva de ontologias (Patel et al., 2003).

4

EXTRAÇÃO DE UMA ONTOLOGIA EM TEXTOS DE TESTAMENTOS

4.1 CONTEXTUALIZAÇÃO DO PROCESSO

Anteriormente, introduzimos o conceito de ontologia e apresentámos alguns estudos realizados na área de extração de ontologias, que incluíram referências aos algoritmos utilizados e aos métodos de avaliação de sistemas ontológicos desenvolvidos. De seguida, apresentaremos o trabalho que foi realizado na extração de uma ontologia a partir de um conjunto de textos de testamentos antigos editados por Barros e Alves (2019) – *O Livro dos Testamentos – Picote, 1780-1803* –, a partir de um manuscrito então inédito do século XVIII do Arquivo Municipal de Miranda do Douro. Neste trabalho definimos e implementámos, também, os mecanismos de exploração da ontologia obtida, de forma a podermos conhecer com detalhe os seus diversos elementos: classes, propriedades, tipos e entidades chave. Apesar do leque tão diverso de técnicas linguísticas e estatísticas apresentadas anteriormente, não foi possível a utilização de algumas delas, devido ao facto de os testamentos disponíveis estarem escritos em português setecentista, apresentando diferenças significativas relativamente ao português atual, sobretudo a nível ortográfico, evidenciando ampla variação fonética e fonológica, mas também a nível morfológico, lexical e sintático (Alves and Barros, 2019: 55-119). Tendo isso em consideração, serão apresentadas, numa primeira parte, as tecnologias e *frameworks* utilizadas neste processo de extração e de posterior exploração da ontologia, passando-se de seguida ao processo de extração aplicado, para o qual serão explicadas as técnicas utilizadas e todo o raciocínio subjacente. Com a ontologia desenvolvida, procedemos à sua exploração através de um sistema de bases de dados orientados por grafos, no qual foi possível visualizá-la de uma forma mais clara e pormenorizada. Por fim, analisaremos e discutiremos os resultados obtidos.

4.2 TECNOLOGIAS E *FRAMEWORKS* UTLIZADAS

Toda a parte de programação até à obtenção da ontologia foi feita utilizando a linguagem Python (Van Rossum and Drake, 2009), com o auxílio de uma biblioteca *open-source* denominada spacy (Honnibal and Montani, 2017), que é especialmente orientada para o processamento de linguagem natural (PLN). Esta biblioteca é muito poderosa e possui vários recursos para extração de informação ou para o processamento de linguagem natural, o que facilita bastante o trabalho de obtenção da ontologia.

Como referido anteriormente, os testamentos com que trabalhámos são textos não estruturados. Assim, para que fosse possível processar e retirar informação relevante dos mesmos foi necessário representar os dados num formato capaz de ser entendido por computadores, daí o papel fundamental do PLN. O PLN é uma subárea da Inteligência Artificial que tem como foco as interações entre computadores e humanos através das linguagens humanas. Consiste no processo de análise, compreensão e derivação de significado das linguagens humanas para os computadores (Singh, 2019). Em relação à persistência de dados e representação gráfica da ontologia resultante da extração, a ferramenta escolhida foi o Neo4J (Santos López and Santos De La Cruz, 2015), uma base de dados NoSQL orientada para grafos.

4.2.1 *Spacy*

A biblioteca Spacy (Honnibal and Montani, 2017) possui diferentes modelos, consoante a língua pretendida. Como, no nosso caso, trabalhamos com a língua portuguesa (séc. XVIII), o modelo escolhido foi o pt_core_news_sm, mesmo sabendo que este modelo foi preparado para o português contemporâneo. Porém, era o modelo que mais de adequava ao processo que queríamos implementar.

Após a importação do modelo, o primeiro passo que demos foi o de aplicar o comando nlp ao texto que se pretendia processar. Com a aplicação deste comando, o spacy faz a tokenização do texto a ser processado, de modo a obter um objeto Doc. De seguida, este objeto Doc é processado em vários passos. Estas etapas estão incluídas naquilo que designamos por pipeline de processamento. Alguns desses passos são realizados pelo *tagger*, que é o responsável por atribuir tags de classe gramatical, e o lemmatizer, que é o responsável por transformar as palavras na sua forma básica, ou o seu lema. Cada componente da pipeline retorna o Doc processado, que, depois, é passado para a componente subsequente. Na Figura 7

podemos observar uma ilustração desse processo. Posteriormente, alguns dos seus passos serão explicados de uma forma mais pormenorizada.

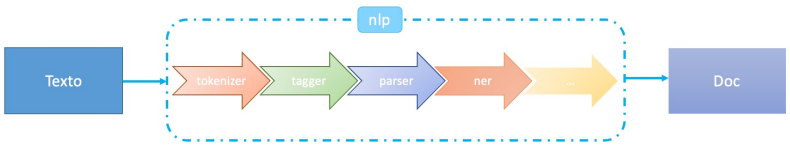


Figura 7: Pipeline de processamento – figura adaptada de Honnibal e Montani (2017).

É importante realçar que o resultado da aplicação de cada componente terá um *output* diferente, consoante a língua na qual o texto a ser processado se encontra escrito. Para além disso, estes modelos são treinados utilizando técnicas de *machine learning* e modelos estatísticos.

4.2.2 *Part-of-speech Tagging*

Após a tokenização é possível, através do *tagger*, atribuir tags gramaticais a cada um dos *tokens*. Este processo designa-se por *part-of-speech tagging* ou POS *tagging*. Assim, utilizando, por exemplo, a frase "O Carlos comprou recentemente uma casa", ao aplicar-se o tagger obtém-se o resultado que está apresentado na Tabela 4.

Tabela 4: - Resultado após aplicação do *tagger*

Texto	**DEP**	**TAG**
O	det	DET
Carlos	nsubj	PROPN
comprou	ROOT	VERB
recentemenente	advmod	ADV
uma	det	DET
casa	obj	NOUN
.	punct	PUNCT

Na Tabela 4 pode-se observar que, de facto, na terceira coluna, designada por TAG, ocorre a atribuição da classe gramatical a cada *token* do texto, no qual a "Carlos" é atribuída a *tag* PROPN, ou nome próprio, e a "comprou" a *tag* VERB, referente a verbo. Para além disso, na segunda coluna, DEP, verifica-se a ocorrência da atribuição das dependências sintáticas, isto é, as relações entre os *tokens* do texto.

4.2.3 *Dependency Parsing*

Na secção anterior vimos que a coluna DEP (Tabela 4) refere as dependências sintáticas entre os *tokens*. Essas dependências podem ser observadas na árvore de *parsing* (Figura 8), em que os tokens estão ligados entre si através de arcos.

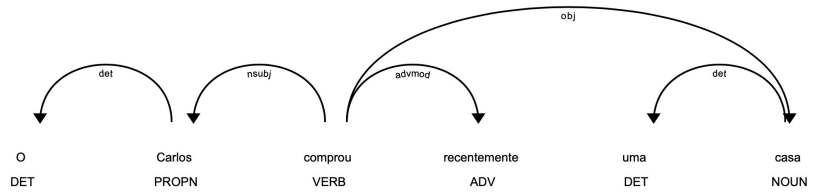


Figura 8: Exemplo de uma árvore de *parsing*

Através do exemplo apresentado na Figura 8 é possível observar que os tokens estão ligados por arcos direcionais, sendo importante saber distinguir os tokens dos quais saem os arcos dos tokens para os quais os arcos estão direcionados. **Filho/Dependente** é a designação dada aos *tokens* para onde o arco aponta e Cabeça aos tokens dos quais saem os arcos. Assim sendo, o token "Carlos" é Filho do *token* "comprou", pois é o sujeito ligado ao verbo *comprar*, e Cabeça do *token* "O", devido ao facto de este ser um determinante ligado ao *token* "Carlos". Como estas relações sintáticas formam uma árvore, cada palavra tem exatamente uma Cabeça. Assim, é possível iterar sobre os arcos na árvore iterando sobre as palavras da frase (Honnibal and Montani, 2017). Este tipo de representação gráfica, em árvore, é fornecida pelo spacy, através de um módulo de visualização específico que pode ser utilizado através do comando displacy.serve, passando-lhe um objeto ou uma lista de objetos Doc.

4.2.4 *Named Entity Recognition (NER)*

Outra componente da pipeline de processamento é o *named entity recognition* (NER), que consiste em atribuir etiquetas aos diferentes tokens da frase. Existem vários de tipos de etiquetas, como lugar, pessoa, empresa, produto, número, dinheiro, data, entre outros.

Utilizando a frase anterior como exemplo, mas agora acrescentando-lhe uma localidade, "O Carlos comprou recentemente uma casa em Londres", e aplicando o NER, obtém-se o resultado que está apresentado na Figura 9. Com isso podemos ver que "Carlos" foi identificado como pessoa (PER) e Londres como uma localidade (LOC).

Figura 9: Exemplo de aplicação do NER

O spacy possui muitas outras funcionalidades, que não serão abordadas neste trabalho, mas que são de grande ajuda no que toca ao processamento de texto. Para um estudo mais pormenorizado desta ferramenta e do que esta é capaz de oferecer, em Honnibal e Montani (2017) podemos encontrar bastante informação sobre as funcionalidades do spacy.

4.2.5 *Neo4J*

O Neo4J é uma base de dados NoSQL que pertence à categoria das bases de dados orientadas para grafos. Segue uma estrutura constituída por nodos e relacionamentos entre estes, na qual são aplicados algoritmos sofisticados e cálculos matemáticos para uma recuperação eficiente de dados. Para além disso, o Neo4j garante o comportamento ACID (Atomicidade, Consistência, Isolamento e Durabilidade), característica que o torna numa das poucas bases de dados NoSQL que suportam operações transacionais (Santos López and Santos De La Cruz, 2015).

A escolha desta tecnologia foi motivada pelo facto de ser aquela que mais se enquadra no contexto em questão, pois também uma ontologia é constituída por relacionamentos entre classes, sendo essas classes aqui representadas como nodos. Para além da facilidade na visualização e interação com os dados, diferentemente das bases de dados relacionais, como por exemplo MySQL, no Neo4J não é necessária a configuração de chaves primárias e chaves estrangeiras para determinar que campos podem ter relacionamentos e para que dados, bastando adicionar qualquer relação entre qualquer nodo sempre que for preciso, o que torna este sistema muito flexível (Adelino, 2019). A plataforma Neo4j Graph, da qual a base de dados Neo4j faz parte, é constituída por várias ferramentas e bibliotecas que facilitam o desenvolvimento, de entre as quais o Neo4j Desktop e o Neo4j Browser. Para gerir instâncias de Neo4J localmente para desenvolvimento utilizámos o Neo4J Desktop, e para a visualização, interrogação e interação dos dados o Neo4J Browser. Assim sendo, é necessário abrir o Neo4J Desktop, um ambiente de gestão de instâncias de Neo4J, onde é possível gerir inúmeros projetos e servidores de bases de dados. Para além disso, esta gestão pode ser feita utilizando apenas a interface que o Desktop fornece, sem necessidade da linha de comandos. Na figura seguinte apresenta-se um exemplo de um projeto

criado, denominado por "Ontology_Extraction" e também de uma base de dados criada dentro deste projeto, designada por "ontologydb".

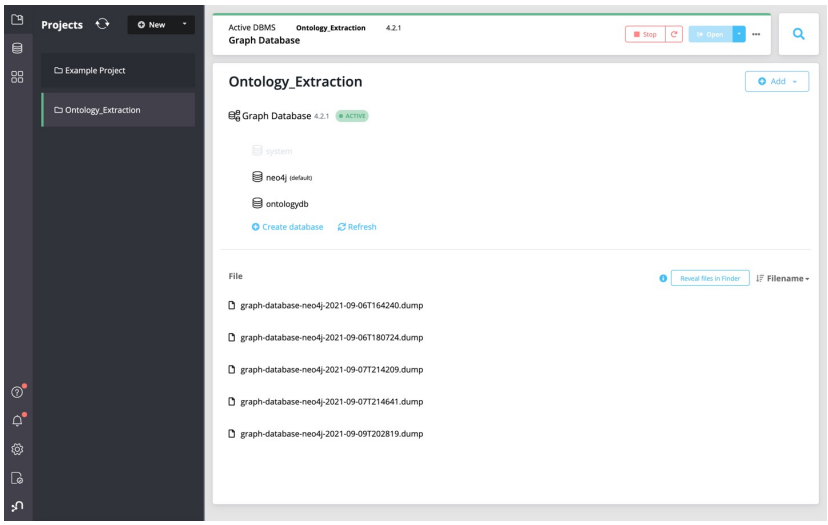


Figura 10: Exemplo de um projeto no Neo4J Desktop

Como foi referido inicialmente, toda a parte de programação foi feita utilizando a linguagem de programação python e, para fazer a ligação com o Neo4J, foi necessário configurar certos aspetos, configuração essa que pode ser consultada em Sullivan (2021). Utilizando agora o exemplo anterior, "O Carlos comprou recentemente uma casa em Londres", e apresentando esta informação no Neo4J, obtém-se o modelo apresentado na Figura 11:

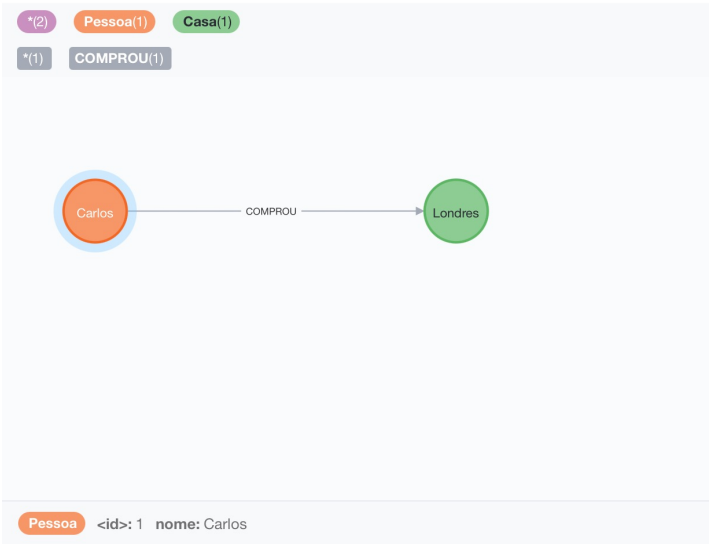


Figura 11: Representação dos dados no Neo4J Browser

A partir da representação dos dados na Figura 11, pode-se observar que existem dois tipos de nodos, o nodo "Pessoa", representado a laranja, e o nodo "Casa", representado a verde. Repare-se, ainda, que cada nodo tem pelo menos um atributo associado a ele, tendo o nodo "Pessoa" o atributo "nome" e o nodo "Casa" o atributo "localidade", que pode ser visto no canto inferior esquerdo, quando se coloca o cursor sobre o nodo. Por fim, podemos ver, ainda, o relacionamento "COMPROU" que liga os nodos referidos.

De seguida criou-se um outro nodo "Pessoa", com o atributo "nome", com o valor 'Maria', e um novo relacionamento "temFilha", que liga o nodo "Pessoa" com nome 'Carlos', ao nodo "Pessoa" com nome 'Maria'. Desta forma, fica explícito que 'Carlos', para além de ter comprado uma casa em 'Londres', tem ainda uma filha cujo nome é 'Maria'.

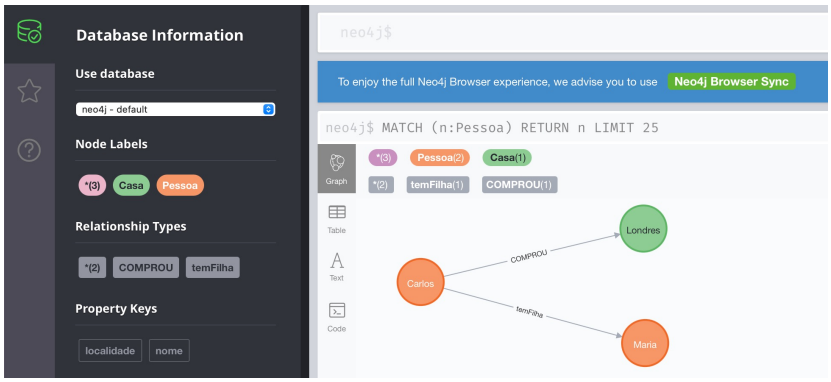


Figura 12: Representação dos dados no Neo4J Browser

4.3 O PROCESSO DE EXTRAÇÃO APLICADO

Após uma análise detalhada da estrutura dos testamentos, foi possível encontrar alguns padrões comuns, o que permitiu, posteriormente, a conceção de uma pipeline referente ao processo de extração. Assim sendo, este pode ser dividido em várias etapas distintas, em cada uma das quais ocorre a aplicação de diferentes métodos, com o intuito de obter o resultado pretendido.

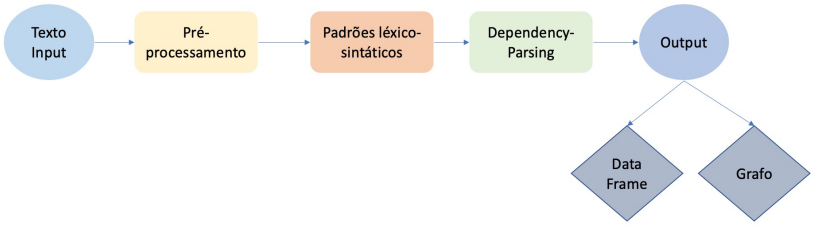


Figura 13: *Pipeline* do processo de extração

Como é possível observar na Figura 13, no início do processo realiza-se um pré-processamento do texto, operação na qual se realizam algumas alterações, de modo a tornar o texto mais legível, como, por exemplo, a remoção de carateres que dificultam a interpretação do texto como "/" e "=". De seguida são aplicados os padrões léxico-sintáticos que foram definidos. Estes consistem, basicamente, num conjunto de estruturas linguísticas generalizadas, ou esquemas, que indicam relações semânticas entre os termos e que podem ser aplicados à identificação de conceitos formalizados e de relações conceptuais em textos em linguagem natural. Nestes padrões são também utilizadas expressões regulares com o intuito de identificar determinados conceitos. Após a aplicação dos padrões realiza-se o *dependency-parsing*, que faz a análise da estrutura gramatical de uma frase e descobre palavras relacionadas, bem como o seu tipo de relacionamento (Jaiswal, 2021), como, por exemplo, o complemento direto, o sujeito, etc. Por fim, obtêm-se todos os triplos referentes à ontologia. A ontologia é representada em dois formatos distintos, numa estrutura de grafos e numa estrutura *dataframe*.

4.3.1 *Pré-Processamento*

Antes da aplicação de qualquer técnica, é necessário garantir que os dados a trabalhar estão prontos a serem usados. Assim sendo, o pré-processamento dos textos é essencial para a posterior análise do conteúdo destes, compreendendo essa etapa prévia os processos de limpeza e transformação dos dados. O processo de limpeza consiste, principalmente, na inclusão de dados em falta e na correção ou remoção de dados considerados incorretos ou irrelevantes. Já o processo de transformação passa pela normalização e reestruturação dos dados, com o intuito de facilitar a posterior extração de informação.

De seguida, na Figura 14, apresentamos como exemplo um excerto do texto de um testamento antes de ter sido realizada qualquer alteração.

Testamento e vltima vontade que mandou fazer / Joze Carreiro deste Leugar de Picotte

Em nome de Deos Amen Saivão queantos este publico / jnstromento de testamento e vltima vontade / virem que semdo no ano do nacimento de noso / Senhor Jesus Christo do ano de mil Setecentos e noventa / e seis anos aos vinte e oito do mes de ovtubro [52] do dito ano neste Leugar de Picotte cazas / da morada de Joze Carreiro do mesmo Luegar aonde / heu escrivão vim por ser chamado hai achei doente e de cama / ao sobredito Joze Carreiro de que dou fe dise parante / min e das testemunhas ao diante nomiadas e no fim deste / jnstromento asinadas que elle se achava gravemente / emfermo[9] porem com seus cinco semtidos e juizo perfeito / e que cem efeito estava ao parecer de min Escrivão e das ditas / testemunhas de que dou fe dise que se temia a morte e que / poriso queria fazer seu testaminto para testar pella sua alma / e dispor de suas coizas como mais comvinente lhe parecese / o que fes pella maneira segente pirmeiramente dise elle testador / que cer e porfesa na llei de christo e nos misterios da Samtisima / trimdade e em todos os dogmas da fe catollica Romana / e que nella protesta viver e morrer como verdadeiro e fiel / christão Jttem dise elle testador que semdo deos servido llevallo / desta vida tramzitoria para a terna quere que o seu corpo / seija sepultado dentro da jgreija Matris de São joão aonde / lhe farão todos os vzos e custumes della = Jttem dise elle / testador que quere que lhe facão hum ovficio de corpo / perzente de nove lliçois pagos os padres a duzentos e cuarenta / = e deixa ao parraco pell' asistenca deste ovficio / seiscentos[10] reis = Jttem dise elle testador que deixa cinco / alquires de caridade as pesouas que acompanharem o / seu corpo para a jgreija = Jttem dise que deixa treze misas / de tencão = Jttem dese que deixa huma misa ao / Santo christo da voua morte e mais oitra misa ao Santo / christo da Santa Crus = Jttem dise que deixa vinte misas / por sua alma = Jttem dise que quere que lhe ovfereca a sua / companheira joana com hum baramdão e hum ano e comforme / he vzo e custume neste Leugar = Jttem dise que elle he / irmão na comfraria da ermandade deste mesmo Leugar e na / comfraria de nosa Senhora do monte de duas jgreijas e que / quere que lhe facão os ovficios pagando os caidos e [52v] Lutuozas = Jttem

Figura 14: Exemplo de um excerto de testamento extraído de Barros e Alves (2019).

Como se pode observar na figura, torna-se muito difícil, por parte do computador, extrair qualquer tipo de informação do documento no estado atual, no qual o texto praticamente não apresenta pontuação, incluindo alguns carateres que dificultam mais esse processo, por exemplo as barras oblíquas "/", indicadoras da mudança de linha no manuscrito, e os sinais de igual "=", que indicam final de frase e princípio de uma nova. Da análise deste documento é possível verificar a existência de carateres que nada acrescentam ao conteúdo do testamento, tornando-o menos legível, nomeadamente os carateres "=" (que equivaleria a um ponto final ou ponto e vírgula), "/", "[ ]" (indicando acrescento de grafemas em falta por parte dos editores), "+", "( )", entre outros. Para além disso, pode-se ainda inferir que a sequência "Jttem disse ..." (equivalendo o latinismo item a idem, também, mais) ou, em outros testamentos, apenas "Disse ..." marcam o início de uma frase. Assim sendo, foram eliminados todos os carateres mencionados anteriormente e foram adicionados pontos finais sempre que apareciam os sinais ou expressões que marcam o início de uma frase, facilitando assim o posterior trabalho da máquina. É de realçar que a expressão referida pode surgir escrita de diferentes maneiras, evidenciando a variação gráfica, fonética e fonológica, como, por exemplo, "Jttim dise ...", "Item dise ...", "Jttim disse ...", "Jtem desse ...", entre outras.

Outra das alterações feitas nesta etapa foi a padronização de certas palavras que ao longo dos testamentos aparecem escritas de diferentes maneiras e que, posteri-

ormente, podem ter significados diferentes no que toca à sua análise gramatical. Veja-se, por exemplo, as palavras "dez", "quarenta", "reis" (= réis, moeda), "feitio" ou "quer". De seguida, na Figura 15, apresenta-se um esquema com as várias fases do pré-processamento dos documentos.

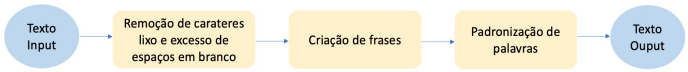


Figura 15: Ilustração das fases do pré-processamento

Após a aplicação dos métodos mencionados anteriormente sobre o texto apresentado na Figura 14 obtivemos o texto que apresentamos na Figura 16.

Testamento e vltima vontade que mandou fazer Joze Carreiro deste Leugar de Picotte

Em nome de Deos Amen Saivão queantos este publico jnstromento de testamento e vltima vontade virem que semdo no ano do nacimento de noso Senhor Jesus Christo do ano de mil Setecentos e noventa e seis anos aos vinte e oito do mes de ovtubro do dito ano neste Leugar de Picotte cazas da morada de Joze Carreiro do mesmo Luegar aonde heu escrivão vim por ser chamado hai achei doente e de cama ao sobredito Joze Carreiro de que dou fe dise parante min e das testemunhas ao diante nomiadas e no fim deste jnstromento asinadas que elle se achava gravemente emfermo porem com seus cinco semtidos e juizo perfeito e que cem efeito estava ao parecer de min Escrivão e das ditas testemunhas de que dou fe dise que se temia a morte e que poriso queria fazer seu testaminto para testar pella sua alma e dispor de suas coizas como mais comvinente lhe parecese o que fes pella maneira segente pirmeiramente dise elle testador que cer e porfesa na llei de christo e nos misterios da Samtisima trimdade e em todos os dogmas da fe catollica Romana e que nella protesta viver e morrer como verdadeiro e fiel christão . Jttem dise elle testador que semdo deos servido llevallo desta vida tramzitoria para a terna quere que o seu corpo seija sepultado dentro da jgreija Matris de São joão aonde lhe farão todos os vzos e custumes della . Jttem dise elle testador que quere que lhe facão um ovficio de corpo perzente de nove lliçois pagos os padres a duzentos e quarenta e deixa ao parraco pell asistenca deste ovficio seiscentos reis . Jttem dise elle testador que deixa cinco alquires de caridade as pesouas que acompanharem o seu corpo para a jgreija . Jttem dise que deixa treze misas de tencão . Jttem dese que deixa uma misa ao Santo christo da voua morte e mais oitra misa ao Santo christo da Santa Crus . Jttem dise que deixa vinte misas por sua alma . Jttem dise que quere que lhe ovfereca a sua companheira joana com um baramdão e um ano e comforme he vzo e custume neste Leugar . Jttem dise que elle he irmão na comfraria da ermandade deste mesmo Leugar e na comfraria de nosa Senhora do monte de duas jgreijas e que quere que lhe facão os ovficios pagando os caidos e Lutuozas . Jttem dise que deixa uma moreira

Figura 16: Excerto de um testamento após ser pré-processado

Após a realização destas modificações no texto base, torna-se mais fácil utilizar o nlp do spacy sobre o texto, o que faz com que seja possível aceder a cada palavra e à informação com que está associada, desde a sua classe gramatical até ao tipo de dependências que possui em relação às restantes palavras da frase na qual está inserida. Apesar de ser possível lematizar cada palavra após a aplicação do nlp do Spacy, esta técnica não foi utilizada devido ao facto de o Spacy possuir um modelo para o português contemporâneo e, visto que os testamentos estão escritos em português setecentista, ao lematizar algumas palavras o spacy confunde e lematiza para palavras próximas do seu modelo. Assim, frequentemente estas ações conduzem a situações de erro.

4.3.2 *Padrões léxico-sintáticos*

Os padrões léxico-sintáticos permitem extrair conceitos através das relações semânticas entre as palavras. Após uma análise detalhada dos testamentos, e tendo em conta a natureza dos documentos disponíveis, verificámos que existem certos conceitos que são comuns em todos eles. Como tal, a aplicação dos padrões teve um grande papel na sua extração. Os termos identificados pelos padrões foram o testador (a pessoa que mandou fazer o seu testamento), o escrivão (pessoa que escreveu o testamento), o local onde o documento foi feito, a data em que o mesmo foi redigido, as testemunhas presentes, os herdeiros e os familiares do testador e ainda os legatários, que são os indivíduos a quem o testador deixa algum legado, ou *deixa*. Por exemplo, o padrão usado para extrair o filho do testador tem a seguinte forma:

- seu filho {nome}+ de? {nome}*

No padrão, os {} significam que as palavras que estes envolvem podem ter qualquer valor, enquanto as restantes são palavras fixas que aparecem exatamente assim no texto. O sinal "+" tem o significado de "um ou mais", isto é, existe pelo menos um nome a seguir à palavra "filho", enquanto o sinal "*" tem o significado de "zero ou mais". Assim, podem ou não existir nomes a seguir à palavra "de". Por fim, o sinal "?" junto à palavra "de" tem o significado de "zero ou 1", podendo esta palavra aparecer ou não na frase e, se aparecer, será no máximo uma vez. Com este padrão é possível extrair do texto processado informação acerca de frases como "Deixa a seu filho Manuel ...", "Deixa a seu filho Manuel Oliveira" ou ainda "Deixa a seu filho Manuel de Oliveira". Este padrão foi utilizado como base para extrair todos os outros membros familiares, alterando-se apenas o grau de parentesco que se pretende extrair. Para extrair qualquer membro poderíamos definir um padrão geral, que poderia ter a seguinte forma:

- (seu | sua) (filho | filha | marido | mulher | primo | prima | sobrinho | sobrinha | afilhado | afilhada | irmão | irmã | cunhado | cunhada) {nome}+ de? {nome}*

Neste padrão, a " | " é um operador lógico com valor de disjunção, ou seja, em (seu | sua) o pronome pessoal a extrair pode ser "seu" ou "sua", consoante o género do grau de parentesco a extrair. No entanto, optámos por criar um padrão distinto para cada grau de parentesco, para que posteriormente fosse mais fácil distinguir o grau de parentesco com que se está a trabalhar. Relativamente à extração dos herdeiros e dos terceiros (qualquer pessoa de confiança que se deixa legalmente

encarregue de fazer cumprir o que se estipula no testamento e de tomar as medidas necessárias conforme o que pede o testador; todos os testamentos o nomeiam, para garantia do seu cumprimento; chama-se terceiro, testamenteiro ou executor do testamento), o procedimento foi um pouco mais complicado, uma vez que foram identificadas várias expressões possíveis para este conceito. Tendo isso em consideração, definimos os seguintes padrões:

- (nomeia | nomiava | deixa) para seu (erdeiro | terceiro) a seu (filho | afilhado | marido | irmão | primo | sobrinho | cunhado) {nome}+ de? {nome}*
- (nomeia | nomiava | deixa) para seu (erdeiro | terceiro) a {nome}+ de? {nome}*
- (nomeia | nomiava | deixa) (para | por) seus (erdeiros | terceiros) todos? (os | aos) seus filhos e filhas {nome}+ {e nome}*
- (nomeia | nomiava | deixa) (para | por) seus (erdeiros | terceiros) a seus filhos {nome}+ {e a nome}*
- (nomeia | nomiava | deixa) (para | por) seus (erdeiros | terceiros) a seu marido {nome}+ {e a nome}*
- (nomeia | nomiava | deixa) (para | por) seus (erdeiros | terceiros) a seu marido {nome}+ e a seu filho {nome}+
- (nomeia | nomiava | deixa) por (erdeiro | terceiro) de todos os seus bens (o | a | os | as) (seu | sua | seus | suas) (marido | mulher | filha | filhas | filho | filhos ... primos) {nome}+ {e nome}*

Estes foram os padrões base que foram definidos para extrair os herdeiros, juntamente com os terceiros ou testamenteiros. Para além destes padrões foram criados outros semelhantes, mas tendo em conta o género feminino e o plural para cada um dos conceitos em tratamento.

No que diz respeito à extração dos legatários, falta apresentar e explicar o padrão utilizado em expressões como “Deixa a Maria ...” ou “Deixa a Maria da Silva ...”, nas quais o testador não menciona nenhum grau de parentesco. Achamos que estas referências dizem respeito a pessoas amigas, vizinhas ou conterrâneas. O referido padrão tem a seguinte forma:

- Deixa a nome+ de? {nome}*

Estando os testamentos escritos em português do século XVIII, para além da escrita de certas palavras, a construção frástica difere muitas vezes do português

atual. A aplicação dos padrões apresentados em Taba e Medeiros Caseli, (2014), mencionados no capítulo anterior, não se mostrou muito eficaz na extração de informação. Porém, ainda no que toca a relações de hiponímia, foram identificados dois padrões semelhantes. Foram eles:

- T1 como são T1, T2 ... (e | ou)? Tn
- T1 que são T1, T2 ... (e | ou)? Tn

em que T1 representa o primeiro termo na relação e T2 o segundo. Para a extração do escrivão, o padrão criado foi:

- Eu? {,}? {nome_do_escrivão}+ {,}? Escrivão

Este padrão aparece em expressões como "Eu, Manuel Carreiro, Escrivão", "Manuel Carreiro, Escrivão", ou "Manuel Carreiro", em que podemos verificar que o pronome "Eu" e as vírgulas podem ou não estar presentes na frase.

Em relação ao testador, ao local onde o testamento foi feito e às testemunhas, ao invés de recorrer a padrões léxico-sintáticos, utilizámos expressões regulares para fazer a extração. No caso do testador e do local, esta escolha deveu-se unicamente ao facto de se saber que estes dois elementos figuram sempre no título do testamento. Como o texto é processado linha a linha, estes conceitos tornam-se de fácil extração. No entanto, também se poderia optar pela utilização de padrões, usando, por exemplo estruturas de reconhecimento como as seguintes:

- lugar de {nome_do_lugar}+ de? {nome_do_lugar}* para a extração do local
- (mandou fazer | fes) {nome_do_testador}+ de? {nome_do_testador}* para a extração do testador

O padrão usado para extrair a localidade pode ser encontrado em expressões como "lugar de Picote", ou "lugar de Fonte de Aldeia", em que se verifica que a primeira parte "lugar de" está sempre presente, enquanto o nome da localidade pode variar entre um nome simples e um nome composto, que pode ou não conter a preposição "de". Em relação ao padrão utilizado para a extração do nome do testador, podemos verificar que a estrutura é muito semelhante ao padrão anterior. Também este possui a parte inicial fixa que pode variar entre "mandou fazer" e "fes" e depois o nome do testador, que pode variar entre um nome simples e um nome composto. Exemplos de expressões em que é possível aplicar este padrão são, por exemplo, "mandou fazer José Fidalgo", "fes José Fidalgo", ou ainda "mandou fazer Manuel de Oliveira".

No processo de extração das testemunhas, verificou-se que estas aparecem em dois momentos distintos: a seguir à expressão "que asinou a rogo da testadora por ella lhe dar liccenca" e, no final de cada testamento, a seguir ao substantivo "Assinaturas:", acrescentado pelos editores para mais fácil compreensão do texto pelo leitor, uma vez que as mesmas já não se acham manuscritas, para fácil determinação da sua natureza. No entanto, como o número de testemunhas não é o mesmo para todos os testamentos, a utilização de padrões léxico-sintáticos não é viável, visto que não se saberia o número total de campos. Assim sendo, recorreu-se, mais uma vez, à definição de expressões regulares específicas capazes de captar esta informação.

- testemunhas\s*perzentes\s*([A-Z]*[a-z]*)+\s*que\s+asinarao'

Com esta expressão regular conseguimos extrair as testemunhas de frases como "testemunhas perzentes António Carreiro José Silva Manuel Miranda que asinarao". Aqui podemos verificar que não existe nenhum delimitador a separar os nomes das testemunhas, o que torna difícil distingui-las. De modo a resolver este problema, e após uma análise detalhada dos vários textos do testamentos, optámos por fazer um processamento em que separamos as testemunhas a cada dois nomes, pois verificámos que na maioria dos casos esta era a solução mais adequada. Assim sendo, no caso acima mencionado, as testemunhas seriam: António Carreiro, José Silva e Manuel Miranda.

Existe ainda outro caso em que as testemunhas são mencionadas, tendo-se criado outra expressão regular para a extração das mesmas:

- testemunhas\s*perzentes\s*([A-Z]*[a-z]*)+\s*que \s+asinou\s+a\s+rogo\s+da\s+testadora\s+([A-Z]*[a-z]*)+\s*que

Esta expressão regular permite extrair as testemunhas de frases como "testemunhas perzentes António Carreiro José Silva Manuel Miranda que asinou a rogo da testadora Maria Martins que".

É de salientar que este caso ocorre apenas quando é uma mulher a mandar fazer o seu testamento. Assim, tal como no caso anterior, a metodologia adotada para a distinção das testemunhas foi a separação a cada dois nomes.

4.3.3 *Dependency Parsing*

Por fim, precisámos de extrair a informação relativa às heranças e legados deixados pelos testadores. Da análise dos testamentos disponíveis, observou-se que também aqui é possível verificar a existência de uma espécie de padrão, surgindo expressões

como “Ittem dise que deixa uma camisa ...”, “Ittem dise que deixa cinco alquires de pão cozido ...”, ou, ainda, “Ittem dise que quer que o seu corpo seija sepultado ...”, em que verbos como deixar, querer, nomear, por exemplo, são usados para indicar coisas que o testador deixa como herança ou coisas que quer que lhe façam depois da sua morte. Apesar de se conseguir identificar estes padrões, o uso da técnica anterior para a extração de informação não é viável nesta situação, devido ao facto de não se saber exatamente o número de campos que vêm a seguir aos verbos e a ordem com que aparecem na frase. Sabendo isso, decidiu-se utilizar a técnica de *dependency parsing*, que, como foi explicado anteriormente (Secção 4.3), consiste na análise das dependências sintáticas entre as palavras. Porém, antes de se aplicar esta técnica, após a extração do nome do testador, na etapa anterior, procedeu-se à substituição das expressões como “Item disse que...”, “Item disse o testador que...”, e outras semelhantes, pelo respetivo nome do testador, obtendo-se assim, por exemplo, “José Folgado deixa vinte missas ...” ao invés de “Item disse que deixa vinte missas...”. Feita esta substituição, foi possível dividir cada frase em três partes, tal como acontece com um triplo numa ontologia (Sujeito, Predicado, Objeto), sendo o predicado neste caso o verbo a captar e o objeto a herança e/ou o legado deixados pelo testador. Em relação ao sujeito existem três casos distintos, por exemplo: “José” ou “José Silva” ou ainda “José da Silva”. No caso em que o sujeito é do tipo simples, obtemos a árvore de *parsing* apresentada na Figura 17.

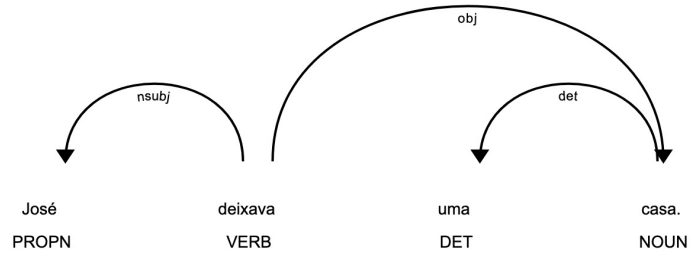


Figura 17: Exemplo de um caso de primeiro tipo de sujeito

No caso em que o sujeito é do tipo composto, o tipo de dependências obtidas está apresentado na Figura 18.

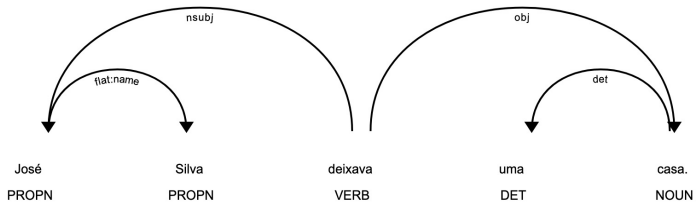


Figura 18: Exemplo de um caso de segundo tipo de sujeito

Ainda no que toca ao tipo de sujeito composto, existe o caso em que este possui a preposição "de", o que altera o tipo de dependências entre as palavras. Na Figura 19 é possível observar a árvore de *parsing* obtida quando este tipo de sujeito está presente.

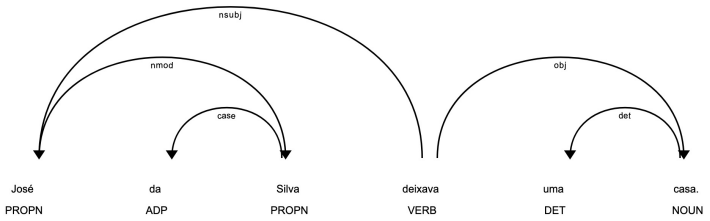


Figura 19: Exemplo de um caso de terceiro tipo de sujeito

Através da análise dos esquemas apresentados nas figuras 17,18 e 19 é possível verificar que as dependências diferem consoante o tipo de sujeito, sendo necessário ter em atenção os diferentes tipos de casos que possam ocorrer. No caso do verbo, para além do exemplo anterior, também existem algumas situações que devem ser tidas em atenção. Vejam-se frases como, por exemplo, "José quer que lhe façam...", "José quer que lhe ofereçam...", entre outros.

Quanto ao terceiro campo, a herança e/ou os legados deixados pelo testador, este foi o que mais trabalho deu, não só pela existência de inúmeros casos, mas também devido ao facto de o modelo do spacy estar preparado para o português atual, o que, em algumas situações, conduziu à atribuição de uma classe gramatical ou dependência sintática incorreta. Utilizando exemplos dos testamentos como "José deixava treze misas de temcão", em que 'temcão' corresponde à palavra 'intenção' (por intenção de algum falecido) e "José deixava cinco alqueires de pão", podemos contruir as respetivas árvores de parsing, ilustradas nas figuras 20 e 21.

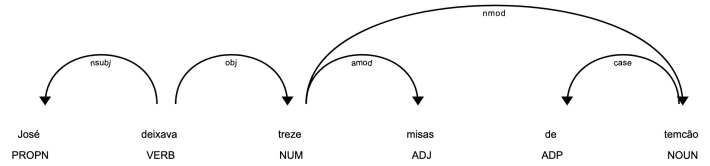


Figura 20: Representação das dependências em "José deixava treze misas de temcão"

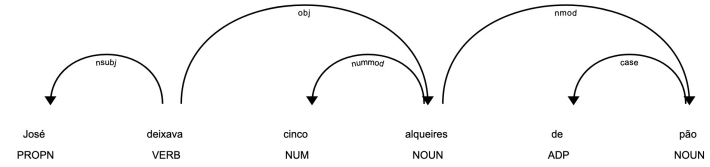


Figura 21: Representação das dependências em "José deixava cinco alqueires de pão"

Note-se que, apesar de a constituição das frases ser semelhante, existe uma diferença clara no tipo de dependências que as palavras apresentam entre si. Para além disso é possível verificar também que foi atribuída a classe gramatical "nome" à palavra 'alqueires', enquanto a palavra 'misas' é considerada como sendo um adjetivo (erradamente, muito provavelmente pelo facto do spacy não reconhecer a palavra e tentar usar outra mais próxima do seu modelo). Para além disso, na frase "José deixava vinte reis.", o "real" é, neste caso, uma moeda (com o plural reais ou réis), a árvore de dependências obtida seria a que está apresentada na Figura 22.

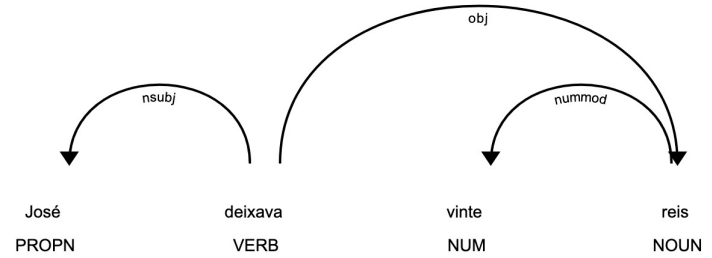


Figura 22: Representação das dependências em "José deixava vinte reis."

No entanto, se o exemplo fosse "José deixava vinte e cinco reis." já obteríamos as dependências que estão apresentadas na Figura 23.

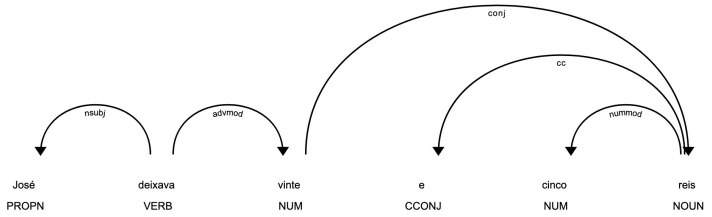


Figura 23: Representação das dependências em "José deixava vinte e cinco reis."

Uma vez que se verifica a existência de uma alteração no tipo de dependências existentes, foi necessário cobrir os vários casos, consoante aquilo que se pretendeu captar. Para a extração das heranças e dos legados deixados pelo testador foram identificados cerca de trinta casos diferentes, de que se apresentam alguns exemplos nas figuras 17,18, 19, 20, 21, 22 e 23.

4.4 EXPLORAÇÃO DA ONTOLOGIA

Terminados os processos de extração dos dados e de construção de triplos da ontologia, passámos à conceção de uma estrutura ontológica que fosse capaz de

representar a informação contida nos testamentos processados. Identificámos três classes principais: "Pessoa", "Testamento" e "Herança".

A classe "Pessoa" abrange todas as outras naquilo que diz respeito a indivíduos identificados no testamento, englobando o próprio testador até aos seus herdeiros. Esta classe foi dividida em seis subclasses, nomeadamente: "Testador", "Escrivão", "Família", "Herdeiro", "Legatário" e "Testemunha". Todas estas classes têm a propriedade 'nome' que identifica o indivíduo e, para além desta, a classe "Família" tem ainda a propriedade 'grau_', que se refere ao grau de parentesco que os indivíduos desta classe possuem com o testador. Também é importante referir a diferença que existe entre as classes "Herdeiro" e "Legatário", identificando a primeira os indivíduos que recebem a herança deixada pelo testador, ou parte dela, e a segunda identifica os indivíduos que recebem um legado por parte do testador.

Quanto às classes "Testamento" e "Herança", ambas têm a propriedade 'designação'. Em relação à classe Herança esta abrange também os legados, apesar de herança e legado serem conceitos distintos, no que toca à extração de dados relativos a cada classe, não é possível fazer a distinção de uma forma não manual. A classe Testamento tem como propriedades a data e a localidade na qual o testamento foi escrito. Na Figura 24 podemos ver um esquema da hierarquia estabelecida entre as classes definidas.

Figura 24: Esquema da hierarquia das classes da Ontologia desenvolvida

A ontologia que foi construída foi exportada para o Neo4J, o que tornou possível explorá-la através da interface do sistema ou através de queries escritas em Cypher, a linguagem de querying do Neo4J. Nas Figuras 25 e 26 temos um dos testamentos

utilizados no processo de extração e na Figura 27 sua posterior representação gráfica da ontologia criada no ambiente do Neo4J.

Testamento e vltima vontade que mandou fazer Luiza / velloza veuva que ficou de Manoel goncalves deste Leugar / de Semdim

Em nome de Deos Amen Saivão quantos este publico / jnstromento de Testamento e vltima vontade virem / que semdo no ano do nacimento de Noso Sinhor[15] Jesus Cristo / do ano de mil Setecemtos e noventa e seis anos aos vinte / e hum dias do mes de Nobembro do dito ano neste Leugar de / Semdim cazas da morada de Luiza velloza do mesmo Leugar / aonde heu Escrivão vim por ser chamado <a↑>hi achei doente / e de cama a sobredita Luiza velloza de que dou fe dise / parante min e das testemunhas ao diante nomiadas e no / fim deste jnstromento asinadas que ella se achava gravemente / emferma porem com seus cinco semtidos e juizo / perfeito e que çem efeito estava ao parecer de / min escrivão e das ditas testemunhas de que dou fe [54] dise ella testadora que se temia a morte e / que por[*i*]so queria fazer seu testamento para testar / pella sua alma e despor de suas coizas como mais / comvinente lhe parecese o que fes pella maneira segente / pirmeiramente dese ella testadora que cer e porfesa / na llei de christo e nos misterios da Samtisima Trimdade / e em todos os dogmas da fe Catollica Romana / e que nella protesta viver e morrer como verdadeira / e fiel cristãun = Jttem dise ella testadora que semdo / Deos servido llevalla desta vida tramzitoria para / a terna quere que o seu corpo seije sepultado dentro / da jgreija Matris de São pedro do mesmo Leugar / aonde lhe farão todos os vzos e custumes della = Jttem / dise ella testadora que quere que lhe facão hum ovficio / de corpo perzente de nove llicois pagos os padres / a duzentos e cuarenta = Jttem dise ella testadora que / deixa seis alqueres de caridade repartidos pellas mesmas / portas dos pobres = Jttem dise ella testadora que deixa / trinta e duas misas por diversas temcois = Jttem dise / que deixa vinte cinco misas por diversos defuntos = / Jttem dise ella testadora que deixa cem misas por sua alma / = Jttem dise ella testadora que ella he irmãn na [*comfraria*] da ermandade / da Senhora dos Rimedios deste Leugar e na ermandade / de <+↑> duas jgreijas e na de ovteiro <+↑>[16] digo de fonte de / aldeia e na de ovteiro e na de ventuzello e quere que lhe digão os ovficios pagando os caidos e llutuozas = / Jttem dise ella testadora que deixa a cada filho tres baras / de llemco e ao genrro oitras tres = Jttem dese que deixa a cada / nora sua camiza = Jttem dese que deixa a seu filho / Jgnacio huma cama e hum llançol framjado = Jttem / dese que deixa a sua filha joaquina huma cama perparada / como lhe pertemce e dois llançois e hum framjado = / dise mais que deixa a mesma sua filha cemcoenta mel / reis que lhe deixou seu tio Joze goncalves dagram barbosa / e para comprimento dos caidos lhe deixa cuatro / baras de llemço = deixa mais a sua filha o que ovver na arca / fateira que não esteija emventeirado = tudo isto / he pello amor de deos e pello travalho de a estar a [*a*]sestir[17] [54v] = Jttem dise que deixa a seu filho Luis + fran[*cisco*] / hum llamçol framjado = Deixa a seu filho Jgnacio hum / traveseiro framjado de duas bocas = Deixa

Figura 25: Testamento utilizado no processo de extração - retirado de Alves e Barros, (2019:250-252)

huma mantilha / a sua neta filha de Maria digo huma mantilha redonda = / Jttem dise que deixa [*a*] Jgnacio e a Joaquina a orta da marrella / e a bu jementa nos subeijos de terça e se não lhe chegar / o pedirão na sua receita = Deixa duas baras de llenco / a tereza moreta = e oitras duas a barbora peres = Jttem / dise que nomeia para seu terceiro a seu filho Luis a / quem pede e roga lhe faca pello bem da sua alma com / toda a caridade christão e lhe deixa por seu travalho / hum cuarto da frauga = Jttem dise [*que*] quere que lhe ovfereca / a sua filha Joaquina com hum baramdão na / sepultura e comforme he vzo e custume neste Leugar / e lhe deixa por seu travalho os seus bestidos = digo que a / comta de cuatro baras de llemço são cuatro peças de llenço / á comta dos juros de cimcoenta mel reis que lhe deixou / seu tio = isto he a mesma sua filha Joaquina = / Jttem dise que deixa mais huma peça de llemço de nove / baras a Jgnacio = declara que o ovficio de corpo perzente / he somente notruno = Jttem dese que depois deste / seu testamento comprido nomeia para seus erdeiros / a seus filhos Manoel e Luis e A Jgnacio e joaquina / e Maria e Antonio =[18]
Em testemunho de verdade asim o pediu e atrogou / a min escrivão lhe escrivese este seu Testamento nesta / nota o que a dita testadora ove por bom firme e / valliozo e por este revoga oitros cuaisquera testam[*e*]ntos / ov coducellios que haja feito em nota ov fora della / e so quere que este valha teinha força e vigor e pede / e roga a todas as justicias asim ecleziasticas como / secullares o cumprão e facão muito inteiramente / comprire e goardar como nelle se cumtem / de que forão testemunhas perzentes ao feitio / deste Joze Canguiro que asinou a rogo da testadora [55] por ella lhe dar llicença Manoel / Pardal veuvo João Cuveiro Manoel faRucu Manoel Paullo Antonio ferador / Joze llourenço que Deixa mais a Luiza melchora huma bara de llemco deixa mais oitra bara de / llenco a Maria melchora = e asinarão todas / as testemunhas como testadores depois de este lhe / ser por min llido que dise ella testadora depois / de este lhe ser por min llido parante min / e das testemunhas que estava na sua vltima / vontade de que dou fe' heu Manoel Domingues / Escrivão dos testamentos do Leugar de Picotte / [Assinaturas:] eu Joze Cang*ueir*o asino a rogo da testadora
eu M*ano*el Dominges Asino a rogo da testadora
Manoel Paullo
João + Coveiro Manoel + Pardal
Joze + llurenço Antonio + feRador
heu Manoel Domingues Escrivão

Figura 26: Testamento utilizado no processo de extração (continuação) - retirado de Alves e Barros, (2019)

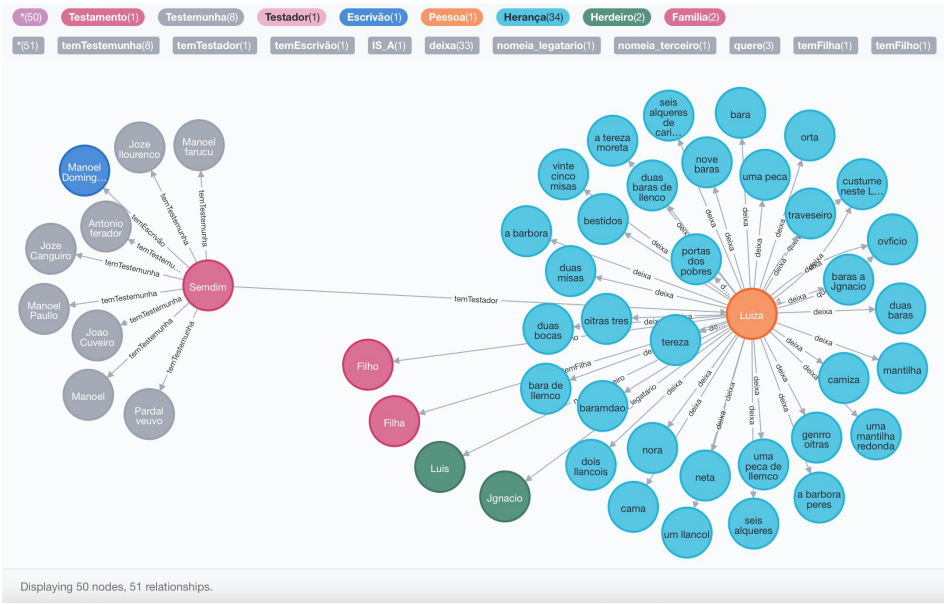


Figura 27: Representação da Ontologia em Neo4j

Através de uma breve análise da Figura 27 podemos ver que, para além dos nodos, que representam as classes mencionadas anteriormente, existem vários tipos de relacionamentos entre si. O nodo "Testamento" relaciona-se com o nodo "Testemunha", "Escrivão" e "Testador" através dos relacionamentos 'temTestemunha', 'temEscrivão' e 'temTestador', respetivamente. Desta forma, sabemos que, durante a escrita do testamento, estiveram presentes oito testemunhas, para além do escrivão 'Manoel Domingues' e que 'Luiza' foi a pessoa que mandou fazer o testamento. Se posicionássemos o cursor sobre o nodo Testamento poderíamos ver todas as suas outras propriedades (Figura 28).

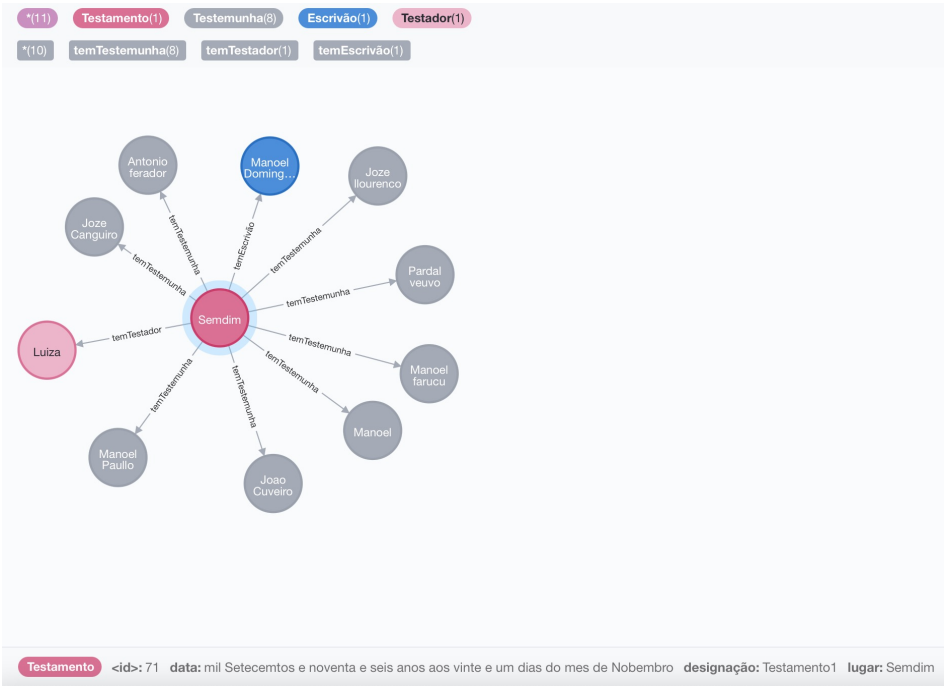


Figura 28: Representação das propriedades do nodo Testamento

Na Figura 29 também podemos observar os relacionamentos que ligam o nodo "Testador" aos nodos "Herdeiro" e "Legatário", nomeadamente 'nomeia_herdeiro'/ 'nomeia_terceiro' e 'nomeia_legatário', respetivamente. Relativamente aos relacionamentos com o nodo Família, estes variam consoante o grau de parentesco. Vejam-se, por exemplo, os relacionamentos 'temFilha' e 'temPrimo'. Por fim, ainda sobre o nodo "Testador", este relaciona-se com o nodo "Herança" através de relacionamentos como 'deixa', 'quer', 'manda', 'empresta', entre outros.

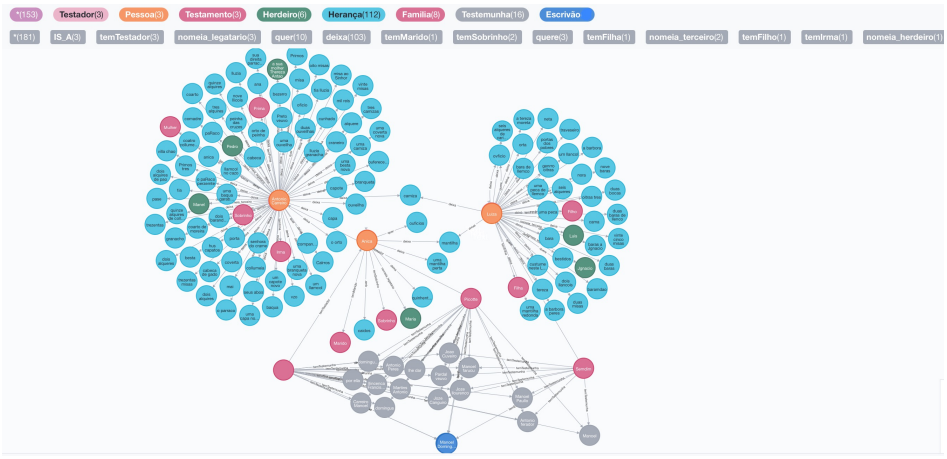


Figura 29: Representação da Ontologia com informação de três testamentos

Até agora foi mostrada a representação de informação correspondente a um testamento. Porém, é também possível representar informação de vários testamentos simultaneamente, como se pode verificar na Figura 29. Nesta podemos observar que existem alguns nodos que não se relacionam apenas com um nodo, mas sim com vários. Isto acontece, por exemplo, com os nodos cinzentos que representam as testemunhas presentes na altura em que os testamentos foram escritos. Como se trata de testamentos de uma localidade pequena, com poucos habitantes, era normal algumas pessoas serem testemunhas de vários testamentos, como é o caso do exemplo representado na Figura 27. Para além disso, podemos também ver que esses testamentos foram escritos pelo mesmo escrivão, 'Manoel Domingues', que está representado no esquema pelo nodo azul. Quanto à hierarquia de classes, mais especificamente as subclasses de Pessoa, esta é representada através do relacionamento 'IS_A', em que por exemplo, "Testador" -> 'IS_A' -> "Pessoa" significa que Testador é uma subclasse de Pessoa.

Por fim, refira-se que através da linguagem Cypher podemos definir a forma como queremos explorar os dados da ontologia. A capacidade e versatilidade da linguagem permite-nos cruzar os dados das formas mais diversificadas. Por exemplo, para sabermos qual o escrivão que tinha um filho chamado 'Joze', ou quantas vezes determinado indivíduo esteve presente como testemunha, ou ainda quais os herdeiros de certo testador, poderíamos utilizar, respetivamente, as seguintes queries em Cypher:

1. MATCH (e:Escrivão)-[:temFilho]-> (f:Familia nome: " Joze") RETURN e.nome
2. MATCH (t:Testamento)-[:temTestemunha]-> (tes:Testemunha nome: " Manoel") RETURN label(t), count(*)
3. MATCH (t:Testador nome:"Francisco")-[:nomeia_herdeiro]-> (h:Herdeiro) RETURN h.nome

4.5 ANÁLISE DE RESULTADOS

Após a extração e a representação gráfica da ontologia foram feitas algumas medições com o objetivo de avaliar a eficácia e a eficiência do sistema desenvolvido. Primeiramente, foram avaliados os resultados da aplicação das técnicas utilizadas em diferentes testamentos. Na Tabela 5 é possível observar alguns dados relativos ao processamento de seis testamentos diferentes:

Tabela 5: Comparação entre o número de palavras processadas, a precisão conseguida com a aplicação das técnicas e o tempo de processamento

Nº palavras processadas	**Padrões léxico-sintáticos (precisão)**	**Padrões léxico-sintáticos + Dependecy parsing (precisão)**	**Tempo de processamento (segundos)**
4130	1.0	0.90	83.8
4715	1.0	0.82	81.9
5020	0.9	0.82	85.7
5206	1.0	0.83	81.7
6138	0.9	0.72	84.1
6832	1.0	0.80	85.4

Através da análise da Tabela 5 pode-se verificar que a aplicação dos padrões léxico-sintáticos nos permitiu garantir uma precisão média de 96.7% na extração de dados, e que a combinação destes padrões com a técnica de dependency parsing originou uma precisão média inferior, de cerca de 82%. Esta diferença pode ser explicada pelo facto de os padrões léxico-sintáticos serem, na maioria dos casos, estruturas fixas, em que sabemos exatamente o que queremos extrair, não havendo grande margem para a extração de dados incorretos. Por outro lado, o *dependency parsing* depende muito das classes gramaticais das palavras e das dependências entre estas, e como o português utilizado nos testamentos é do século XVIII, por vezes ocorreram erros na atribuição da classe gramatical feita pelo spacy às palavras dos textos processados e, consequentemente, no tipo de dependências que essas palavras têm umas com outras. A análise dos dados da Tabela 5 também nos revela que o tempo de processamento de cada testamento é quase idêntico, ainda que o número de palavras processadas por testamento varie entre 4130 e 6832.

De seguida, foi analisado o número de triplos semânticos extraídos em cada um dos testamentos mencionados anteriormente. Da realização desta análise obtivemos os dados que estão apresentados na Tabela 6.

Tabela 6: Análise do número de triplos semânticos extraídos

Testamento	**Nº total de triplos extraídos**	**Nº de triplos extraídos (Padrões)**	**Nº de triplos extraídos (*Dependency Parsing*)**	**Nº de triplos extraídos incorretamente**
1	32	10	19	3
2	27	8	14	5
3	39	11	21	7
4	34	11	17	6
5	39	9	19	11
6	36	12	17	7

Através da análise dos dados da Tabela 6 é possível verificar que, apesar de o número de triplos corretamente extraídos ser bastante razoável (de acordo com a precisão obtida e representada anteriormente na Tabela 5), em alguns casos o número de triplos incorretamente extraídos é relativamente elevado, pelo motivo referido anteriormente. Também é possível verificar que, apesar da ocorrência de alguns erros com a aplicação do *dependency parsing*, o número de triplos extraídos com esta técnica é, na maioria das vezes, superior ao número de triplos extraídos com o uso dos padrões. Este resultado pode ser explicado pelo facto de o número de heranças e legados deixados pelo testador ser relativamente superior ao número de indivíduos mencionados no testamento.

Tabela 7: Comparação dos resultados obtidos em testamentos de várias localidades

Localidade	**Precisão**
Picote	0.83
Sendim	0.82
Fonte da Aldeia	0.75
Malhadas	0.55
Póvoa	0.1

Os testamentos analisados são da localidade de Picote, uma freguesia de Miranda do Douro. Todavia, para verificarmos a precisão do processo de reconhecimento que efetuámos sobre estes textos, decidimos compará-los com os resultados equivalentes

realizados sobre testamentos da mesma época, mas de outras localidades. Após a realização de um conjunto vasto de testes obtivemos os resultados que estão apresentados na Tabela 7. Através desta tabela, pode-se observar que as localidades de Sendim e Picote apresentam precisões idênticas no que toca à ontologia extraída. A localidade de Fonte de Aldeia, por exemplo, apresenta uma precisão bastante inferior. No entanto, as maiores diferenças verificam-se nas localidades de Malhadas e da Póvoa. Esta diferença pode ficar a dever-se ao facto de o testador e o escrivão serem diferentes em cada uma das localidades, o que, consequentemente, se reflete num tipo de escrita diferente e por serem regiões mais afastadas, quando comparamos as suas localizações com Picote, Sendim ou Fonte da Aldeia, o que, como sabemos, pode levar a que tenham grafias diferentes, algumas espelhando uma realização fonética distinta. Por fim, foram analisados os tempos de processamento de diferentes números de testamento. Os resultados desta análise estão apresentados na Tabela 8. Uma rápida análise destes resultados permite-nos concluir que, nestes casos, os tempos de processamento são relativamente proporcionais ao número de testamentos e ao número de palavras processadas.

Tabela 8: Comparação dos tempos de processamento em diferentes conjuntos ou números de testamentos

Nº de testamentos	**Nº de palavras**	**Tempo de processamento (segundos))**
1	5110	83.4
10	40520	855.2
20	83856	1613.5
50	211520	4035.9

5

CONCLUSÕES E TRABALHO FUTURO

5.1 CONCLUSÕES

As ontologias têm vindo a ganhar cada vez mais importância no que toca ao desenvolvimento de sistemas baseados em conhecimento. No entanto, para além de ainda haver certa dificuldade em compreender o seu método de construção, a construção manual destas é muito dispendiosa tanto a nível de recursos como de tempo e, após a construção, é necessário manter a ontologia atualizada consoante os novos requisitos que poderão surgir. Nesta dissertação abordámos o conceito de ontology learning, sendo esta a área correspondente à construção (semi)-automática de ontologias. Esta construção (semi)-automática torna-se muito mais vantajosa, principalmente no que toca aos recursos utilizados e ao tempo despendido para este processo. Existem várias metodologias para a extração (semi)-automática de ontologias, porém o modelo tipicamente utilizado é composto por cinco etapas, em que, na etapa inicial, de modo a eliminar inconsistências semânticas, ocorre um pré-processamento do texto usando técnicas linguísticas como, por exemplo, o *part of speech tagging*, o *parsing* ou a lematização. Após este pré-processamento, faz-se a extração de termos e conceitos relevantes do domínio, usando várias técnicas de processamento de linguagem natural, nomeadamente *syntatic parsing* e *subcategorization frames*, juntamente com técnicas de domínio estatístico, tais como a *C/NC value*, a *contrastive analysis* ou a *latent semantic analysis*. Para além da obtenção dos *clusters* de conceitos, é necessário, também, extrair as relações taxonómicas e não-taxonómicas que possam ser estabelecidas entre estes, usando, mais uma vez, técnicas de processamento de linguagem natural e técnicas de domínio estatístico. Ainda no que toca à extração, também os axiomas são extraídos, usando técnicas de programação lógica indutiva. Por fim, de modo a avaliar a integridade da ontologia desenvolvida, são utilizadas algumas técnicas próprias de avaliação de ontologias. Neste trabalho, o processo aplicado para a extração da ontologia a partir de testamentos antigos, que foram editados por Alves e Barros (2019), *O Livro dos Testamentos – Picote, 1780-*

1803, foi composto por três etapas. A primeira consistiu num pré-processamento dos textos dos testamentos, na qual, de modo a simplificar a interpretação destes, foram eliminados vários carateres indesejáveis, como as barras e os parênteses retos. Para além disso, padronizou-se a escrita de algumas palavras, que, ao longo dos testamentos, apresentavam numerosas variantes orgráficas, o que levava à ocorrência de erros e inconsistências. De seguida, após um estudo dos vários algoritmos disponíveis para a extração de termos e relações, optámos pela aplicação de padrões léxico-sintáticos e da técnica de dependency parsing, por serem aqueles que mais se adequavam. Os padrões permitiram-nos extrair relações do tipo hipónimo/hiperónimo, tendo sido fundamentais para a posterior construção da hierarquia de classes. Em relação ao *dependency parsing*, uma técnica que consiste na análise de dependências entre palavras, esta foi fundamental para a extração das heranças e legados ou, por vezes, encargos deixados pelo testador. Por fim, após a extração dos conceitos e relações, foi feita a exportação dos dados extraídos, em formato de triplos semânticos, para uma base de dados orientada a grafos, designada de Neo4J. Esta representação gráfica permitiu-nos uma melhor exploração e consulta da ontologia obtida. Concluída a fase de extração da ontologia, e de modo a avaliar a integridade desta, foi feita uma comparação da ontologia com os dados obtidos com a extração manual de algumas dezenas de testamentos. Foi possível verificar que a combinação das técnicas de *dependency parsing* e padrões léxico-sintáticos resultou numa precisão média de 82%, no que toca a dados corretamente extraídos, um resultado que pode ser considerado bastante positivo, visto que se trata de documentos escritos em português setecentista, em que existem diferenças significativas na ortografia, bem como no próprio léxico, na morfologia e na sintaxe. Para além disso, foi feita uma comparação com testamentos de outras localidades, nomeadamente Fonte de Aldeia, Malhadas e Póvoa, na qual se verificou que a precisão da ontologia obtida foi diminuindo. Essa diminuição pode ser explicada pelo facto de o testador ser diferente em cada uma das localidades (bem como, na maior parte dos casos, o escrivão), o que, consequentemente, resulta num tipo de escrita diferente, e ainda por serem regiões mais afastadas, quando comparamos as suas localizações com a de Picote. Por fim, foram ainda comparados os tempos de processamento para diferentes quantidades de testamentos, verificando-se que, à medida que aumenta o número de testamentos, o tempo de processamento aumenta de uma forma relativamente proporcional. Este resultado pode ser explicado pelo facto de os testamentos terem uma quantidade de palavras bastante semelhante. Em suma, os resultados obtidos na extração da ontologia e na sua posterior representação gráfica foram bastante satisfatórios, e através da exploração da ontologia foi possível adquirir conhecimento importante

relativamente aos testamentos de Picote, desde os vários testadores até às heranças, legados e deveres deixados por estes.

5.2 TRABALHO FUTURO

A criação deste sistema ontológico permitiu a aquisição de conhecimento no domínio dos testamentos e a sua posterior representação e exploração num sistema de grafos. Assim, foi possível analisar o conteúdo de cada testamento, permitindo-nos retirar informações como o tipo de legados e bens deixados por cada testador ou os membros da sua família. Apesar de considerarmos os resultados obtidos bastante positivos, conseguimos perceber que existe margem para melhorias, que podem ser feitas com o intuito de tornar este sistema ainda mais consistente. Assim, de forma a reduzir a quantidade de ruído, resultante dos conceitos incorretamente classificados, uma das estratégias que poderiam ser adotadas era a de criar um modelo específico no spacy, direcionado para o português utilizado nos testamentos trabalhados. Este modelo permitiria, então, obter melhores resultados quando aplicada a técnica do *dependency parsing*, atribuindo a cada palavra a classe gramatical correta e criando triplos semânticos válidos.

Outra estratégia a ser adotada seria a diminuição da automatização deste sistema ontológico, mais especificamente na fase final de validação dos triplos construídos. Assim, este processo teria uma parte supervisionada, em que o utilizador teria a opção de validar os triplos a serem adicionados à ontologia. Esta modificação, apesar de requerer trabalho manual, melhoraria o produto final, obtendo-se, assim, uma ontologia mais coerente e completa, uma vez que os testamentos são textos muito precisos, obedecendo a fórmulas, de tamanho relativamente pequeno, com um número relativamente limitado de elementos e a presença previsível de outros, permitindo um controlo manual (quase) sem falhas. Por fim, de modo a facilitar o acesso e leitura dos dados a outro tipo de ferramentas, poderá ser gerado um ficheiro em formato turtle com toda a informação relativa à ontologia construída. Esta alteração tornará possível a representação da informação em ferramentas como, por exemplo, o Protégé, que oferece como uma das vantagens a facilidade na criação de relacionamentos inversos.

BIBLIOGRAFIA

F. Adelino. O que é Neo4j - TechNote Inc., 2019. URL https://technoteinc.blogspot.com/2019/07/o-que-e-neo4j.html.

A. Alves and A. Barros. *O Livro dos Testamentos - Picote 1780-1803*. 11 2019. ISBN 978-989-99411-8-2.

M. Asim, M. Wasim, M. Khan, M. Waqar, and H. Abbasi. A survey of ontology learning techniques and applications. *Database*, pages 1–24, 2018.

C. Baccigalupo and E. Plaza. Poolcasting: A social Web radio architecture for group customisation. *Proceedings - 3rd International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution, AXMEDIS 2007*, pages 115–122, 2007.

T.L. Baségio. Uma Abordagem Semi-automática para Identificação de Estruturas Ontológicas a partir de Textos na Língua Portuguesa do Brasil Uma abordagem Semi-Automática para Identificação de Estruturas Ontológicas a partir de Textos na Língua Portuguesa do Brasil. pages 1–124, 2007. URL http://tede.pucrs.br/tde_busca/processaArquivo.php?codArquivo=2143.

K. Belhoucine and M. Mourchid. A Survey on Methods of Ontology Learning from Text. pages 113–123, 2020.

K. Bhaskar and S. Savita. A Comparative Study of Ontology building Tools in Semantic Web Applications. *International journal of Web Semantic Technology*, 2010.

K K Breitman, M a Casanova, and W Truszkowski. *NASA Monographs in Systems and Software Engineering*. 2007.

P. Cimiano and J. Völker. Natural Language Processing and Information Systems. 2002. URL https://www.actapress.com/Abstract.aspx?paperId=26042.

P. Cimiano, A. Hotho, and S. Staab. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *Frontiers in Plant Science*, pages 305–339, 2016.

L. Drumond and R. Girardi. A survey of ontology learning procedures. *CEUR Workshop Proceedings*, 2008.

E. Drymonas, K. Zervanou, and E. G.M. Petrakis. Unsupervised ontology acquisition from plain texts: The OntoGain system. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 277–287, 2010.

A. Escórcio and J. Cardozo. Editing tools for ontology creation. *Semantic Web Services: Theory, Tools and Applications*, 2007.

A. Farquhar. Ontolingua , 1997. URL http://www.ksl.stanford.edu/software/ontolingua/.

D. Faure and C. Nedellec. A Corpus-based Conceptual Clustering Method for Verb Frames and Ontology Acquisition. *Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC)*, 1998.

M. Fernández, A. Gómez-Pérez, and N. Juristo. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. 1997. URL www.aaai.org.

I. Neves Ferraz and A. Cristina Bicharra Garcia. Ontology in association rules. *SpringerPlus*, pages 1–12, 2013.

M.C. Freitas and V. Quental. Subsídios para a Elaboração Automática de Taxonomias. pages 1585–1594, 2007. URL http://www.de9.ime.eb.br/$\sim$sousamaf/cd/pdf/arq0163.pdf.

D. Gasevic, D. Djuric, and V. Devedzic. *Model Driven Architecture and Ontology Development*. 2006.

T. R. Gruber. A Translation Approach to Portable Ontology Specifications by KNOWLEDGE SYSTEMS LABORATORY Computer Science Department. *Knowledge Aquisition*, pages 199–220, 1993. URL https://pdfs.semanticscholar.org/5120/f65919f77859a974fcc1ad08f72b2918b8ec.pdf%0Ahttps://ebiquity.umbc.edu/get/a/publication/501.pdf.

M. Hazman, S. R. El-Beltagy, and A. Rafea. A survey of ontology learning approaches. *CEUR Workshop Proceedings*, pages 36–43, 2008.

M. A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora Lexico-Syntactic for Hyponymy Patterns. *International Conference on Computational Linguistics*, pages 23–28, 1992.

M. Honnibal and I. Montani. spaCy · Industrial-strength Natural Language Processing in Python. 2017.

S. Jaiswal. Natural Language Processing — Dependency Parsing | by Shivanee Jaiswal | Towards Data Science. *Industrial Data*, 2021. URL https://towardsdatascience.com/natural-language-processing-dependency-parsing-cf094bbbe3f7.

X. Jiang and A. Tan. Full-Text Citation Analysis : A New Method to Enhance. *Journal of the American Society for Information Science and Technology*, pages 1852–1863, 2013.

M. Khattak, R. Batool, Z. Pervez, M Khan, and S. Lee. Ontology evolution and challenges. *Journal of Information Science and Engineering*, pages 851–871, 2013.

T. K. Landauer, P.W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse Processes*, pages 259–284, 1998.

K. Liu, William R. Hogan, and Rebecca S. Crowley. Natural Language Processing methods and systems for biomedical ontology learning. *Journal of Biomedical Informatics*, 44:163–179, 2011. URL http://dx.doi.org/10.1016/j.jbi.2010.07.006.

P. Machado and V. Strube de Lima. Extração de relações hiponímicas em um corpus de língua portuguesa. 2015.

M.A Mason. MS Windows NT kernel description, 2015. URL https://protege.stanford.edu/.

D.L McGuinness. Ontologies come of age: The web ' s growing needs. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, pages 1–13, 2005.

E. Merelli and M. Luck. *Technical Forum Group on Agents in Bioinformatics*. 2004. URL http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Technical+Forum+Group+on+Agents+in+Bioinformatics+1#0.

M.A. Musen. Protégé. . 2015. URL https://protege.stanford.edu/.

R. Navigli, P. Velardi, and A. Gangemi. Ontology Learning and Its Application to Automated Terminology Translation. *IEEE Intelligent Systems*, pages 22–31, 2003.

N.F. Noy and D.L. McGuinness. A Guide to Creating Your First Ontology. *Biomedical Informatics Reseach*, pages 7–25, 2001. URL http://bmir.stanford.edu/file_asset/index.php/108/BMIR-2001-0880.pdf.

M. Obitko. Ontologies - Description and Applications. *Laboratory for Intelligent Decision Making and Control Series of Research Reports*, pages 1–36, 2001. URL http://cyber.felk.cvut.cz/gerstner/reports/GL126.pdf.

A. Öhgren. Ontology Development and Evolution: Selected Approaches for Small-Scale Application Contexts. 2004.

J. Park, W. Cho, and S. Rho. Evaluating ontology extraction tools using a comprehensive evaluation framework. *Data and Knowledge Engineering*, pages 1043–1061, 2010. URL http://dx.doi.org/10.1016/j.datak.2010.07.002.

C. Patel, K. Supekar, Y. Lee, and E. K. Park. OntoKhoj. page 58, 2003.

F.M. Santos López and E.G. Santos De La Cruz. Literature review about Neo4j graph database as a feasible alternative for replacing RDBMS. *Industrial Data*, page 135, 2015.

M. Shamfard and A Barforoush. An Introduction to Hasti: An Ontology Learning System. 2002. URL https://www.actapress.com/Abstract.aspx?paperId=26042.

T. Singh. Natural Language Processing With spaCy in Python – Real Python, 2019. URL https://realpython.com/natural-language-processing-spacy-python/.

L. Stojanovic. Methods and tools for ontology evolution. pages 1–249, 2004.

N. Stojanovic, L. Stojanovic, and S. Handschuh. Evolution in the ontology-based knowledge management systems. pages 840–850, 2002.

CJ. Sullivan. Create a graph database in Neo4j using Python | by CJ Sullivan | Towards Data Science, 2021. URL https://towardsdatascience.com/create-a-graph-database-in-neo4j-using-python-4172d40f89c4.

Y. Sure, S. Staab, and R. Studer. *Ontology Engineering Methodology*. 2009.

L.Sameshima Taba and H. De Medeiros Caseli. Automatic semantic relation extraction from Portuguese texts. pages 2739–2746, 2014.

M. Uschold and M. Gruninger. Ontologies : Principles , methods and applications Ontologies : Principles , Methods and Applications Mike Uschold Michael Gruninger AIAI-TR-191 February 1996 To appear in Knowledge Engineering Review Volume 11 Number 2 , June 1996 Mike Uschold Tel : Mi. *Knowledge Engineering Review*, 1996.

M. Uschold and M. King. Towards a Methodology for Building Ontologies conjunction with IJCAI-95 Abstract. *Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95*, pages 6.1 – 6.10, 1995.

G. Van Rossum and F.L. Drake. *Python 3 Reference Manual*. 2009.

P. Velardi, R. Navigli, A. Cucchiarelli, and F. Neri. Evaluation of OntoLearn, a methodology for automatic learning of domain ontologies. *Ontology Learning from Text: Methods, evaluation and applications*, page 92, 2005.