• U  C •

Proposta de Tese
Mestrado em Engenharia Informática
Sistemas Inteligentes

Construção automática de uma wordnet com medidas de confiança associadas

Fábio António Areia dos Santos

Orientador
Hugo Gonçalo Oliveira

Departamento de Engenharia Informática
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

31 de Agosto de 2015

Constituição do Júri

- Professor Luís Macedo
- Professor Nuno Pimenta
- Professor Hugo Gonçalo Oliveira

Resumo

Resumo

Numa wordnet, conceitos são representados através de grupos de palavras, vulgarmente chamados de synsets, e cada pertença de uma palavra a um synset representa um diferente sentido dessa mesma palavra. Mas como os sentidos são entidades complexas, sem fronteiras bem definidas, para lidar com eles de forma menos artificial, sugerimos que synsets sejam tratados como conjuntos difusos, em que cada palavra tem um grau de pertença, associado à confiança que existe na utilização de cada palavra para transmitir o conceito que emerge do synset. Propomos então uma abordagem automática para descobrir um conjunto de synsets difusos a partir de uma rede de sinónimos, idealmente redundante, por ser extraída a partir de várias fontes, e o mais abrangentes possível. Um dos princípios é que, em quantos mais recursos duas palavras forem consideradas sinónimos, maior confiança haverá na equivalência de pelo menos um dos seus sentidos. A abordagem proposta foi aplicada a uma rede extraída a partir de três dicionários do português e resultou num novo conjunto de synsets para esta língua, em que as palavras têm pertenças difusas, ou seja, *fuzzy synsets*. Para além de apresentar a abordagem e a ilustrar com alguns resultados obtidos, baseamo-nos em três avaliações – comparação com um tesauro criado manualmente para o português; comparação com uma abordagem anterior com o mesmo objetivo; e avaliação manual – para acreditar que os resultados são positivos, e poderão no futuro ser expandidos através da exploração de outros fontes de sinónimos.

Abstract

In a wordnet, concepts are typically represented as groups of words, commonly known as synsets, and each membership of a word to a synset denotes a different sense of that word. However, since word senses are complex entities, without well-defined boundaries, we suggest to handle them less artificially, by representing them as fuzzy objects, where each word has its membership degree, which can be related to the confidence on using the word to denote the concept conveyed by the synset. We thus propose an approach to discover synsets from a synonymy network, ideally redundant and extracted from several broad-coverage sources. The more synonymy relations there are between two words, the higher the confidence on the semantic equivalence of at least one of their senses. The proposed approach was applied to a network extracted from three Portuguese dictionaries and resulted in a large set of fuzzy synsets. Besides describing this approach and illustrating its results, we rely on three evaluations – comparison against a handcrafted Portuguese thesaurus;

comparison against the results of a previous approach with a similar goal; and manual evaluation – to believe that our outcomes are positive and that, in the future, they might my expanded by exploring additional synonymy sources.

Palavras-Chave

wordnets, synsets, fuzzy clustering, rede léxico-semântica, sinónimos, confiança, dicionários

Glossário

- **Cartão** - Rede formada pelo conjunto das redes Wikicionario, Papel, e Dicionário Aberto.
- **CBC** - Clustering By Committee. **Clip** - Recurso linguístico para a língua portuguesa, no qual são utilizadas medidas de pertença Gonçalo Oliveira (2013).
- **Clustering** - Técnica utilizada para agrupar instâncias em grupos cujos elementos partilhem características comuns.
- **CW** - Chinese Whispers.
- **Dicionário Aberto** - É um Dicionário livre e gratuito, baseado no Novo Dicionário da Língua Portuguesa de 1913 por Cândido de Figueiredo. [1]
- **ECO** - Extraction Clustering Ontologization (Gonçalo Oliveira and Gomes, 2014).
- **FCM** - Fuzzy C-Means.
- **Fuzzy clustering** - Diz-se de um técnica de agrupamentos na qual é associado uma medida de confiança, pertença ou probabilística aos elementos pertencentes a cada um dos vários agrupamentos.
- **Hard Clustering** - Diz-se de um método de agrupamento que classifica as intancias como pertencentes ou não a cada cluster, não havendo métricas associadas á sua pertença ou confiança.
-
- **LSA** - Latent Semanti Analysis.
- **MCL** - Markov Cluster Algorithm.
- **Onto.pt** - Wordnet para a lingua portuguesa, desenvolvda nos CISUC por Gonçalo Oliveira and Gomes (2014).
- **Overlapping Clustering** - Diz-se de método de agrupamento que permite que um instância possa pertence a dois ou mais conjuntos.
- **Papel** - O PAPEL é um recurso criado pela Linguateca a partir do Dicionário PRO de Língua Portuguesa da Porto Editora através de um protocolo de colaboração com o departamento de dicionários desta empresa. [2].

[1] Ver `https://log.pt/dicionarioaberto/` (Agosto 2015)
[2] Ver `http://www.linguateca.pt/PAPEL/` (Agosto 2015)

- **Passeio Aleatório** - O nome em português para **Radom Walk** é também conhecido como o "caminho do bêbado".
- **PLN** - Processamento de Linguagem Natural.
- **PMI** - Poitntwise Mutual Information.
- **Random Walk** - É uma formalização matemática da ideia intuitiva da tomada de vários passos consecutivos, cada qual em um direção aleatória. Ex o movimento de uma molécula.
- **Strict Partitioning** - Diz-se de um método de agrupamento que apenas permita 1 agrupamento por instância.
- **Synset** - Conjunto de palavras que partilham um mesmo sentido.
- **TeP** - Recurso criado manualmente para a lingua portuguesa.
- **Wikicionario** - É um projeto da wikimédia foundation dona da wikipedia, cujo objetivo é criar um dicionário de conteúdo livre disponível em várias línguas, neste documento salvo notação em contrário é utilizada apenas a versão para a língua portuguesa.
- **Wordnet** - Uma base de dados lexical, onde as palavras estão organizadas de acordo com as suas relações semânticas.

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Ao contrário das linguagens formais, a linguagem natural é ambígua. Enquanto que as primeiras estão estruturadas para não haver ambiguidade, na linguagem natural não é claro determinar o significado/contexto de cada palavra, visto que esta pode ter diferentes significados consoante o contexto onde está inserida, não existindo uma lista de regras definidas de modo a tratar esses casos.

Temos atualmente acesso a uma grande quantidade de informação, na sua maioria disponível na Internet, e precisamente em linguagem natural. De modo a possibilitar a extração de conhecimento estruturado a partir dessa informação, é necessário processá-la computacionalmente, através de ferramentas específicas para esta tarefa. Para que as extrações sejam o mais coerentes possível será necessário dotar essas mesmas ferramentas da capacidade de desambiguação.

A elaboração de recursos linguísticos de forma manual, quando feita por humanos, é uma tarefa que requer muito tempo e dedicação. No caso específico da criação de recursos que permitam a consulta dos vários significados de cada palavra, acrescenta-se a subjetividade inerente, e ainda a impossibilidade de cobrir toda a língua. Se catalogar todo o vocabulário já é por si só uma tarefa interminável, é ainda necessário lidar com a mutabilidade da linguagem natural, o que requer um trabalho constante de manutenção. Por exemplo, ao longo do tempo, novas palavras são inseridas no vocabulário, enquanto outras caem em desuso.

Para atenuar estes últimos problemas surgiram técnicas de indução do sentido das palavras (Nasiruddin, 2013), tarefa de processamento de linguagem natural (doravante PLN), que dado uma coleção de textos, procura identificar os significados/-contextos possíveis de uma determinada palavra. Uma forma de o fazer passa por procurar conjuntos de palavras através da sua co-ocorrência em frases ou documentos.

De modo a abordar a ambiguidade das palavras do ponto de vista computacional, é comum organizá-las em conjuntos que partilhem o mesmo significado, comummente designados por synsets, presentes em bases de conhecimento lexical, como as wordnets (Fellbaum, 1998). No caso de uma tarefa próxima e mais genérica, a desambiguação dos sentidos das palavras (Navigli, 2009), consiste no objetivo de determinar precisamente qual o sentido de uma palavra num contexto específico, através da sua associação a uma representação discreta, que pode ser um synset ou uma entrada num dicionário.

Porém, do ponto de vista linguístico, os sentidos das palavras não são discretos, nem podem ser separados com fronteiras bem definidas (Kilgarriff, 1996). Seguindo

essa visão, há palavras que não só pertencem a mais do que um synset, mas que poderão ter maior proximidade a alguns synsets do que a outros, e assim synsets em que diferentes palavras têm diferentes graus de pertença. Ou seja, a construção dos synsets de forma discreta é normalmente artificial.

Uma abordagem mais próxima da realidade seria a categorização dos synsets em forma difusa (fuzzy), ou seja, uma determinada palavra pertencer a um synset com um determinado grau. Por exemplo, a palavra **automóvel** poderá ter um maior grau de pertença ao synset que representa o significado de :"veículos com quatro rodas, com motor, com portas, que serve para transportar pessoas", do que a palavra veículo ao mesmo synset, uma vez que esta palavra não transmite exatamente o mesmo significado que a palavra carro, e por sua vez, carro e automóvel (presentes no synset referido) aparentam partilhar maior semelhança contextual. Assim, pretende-se criar um recurso para a língua portuguesa, onde se irá tirar partido de vários recursos incluindo dicionários. As relações extraídas através destes recursos formam um grafo de palavras, que será um dos materiais base do trabalho aqui apresentado.

Os objetivos principais do trabalho são, por um lado, estudar diferentes formas de obter este tipo de estruturas, a que podemos chamar fuzzy synsets e, por outro, tentar perceber até que ponto os graus de pertença podem ser utilizados como uma medida de confiança acerca da utilização de determinada palavra com o sentido transmitido pelo synset. Será futuramente integrado na abordagem de construção do **Onto.PT** (Gonçalo Oliveira and Gomes, 2014), que já possui os synsets, mas representados de forma discreta. Desta forma pretende-se criar uma futura versão deste recurso, onde esta informação esteja disponível, e que possa ser explorada pelos futuros utilizadores do recurso. Assim, serão aplicados neste projeto diversos algoritmos de clustering, com especial foco nos de fuzzy clustering.

Para a realização deste trabalho será desenvolvida uma aplicação que classifique, e organize as palavras em diversos fuzzy synsets, que se espera também poder vir a disponibilizar.

1.1 Motivação

Este trabalho tem como objetivo a criação de uma wordnet para a Língua Portuguesa, existindo já várias wordnets para a língua portuguesa, com graus de cobertura distintos entre si.No entanto existem alguns problemas associados às wordnets clássicas que se pretendem minimizar, por exemplo partindo de uma definição clássica de sinónimos "Duas expressões são sinónimas se num determinado contexto a substituição de uma por outra numa determinada expressão, não altera o valor lógico dessa mesma expressão"[1], ou seja, seguindo esta definição permite-se a criação de synsets cujas palavras partilhem exatamente os mesmos significados, e palavras que mesmo partilhando um grande grau de semelhança de significado possuem algumas "variações", sendo essas não desprezáveis, mas mesmo assim relevantes para se encontrarem no mesmo synset. Tomando como exemplo um excerto de um synset "criatura, animal, bixo, fera,", não existindo uma métrica que avalie a "intensidade"dessas mesmas "variações"para as wordnets disponíveis, atualmente apenas é possível avaliar se existe ou não relação de sinonímia e não o "grau de semelhança". A utilização deste tipo de métrica teria ainda outras vantagens: a

[1]Ver `http://www.instituto-camoes.pt/lextec/sobre.html` (Agosto 2015)

avaliação do grau de intensidade de uma relação, pois como referido anteriormente o domínio linguístico não possui fronteiras discretas (Kilgarriff, 1996). Por outro lado, o utilizador poderia escolher um grau de cobertura (maior tamanho vs maior precisão), sendo que as medidas associadas poderão ainda ser úteis para tarefas de desambiguação do sentido das palavras (Navigli, 2009). Deste modo acreditamos que a disponibilização deste recurso será útil e criará valor.

1.2 Contribuições

Deste trabalho resultaram as seguintes contribuições:

- Abordagem que tira partido de um algoritmo de hardclustering e strict partitioning de modo a permitir fuzzyclustering e overlaping clustering, que se adequa mais ao domínio do problema;
- Construção de datasets para avaliação manual, estes datasets são flexíveis podendo ser utilizados para avaliar outras abordagens;
- Esta tese que descreve e contextualiza uma abordagem que permite o agrupamento de palavras em synsets difusos;
- Artigo cientifico que resume as principais contribuições deste trabalho (Santos and Gonçalo Oliveira, 2015), aceite para publicação na edição de Dezembro 2015 da Linguamática – Revista para o Processamento Automático das Línguas Ibéricas[2].

1.3 Estrutura do documento

A informação presente neste documento está estruturada da seguinte forma:

- Neste capítulo foi feita uma contextualização introdutória ao domínio do problema e apresentada a nossa motivação;
- O capítulo 2 apresenta alguns conceitos base, necessários para uma melhor compreensão do trabalho aqui relatado;
- O capítulo 3 enumera alguns trabalhos que de alguma forma estão relacionados com a abordagem seguida neste trabalho;
- O capítulo 4 descreve os vários passos da abordagem proposta para a descoberta de synsets difusos a partir de redes de sinonímia;
- O capítulo 5 descreve algumas das experiências efetuadas e apresenta alguns dos seus resultados;
- O capítulo 6 resume as principais conclusões desta dissertação e linhas para trabalho futuro.

[2] `http://linguamatica.com/`

Capítulo 2

Conhecimento Prévio

Neste capítulo serão abordados alguns conhecimentos necessários para contextualizar o trabalho desta dissertação. Como tal, será feita uma breve explicação de conceitos acerca PLN, e no que consistem as wordnets. Após a explicação do domínio do problema, irão ser abordados conhecimentos acerca de grafos, visto que será a estrutura utilizada para representar as ligações entre palavras, e noções de técnicas de agrupamento (vulgo clustering).

2.1 Processamento de Linguagem Natural e Wordnets

O tema do processamento da linguagem natural (Natural Language Processing), descrito extensivamente por Jurafsky and Martin (2009), é muitas vezes apresentado com a ajuda de referências da cultura popular, onde *robots* são capazes de estabelecer e manter uma conversa com pessoas, usando linguagem humana. Estas visões são frequentemente representadas por personagens de filmes e séries, tais como HAL9000 em the Stanley Kubrick's classic *2001: A Space Odyssey*[1].

O PLN é uma componente da Inteligência Artificial (AI, Russell and Norvig (1995)) cujo maior objetivo é permitir às máquinas perceber a linguagem das pessoas e, consequentemente comunicar connosco, na nossa própria linguagem, como se as máquinas fossem pessoas. Posto isto, a linguagem natural, usada pelos humanos na sua comunicação, é provavelmente a forma mais natural de codificar, transmitir e raciocinar sobre o conhecimento. A maior parte dos repositórios e bases de conhecimento estão escritos desta forma (Santos, 1992). Como tal, a importância do campo na Inteligência Artificial não é de todo surpreendente.

Como já referido, um dos maiores problemas referentes à linguagem natural é que esta difere de linguagens formais,como por exemplo as linguagens de programação, pois nestas últimas, a cada letra ou cada símbolo, é frequente que apenas corresponda apenas um único sentido sendo que nos poucos casos em que um símbolo tem vários significados possíveis as regras de desambiguação de uma linguagem ser claras.

A ambiguidade acontece quando não é possível determinar um único significado possível a uma determinada forma de comunicação, consequentemente esta pode ser interpretada de várias formas. Enquanto os humanos conseguem inferir de forma natural o significado e o contexto das palavras, para os computadores a tarefa é muito

[1] Ver `http://www.imdb.com/title/tt0062622/` (Agosto 2015)

mais complexa. Os computadores necessitam de transformar o texto em estruturas de dados, que em seguida devem ser analisadas para inferir o seu significado. Para uma máquina conseguir determinar o significado de cada palavra necessita de ter acesso a um inventário repositório de significados/sentidos/contextos (Navigli, 2009).

Uma wordnet é uma base de conhecimento normalmente utilizada como inventário de sentidos.

2.2 Wordnets

As wordnets podem ser utilizadas para desambiguação e significados de uma palavra, *information retrieval*, classificação automática de texto, sumarização automática de texto, tradução automática entre outros (Snow et al., 2007).

2.2.1 Estrutura

As wordnets comummente dividem as palavras em quatro grupos principais (Nomes, Verbos, Adjetivos e Advérbios), ignorando as preposições e determinantes, provavelmente por estas serem classes fechadas e serem palavras de função, ou seja, sem significado por si só quando isoladas.

Numa wordnet é comum guardar a informação em grupos de sinonímia chamados de synsets. Cada synset conecta-se a outros synsets através de vários tipos de relações conceituais, que vêm normalmente acompanhados de uma definição do seu significado e, por vezes, também de frases ilustrativas da utilização das suas palavras.

Abaixo segue-se uma explicação sobre algumas propriedades e relações presentes em Wordnets.

As palavras possuem relações formais entre si, quer ao nível do som produzido, do grafismo, e do nível semântico. Sendo este trabalho mais focado no contexto e significado das palavras será dado mais importância às relações semânticas onde se destacam as seguintes:

Hiperonímia e Hiponominia

A hiperonímia é um tipo de relação de ordenação hierárquica sendo que o hiperónimo é o conceito mais abrangente e hipónimo a relação inversa da hiperonímia, ou seja são as palavras que pertencem ao hiperónimo correspondente. A hiperónimia tem um papel muito importante na definição das palavras. A definição é uma perífrase sinónima da palavra que se define. A **hiponominia** é a relação inversa da hiperonímia, sendo que hipónimos são as palavras que pertecem à classe do hiperónimo correspondente. Por exemplo, carro é um hipónimo de veículo e veículo é hiperónimo de carro.

Holonímia e Meronímia

A holonímia é a relação de inclusão semântica entre duas unidades lexicais, uma denota um todo (holónimo) sem impor obrigatoriamente as suas prioridades semânticas à outra, considerada sua parte, já a relação inversa, a meronímia é uma relação de inclusão semântica entre duas unidades lexicais, uma denotando a parte (merónimo)

e criando uma relação de dependência ao implicar a referência a um todo (holónimo), relativo a essa parte. Por exemplo roda é merónimo de carro e roda é holónimo de pneu.

Antonímia

Relação entre duas palavras que transmite uma ideia de oposição de significado semântico. Uma vez que as palavras pertencentes ao mesmo synset partilham o seu significado, partilham também as relações semânticas entre elas. Por exemplo *bom* e *mau* são palavras antónimas uma da outra.

Existem ainda muitas outras relações entre os synsets, mas são menos utilizadas que as acima descritas.

2.2.2 Criação de wordnets

Historicamente a primeira wordnet, a WordNet de Princeton, é um projeto iniciado em 1985 no Laboratório da Ciência cognitiva da Universidade de Princeton sob a direção do professor George Armitage Mille Miller (1995); Fellbaum (1998).

Como as wordnets são tradicionalmente criadas e mantidas de forma manual, a sua cobertura nem sempre é a desejada para todas as tarefas. Assim, alguns investigadores tiveram que recorrer a técnicas automáticas de extração de relações semânticas a partir de texto, que exploram as regularidades em que palavras relacionadas ocorrem. Entre esses trabalhos, destacam-se as abordagens baseadas em padrões (Hearst, 1992), abordagens supervisionadas (Snow et al., 2005), e ainda abordagens levemente supervisionadas (Pantel and Pennacchiotti (2006)). Este tipo de trabalho também foi feito para o português (ver vários trabalhos em (Collovini de Abreu et al., 2013), inclusivamente no CISUC, onde foram utilizados padrões para extrair relações de dicionários (Gonçalo Oliveira et al., 2011) e da Wikipédia (Gonçalo Oliveira et al., 2010).

Segundo a ***The Global WordNet Association***[2] existem várias wordnets para a língua portuguesa, entre as quais:

- Wordnet.PT (Marrafa et al., 2011)
- MultiWordNet.PT[3]
- OpenWordNet-PT (de Paiva et al., 2012)
- Onto.PT (Gonçalo Oliveira and Gomes, 2014)

2.2.3 Onto.PT

A **Onto.PT** foi elaborada no **Centro de Informática e Sistemas da Universidade de Coimbra**, no âmbito do doutoramento do Professor Hugo Gonçalo Oliveira, com a orientação do Professor Paulo Gomes. No início do seu desenvolvimento, já existia a WordNet.PT que é desenvolvida de forma manual e não estava, nem está, disponível para outro tipo de utilização que não uma mera busca na sua

[2] `http://globalwordnet.org/wordnets-in-the-world/`
[3] `mwnpt.di.fc.ul.pt`

página *web*. Por outro lado, a Onto.PT não só tem utilização livre, como é criada de forma automática, sendo assim maior que a WordNet.PT e as outras wordnets da língua portuguesa, tanto em termos de palavras, como synsets e relações semânticas.

Para o desenvolvimento da Onto.PT, o autor utilizou vários dicionários de língua portuguesa, explorados através do modelo ECO (Gonçalo Oliveira and Gomes, 2014), que consiste em 3 passos:

- Extração: em que são extraídas relações entre palavras, com base em padrões elaborados previamente;
- *Clustering*: em que são identificados grupos de palavras que partilham o mesmo significado;
- Ontologisação: em que são identificados os synsets alvo para o argumento de cada relação.

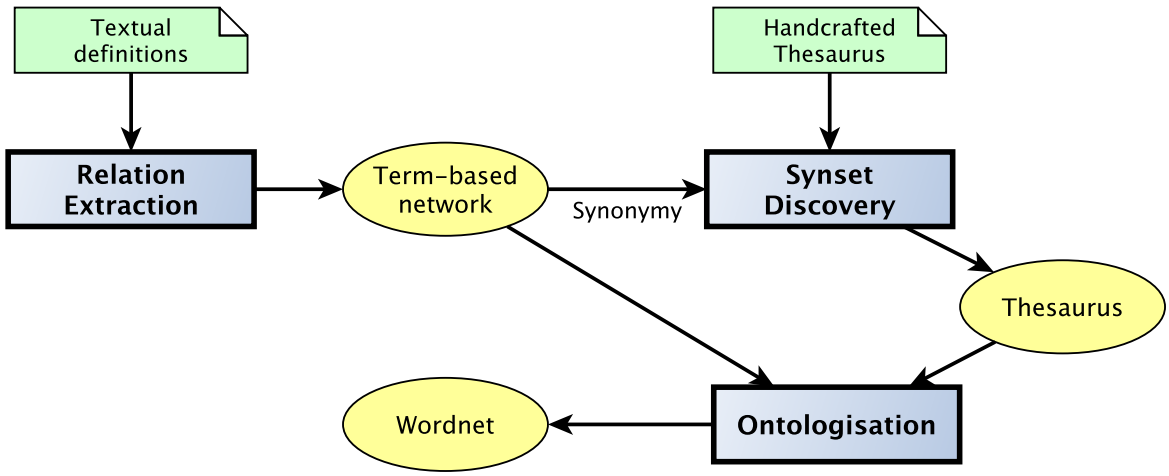

Figura 2.1: Diagrama da abordagem ECO

Ao contrário de Lin (1998) e outros, que exploram o contexto onde as palavras estão inseridas, no Onto.PT o contexto não é explorado durante a fase de extração o que permite uma maior flexibilidade, sendo assim mais fácil integrar vários recursos (dicionários, *thesaurus*, etc). Ao invés de se agrupar as palavras por similaridade de contexto, o autor opta por agrupar as palavras por similaridade de sinónimos. Devido a esta abordagem, os agrupamentos obtidos partilham o mesmo significado (synsets) ao invés de palavras utilizadas no mesmo contexto (Gonçalo Oliveira and Gomes, 2011).

Como já referido, do ponto de vista linguístico, o agrupamento de sinónimos de forma discreta é artificial. Para resolver este problema, o autor utilizou na fase clustering uma representação difusa para os synsets (Gonçalo Oliveira and Gomes, 2011). Porém, esta abordagem é bastante simplista e, no final, as pertenças são mesmo descartadas. De modo a mitigar este problema, a abordagem aqui apresentada foca-se primordialmente na segunda fase, a fase de clustering.

2.3 Grafos

Formalmente, um grafo é um conjunto ordenado de pares $G = (V, A)$, em que V é um conjunto de vértices, A é um conjunto de arestas, e em que cada aresta liga dois vértices, $A \subset V^2$. Se for possível deslocar-se do nó A para o nó B através de pelo menos um caminho de arestas diz-se que nó A e o nó B estão na mesma componente conectada, sendo que um grafo pode possuir uma ou mais componentes conectadas.

Muitos problemas podem ser resolvidos com o recurso a grafos, por exemplo, otimização de rotas, cálculo de distâncias entre locais, fluxo numa rede informática, etc.

2.3.1 Tipos de grafos

Pode-se dividir os grafos em várias categorias entre as quais:

- **Grafos Simples**: existe no máximo uma aresta entre cada par de nós, por exemplo, a aresta entre os nós x e y pode representar-se por $A(x, y)$;
- **Multi Grafos**: permitem mais do que uma aresta entre cada par de nós;
- **Grafos direcionados**: as arestas possuem uma direção associada, ou seja, a aresta $A(x, y)$ é diferente de $A(y, x)$;
- **Grafos com pesos**: as arestas têm um peso numérico associado, ou seja, uma aresta passa a ser um triplo $A(x, y, p)$.

Um grafo pode possuir uma ou mais das propriedades acima descritas.

2.3.2 Operações e propriedades sobre grafos

Coeficiente de clustering

Uma métrica que se pode aplicar a grafos e a subgrafos consiste no coeficiente de clustering também conhecida como transitividade, tendo em conta que esta métrica avalia a intercomunicação entre os vizinhos de um nó, ou seja se todos os vizinhos de A estiverem ligados entre eles o resultado do coeficiente de clustering para o nó A é 1, se nenhum dos vizinhos de A possuir ligações a nenhum vizinho de A então o coeficiente de clustering é 0. Na figura 2.2 é possível visualizar a disposição de alguns exemplos de subgrafos com diferentes valores de coeficientes de cluster

Para calcular o coeficiente de clustering é necessário identificar trios de nós e categoriza-los como trios abertos caso existam duas ligações dentro do trio, ou como trios fechados caso existam três ligações entre os nós desse trio. Sendo que a equação 2.1 permite calcular o coeficiente de clustering para um dado grafo.

$$C = \frac{Trios\ Fechados}{Trios\ fechados + Trios\ abertos} \tag{2.1}$$

Densidade do grafo

Um grafo diz-se completo (densidade =1) se entre quaisquer dois nós existir uma ligação. A densidade de um grafo é a razão entre a quantidade de arestas do grafo

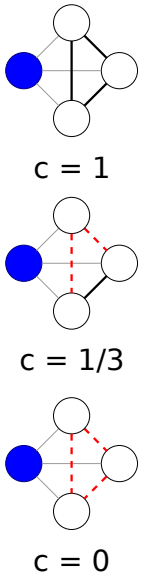

Figura 2.2: Exemplos de subgrafos com diferentes valores de coeficientes de cluster

e a quantidade de arestas do grafo completo com o mesma quantidade de vértices. Para N nós e V vértices a equação que permite calcular a densidade D de um grafo não direcionado

$$D = \frac{2\,|V|}{|N|\,(|N|-1)} \tag{2.2}$$

Sendo que para um grafo direcionado a equação é

$$D = \frac{|V|}{|N|\,(|N|-1)} \tag{2.3}$$

Ponto de corte

Para dividir um grafo em sub-grafos é utilizada a técnica de corte. A métrica "tamanho do corte"é definida pela quantidade de arestas que conectam vértices de um sub-grafo ao outro sub-grafo. Pretende-se que entre clusters esta métrica tenha tamanho reduzido.

Matriz de adjacências

Pode-se representar um grafo através de uma matriz de adjacências que é descritiva das ligações, ou seja quando existe ligação coloca-se um valor diferente de 0. No caso de grafos sem pesos coloca-se por norma 1, em grafos com pesos o valor do peso, e o valor 0 para quando não existe ligações (em ambos os casos referidos) (Schaeffer, 2007). Um exemplo de um grafo está representado na figura 2.3 com a respectiva matriz de adjacencias na tabela 2.1.

2.4 Clustering

O clustering é uma técnica de aprendizagem não supervisionada. Os métodos de aprendizagem não supervisionados diferem dos supervisionados na medida em que

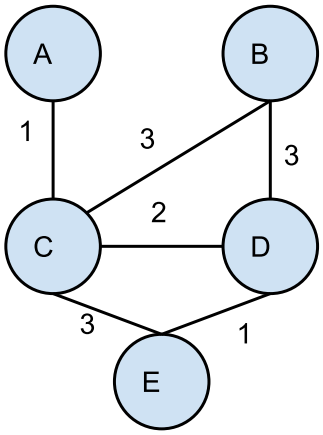

Figura 2.3: Exemplo de um grafo

	A	**B**	**C**	**D**	**E**
A		0	1	0	0
B	0		3	3	0
C	1	3		2	3
D	0	3	2		1
E	0	0	3	1	

Tabela 2.1: Tabela de adjacências da figura 2.3

classificam os dados com base apenas em regularidades identificadas nas características dos seus elementos. Por outro lado, nos métodos de aprendizagem supervisionada existe um conjunto de dados de treino onde estão definidas as várias classes de dados e de onde é possível analisar características comuns aos elementos da mesma classe.

O clustering é aplicado muitas vezes no dia-a-dia das pessoas. Por exemplo, no vestuário seria impensável haver um tamanho único para todas as pessoas, porque esse tamanho não iria servir a um elevado número de pessoas, e ao mesmo tempo não seria economicamente viável fabricar toda a roupa à medida de cada pessoa. Para resolver esta questão foram encontrados os "clusters"neste caso os tamanhos (***Small***, ***Medium***, ***Large***, etc).

O objetivo principal dos métodos de clustering é identificar grupos (ou classes) de instâncias/objetos num determinado universo. Idealmente cada grupo deve possuir apenas indivíduos semelhantes entre si, e os indivíduos em grupos distintos devem ser pouco semelhantes. Os algoritmos de clustering podem dividir-se em diferentes grupos principais, os hierárquicos e os de partição simples. Caso os elementos possuam uma métrica que avalie o nível de pertença ao seu grupo designa-se a essa característica de fuzzy clustering.

Apresentam-se de seguida exemplos de algoritmos que se enquadram em cada um dos grupos.

2.4.1 Hierárquicos

Os algoritmos de clustering hierárquico produzem uma hierarquia de clusters, ou seja, cada cluster pode possuir vários sub-clusters. Cada um deles é um filho, e

quando o cluster não possui nenhum filho, este é denominado de folha. A estrutura aqui descrita tem o nome de dendograma, um exemplo dessa estrutura está representado na figura 2.4 [4]

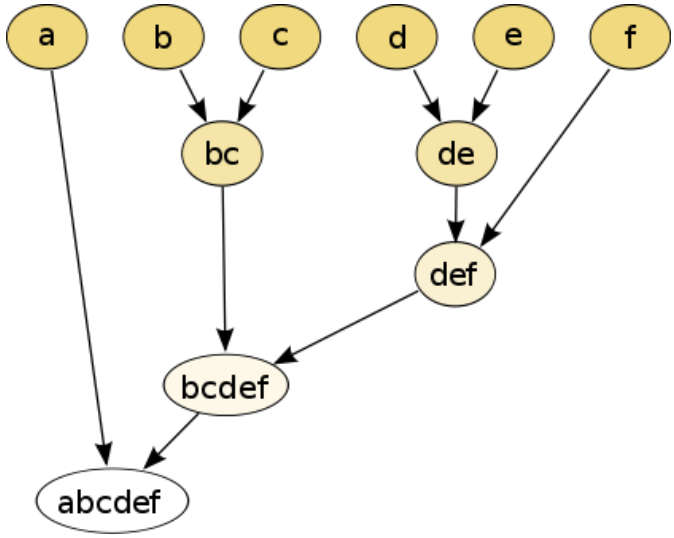

Figura 2.4: Exemplo de um dendograma.

Dentro dos algoritmos de clustering hierárquico existem duas categorias principais de acordo com a hierarquia gerada:

- Algorimos aglomerativos geram uma hierarquia por agrupamentos de grupos mais pequenos em grupos maiores;
- Algoritmos divisivos geram uma hierarquia por divisão de grupos maiores em grupos mais pequenos (algoritmos divisivos).

Um dos algoritmos mais utilizados como base para outras variantes é o **Complete-Link**. Este algoritmo é aglomerativo, ou seja, o sistema inicia-se categorizando cada instância como um cluster. Uma caraterística deste algoritmo para o cálculo das distâncias entre clusters, é a distância menor de todos os elementos do cluster A com todos os elementos do cluster B.

O pseudocódigo para a implementação do algoritmo é:

1. Considerar cada instância como um cluster;
2. Para todos os pares de instâncias calcular as respectivas distâncias;
3. Ordenar a lista de distâncias por ordem crescente;
4. Agrupar no grafo o par com menor distância;
5. Repetir até todas as instâncias ficarem num grafo conexo.

O algoritmo Complete-Link tem uma abordagem semelhante, mas o método de cálculo de distância é feito através do par com a distância maior ao invés da distância menor.

[4]Imagem disponivel em `https://en.wikipedia.org/wiki/Dendrogram#/media/File:Hierarchical_clustering_simple_diagram.svg`

2.4.2 Partição simples

O objetivo dos métodos de Partição é particionar os dados em conjuntos que maximizem a homogeneidade interna e/ou heterogeneidade externa. Em muitos algoritmos de particionamento (não todos) é necessário definir *a priori* o número de grupos. Os algoritmos de partição, tipicamente tem como objetivo encontrar centros de modo a que cada objeto pertença ao centro mais próximo deste. Neste grupo, destacam-se o K-Means e o Potencial Subtrativo.

K-Means

Antes de mais, os algoritmos *K-means* têm uma grande desvantagem no âmbito deste problema, que é o facto de ser necessário introduzir o número de clusters. O algoritmo K-means de uma forma concisa consiste em escolher aleatoriamente k pontos, e em seguida verificar em cada instância qual o ponto k (clusters) que está mais próximo. Em seguida é necessário calcular as coordenadas do centro dos vários clusters, para os pontos k deslocarem-se até esse mesmo centro. Repete-se estes dois últimos passos até o sistema estabilizar, ou seja, os pontos não se movimentarem entre duas iterações.

Potencial Subtrativo

Neste algoritmo existe a vantagem de não ser necessário indicar o número de clusters. O algoritmo consite nos seguintes passos:

1. Considerar um ponto nas coordenadas de cada instância;
2. Calcular o potencial recebido em cada ponto "emanado" pelas outras instâncias;
3. As coordenadas do primeiro ponto serão as coordenadas do ponto com maior potencial ;
4. Para evitar a concentração dos centros é necessário retirar o potencial dos pontos que estão próximos dos centros já calculados no ponto 3 ;
5. Calcular o ponto com maior potencial (depois de reduzidos os potencias no ponto 4) ;
6. Repetir 3, 4, 5 até um critério de paragem (número de clusters, potencial abaixo de um valor, etc).

Este algoritmo não classifica as instâncias, apenas dá os pontos para clustering. Posto isto é possível forçar que cada instância pertença a um, e a só um cluster, ou pelo contrário fazer com que as instância pertençam aos clusters com graus de pertença (conforme a distância ao centro).

2.4.3 Fuzzy

Os algoritmos de fuzzy clustering diferem dos métodos de partição simples na medida em que não se limita o número de classes a que uma instância pode pertencer. Nos métodos Fuzzy instancia i pertence às classes C_j com probabilidade P_{ij}. Neste

grupo, destaca-se o algoritmo Fuzzy C-Means (Doravante FCM), que é muito semelhante ao K-means descrito em **Partição**. A diferença é que no Fuzzy C-means todas as instâncias fazem parte dos vários clusters, mas os centros são calculados com base nas pertenças, ou seja, as instâncias que se encontram mais longe de determinado centro, fazem mover de maneira menos significativa.

Dado um conjunto finito de dados, com N elementos em $X = [x1, x2, x3, ...xn]$ e C centros de clusters c $C = [c1, c2, c3, ..., cc]$ e uma matriz de partição em que $W = w[i, j]$ onde cada elemento $w[i, j]$ apresenta o grau de pertença que a instância Xi tem ao cluster cj

A função objetivo a minimizar é a seguinte

$$\sum_{i=1}^{n}\sum_{j=1}^{c} w_{ij}^{m} \left\|\mathbf{x}_i - \mathbf{c}_j\right\|^2 \tag{2.4}$$

A função que calcula a pertença da instância i ao cluster c é a seguinte

$$w_{ij}^{m} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|\mathbf{x}_i - \mathbf{c}_j\|}{\|\mathbf{x}_i - \mathbf{c}_k\|}\right)^{\frac{2}{m-1}}}. \tag{2.5}$$

O parâmetro m permite manipular o algoritmo de modo a fazê-lo tender para uma abordagem mais discreta (a pertença tende a dar 0 ou 1 quando m tem um valor que se aproxime de 1), ou mais difusa (menores pertenças mas mais clusters onde cada instância pertence de forma proporcional ao tamanho m)

Capítulo 3

Trabalho Relacionado

Neste capítulo serão apresentados alguns exemplos de trabalho relacionado nomeadamente acerca de clustering sobre palavras, clustering sobre grafos e por fim clustering em grafos palavras. São ainda analisadas várias ferramentas de construção/manipulação/visualização de grafos.

A desambiguação e deteção de significados é um problema em aberto. Encontra-se bem definido com precisão, mas no entanto ainda não foi descoberta uma solução ótima, de processamento de linguagem natural. Tomando como exemplo a palavra "Canto", esta poderá estar associada a significados diferentes (Figura 3.1). Assim, na frase "O **canto** do pássaro é melodioso"e "O extintor está no **canto** da parede", o ser humano consegue distinguir o seu significado em cada uma destas frases com facilidade, mas uma máquina não.

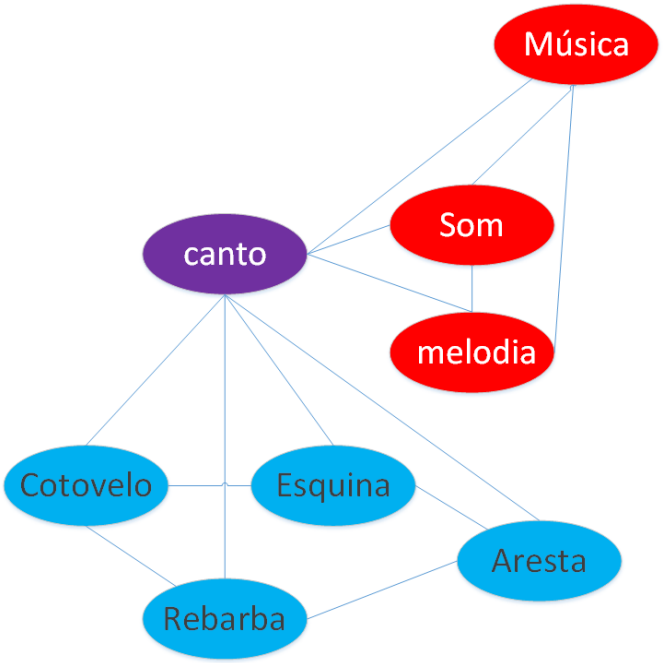

Figura 3.1: Exemplo de polissemia na palavra "Canto"

3.1 Clustering sobre palavras

A hipótese distribucional parte do principio que as palavras que ocorrem nos mesmos contextos tendem a ter significados semelhantes (Harris, 1954). A ideia subjacente de que "uma palavra é caracterizada pela sua companhia" foi popularizado por Firth (1957).

Os modelos de distribuição semântica são modelos computacionais que transformam a hipótese distribucional numa estrutura experimental para analise semântica. Para a criação dessa mesma estrutura tira-se partido da co-ocorrência das palavras, sendo que para cada palavra é criado um vector com o contador de cada co-ocorrência.

No entanto, a similaridade nesta abordagem, não distingue os tipos de relações, ou seja, por exemplo gato é semelhante a cão e a animal, no entanto animal é hipónimo de gato, e cão é co-hipónimo de gato, ou seja homónimos que partilham um mesmo hiperónimo, nas wordnets tradicionais, como por exemplo no Onto.PT, onde as relações são tipificadas.

De seguida seguem-se algumas abordagens de clustering sobre palavras que se enquadram no conceito de hipótese distribucional

Latent semantic analysis (**LSA**, Deerwester et al. (1990)) é uma técnica matemática/estatística totalmente automática usada para extrair, e inferir relações de uso contextual, extraídas através de palavras contidas em excertos de um discurso. Não utiliza nenhum conhecimento humano prévio, por exemplo, dicionários, bases de conhecimento, redes semânticas etc. Consiste na construção de uma matriz, em que cada linha corresponde a uma palavra, e cada coluna corresponde à identificação de um excerto textual, sendo que cada célula representa a frequência de vezes que a palavra apareceu naquele excerto. É utilizada a técnica matemática decomposição em valores singulares para reduzir o tamanho da matriz, reduzindo o número de linhas, mas mantendo a estrutura de similaridade entre as colunas. A técnica de rank lowering também poderá ser utilizada, esperando mitigar-se o problema de sinonímia, visto que, com o rank lowering é esperado por exemplo, que as dimensões associadas a termos que tenham o mesmo significado se juntem.

Existem algumas limitações do algoritmo LSA entre as quais: não consegue detetar a polissemia, dado que, para o algoritmo a palavra canto terá o mesmo "significado"na frase "O **canto** do pássaro é melodioso", e na frase "O extintor está no **canto** da parede". Para o algoritmo, o vetor representará a média dos significados, ou seja, se o significado ocorrer de forma dominante no conjunto de texto será esse o significado representado no vetor. As dimensões resultantes poderão ser justificáveis matematicamente, porém não têm um significado interpretável em linguagem natural.

Pointwise Mutual Information (**PMI**, Turney (2001)) é uma outra métrica, denominada de PMI-IR - Pointwise Mutual Information. Baseada na co-ocorrência, sendo que esta medida é obtida através de medidas probabilísticas baseadas na ocorrência das palavras com que se pretende calcular similaridade, por exemplo, a seguinte equação:

$$score(choice) = \frac{\mathbf{p}(problem\&choice)}{\mathbf{p}(choice)}$$

Calcula a similaridade entre as palavras *choice* e *problem* consoante o número de vezes que elas ocorrem no texto juntas e, o número de vezes em que a palavra *choice* ocorre isolada da palavra *problem*. Existem outras variações deste cálculo probabilístico que podem ser utilizadas.

Dekang Lin opta por explorar a existência de triplos num texto (duas palavras e uma relação gramatical entre elas - ||w,r,w'||), efetuando uma contagem sobre os triplos existentes no texto (Lin, 1998). De seguida são cálculadas métricas de semelhança entre os vários componentes de cada triplo, de modo a encontrar pares de sinonímia.

Clustering by committee (Lin and Pantel, 2002), doravante CBC, é um algoritmo de clustering de partição que consiste em três fases:

1. Construir uma matriz que represente a similaridade entre os pares de palavras possíveis, verificando quais deles têm mais similaridade;

2. Detetar os vários "candidatos"a clusters aqui denominados por *committees*. Os elementos pertencem a um *comittee* quando o grau de semelhança é maior que um determinado valor (threshold);

3. Atribuir cada elemento (palavra/lema) ao cluster com mais similaridade. Quando se atribui um cluster a um elemento são retiradas as caraterísticas de sobreposição entre o elemento e o cluster. Deste modo pretende-se descobrir os vários significados de uma palavra, evitando repetição. Por exemplo, quando se atribui ao cluster instituição financeira a palavra banco, retira-se à palavra as caraterísticas em comum da palavra com o cluster. Deste modo em teoria ficarão presentes as caraterísticas que transmitem o significado de assento.

Pode fazer-se um paralelismo com o K-means na medida em que os objetos pertencem à classe mais próxima, e com o algoritmo denominado de potencial subtrativo, visto que, quando se atribui um objeto a um comittee são retiradas as caraterísticas do comittee ao objeto inserido. Porém, este algoritmo difere do K-means na medida em que, no CBC não é necessário especificar o número de clusters. Os centroides dos clusters não mudam quando lhes são atribuídos novos elementos.

3.2 *Clustering* em grafos

A estrutura das ligações de um grafo é conhecido como a sua estrutura. A própria estrutura dos grafos muitas vezes contêm informação útil para o domínio do problema que este representa. Segue-se nas figuras 3.2, 3.3, 3.4, 3.5 e 3.6 alguns exemplos de representação várias estruturas presentes em grafos na

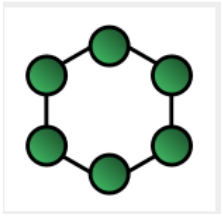

Figura 3.2: Estrutura em anel

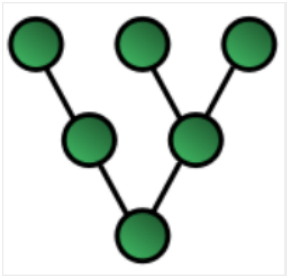

Figura 3.3: Estrutura em árvore

Figura 3.4: Estrutura em estrela

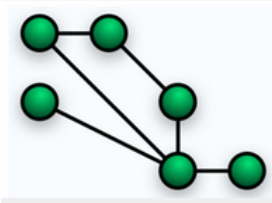

Figura 3.5: Estrutura em malha

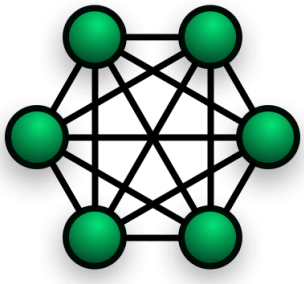

Figura 3.6: Estrutura completamente conectada

Se considerarmos cada um destes exemplos como um cluster, visualmente

percebe-se que estas diferentes estruturas possuem coeficientes de clustering distintos, sendo a estrutura com coeficiente de clustering mais elevado o completamente conectado e como tal o agrupamento mais óbvio.

Uma maneira mais fácil de perceber o conceito de agrupamentos aplicados a grafos é a perceção de como funcionam os círculos sociais dos mais diversos indivíduos. Se representarmos cada pessoa como um nó de um grafo e a relação entre duas pessoas uma ligação entre os respetivos nós, é fácil de verificar a existência de vários círculos sociais. Ou seja, imaginando o indivíduo A, as suas relações e as relações entre os indivíduos que A conhece é possível visualizar agrupamentos como "família" "colegas de trabalho" "colegas de *hobbies*" etc.

A título de exemplo é aqui apresentado (na figura 3.7) um grafo de uma da rede social presente em "Les Miserables" de Victor Hugo, onde cada agrupamento/circulo social é apresentado com uma cor distinta[1].

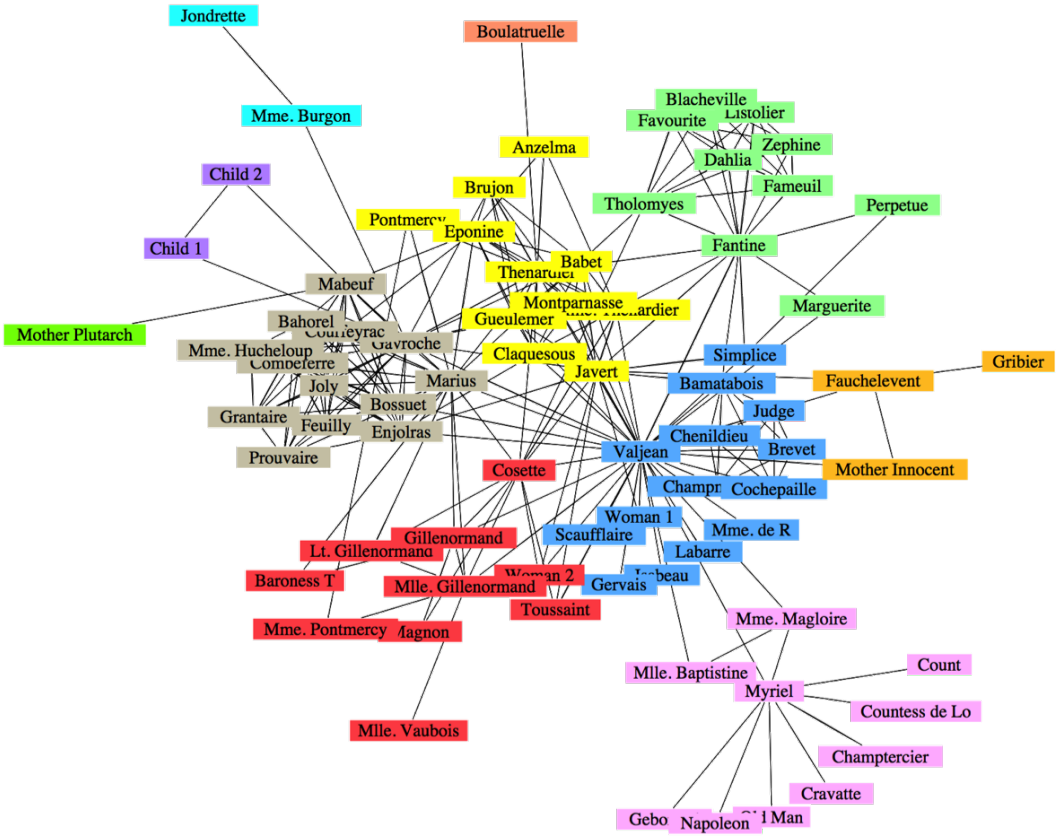

Figura 3.7: Exemplo de clustering sobre o grafo de uma rede social.

Um grafo pode ser representado através de um conjunto de triplos, triplos esses compostos por dois nós e uma relação de ligação que os relaciona, no nosso caso, os nós são palavras, e as ligações são as relações existentes entre elas. Este ponto está devidamente detalhado no capítulo 4. Deste modo achou-se pertinente o estudo sobre os agrupamentos em grafos, sendo analisados neste capítulo alguns agrupamentos, tirando-se já algumas conclusões relacionadas com o domínio do objetivo

[1]Imagem adaptada de `https://punkrockor.wordpress.com/2011/07/27/five-nifty-social-networks`

pretendido neste trabalho. Nas subsecções seguintes são abordados dois métodos de clustering sobre grafos, o Markov Clustering e o Chinese Whispers.

3.2.1 Markov Clustering

Antes de aplicar qualquer algoritmo de clustering é possível ordenar a matriz de adjacências, pois torna-se mais visível a existência de clusters, além de que existem vários algoritmos de clustering que tomam partido da pré-ordenação para diminuir o tempo de processamento.

O Markov Clustering (van Dongen, 2000) (doravante MCL) é um algoritmo de clustering baseado em random walks. Esta formalização matemática defende que, como existem muito mais ligações dentro de um cluster se se escolher aleatoriamente um caminho de um determinado nó do grafo, a probabilidade de ficar dentro do cluster pertencente a esse nó, é muito maior do que sair fora desse mesmo cluster.

Na figura 3.8 está representado um exemplo de um grafo com a respectiva matriz probabilística na tabela 3.1.
Aplicando o conceito acima descrito, quando se executam random walks é possível verificar em que pontos do grafo existe convergência, pontos esses candidatos a clusters. random walks podem ser calculados usando Markov Chains (cadeias de Markov).

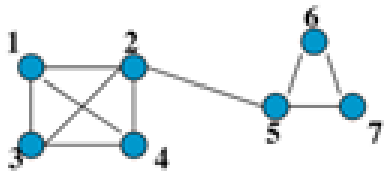

Figura 3.8: Exemplo de um grafo

	1	2	3	4	5	6	7
1	0	0.25	0.33	0.33	0	0	0
2	0.33	0	0.33	0.33	0.33	0	0
3	0.33	0.25	0	0.33	0	0	0
4	0.33	0.25	0.33	0	0	0	0
5	0	0.25	0	0	0	0.5	0.5
6	0	0	0	0	0.33	0	0.5
7	0	0	0	0	0.33	0.5	0

Tabela 3.1: Tabela de transição/probabilística

Segundo a cadeia de Markov de primeira ordem, a matriz probabilística representa a probabilidade do estado presente. O passado e o futuro são probabilisticamente independentes, ou seja, a probabilidade para o estado seguinte depende apenas do estado presente.

Quando as ligações dentro do grafo têm pesos é necessário normalizar a matriz de probabilidade. Para isso pode-se dividir o valor correspondente pela soma total das ligações que saem de determinado nó.

O fluxo dentro do grafo é mais "fácil" ocorrer em áreas densas, que em áreas esparsas no entanto esse efeito é reduzido ao longo das iterações. Como tal, analisando a matriz podemos verificar ao longo das colunas uma relação desses valores com os clusters: onde estão concentrados os valores mais altos há uma grande probabilidade de ser um cluster.

Para aumentar esse efeito, poderá ser usada uma técnica chamada "Inflação", que consiste em elevar os valores de uma determinada coluna a um valor não negativo, e de seguida voltar a normalizar. Com isto acentuam-se os valores fortes, e enfraquecem-se os valores fracos. Tecnicamente, o parâmetro r da "Inflação"pode influenciar a granularidade dos clusters. Existe outra técnica chamada "Expansão" que consiste em expandir a matrix n vezes. A expansão é responsável por permitir o "fluxo" para conectar diferentes zonas do grafo.

Uma versão simplificada do algoritmo de Markov é:

1. Acesso ao grafo com ligações entre os nós;
2. Criar a matriz associada as ligações entre os nós;
3. Normalizar a matriz;
4. Expandir a matriz utilizando um valor previamente definido;
5. "Inflacionar" a matriz utilizando um valor previamente definido;
6. Repetir 4 e 5 alternadamente até convergir, ou seja os valores não se alterarem;
7. Analisar a matriz resultante para ponderar os resultados.

Por vezes a "Inflação" e a "Expansão" anulam-se uma à outra. De forma a resolver essa situação é necessário modificar ligeiramente o valor de um ou dos dois parâmetros.

O grafo irá então ser dividido em duas categorias: Os "atraidores", que vão atrair outros vértices, e os vértices que serão atraídos. Cada "atraidor"tem que ter pelo menos um valor positivo na sua linha, e os atraídos valores positivos nas mesmas colunas que o "atraiador".

Segundo o algoritmo aqui apresentado, só é possível identificar overlapping clusters, quando os vários clusters de um vértice o "atraem"de igual modo, e quando são isomórficos. Ou seja, espera-se que neste projeto, este algoritmo não vá ter um bom desempenho a detetar os Overlaping clusters, visto que, apenas os identifica em casos muito específicos.

O custo temporal do MCL para n vértices é de $O(n^3)$ na fase de "expansão"e de $O(n^2)$ na fase de "inflação".

O prunning da matriz neste caso transformar os valores residuais em 0 reduz bastante os tempos de cálculo, visto que os zeros não sofrem alterações durante as várias iterações deste algoritmo.

O número de iterações necessárias para que a matriz convirja, ainda não está matematicamente provado, no entanto é frequente essa ocorrência entre 10 a 100 iterações (entre "Inflação"e "Expansão").[2].

Este algoritmo classificasse como hard-clustering e strict partioning clustering, sendo que não é necessário indicar o número de agrupamentos

[2]Ver `https://www.cs.ucsb.edu/~xyan/classes/CS595D-2009winter/MCL_Presentation2.pdf` (Agosto 2015)

3.2.2 Chinese Whispers

O algoritmo denominado por Chinese Whispers (Biemann, 2006) (doravante CW) baseia-se no popular jogo do Telefone Estragado, jogo em que num grupo de pessoas é designada uma, para criar uma frase e sussurrando-a à pessoa seguinte, pessoa essa que repete o processo, até à ultima pessoa do grupo ouvir a mensagem que a repetirá desta vez em voz alta. Devido a mensagem passar por vários "canais"com ruído (a mensagem é segredada e muitas vezes o recetor não ouve com clareza o que foi dito, sendo deste modo propagado e aumentado o erro a cada passagem) a mensagem vai sendo transformada ao longo das iterações. O algoritmo de CW baseia-se, tal como o MCL em random walks. Inicialmente, o seu funcionamento consiste na atribuição de uma classe distinta a cada nó de um dado grafo. Por cada iteração do algoritmo é atribuído a cada nó a classe cujos vizinhos lhe transmitam a maior "força". Essa "força"é medida através da soma dos pesos de cada classe dos nós conectados, sendo executado diversas vezes até se encontrar um ponto de estabilização. Este algoritmo à semelhança do já analisado MCL classifica-se como hard-clustering e strict partioning clustering, sendo que mais uma vez não é necessário indicar o número de agrupamentos.

Um facto interessante e relevante para o domínio deste trabalho é que o CW sendo um algoritmo de graph clustering que pode ser aplicado a problemas genérico, o autor explora as co-ocorrências presentes em vários tipos de textos de modo a construir um grafo, onde aplica o algoritmo Chinese Whisperers para agrupar as palavras em synsets. O autor constrói ainda um grafo de segunda ordem de modo a agrupar as palavras em classes (ex nomes próprios, nomes de plantas, nomes de profissões, etc)

3.3 Manipulação de Grafos

Os nós de um grafo podem representar diferentes entidades, inclusivamente palavras, podendo as arestas representar relações semânticas ou de co-ocorrência entre palavras. Tratando-se de um grafo, as técnicas de graph clustering são também aplicáveis. Destacam-se alguns trabalhos em que foram aplicadas técnicas deste tipo a grafos de palavras:

- Biemann (2006) como já referido o autor do algoritmo CW, aplicou o seu algoritmo ao domínio de PLN aplicando-o a um grafo de co-ocorrências;

- Gfeller et al. (2005) utiliza um dicionário de sinónimos para construir um grafo. Após a sua construção é aplicado o algoritmo MCL para identificar clusters;

- Gonçalo Oliveira et al. (2010) Aplica um procedimento inspirado no anterior foi também aplicado ao português, na descoberta automática de synsets Gonçalo Oliveira et al. (2010);

- Navarro et al. (2009) categoriza e identifica vários synsets, a partir do Wikcionário. Os autores optaram por explorar o número de traduções de cada uma das palavras partindo do principio que se duas palavras partilham muitas traduções então é provável que estas sejam sinónimas;

- (Dorow, 2006) A autora opta após a criação do grafo de sinonímia a construção de um grafo em que cada nó é um par de palavras e as ligações são as ligações comuns do grafo original, para posterior aplicação do markov clustering;

- Gonçalo Oliveira and Gomes (2011) descobrem clusters em redes de sinónimos, também extraídas de dicionários, aproximando-os depois aos synsets de uma wordnet.

3.4 Manipulação de grafos

Como referido anteriormente, as redes extraídas através de dicionários podem ser vistas como grafos. Devido a esse facto, achou-se pertinente a utilização de algumas ferramentas que auxiliem na sua construção e manipulação, visto que não fazia sentido a sua implementação de raiz. As próximas duas secções analisam estas ferramentas de acordo com com as funcionalidades disponibilizadas por cada uma delas.

3.4.1 Frameworks

Existem várias frameworks para a construção e manipulação de grafos através de uma linguagem de programação. De forma a escolher a framework a utilizar neste trabalho, foram efetuados alguns comparativos entre as várias frameworks deste tipo. Estas ferramentas – JUNG[3], JGraphT[4], GraphStream[5], Sigma.fs[6], Gexf4j[7], e Gephi[8] – foram selecionadas como base numa pesquisa rápida e considerando alguns critérios, incluindo o tipo de licenciamento e a data da última atualização, ou seja, se o desenvolvimento se encontra ativo ou não.

Pode dividir-se as frameworks em três principais funcionalidades: a construção da estrutura do grafo, visualização do grafo, e algoritmos implementados de machine learning. Grande parte das frameworks enquadra-se em mais que um destes grupos, como apresentado na tabela 3.2.

Na tabela 3.2 a propriedade **Visualização** indica se a ferramenta consegue apresentar de modo gráfico um determinado grafo, a propriedade **Clustering** indica se é possível ou não a execução de algoritmos de cluster de modo a inferir agrupamentos, o **Construção** indica se a ferramenta disponibiliza algum meio que permita a construção de um grafo de raiz, o atributo **Operações** indica se é permito ao utilizador executar operações sobre os grafos, como por exemplo a consulta de estatísticas (grão médio, coeficiente de cluster, etc), seja a manipulação da estrutura do grafo propriamente dita (adição/remoção de nós/ligações, alteração de pesos, etc), a **Linguagem** refere-se à linguagem de programação para o qual as diversas API (Application Program Interface) estão desenhadas. A **Atualização** é a data da última versão, e a **Licença indica** o tipo de licenciamento em que a API/Ferramenta é disponibilizada

[3] `http://jung.sourceforge.net/`
[4] `http://jgrapht.org/`
[5] `http://graphstream-project.org/`
[6] `http://www.sigmafs.com/`
[7] `https://github.com/francesco-ficarola/gexf4j`
[8] `http://gephi.github.io/`

	JUNG	JGraphT	GraphStream	Sigma.fs	Gexf4j	Gephi
Visualização	✓	✓	✓	✓		✓
Clustering	✓		✓			✓
Construção	✓	✓	✓		✓	✓
Operações	✓	✓	✓	✓		✓
Linguagem	Java	Java	Java	JavaScript	Java	Java
Atualização	01/2010	12/2013	07/2014	08/2014	2013	01/2013
Licença	BSD	LGPL	GPL	MIT	Apache	GPL

Tabela 3.2: Tabela comparativa entre várias ferramentas referentes a grafos

Como podemos verificar na Tabela 3.2 o **Gephi** aparenta ser a *framework*, que mais opções possibilita, é equiparável ao Graphstream porém o Gephi possui uma aplicação de visualização de grafos (directamente relacionada com a API) com bastantes funcionalidades como por exemplo apresentar os vizinhos de segundo e terceiro graus e para o qual se encontra mais documentação disponível na internet. O Gephi é ainda modular ou seja é existem e é permitido o desenvolvimento de plugins para este software, o que se torna muito vantajoso porque uma abordagem de clustering sobre grafos é interessante desenvolver sobre a forma de plugin.

O **JUNG** e o **Graphstream** também se apresentam como bons candidatos, no entanto a disponibilização de plugins e a ferramenta de visualização foram os fatores que nos fizeram optar por esta solução (API Gephi e Ferramenta de visualização Gephi). O **JGrapht** é uma ferramenta eficaz para construção e manipulação de grafos, mas fica aquém em relação aos métodos que permitam o clustering e métricas de avaliação.

3.4.2 Formatos de serialização

Existem diversos formatos de serialização de grafos para ficheiros, tais como: GraphSON, GEXF, GraphML, GDF, GML, Pajek NET, GraphViz DOT, entre outros.

Estes formatos possuem diferentes características e funcionalidades,sendo comum verificar que as diferentes ferramentas normalmente privilegiam um formato (por ser o formato por omissão, e/ou por suportarem mais funcionalidades, etc), no entanto, não se procedeu a um estudo exaustivo sobre os vários formatos por vários motivos entre os quais:

- No nosso trabalho a construção da rede propriamente dita é visivelmente mais rápida (na ordem de poucos segundos) em comparação com os restantes passos necessários (na ordem de horas em redes mais complexas);
- Os requisitos necessários para os grafos são simples, (grafos unidirecionais, com diferentes pesos em cada ligação, suporte de um identificador/label do tipo string para identificação rápida da palavra respetiva a cada nó);
- Verificar se esse formato se adequa ao nosso objetivo ou seja se cumpre os requisitos acima apresentados, como visto mais a frente o formato pre-definido da ferramenta escolhida.

Sendo assim a escolha para o formato de serialização foi a seguinte:

1. Análise das ferramentas disponíveis;

2. Verificar qual o formato pré-definido/ou recomendado pela ferramenta escolhida;

3. Verificar se esse formato se adequa ao nosso objetivo ou seja se cumpre os requisitos acima apresentados.

Neste caso foi o formato de serialização a ser utilizado foi o GEXF visto que é o formato padrão da ferramenta escolhida que como veremos à mais à frente foi o Gephi suportando todos os requisitos necessários para o nosso objetivo.

A equipa do Gephi elaborou inclusivamente um quadro comparativo entre os vários formatos de serialização, comparativo esse apresentado na figura 3.9.

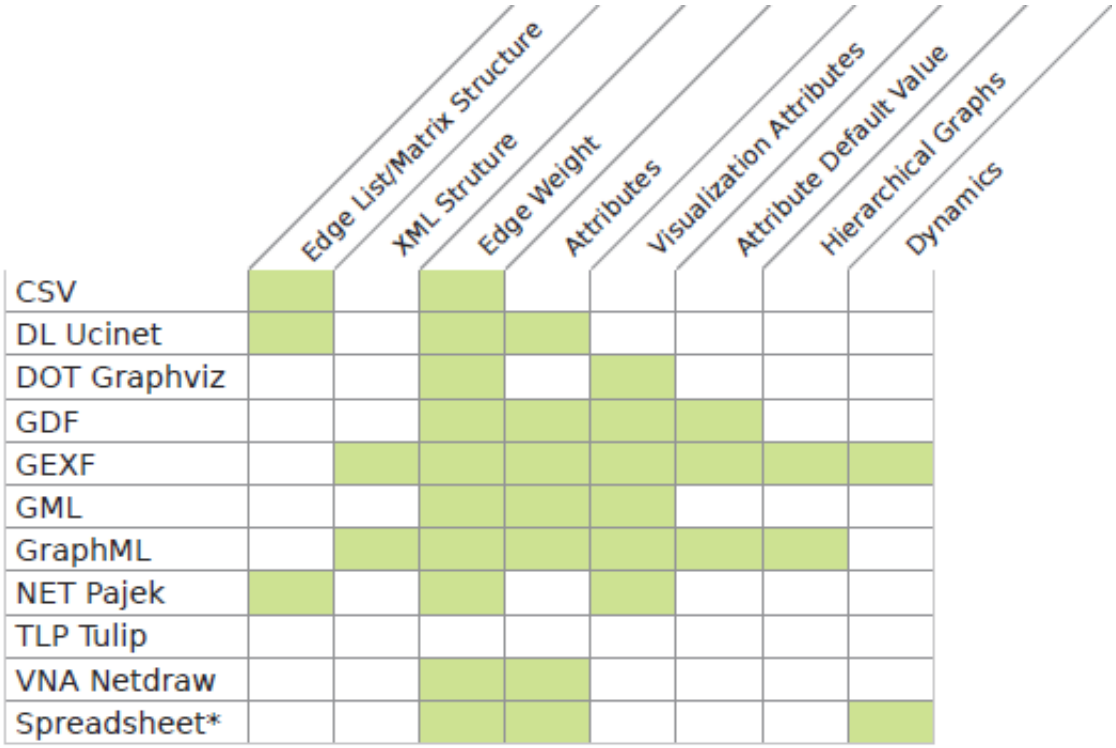

Figura 3.9: Quadro comparativo dos formatos suportados pelo Gephi

Neste gráfico podemos verificar que o GEXF é realmente o formato com mais funcionalidades suportadas pelo gephi.

3.5 Discussão

Após a análise dos algoritmos apresentados nas primeiras secções deste capítulo, chegamos à conclusão que nenhum deles se adequa ao problema aqui apresentado, uma vez que todos possuem restrições que os tornam inviáveis no domínio do nosso problema. Problemas esses que são:

- A abordagem fuzzy do K-means denominada como Fuzy C-Means não se adequa a este problema visto requerer que seja conhecido o número de clusters à o priori. No entanto pretende-se uma abordagem suficientemente flexível que

a partir de diferentes grafos identifique o número de clusters mais adequado consoante a sua estrutura;

- A maior parte dos algoritmos analisados não suportam a sobreposição de clusters o que os torna inviáveis devido à existência de palavras polissémicas, ou seja palavras que estariam presentes em mais do que um cluster;
- A maior parte dos algoritmos estudados apenas avaliam se o elemento está ou não contido no seu cluster, devido a esse facto é impossível a sua utilização para elaboração de clusters com medidas de pertença associadas.

Dadas estas limitações, no capítulo seguinte é descrito uma abordagem proposta para a resolução do nosso problema, tirando partido da framework Gephi que foi a escolhida após a análise, escolhendo consequentemente o formato de serialização GEXF visto que é o formato padrão por esta framework.

Capítulo 4

Abordagem proposta

Nesta tese são estudadas as relações presentes em vários dicionários. O nosso objetivo principal é a criação de agrupamentos de palavras com o mesmo significado, em que cada elemento de cada agrupamento possua um valor que avalie o grau de pertença ao respetivo agrupamento. Deseja-se que esta métrica contenha um valor numérico de modo a que quanto maior o valor mais representativo do synset seja.

Este trabalho toma partido da abordagem ECO (Gonçalo Oliveira and Gomes, 2014), referenciada no capítulo de Conhecimento Prévio, focando-se na fase de clustering. No entanto, ao invés dos agrupamentos discretos é aplicado uma heurística de modo a avaliar o grau de significância/pertença de cada palavra aos grupos onde esta se encontra.

Utilizando a abordagem ECO referida anteriormente são gerados triplos (Palavra Relação Palavra) que podem ser representados como um grafo (Palavras representadas como nós, relações representadas como arestas) e deste modo obtemos um grafo para cada categoria gramatical (ex Substantivos, Verbos, Nomes). Assim o objetivo será obter agrupamentos de palavras, que no grafo partilhem muitas ligações entre si, acreditando que em teoria isso representará conjuntos de palavras que partilhem o mesmo significado, e na solução aqui apresentada, uma métrica que avalie para cada palavra o grau de pertença aos conjuntos onde estas se encontram inseridas.

4.1 Solução proposta

Como visto no capítulo anterior, nenhuma das abordagens relacionadas possui os requisitos necessários para o objetivo aqui apresentado. Propomos então uma abordagem que combina algumas caraterísticas de vários algoritmos, e que consiste em dois passos principais:

1. Identificação de um conjunto de centróides;

2. Cálculo dos graus de pertença, com base na distância aos centróides.

No caso aqui apresentado, os centróides são nada mais nada menos que clusters discretos base, identificados a partir da estrutura do grafo e onde não há qualquer sobreposição. De certa forma, podem ser vistos como uma estrutura inicial, tal como os commitees no CBC, que será numa segunda fase aumentada. Para a sua identificação, contudo, deve ser utilizado um algoritmo eficiente que tire partido da estrutura do grafo, tal como o CW.

No segundo passo, os graus de pertença de cada palavra são calculados com base na semelhança entre as características (outras palavras) da palavra que são relevantes para o centróide e as palavras do próprio centróide, o que de certa forma se assemelha ao cálculo das pertenças no FCM. No entanto, não será necessário realizar novas iterações, porque acreditamos que o CW gerará centróides que já incluirão palavras com um elevado grau de proximidade.

Formalizando, a abordagem proposta é aplicada a uma rede de sinonímia $G = (P, R)$, onde P é o conjunto de palavras e R é o conjunto de pares de sinonímia. A rede G pode ser representada através de uma matriz de adjacências $A(|P| \times |P|)$, onde $A_{ij} = \omega_{ij}$, um peso que reflete o número de vezes que um par de sinónimos, $R(P_i, P_j)$, ocorre nas fontes utilizadas (dicionários, por exemplo). O peso máximo m é portanto uma constante, igual ao número de fontes utilizadas.

No primeiro passo, o algoritmo de clustering aplicado resulta num primeiro conjunto de clusters centróide C.

No segundo, o valor de pertença da palavra P_i ao centróide C_k, $\mu(P_i, C_k)$, é calculado através da equação 4.1, onde T é o conjunto de palavras relevantes para o cálculo, ou seja, todas as palavras do centróide C_k e ainda a palavra P_i, que pode ou não estar no centróide (ver equações 4.2). Este calculo tenta tirar partido das semelhanças entre as palavras, neste caso palavras que tenham relações de sinonímia semelhantes, porém apenas se consideraram semelhanças como "relevantes"caso exista pelo menos uma ligação a pelo menos uma palavra dentro do cluster, uma vez que consideramos que a semelhança do facto de não haver ligação não é de todo tão significativa como a existência de semelhanças quando existem efetivamente ligações comuns. O facto de ignorar as características "não relevantes", ou seja, não classificar duas palavras semelhantes por partilharem o facto de não estarem ligadas a determinada palavra, como semelhança é uma das diferenças mais importantes da abordagem de Gonçalo Oliveira and Gomes (2011), onde são utilizados o cálculo de cossenos dos vectores considerando todas as características e não apenas as relevantes, que correspondem a relações efetivas.

$$\mu(P_i, C_k) = \frac{\sum_{j=0}^{|C_k|} A_{ij}}{m \times |T|} \tag{4.1}$$

$$\begin{aligned} T &= \{C_k \cup P_i\}, \text{ou seja} \\ |T| &= \{|C_k|, P_i \in C_k \vee |C_k| + 1, P_i \notin C_k\} \end{aligned} \tag{4.2}$$

Devido a utilização de estruturas iniciais, estruturas essas que já se encontram preenchidas com elementos é necessário tomar em conta se a palavra cuja pertença está a ser calculada já se encontra, ou não no cluster inicial. Esta diferenciação é importante devido a coerência matemática, já que neste caso, para a pontuação máxima, ou seja, a obtenção do valor 1, a palavra deve ter o peso máximo em todas as componentes representativas do cluster, sendo posteriormente dividida pelo número de características relevantes. Estas resultam da reunião entre a palavra a avaliar e as palavras do cluster, ou seja, evita-se assim que a posição no vetor de características seja contabilizada duas vezes.

4.2 Exemplos

A abordagem é ilustrada com auxílio do grafo na figura 4.1, centrado na palavra *banana*. Em português europeu, esta palavra tanto pode ser o nome de uma fruta, como pode ter o sentido figurado de uma pessoa sem iniciativa. Suponha-se que o grafo é extraído a partir de três dicionários ($m = 3$) e que o algoritmo CW, aqui utilizado para o primeiro passo, identifica os dois *clusters* centróide representados na tabela 4.1. Para calcular o valor de pertença de *banana* ao centróide C_A, devem ser consideradas as ligações às palavras *musa* e *bananeira*, ou seja, apenas 1. Este número é dividido por $3 \times |T|$, em que as palavras relevantes $T = \{musa, bananeira, banana\}$. Portanto, neste caso: número de ligações ao cluster = 1 número de dicionários = 3 número de palavras relevantes = 3 então $\mu(banana, C_A) = \frac{1}{3\times3} = \frac{1}{9}$.

Para o cálculo da pertença da palavra *banana* ao centróide C_B, as características relevantes são o número de ligações com todas as palavras do centróide C_A, apenas 1, para a palavra *pateta*. Considera-se ainda que cada palavra tem o número máximo de "ligações"a si própria, por isso, neste caso, como $banana \in C_B$, soma-se 3 ao número de ligações relevantes. Ou seja, o numerador será 4 (3 da palavra própria + 1 ocorrência entre pateta e uma palavra do cluster = 4) e o denominador será 21 (numero dicionários = 3 × numero de palavras relevantes = 7), e assim, $\mu(banana, C_B) = \frac{4}{21}$.

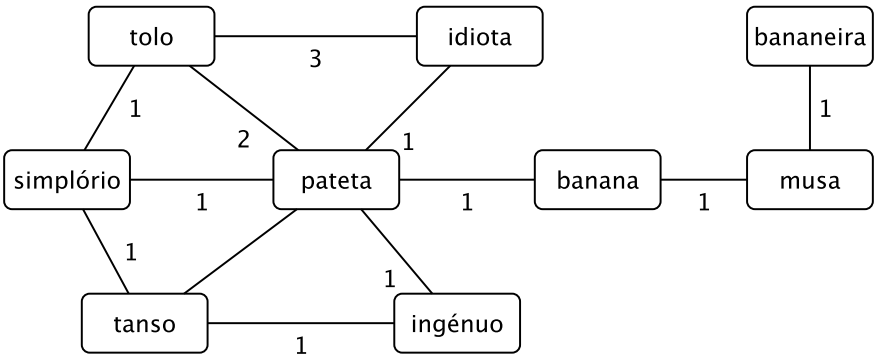

Figura 4.1: Rede de sinonímia com palavras e pesos das ligações.

C_A	*musa*, *bananeira*
C_B	*banana*, *pateta*, *idiota*, *tolo*, *simplório*, *tanso*, *ingénuo*

Tabela 4.1: Centróides descobertos a partir da rede da figura 4.1, com o algoritmo Chinese Whispers.

C_A	*bananeira*(0.666);*musa*(0.666);*banana*(0.111)
C_B	*pateta*(0.476);*tolo*(0.428);*idiota*(0.333);*simplório*(0.285);...;*musa*(0.041)

Tabela 4.2: Pertenças cálculadas com base nos clusters discretos presentes na 4.1

A tabela 4.2 mostra os clusters obtidos após a utilização do algoritmo aqui apresentado, como podemos verificar foram adicionadas algumas palavras aos clusters iniciais, como por exemplo *banana* e *musa* nos clusters C_A e C_B respetivamente, podemos constatar ainda que as pertenças parecem fazer sentido na medida em que

as palavras com maior valor de pertença aparentam partilhar "mais semelhança"que as palavras com menor valor,

4.3 Escolha do algoritmo para o primeiro passo

A abordagem proposta é suficientemente flexível para, no primeiro passo, aceitar qualquer tipo de algoritmo de clustering sobre grafos. De forma a escolher o algoritmo a aplicar nas experiências realizadas, os dois algoritmos candidatos, MCL e CW, foram aplicados a redes de sinonímia extraídas automaticamente a partir da versão portuguesa do Wikcionário[1], no âmbito do projeto Onto.PT (neste caso, na rede CARTÃO (Gonçalo Oliveira et al., 2011), apresentada na próxima secção). Para este trabalho adaptamos ainda o CBC para grafos, integrando este algoritmo na API do Gephi, juntando assim o CBC ao MCL e CW na para a lista de algoritmos disponíveis.

Para testar os vários algoritmos e comparar os seus resultados foi escolhido um subconjunto da rede CARTÃO. Para aumentar a rapidez de execução destes testes, foram utilizadas apenas as relações de sinonímia extraídas do dicionário mais pequeno desta rede, o Wikcionário.PT, que também tem uma rede mais pequena, e assim mais rápida de processar. Apesar de este teste ter sido mais tarde replicado nas outras redes, os resultados obtidos foram semelhantes e não se incluem aqui por não haver necessidade de aumentar a complexidade desta análise.

A tabela 4.3 apresenta algumas propriedades da rede de sinonímia em questão. Para o sub-grafo de cada categoria gramatical, é indicado o número de vértices (palavras, $|P|$) e arestas distintas (relações de sinonímia, $|R|$), o grau médio dos vértices ($\overline{deg}(P)$), o coeficiente médio de *clustering* ($\overline{CC}$), o número de componentes conectadas ($|Comp|$), e o número de palavras da componente maior ($|P|_{mc}$). A tabela 4.4 mostra as propriedades dos clusters (neste caso tambem synsets) obtidos a partir da rede do Wikcionário, com os algoritmos CW, MCL e CBC. Para além do número médio de sentidos por palavra ($\overline{sents}$) e número de sentidos da maior palavra ($max(\#sents)$), é indicado o total de synsets (#), synsets com apenas uma palavra ($tam = 1$), duas palavras ($tam = 2$) e mais de 25 palavras ($tam > 25$), e ainda o tamanho do maior synset ($max(tam)$).

POS	$\|P\|$	$\|R\|$	$\overline{deg}$(G)	$\overline{CC}$	$\|Comp\|$	$\|P\|_{mc}$
N	14743	15462	2.10	0.154	14743	14742
V	3953	5494	2.78	0.147	3953	3953
Adj	5968	7421	2.49	0.150	5968	5968
Adv	386	297	1.54	0.084	386	386

Tabela 4.3: Propriedades numéricas da rede de sinonímia do Wikicionário.

Como podemos verificar o MCL, é o único que consegue agrupar palavras polissémicas, mas a um nível muitíssimo reduzido visto que são necessárias condições muito especificas, para uma instância ser classificada em mais do que uma categoria, facto esse já referido na secção anterior. Esses nós considerados instáveis foram até explorados por Dorow (2006) e Gfeller et al. (2005).

[1] `http://pt.wiktionary.org`

	Cat	Palavras		Synsets					
		$\overline{sents}$	$max(\#sents)$	$\#$	$\overline{tam}$	$tam = 1$	$tam = 2$	$tam > 25$	$max(tam)$
CW	**N**	1	1	3608	4,09	0	2005	50	165
	V	1	1	601	6,58	0	223	24	116
	Adj	1	1	1200	4,97	0	613	30	120
MCL	**N**	1,01	4	7326	2,58	173	4909	0	23
	V	1,11	3	1587	2,77	103	2005	0	17
	Adj	1,22	4	2738	2,66	125	1648	0	15
CBC	**N**	1	1	7984	1,84	5170	5170	1	27
	V	1	1	1608	2,45	718	332	0	15
	Adj	1	1	2835	2,10	1609	1609	0	15

Tabela 4.4: Propriedades numéricas dos synsets obtidos a partir da rede do Wikcionário.

Um dos graves problemas do MCL é a memória que é necessária alocar. Com a implementação do MCL suportada pela API do Gephi é necessária alocar muito mais memória RAM que as outras duas, ultrapassando largamente a memória disponível que no caso da maquina de testes era de 24 GB, uma vez que os outros dois algoritmos testados não apresentaram este tipo de problemas para as mesmas redes, este algoritmo foi descartado.

Uma vez que os autores do CBC projetaram o algoritmo para matrizes de co-ocorrência e o CW e o MCL foram diretamente desenhados para grafos, consideramos este um fator que favorece o CW e o MCL, visto que é sobre grafos que a nossa abordagem opera. Outro fator que nos fez descartar o CBC foi o elevado numero de clusters com apenas um elemento e visto que a nossa abordagem tira partido das ligações a grupo de palavras (cluster), achamos este fator relevante para a sua exclusão

A título de exemplo e como método de comparação foram executados alguns testes já com a solução de integração aqui apresentada. Para a rede de substantivos, do Wikicionario o CW conseguiu efetuar os agrupamentos em 26 minutos, o CBC conseguiu resolver em 138 minutos, e o MCL em 37 minutos.

Com base na realização das várias experiências, optámos por utilizar o CW para esta fase da abordagem.

Capítulo 5

Experimentação

Esta capítulo apresenta os resultados da aplicação da abordagem proposta à rede léxico-semântica CARTÃO, que se começa por descrever, seguida de uma visão numérica dos resultados e, por fim, de exemplos ilustrativos, com alguns dos synsets difusos obtidos.

5.1 Rede Léxico-Semântica

A rede léxico-semântica utilizada para a descoberta de synsets difusos foi o CARTÃO (Gonçalo Oliveira et al., 2011), disponível gratuitamente, e extraída de forma automática a partir de três dicionários da língua portuguesa, incluindo o Wikcionário, com base em padrões textuais nas suas definições.

Para ajudar a caracterizar esta rede, a tabela 5.1 apresenta algumas das suas propriedades numéricas, mais propriamente para os sub-grafos de sinonímia entre substantivos (N), verbos (V), adjetivos (Adj) e advérbios (Adv).

Para cada sub-grafo, é indicado o número de vértices (palavras, $|P|$) e arestas distintas (relações de sinonímia, $|R|$), o grau médio dos vértices ($\overline{deg}(P)$), o coeficiente médio de *clustering* ($\overline{CC}$), o número de componentes conectadas ($|Comp|$), e o número de palavras da componente maior ($|P|_{mc}$). Tal como outros investigadores (por exemplo, Gfeller et al. (2005)), também nos apercebemos que estes sub-grafos, extraídos de dicionários, são constituídos por uma grande componente, e várias pequenas. Os coeficientes de clustering são comparáveis aos de outras redes de pequeno mundo (em inglês, *small-world networks*), em que a distância média entre dois vértices é curta. Comparando os três sub-grafos, o sub-grafo dos verbos possui um grau médio mais elevado, o que significa que os verbos terão mais sinónimos e/ou serão mais ambíguos. Também se observa que o sub-grafo de advérbios é significativamente mais pequeno que os demais, por isso acabou por não ser utilizado nas experiências apresentadas nas próximas secções.

POS	$\|P\|$	$\|R\|$	$\overline{deg}$(G)	$\overline{CC}$	$\|Comp\|$	$\|P\|_{mc}$
N	43,724	65,127	2.98	0.21	5,812	28,734
V	10,380	26,266	5.06	0.25	362	9,549
Adj	31,014	17,368	3.57	0.23	2,049	12,343
Adv	1,271	1,296	2.04	0.18	160	819

Tabela 5.1: Propriedades numéricas da rede CARTÃO.

5.2 Propriedades dos synsets descobertos

Ao correr o algoritmo proposto no CARTÃO, obtemos um conjunto com quase 15 mil synsets difusos C', com as propriedades apresentadas nas tabelas 5.2 e 5.3 para cada categoria gramatical, nomeadamente: número de palavras (#pals), média de sentidos por palavra ($\overline{sents}$), palavra com mais sentidos (max(#sents)), total de synsets (#synsets), média de palavras por synset ($\overline{|synset|}$), synsets com apenas duas palavras ($|synset| = 2$), synsets com mais de 25 palavras ($|synset| > 25$), e tamanho do maior synset (max($|synset|$)). Na mesma tabela, incluem-se as mesmas propriedades para abordagem de (Gonçalo Oliveira and Gomes, 2011), onde foi utilizada uma versão anterior do CARTÃO (originalmente Padawik (Gonçalo Oliveira and Gomes, 2011), depois rebaptizado como CLIP (Gonçalo Oliveira, 2013)), com um ponto de corte de 0,01, e ainda as propriedades dos synsets no tesauro TeP 2.0 (Maziero et al., 2008), criado manualmente para o português do Brasil e aqui usado como referência.

Quando comparado com o CLIP, parece haver menos ruído. Isto porque existem menos synsets, em média mais pequenos para os nomes e adjetivos, e de tamanho comparável para os verbos. As palavras são também menos ambíguas. No TeP, o número médio de palavras por synset é mais baixo, tal como o número médio de sentidos por palavra, o que já era esperado, não só pela abordagem difusa, mas também pelo maior grau de cobertura do nosso tesauro. Recordamos, no entanto, que pode ser aplicado um ponto de corte aos synsets difusos, de modo que estes fiquem pequenos e, tendencialmente, mais confiáveis. Por outro lado, também no TeP, o número de synsets de verbos e adjetivos é mais do dobro, e ligeiramente mais baixo para os substantivos, No entanto, os nossos synsets cobrem quase o dobro das palavras do TeP (cerca de 70 mil contra 40 mil), mais propriamente um número próximo de verbos, ligeiramente superior de adjetivos, e mais do dobro de substantivos.

	Cat	Palavras		
		#pals	$\overline{sents}$	max(#sents)
Actual	**N**	43.721	1,92	42
	V	10.380	3,15	54
	Adj	17.368	2,28	44
CLIP 0,01	**N**	39.354	7,78	46
	V	11.502	14,31	42
	Adj	15.260	10,36	43
TeP 2.0	**N**	17.158	1,71	21
	V	10.827	2,08	41
	Adj	14.586	1,46	19

Tabela 5.2: Propriedades das palavras

As tabelas 5.4 e 5.5 ilustram os resultados obtidos através de uma seleção de palavras polissémicas da língua portuguesa – respectivamente substantivos e adjectivos, e verbos – e alguns dos synsets difusos que as incluem, organizados de acordo com o conceito que transmitem (frequentemente clarificado pela palavra com maior pertença) e onde as palavras são apresentadas por ordem decrescente do grau de pertença. Numa observação global, podemos observar nas tabelas que tanto a cons-

	Cat	Synsets				
		#synsets	$\overline{\lvert synset \rvert}$	$\lvert synset \rvert = 2$	$\lvert synset \rvert > 25$	$\max(\lvert synset \rvert)$
Actual	**N**	9.881	8,49	4.147	632	554
	V	1.438	22,76	289	370	500
	Adj	3.571	11,.07	1.530	367	322
CLIP 0,01	**N**	20.102	15,23	3,885	3,756	109
	V	7.775	21,17	307	2,411	89
	Adj	8.896	17,77	1,326	2,157	109
TeP 2.0	**N**	8.254	3,56	3.079	0	21
	V	3.978	5,67	939	48	53
	Adj	6.066	3,50	3.033	19	43

Tabela 5.3: Propriedades dos synsets

tituição dos synsets como os graus de pertença parecem fazer sentido ou seja as palavras com maior pertença dentro do mesmo synset partilham mais significância entre si. Num dos exemplos presente nas tabelas 5.4 e 5.5, é possível verificar que para o synset que transmite o significado de mistura as palavras *mistura*(0,333), *amálgama*(0,127) e *mescla*(0,111) possuem uma maior semelhança que as palavras *cocktail*(0,015), *combinação*(0,015) e *logro*(0,015), que são as palavras com a pontuação menor

No entanto é visível que nos synsets maiores tendencialmente ocorrem pertenças menores que nos synsets com menos elementos como por exemplo: *degelo(0.778)*, *descongelação*(0.556), *descoalho*(0.556) e *fusão(0.0833)*, uma possível solução para este facto seria efetuar uma normalização dentro de cada synset.

5.3 Avaliação

De modo a avaliar de forma mais objetiva o desempenho da abordagem aqui apresentada, foram utilizados dois procedimentos

1. Avaliação contra um recurso dourado;
2. Avaliação manual.

5.3.1 Avaliação com recurso dourado

O recurso escolhido para a avaliação automática foi o TeP uma vez que o TeP, é um tesauro criado manualmente para o português, há alguma confiança nos seus conteúdos. Para além disso, foi criado de forma completamente independente do CARTÃO. Daí ter sido o TeP a nossa primeira opção para verificar a qualidade dos synsets descobertos.

A avaliação foi feita através da confirmação de todos os pares de sinonímia nos synsets de cada um dos tesauros descobertos. Considera-se que um par de sinonímia, $R(w_a, w_b)$, é um conjunto de duas palavras que pertencem ao mesmo synset C_x, ou seja, $R(w_a, w_b) \rightarrow \exists C_x : w_a \in C_x \wedge w_b \in C_x$. Então, para cada par nos tesauros descobertos, verificou-se se existia pelo menos um synset no TeP que contivesse as duas palavras. A tabela 5.6 apresenta a proporção de pares confirmados para os synsets de cada categoria gramatical, não só para os resultados da abordagem actual,

Palavra	Conceito	Synsets difusos
pasta	mistura	*mistura*(0.333), *amálgama*(0.127), *mescla*(0.111), *matalotagem*(0.079), *anguzada*(0.079), *comistão*(0.079), *misto*(0.079), *landoque*(0.079), *salsada*(0.0758), *confusão*(0.0758), *enovelamento*(0.063), *cacharolete*(0.063), *macedónia*(0.063), *mexedura*(0.063), *caldeação*(0.063), *mixagem*(0.063), *pasta*(0.063), *angu*(0.063), *amalgamação*(0.063), *comistura*(0.063), *impurezas*(0.063), *mistão*(0.063), *estricote*(0.063), *fusão*(0.045), *temperamento*(0.03), *pot-pourri*(0.015), *imisção*(0.015), *conjunto*(0.015), *ensalsada*(0.015), *envolta*(0.015), *agrupamento*(0.015), *baralha*(0.015), *marinhagem*(0.015), *salgalhada*(0.015), *misturada*(0.015), *miscelânea*(0.015), *têmpera*(0.015), *imperfeição*(0.015), *cocktail*(0.015), *combinação*(0.015), *logro*(0.015), ...
	dinheiro	*dinheiro*(0.28), *bufunfa*(0.069), *caroço*(0.053), *tutu*(0.042), *pataco*(0.037), *bagalhoça*(0.037), *guines*(0.037), *cobre*(0.032), *pecúnia*(0.032), *gaita*(0.032), *cacique*(0.032), *pílula*(0.026), *morubixaba*(0.026), *pila*(0.026), *cacau*(0.026), *arame*(0.026), *calombo*(0.026), *patacaria*(0.026), *gimbo*(0.026), *maco*(0.026), *bubão*(0.026), *chelpa*(0.026), *roço*(0.026), *levação*(0.026), *íngua*(0.026), *vénus*(0.021), *verdinha*(0.021), *mondrongo*(0.021), *pírula*(0.021), *dindim*(0.021), *trocado*(0.021), *curaca*(0.021), *pataca*(0.021), *massaroca*(0.021), *bagalho*(0.021), *carcanhol*(0.021), *pilim*(0.021), *encórdio*(0.021), *teca*(0.021), *coronel*(0.021), *matambira*(0.021), *mussuruco*(0.021), *cinco-réis*(0.021), *metal*(0.021), *cunques*(0.021), *jan-da-cruz*(0.021), *boro*(0.021), *cum-quibus*(0.021), *bilhestres*(0.021), *calique*(0.021), *parrolo*(0.021), *zerzulho*(0.021), *caronha*(0.021), *nhurro*(0.021), *baguines*(0.021), *pecuniária*(0.021), *pecunia*(0.021), *marcaureles*(0.021), *china*(0.021), *fanfa*(0.021), *dieiro*(0.021), *influente*(0.021), *guino*(0.021), *grana*(0.02), *tostão*(0.01), *riqueza*(0.01), *cabedal*(0.01), *posses*(0.01), *inchaço*(0.01), ...
planta	vegetal	*vegetal*(0.667), *plantas*(0.667), *planta*(0.111)
	plano	*plano*(0.379), *projecto*(0.23), *tenção*(0.207), *desígnio*(0.207), *traçado*(0.161), *propósito*(0.161), *intenção*(0.149), *pressuposto*(0.138), *intento*(0.138), *prospecto*(0.126), *desenho*(0.126), *planta*(0.126), *programa*(0.115), *traça*(0.0.115), *mente*(0.092), *risco*(0.089), *resolução*(0.089), *prospeto*(0.08), *arquitectura*(0.08), *ideia*(0.078), *pressuposição*(0.069), *traçamento*(0.069), *prepósito*(0.069), *presuposto*(0.069), *intuito*(0.067), *vista*(0.067), *alçado*(0.057), *planificação*(0.057), *design*(0.057), *pranta*(0.057), *esboço*(0.055), *planejamento*(0.045), *fundição*(0.046), *gizamento*(0.046), *caruru*(0.046), *aspecto*(0.044), *medida*(0.044), *fim*(0.044), *vontade*(0.044), *desejo*(0.044), *objectivo*(0.033), *conjectura*(0.033), *escopo*(0.033), *sentido*(0.033), *cálculo*(0.033), *disposição*(0.033), *espírito*(0.033), ...
sede	centro	*centro*(0.6), *núcleo*(0.4), *sensório*(0.333), *foco*(0.333), *club*(0.267), *sede*(0.222), *âmago*(0.222), *meio*(0.167), *coração*(0.167), *metrópole*(0.111), *escol*(0.056), *pólo*(0.056), *clube*(0.056), *umbigo*(0.056), *cérebro*(0.056), *fundo*(0.056), *gema*(0.056), *cadeira*(0.056), *casco*(0.056), *aglomeração*(0.056), *grupo*(0.056), *empório*(0.056), *essência*(0.056), *casino*(0.056), *profundeza*(0.056), *caroço*(0.056), *sociedade*(0.056)
	secura	*sede*(0.429), *secura*(0.333), *sequidão*(0.286), *seda*(0.238), *cerdas*(0.19), *sieda*(0.19), *seeda*(0.19), *aridez*(0.083), *centro*(0.083), *cerda*(0.042), *foco*(0.042), *impassibilidade*(0.042), *mortalha*(0.042), *cadeira*(0.042), *núcleo*(0.042), *diocese*(0.042), *ambição*(0.042), *impaciência*(0.042), *banco*(0.042), *apetite*(0.042), *avidez*(0.042), *ânsia*(0.042), *insensibilidade*(0.042), *capital*(0.042), *polidipsia*(0.042), *luxo*(0.042), *frieza*(0.042), *seta*(0.042), *magreza*(0.042)
	impaciência	*impaciência*(0.533), *frenesi*(0.467), *rabujice*(0.267), *despaciência*(0.267), *farnesia*(0.267), *inquietação*(0.222), *sofreguidão*(0.167), *pressa*(0.167), *desespero*(0.111), *nervosismo*(0.111), *ansiedade*(0.111), *exaltação*(0.111), *cócegas*(0.111), *freima*(0.111), *freimaço*(0.056), *formigueiro*(0.056), *precipitação*(0.056), *agastamento*(0.056), *impertinência*(0.056), *sofreguice*(0.056), *sede*(0.056), *inguinação*(0.056), *ira*(0.056), *furor*(0.056), *excitação*(0.056), *prurido*(0.056), *fúria*(0.056), *afã*(0.056)
verde	cor verde	*verde*(0.274), *virente*(0.137), *verdejante*(0.137), *relvoso*(0.118), *gramíneo*(0.098), *esmeraldino*(0.098), *prásino*(0.098), *desassazonado*(0.098), *viridente*(0.098), *ervoso*(0.098), *verdoso*(0.098), *ecológico*(0.078), *dessazonado*(0.078), *graminoso*(0.078), *viridante*(0.078), *herboso*(0.078), *porráceo*(0.078), *viçoso*(0.055), *inoportuno*(0.037), *fresco*(0.037), *esverdeado*(0.037), ...
	amador	*inexperiente*(0.917), *noviço*(0.067), *novato*(0.067), *inexperto*(0.417), *novel*(0.267), *ingénuo*(0.267), *inocente*(0.267), *principiante*(0.133), *novo*(0.133), *viçoso*(0.133), *matumbo*(0.067), *incompetente*(0.067), *amador*(0.067), *verde*(0.067), *moço*(0.067), *bisonho*(0.067), *ingênuo*(0.067), ...

Tabela 5.4: Synsets difusos de palavras polissémicas (substantivos e adjectivos).

mas também para o CLIP. Na mesma tabela, de forma a verificar se as medidas de pertença fazem sentido, apresenta-se, para cada tesauro criado de forma automática, a média das medidas de pertença dos pares confirmados ($\overline{\mu_c}$) e não confirmadas ($\overline{\mu_c}$)

Palavra	Conceito	Synsets difusos
limpar	tornar limpo	*limpar*(0.262), *purificar*(0.126), *enxugar*(0.098), *expurgar*(0.066), *mundificar*(0.06), *desinfectar*(0.06), *purgar*(0.055), *secar*(0.055), *depurar*(0.049), *mirrar*(0.049), *lavar*(0.049), *descontaminar*(0.044), *despoluir*(0.038), *desinçar*(0.038), *virginizar*(0.038), *esburgar*(0.038), *dessecar*(0.038), *assear*(0.038), *luir*(0.038), *varrer*(0.038), *esmirrar*(0.033), *desensopar*(0.033), *desenxovalhar*(0.033), *absterger*(0.033), *tamisar*(0.027), *virginalizar*(0.027), *desparasitar*(0.027), *vassourar*(0.027), *desenxamear*(0.027), *emundar*(0.027), *desecar*(0.027), *desempestar*(0.027), *desenodoar*(0.027), *desenfarruscar*(0.027), *perlavar*(0.027), *detergir*(0.027), *achicar*(0.027), *estomentar*(0.027), *desencharcar*(0.027), ...
	podar	*desramar*(0.778), *escamondar*(0.556), *mondar*(0.556), *limpar*(0.25), *petelar*(0.083), *desgalhar*(0.083), *derramar*(0.083), *alveitarar*(0.083), *carpir*(0.083), *capinar*(0.083), *corrigir*(0.083)
	peneirar	*joeirar*(0.533), *escrivar*(0.333), *utar*(0.267), *acrivar*(0.267), *outar*(0.267), *peneirar*(0.111), *limpar*(0.111), *tamisar*(0.056), *crivar*(0.056), *cirandar*(0.056), *brocar*(0.056)
	roubar	*ripar*(0.533), *bifar*(0.467), *ripançar*(0.4), *surrupiar*(0.267), *palmar*(0.267), *surripiar*(0.222), *furtar*(0.111), *limpar*(0.111), *pifar*(0.056), *raspar*(0.056), *arrancar*(0.056), *puxar*(0.056)
estimar	apreciar	*apreciar*(0.444), *valorar*(0.333), *estimar*(0.333), *avaliar*(0.333), *cotar*(0.222), *valorizar*(0.222), *admirar*(0.222), *ponderar*(0.19), *considerar*(0.143), *amar*(0.095), *discernir*(0.095), *julgar*(0.095), *equacionar*(0.048), *ustir*(0.048), *trutinar*(0.048), *estranhar*(0.048), *qualificar*(0.048), *apreçar*(0.048), *gostar*(0.048), *desfrutar*(0.048), *adular*(0.048), *conhecer*(0.048), *recensear*(0.048), *aquilatar*(0.048), *numerar*(0.048), *desejar*(0.048), *sentir*(0.048), *reputar*(0.048), *calcular*(0.048), *revelar*(0.048), *vêr*(0.048)
	avaliar	*avaliar*(0.625), *aquilatar*(0.375), *quilatar*(0.292), *apreçar*(0.292), *equacionar*(0.208), *almotaçar*(0.208), *conceituar*(0.208), *aderar*(0.208), *julgar*(0.185), *estimar*(0.148), *apreciar*(0.148), *pesar*(0.111), *conhecer*(0.111), *louvar*(0.111), *calcular*(0.111), *ajuizar*(0.074), *quantiar*(0.074), *aferir*(0.074), *computar*(0.074), *aperfeiçoar*(0.074), *ponderar*(0.074), *reputar*(0.074), *cotar*(0.037), *valorar*(0.037), *arbitrar*(0.037), *mensurar*(0.037), *qualificar*(0.037), *contrastar*(0.037), *orçar*(0.037), *montar*(0.037), *taxar*(0.037), *apurar*(0.037), *discernir*(0.037), *examinar*(0.037), *tomar*(0.037)

Tabela 5.5: Synsets difusos de palavras polissémicas (verbos).

no TeP, e ainda a média das pertenças mínimas de cada par confirmado $\overline{min(\mu_c)}$ e não confirmado $\overline{min(\mu_{!c})}$.

Relativamente ao CLIP, há agora mais pares de sinonímia confirmados no TeP, ainda que esta proporção esteja sempre abaixo dos 42%. Uma razão para isto acontecer é o facto de TeP e CARTÃO serem recursos, até certo ponto, complementares, não só relativamente a lemas, mas também a pares de sinonímia (veja-se a comparação realizada em (Gonçalo Oliveira et al., 2011)). Aliás, o TeP foi criado para o português do Brasil, enquanto que o CARTÃO se baseia principalmente em dicionários feito para o português de Portugal. Ou seja, alguns dos pares não confirmados podem ser sentidos não cobertos pelo TeP.

Relativamente aos graus de pertença, para ambos os tesauros é visível que se tem medidas mais elevadas para pares confirmados do que para pares não confirmados. Ora, isto é precisamente o desejado, visto que, com os graus de pertença, queremos medir a confiança, que deverá ser mais baixa para palavras cuja inclusão no synset é mais duvidosa ou mesmo incorreta.

Um dado importante a ter em conta na análise de resultados é o facto de que, em synsets grandes, a ausência de uma palavra representa uma grande perda em combinações confirmadas, uma vez que se perdem todas as combinações entre as palavras restantes. A fórmula para calcular o número de pares necessários para que a cobertura no TeP seja total está presente na fórmula da figura 5.1. Por exemplo, um synset com 3 palavras apenas tem 2 combinações de sinonímia possíveis enquanto que com 10 elementos esse valor sobe para 45 pares.

Com isto chega-se facilmente à conclusão que os synsets menores terão uma maior

$$\binom{n}{p} = \frac{n!}{(n-p)!p!}$$

Figura 5.1: Fórmula para calcular o número de pares possíveis

probabilidade de ser classificados como perfeitos no TeP. Ou seja quanto maior o synset no caso de uma palavra que não esteja no TeP maior a pontuação decrescida porque todos os pares dessa palavra em com todas as outras palavras do synset serão consideradas como erradas.

	CLIP 0,01					Actual				
Cat	**Confirmados**	$\overline{\mu_c}$	$\overline{\mu_{!c}}$	$\overline{min(\mu_c)}$	$\overline{min(\mu_{!c})}$	**Confirmados**	$\overline{\mu_c}$	$\overline{\mu_{!c}}$	$\overline{min(\mu_c)}$	$\overline{min(\mu_{!c})}$
N	27,89%	0,25	0,15	0,13	0,05	30,00%	0,33	0,17	0,16	0,06
V	27,91%	0,19	0,13	0,13	0,05	33,27%	0,29	0,16	0,12	0,05
Adj	32,37%	0,29	0,16	0,13	0.05	41,44%	0,38	0,18	0,13	0.06

Tabela 5.6: Confirmação de sinónimos no TeP.

Também a ter em conta é o facto de serem avaliadas todas as combinações de pares possíveis uma vez que não foi utilizado nenhum ponto de corte, ou seja os pares compostos por uma palavra com pontuação alta e uma palavra com pontuação baixa, a pontuação da palavra mais alta irá aumentar de sobremaneira a pontuação média do par, No exemplo (*piolheira(0,78)*, *piolhada(0,67)*, *piolharia(0,44)*, *cabeça(0,083)* ,*porcaria(0,083)*, *galinheiro(0,083)*, *pocilga(0,083)*,) **piolheira** tem uma pontuação elevada e **galinheiro** uma pontuação baixa, no entanto a pontuação média desse pare será de 0,431 uma pontuação relativamente alta, (visto que as pontuações de cada palavra são referentes ao synset no seu todo e não apenas ao par a ser comparado) para uma situação que claramente vai dar como falsa, tendo em conta este facto foi utilizado a métrica pertença mínima de cada par para mitigar esse facto. Analisando essa métrica é possível verificar, tal como seria de esperar, que as pontuações são bastante reduzidas em comparação com a sua média. No entanto as pontuações nos pares não confirmados tiveram pontuações marginais tanto no CLIP como na solução aqui apresentada, sendo que considerando esta métrica a pontuação dos pares confirmados sobe em relação á abordagem CLIP. Da utilização desta métrica é possível inferir que a utilização de um ponto de corte relativamente baixo aumentaria a confiança da constituição destes synsets.

5.3.2 Avaliação manual

Devido às limitações já referidas do TeP, decidimos efetuar uma avaliação adicional, desta vez manual, seguindo as mesmas regras que na avaliação feita ao CLIP, detalhada em (Gonçalo Oliveira and Gomes, 2011) e (Gonçalo Oliveira, 2013). Mais precisamente, esta avaliação passou pelas seguintes fases:

1. Remoção (automática) dos synsets de todas as palavras que não ocorrem nos corpos do AC/DC (Santos and Bick, 2000);

2. Seleção (automática) apenas dos synsets onde todas as palavras têm uma frequência superior a 100, nos mesmos corpos;

3. Escolha (automática), de n pares de palavras, sendo que cada par tem duas palavras provenientes do mesmo synset;

4. Classificação manual de cada par, como correto ou incorreto.

Os dois primeiros passos foram feitos para tornar a avaliação mais rápida e focada em palavras conhecidas, por serem frequentes. No terceiro passo, optamos por gerar três conjuntos aleatórios: 150 pares de nomes, 150 pares de verbos e 150 pares de adjetivos. No quarto passo, cada par foi classificado por dois avaliadores humanos, de forma independente, a quem foi sugerido a consulta de dicionários na rede, em caso de dúvida. A tabela 5.7 apresenta os resultados obtidos por avaliador e a sua concordância κ, assim como os resultados da avaliação manual do CLIP, mas apenas para os nomes, tal como apresentada em (Gonçalo Oliveira, 2013). Apresentam-se ainda as médias das medidas de pertença dos pares classificados como correctos por ambos os avaliadores ($\overline{\mu_c}$), pares onde não houve concordância entre avaliadores ($\overline{\mu_d}$), e pares classificados como incorrectos por ambos ($\overline{\mu_i}$). Não nos foi possível recuperar os dados de avaliação manual do CLIP, o que não nos permite fazer a análise dos graus de pertença para a abordagem anterior.

	CLIP 0,01		**Actual**				
Cat	**Correctos**	κ	**Correctos**	κ	$\overline{\mu_c}$	$\overline{\mu_d}$	$\overline{\mu_i}$
N	75.0%	0.43	84.7-88.0%	0.75	0.30	0.26	0.22
V	N/A	N/A	68.7-68.7%	0.65	0.25	0.19	0.15
Adj	N/A	N/A	74.7-77.3%	0.74	0.17	0.18	0.25

Tabela 5.7: Resultados da avaliação manual e média dos graus de pertença para cada classe de pares de sinonima.

5.4 Utilização de outros recursos

Os resultados apresentados anteriormente constituíram a primeira experiência na aplicação da abordagem proposta a relações de sinonímia extraídas a partir de três dicionários da língua portuguesa. No entanto, relações de outros tipos podem também transmitir informação relevante no cálculo da pertença (confiança) das palavras a synsets. Nesta secção apresentam-se alguns resultados preliminares em que foram considerados, primeiro, relações de hiperonímia e, segundo, relações comuns a vários elementos de um synset base.

5.4.1 Utilização de Hiperónimos

Uma das primeiras experiências onde, para além da utilização das relações de sinonímia, da mesma forma que o anteriormente relatado, foi a utilização de outros tipos de relação, considerando-se também relações de hiperonímia (ex "*animal* hiperónimo de *cão*"e "*animal* hiperónimo de *gato*"), no cálculo da pertença. Este tipo de relação foi escolhido não só por ter muitas instâncias no CARTÃO (cerca de 115 mil, 95 mil distintas), mas principalmente por indicar uma generalização/especificação. Ou seja, hipónimos partilham um conjunto de características com os seus hiperónimos, e por isso podem considerar-se semanticamente próximos, por exemplo as palavras *carro* e *automóvel* são ambos hipónimos

de *veículo*. Existem mesmo várias medidas para o cálculo de similaridade semântica com base nestas relações, na WordNet (por exemplo, (Resnik, 1995) ou (Leacock and Chodorow, 1998)).

Com base nas considerações anteriores, as relações de hiperonímia vão apenas aumentar a pertença em casos em que uma palavra está em relações de sinonímia com algumas das palavras do synset base, mas também em relações de hiperonímia com outras dessas palavras. Estes casos serão, acreditamos, situações em que, na própria linguagem, a utilização do hipónimo e do hiperónimo se confundem e acabam por ser usadas para referir o mesmo. Ao mesmo tempo, num dicionário que já inclua uma relação de sinonímia entre duas palavras, não serão consideradas relações de hiperonímia entre as mesmas palavras. Assim, o peso vindo de cada fonte nunca pode ser superior a 1. Um exemplo de uma relação válida seria num dicionário existir uma relação "*folheto* sinónimo de *panfleto*"e em outro dicionário existir a relação "*folheto* hiperónimo de *panfleto*". Devemos acrescentar que, como nos dicionários utilizados as relações de hiperonímia se estabelecem apenas entre substantivos, esta experiência foi aplicada somente a palavras desta categoria gramatical.

Para confirmar rapidamente que a consideração dos hiperónimos desta forma alterava as pertenças da forma desejada, aproveitamos os dados da avaliação manual anterior. A tabela 5.8 apresenta os valores médios das pertenças de pares de palavras do mesmo synset, primeiro, sem a consideração das relações de hiperonímia e, segundo, quando as relações de hiperonímia são consideradas com um peso que é metade dos das relações de sinonímia. Ou seja, para calcular a pertença, antes de aplicar a equação 4.1, a matriz de adjacências da rede, A, é alterada, de forma a que, sempre que haja uma relação de hiperonímia entre duas palavras $H(P_i, P_j)$, se também existir pelo menos uma relação de sinonímia, $S(P_i, P_j)$, a ligação entre as palavras é reforçada, $A_{ij} += \alpha$.

Peso hiperonímia	$\overline{\mu_i}$	$\overline{\mu_d}$	$\overline{\mu_c}$
0,0	0,21957	0,26132	0,29960
0,1	0,21985	0,2615	0,30041
0,2	0,22012	0,2617	0,30123
0,5	0,22096	0,26234	0,30368
Diferença	0,00139	0,00102	0,00408
Ganho	0,2597	0,0584	0,0320

Tabela 5.8: Diferenças e ganhos nas pertenças médias de pares de sinonímia corretos, discordantes e incorretos utilizando as relações de hiperonímia.

De forma a observar a evolução nas pertenças médias, a tabela 5.8 apresenta também a diferença entre o valor destas antes (peso=0,0) e depois de considerar as relações de hiperonímia (peso=0,5), e mostra ainda o ganho em cada média (equação 5.1). Como apenas os casos em que existiam relações de hiperonímia são valorizados, e não havia mais nenhuma alteração, os valores das pertenças ou se mantinham, ou aumentavam. Ou seja, o ganho seria zero ou positivo. Na tabela 5.8 verifica-se que, apesar do ganho ser sempre positivo, é ligeiramente superior nos casos em que ambos os anotadores concordaram que as duas palavras do par eram sinónimos, ou seja, tornou o valor das pertenças um pouco mais fiel a uma medida de confiança.

$$Ganho = \frac{Valor_{nova} - Valor_{anterior}}{Valor_{anterior}} \tag{5.1}$$

Com base nos valores obtidos, decidimos começar a utilizar também as relações de hiperonímia no cálculo das pertenças aos synsets difusos. A título de exemplo, a tabela 5.9 apresenta três synsets difusos antes e depois de serem consideradas as relações de hiperonímia.

Antes	**Depois**
ramada(0.67), *ramagem*(0.52), *rama*(0.52), *enramada*(0.29), *ramosidade*(0.24), *arramada*(0.19), *fronde*(0.19), *parreira*(0.13), *latada*(0.083), *França*(0.042), *ramaria*(0.042), ***folhagem*(0.042)**	*ramada*(0.67), *ramagem*(0.52), *rama*(0.52), *enramada*(0.29), *ramosidade*(0.24), *arramada*(0.19), *fronde*(0.19), *parreira*(0.13), *latada*(0.083), ***folhagem*(0.063)**, *França*(0.042), *ramaria*(0.042)
panfleto(0.83), *libelo*(0.83), *querela*(0.11), ***folheto*(0.11)**	*panfleto*(0.83), *libelo*(0.83), ***folheto*(0.17)**, *querela*(0.11)
apelido(0.46), ***nome*(0.46)**, ***alcunha*(0.40)**, *cognome*(0.31), ***epíteto*(0.23)**, ***sobrenome*(0.23)**, *designação*(0.17), *denominação*(0.17), *qualificação*(0.15), ...	***nome*(0.48)**, *apelido*(0.46), ***alcunha*(0.41)**, *cognome*(0.31), ***sobrenome*(0.25)**, ***epíteto*(0.24)**, *designação*(0.17), *denominação*(0.17), *qualificação*(0.15), ...
mortalha(1,0), sudário(1,0), **lençol(0,22)**, seda(0,11)	mortalha(1,0), sudário(1,0), **lençol(0,28)**, seda(0,11)

Tabela 5.9: Exemplos de synsets difusos com pertenças das palavras antes e depois de considerar as relações de hiperonímia.

5.4.2 Utilização de relações em comum

Outra experiência realizada procurou tirar partido das relações em comum entre vários elementos do mesmo synset base, como por exemplo "X parte de A"e "X parte de B"considera-se que A e B possuem uma relação em comum, neste caso a relação "parte de"com X.

Mais uma vez para verificarmos se as pertenças são alteradas de forma desejada aproveitamos os dados da avaliação manual. A tabela 5.10 apresenta os valores médios das pertenças de pares de palavras do mesmo synset, primeiro, sem a consideração das relações comuns e, segundo, quando as relações comuns são consideradas com um peso que é de um décimo das relações de sinonímia, foi considerado este valor porque existem muitos tipos de relação, existindo a possibilidade de haver casos em que haja muitas relações em comum, acreditamos com isso que a relevância de relações comuns seja de menor significância que a existência de uma relação de hiperonímia. Ou seja, para calcular a pertença, antes de aplicar a equação 4.1, a matriz de adjacências da rede, A, é alterada, de forma a que, sempre que haja duas palavras que partilhem uma mesma relação com uma terceira$R(P_i, P_j)$, se também existir pelo menos uma relação de sinonímia, $R(P_i, P_j)$, a ligação entre as palavras é reforçada, $A_{ij} += 0.1$.

Peso relações em comum	$\overline{\mu_i}$	$\overline{\mu_d}$	$\overline{\mu_c}$
0,0	0,2195	0,2613	0,2996
0,1	0,2351	0,2766	0,3092
Diferença	0,01553	0,01528	0,0096
Ganho	0,0707	0,00390	0,01362

Tabela 5.10: Diferenças e ganhos nas pertenças médias de pares de sinonímia correctos, discordantes e incorrectos utilizando as relações em comum.

Mais uma vez esta abordagem apenas incrementa valor de pertença quando existem relações comuns entre palavras não havendo nenhuma situação em que a pontuação seja decrementada. Nesta experimentação, ao contrário da utilização das relações de hiperonímia, o desempenho é negativo visto que a pontuação média dos pares considerados como não sinónimos, subiram muito mais quer no valor absoluto quer em proporção como se pode ver no valor de ganho. Ou seja, para já, esta alteração ao algoritmo não foi considerada, ainda que haja várias experiências a realizar para confirmar os seus benefícios ou não, por exemplo, alterar o valor do peso ou considerar diferentes tipos de relação de forma diferente.

Capítulo 6

Conclusões

Com vista à descoberta de conceitos, descritos por conjuntos de palavras com pertenças variáveis, apresentamos uma nova abordagem para a descoberta de synsets difusos através de redes léxico-semânticas. Esta abordagem tira partido da redundância em redes extraídas a partir de várias fontes, neste caso dicionários, por isso o valor da pertença pode, de certa forma, quantificar a confiança na utilização da palavra para se referir ao conceito que emerge do synset.

A abordagem proposta diferencia-se de uma abordagem anterior para o mesmo fim por ser realizada em dois passos e por considerar apenas as adjacências relevantes para o cálculo das pertenças de cada palavra a um synset. Isto diminuiu o ruído e tornou o valor das pertenças mais facilmente interpretável, o que se confirma não só pela avaliação manual de ambas as abordagens, mas também pela comparação do valor das pertenças de diferentes pares de palavras. Como esperado numa medida de confiança, pares de palavras que devem estar no mesmo synset (sinónimos) têm em média uma pertença superior a pares que, de acordo com anotadores humanos, não são sinónimos.

Ainda assim, apesar dos resultados positivos, os valores da avaliação mostram que há ainda muita margem de melhoria. Por exemplo, enquanto cerca de 88% dos pares de substantivos pertencentes ao mesmo synset são efetivamente sinónimos, para os verbos, este número desce para 68%.

O recurso resultante deste trabalho será uma wordnet para a língua portuguesa, criada de forma automática, e em que haverá valores de confiança associados a algumas das decisões tomadas, incluindo não só a inclusão de palavras em synsets, como também o estabelecimento de relações entre synsets, que será uma das próximas fases do trabalho. Acreditamos que este recurso, a ser disponibilizado em breve, possa ser de grande utilidade para aqueles que procuram uma wordnet para o português em que o balanço entre cobertura e confiança possa ser personalizado de acordo com as necessidades da aplicação.

Trabalho Futuro

Nos próximos passos a realizar neste âmbito, pretendemos realizar novas experiências para averiguar a melhor forma de considerar outros tipos de relação. Por exemplo, uma ideia a seguir é que palavras sinónimas devem estar relacionadas da mesma forma com as mesmas palavras (por exemplo, tanto *carro*, como *automóvel* devem ser hipónimos de *veículo* e ter como partes *roda* ou *motor*).

Devido aos testes já descritos no capítulo da abordagem proposta, os resultados

focaram-se essencialmente nas estruturas obtidas com o CW. Porém seria interessante executar a abordagem descrita neste documento, a outros algoritmos de strict clustering como por exemplo o MCL e o CBC, entre outros.

Como referido, já foi feito algum trabalho na exploração de palavras com relações comuns, no entanto consideramos que ainda há margem para melhorar os resultados, nomeadamente o estudo de quais as relações mais importantes a considerar e/ou a atribuição de pesos diferenciados a cada tipo de relação.

A abordagem aqui proposta poderá ser aplicada a outras redes de sinonímia. Um dos exemplos seria a inclusão do recurso criado manualmente TeP, que é aqui utilizado, mas neste caso como recurso linguístico de entrada e não como recurso para avaliação. Outro exemplo de recurso a ser adicionado seria a utilização de outras wordnets para a língua portuguesa, tais como por exemplo o TeP (Maziero et al., 2008), a OpenWN-PT (de Paiva et al., 2012) ou a PULO (Simões and Guinovart, 2014).

O próximo passo seria classificar as relações entre synsets com medidas de pertença, ou seja integrando esta abordagem na terceira fase da abordagem ECO a de Ontologisação.

A ter em conta, como referido em (Biemann, 2006), é o facto do CW não ser totalmente determinístico. Neste sentido, poderá ser interessante correr o algoritmo diversas vezes de modo a avaliar os diferentes resultados. Ou, mesmo utilizar o número de vezes em que cada palavra é incluída em cada synset como característica no cálculo da sua pertença.

Os resultados desta abordagem serão ainda disponibilizados livremente à comunidade, na página do Onto.PT[1].

Contribuições resultantes

Após a realização deste trabalho, resultou uma nova abordagem que permite através da utilização de redes extraídas de dicionários, o agrupamento de palavras em conjuntos de elementos que partilham um mesmo significado, avaliando o valor de proximidade ao significado do conjunto.

Para a avaliação desta abordagem foram gerados datasets compostos por pares de palavra e a classificação atribuída por dois avaliadores, estes datasets estão construidos e podem ser utilidados como meio de avaliação de outras abordagens com objetivos semelhantes.

Em paralelo à realização deste documento foi redigido um artigo que está aprovado para publicação na revista Linguamática, que tem como tema o processamento automático das línguas ibéricas. O nosso artigo resume os passos principais da abordagem presente neste documento.

Do trabalho executado ao longo de vários meses resulta este mesmo documento, que descreve uma abordagem para a descoberta de synsets difusos, apresentando e descrevendo alguns dos testes realizados demonstrando os seus resultados, e conclusões inferidas. É ainda contextualizado alguns conceitos e trabalhos relacionados com o tema aqui apresentado.

[1] `http://ontopt.dei.uc.pt/`

Bibliografia

Biemann, C. (2006). Chinese Whispers: An efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, TextGraphs-1, pages 73–80, Stroudsburg, PA, USA. ACL Press.

Collovini de Abreu, S., Bonamigo, T. L., and Vieira, R. (2013). A review on relation extraction with an eye on portuguese. *Journal of the Brazilian Computer Society*, 19(4):553–571.

de Paiva, V., Rademaker, A., and de Melo, G. (2012). OpenWordNet-PT: An open brazilian wordnet for reasoning. In *Proceedings of 24th International Conference on Computational Linguistics*, COLING (Demo Paper).

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41:391–407.

Dorow, B. (2006). *A Graph Model for Words and their Meanings*. PhD thesis, Institut fur Maschinelle Sprachverarbeitung der Universitat Stuttgart.

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

Firth, J. R. (1957). A synopsis of linguistic theory 1930-55. 1952-59:1–32.

Gfeller, D., Chappelier, J.-C., and De Los Rios, P. (2005). Synonym Dictionary Improvement through Markov Clustering and Clustering Stability. In *Proceedings of International Symposium on Applied Stochastic Models and Data Analysis*, ASMDA 2005, pages 106–113, Brest, France.

Gonçalo Oliveira, H. (2013). *Onto.PT: Towards the Automatic Construction of a Lexical Ontology for Portuguese*. PhD thesis, University of Coimbra.

Gonçalo Oliveira, H., Antón Pérez, L., Costa, H., and Gomes, P. (2011). Uma rede léxico-semântica de grandes dimensões para o português, extraída a partir de dicionários electrónicos. *Linguamática*, 3(2):23–38.

Gonçalo Oliveira, H., Costa, H., and Gomes, P. (2010). Extracção de conhecimento léxico-semântico a partir de resumos da Wikipédia. In *Actas do II Simpósio de Informática*, INFORUM 2010, pages 537–548, Braga, Portugal. Universidade do Minho.

Gonçalo Oliveira, H. and Gomes, P. (2011). Automatic Discovery of Fuzzy Synsets from Dictionary Definitions. In *Proceedings of 22nd International Joint Conference on Artificial Intelligence*, IJCAI 2011, pages 1801–1806, Barcelona, Spain. IJCAI/AAAI.

Gonçalo Oliveira, H. and Gomes, P. (2014). ECO and Onto.PT: A flexible approach for creating a Portuguese wordnet automatically. *Language Resources and Evaluation*, 48(2):373–393.

Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of 14th Conference on Computational Linguistics*, COLING 92, pages 539–545, Morristown, NJ, USA. ACL Press.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, Englewood Cliffs, NJ, 2nd edition.

Kilgarriff, A. (1996). Word senses are not bona fide objects: implications for cognitive science, formal semantics, NLP. In *Proceedings of 5th International Conference on the Cognitive Science of Natural Language Processing*, pages 193–200.

Leacock, C. and Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. In Fellfaum, C., editor, *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*, pages 265–283, Cambridge, Massachusetts. The MIT Press.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, pages 768–774, Morristown, NJ, USA. Association for Computational Linguistics.

Lin, D. and Pantel, P. (2002). Concept discovery from text. In *Proceedings of 19th International Conference on Computational Linguistics*, COLING 2002, pages 577–583.

Marrafa, P., Amaro, R., and Mendes, S. (2011). WordNet.PT Global – extending WordNet.PT to Portuguese varieties. In *Proceedings of 1st Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*, pages 70–74, Edinburgh, Scotland. ACL Press.

Maziero, E. G., Pardo, T. A. S., Felippo, A. D., and Dias-da-Silva, B. C. (2008). A Base de Dados Lexical e a Interface Web do TeP 2.0 - Thesaurus Eletrônico para o Português do Brasil. In *VI Workshop em Tecnologia da Informação e da Linguagem Humana (TIL)*, pages 390–392.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Nasiruddin, M. (2013). A state of the art of word sense induction: A way towards word sense disambiguation for under resourced languages. *TALN/RECITAL 2013*.

Navarro, E., Sajous, F., Gaume, B., Prévot, L., Hsieh, S., Kuo, T. Y., Magistry, P., and Huang, C. R. (2009). Wiktionary and NLP: Improving synonymy networks. In *Proceedings of Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 19–27, Suntec, Singapore. ACL Press.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69.

Pantel, P. and Pennacchiotti, M. (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of 21st International*

Conference on Computational Linguistics and 44th annual meeting of the Association for Computational Linguistics, pages 113–120, Sydney, Australia. ACL Press.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA.

Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice-Hall.

Santos, D. (1992). Natural Language and Knowledge Representation. In *Proceedings of the ERCIM Workshop on Theoretical and Experimental Aspects of Knowledge Representation*, pages 195–197.

Santos, D. and Bick, E. (2000). Providing Internet access to Portuguese corpora: the AC/DC project. In *Proceedings of 2nd International Conference on Language Resources and Evaluation*, LREC 2000, pages 205–210.

Santos, F. and Gonçalo Oliveira, H. (2015). Descoberta de synsets difusos com base na redundância em vários dicionários. *Linguamática*, page aceite para publicação.

Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review*, 1(1):27–64.

Simões, A. and Guinovart, X. G. (2014). Bootstrapping a portuguese wordnet from galician, spanish and english wordnets. *Advances in Speech and Language Technologies for Iberian Languages*, 8854:239–248.

Snow, R., Jurafsky, D., and Ng, A. Y. (2005). Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems*, pages 1297–1304. MIT Press, Cambridge, MA.

Snow, R., Prakash, S., Jurafsky, D., and Ng, A. Y. (2007). Learning to Merge Word Senses. In *Proceedings of the Joint Meeting of the Conference on Empirical Methods on Natural Language Processing and the Conference on Natural Language Learning*, pages 1005–1014.

Turney, P. D. (2001). Mining the Web for synonyms: PMI–IR versus LSA on TOEFL. In *Proceedings of 12th European Conference on Machine Learning, ECML 2001*, volume 2167 of *LNCS*, pages 491–502. Springer.

van Dongen, S. M. (2000). *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht.