Rui Filipe dos Santos Rodrigues

Métrica de dissimilaridade semântica baseada na wikipédia

Dissertação de Mestrado em
Informática de Gestão

Novembro de 2015

Agradecimentos

Agradeço em primeiro lugar à instituição Escola Superior de Tecnologia de Setúbal por ter disponibilizado as condições mínimas necessárias para a preparação do presente trabalho.

Ao Professor Joaquim Filipe, por ter proposto, orientado e apoiado no decorrer dos trabalhos, apresentando sempre um grande entusiasmo, ambição e esperança nas ideias aqui apresentadas.

Ao meu grupo de trabalho profissional por ter discutido inúmeras ideias, desta forma contribuindo para diversas melhorias ao nível da eficiência e eficácia da solução.

Por fim, um agradecimento especial a toda a minha família por toda a motivação e apoio durante os últimos meses. Em particular à Susana, que acompanhou de perto todas as dificuldades.

Resumo

Não obstante a vasta quantidade de informação disponibilizada nem sempre é fácil obter o conhecimento que se almeja alcançar, devido à dificuldade de catalogar a informação. Os sistemas de "descoberta de conhecimento" atuais centram-se na procura de palavras idênticas, podendo aqui observar-se variadas limitações, entre elas a falta de capacidade de interpretação. A compreensão do significado semântico do conjunto de expressões é uma característica do ser humano, sendo esta difícil de replicar em sistemas computacionais. O objetivo principal deste trabalho consiste na criação de um sistema de cálculo de semelhança semântica entre classes abstratas, sistema esse que deve possuir por base uma ontologia de conhecimento. Para atingirmos o objetivo proposto começou-se por identificar e analisar a necessidade de uma máquina conseguir simular ou melhorar a apreciação do ser humano relativamente à interpretação semântica. Após a definição e enquadramento do problema na área de conhecimento respetiva partiu-se para a criação do sistema capacitado de calcular uma medida de semelhança entre entidades, tendo em consideração a importância que o desempenho apresenta neste tipo de sistema.

Palavras-chave: Descoberta de Conhecimento, Ontologias, Base de Conhecimento, Similaridade Semântica, Entidades, Categorias da Wikipedia.

Abstract

Despite the vast amount of information available it is not always easy to obtain the knowledge that we aim to achieve because of the difficulty of cataloging information. Current systems of "knowledge discovery" focus in the search for identical words, which can have various limitations, including the lack of interpretation. An understanding of the semantic meaning of the set of expressions is a characteristic of the human being, which is difficult to replicate in computer systems. The main objective of this work is to create a semantic similarity calculation system between abstract classes. This system should have a knowledge based ontology. To achieve the proposed objective we started by identifying and analyzing the need for a machine to simulate or enhance the appreciation of the human being with regard to semantic interpretation. After defining the problem in the respective area of expertise we started to create a system capable of calculating a similarity measure between entities, taking into account the importance that the performance features in this type of system.

Keywords: Knowledge Discovery, Ontologies, Knowledge Base, Semantic Relatedness, Entities, Wikipedia Categories.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Siglas e Acrónimos

API	*Application Programming Interface*
LCS	*List Common Subsumer*
PLN	*Processamento de Linguagem Natural*
REGEX	*REGular EXpression*
SQL	*Structured Query Language*
XML	*eXtensible Markup Language*

Capítulo 1

Introdução

Atualmente pode-se observar a geração de grandes quantidades de informação à escala global a uma velocidade impressionante, sendo essa informação produzida pelas mais variadas fontes e disseminada eletronicamente por todo o mundo. Esta quantidade impressionante de dados inunda o dia-a-dia de qualquer ser humano, tornando complicada a tarefa de se chegar ao conhecimento que se pretende obter. Apesar de parecer uma tarefa trivial, esta apresenta-se no entanto como um exercício extremamente complexo. Cada entidade possui as suas preferências e os seus gostos, isto é, a entidade é caracterizada por um determinado conjunto de atributos. Assim sendo, determinar se duas entidades são próximas corresponde a descobrir se os seus atributos se encontram próximos em termos semânticos.

No estudo das comunicações, a semântica integra uma das vias da investigação dos símbolos e sinais usados por agentes, mais conhecido por semiótica. De acordo com Charles Sanders Peirce, a semiótica pode ser definida como o estudo dos sinais e dos seus significados, palavras, gestos, sons, imagens e objetos. Existem três principais campos de investigação na semiótica, nomeadamente sintática (a ordem e relação entre sinais), pragmática (forma como são usados e interpretados) e semântica (relação entre os sinais e o contexto que são usados). O processo de computação da similaridade semântica é um dos tópicos mais fascinantes em desenvolvimento nas áreas de inteligência artificial e processamento de linguagem. É um problema com alguns anos de existência, sendo que as primeiras investigações científicas neste tema datam dos anos 70 (Charniak, 1973). Vários sistemas de motores de procura simulam este tipo de comportamento há algum tempo, embora se centrem sempre na similaridade sintática entre as expressões de procura e páginas indexadas. Este mecanismo peca no entanto pela ignorância do significado da frase (compreende-se por frase o sentido gerado pelo conjunto das palavras individuais como um todo).

A determinação de similaridade semântica entre entidades pode ser categorizada em duas principais metodologias, nomeadamente a avaliação da similaridade através do conteúdo da informação e a procura de similaridade com recurso a base de conhecimento externa. No caso do primeiro método, pode-se dar o exemplo de comparação entre dois artigos, sendo que a similaridade é medida estatisticamente pelo número de expressões que partilham. Neste momento existem diversas técnicas bastante consolidadas que têm por base a análise sintática, sendo que a contagem de termos e a aplicação do “term frequency, inverse term frequency” (tf-idf) é das mais utilizadas (Salton & Buckley, 1988). Conceptualmente, pretende-se refletir o quanto uma palavra é importante para um documento numa coleção de documentos, sendo que esta técnica é de importância fundamental para

sistemas de “Information Retrieval” e “text mining”. A segunda opção possui como componente central uma ontologia, isto é, existe uma grande dependência em termos de eficácia/eficiência de resultados por parte da aplicação em relação à base de conhecimento utilizada. Numa ontologia aproximada à realidade deve-se ter em conta que por vezes existam imperfeições em termos de exploração das mais diversas áreas, visto que qualquer base de conhecimento deste género não é mais do que um modelo de aproximação à própria realidade.

Cada ponto da ontologia é representado por uma entidade, na qual esta pode ser definida como um objeto abstrato e genérico que tem como propriedades os seus tópicos, ou seja, toda e qualquer entidade é caracterizada pelo seu conjunto de temas. Ao efetuar a comparação semântica de duas entidades, procede-se ao cálculo das distâncias conceptuais entre todos os seus tópicos, obtendo um resultado final. Tomemos o seguinte exemplo: assumindo que se tem por entidade uma pessoa que seria definida pelos seus gostos pessoais (tópicos) e uma segunda entidade que seria uma revista caracterizada pelas suas áreas. Ao calcular as distâncias entre tópicos de E1 e áreas de E2 conseguiríamos saber se a pessoa irá gostar da revista ou não.

1.1. Definição do Problema

O volume de dados disponível na internet está a crescer exponencialmente, estimando-se que circularam mais de 500 mil Petabytes de dados só no ano de 2014 (CISCO, 2014). Trata-se portanto de uma elevada circulação de notícias, artigos, correio eletrónico, anúncios, etc.. Hoje em dia os sistemas de “descoberta de conhecimento” centram-se na procura de palavras idênticas, podendo aqui observar-se variadas limitações, entre elas a falta de capacidade de interpretação. A compreensão do significado semântico do conjunto de expressões é uma característica do ser humano, sendo esta difícil de replicar em sistemas computacionais. O exemplo mais notório deste cenário consiste na utilização de sinónimos em frases diferentes, que conceptualmente significam exatamente o mesmo. Torna-se ainda mais complexo se assumirmos a existência de palavras que têm diferentes significados, consoante o contexto que se inserem.

A análise semântica consiste em interpretar o significado de elementos sintáticos, ou seja, corresponde à interpretação de conjuntos de palavras deduzindo o seu significado conceptual. O ser humano apresenta facilidade em executar este tipo de tarefas graças à sua experiência, capacidade de julgamento e conhecimento do ambiente que o rodeia. Este tipo de características é absorvido pelo ser humano ao longo da sua formação e desenvolvimento, podendo-se comparar a uma base de dados de conhecimento acumulado ao longo da vida. Para os humanos é tipicamente complicado expor as suas ideias através de uma só expressão. Em vez disso, são elaboradas frases que são absorvidas e transformadas num significado que se situa num nível de abstração mais elevado. As palavras são

pequenas unidades fragmentadas, que em conjunto com a capacidade de interpretação, transformam-se em informação, permitindo assim a transmissão de ideias entre seres humanos.

Bastante trabalho já foi desenvolvido no que diz respeito ao cálculo de similaridade baseado puramente em análise estatística. Contudo, este tipo de abordagem menospreza a importância do conhecimento das ligações e o relacionamento entre as entidades existentes no nosso dia-a-dia, o que poderá ser um ponto-chave para o aumento de qualidade de resultados. (Deerwester et al., 1990).

O armazenamento organizado de grandes quantidades de conhecimento é, por si só, uma tarefa extremamente complexa. Nem a mais completa enciclopédia do mundo contempla toda a informação sobre tudo o que existe no nosso quotidiano. Analisando os números da Enciclopédia Britannica, provavelmente a obra mais completa sobre a atividade e saber humano, conclui-se que possuí 44 milhões de palavras distribuídas por 65 mil artigos na sua última versão. A categorização e relacionamento de toda esta informação de uma forma lógica para o ser humano é uma tarefa que exige um amplo conhecimento sobre todos os domínios de aplicação. A dificuldade mencionada relaciona-se com a criação e utilização correta de uma ontologia, ou seja, uma descrição de conceitos e relações que podem existir formalmente para um agente ou comunidade de agentes de grandes dimensões, na medida em que processar, relacionar e extrair informação em tempo útil, poderão revelar-se um obstáculo complexo.

A aproximação através da correspondência da língua Inglesa é um projeto relativamente conhecido, denominado por WordNet, e que corresponde a uma conhecida base de dados léxica que descreve a estrutura e relaciona termos linguísticos entre si. Este projeto recorre a uma técnica de agrupamento de sinónimos em grupos de significado, efetuando ligações entre estes respetivos conjuntos. O projeto está a ser conduzido por pessoas altamente especializadas na língua inglesa, em detrimento de pessoas especializadas no domínio de aplicação de cada uma das áreas de conhecimento das nossas ciências. Utilizando este tipo de ferramenta, consegue-se uma diminuição de um dos grandes problemas do processamento de linguagem, nomeadamente a desambiguação do significado de palavras através da identificação de sinónimos, hiperónimos e definindo métricas com bases nestes. Este tipo de recursos porém tem uma orientação direcionada ao significado da palavra como uma entidade individual, ignorando o conhecimento do conjunto agrupado de entidades em geral.

Devido à enorme quantidade de informação existente no que toca a efetuar a correta correspondência de conhecimento e também a impossibilidade de reunir pessoas altamente especializadas em todas as áreas para nos ajudar, conclui-se que a utilização de uma ontologia imperfeita e em desenvolvimento será o caminho a seguir. Deve-se também ter em conta que a realidade está em constante mutação e evolução, portanto a ontologia necessita de acompanhar este fenómeno.

Cenários como o desenvolvimento pormenorizado de uma determinada área de conhecimento, por especialistas de um determinado domínio em detrimento de outras áreas, poderá também gerar imperfeições que não correspondentes à realidade.

Tendo em conta que em todas as línguas existem expressões que podem ter mais do que um significado, tal pode ser um problema perante a necessidade de calcular a relação da distância semântica entre entidades. Tomando como exemplo o tópico Washington: pode-se estar a falar de uma cidade, de um estado ou até de uma pessoa. Não basta possuir o mecanismo de identificação deste tipo de situações. É necessário possuir inteligência suficiente para depois de se deparar com a tentativa de desambiguação, ter a capacidade de optar pela propriedade mais adequada ao contexto.

Posto isto, torna-se essencial realçar a necessidade de facilidade em termos da utilização da ferramenta, visto que o utilizador não irá querer esperar mais do que alguns momentos para saber o resultado da semelhança entre as suas entidades. Tendo em conta que os cálculos poderão ser bastante complexos, a melhoria de resultados sem sacrifício do tempo de espera é um desafio constante. Também se espera uma interface de utilização compatível com qualquer tipo de sistema, visto que as mais diversas aplicações podem necessitar de fazer uso das vantagens de obter distância semântica entre entidades. Aplicações clientes apenas se têm de preocupar em enviar entidades caracterizadas com base em tópicos de interesse.

1.2. Objetivos

Tendo em vista a resposta à problemática exposta na secção anterior, foram definidos diversos objetivos que se irão tentar atingir no decorrer do desenvolvimento deste trabalho.

Em primeiro lugar, pretende-se investigar e consolidar a definição de entidade no contexto deste trabalho. Seguidamente irá propor-se a criação de um sistema de cálculo de semelhança semântica entre entidades. Este sistema deve ter como base uma ontologia de conhecimento. No entanto, devido à impossibilidade de mobilizar todos os especialistas necessários para definir uma ontologia do domínio de aplicação, a Wikipédia será alvo deste estudo. O sistema deverá permitir a navegação entre nós da ontologia e retornar informação sobre os respetivos caminhos. Este caminho deverá ser pós-processado tendo em vista diversos parâmetros que nos permitam reduzir a incerteza e que sejam facilmente ajustáveis. O sistema deverá possuir a capacidade de receber pedidos num formato *standard* a definir. Deverá ser efetuada a recolha da base de dados operacional da ontologia, utilizando de seguida as ferramentas de navegação pelas suas páginas e consequentes categorias. Devem ser discutidas, sempre que possível, oportunidades de melhorias aos tempos de processamento. Deverão também ser alvo de estudo e análise diversas técnicas de aperfeiçoamento da ontologia, tais como:

- Nós vizinhos partilhados;
- Cálculo de representantes de tópicos demasiado específicos;
- Densidade da região onde foi encontrado um determinado caminho;
- Distância constante à categoria raiz;
- Mecanismo de desambiguação de expressões.

Todas estas oportunidades deverão ser estudadas, analisadas e colocadas com o respetivo peso no cálculo da medida de similaridade semântica entre entidades, com base na ontologia. O mecanismo deverá permitir a afinação do peso de todos os componentes que fazem parte da medida de uma forma simples, sem ser necessário voltar a recompilar o sistema para uma nova bateria de testes. Todas as variáveis da medida terão de ser estudadas, visto existir a necessidade de uma justificação para o seu uso. Naturalmente serão utilizados variados conjuntos de testes reais, entre os quais se encontram conjuntos anteriormente validados por especialistas dos respetivos domínios.

Deverá ser arquitetado um mecanismo de atualização constante da base de conhecimento, consistindo nos seguintes passos:

- Verificação diária de novas versões, através de um agente;
- Efetuar *downloads* da base de dados atualizada;
- Importação dos dados;
- Logo que a nova instância da base de conhecimento esteja ativa, pretende-se que a ligação do programa seja automaticamente redirecionada.

1.3. Principais Contribuições

O presente trabalho tem como intuito explorar diversas soluções nas áreas da computação da relação semântica, incluindo os seguintes aspetos:

- Criação de condições para a utilização de uma ontologia como base de conhecimento, navegação nos nós da sua ontologia e identificação de caminhos entre os respetivos nós.
- Conceção e desenvolvimento de uma medida que auxilie no aumento da qualidade do cálculo das distâncias entre as entidades.
- Desenvolvimento de mecanismo suficientemente capaz para comunicar com qualquer tipo de sistema, seja em que tecnologia for.

1.4. Metodologia de Desenvolvimento

A elaboração deste trabalho de mestrado começou por identificar e analisar a necessidade atual de fazer com que uma máquina consiga simular uma correta apreciação do ser humano ou mesmo melhorar essa apreciação em alguns casos, no que diz respeito à interpretação semântica.

Definido o problema, será necessário enquadrá-lo nas suas respetivas áreas, neste caso: reconhecimento de padrões, processamento de linguagem natural e ontologia de entidades, efetuando para tal uma investigação no mundo científico sobre os últimos desenvolvimentos e avanços deste tipo de temáticas. Identificou-se a tendência de muitos artigos, cujos autores investiram em técnicas que têm por base sistemas de armazenamento de conhecimento, como por exemplo a wikipédia.

Decidiu-se avançar com a criação de um sistema capaz de calcular uma medida de semelhança. Tendo sempre em atenção que o desempenho é um dos fatores mais importantes para o sucesso do trabalho.

Será necessária, numa fase posterior, proceder à validação de resultados, preferencialmente com conjuntos conhecidos e contra o ser humano especialista do domínio. Trata-se de um passo bastante importante na medida em que o objetivo máximo será desenvolver um sistema capaz de calcular a medida de semelhança e usá-la para automatizar tarefas que hoje são realizadas por pessoas.

1.5. Estrutura do Documento

Torna-se necessário proceder à descrição da estrutura deste trabalho, que será apresentado ao longo de seis capítulos.

No capítulo de introdução é apresentado o trabalho, assim como o conceito onde este se insere, problemática, motivação e investigação. Também são descritos os objetivos a que nos propomos e a própria metodologia e estrutura do trabalho.

O capítulo 2 tem como objetivo demonstrar as várias áreas científicas de suporte a este trabalho, explicando sucintamente a sua utilidade como base de estudo da computação da similaridade. A abordagem utilizada começa por expor tópicos de alto nível, como processamento de linguagem natural e reconhecimento de padrões, e seguidamente aprofundar algumas técnicas que nos pareceram importantes para a evolução do projeto.

É no capítulo 3 que se investiga a semântica, visto ser parte essencial para este trabalho, começando por recorrer à área da semiótica como base teórica. Seguidamente entramos no tópico das ontologias, inicialmente numa abordagem mais filosófica, e posteriormente aplicada às tecnologias de informação. Terminamos com a análise prática de duas das principais ontologias de suporte a sistemas.

O capítulo 4 é seguramente o mais importante deste trabalho. É aqui que explicamos todas as técnicas implementadas para a resolução dos problemas propostos anteriormente, assim como, se apresenta uma explicação do próprio software a desenvolver. Para ta, é traçada uma explicação dos próprios algoritmos recorrendo a ilustrações, de forma a reduzir a complexidade de algumas das técnicas.

No capítulo 5 demonstram-se os resultados pormenorizadamente, recorrendo a comparação com resultados facultados pelo especialista. Pretende-se apresentar também a evolução dos resultados através da introdução das diversas técnicas explicadas no capítulo anterior.

O sexto capítulo apresenta as conclusões retiradas a partir deste projeto e descreve o trabalho que poderá ser realizado no futuro para melhorar o sistema.

Capítulo 2

Computação de Similaridade

Medidas de similaridade têm um papel importante no processamento de linguagem natural. Contudo, diversos trabalhos nestas áreas demonstram uma evidente necessidade de evolução, visto que nos dias de hoje medir robustamente a distância entre duas palavras é ainda uma tarefa bastante árdua (Bollegala, Matsuo, & Ishizuka, 2007). O estudo da similaridade entre palavras já faz parte integral do processamento natural de linguagem praticamente desde o início do seu estudo. A similaridade entre entidades pode mudar ao longo do tempo e em diferentes domínios de aplicação. Tomemos como exemplo, a palavra “apple”, a qual é frequentemente associada a computadores no contexto da tecnologia, embora esta associação não seja feita na maioria dos dicionários. Novas palavras são criadas todos os dias, assim como novos significados são atribuídos a palavras já existentes.

Na abordagem sintática, pode-se dar o exemplo mais comum em termos de programação, que consiste no uso de “Regular Expressions”, para identificação de padrões básicos. Outras soluções simples são também a correspondência sintática entre os dois conjuntos, calculando seguidamente uma pontuação baseada no número de palavras que ocorrem nos dois textos. Ao longo dos anos foram propostas várias ideias que melhoram este tipo de abordagem, tais como, o “stemming”, remoção de “stopwords” e “speech-tagging”. Em reconhecimento de padrões existem dois principais caminhos para busca de informação e classificação de conjuntos de dados. Na aprendizagem supervisionada disponibiliza-se ao algoritmo informação devidamente etiquetada, ou seja, exemplos de treino que consistem no objeto e input e o seu respetivo output. Na aprendizagem não supervisionada tenta-se encontrar uma estrutura desconhecida em dados não etiquetados, sendo esta técnica conhecida como “clustering”. Existem diferentes níveis de dificuldade na busca pela informação e pelo conhecimento, dependendo estes também da estrutura dos dados. No caso do “Data Mining” estamos num contexto de informação estruturada, normalmente bases de dados SQL. Se se quiser ir mais longe e retirar informação de dados não estruturados, entramos no universo do “Information Retrieval”, onde o exemplo mais comum são os motores de busca na internet. Transformação de informação não estruturada em dados estatísticos é um dos caminhos mais comuns, para que seja possível a aplicação dos algoritmos de “data mining”. Este tipo de técnicas passam pela criação de vetores “bag of words” para cálculo de frequência de termos, assim como a frequência inversa “TF-IDF”. Com esta informação podem-se aplicar, por exemplo, fórmulas simples para cálculo de distâncias.

2.1. Processamento de Linguagem Natural

Linguagens naturais são uma ferramenta que evoluiu naturalmente ao longo dos tempos sendo usadas pelos seres humanos com o propósito de comunicar. Assim, o português e o inglês são exemplos de linguagens naturais. O processamento de linguagem natural é uma subárea das ciências da computação e da própria linguística que trata da análise da linguagem natural com a ajuda de computadores. O processamento de linguagem natural é neste momento considerado um subdomínio da inteligência artificial. O termo linguagem natural foi empregue com o objetivo de criar uma distinção entre linguagens humanas (Português, Inglês) e linguagens de programação (c-sharp, java).

A habilidade de um computador processar linguagem natural tão bem como um ser humano é sinal de uma máquina inteligente. A base desta teoria é o facto do uso da linguagem para comunicar pelo ser humano estar diretamente ligado à sua capacidade cognitiva. Alan Turing foi das primeiras pessoas a aperceber-se deste facto, o que pode ser comprovado no "Teste de Turing" onde foi sugerido julgar a inteligência de um computador através de um jogo entre Humanos e Máquina: se um humano (C) não conseguir distinguir entre outro humano (B) e uma máquina (A) através de perguntas em linguagem natural num determinado período de tempo, então conclui-se que esta máquina é inteligente. Turing previu que pelo no final do século XX, uma máquina com dez gigabytes de memória teria cerca de 30% de hipóteses de enganar um humano durante 5 minutos. Podemos ver uma ilustração do teste de Turing na figura seguinte (Turing, 1950).

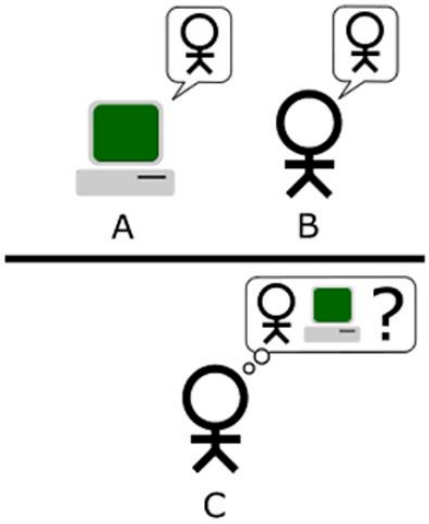

Figura 1-Teste de Turing (extraído de (Wikipédia, 2015)).

A maior parte da comunicação entre humanos ao longo da sua vida ocorre através de discurso oral, sendo que a comunicação escrita aparece logo a seguir, em termos de importância. Mas o processamento de linguagem escrita, apesar de complexo, apresenta menos desafios que o processamento de linguagem oral. Logo torna-se útil dividir o problema em dois níveis: num nível, compreender o texto escrito usando recursos sintáticos e semânticos da linguagem, assim como

informação do mundo real; no outro nível, a compreensão da linguagem falada, usando todos os recursos da escrita juntando o conhecimento adicional de fonologia e ambiguidades de discurso. Nos últimos anos a qualidade do processamento de linguagem natural tem vindo a aumentar sobretudo devido ao consenso de que as “features” semânticas têm mais importância do que se pensava. Também em relação aos contextos, considera-se agora de extrema importância na interpretação do texto, o conhecimento do domínio que é frequentemente aplicado pelas pessoas para a correta compreensão de mensagens. Isto é, os ingredientes apropriados para a extração de significado estão muitas vezes fora das palavras da frase que estamos a tentar interpretar (Kumar E. , 2011).

Os principais objetivos do processamento de linguagem natural são:

- Interfaces de linguagem natural para bases de dados
- Sistemas de tradução automática para linguagem máquina
- Sistemas de análise de textos
- Sistemas de reconhecimento da fala

Seguidamente apresentam-se as principais tarefas do processamento de linguagem natural:

- Resumo automático de textos
- Ajuda na interação com línguas desconhecidas
- Extração de informação
- Recuperação de informação
- Tradução entre linguagens naturais
- Reconhecimento automático de entidades (Países, cidades)
- Geração de linguagem natural
- Compreensão de linguagem natural
- Capacidade de resposta a perguntas em linguagem natural
- Reconhecimento da fala
- Capacidade de transformar texto escrito numa conversa oral
- Verificação de texto
- Análise de opinião e sentimento
- Desambiguação

A grande generalidade dos programas de processamento de linguagem natural têm uma estrutura semelhante à apresentada na figura abaixo. Num sistema deste género os “inputs” são dados em linguagem natural, seguidamente tratados pelo “parser” e generalizados numa estrutura sintática da frase. O interpretador de semântica capta os detalhes semânticos gerados pela frase estruturada e torna-a compatível com a estrutura de base de dados. O processo ao contrário é usado na geração de linguagem natural. Na figura seguinte é observável um exemplo de um sistema de PLN.

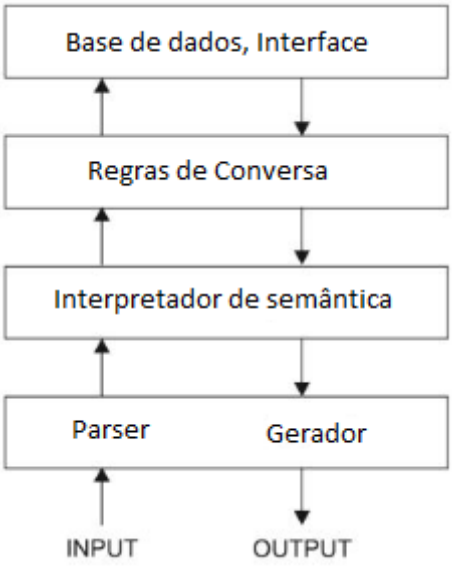

Figura 2- Sistema de PLN.

Para desenvolver um sistema de processamento de linguagem natural, a implementação do parser tem que ser feita numa linguagem adequada. As linguagens de programação tendem a ser específicas e livres de contexto, sendo que desta forma podem-se escrever soluções rápidas e bastante eficientes.

2.2. Processamento de Texto

Numa abordagem mais generalizada, o processamento de texto é simplesmente o efetuar de uma qualquer operação com uma dada informação textual. Pode ser uma restruturação, uma reformatação, extração de informação, ou mesmo efetuar cálculos tendo por base a informação contida. O texto é um conjunto de dados que estão num formato em que o ser humano, devidamente instruído, consegue facilmente convertê-lo em informação (Mertz, 2003).

O processador de texto mais comum é provavelmente o editor de texto, que está presente em todos os computadores hoje em dia. Este software foi desenvolvido com o intuito de facilitar a vida dos seus utilizadores num grande número de tarefas de processamento de texto, tais como copiar e colar texto, procura de palavras e até mesmo tarefas mais complexas, como por exemplo a execução de “macros”.

Processamento de texto é uma das principais tarefas dos programadores de software. A informação que está contida nos sistemas de software é em grande parte composta por texto do domínio de aplicação e guardada em bytes nas respetivas bases de dados de suporte.

2.2.1. Expressões Regulares

No contexto deste trabalho, uma expressão regular é um tipo de texto padrão específico que pode ser usado em várias linguagens de programação modernas. Pode usar-se para determinar se o conteúdo de uma caixa de texto cumpre as regras necessárias, encontrar texto que tem um determinado padrão num documento extenso, efetuar trocas de texto que tenham um determinado padrão ou partir um documento em blocos de subtextos.

O termo ”Expressão Regular” deriva da matemática e da teoria das ciências da computação, onde reflete um conceito matemático conhecido por “regularidade”, estudado e formulado na álgebra de Kleene (Kozen, 1991) . Estas expressões podem ser implementadas recorrendo a “software” usando técnicas de máquinas de estados finitas determinísticas. Uma representação de uma máquina de estados finita encontra-se na figura seguinte (Wikipedia, 2015).

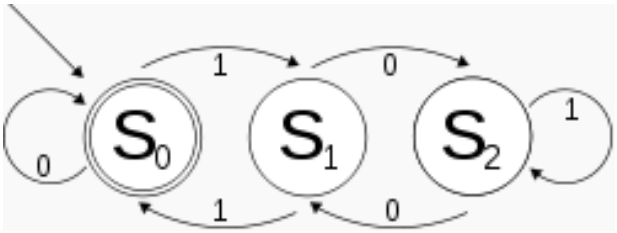

Figura 3 - Máquina de estados finita determinística que aceita números binários, sendo o estado S0 o estado inicial (extraído de (Wikipedia, 2015).

Os padrões de texto usados pelas primeiras ferramentas de pesquisa de diretórios dos sistemas operativos utilizam principalmente técnicas matemáticas de procura. Nos dias de hoje, o nome da ferramenta foi mantido, mas expressões regulares modernas como as baseadas em “Perl” muito pouco têm a ver com as técnicas usadas inicialmente (Goyvaerts, 2009).

Qualquer programador que use de forma correta expressões regulares consegue simplificar, em grande parte, o código de processamento de texto e permite também executar tarefas que de outra forma não seria possível. Construir uma função que retire todos os e-mails de um documento extenso poderia ser constituída por dezenas de linhas de código, pouco motivador de escrever e difícil de manter. Mas com expressões regulares, apenas necessitamos de uma linha de código para implementar esta tarefa de forma eficiente.

Não existe um “standard” oficial para os padrões das expressões regulares, ou seja, não existe um guia que diga que o caracter A é um “wild-card”, por exemplo. Como se pode imaginar, cada linguagem de programação e cada aplicação de processamento de texto tem a sua ideia de como fazer as coisas. Felizmente a maior parte dos designers de linguagens de programação pensaram da mesma forma: “Porquê criar algo radicalmente diferente quando podemos reutilizar ideias?”. Como resultado, hoje em dia todas as linguagens de programação usam sintaxe similar e compatível com o estilo do “Perl”, muito

à imagem do que aconteceu com o SQL em que a Microsoft e a Oracle construíram os seus sistemas com pequenas diferenças, apesar da base ser a mesma.

2.2.2. Stop Words

Stopwords ou palavras comuns são termos que apesar de muito úteis na arquitetura do discurso entre seres humanos, geram bastante ruido quando uma máquina tenta extrair significado de um determinado texto. São consideradas palavras não discriminativas. Isto é, são expressões que por elas mesmas trazem muito pouca informação sobre o conteúdo e por consequência podem fazer com que as palavras realmente importantes passem despercebidas. Outro problema associado às stopwords prende-se com a carga de processamento que adicionam quando se quer processar e analisar textos em linguagem natural. Normalmente na abordagem de um problema deste género, podem-se categorizar "stopwords" em dois grupos: palavras gerais e palavras dependentes do domínio. O primeiro grupo inclui as palavras "standard", que são bastante conhecidas e estão disponíveis em diversos repositórios, pelo menos no que diz respeito à língua inglesa. No segundo grupo encontram-se palavras que são geradas tendo em conta o nosso universo de trabalho ou a nossa categorização de texto. O tipo de "stopwords" específicas varia bastante de domínio para domínio, por exemplo, o termo "Aprendizagem" pode ser considerado uma "stopword" no contexto de educação, mas no contexto das ciências da computação poderia ser considerado uma palavra com valor na identificação de significado. Stopwords específicas dependentes do contexto podem ser descobertas e usadas nos mais diversos domínios, desde gestão de recursos humanos até à bioinformática, por exemplo (Kamel, 2008).

A necessidade de criação automática de listas desta natureza não foi evidente para os primeiros investigadores nestas áreas, sendo que a primeira iniciativa neste sentido foi criada por Van Rijsbergen em 1979 e continua a ser das mais usadas nos dias de hoje (Van Rijsbergen, 1979). A figura seguinte ilustra um exemplo de uma lista de stopwords (Apple Computer, Inc, 2015).

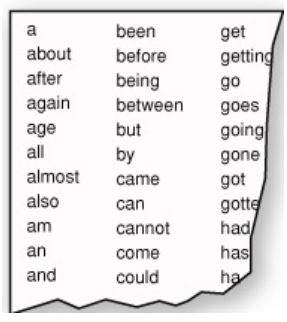

Figura 4 - Exemplo de lista de stopwords (extraído de (Apple Computer, Inc, 2015)).

Existem vários métodos que possibilitam a limpeza de "stopwords", nomeadamente através da consulta de uma lista com todas as entradas, por um filtro de alta frequência de expressões, um sistema de ranking de termos, ou pela combinação destes três métodos. Para os motores de busca mais comuns, entre as “stopwords” mais recorrente encontramos expressões como "the", "is", "at", "which", "on". Como podemos observar, as expressões dependem diretamente da língua natural em que estamos a trabalhar (neste exemplo a língua inglesa).

2.2.3. Stemming

Em Inglês, como em muitas outas línguas, palavras com o mesmo significado ocorrem em mais do que uma forma. A palavra “Book” e “Books” são a mesma expressão em duas formas diferentes. Na maior parte das vezes torna-se vantajoso eliminar este tipo de variações antes de avançar com o processamento do texto, ou seja, normalizar as palavras numa única “book”. Quando o objetivo da normalização é regularizar as variações gramáticas, como singular/plural ou passado/presente, estamos a falar de “inflectional stemming”. Para uma linguagem como o inglês com variadas irregularidades nas formas das palavras, torna-se por vezes complicado generalizar regras de “stem”.

Soluções mais sofisticadas criadas para resolver este tipo de problemas recorrem a dicionários, ou seja, todas as tentativas baseadas apenas na sintaxe, ignorando a informação linguística irão resultar invariavelmente em situações imperfeitas. Sendo que perante aplicações que usam funções de “stem” normalmente não necessitam de acertos de 100%, um algoritmo rápido que consiga uniformizar a maior parte do texto, normalmente é o suficiente para a maioria das aplicações garantirem resultados satisfatórios a este nível (Sholom, Indurkhya, & Zhang, 2010). Em 1980 Martin Porter (Porter, 1980) propôs um dos mais famosos algoritmos de “stemming” para a língua inglesa. Atualmente este algoritmo ainda é uma referência para resolver problemas desta natureza. A figura seguinte representa o pseudocódigo deste algoritmo.

```
Parâmetros: palavra(token) e dicionário. Regras:
If token length < 4
        return token
If token is number
        return token
If token is acronym
        return token
If token in dictionary
        return the stored stem
If token ends in s'
        strip the ' and return stripped token
If token ends in 's
        strip the 's and return stripped token
If token ends in "is", "us", or "ss"
        return token
If token ends in s
        strip s, check in dictionary and return stripped token if there
If token ends with es
        strip es, check in dictionary and return stripped token if there
If token ends in ies
        replace ies by y and return changed token
If token ends in s
        strip s and return stripped token
If token doesn't end with ed or ing
        return token
If token ends with ed
        strip ed, check in dictionary and return stripped token if there
If token ends in ied
        replace ied by y and return changed token
If token ends in eed
        remove d and return stripped token if in dictionary
If token ends with ing
        strip ing (if length > 5) and return stripped token if in dictionary
If token ends with ing
        length ≤ 5 return token
```

Figura 5 – Pseudocódigo de algoritmo simples de stemming para a língua inglesa baseado em Porter.

2.3. Reconhecimento de Padrões

Reconhecimento de padrões é uma disciplina científica cujo principal objetivo é classificar objetos tendo em conta um conjunto de categorias ou classes. Estes objetos são caracterizados por diversas características.

Este tipo de tarefas são de extrema dificuldade para uma máquina, onde mesmo dentro de um domínio de aplicação restrito é exigido um grande esforço para tomar uma decisão suficientemente inteligente.

Se o objetivo for restrito de tal forma que se obtenha um pequeno conjunto de hipóteses de categorização, pode-se então limitar o problema à escolha de categoria mais apropriada para a instância. Este tipo de abordagem é bastante útil para resolução dos mais variados problemas, nomeadamente, processamento de imagem, reconhecimento da fala, tarefas de diagnóstico ou identificação de padrões em texto. A obtenção da decisão pela máquina pode ser considerado uma das finalidades mais importantes no que diz respeito ao reconhecimento de padrões (Marques, 2005).

Tal como representado na figura abaixo, habitualmente um sistema de reconhecimento de padrões possui duas principais componentes, o identificador de caraterísticas e o classificador. A primeira

componente seleciona a informação para o processo de decisão, analisando e transformando dados num conjunto mais pequeno, as características. As características são utilizadas pelo classificador para escolher a classe mais apropriada para o objeto. Seguidamente ilustra-se um sistema de reconhecimento de padrões.

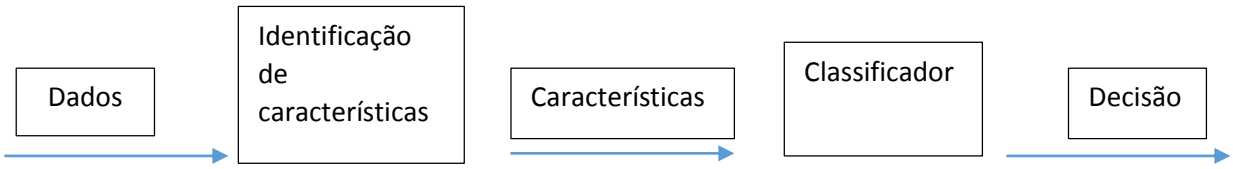

Figura 6 - Sistema de reconhecimento de padrões (extraído de (Marques, 2005)).

2.3.1 Características

A escolha de características tem uma importância vital para qualquer mecanismo de reconhecimento de padrões. Se as propriedades relevantes para a decisão forem conhecidas e em número reduzido, pode-se simplesmente incluir toda a informação no vetor de características. Contudo, quando o número é muito elevado, a escolha de um conjunto de características que contenha a informação relevante para a classificação é decisiva.

Entende-se por **extração de características** como a transformação das características existentes num espaço de dimensionalidade mais reduzido. Esta necessidade é evidente quando se possui uma quantidade excessiva de dados, muitas vezes redundantes e que atrasam o processo. A análise das componentes principais "Principal Component Analysis", ou o Singular Value Decomposition (SVD), são também muito usados em tarefas deste género.

Na **seleção de características** para atingir também o objetivo de redução de informação seleciona-se um subgrupo das características existentes e que consideramos mais importantes para o problema (Kumar D. K., 2013).

2.3.2 Classificador

O classificador é independente da problemática que pretendemos resolver, isto é, diferente da identificação de características, que possuem extrema dependência do ambiente em que estamos inseridos. Esta diferença permite que o sistema de decisão tenha uma componente que depende da natureza do problema e outra de processamento estatístico. Sendo assim, podem-se aplicar as técnicas de classificação em contextos variados, por exemplo, deteção de avarias, processamento de sinais, ou mesmo reconhecimento da fala.

2.3.3 Aprendizagem Supervisionada

De modo a aprender ou compreender fenómenos novos, as pessoas tentam apreender quais as características que os descrevem e compará-los com outros objetos ou fenómenos conhecidos baseando-se na similaridade entre eles. Na abordagem supervisionada admite-se que se conhece a classe que gerou cada padrão do conjunto de treinos. O classificador é treinado para replicar a decisão considerada correta para todos os padrões de treino existentes.

A aprendizagem supervisionada assemelha-se ao tipo de aprendizagem que os seres humanos são submetidos nos primeiros anos de vida. Este processo conta com a presença de supervisores que estimulam procedimentos considerados corretos e corrigem comportamentos menos corretos (Salvador, 2005). Sistemas de reconhecimento de padrões com uma abordagem de aprendizagem supervisionada usam uma estratégia comum, qualquer que seja o algoritmo aplicado. Tipicamente são executados os seguintes passos:

- Seleção do treino, afinação do conjunto de testes, que consiste em objetos que fazem parte de classes conhecidas e que os valores das suas características são conhecidos.
- Seleção e extração de características. Aquelas características que contêm informação para a concretização da classificação pretendida são mantidas, ao passo que outras características que apenas adicionem ruido são excluídas.
- Construção de um algoritmo tendo por base o conjunto de treino. Um modelo matemático derivado do número de características medidas nas amostras que constituem o conjunto de treino e as suas categorias conhecidas.
- Validação da aplicação usando amostras de teste independentes com o objetivo de avaliar a confiabilidade da aplicação.

2.3.4 Aprendizagem Não Supervisionada

A aprendizagem não supervisionada é um tópico de aprendizagem automática que trata de processar dados não classificados em conjuntos com alguma similaridade, de tal modo que essa organização revele a estrutura (estatística ou geométrica) subjacente ao conjunto de dados, sem qualquer tipo de informação à priori. No contexto de agrupamento de padrões, assume-se que os dados a tratar constituem a descrição de um conjunto de objetos ou padrões que pode ser processada por um computador. A aprendizagem não supervisionada é muito importante no processamento de conteúdos multimédia, por exemplo, uma vez que o particionamento de dados é muitas vezes um requisito na ausência de classes devidamente etiquetadas (Greene, Cunningham, & Mayer, 2008).

O objetivo do agrupamento de padrões consiste então na organização de um conjunto finito de dados não classificados em estruturas ”naturais “de dados. Para tal, um conjunto de padrões é organizado em grupos “clusters”, mais ou menos homogéneos com base numa medida de similaridade

previamente escolhida, de tal modo que a semelhança entre padrões dentro do mesmo grupo é maior que a semelhança entre padrões pertencentes a grupos diferentes. Na figura seguinte pode-se observar o processamento através de aprendizagem não supervisionada.

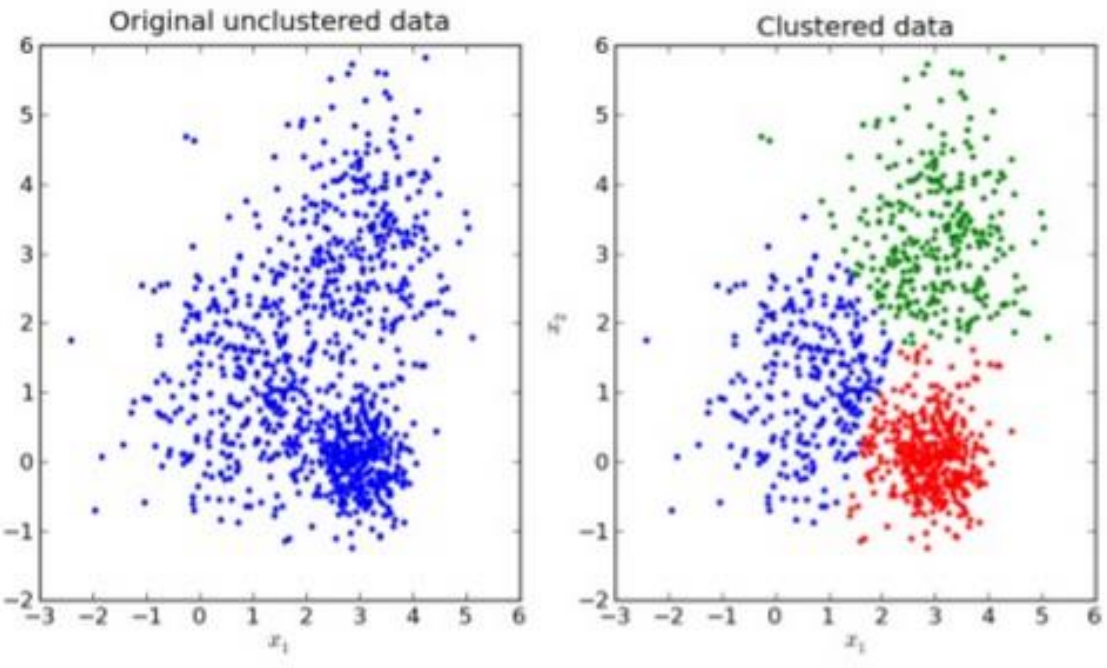

Figura 7- Exemplo de processamento usando aprendizagem não supervisionada.

2.4 Precisão/Recall[1]

Algumas técnicas para medir o desempenho de soluções foram estudadas para este tipo de problemas. Originalmente, o "Precisão/recall" foram duas medidas estatísticas largamente usadas em áreas de "information retrieval", mas que são bastante uteis na maioria dos problemas de processamento de linguagem natural (Blockeel, Ramon, Shavlik, & Tadepalli, 2007). A Precisão significa dentro do conjunto retornado, a percentagem de entidades que são de facto relevantes, enquanto o "Recall" é a percentagem de entidades relevantes que foram retornadas de todo o universo. Muito usada na literatura é também a notação de erros do tipo I ou falsos positivos e os erros do tipo 2 ou falsos negativos. Na figura abaixo, através de uma ilustração, explica-se graficamente o significado destes conceitos e medidas.

[1] Usa-se o termo técnico "Recall" ao longo de todo o documento em virtude de não existir termo equivalente em português.

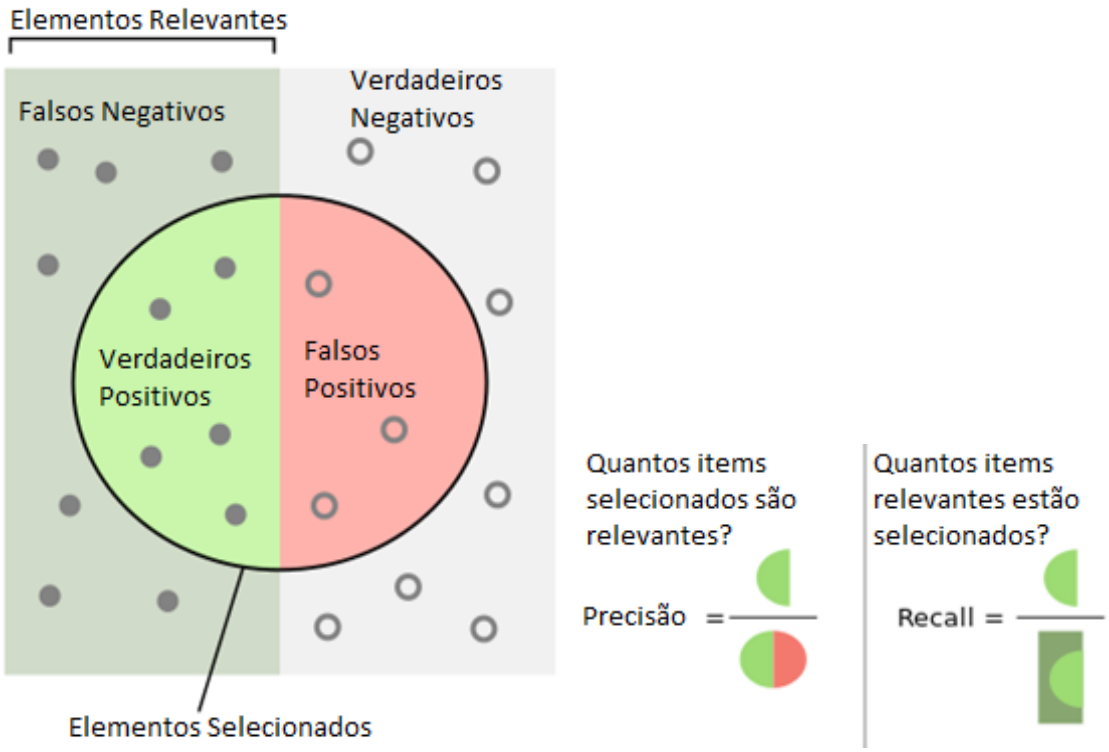

Figura 8 - Precisão/recall.

Suponhamos que efetuamos uma pesquisa que devolve 30 resultados de pesquisa sendo que desse conjunto, apenas 20 dos quais são relevantes. No entanto, o motor falhou o retorno de 40 resultados relevantes. Neste caso, a sua precisão resulta em 20/30 = 2/3 e o seu recall é de 20/60 = 1/3.

2.5 Similaridade vs Distância

A distância é a separação entre dois pontos, ou seja, é a medida pelo comprimento do segmento de reta que os conecta. No caso de dois exemplos físicos à superfície da Terra, então a distância é o mínimo comprimento entre as possíveis trajetórias partindo do primeiro ponto até chegar ao segundo. Uma distância tem que obedecer às seguintes propriedades:

O valor da distância é sempre um número positivo $d(x,y) \geq 0$.

É simétrica, ou seja, $d(x,y) = d(y,x)$.

Tem de se verificar a regra da triangularidade $d(x,z) \leq [d(x,y) + d(y,z)]$

Para dois pontos iguais tem de ser nula $d(x,y) = 0 \Rightarrow x = y$.

Qualquer solução que não verifique as propriedades acima referidas não será denominada de distância, visto que se trata de um abuso de linguagem.

2.6 Conclusões

Mesmo usando as técnicas acima referidas, é extremamente complicado com métodos de similaridade tradicionais quantificar o nível de relação semântica entre um “smartphone” e um “telemóvel”, ou entre as expressões “programar um método” e “codificar um procedimento”. Este tipo de associações é uma tarefa relativamente banal para um ser humano, mas é um obstáculo complexo para uma máquina. A grande diferença é que o ser humano não processa informação ao nível da palavra, mas sim ao nível do conjunto, sendo as palavras um mero instrumento para disparar conceitos bastante mais complexos relacionados com o seu conhecimento e experiência.

O conhecimento semântico é definitivamente um requisito fundamental na melhoria dos resultados deste tipo de aplicações. A organização e relacionamento dos dados é também de grande importância, visto que terá de refletir uma grande quantidade de domínios de aplicação. Acredita-se que a exploração de uma base de dados de conhecimento será portanto o caminho a seguir, tendo em conta os objetivos a que nos propomos.

Capítulo 3

Similaridade Semântica

Enquanto muitos dos avanços no processamento de linguagem natural foram executados por métodos maioritariamente estatísticos, acredita-se que futuros desenvolvimentos necessitarão da ajuda de semântica e de bases de conhecimento (Ontologias).

Entende-se por ontologia, no contexto das ciências da computação, como uma descrição de conceito e as suas respetivas relações. Consiste pois na forma de representação e especificação de classes, propriedades e relações entre entidades.

A semântica faz parte integrante do estudo dos símbolos e sinais usados por agentes, a semiótica. Charles Sanders Peirce considera que a semiótica pode ser descrita como a investigação dos sinais. Existem três principais áreas de estudo na semiótica, sintática, pragmática e a semântica. Posteriormente surgiram outras abordagens: Stamper, por exemplo, realça a importância de estudar outros aspetos técnicos dos sinais, como o nível físico e o mundo social. A análise semântica consiste na extração de significado de alto nível de estruturas sintáticas simples. Este processo pode ser automatizado por máquinas e normalmente consegue-se sempre tendo em base as relações entre os conceitos. A computação da similaridade semântica e relacionamento de palavras necessita de acesso a uma grande carga de Informação, tal como o conhecimento específico dos mais diversos domínios de aplicação. Duas das bases de conhecimento mais estudadas no mundo são a “WordNet”, base de dados léxica e “Wikipédia", enciclopédia digital.

A “Wordnet” é uma base de dados léxica que descreve estruturas e ligações entre palavras para mais de 100 000 conceitos da língua inglesa. Foi criada e é mantida por investigadores de linguística, o que pode ser interpretado como uma desvantagem, visto que não possui conhecimento de domínio específico dos especialistas dos diversos assuntos.

A “Wikipédia” está a tornar-se num corpus de grande valor para a ciência em diversas áreas, entre elas, Inteligência Artificial, Processamento de Linguagem Natural e Web semântica. É uma base de dados que guarda um grande número de conceitos dos mais diversos temas da humanidade, cobrindo desta forma diversos tópicos de domínio específico. Neste momento conta com cerca de 4.9 milhões de artigos, mas as suas mais-valias não se resumem ao seu tamanho. A Wikipedia inclui também diversas ligações entre artigos, capacidade de desambiguação de palavras, uma categorização ontológica e encontra-se em constante atualização. As características listadas contribuem para que de facto seja possível retirar conhecimento válido desta ontologia, com objetivos diversos, tais como medir relações semânticas, extração de relações e relações de multilinguismo.

3.1 Semiótica

Tradicionalmente a semiótica tem sido alvo de estudo em três dimensões separadas, embora fortemente interligadas, o que reflete as raízes da filosofia do tema. Estas dimensões foram designadas como sintáticas, semânticas e pragmáticas (Filipe, 2000).

Alguns autores como Stamper mencionaram a importância de estudar outros aspetos técnicos dos sinais, incluindo a necessidade da dimensão social dos sinais e do estudo dos aspetos sociais evoluídos, os quais incluem a criação e modificação de outras relações sociais.

Assim, Stamper expandiu a escada da semiótica de três para seis níveis, como observado na figura seguinte:

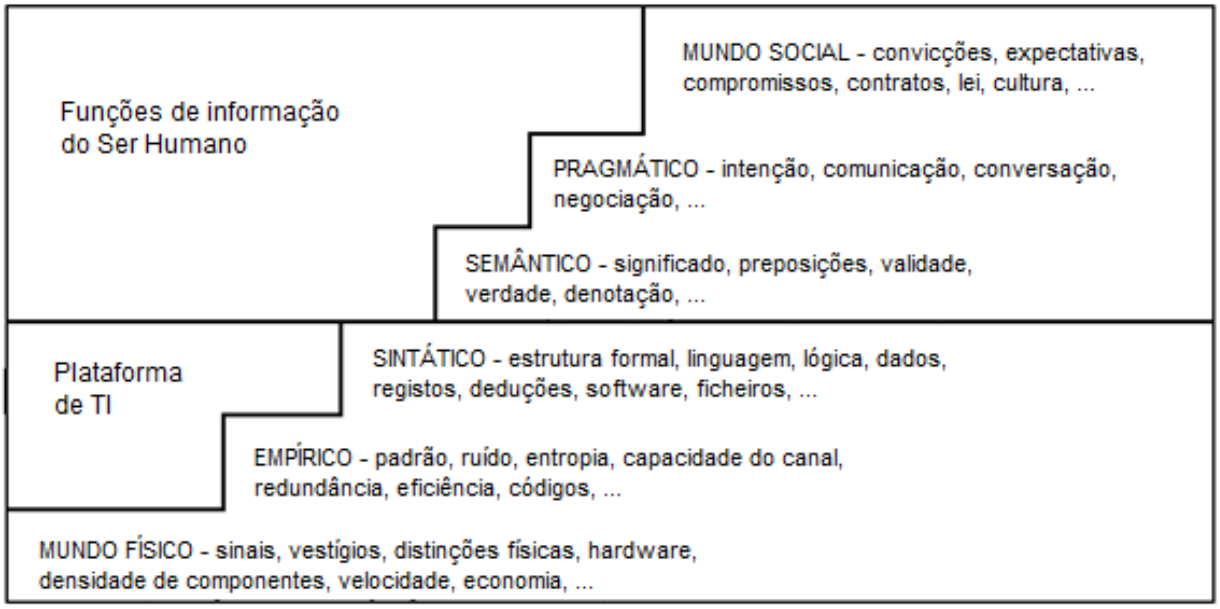

Figura 9 - Escada da semiótica.

3.1.1. Nível Físico

Diz respeito à economia da informação. Sinais são modelos para sinais físicos (variantes no tempo) e marcas (estáticas no tempo), as suas origens e destinos sobre os quais são transmitidos. A informação é uma coleção de tokens físicos. Uma comunicação bem-sucedida é atingida quando uma cadeia de tokens físicos, transmitidos ao longo de um canal é recebida no final pelo recetor, conservando as mesmas propriedades físicas.

3.1.2. Nível Empírico

O principal interesse do nível empírico é a análise das suas propriedades estatísticas. Uma comunicação é bem-sucedida a este nível quando o recetor pode reconstruir a mesma sequência de símbolos que foi enviada pelo emissor, independentemente de quaisquer problemas a nível físico. Envolve os estudos de dispositivos de comunicação, maximização do sinal e deteção/correção de erros.

3.1.3. Nível Sintático

Nele os sinais correspondem a tokens lógicos que representam alguns tokens físicos ou tokens que são muitas vezes designados por símbolos. A sintaxe diz respeito à análise e geração de estruturas simbólicas pela manipulação de símbolos usando regras de produção. Conceitos como complexidade, ambiguidade e estrutura são alvo de investigação neste nível. Um dos conceitos sintáticos mais úteis é o mecanismo generativo, no qual expressões simbólicas (fórmulas) são mapeadas noutras de acordo com um conjunto de regras (axiomas). Este processo pode também ser designado por "dedução". A comunicação neste nível presume a existência de dispositivos capazes de processar expressões simbólicas. A comunicação é bem-sucedida se os dispositivos são capazes de identificar internamente e reconstituir cada uma das expressões, independentemente dos símbolos usados.

3.1.4. Nível Semântico

Semântica diz respeito ao significado de sinais. Contudo, há posturas filosóficas diferentes na definição do seu significado. Se uma visão objetivista é adotada, então é adequado colocar símbolos num conjunto bem organizado, definido a nível de sintaxe, mapeado para o mundo real. Contudo, num ambiente social, onde o consenso não é garantido, é possível que agentes tenham diferentes visões do mundo. Neste caso torna-se necessário negociar e unificar as funções de correspondência individuais ou até negociar e completar uma ou mais das visões individuais do mundo através de relações e conceitos em falta.

O princípio construtivista do significado da semântica é essencial em muitos aspetos dos negócios organizacionais, uma vez que as interações entre pessoas requerem sempre unificação semântica dos conceitos trocados no discurso. O conhecimento da importância do ouvinte na comunicação, não apenas para decifrar a mensagem, mas também para a perceber e aceitar, reforça a relevância do princípio construtivista do significado semântico.

Neste nível, uma comunicação diz-se bem-sucedida se as interpretações forem equivalentes, tanto no orador como do ouvinte. Se um discurso pode ser interpretado de diferentes modos, tal significa que a interpretação escolhida pelo orador deve ser a mesma escolhida pelo ouvinte.

3.1.5. Nível Pragmático

A pragmática é o nível da semiótica que diz respeito à relação entre sinais e o principal comportamento dos agentes responsáveis, num contexto social. Intenções desempenham um papel importante neste nível. Até mesmo uma não-ação pode ser uma ação caso reflita a resposta do ouvinte à intensão do orador. A pragmática relaciona intensões com comportamento. Deste modo, uma comunicação considera-se bem-sucedida neste nível se o ouvinte entender as intensões do orador independentemente da correção sintática e precisão semântica. A comunicação pode ser bem-sucedida mesmo que o ouvinte não execute o que o orador pretendia, desde que o ouvinte interprete corretamente a intensão expressa pelo orador.

3.1.6. Nível Social

Tipicamente os atos de comunicação têm efeitos persuasivos, os quais correspondem à criação de atitudes à priori das ações realizadas pelo ouvinte como resultado da sua interpretação do ato de comunicação. Estas ações podem criar, manter ou alterar relações sociais, sendo portanto a consequência social do sinal de comunicação.

A comunicação no nível social requer que o orador e o ouvinte partilhem normas sociais. Mesmo que o ouvinte aparentemente não faça nada, a comunicação é bem-sucedida a este nível desde que como resultado de um ato comunicativo, qualquer compromisso social seja criado ou modificado. O conhecimento a nível social é essencialmente definido em termos de normas, i.e. regularidades de comportamento, perceção, julgamento e crença.

3.2 Ontologia

Existe consenso, em termos científicos, que o estudo da ontologia diz respeito aos tipos de coisas que fazem parte da nossa existência. Neste contexto, “tipo” significa “categoria”, tendo sido usado por Aristóteles para discutir sobre entidades. Os sistemas de categorias são, em grande parte das vezes, estruturados em níveis hierárquicos, na forma de uma árvore invertida.

3.2.1 Origem na Filosofia

Aristóteles foi o primeiro filósofo a usar a palavra grega “kategoria“ como termo técnico. De acordo com Aristóteles, um sistema de categorias deve ser capaz de criar uma lista das coisas que existem. Deve conter uma lista de tipos de alto nível, denominadas categorias. Os objetos mais comuns da experiencia humana são atribuídos a classes com um nível de generalidade superior. Para Aristóteles, cada entidade possui a sua essência real. Para uma entidade ser de um certo tipo, esta

deve compartilhar um conjunto de propriedades com outros membros daquele tipo. Com base nesta noção foi criado o método para distinguir essências que usa a distinção género-espécie. De acordo com a distinção género-espécie, a essência real de uma espécie é a combinação de seu género e da sua diferença, a qual é usada para distinguir uma espécie de outra do mesmo género. Por exemplo, a espécie humana pertence ao género Homo e sua diferença é a racionalidade. A ontologia de Aristóteles proporciona assim um método para ordenar categorias de acordo com suas características.

3.2.2 Aplicada à Computação

Além da sua utilização na filosofia, o termo ontologia tem sido usado noutras áreas da ciência, como por exemplo, nas ciências da computação. Em várias áreas das ciências da computação, tais como Inteligência Artificial, Engenharia de Software ou Bioinformática, o termo ontologia é usado para referir uma estrutura de termos organizados logicamente. Considerando as atividades e os agentes envolvidos na representação do conhecimento, é possível entender melhor o papel da ontologia. Pode-se também ver uma ontologia como uma teoria representativa dos principais factos e regras que governam parte da realidade, com fins computacionais.

3.2.3 Entidade

O termo entidade é uma expressão abstrata por natureza cujo significado aponta para a existência de algo, que pode ser material ou não. Uma entidade pode adquirir os mais diversos significados, dependendo em grande parte do contexto em que é empregue. Uma entidade é qualquer coisa que existe como uma unidade particular, sendo que pessoas e organizações são entidades equivalentes perante a lei (American Heritage, 2000). Em temos filosóficos, uma entidade é qualquer coisa que tem uma real e distintiva existência, isto é, uma coisa considerada independente de qualquer outra (Collins english dictionary, 2003). Um ser ou uma existência, considerado como único, independente e autossuficiente. Na Wordnet 3.0 entidade é definida como “Que é percebido ou conhecido ou inferido por ter a sua distinta existência, vivo ou não vivo”. De acordo com o projeto Thesaurus, os sinónimos mais comuns são: uma coisa, um ser, um corpo, um individuo, uma presença, uma existência, uma substancia, uma criatura ou mesmo um organismo. No que diz respeito a sistemas de informação, o conceito de entidade começou a ser empregue por volta dos anos 70, principalmente no que diz respeito a modelação de dados para sistemas de bases de dados. Como exemplo temos o modelo relacional sugerido por Edgar F. Codd em 1970.

Geralmente as entidades são caracterizadas por atributos, e estes atributos determinam a natureza das instâncias em questão. Todas as entidades têm a capacidade de se relacionarem com outras para que desta forma seja possível encontrar caminhos e determinar distâncias entre os objetos.

3.3 Wordnet

Wordnet é uma base de dados léxica para a língua inglesa, que agrupa as palavras em conjuntos conforme o seu relacionamento, disponibilizando também pequenas definições e exemplos de utilização. Foi criada com o intuito de ajudar aplicações de processamento de linguagem natural, mas existe também uma versão web disponível a qualquer pessoa e que pode ser consultada através de qualquer browser. A sua base de dados e bibliotecas de acesso estão também disponíveis para download gratuito. O projeto foi criado na Universidade de Princeton em 1985. Neste momento encontra-se na versão 3.1, que conta com cerca de 155 mil palavras organizadas nas categorias de nomes, verbos, adjetivos e advérbios. Uma das características chave desta base de dados são os conjuntos (synsets), que agrupam as palavras que são da mesma categoria e que têm significados semânticos similares (carro, Automóvel). Neste momento existem 117 mil conjuntos interligados entre si.

Na figura seguinte ilustra-se o exemplo do resultado da pesquisa do termo “automobile” na interface do Wordnet.

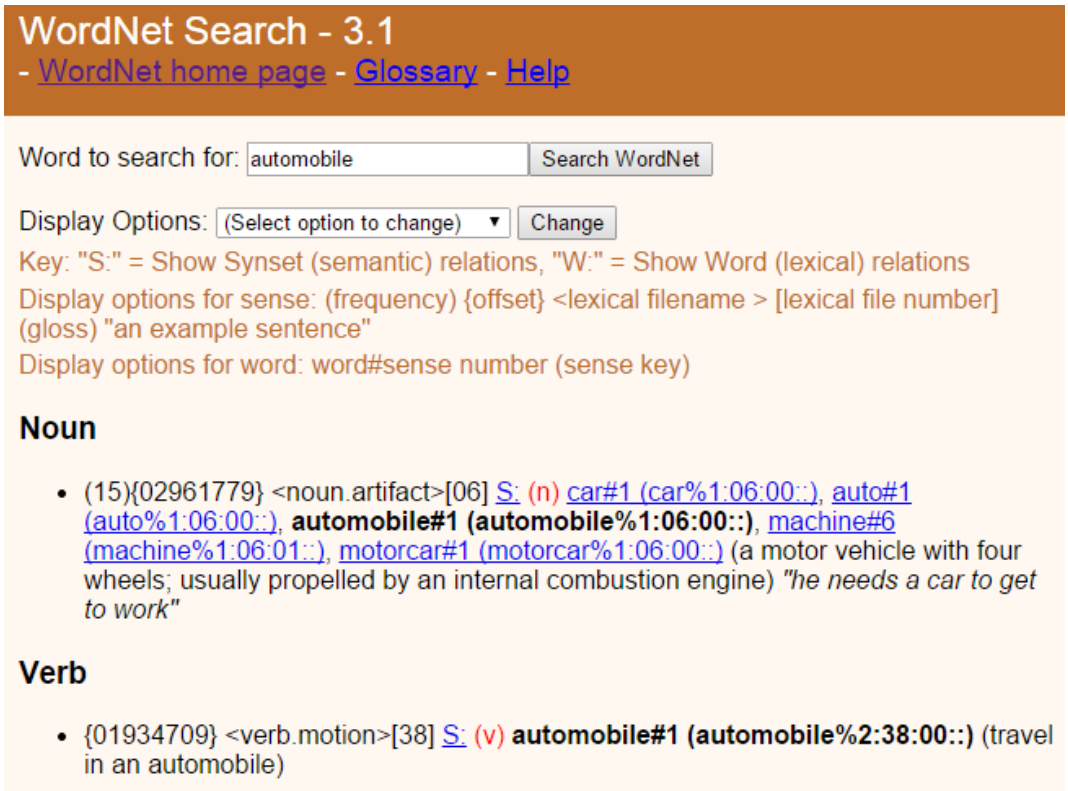

Figura 10 - Termo “automobile” na interface web da Wordenet (extraído de (Wordnet, 2015)).

Todos os conjuntos estão relacionados entre si, sendo que os tipos de relações léxicas mais comuns encontradas entre estes conjuntos são hiperónimos e hipónimos. Por um lado os hiperónimos são termos mais gerais que possuem instâncias mais específicas, os hipónimos. Por outro lado, um hipónimo é uma palavra ou frase onde o seu significado semântico tem um grande nível de especificidade. Hiperónimos e hipónimos são assimétricos relacionalmente, ou seja, faz sentido dizer que um computador é um tipo de máquina, mas uma máquina não é um tipo de computador.

Como podemos observar na figura seguinte, os hipónimos têm propriedade transitiva, ou seja, se se considerar que X é hipónimo de Y, e Y é hipónimo de Z, então X é hipónimo de Z. Também é normal observar-se palavras que são simultaneamente hiperónimos e hipónimos, como a palavra Purple, exemplificada na figura seguinte. Neste tipo de relação, o topo das estruturas de relacionamento é sempre composto por expressões mais abstratas, enquanto, no fundo da estrutura temos expressões bastante específicas.

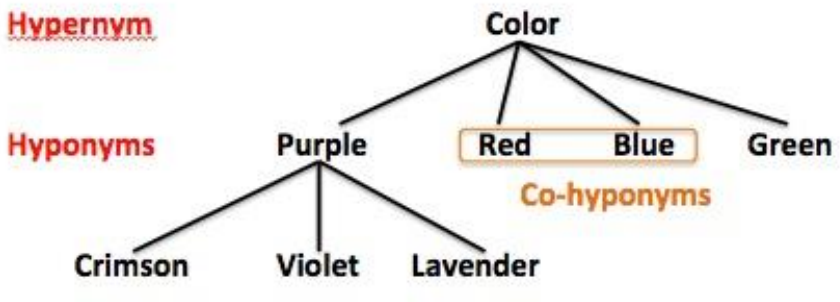

Figura 11 - Exemplo de uma relação léxica (extraído de (Wikipedia, 2015)).

A relação hiperónimo/hipónimo entre conjuntos de sinónimos pode ser interpretada como uma especialização entre categorias.

A Wordnet já foi usada para os mais diversos objetivos em termos de tecnologias de informação, tais como: desambiguação de palavras, classificação e sumarização e tradução de textos. Mas o uso mais comum é mesmo a determinação de similaridade entre palavras, sendo que vários algoritmos foram propostos ao logo dos anos, como por exemplo a distância entre conjunto de sinónimos. Na sua base de dados não está incluída informação sobre o uso das expressões, assim como falta de informação sobre o domínio específico da expressão. Mesmo na própria desambiguação de palavras em texto, o desempenho está ainda longe de ser comparável com a do ser-humano.

3.4 Wikipédia

É uma enciclopédia livre de acesso gratuito, sem fins lucrativos e que se encontra entre os websites mais visitados da internet. Todos os utilizadores da Wikipédia podem contribuir para melhorar artigos existentes, assim como, contribuir com a criação de novos artigos. Foi lançada a 15 de janeiro de 2001, inicialmente em inglês, mas rapidamente se tornou Multilingual através do desenvolvimento de versões similares em diferentes línguas que diferem sempre no que diz respeito ao conteúdo, existindo neste momento mais de 200 versões. A versão inglesa continua a ser a mais completa com 4.9 milhões de artigos, 18 mil milhões de visualizações de páginas e 500 milhões de visitas por mês (Wikipedia, 2015).

Ao contrário das enciclopédias tradicionais, que passam por longos processos de revisão e aperfeiçoamento até chegarem ao público, a Wikipédia disponibiliza rapidamente os artigos escritos ou editados por quaisquer dos seus utilizadores. Sendo que a ideia é estimular a comunidade para ir melhorando os artigos ao longo do tempo, existe a desvantagem de se poder assistir a erros, incoerências e mesmo vandalismo. Como solução para este problema, a Wikipédia decidiu criar políticas de proteção das suas páginas e só alguns editores podem efetuar/aprovar alterações em determinadas páginas. Os princípios da Wikipédia estão assentes em cinco pilares: Acima de tudo a Wikipédia é uma enciclopédia e combina características gerais das principais enciclopédias mundiais. Todos os seus artigos são escritos do ponto de vista neutro, é sempre privilegiada a abordagem da explicação dos vários pontos de vista sobre o assunto em vez de verdades únicas. A Wikipédia é gratuita e todas pessoas podem consultar, editar e contribuir. Todos os utilizadores devem respeitar as opiniões e tratar com civismo todos os outros utilizadores. Não existem regras rígidas, as linhas orientadoras estão em constante evolução.

Cada artigo da Wikipédia tem associado uma página de conversa. Esta é a forma primária que os editores têm para discutir, coordenar e debater os temas. Um relatório de 2014 indica-nos que a Wikipédia conta com cerca de 80 000 editores, sendo que só metade deles se podem considerar como editores ativos.

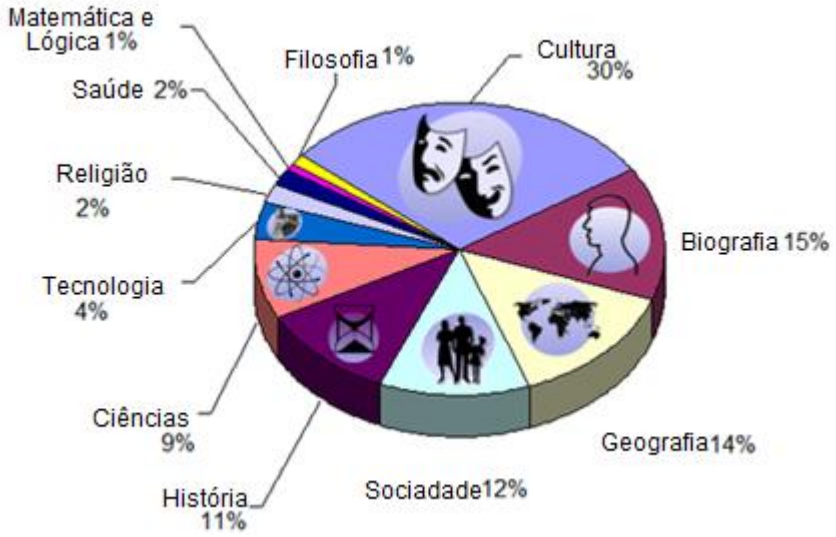

Figura 12 - Áreas da Wikipédia.

Artigos de enciclopédias tradicionais como a Encyclopedia Britannica são cuidadosamente escritos por pessoas altamente especializadas nos respetivos temas, levando estas obras a terem um elevado nível de credibilidade. Por vezes a Wikipédia é acusada de possuir informação incompleta ou pouco estruturada. Foi efetuado em 2005 um estudo de comparação de 42 entidades cientificas da Wikipédia e da Encyclopædia Britannica pelo journal "Nature" e descobriu-se que a diferença de qualidade é mínima, concluindo-se que em média por cada artigo na Encyclopædia Britannica foram identificados 3 pontos fracos na Wikipédia 4 (Giles, 2005).

3.5 Conclusão

Neste capítulo começou-se por apresentar o que se entende por ontologia e análise semântica, referindo duas das bases de conhecimento mais estudadas da atualidade.

No que diz respeito à semiótica, é mencionado o trabalho de Stamper em relação à existência de seis níveis na escada da semiótica, nomeadamente nível físico (economia da informação), nível empírico (análise de propriedades estatísticas), nível sintático (análise e geração de estruturas simbólicas), nível semântico (significado de sinais), nível pragmático (relação entre sinais e comportamento dos agentes responsáveis) e nível social (consequência social do sinal de comunicação).

Relativamente ao conceito de ontologia, embora o mesmo tenha origem na filosofia, por meio de Aristóteles, pode este conceito ser aplicado à computação, sendo usado na referenciação de uma estrutura de termos organizados logicamente. Associado a estes conceitos apresenta-se a noção de entidade.

Quanto às bases de conhecimento mencionadas, Wordnet e Wikipedia, foi estudada e utilizada a segunda devido às suas características. Ao passo que a Wordnet é uma base de dados léxica, criada e desenvolvida por investigadores da linguística não possuindo assim conhecimento de domínio específico dos especialistas, a Wikipédia surge como uma base de dados com um vasto número de conceitos de variados temas, cobrindo assim diversos tópicos de domínio específico.

Capítulo 4

Similaridade Semântica entre Entidades na Wikipédia

A medida de similaridade proposta é baseada no princípio de que todos os objetos da ontologia utilizada estão conectados entre si, para que desta forma seja possível o cálculo de caminhos necessários entre os mais diversos conceitos.

A base de conhecimento para resolução do problema proposto será a ontologia de categorias da Wikipédia, que como se verificou no capítulo anterior, trata-se da maior e mais completa ontologia de conhecimento disponível. Todos os artigos encontram-se catalogados e estas categorias estão inseridas na ontologia.

No presente contexto uma entidade pode assumir os mais diversos significados. Porém, estas entidades têm de ser caracterizadas por propriedades ou características. O exemplo mais comum de entidade que podemos dar é o de uma pessoa, em que neste caso, como características, poderíamos selecionar os seus interesses.

Numa ontologia, o "Least Common Subsumer" de dois conceitos A e B é descoberto quando se pretende efetuar o cálculo do caminho entre ambos, isto é, o conceito mais específico e antecessor de A e B. Trata-se do primeiro objeto pai comum entre os outros dois que se busca o caminho. É uma ferramenta importante para o projeto, porque, por exemplo a sua profundidade pode ser uma variável interessante. Seguidamente pode-se observar a representação do conceito de "Least Common Subsumer".

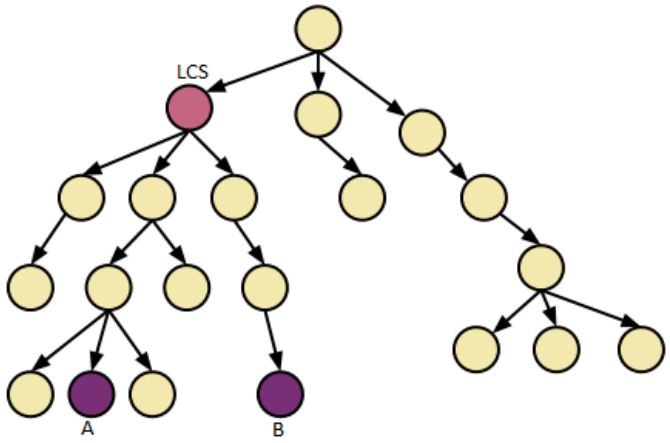

Figura 13 - Least Common Subsumer (extraído de (Stoimen, 2015)).

Para se conseguir uma relação apurada entre entidades é necessário realizar o cálculo exaustivo de todos os caminhos de todas as propriedades de uma entidade E1 com todas as propriedades de uma entidade E2, recorrendo à ontologia. Desta forma obtém-se a ponderação de todas as variáveis envolvidas, com o objetivo de retornar o valor de similaridade de E1 e E2. A identificação de nós vizinhos partilhados através dos vários caminhos encontrados entre as propriedades é também considerada uma variável a ter em conta no cálculo final da similaridade entre duas entidades.

Por vezes encontram-se dificuldades na tarefa de correspondência das propriedades de uma entidade com os conceitos da Wikipédia, ou seja, somos confrontados com a necessidade de desambiguar uma expressão. Várias técnicas de desambiguação automática foram estudadas com o objetivo de solucionar este problema. Tendo em conta diversos testes e análise de resultados, verificou-se que existem diversas partes da ontologia de categorias da Wikipédia que não se encontram desenvolvidas uniformemente. Parte do desafio de conseguir uma similaridade consistente passa por atenuar este tipo de fenómenos. Desta forma pretende-se a implementação de algoritmos capazes de se adaptar à densidade da região em que o caminho entre dois nós está a ser pesquisado.

4.1 Implementação da Medida

A implementação da medida proposta tem por base o princípio de que todos os conceitos da ontologia estão conectados entre si, tal como se pode observar na figura abaixo para o par de tópicos ”Feature Learning” e ”Boosting”. Nesta figura pode-se verificar que o “Least Common Subsumer” (LCS) destes dois tópicos é o termo “Machine Learning”.

A medida de relação semântica proposta é executada tendo em conta a seguinte sequência de passos:

Cálculo de semelhança entre conceitos

Consiste no “comprimento” do caminho entre conceitos. Assumindo-se c1 e c2 dois conceitos representados na rede de categorias da Wikipédia, a primeira preocupação é encontrar o caminho mais curto entre c1 e c2; seguidamente teremos de achar o número de conceitos total entre c1 e o LCS e por fim o número total de conceitos entre c2 e o LCS. Abaixo exemplifica-se um caminho de categorias entre dois conceitos.

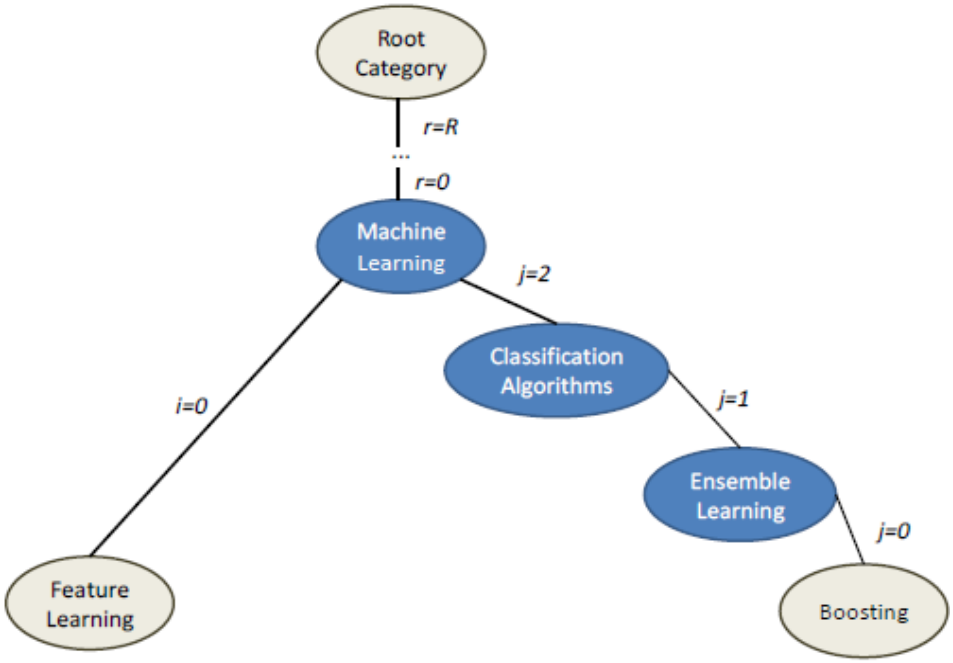

Figura 14 - Caminho de categorias entre os conceitos "Feature Learning" e "Boosting" (extraído de (Medina, Fred, Rodrigues, & Filipe, 2012)).

O valor da relação entre os dois conceitos é dada por:

$$r\,(c1,c2) = \frac{\sum_{i=0}^{I} w_i^1 + \sum_{i=0}^{j} w_i^2}{\sum_{i=0}^{R} w_i^1 + \sum_{i=0}^{R} w_i^2} \quad (1)$$

Onde w_i^1 é o peso da aresta com o índice i no caminho entre c1 e o tópico LCS, w_i^2 é o peso da aresta com o índice i no caminho entre c2 e o tópico LCS, I é o número de arestas entre c1 e o LCS e o J é o número de arestas que conecta c2 e o LCS. Com R a representar o número de arestas entre o nó raiz e o nó que se está a processar, com a restrição que este caminho tem de incluir o LCS calculado anteriormente.

Similaridade entre entidades

Dadas duas entidades E1 e E2 representadas pelos conjuntos de conceitos C1 = $\{c_1^1, \dots, c_n^1\}$ e C2 = $\{c_1^2, \dots, c_m^2\}$, respetivamente, a nossa definição para a medida de similaridade entre estas duas entidades r(E1,E2) é dada por:

$$r(E_1,E_2) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} r\,(ci,cj)}{n * m}$$

Peso na medida de similaridade dos nós vizinhos mais próximos partilhados (SNN)

Sejam c_1 e c_2 dois conceitos em que $c_1 \in E_1$ e $c_2 \in E_2$. Começa-se por calcular todos os caminhos mais curtos entre c_1 e todos os concentos pertencentes a E_2 , sendo C^1 o conjunto de categorias contido nestes caminhos. Seguidamente segue-se o mesmo procedimento para todos os pares possíveis de c_2 com os conceitos pertencentes a E_1 sendo C^2 o conjunto de categorias contido nestes caminhos.

Referimo-nos a C^1 e C^2 como os conjuntos de nós vizinhos mais próximos de c_1 e c_2 respetivamente. Os nós vizinhos mais próximos partilhados de c_1 e c_2 correspondem à interseção $C^1 \cap C^2$. Seguidamente define-se o peso deste componente, que é proporcional ao número de nós partilhados.

$$SNN(c_1 \text{ e } c_2) = \frac{|C^1 \cap C^2|}{|C^1|+|C^2|}$$

Esta equação pode ser generalizada para medir o peso dos nós partilhados de categorias entre duas entidades da seguinte forma:

$$SNN(E_1,E_2) = \sum_{i \in E1}^{n} \quad \sum_{j \in E2}^{m} \quad \frac{|C^i \cap C^j|}{|C^i|+|C^j|}$$

A localização de um nó na rede de categorias pode influenciar a sua relevância na computação total do relacionamento entre entidades. Um nó localizado a uma profundidade elevada na hierarquia é mais específico, e por isso mais relevante para a caracterização semântica de proximidade entre conceitos. Se um determinado caminho contém algumas categorias localizadas em níveis profundos da hierarquia, então os conceitos contidos neste caminho contribuem mais para a qualidade dos resultados da medida de similaridade do que caminhos localizados em níveis superficiais.

Ao adicionarmos esta componente à nossa equação, ficamos com:

$$Wsnn\ (E_1,E_2) = \frac{\sum l \in C^i \cap C^j\ w_n(l)}{\sum l \in C^i\ w_n(l) + \sum l \in C^j\ w_n(lcs)}$$

Onde $w_n(l)$ é a profundidade do nó (número de arestas entre o nó corrente e a raiz da ontologia).

A medida que propomos resulta da combinação das componentes acima apresentadas:

$$M\,(E1,E2) = \frac{\alpha_1\ Snn(E_1,E_2)+\alpha_2\ Wsnn(E_1,E_2)}{\alpha_1\ +\alpha_2},$$

Onde α1 e α1 são parâmetros pré-definidos.

Peso dos nós "Least Common Subsumer" partilhados (S*lcs*)

Esta técnica reflete o pressuposto de que se duas entidades partilham muitos dos *lcs* e estes encontram-se a uma profundidade considerável, então estas entidades devem ter uma relação forte. Por outro lado, se têm poucos *lcs* partilhados e estes estão localizados superficialmente, então o nível de semelhança entre as entidades deve ser fraco.

A medida seguinte é assignada aos nós *lcs* de duas entidades:

$$S_{lcs}\,(E_1,E_2) = \frac{\sum lcs\in\{C^1\cap C^2\}\ w_n(lcs)}{\sum lcs\in C^1\ w_n(lcs) + \sum l\in C^2\ w_n(lcs)}$$

Onde $w_n(l)$ corresponde à sua profundidade na hierarquia.

Desta forma, a nossa equação fica com o seguinte aspeto:

$$M\,(E1,E2) = \frac{\alpha_1\ S(E_1,E_2)+\alpha_2\ Slcs(E_1,E_2)+\ \alpha_3\ Snn(E_1,E_2)}{\alpha_1\ +\alpha_2\ +\ \alpha_3}$$

onde $\alpha_1\ +\alpha_2\ +\ \alpha_3$ são parâmetros pré-definidos.

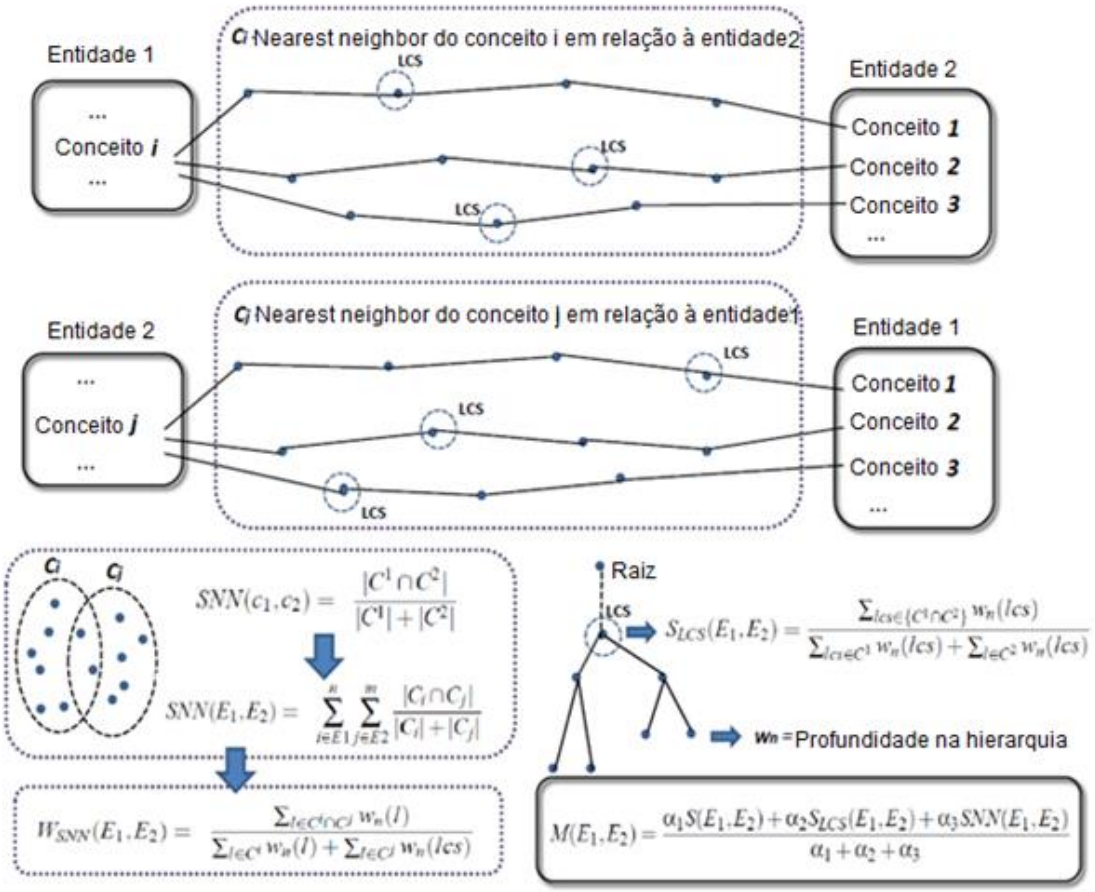

Figura 15 - Ilustração dos vários componentes da medida de similaridade (extraído de (Medina, Fred, Rodrigues, & Filipe, 2012)).

4.2 Computação dos Caminhos mais Curtos

Várias versões da base de dados da Wikipédia podem ser acedidas em http://dumps.wikimedia.org/backupindex.html. Para os resultados apresentados neste trabalho, resolveu-se utilizar a mais recente versão inglesa. O armazenamento é feito recorrendo a uma estrutura baseada em MySQL fornecida pela Java Wikipédia Library API (disponível em http://www.ukp.tudarmstadt.de/software/jwpl/), descrita em detalhe por (Zesch, Müller, & Gurevych, 2008). Esta API ajuda-nos igualmente a detetar se se tratam de páginas de desambiguação ou não. Foi criada uma nova camada de código que se encontra num nível de abstração superior a esta API e que nos permite calcular caminhos mais curtos entre nós da ontologia.

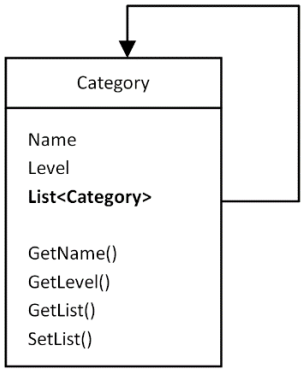

Figura 16 - Objeto do tipo Categoria.

Cada objeto da Wikipédia do tipo categoria possui o nível da procura corrente e a lista dos vizinhos mais próximos quando se está a desempenhar a tarefa de procura de caminho. Um exemplo do objeto categoria encontra-se na **Error! Reference source not found.**. Como podemos observar esta técnica é em tudo bastante similar a uma procura em árvore dos caminhos mais curtos, onde cada instância da categoria será uma folha da árvore.

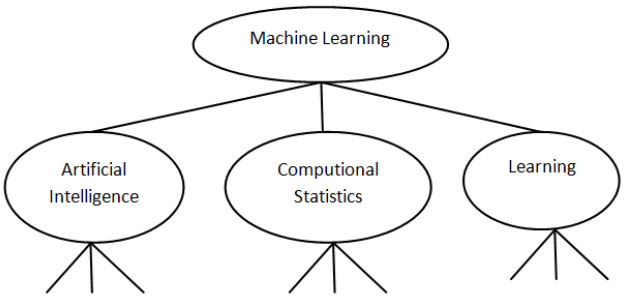

Figura 17 - Exemplo do Conceito Machine Learning.

Depois da instanciação do conceito de “Machine Learning”, todas as categorias descendentes (“Artificial intelligence”, “Learning”, e “Computional Statistics”) vão ter o atributo “level” com o valor dois. A instanciação de cada um destes conceitos de nível dois irá originar conceitos descendentes com o atributo “level” a três e assim sucessivamente. O seguinte pseudocódigo ilustra esta técnica. O método GetAboveLevelCategories() procura as categorias descendentes da categoria que está a ser iterada. Por cada caminho entre conceito são contruídas duas árvores de categorias, uma por cada conceito; o nível de profundidade irá crescer até que exista uma categoria comum, o LCS.

```
1 Category nextLevel(Category c)
2       Begin
3       ForEach Category in c.List
4       Begin
5               If ( IteratedCategory .List = null)
                -    >leaf node
6               Begin
7                       WikiList=wikipédia.
                        GetAboveLevelCategories();
8                       c.List=newList;
9               End
10              Else
11              Begin
12                      NextLevel( IteratedCategory );
                                ->recursive method
13              End
14      End
15 End
```

Pseudo-código de análise do próximo nível de nós da ontologia.

```
1 Void Main ()
2 Begin
3       List c1 = wikipédia.
        GetAboveLevelCategories (" concept1 ");
4       List c2 = wikipédia.
        GetAboveLevelCategories (" concept2");
5
6       While(ExistMatch(c1 ,c2) )
7       Begin
8               nextLevel(c1);
9               nextLevel(c2);
10      End
11 End
```

Pseudocódigo que permite a geração do caminho até à existência de LCS

Neste segundo exemplo de pseudocódigo, o método “ExistMatch” verifica todos os nós das duas árvores que estão a ser geradas: se é encontrada uma correspondência, então esse mesmo nó é considerado a “Least Common Subsummer” e é retornado. O caminho mais curto é seguidamente

encontrado através da análise dos dois nós de origem e o LCS. Com o objetivo de reduzir o custo de processamento todos os caminhos são guardados em cache.

4.3 Densidade

Depois de se observarem diferenças substanciais na densidade de relações na rede de categorias ao longo de todo a ontologia, concluiu-se que estas irregularidades poderiam levar a imperfeições nos resultados da medida de semelhança entre entidades. Assim sendo, decidiu-se então adicionar outro componente à fórmula de cálculo descrita no subcapítulo 4.1.

A ideia geral por trás desta componente pode ser introduzida através do conceito matemático genérico. Na teoria de grafos, um grafo denso é um grafo em que o número de arestas se encontra próximo do número máximo possível entre os vértices contidos.

Num grafo completo a densidade máxima é de um para um e a densidade mínima é próxima de zero para um grafo muito esparso (Coleman & More, 1983). No nosso caso, não se pretende proceder ao cálculo preciso da densidade de toda a rede de categorias da Wikipédia. Em vez disso, analisam-se todos os nós contidos no caminho mais curto, descobrindo assim todos os seus graus.

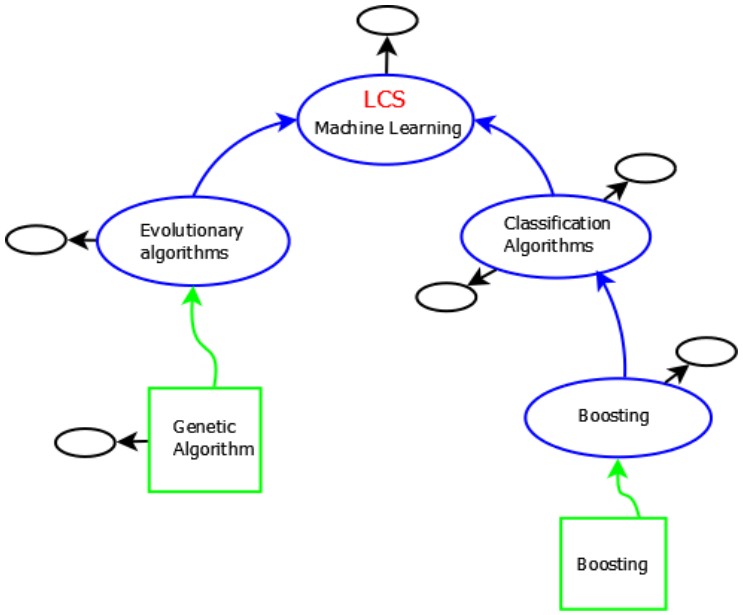

Figura 18 – Caminho pouco denso.

Na figura acima, os círculos mais largos representam as categorias da Wikipédia no caminho calculado, sendo que os pequenos círculos representam os vizinhos e os quadrados indicam as páginas da Wikipédia. Como podemos observar na figura seguinte, esta representa uma situação de alta densidade, uma vez que o grau dos vértices é bastante mais elevado.

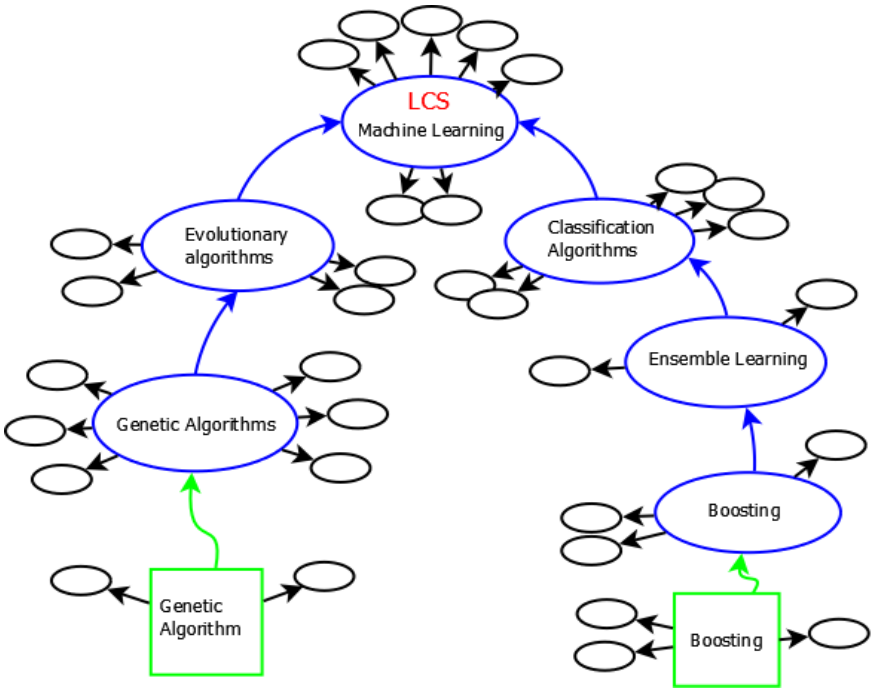

Figura 19 - Caminho muito denso.

Quando o cálculo do caminho ocorre, os graus de todas as categorias são guardados. O objetivo é guardar o máximo de informação possível sobre o número de vizinhos do caminho calculado. Como abordado anteriormente, as propriedades das entidades são mapeadas para a página correta que representa o conceito na Wikipédia. Na figura acima temos a ilustração da relação semântica entre a propriedade “generic algorithm” e a propriedade “boosting”. Estes dois conceitos estão contidos em dois conjuntos de propriedades, cada um deles representando a sua entidade.

Por cada caminho explorado entre conceitos das duas entidades, guarda-se informação sobre os vizinhos de todas as categorias dos caminhos. Isto significa que a informação da densidade que se encontrou nos caminhos entre conceitos das duas entidades é mantida entre cálculos de diferentes caminhos, e baseado nesta informação consegue-se entender melhor a densidade da parte da ontologia que estamos a usar no momento.

Como nas componentes anteriores da medida, o nó LCS tem bastante significado neste contexto (Medina, Fred, Rodrigues, & Filipe, 2012). Assim torna-se proveitoso ter em conta a densidade do caminho do LCS do nosso caminho e do nó raiz da árvore. A estratégia é em tudo semelhante ao cálculo de densidade entre propriedades regulares, só que os conceitos dão lugar ao LCS e à raiz da árvore de categorias.

4.4 Desambiguação

Durante os cálculos da medida de semelhança entre entidades, observou-se que várias propriedades acabavam por ser mapeadas em páginas ambíguas da Wikipédia. O processo de desambiguação pretende resolver os conflitos que nos surgem aquando um único conceito possui mais do que um significado na Wikipédia. Por exemplo, o conceito “Matrix” pode referir-se a um tópico matemático, um álbum de música, ou um filme de cinema, entre outros. Para evitar este tipo de ocorrências, determina-se o contexto dado pelas entidades não ambíguas e a partir daqui tenta-se encontrar a sugestão de desambiguação mais acertada.

Como resultado desta técnica, consegue-se incrementar o número de propriedades das entidades mapeadas corretamente. Consequentemente o número de caminhos calculados aumenta, aumentando assim também a eficácia da medida de similaridade.

4.4.1 Correspondência de Conceitos

Tendo em conta a técnica descrita no subcapítulo 4.1. , sabe-se que o primeiro passo é encontrar o artigo da Wikipédia que representa o conceito para o qual se pretende encontrar o caminho. Existem duas principais formas de encontrar o artigo correspondente a um conceito:

1. Encontrar a correspondência sintaticamente, comparando o texto do conceito com o título da página da Wikipédia.
2. Decompor o conceito, separando as várias palavras que o constituem, com o objetivo de encontrar correspondência sintática de alguma delas.

Para que estas duas soluções produzam resultados, foram desenvolvidas as seguintes ferramentas:

- Pesquisa direta – encontrar um artigo da Wikipédia através da comparação sintática, singular e plural da expressão, com o título do artigo. Este método é também reutilizado pelos próximos métodos.
- Uso de REGEX – criação de expressões regulares que retornam resultados dos casos mais comuns de conceitos com várias expressões: a palavra “and” ou símbolos de pontuação, tais como a vírgula ou o ponto e vírgula. Este tipo de elementos são bastante comuns e úteis para apanhar este tipo de expressões com sucesso.
- Decomposição – Este método decompõe o conceito por fases, sendo que em cada uma delas uma palavra é removida do conceito, até um mínimo de duas palavras.

4.4.2 Técnica de Desambiguação

Ao realizar a correspondência de todos os conceitos da nossa entidade, descobrimos quais tiveram ligação direta com a página correta e quais foram encaminhados para páginas de desambiguação. Neste momento o conjunto de categorias das páginas descobertas diretamente ganha uma grande relevância para a nossa solução. Chamemos a este conjunto o nosso contexto geral. Como pré-condição à desambiguação admite-se que existe pelo menos um conceito da nossa entidade mapeado de forma direta, isto é, o nosso contexto geral tem que ser constituído pelo menos, pelas categorias de um artigo.

Um segundo tipo de contextos é criado, sendo os contextos dos artigos sugeridos pela página de desambiguação. Para construir o contexto de um artigo são analisadas as suas categorias, as categorias dos "inlinks" e as categorias dos "outlinks". Entenda-se como "inlinks" as páginas da Wikipédia que apontam para a nossa página de desambiguação sugerida e "outlinks" as páginas para as quais apontamos. Podemos observar esta notação na figura abaixo.

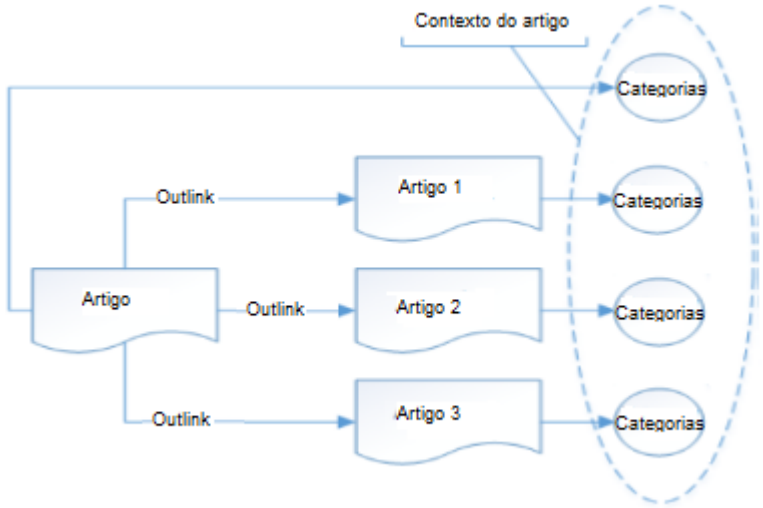

Figura 20 - Contexto do artigo.

Dentro dos referidos contextos existem categorias com mais peso do que outras, isto devido ao número de repetições das categorias nas páginas selecionadas para fazerem parte do contexto em questão.

Para se descobrir a página correta entre as sugestões de desambiguação, é necessário entender qual o contexto de artigo que se assemelha mais ao nosso contexto geral, com a ajuda do "cosine similarity", que é bastante usado neste tipo de problemas (Tan, Steinbach, & Kumar, 2005).

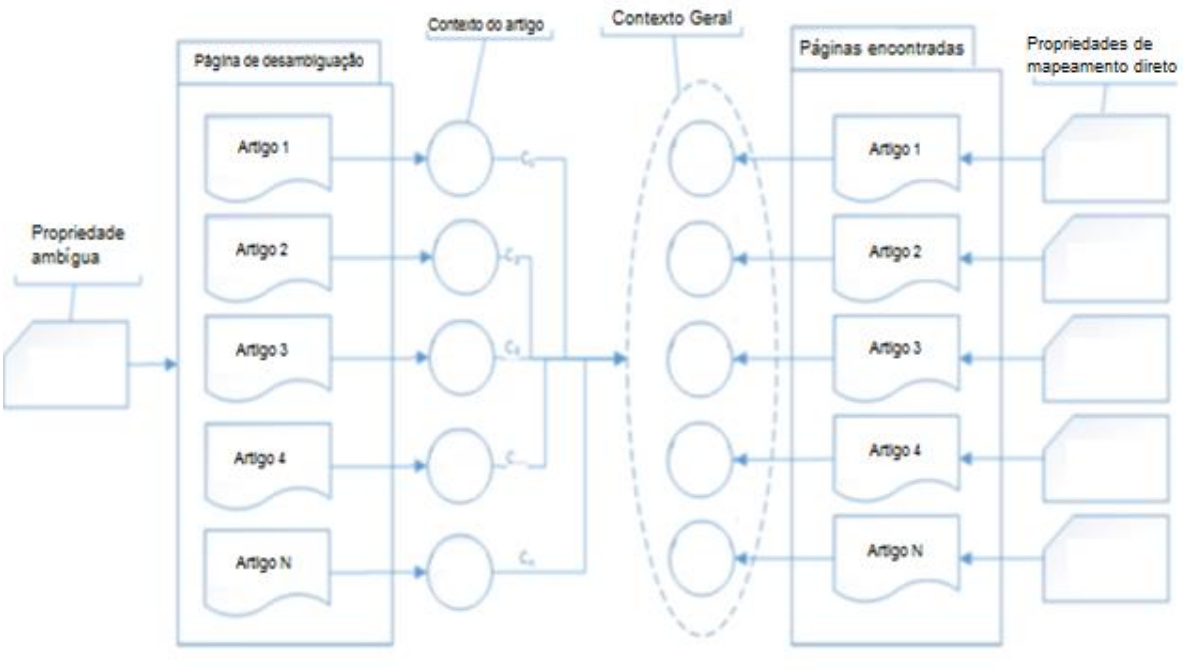

Figura 19 - Desambiguação baseada no contexto.

Finalmente, o artigo com maior similaridade será o mais adequado para ser mapeado com o nosso conceito. Este procedimento é descrito na figura acima.

4.5 Conclusão

Neste capítulo descreve-se a medida de similaridade proposta no âmbito desta tese de mestrado, medida essa baseada no princípio de que todos os objetos da ontologia usada estão conectadas entre si, de modo a possibilitar o cálculo de caminhos necessários entre os mais diversos conceitos.

A base de conhecimento utilizada para a resolução do problema mencionado em capítulos anteriores é a ontologia de categorias da wikipedia.

A noção de lcs “Least Common Subsumer” revelou-se bastante importante e útil no desenvolvimento deste projeto, foi amplamente explorada nas diversas componentes que contituem a medida de similaridade.

Para fazer frente às dificuldades na correspondência de propriedades de uma entidade com conceitos da Wikipedia foram aplicadas novas metodologias de desambiguação de conceitos, tendo por base a identificação de contexto.

A implementação da medida proposta implicou o estudo dos seguintes assuntos:

- Desambiguação de conceitos

- Cálculo de semelhança entre conceitos
- Nós vizinhos mais próximos partilhados
- Densidade da ontologia nos caminhos calculados
- Similaridade entre entidades
- Peso dos nós “Least Common Subsumer”

Capítulo 5

Sistema

Neste capítulo dar-se-á a conhecer a informação pertinente relativa à implementação da ferramente descrita nos capítulos anteriores.

Começar-se-á por apresentar o diagrama de classes, sendo feita uma pequena expplicação das classes que o constituem.

Será também apresentado o modelo entidade-relação, onde são apresentadas as tabelas existentes na base de dados.

5.1 Diagrama de classes

A programação orientada a objetos possibilita-nos garantir a modularidade do código produzido. Isto viabiliza-nos executar uma boa manutenção do projeto longo da sua evolução e afinação, assim como o aproveitamento de bibliotecas de funcionalidades já desenvolvidas por terceiros, acelerando desta forma o desenvolvimento. Seguidamente apresenta-se o diagrama de classes do projeto.

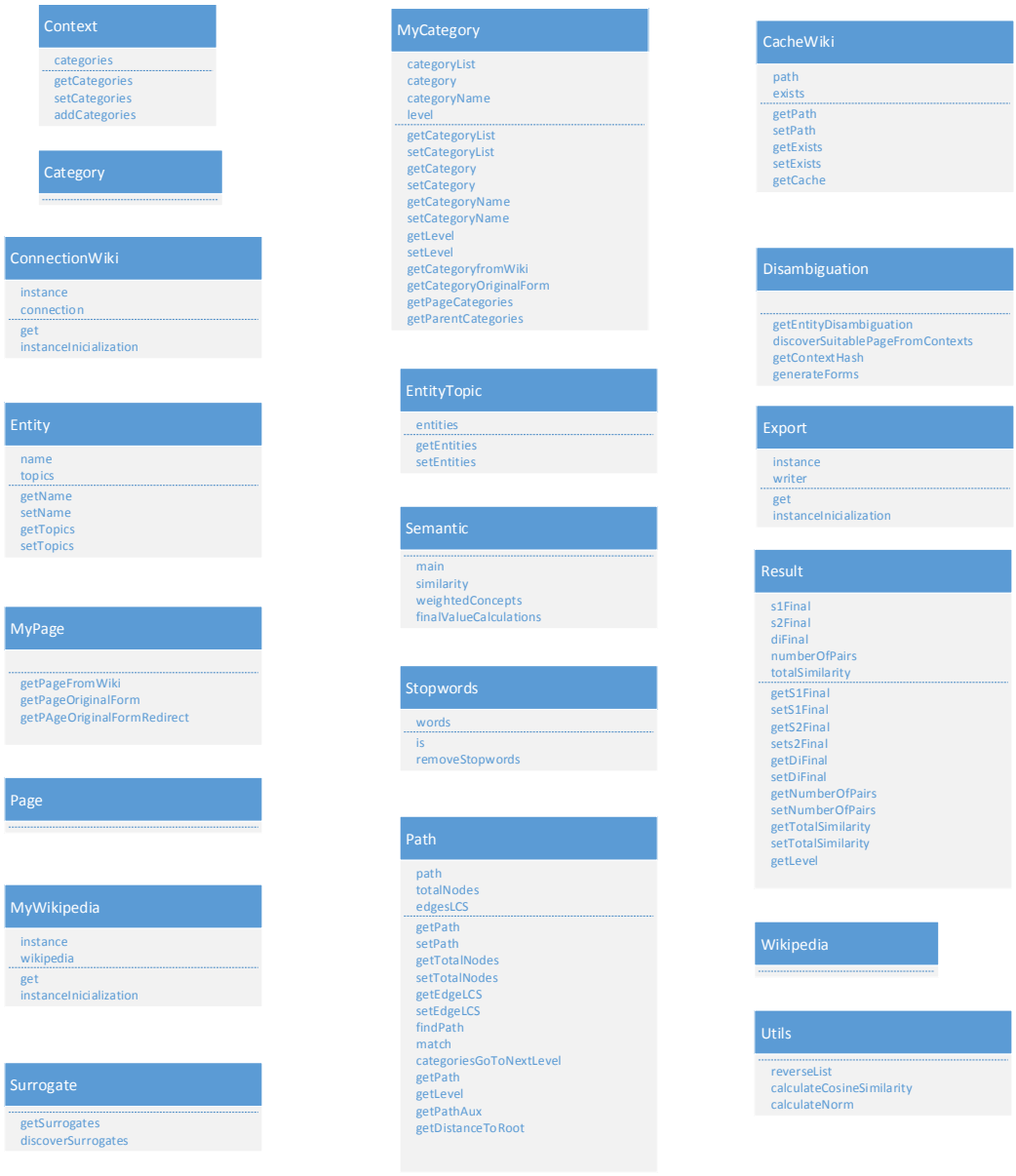

Figura 20 - Diagrama de classes.

5.1.1 Context

Esta classe serve de suporte à metodologia de desambiguação apresentada anteriormente. Ao instanciar esta classe torna-se possível a construção de contextos genéricos, que tanto podem ser do âmbito geral de várias entidades ou do âmbito específico de uma única entidade. Esta estrutura é constituída por um atributo principal do tipo lista de categoria (Wikipédia API), sendo necessário criar todos os seletores e modificadores do atributo lista. Também se sentiu a necessidade de criação de um método que facilitasse o acesso direto para adição de categorias ao contexto.

5.1.2 MyCategory

Tonou-se necessário a construção de uma camada de classes que trabalha sobre a API da Wikipédia. Isto porque as funcionalidades fornecidas pela API, apesar de serem úteis, são insuficientes para se atingir os objetivos propostos. Decidiu-se usar o prefixo “My” em todas as classes criadas com este propósito, visto ser uma forma simples de distinção. Esta classe contem os atributos “category”, “categoryList”, “categoryName” e “Level”, que são fundamentais para encontrar caminhos entre tópicos na ontologia de categorias da Wikipédia. Além dos modificadores e seletores dos atributos mencionados anteriormente, foi necessário criar um conjunto de métodos auxiliares para ajudarem a retornar a categoria através de uma string de um tópico, as categorias pais de uma categoria e as páginas de uma determinada categoria.

5.1.3 CacheWiki

A necessidade desta classe surgiu visto que, através de observação, podíamos concluir que diversos caminhos eram procurados repetidas vezes na ontologia de categorias. Desse modo a eficiência poderia ser incrementada ao guardarmos todos os caminhos calculados ao longo do tempo, ficando estes disponíveis para futuros cálculos. Esta classe contem como atributo principal um “Path”: objeto caminho. Todos os seletores e modificadores estão disponíveis, assim como um método auxiliar para pesquisar se o caminho já foi calculado anteriormente.

5.1.4 Disambiguation

Trata-se de uma classe auxiliar cujo método principal recebe uma entidade com tópicos e por cada um destes tópicos verifica-se a sua existência e ambiguidade na base de dados da Wikipédia. Tendo isto em conta, o tópico é adicionado a uma variável do contexto Geral, ou à lista de desambiguações a executar. O método “GenerateForm” tenta gerar todas as formas sintáticas possíveis para um determinado tópico desconhecido. E por último, o método “DiscoverSuitablePageFromContexts” é responsável por descobrir qual a opção correta entre o conjunto de sugestões para desambiguação, com a ajuda do contexto.

5.1.5 Entity

A classe “Entity” foi idealizada para ser suficientemente genérica para conseguir receber todo o tipo de solicitações idealizadas para o contexto deste projeto. Possui como atributos uma string para guardar o seu nome e também uma lista de tópicos que guardam a sua caracterização e significado.

As instanciações desta classe são usadas ao longo de todo o projeto, logo os métodos de seleção dos atributos dos objetos são essenciais.

5.1.6 EntityTopic

A missão desta classe passa por receber e interpretar os pedidos externos de verificação de similaridade entre conceitos. Por omissão estes pedidos são executados recorrendo a xml (exemplo em anexo), tendo o pedido de conter os seguintes nós:

<Entities> (raiz)
    <Entity> (1,*)
        <Name> (1)
        <Topics> (1)
            <Topic> (1,*)

Nesta classe existe um atributo lista de entidades, que é inicializado com base na interpretação do ficheiro xml com os pedidos de processamento.

5.1.7 Export

Uma classe auxiliar que é fundamental para visualizar resultados dos cálculos da medida de semelhança e também, por vezes, efetuar o debug necessário do código. O ficheiro exportação é construído em formato CSV (comma separated values). Este ficheiro inclui vários detalhes, tais como todas as desambiguações, todos os caminhos calculados, todos os LCS, todos os valores intermédios e finais de similaridade.

5.1.8 MyPage

Tal como na classe MyCategory, foi mais uma vez necessário criar uma classe para trabalhar numa camada do objetos superior à da Wikipédia API. Neste caso específico, a MyPage. Possui o método “getPageFromWiki”, que recebe como argumento o nome do tópico e tenta extrair a página correta da base de dados. No método “GetPageOriginalForm” caso não se consiga encontrar a página na primeira tentativa, consulta-se a tabela de redireccionamentos da Wikipédia; desta forma, na maior parte das vezes pode-se retornar o id da página pretendida.

5.1.9 Stopwords

A classe Stopwords é usada para diversas operações de limpeza de palavras sem qualquer significado semântico nos nossos tópicos. Existe uma grande carga de solicitações sobre esta classe, logo surgiu a necessidade de assegurar uma eficiência de alto nível nas suas operações. O atributo “m_words” do tipo “HashSet” guarda todas as palavras indesejadas; este tipo de objetos permite um

acesso de desempenho extremo ao seu conteúdo. O método auxiliar “is” recebe uma string e verifica se esta está contida no “HasSet” de stopwords. O método “removeStopWords” recebe um conjunto de palavras e retorna esse mesmo conjunto sem stopwords.

5.1.10 Result

A necessidade de guardar um número substancial de variáveis com valores necessários ao cálculo final da medida de similaridade levou à construção desta classe. Os seus atributos são maioritariamente do tipo “double” e contam com os respetivos seletores e modificadores.

5.1.11 Path

Classe bastante importante para este projeto, visto ser responsável por assegurar que se encontram os caminhos necessários entre tópicos na ontologia de categorias da Wikipédia. Possui como atributos uma string para guardar os tópicos do caminho, o número de nós que existe no caminho e o número de nós que existe entre o nosso LCS e a raiz. Alem dos respetivos modificadores e seletores, contem um método para efetuar a procura do caminho, “GetPath” e respetivos métodos auxiliares. O método “getLevel” retorna a profundidade do caminho com que estamos a trabalhar.

5.1.12 MyWikipédia

Este objeto guarda a instância que permite a convecção à API da Wikipédia. Tendo em vista a maximização de desempenho e redução de complexidade, foi utilizado o padrão de desenho de software “Singleton”. Desta forma a classe é instanciada uma única vez e por conseguinte utiliza-se sempre o mesmo objeto de acesso à API ao longo de toda a execução, evitando o constante carregamento de dados da Wikipédia.

5.1.13 Surrogate

Por vezes os tópicos com que estamos a trabalhar podem ser demasiado específicos e existe necessidade de encontrar representantes não tão específicos. O método “GetSurrogates” recebe a lista de tópicos e retorna uma nova lista com representantes um pouco mais gerais, usando todas as capacidades das classes já comentadas anteriormente.

5.1.14 Utils

Esta classe possui uma coleção de métodos bastante úteis, mas que não fazem parte das bibliotecas do java neste momento. “isAllUpper” verifica se todos os caracteres de uma string são maiúsculos, “reverseList” inverte a ordem dos elementos de uma lista e “calculateCosineSimilarity” recebe os parâmetros necessários e efetua o cálculo da similaridade do cosseno.

5.1.15 Semantic

É a classe principal do projeto, encontrando-se aqui o método principal que se encarrega de iterar todas as entidades para as quais se pretende calcular a similaridade. O método “similarity” recebe duas listas de tópicos e solicita valores entre cada uma das ocorrências de cada lista, enquanto o método “weightedConcepts” recebe como parâmetro dois tópicos e calcula os respetivos caminhos e similaridade resultante.

5.2 Modelo Entidade Relação

A estrutura que guarda os dados da Wikipédia que serve de suporte a este projeto usa uma base de dados relacional como metodologia de armazenamento. O modelo relacional foi apresentado por por Peter Chen e publicado num artigo de 1976 (Chen, 1976). Nos dias de hoje é massivamente usado por toda a indústria de software. Um modelo entidade relação (modelo ER) é um modelo de dados para descrever dados ou representar aspetos da informação de um domínio de negócio. Os principais componentes dos MER são as entidades e as suas propriedades, que correspondem às tabelas e campos de uma base de dados comum.

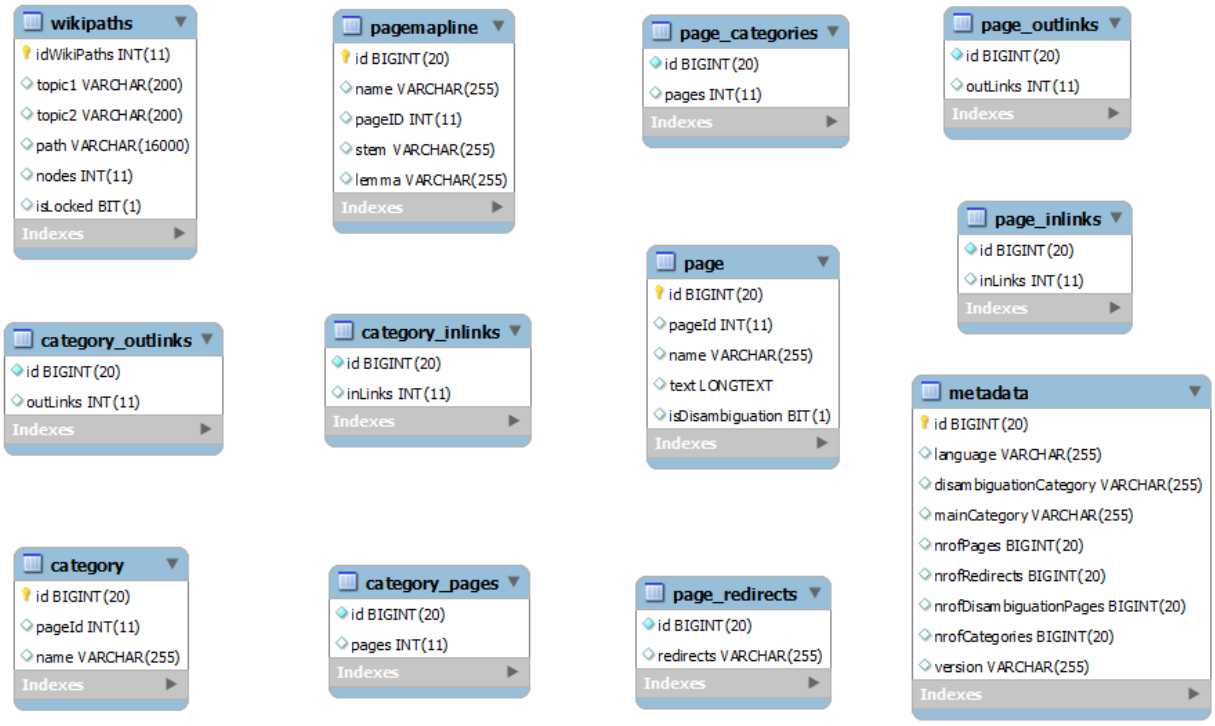

Figura 21- Modelo Entidade Relação

5.2.1 *Page*

A tabela “page” é responsável por guardar todas as páginas que existem na Wikipédia, (no caso da Wikipédia inglesa cerca de 5 milhões de páginas). Os campos desta tabela são o id único da página, o título da página, o seu conteúdo e por último a informação acerca de se a página é de desambiguação ou não.

5.2.2 *Page_inlinks*

Aqui guarda-se informação acerca de todas as páginas que apontam para uma outra determinada página. Ou seja, dado um id de página “P1” da tabela page, ao ser pesquisado na tabela de “inlinks” iremos obter como resultado os ids de todas as páginas que têm uma hiperligação para “P1” na Wikipédia.

5.2.3 *Page_outlinks*

Também é muito útil manter a informação acerca de todas as páginas para onde aponta uma determinada página. Similar à tabela anterior, mas neste caso para disponibilizar informação sobre quais os sítios para onde aponta cada página.

5.2.4 *Page_redirects*

A Wikipédia guarda diferentes formas de escrever os títulos das suas páginas. Uma vez que existem cerca de 70 milhões de expressões nesta tabela, faz uma média de 14 formas diferentes de escrever o título de cada página. A estrutura da tabela é relativamente simples, id da página representante e título alternativo.

5.2.5 *Page_categories*

Nesta tabela conseguimos obter informação acerca de quais categorias foram aplicadas a uma determinada página. Apenas se guardam os ids das páginas e ids das respetivas categorias.

5.2.6 *Category*

Possivelmente a tabela mais importante para este projeto, visto ser na rede de categorias que se encontra o caminho entre as propriedades das entidades. Aqui armazena-se todas as categorias que existem na Wikipédia, num estrutura simples, o identificador da categoria e o seu nome.

5.2.7 *Category_inlinks*

Tal como nas páginas, aqui nas categorias também existe grande interesse em saber quais as categorias que apontam para uma outra determinada categoria. Este tipo de ligação permite que seja possível calcular caminhos entre categorias.

5.2.8 *Category_outlinks*

Similar ao anterior, mas aqui obtém-se informação acerca das ligações de saída de uma determinada categoria.

5.2.9 *Category_pages*

Tabela de relacionamento entre categorias e páginas; aqui consegue-se saber, dado o id de uma categoria, todas as páginas que estão relacionadas com esta. A estrutura é simples, dois campos que servem como identificadores da categoria e das páginas.

5.2.10 *Wikipaths*

Esta tabela foi criada com o objetivo de guardar caminhos calculados anteriormente, evitando assim desperdício de recursos a calcular o mesmo caminho várias vezes ao longo do tempo. Em termos de estrutura, existe um campo para cada tópico, outro para o caminho e por fim o número de nós do caminho.

5.3. Conclusão

Em termos de desenvolvimento do software optou-se pela linguagem de programação java. A metodologia de programação é orientada por objetos, de modo a garantir a modularidade do código produzido. Tal possibilita uma boa manutenção do projeto ao longo da sua evolução e afinação.

Neste capítulo são também descritas as várias classes que compõem o software deste projeto, bem como o modelo entidade-relação da base de dados de suporte (tecnologia MySql).

Capítulo 6

Resultados, Validação e Verificação

Com o objetivo principal de averiguar a qualidade de resultados do sistema, decidiu-se sujeitá-lo a um processo de verificação e validação de outputs.

Uma das definições mais comuns para a área de verificação e validação de sistemas de software inteligentes é:

- Verificação – processo de reconhecimento de que o software implementa corretamente as funções que satisfazem a sua especificação.
- Validação – processo de certificação de que o output do sistema inteligente é equivalente ao do perito humano, quando fornecido os mesmos inputs.

Mesmo tratando-se de uma peça de software que se pretende inteligente com toda a complexidade que isto acarreta, é logico que se possam aplicar técnicas de verificação e validação de software tradicionais.

Seguidamente pretende-se fazer uma análise das técnicas mais conhecidas, começando pelos testes tradicionais, análise em tempo real, análise estática e também uma perspetiva de desempenho humana. Estes métodos variam tanto na confiança que transmitem, como no nível de conhecimento necessário para os executar.

6.1 Casos de Teste

Tradicionalmente, a verificação de software é executada através do teste de diversos cenários. O sistema que se quer testar é colocado num fluxo onde lhe são fornecidos os inputs e outputs. Cada teste que corre é uma sequência de inputs fornecidos e outputs esperados, correspondentes ao cenário pretendido. Os erros são assinalados quando o output recebido não corresponde ao esperado.

Até para sistemas de informação simples, a manutenção dos casos de teste é bastante complexa e dispendiosa. Isto porque é requerido um conhecimento profundo do sistema para assegurar que o número máximo de situações é coberto, usando o número mínimo de testes possível. A própria tarefa de execução dos testes consome bastante tempo, porque todo o código do programa tem de ser executado e seguidamente tudo tem de ser reiniciado antes de um novo arranque. Em sistemas de alguma dimensão, é perfeitamente natural que se gastem mais recursos na fase de testes do que na de desenvolvimento.

Enquanto os testes tradicionais podem ser o suficiente para testar sistemas de informação convencionais, tal pode não ser suficiente para sistemas com nível de complexidade superior, principalmente devido às numerosas situações de teste. Projetos deste género normalmente incorporam resposta a um leque muito abundante de inputs e os casos de teste só podem testar um número limitado de configurações possíveis. Desta forma é seriamente complexo medir o risco de erro do sistema.

6.2 Verificação em Tempo Real

A verificação em tempo real refere-se à identificação de potenciais problemas de um programa que se encontra em execução, através da análise dos valores das variáveis e dos eventos que são executados e desta forma é possível detetar comportamentos incorretos. Métodos simples deste género são usados há várias décadas. Por exemplo, praticamente todos os programadores criam “logs” nas suas aplicações, com o intuito de facilitar o “debug” das mesmas. Uma verificação em tempo real pode interpretar estes “logs” e automaticamente identificar situações anormais, visto que é difícil a análise manual na maior parte das vezes.

Esta análise pode ser levada a cabo em históricos de execução, mas também em tempo real, durante a execução do programa. Neste caso podem-se lançar mecanismos de recuperação do sistema como proteção contra falhas. Este tipo de mecanismo tem como vantagem usar muito poucos recursos computacionais e adapta-se a sistema de grande complexidade com relativa facilidade. Por outro lado não nos garante a deteção de todos os problemas, gerando por vezes falsos negativos.

6.3 Análise Estática

A análise estática consiste em explorar a estrutura do código fonte de um programa sem o executar. Inclui aspetos como o controlo do fluxo de execução do código, aproximação da gama de valores admitidos pelas variáveis e identificação de código importante a um set de variáveis ou funções.

A análise estática pode ser aplicada a um programa em fases iniciais do seu desenvolvimento, assim como na sua manutenção, ajudando a manter a sua qualidade. Normalmente ferramentas deste género são totalmente automáticas, analisando e informando o programador de potenciais melhorias. Existem também pontos fracos nesta abordagem, visto que algoritmos de alta precisão são também bastante complexos e necessitam de muito tempo para efetuar a análise do código. Algoritmos mais eficientes fazem aproximações que podem resultar em números altos de falsos positivos.

6.4 Desempenho Humano como Objetivo

Os pontos discutidos anteriormente têm como alvo principal a qualidade do código e ausência de bugs nas suas funcionalidades. Não minimizando a sua importância, pretende-se ir mais longe na verificação e validação desta abordagem, visto tratar-se de um sistema que se pretende inteligente.

Quando são verificadas todas as opções formais discutidas anteriormente, existe motivação para regressar aos objetivos mais primários da questão da verificação e validação de sistemas deste género, ou seja, garantir que o sistema tem como resultado uma solução ou comportamento equivalente ao de um especialista da área, tendo em conta os mesmos dados iniciais. Qual a razão para que a replicação da sabedoria humana seja tão universalmente aceite na verificação e validação de sistemas desta natureza? Computadores podem ser descritos como infraestruturas de modelação. Estes podem ser usados para simular vários sistemas físicos – naturais ou feitos pelo homem, entre eles a simulação de turbulência de ar através das asas de um avião, simulação de circuitos elétricos, campos eletromagnéticos entre outros fenómenos complexos.

Entre as várias definições de inteligências artificial, a maior parte delas faz referência à inteligência humana. Partindo da perspetiva de que um computador é uma máquina de modelos, a inteligência artificial não é mais do que uma tentativa de simulação da forma de trabalhar do cérebro humano. Existem vários domínios que não podem ser diretamente modelados, seja por não estarem corretamente compreendidos, ou por serem demasiado complexos. Mas os humanos como resultado da sua experiencia com estes domínios, são capazes de os modelar com recurso a heurísticas imperfeitas ou outras técnicas lhes permitem trabalhar no domínio. Sistemas inteligentes representam domínios que são difíceis de modelar. Mais precisamente, sistemas periciais não modelam diretamente o domínio, mas sim a interpretação humana do domínio. Assim sendo, é logico que seja uma avaliação humana a determinar a qualidade do sistema inteligente. Esta verificação e validação serão efetuadas recorrendo ao conhecimento do especialista do domínio de aplicação, em que o seu julgamento será diretamente comparado com o output da máquina. A capacidade do sistema será quantificada da seguinte forma: a aproximação dos resultados do algoritmo ao julgamento humano será considerado "feedback" positivo, o inverso será considerado resultados negativos. Será efetuada uma seleção prévia de entidades e respetivas propriedades de um determinado domínio de aplicação. A tarefa do especialista será, tendo em conta os interesses (propriedades), relacionar o conjunto de entidades entre si da melhor forma possível.

Na grande generalidade dos projetos de sistemas computorizados atribui-se grande importância à reação dos utilizadores no que concerne à aplicação e aprendizagem de funcionamento com as diversas funcionalidades do mesmo. No entanto, neste projeto pretende-se focar a atenção na melhoria contínua da capacidade de ajustamento de parâmetros, bem como na visualização e exploração de resultados (Charniak, 1973). Neste tipo de sistemas a validação será particularmente

importante dada a necessidade de afinação dos diversos parâmetros que contribuem para o resultado final fornecido pelo algoritmo.

6.4.1 Comparação de Similaridade: Sistema vs. Base

A tabela abaixo ilustra a comparação entre duas entidades, tendo como intuito validar o valor obtido pelo software (representado na coluna "Máquina") em comparação com a coluna "Base". As colunas "Entidade 1" e "Entidade 2" correspondem às entidades a comparar, sendo que a coluna "Máquina" apresenta o valor retornado pelo sistema automático e a coluna "Base" apresenta os valores obtidos através de uma metodologia *Ground Truth* (na impossibilidade de mobilizar especialistas humanos). O qualificador "Sim", significa que a Entidade 1 está próxima da Entidade 2 ao passo que o "Não" significa que se considerou que os assuntos não estão relacionados entre si.

Em reconhecimento de padrões o termo *Ground Truth* refere-se à informação fornecida através de observação direta, em contraste com a descoberta de informação por inferência. Ou seja, a coluna "Base" irá marcar com o qualificador "Sim" as ocorrências em que a Entidade 1 e o Evento da Entidade 2 são semelhantes.

Tabela 1- Cálculo de similaridade entre eventos e artigos científicos

Entidade 1	Entidade 2	Evento da Entidade 2	Máquina	Base
MODELSWARD	Variability Management supporting the Model-Driven Design of User Interfaces	MODELSWARD	0.837	Sim
MODELSWARD	Using Software Categories for the Development of Generative Software	MODELSWARD	0.881	Sim
MODELSWARD	Characterizing Workload Pattern	MODELSWARD	0.837	Sim
MODELSWARD	A Modular Method for Global System Behaviour Specification	MODELSWARD	0.833	Sim
MODELSWARD	Towards Product Lining Model-Driven Development Code Generators	MODELSWARD	0.962	Sim
MODELSWARD	Low Aerodynamic Drag Suit for Cycling - Design and Testing	ICSPORTS	0.555	Não
MODELSWARD	AT Training Zones for Different Pace Plans of 200 M Swimmers	ICSPORTS	0.688	Não
MODELSWARD	Links Between Mood States, Time Of Practise And Game Results In Brazilian Voleibol Players	ICSPORTS	0.688	Não
MODELSWARD	Flexible studs in prevention of football injuries	ICSPORTS	0.555	Não
MODELSWARD	Relationship Between Initiation of Gaze Stabilization and Angle of Head and Trunk Movement During a Jump with Full Turn	ICSPORTS	0.688	Não
MODELSWARD	Cogn-Evo - Document Clustering Based on a Cognitive-Evolutionary System	ICPRAM	0.753	Não
MODELSWARD	HyperSAX	ICPRAM	0.624	Não

MODELSWARD	Automatic Image Annotation using Convex Deep Learning Models	ICPRAM	0.823	Não
MODELSWARD	ZiZo	ICPRAM	0.671	Não
MODELSWARD	Binary Quantization for Selecting Central Points of Dimension Reduction Projection Simple-Map	ICPRAM	0.891	Não
MODELSWARD	TE Modes in Liquid Crystal Optical Fibers Embedded with Conducting Tape Helix Structure	PHOTOPTICS	0.551	Não
MODELSWARD	DPSK Signals Demodulation based on a Graded-index Multimode Fiber Mismatch Spliced between Two Single-mode Fibers	PHOTOPTICS	0.682	Não
MODELSWARD	Mechanical Characterisation of the Four Most Used Coating Materials for Optical Fibres	PHOTOPTICS	0.682	Não
MODELSWARD	Dual-core Photonic Crystal Fiber Magnetic Field Sensor based on selectively Magnetic Fluids Infiltrated	PHOTOPTICS	0.682	Não
MODELSWARD	Resource allocation and scheduling based on emergent behaviours in multi-agent scenarios	PHOTOPTICS	0.606	Não
ICSPORTS	Variability Management supporting the Model-Driven Design of User Interfaces	MODELSWARD	0.515	Não
ICSPORTS	Using Software Categories for the Development of Generative Software	MODELSWARD	0.426	Não
ICSPORTS	Characterizing Workload Pattern	MODELSWARD	0.818	Não
ICSPORTS	A Modular Method for Global System Behaviour Specification	MODELSWARD	0.796	Não
ICSPORTS	Towards Product Lining Model-Driven Development Code Generators	MODELSWARD	0.725	Não
ICSPORTS	Low Aerodynamic Drag Suit for Cycling - Design and Testing	ICSPORTS	0.834	Sim
ICSPORTS	AT Training Zones for Different Pace Plans of 200 M Swimmers	ICSPORTS	0.781	Sim
ICSPORTS	Links Between Mood States, Time Of Practise And Game Results In Brazilian Voleibol Players	ICSPORTS	0.752	Sim
ICSPORTS	Flexible studs in prevention of football injuries	ICSPORTS	0.834	Sim
ICSPORTS	Relationship Between Initiation of Gaze Stabilization and Angle of Head and Trunk Movement During a Jump with Full Turn	ICSPORTS	0.752	Sim
ICSPORTS	Cogn-Evo - Document Clustering Based on a Cognitive-Evolutionary System	ICPRAM	0.563	Não
ICSPORTS	HyperSAX	ICPRAM	0.469	Não
ICSPORTS	Automatic Image Annotation using Convex Deep Learning Models	ICPRAM	0.912	Não
ICSPORTS	ZiZo	ICPRAM	0.498	Não
ICSPORTS	Binary Quantization for Selecting Central Points of Dimension Reduction Projection Simple-Map	ICPRAM	0.768	Não
ICSPORTS	TE Modes in Liquid Crystal Optical Fibers Embedded with Conducting Tape Helix Structure	PHOTOPTICS	0.306	Não

ICSPORTS	DPSK Signals Demodulation based on a Graded-index Multimode Fiber Mismatch Spliced between Two Single-mode Fibers	PHOTOPTICS	0.971	Não
ICSPORTS	Mechanical Characterisation of the Four Most Used Coating Materials for Optical Fibres	PHOTOPTICS	0.395	Não
ICSPORTS	Dual-core Photonic Crystal Fiber Magnetic Field Sensor based on selectively Magnetic Fluids Infiltrated	PHOTOPTICS	0.594	Não
ICSPORTS	Resource allocation and scheduling based on emergent behaviours in multi-agent scenarios	PHOTOPTICS	0.971	Não
ICPRAM	Variability Management supporting the Model-Driven Design of User Interfaces	MODELSWARD	0.79	Não
ICPRAM	Using Software Categories for the Development of Generative Software	MODELSWARD	0.682	Não
ICPRAM	Characterizing Workload Pattern	MODELSWARD	0.764	Não
ICPRAM	A Modular Method for Global System Behaviour Specification	MODELSWARD	0.745	Não
ICPRAM	Towards Product Lining Model-Driven Development Code Generators	MODELSWARD	0.934	Não
ICPRAM	Low Aerodynamic Drag Suit for Cycling - Design and Testing	ICSPORTS	0.751	Não
ICPRAM	AT Training Zones for Different Pace Plans of 200 M Swimmers	ICSPORTS	0.792	Não
ICPRAM	Links Between Mood States, Time Of Practise And Game Results In Brazilian Voleibol Players	ICSPORTS	0.792	Não
ICPRAM	Flexible studs in prevention of football injuries	ICSPORTS	0.751	Não
ICPRAM	Relationship Between Initiation of Gaze Stabilization and Angle of Head and Trunk Movement During a Jump with Full Turn	ICSPORTS	0.792	Não
ICPRAM	Cogn-Evo - Document Clustering Based on a Cognitive-Evolutionary System	ICPRAM	0.94	Sim
ICPRAM	HyperSAX	ICPRAM	0.679	Sim
ICPRAM	Automatic Image Annotation using Convex Deep Learning Models	ICPRAM	0.994	Sim
ICPRAM	ZiZo	ICPRAM	0.778	Sim
ICPRAM	Binary Quantization for Selecting Central Points of Dimension Reduction Projection Simple-Map	ICPRAM	0.989	Sim
ICPRAM	TE Modes in Liquid Crystal Optical Fibers Embedded with Conducting Tape Helix Structure	PHOTOPTICS	0.674	Não
ICPRAM	DPSK Signals Demodulation based on a Graded-index Multimode Fiber Mismatch Spliced between Two Single-mode Fibers	PHOTOPTICS	0.684	Não
ICPRAM	Mechanical Characterisation of the Four Most Used Coating Materials for Optical Fibres	PHOTOPTICS	0.684	Não
ICPRAM	Dual-core Photonic Crystal Fiber Magnetic Field Sensor based on selectively Magnetic Fluids Infiltrated	PHOTOPTICS	0.684	Não

ICPRAM	Resource allocation and scheduling based on emergent behaviours in multi-agent scenarios	PHOTOPTICS	0.644	Não
PHOTOPTICS	Variability Management supporting the Model-Driven Design of User Interfaces	MODELSWARD	0.555	Não
PHOTOPTICS	Using Software Categories for the Development of Generative Software	MODELSWARD	0.528	Não
PHOTOPTICS	Characterizing Workload Pattern	MODELSWARD	0.603	Não
PHOTOPTICS	A Modular Method for Global System Behaviour Specification	MODELSWARD	0.686	Não
PHOTOPTICS	Towards Product Lining Model-Driven Development Code Generators	MODELSWARD	0.786	Não
PHOTOPTICS	Low Aerodynamic Drag Suit for Cycling - Design and Testing	ICSPORTS	0.696	Não
PHOTOPTICS	AT Training Zones for Different Pace Plans of 200 M Swimmers	ICSPORTS	0.514	Não
PHOTOPTICS	Links Between Mood States, Time Of Practise And Game Results In Brazilian Voleibol Players	ICSPORTS	0.68	Não
PHOTOPTICS	Flexible studs in prevention of football injuries	ICSPORTS	0.971	Não
PHOTOPTICS	Relationship Between Initiation of Gaze Stabilization and Angle of Head and Trunk Movement During a Jump with Full Turn	ICSPORTS	0.68	Não
PHOTOPTICS	Cogn-Evo - Document Clustering Based on a Cognitive-Evolutionary System	ICPRAM	0.673	Não
PHOTOPTICS	HyperSAX	ICPRAM	0.507	Não
PHOTOPTICS	Automatic Image Annotation using Convex Deep Learning Models	ICPRAM	0.753	Não
PHOTOPTICS	ZiZo	ICPRAM	0.712	Não
PHOTOPTICS	Binary Quantization for Selecting Central Points of Dimension Reduction Projection Simple-Map	ICPRAM	0.774	Não
PHOTOPTICS	Automatic Waveguide-fiber Alignment Algorithm based on Coupling Model	PHOTOPTICS	0.949	Sim
PHOTOPTICS	Mechanical Characterisation of the Four Most Used Coating Materials for Optical Fibres	PHOTOPTICS	0.69	Sim
PHOTOPTICS	Bragg grating solitons in semilinear dual-core system with cubic-quintic nonlinearity	PHOTOPTICS	0.69	Sim
PHOTOPTICS	Design of an optimized distal optic for non linear endomicroscopy	PHOTOPTICS	0.869	Sim
PHOTOPTICS	The Sensors Are Innovative in Internet of Things	PHOTOPTICS	0.585	Sim

Tabela 2- Cálculo de similaridade entre eventos e revisores

Entidade 1	Entidade 2	Evento da entidade 2	Máquina	Qualificador
MODELSWARD	Cesar Gonzalez-Perez	MODELSWARD	0.837	Sim
MODELSWARD	Cinzia Cappiello	MODELSWARD	0.869	Sim
MODELSWARD	Damiano Distante	MODELSWARD	0.935	Sim
MODELSWARD	Fergal Mc Caffery	MODELSWARD	0.837	Sim
MODELSWARD	Guglielmo de Angelis	MODELSWARD	0.981	Sim
MODELSWARD	Abdülkerim Kasim Baltaci	ICSPORTS	0.718	Não
MODELSWARD	Amir Ali Mohagheghi	ICSPORTS	0.585	Não
MODELSWARD	Andrey Koptyug	ICSPORTS	0.555	Não
MODELSWARD	Anthony Leicht	ICSPORTS	0.555	Não
MODELSWARD	Jan Cabri	ICSPORTS	0.641	Não
MODELSWARD	Apostolos Papadopoulos	ICPRAM	0.867	Não
MODELSWARD	Bjoern Schuller	ICPRAM	0.633	Não
MODELSWARD	Elena Marchiori	ICPRAM	0.633	Não
MODELSWARD	Fabio Gonzalez	ICPRAM	0.768	Não
MODELSWARD	Francisco Martínez Álvarez	ICPRAM	0.823	Não
MODELSWARD	Javier PELAYO ZUECO	PHOTOPTICS	0.682	Não
MODELSWARD	Alexander Argyros	PHOTOPTICS	0.682	Não
MODELSWARD	André Nicolet	PHOTOPTICS	0.682	Não
MODELSWARD	Carlos Saavedra	PHOTOPTICS	0.682	Não
MODELSWARD	David Plant	PHOTOPTICS	0.682	Não
ICSPORTS	Cesar Gonzalez-Perez	MODELSWARD	0.818	Não
ICSPORTS	Cinzia Cappiello	MODELSWARD	0.688	Não
ICSPORTS	Damiano Distante	MODELSWARD	0.901	Não
ICSPORTS	Fergal Mc Caffery	MODELSWARD	0.818	Não
ICSPORTS	Guglielmo de Angelis	MODELSWARD	0.777	Não
ICSPORTS	Abdülkerim Kasim Baltaci	ICSPORTS	0.934	Sim
ICSPORTS	Amir Ali Mohagheghi	ICSPORTS	0.936	Sim
ICSPORTS	Andrey Koptyug	ICSPORTS	0.834	Sim
ICSPORTS	Anthony Leicht	ICSPORTS	0.834	Sim
ICSPORTS	Jan Cabri	ICSPORTS	0.883	Sim
ICSPORTS	Apostolos Papadopoulos	ICPRAM	0.563	Não
ICSPORTS	Bjoern Schuller	ICPRAM	0.435	Não
ICSPORTS	Elena Marchiori	ICPRAM	0.576	Não
ICSPORTS	Fabio Gonzalez	ICPRAM	0.771	Não
ICSPORTS	Francisco Martínez Álvarez	ICPRAM	0.75	Não
ICSPORTS	Javier PELAYO ZUECO	PHOTOPTICS	0.395	Não
ICSPORTS	Alexander Argyros	PHOTOPTICS	0.594	Não
ICSPORTS	André Nicolet	PHOTOPTICS	0.696	Não
ICSPORTS	Carlos Saavedra	PHOTOPTICS	0.696	Não

ICSPORTS	David Plant	PHOTOPTICS	0.395	Não
ICPRAM	Cesar Gonzalez-Perez	MODELSWARD	0.764	Não
ICPRAM	Cinzia Cappiello	MODELSWARD	0.867	Não
ICPRAM	Damiano Distante	MODELSWARD	0.959	Não
ICPRAM	Fergal Mc Caffery	MODELSWARD	0.764	Não
ICPRAM	Guglielmo de Angelis	MODELSWARD	0.933	Não
ICPRAM	Abdülkerim Kasim Baltaci	ICSPORTS	0.825	Não
ICPRAM	Amir Ali Mohagheghi	ICSPORTS	0.663	Não
ICPRAM	Andrey Koptyug	ICSPORTS	0.751	Não
ICPRAM	Anthony Leicht	ICSPORTS	0.751	Não
ICPRAM	Jan Cabri	ICSPORTS	0.847	Não
ICPRAM	Apostolos Papadopoulos	ICPRAM	0.954	Sim
ICPRAM	Bjoern Schuller	ICPRAM	0.765	Sim
ICPRAM	Elena Marchiori	ICPRAM	0.866	Sim
ICPRAM	Fabio Gonzalez	ICPRAM	0.993	Sim
ICPRAM	Francisco Martínez Álvarez	ICPRAM	0.956	Sim
ICPRAM	Javier PELAYO ZUECO	PHOTOPTICS	0.684	Não
ICPRAM	Alexander Argyros	PHOTOPTICS	0.684	Não
ICPRAM	André Nicolet	PHOTOPTICS	0.684	Não
ICPRAM	Carlos Saavedra	PHOTOPTICS	0.684	Não
ICPRAM	David Plant	PHOTOPTICS	0.684	Não
PHOTOPTICS	Cesar Gonzalez-Perez	MODELSWARD	0.603	Não
PHOTOPTICS	Cinzia Cappiello	MODELSWARD	0.682	Não
PHOTOPTICS	Damiano Distante	MODELSWARD	0.774	Não
PHOTOPTICS	Fergal Mc Caffery	MODELSWARD	0.686	Não
PHOTOPTICS	Guglielmo de Angelis	MODELSWARD	0.813	Não
PHOTOPTICS	Abdülkerim Kasim Baltaci	ICSPORTS	0.824	Não
PHOTOPTICS	Amir Ali Mohagheghi	ICSPORTS	0.686	Não
PHOTOPTICS	Andrey Koptyug	ICSPORTS	0.771	Não
PHOTOPTICS	Anthony Leicht	ICSPORTS	0.593	Não
PHOTOPTICS	Jan Cabri	ICSPORTS	0.81	Não
PHOTOPTICS	Apostolos Papadopoulos	ICPRAM	0.823	Não
PHOTOPTICS	Bjoern Schuller	ICPRAM	0.708	Não
PHOTOPTICS	Elena Marchiori	ICPRAM	0.664	Não
PHOTOPTICS	Fabio Gonzalez	ICPRAM	0.7	Não
PHOTOPTICS	Francisco Martínez Álvarez	ICPRAM	0.759	Não
PHOTOPTICS	Javier PELAYO ZUECO	PHOTOPTICS	0.719	Sim
PHOTOPTICS	Alexander Argyros	PHOTOPTICS	0.878	Sim
PHOTOPTICS	André Nicolet	PHOTOPTICS	0.718	Sim
PHOTOPTICS	Carlos Saavedra	PHOTOPTICS	0.869	Sim
PHOTOPTICS	David Plant	PHOTOPTICS	0.719	Sim

6.4.2 *Análise de Resultados*

Tendo em conta os resultados obtidos na subsecção anterior, pretende-se agora efectuar uma análise mais cuidada dos dados, começando pela construção do histograma de frequência dos resultados da máquina, cálculo do desvio padrão e variância. Através da análise deste histograma espera-se compreender melhor a frequência dos resultados e com isto proceder à criação de classes para comparação direta com a coluna Base (*Ground Truth*).

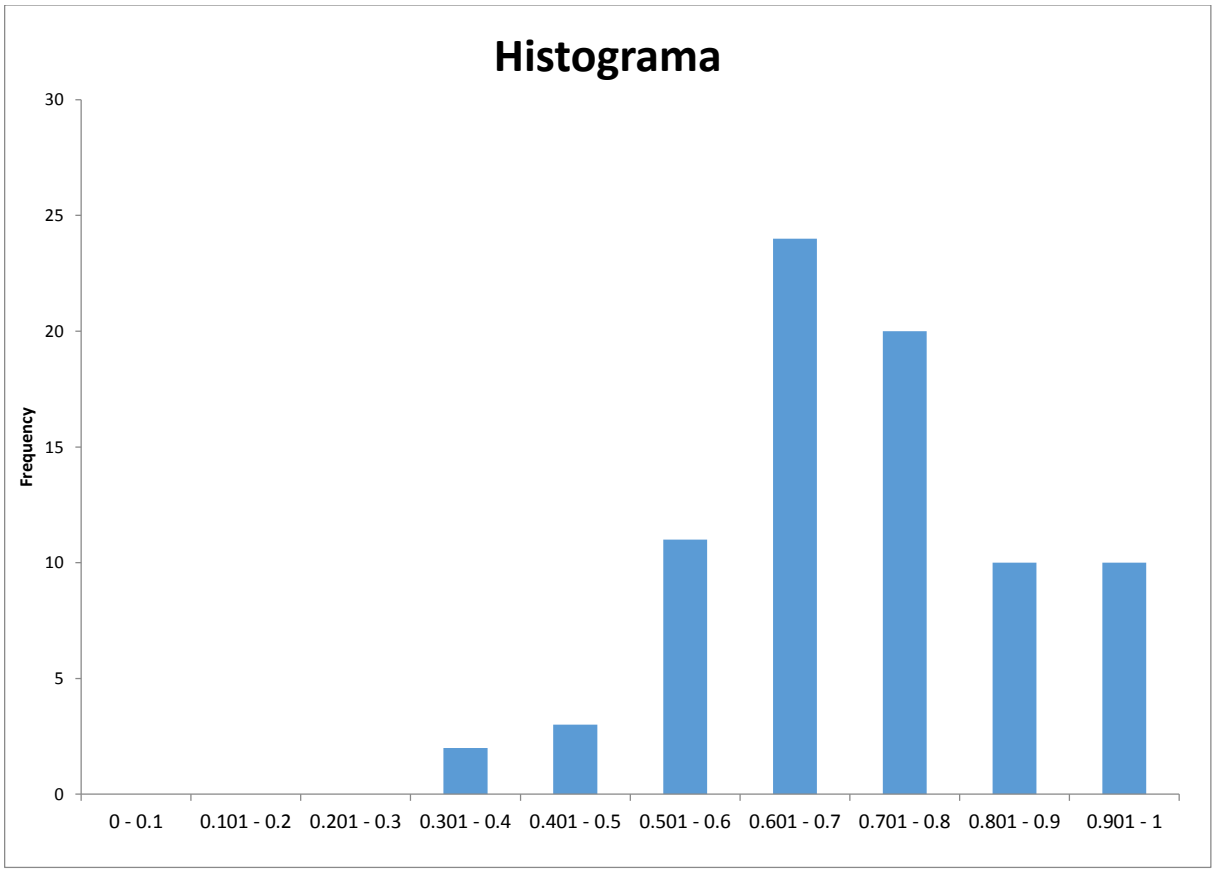

Figura 22-Histograma dos resultados da máquina

Estatisticamente temos uma média de resultados de 0.72, apresentado um desvio padrão de 0.146 e uma variância no valor de 0.02. O desvio padrão poderá ser usado para a criação de diversas classes no intervalo de valores do output da máquina, por exemplo. Para esta validação pretende-se apenas a criação de 2 classes, ficando o "threshold" situado no valor da média calculado anteriormente.

Em termos de output obtido pela máquina, valores inferiores a "0,72" irão pertencer à classe "Não", enquanto que os valores superiores a este número, irão pertencer à classe "Sim". Desta forma estamos em condições de comparar diretamente a decisão da máquina com a coluna "Base".

A seguinte tabela demonstra a informação sobre a Precisão e o Recall (mais informação sobre estes conceitos pode ser consultada na secção 2.4 deste documento) dos resultados da máquina devidamente classificados em relação à informação contida em “Base”.

Tabela 3 - Precisão e Recall

	Entidades Compatíveis(Sim)	Entidades não Compatíveis(Não)
Precisão	0.407	0.911
Recall	0.825	0.6

6.5 Conclusão

Neste capítulo verificaram-se vários métodos de verificação e validação, com intuito de selecionar aquele que mais se adequa ao projeto desenvolvido.

Efectuaram-se também várias iniciativas de análise de resultados, com o objetivo de verificar/validar e compreender os resultados fornecidos pela ferramenta. De uma forma geral, considera-se os resultados estão dentro do esperado e que são bastante promissores no que diz respeito a futuras evoluções do software. De salientar que é justificavel a baixa precisão nas “Entidades Compatíveis”, uma vez que os presentes eventos partilham diversos tópicos, existindo assim compatibilidade entre entidades, compatibilidade essa que não está refletida na abordagem de comparação *Ground Truth*. Neste caso, os eventos têm tópicos em comum e portanto não são totalmente independentes.

Acredita-se que os níveis de precisão facilmente serão incrementados se for possível a colaboração de especialistas de cada dominio de cada evento, visto que estes têm uma ampla capacidade de inferir conhecimento na sua decisão de compatibilidade.

Capítulo 7
Conclusão

Neste capítulo encontrar-se-ão as conclusões relativamente ao projeto elaborado, bem como o trabalho futuro que pode ser desenvolvido por forma a melhorar a métrica de dissimilaridade desenvolvida.

7.1. Conclusões

Quando fazemos uma pesquisa na internet pretendemos que o resultado obtido vá ao encontro do esperado, o que nem sempre acontece devido à dificuldade de classificar a informação. Os sistemas de descoberta do conhecimento centram-se na procura de palavras idênticas, o que leva a várias limitações, entre as quais a falta de capacidade de interpretação. A compreensão do significado semântico de uma expressão, ou conjunto de expressões, embora intuitiva nos seres humanos, é de difícil replicação em sistemas computacionais.

A análise semântica, isto é, interpretar conjuntos de palavras deduzindo o seu significado conceptual, é um aspeto crucial para a catalogação de informação. Em termos de ciências da computação já existe bastante trabalho referente ao cálculo de similaridade baseado em análise sintática estatística, embora esta vertente menospreze a importância do conhecimento das ligações bem como a relação entre as entidades existentes no nosso quotidiano. O próprio ato de armazenar ordenadamente quantidades massivas de informação é uma tarefa complexa. A categorização e relacionamento de toda a informação de uma forma lógica exige um vasto conhecimento sobre todos os domínios de informação. Esta dificuldade está relacionada com a criação e utilização correta de uma ontologia, uma vez que processar, relacionar e extrair informação em tempo útil podem revelar-se problemas complexos.

Tendo em conta esta problemática pretendeu-se criar um sistema de cálculo de semelhança semântica entre entidades, sistema esse a ter por base uma ontologia de conhecimento de categorias da Wikipédia. Este sistema permite a navegação entre nós da ontologia, retornando informação sobre os repetivos caminhos.

A implementação da medida proposta tem por base o princípio de que todos os conceitos da ontologia estão conectados entre si. A medida de semelhança é executada mediante uma sequência de passos apoiados em fórmulas matemáticas: cálculo de semelhança entre conceitos, similaridade entre entidades, cálculo do peso na medida de similaridade dos nós vizinhos mais próximos. A medida proposta é pois uma combinação dos componentes calculados nos passos anteriormente

mencionados, tomando em conta o peso dos nós LCS (Least Common Subsumer) bem como aspetos relacionados com a computação da densidade (componente introduzida no cálculo da medida de semelhança de modo a diminuir as diferenças na densidade de relações na rede de categorias ao longo de toda a ontologia) e com a desambiguação de termos.

7.2. Trabalho Futuro

Trata-se de um metodologia e de um software que não estão completamente terminados e que estão em progresso e evolução continua, incluindo tarefas de "Fine-tuning" que são executadas regularmente tendo em conta a melhoria dos resultados. Destas tarefas poderá sempre resultar a necessidade de exploração de novas técnicas, como foi o caso do artigo científico da densidade de caminhos em ontologias variáveis (Rodrigues, Filipe, & Fred, 2014). A actualização contínua da base de conhecimento da Wikipédia também contribui em grande escala para a melhoria de resultados, sendo que esta característica é uma das grandes vantagens da abortagem utilizada neste projeto.

As áreas de aplicação desta ferramenta são bastante vastas e não se restringem aos casos apresentados na secção de validação, nomeadamente:

- Concorrência entre eventos - caso em que ambas as entidades comparadas são do mesmo tipo, naturalmente as entidades mais próximas entre si semanticamente são consideradas concorrentes.
- Sugestão de comunidades – relação semântica entre autores através dos seus interesses de científicos.
- Distribuição de artigos científicos por revisores – tópicos dos artigos científicos são medidos semanticamente contra os interesses dos revisores, os mais próximos semanticamente devem ser atribuídos.
- Sugestão de artigos científicos a autores – se um determinado autor está a publicar artigos numa determinada área, o sistema pode comparar os tópicos deste artigo com os existentes e efectuar sugestões de leitura de respetiva citação.
- Evento mais adquado para submissão – encaminhar o autor para o evento que mais se relaciona com o artigo que pretende submeter.
- Sugestão de sessões a assistir – tendo em conta os papers que são apresentados numa determinada sessão e os interesses do autor, sugerir automaticamente as sessões que que não deve perder durante o evento.

Bibliografia

Apple Computer, Inc. (2015). *Search Basics*. Retrieved from https://developer.apple.com/library/mac/documentation/UserExperience/Conceptual/SearchKitConcepts/searchKit_basics/searchKit_basics.html

Blockeel, H., Ramon, J., Shavlik, J., & Tadepalli, P. (2007). *Proceedings of the 17th International Conference on Inductive Logic Programming.* Berlin, Heidelberg: Springer-Verlag.

Bollegala, D., Matsuo, Y., & Ishizuka, M. (2007). Measuring Semantic Similarity between Words Using Web Search Engines. *In International World Wide Web Conference*.

Charniak, E. (1973). Jack and Janet in search of a theory of knowledge. *Advanced Papers from the Third International Joint Conference on Artificial Intelliente* (pp. 337-343). Stanford, Cal.: Los Altos, Cal. W. Kaufmann.

Chen, P. (1976). The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems (TODS)*, 9-36.

CISCO. (2014, June 10). *Cisco Visual Networking Index: Forecast and Methodology, 2013–2018*. Retrieved from http://www.cisco.com/c/en/us/solutions/collateral/serviceprovider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf

Coleman, T. F., & More, J. J. (1983). Estimation of sparse Jacobian matrices and graph coloring problems. *SIAM Journal on Numerical Analysis*, 187-209.

Deerwester et al., (. S. (1990). Indexing by latent semantic analysis. *JASIS,* , 391-407.

Filipe, J. B. (2000). *Normative organisational modelling using intelligent multi-agent systems.* Staffordshire University.

Giles, J. (2005). Internet encyclopedias go head to head. *Nature*.

Goyvaerts, J. &. (2009). *Regular expressions cookbook.* O'reilly.

Greene, D., Cunningham, P., & Mayer, R. (2008). *Unsupervised Learning and Clustering In Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval.*

Kamel, M. M. (2008). Automatic extraction of domain-specific stopwords from labeled documents. *Advances in information retrieval*, 222-233.

Kozen, D. (1991). A Completeness Theorem for Kleene Algebras and the Algebra of Regular Events. *Proceedings of the 6th Annual IEEE Symposium on Logic in Computer Science*, 214-225.

Kumar, D. K. (2013). Network Anomaly Detection: A Machine Learning Perspective. *Chapman*.

Kumar, E. (2011). *Natural language processing.* IK International Pvt Ltd.

Marques, J. S. (2005). *Reconhecimento de Padrões: métodos estatísticos e neuronais.* IST Press.

Medina, L. A., Fred, A. L., Rodrigues, R., & Filipe, J. (2012). Measuring entity semantic relatedness using wikipedia. *KDIR*, 431–437.

Mertz, D. (2003). *Text Processing with Python.* Boston: Addison-Wesley Longman Publishing Co.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 130-137.

Rodrigues, R., Filipe, J., & Fred, A. (2014). Semantic Relatedness with Variable Ontology Density.

Salton , G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage*, 513-523.

Sholom, M. W., Indurkhya, N., & Zhang, T. (2010). Fundamentals of Predictive Text Mining(1st ed.). *Springer Publishing Company*.

Stoimen. (2015). *Computer Algorithms: Finding the Lowest Common Ancestor*. Retrieved from http://www.stoimen.com/blog/2012/08/24/computer-algorithms-finding-the-lowest-common-ancestor/

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). Introduction to Data Mining. Minnesota : Addison-Wesley.

Turing, A. (1950). Computing machinery and intelligence. *Mind,* , 433-460.

Van Rijsbergen, C. (1979). *Information Retrieval.* University of Glasgow.

Wikipedia. (2015, September 5). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Wikipedia

Wikipedia. (2015). *Deterministic finite automaton - Wikipedia, the free encyclopedia*. Retrieved from https://en.wikipedia.org/wiki/Deterministic_finite_automaton#/media/File:DFA_example_multiplies_of_3.svg

Wikipedia. (2015). *Hyponymy and hypernymy - Wikipedia, the free encyclopedia*. Retrieved from https://en.wikipedia.org/wiki/Hyponymy_and_hypernymy#/media/File:Hyponymsandhypernyms.jpg

Wikipédia. (2015). *Teste de Turing – Wikipédia, a enciclopédia livre*. Retrieved from https://pt.wikipedia.org/wiki/Teste_de_Turing#/media/File:Turing_Test_version_3.png

Wordnet. (2015). Retrieved from http://wordnetweb.princeton.edu/perl/webwn?s=automobile&sub=Search+WordNet&o2=&o0=1&o8=1&o1=1&o7=&o5=&o9=&o6=&o3=&o4=&h=

Zesch, T., Müller, C., & Gurevych, I. (2008). Extracting Lexical Semantic Knowledge fromWikipedia and Wiktionary. *In Proceedings of the Conference on Language Resources and Evaluation*.

Anexo I

Artigo

O artigo científico aqui anexado constitui um trabalho de síntese de algumas ideias que foram criadas e desenvolvidas no âmbito deste projeto. Foi publicado e apresentado oralmente numa conferência internacional, nomeadamente a *International Conference on Knowledge Discovery and Information Retrieval* 2014 decorrida em Roma, em Outubro de 2014.

Semantic Relatedness with Variable Ontology Density

Rui Rodrigues[1], Joaquim Filipe[1] and Ana L. N. Fred[2]
[1]*Escola Superior de Tecnologia, Instituto Politécnico de Setúbal, Setúbal, Portugal*
[2]*Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa, Portugal*
rrodrigues@insticc.org, j.filipe@est.ips.pt, afred@lx.it.pt

Keywords: Semantic Relatedness, Wikipédia Categories, Ontological Entities, Taxonomical Density.

Abstract: In a previous work, we proposed a semantic relatedness measure between scientific concepts, using Wikipédia categories network as an ontology, based on the length of the category path. After observing substantial differences in the arc density of the categories network, across the whole graph, it was concluded that these irregularities in the ontology density may lead to substantial errors in the computation of the semantic relatedness measure. Now we attempt to correct for this bias and improve this measure by adding the notion of ontology density and proposing a new semantic relatedness measure. The proposed measure computes a weighed length of the category path between two concepts in the ontology graph, assigning a different weight to each arc of the path, depending on the ontology density in its region. This procedure has been extended to measure semantic relatedness between entities, an entity being defined as a set of concepts.

1 INTRODUCTION

The *semantic relatedness* between two concepts indicates the degree in which these concepts are related in a conceptual network, by computing not only their semantic similarity but actually any possible semantic relationship between them (Ponzetto and Strube, 2007) (Gracia and Mena, 2008).

Computational semantic relationship techniques can be placed into one of two categories:

- Distributional measures that rely on unstructured data, such as large sets. The underlying assumption is that if similar words appear in similar contexts, then they should have similar meanings.
- Measures based on structured databases, such as taxonomies or ontologies, where semantic relationships are captured.

This work focuses on the second relationship measuring category, and uses Wikipédia page categories as a taxonomy.

The proposed measure considers not merely the number of arcs in the graph between the nodes that represent each concept, but also their relationship in the taxonomy. In our work we used the English version of Wikipédia[1], based on which we analyzed relationships between concepts by building paths from start to destination nodes in the category network.

[1]http://en.wikipédia.org

This procedure has been extended to measure semantic relatedness between entities, an entity being defined as a set of properties, i.e. concepts.

Since the Wikipédia ontology is in constant development, it was observed that some regions are far more developed than others. In this paper we propose to add to our previously developed measure of semantic relatedness (Medina et al., 2012) the notion of density, as a function of the number of incoming and outgoing links to/from a node in the conceptual graph. This means that the semantic relatedness between concepts needs is inversely weighed by the density of the region of the path between concepts. We will propose a technique to compute the density of this region.

The remaining sections of this document are organized as follows: in Section 2 we describe related work in this area; Section 3 presents the proposed measure of semantic relatedness with the inverse density ponderation and in section 4 are presented the results obtained after applying this measure to a set of entities. Finally, in Section 5 we draw the main conclusions and identify opportunities for future work.

2 RELATED WORK

Given two words or expressions represented in a taxonomy, the computation of the semantic relatedness between these two objects may be transformed into the evaluation of their conceptual distance in the conceptual space generated by a taxonomy (Jiang and Conrath, 1997), being that each object is represented by a node in the resulting graph.

Semantic relatedness measures in hierarchical taxonomies can be categorized into three types (Slimani et al., 2006):

1. **Information Content or Node-based:** evaluation of the information content of a concept represented by a node such as described in (Resnik, 1999). The semantic relatedness between two concepts reflects the amount of shared information between them, generally in the form of their least common subsumer (LCS).
2. **Path or Edge-based:** evaluation of the distance that separates concepts by measuring the length of the edge-path between them (Wu and Palmer, 1994) (Rada et al., 1989). A weight is assigned to each edge, being that the weight computation must reflect some of the graph properties (network density, node depth, link strength, etc.) (Jiang and Conrath, 1997)
3. **Hybrid:** a combination of the former two (Jiang and Conrath, 1997) (Leacock and Chodorow, 1998).

Lexical databases, such as WordNet, have been explored as knowledge bases to measure the semantic similarity between words or expressions. However, WordNet provides generic definitions and a somewhat rigid categorization that does not reflect the intuitive semantic meaning that a human might assign to a concept.

In this paper we use the english version of the Wikipédia[2], a web-based encyclopedia which has approximately 4 million articles edited and reviewed by volunteers. The contributors are asked to assign these articles to one or more categories: Wikipédia may be thus viewed as either a folksonomy (Nastase and Strube, 2008) or a Collective Knowledge Base (Zesch et al., 2008), where human knowledge and human intuition on semantic relationships emerges in the form of a category network. It is then natural that this webresource has been increasingly explored as a conceptual feature space, such that articles and categories are represented as nodes in the Wikipédia graph.

Techniques such Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007) represent texts in the high-dimensional concept space of Wikipédia as weighted vectors. A textual fragment is thus considered as a weighted mixture of a predetermined set of "natural" concepts. Wikipédia Linkbased Measure (WLM), first described in (Milne and Witten, 2008), uses its hyperlink structure, rather than the category hierarchy or textual content to compute semantic relatedness. In (Gouws et al., 2010) semantic relatedness is computed by spreading activation energy over the aforementioned hyperlink structure.

Measurement of semantic similarity between concept sets can provide particular value for tasks concerning the semantics of entities (Liu and Birnbaum, 2007). An entity may represent, for instance, (1) an author, by means of his/her research interests, (2) a publication, such as a scientific journal, by means of its main topics, (3) a conference, by means of its submission topics. In Information Retrieval, the similarity between documents is generally estimated by means of their Vector Space Models. Each feature vector represents the bag-of-words of the respective document, assigning a weight to each feature/term that reflects its importance in the overall context of either the document or the document set. The definition of entity can also be extended to represent a document, where instead of a weighted feature vector, we have a set of terms that can be related to other entities (which may also be documents or other types of entities) by means of a semantic relatedness measure between entities, such as the one presented in this paper.

3 SIMILARITY RELATEDNESS MEASURE

The implementation of the proposed measure is based on the assumption that each pair of concepts is connected by a category path.

The proposed relatedness measure is computed from the following sequence of steps

Distance between Concepts - Weighted Edges Sum. Let c_1 and c_2 be two concepts represented in the Wikipédia categories network. Find the shortest category path between the concepts. Compute the edge-based semantic relatedness between c_1 and the LCS node, which is the sum of the weights of the edges that link c_1 to the LCS node. Repeat this procedure to find the edge-based semantic relatedness between c_2 and the LCS node.

The overall edge-based relatedness measure between the two concepts is given by

$$d(c_1, c_2) = \frac{\sum_{i=0}^{I} w_i^1 + \sum_{i=0}^{J} w_i^2}{\sum_{i=0}^{R} w_i^1 + \sum_{i=0}^{R} w_i^2} \qquad (1)$$

where $w^1{}_i$ is the weight of the edge with index i in the

[2] http://en.wikipédia.org

category path between c_1 and the LCS category, w^2_i is the weight of the edge with index i in the category path between c_2 and the LCS category, I is the depth of the last edge of the path that connects c_1 to the LCS and J is the depth of the last edge of the path that connects c_2 to the LCS category, with R denoting the index of the last edge in the path between the root node and a concept node, with the restriction that this path must include the LCS.

Edge-based Similarity between Entities. Given two entities E_1 and E_2 represented by discrete sets of concepts $C_1 = \{c^1_1,...,c^1_n\}$ and $C_2 = \{c^2_1,...,c^2_m\}$, respectively, we define the edge-based distance between sets $D(E_1,E_2)$ is

$$D(E_1,E_2) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} d(c_i,c_j)}{n \times m} \qquad (2)$$

Finally, we have the following similarity measure between entities

$$S(E_1,E_2) = 1 - D(E_1,E_2) \qquad (3)$$

3.1 Computation of Shortest Paths in the Wikipédia Graph

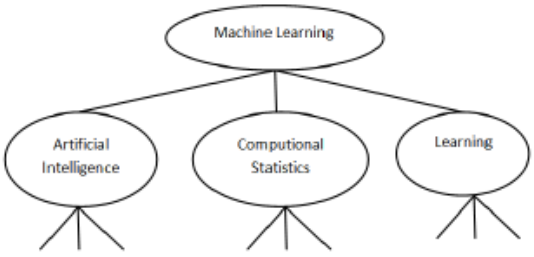

Figure 1: Instatiation of the concept ”Machine Learning”.

Several versions of the Wikipédia maybe be accessed at http://dumps.wikimedia.org/backup-index.html. For the results presented in this paper, we used a recent english version. To store all the Wikipédia pages and links we used the MySQL structure provided by the Java Wikipédia Library API (available at http://www.ukp.tudarmstadt.de/software/jwpl/), further described in (Zesch et al., 2008). This API also helps us to determine if a page is of the ”Disambiguation pages” type. We did not, however, used the API to build category paths, having specifically devised procedures for this task.

Each Wikipédia object of the type CATEGORY is assigned a level within the current search and a list of its nearest neighbors, when examined for shortest path computation. By regarding this shortest-path search as a tree-search, each instance of the object category will be a leaf of the tree.

```
1  Category nextLevel(Category c)
2        Begin
3        ForEach Category in c.List
4              Begin
5                    If (IteratedCategory.List = null)
                           ->leaf node
6                          Begin
7                          WikiList=wikipédia.
                                 GetAboveLevelCategories ();
8                          c.List=newList;
9                    End
10                   Else
11                         Begin
12                         NextLevel(IteratedCategory);
                                 ->recursive method
13                         End
14                   End
15             End
```

Listing 1: Procedure to examine the upper level of a node.

After the instantiation of concept *Machine Learning* (see Figure 1), all of its parent categories (”Artificial Intelligence”, ”Learning” and ”Computational Statistics”) will have a level attribute of two. The instatiation of each of these categories will return their corresponding list of parent categories and a level attribute of 3 and so on. The pseudo code in Listing 1 illustrates this procedure.

For each computation of a category path between two concepts, two trees are built, one for each concept. The level attribute will grow until the algorithm finds a common ancestor (the LCS).

```
1  Void Main()
2        Begin
3          List c1 = wikipédia.
                 GetAboveLevelCategories("concept1") ;
4            List c2 = wikipédia.
                 GetAboveLevelCategories("concept2 ");
5
6          While(ExistMatch(c1,c2) )
7                Begin
8                 nextLevel(c1);
9                 nextLevel(c2);
10               End
11       End
```

Listing 2: Procedure to find a path by means of a least common subsumer search.

The pseudo code in Listing 2 illustrates this procedure for example concepts ”concept1” and ”concept2”.

These procedures were implemented with Java. Java is not the most adequate technology for this type of tree search, since it lacks tail call elimination for security reasons, as further detailed in the Bugs section of Oracles website[3] but it was sufficiently effective to accomplish our goals.

3.2 Density

The general idea behind the proposed improvement to the measure can be introduced with some generic mathematics concepts. In graph theory, a dense graph is a graph in which the number of edges is close to the maximal possible between all the vertices contained.

The maximum density is 1 and the minimum is 0 for a very sparse graph (Coleman and Moré, 1983).

In our case, we do not intend to calculate the complete density of the Wikipédia category ontology. Instead, we propose to analyse all the nodes in the shortest path and find all the degrees. Then it is necessary to find a way of discover the approximate level of density with the degrees as a clue.

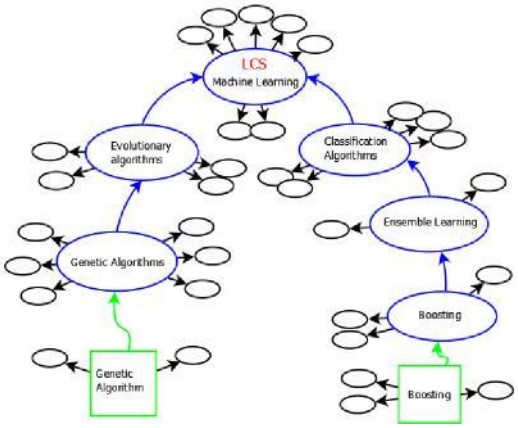

Figure 2: Example of an High Density path.

In the figure above, the larger circles represents Wikipédia categories in the path, the small ones represent the neighbors and the squares represents the Wikipédia pages. As it can be observed, Figure 2 represents a situation of higher density than Figure 3 and as we can conclude, the number of degrees for each node follows the tendency.

[3]See http://bugs.sun.com/view bug.do?bug id=4726340

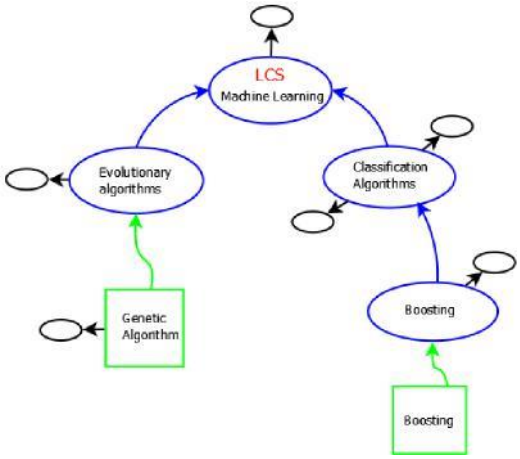

Figure 3: Example of a Low Density Path.

When path calculation occurs, the degrees of all vertices in the calculated path are saved, even the number of categories at the original pages.The objective is to save as much information as possible about the number neighbors of the calculated path. As it was previous explained, the properties of our entities were mapped to the correct page that represents the concept in Wikipédia. Figure 2 aims to find the distance between the property generic algorithm and the property boosting. These two concepts are contained in two different sets of properties, each one representing it own entity.

For every path explored between the entity properties the average number of neighbors is calculated. This means that information about the average number of nodes that are between the two entities is kept across the paths calculations, and based on that, allows to understand if the universe that we are working at the moment is more or less dense.

$$PathDensity(p1,p2) = \frac{\sum_1^n | E | (i)}{n} \qquad (4)$$

$$PropertyDensity = \frac{PathDensity(p1,p2)}{PathDensity(p1,p2)+RootDensity(lcs,root)} \qquad (5)$$

On top of that, as we saw before, the LCS node is very important to us. In a previous work it was concluded that it is very helpful to calculate how deep is this node (Medina et al., 2012). Now we propose to use additional information, by taking into account the density of the path between a given LCS and the root. The strategy is similar to the density calculation property, with the difference that now our path is between our LCS and the Wikipédia root category. We count the neighbors at each vertex and calculate the average at the end.

Table 1: Entities represented as sets of scientific topics. The first three entities represent actual conferences (CVPR, KDD and RECOMB respectively). The other three entities represent authors.

E1	E2	E3	E4	E5	E6
Computer Vision	Knowledge Discovery	Molecular Biology	Computer Vision	Information Extraction	Genetics
Object Recognition	Data Mining	Gene Expression	Robotics	Machine Learning	Human Genome
Structure from Motion	Web Mining	Computational Biology	Object Recognition	Natural Language Processing	Gene Expression
Image Segmentation	Recommender Systems	Genomics	Structure from Motion	Information Retrieval	Systems Biology
Image Processing	Cluster Analysis	Population Genetics	Human Computer Interaction	Data Mining	Clinical Medicine
Object categorization	Text Mining		Virtual Reality	Graphical Model	Bioinformatics
Optical Flow	Data Analytics		Facial Expression	Social Network	
Pattern Recognition	Structure Mining			Reinforcement Learning	
				Web Mining	

Table 2: Proposed similarity measure results.

Pair	(E_1,E_4)	(E_1,E_5)	(E_1,E_6)	(E_2,E_4)	(E_2,E_5)	(E_2,E_6)	(E_3,E_4)	(E_3,E_5)	(E_3,E_6)
S Eq.3	0.948	0.931	0.912	0.869	0.954	0.826	0.786	0.862	0.989
S Den	1.0	0.958	0.682	0.812	1.0	0.724	0.847	0.826	1.0

4 RESULTS AND DISCUSSION

With test result tracking in mind, we decided to use the same battery of entity tests from last paper (Medina et al., 2012). The results can be consulted on table 2, where the first row contains the values previously obtained on our first paper, and the second row contains the results achieved by the new method. The set is composed by six entities: three of them represent well known conferences (CVPR[4], KDD[5] and RECOMB[6]). These conferences were chosen because each one corresponds to a distinct scientific research area: CVPR to Computer Vision and Pattern Recognition; KDD to Data Mining and Knowledge Discovery; RECOMB to Computational Molecular Biology. The other three entities represent well known authors. Each of these authors was chosen based on the strong correspondence of their research interests with one of the three conferences:

- **Author** E_4**:** from computer vision area, which matches CVPR represented by E_1.
- **Author** E_5**:** from data mining and machine learning areas, which are more related to the KDD, represented by E_2.
- **Author** E_6**:** from genetics and bioinformatics areas, which is more closely related to RECOMB (represented by E_3).

Some concepts listed here do not have a direct correspondence with a Wikipédia page, so it lead us to a disambiguation problem. A quick solution for the first case was to replace the concept with a similar concept. For instance, the concept Image Segmentation of E1 had to be replaced with the page Segmentation (image processing). We will propose a technique to automation this kind of tasks in a future work.

From these results, as before, we observed a high value of similarity for the following entity pairs: (E_4,E_1), (E_5,E_2), and (E_6,E_3), but at this time, there is even stronger. This is expected due to the semantic overlapping of properties in the sets. It was also expected that the similarity values for (E_1,E_5) would be much lower than the value found for (E_1,E_4). It was observed that in many cases this distance continues to return very high similarity values that are not quite differentiated. We believe this is due to the proximity of a single pair of features, among many features. The fact that similarity between entities is determined by the maximum value of the similarity between all combinations of pairs of entities features introduces "bias" to 1. Hence it is needed, eventually, in the future to introduce mechanisms to extend the dynamic range of the similarity between entities. However, the high similarity E1-E5 is explained due to some very similar features, such as Pattern Recognition vs. Machine Learning, for example.

5 CONCLUSIONS AND FUTURE WORK

In this paper we continue our journey to find a more balanced semantic relatedness measure between entities. As before, we belive that the use of Wikipédia and the hierarchy of scientific categories contained in it, is the most promising way to accomplish your goal.

[4]http://www.cvpr2012.org/
[5]http://kdd2012.sigkdd.org/
[6]http://bioinfo.au.tsinghua.edu.cn/recomb2013/

The devised measure examines the Wikipédia category paths between all the possible concept pairs of two distinct entities, assigning weights according to the category's relevance. With this new attempt, we improve this measure by adding the notion of ontology density. We examined and compared new results with the old ones and concluded, by observing, that these matches are a step in the right direction. Although there is room for future developments, mainly regarding the range of result values, the differentiation of values and eventually the introduction of a threshold. The issue is to determine where to place the threshold to make the right decision. Usually the threshold is set "half-way", however, for this test case, it should be placed above 0.73, which is a high value.

Future work includes continuing exploration of the measure for other contexts as well as a comparison of our measure with other state-of-the-art metrics. It is also very important the effort to make the process as much autonomous as possible, by giving to the process the ability of automatic disambiguation.

REFERENCES

Coleman, T. F. and Moré, J. J. (1983). Estimation of sparse Jacobian matrices and graph coloring problems. *SIAM Journal on Numerical Analysis*, 20(1):187–209.

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipédia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 1606–1611. Morgan Kaufmann Publishers Inc.

Gouws, S., Rooyen, G., and Engelbrecht, H. (2010). Measuring conceptual similarity by spreading activation over wikipédia's hyperlink structure. In *Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*.

Gracia, J. and Mena, E. (2008). Web-based measure of semantic relatedness. In *In Proc. of 9th International Conference on Web Information Systems Engineering (WISE 2008), Auckland (New Zealand*, pages 136– 150. Springer.

Jiang, J. J. and Conrath, D. W. (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *International Conference Research on Computational Linguistics (ROCLING X)*.

Leacock, C. and Chodorow, M. (1998). *Combining Local Context and WordNet Similarity for Word Sense Identification*, chapter 11, pages 265–283. The MIT Press.

Liu, J. and Birnbaum, L. (2007). Measuring semantic similarity between named entities by searching the web directory. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, WI '07, pages 461–465.L

Medina, L. A. S., Fred, A. L. N., Rodrigues, R., and Filipe, J. (2012). Measuring entity semantic relatedness using wikipédia. In Fred, A. L. N., Filipe, J., Fred, A. L. N., and Filipe, J., editors, *KDIR*, pages 431–437. SciTePress.

Milne, D. and Witten, I. H. (2008). An effective, low-cost measure of semantic relatedness obtained from wikipédia links. In *In Proceedings ofAAAI 2008*.

Nastase, V. and Strube, M. (2008). Decoding wikipédia categories for knowledge acquisition. In *AAAI*, pages 1219–1224.

Ponzetto, S. P. and Strube, M. (2007). Knowledge derived from wikipédia for computing semantic relatedness. *J. Artif. Int. Res.*, 30:181–212.

Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19(1):17–30.

Resnik, P. (1999). Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*, 11:95–130.

Slimani, T., Yaghlane, B. B., and Mellouli, K. (2006). A New Similarity Measure based on Edge Counting. In *Proceedings of world academy of science, engineering and technology*, volume 17.

Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 133–138. Association for Computational Linguistics.

Zesch, T., Müller, C., and Gurevych, I. (2008). Extracting Lexical Semantic Knowledge from Wikipédia and Wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*.

Anexo II

Formato input

O ficheiro xml aqui anexado representa um ficheiro com a estrutura necessária para fornecer entidades para de cálculo de semelhança ao projeto. Neste caso particular, cinco entidades de origem em diferentes áreas científicas.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<entities>
	<entity name="E1">
		<topic>Computer! and Visions</topic>
		<topic>Object Recognition</topic>
		<topic>Structure from Motion teste teste</topic>
		<topic>Image Segmentation</topic>
		<topic>Image Processing</topic>
		<topic>Object Categorization from Image Search</topic>
		<topic>Optical Flow</topic>
		<topic>Pattern Recognition</topic>
	</entity>
	<entity name="E2">
		<topic>Knowledge extraction</topic>
		<topic>Data Mining</topic>
		<topic>Web Mining</topic>
		<topic>Recommender Systems</topic>
		<topic>Cluster Analysis</topic>
		<topic>Text Mining</topic>
		<topic>Data Analytics</topic>
		<topic>Structure Mining</topic>
	</entity>
	<entity name="E3">
		<topic>Molecular Biology</topic>
		<topic>Gene Expression</topic>
		<topic>Genomics</topic>
		<topic>Population Genetics</topic>
	</entity>
	<entity name="E4">
		<topic>car</topic>
		<topic>engine</topic>
		<topic>rpm</topic>
		<topic>Matrix</topic>
	</entity>
	<entity name="E5">
		<topic>brad pitt</topic>
		<topic>hollywood</topic>
		<topic>scientific fiction</topic>
		<topic>Matrix</topic>
	</entity>
</entities>
```

Anexo III

Propriedades de Eventos

Neste anexo são apresentados os tópicos associados às respetivas áreas de conhecimento, para cada um dos eventos estudados.

```
<entities>
	<entity name="MODELSWARD">
		<topic>Domain-specific Modeling</topic>
		<topic>General-purpose Modeling Languages and Standards</topic>
		<topic>Model Transformation</topic>
		<topic>Syntax and Semantics of Modeling Languages</topic>
		<topic>Meta-modeling: Foundations and Tools</topic>
		<topic>Reasoning about Models</topic>
		<topic>Constraint Modeling and Languages</topic>
		<topic>Model-Driven Architecture</topic>
		<topic>Service Oriented Architectures</topic>
		<topic>Systems Engineering</topic>
		<topic>Model Transformations and Generative Approaches</topic>
		<topic>Frameworks for Model-Driven Development</topic>
		<topic>Hybrid Multi-Modeling Approaches</topic>
		<topic>Modeling for the Cloud</topic>
		<topic>Software Process Modeling, Enactment and Execution</topic>
		<topic>Workflow Management Systems</topic>
		<topic>Business Process Modeling</topic>
		<topic>Agile Model-Driven Development</topic>
		<topic>Model-Driven Project Management</topic>
		<topic>Model-based Testing and Validation</topic>
		<topic>Model Execution and Simulation</topic>
		<topic>Model Quality Assurance Techniques</topic>
		<topic>Executable UML</topic>
		<topic>Meta-Programming</topic>
		<topic>Component-based software engineering</topic>
		<topic>Software Factories and Software Product Lines</topic>
		<topic>Generative Programming</topic>
	</entity>
	<entity name="ICSPORTS">
		<topic>Computer Supported Training and Decision Support Systems</topic>
		<topic>Multimedia and Information Technology</topic>
		<topic>Simulation and Mathematical Modeling</topic>
		<topic>Sport Statistics and Analyses </topic>
		<topic>Applied Physiology and Exercise</topic>
		<topic>Coaching </topic>
		<topic>Health and Fitness</topic>
		<topic>Sports Psychology</topic>
		<topic>Training and Testing </topic>
		<topic>Biosignals and Biodevices Engineering</topic>
		<topic>Gait and Posture</topic>
		<topic>Motor Control and Coordination</topic>
		<topic>Motor Learning</topic>
		<topic>Neuromuscular Physiology</topic>
		<topic>Sports Biomechanics</topic>
		<topic>Injury Prevention</topic>
```

```
        <topic>Physical Education and Sports for Handicapped</topic>
        <topic>Physiotherapy and Rehabilitation</topic>
</entity>
<entity name="ICPRAM">
        <topic>Exact and Approximate Inference</topic>
        <topic>Density Estimation</topic>
        <topic>Bayesian Models</topic>
        <topic>Gaussian Processes</topic>
        <topic>Model Selection</topic>
        <topic>Graphical and Graph-based Models</topic>
        <topic>Missing Data</topic>
        <topic>Ensemble Methods</topic>
        <topic>Neural Networks</topic>
        <topic>Kernel Methods</topic>
        <topic>Large Margin Methods</topic>
        <topic>Classification</topic>
        <topic>Regression</topic>
        <topic>Sparsity</topic>
        <topic>Feature Selection and Extraction</topic>
        <topic>Spectral Methods</topic>
        <topic>Embedding and Manifold Learning</topic>
        <topic>Similarity and Distance Learning</topic>
        <topic>Matrix Factorization</topic>
        <topic>Clustering</topic>
        <topic>ICA, PCA, CCA and other Linear Models</topic>
        <topic>Fuzzy Logic</topic>
        <topic>Active Learning</topic>
        <topic>Cost-sensitive Learning</topic>
        <topic>Incremental Learning</topic>
        <topic>On-line Learning</topic>
        <topic>Structured Learning</topic>
        <topic>Multi-agent Learning</topic>
        <topic>Multi-instance Learning</topic>
        <topic>Reinforcement Learning</topic>
        <topic>Instance-based Learning</topic>
        <topic>Knowledge Acquisition and Representation</topic>
        <topic>Meta Learning</topic>
        <topic>Multi-strategy Learning</topic>
        <topic>Case-Based Reasoning</topic>
        <topic>Inductive Learning</topic>
        <topic>Computational Learning Theory</topic>
        <topic>Cooperative Learning</topic>
        <topic>Evolutionary Computation</topic>
        <topic>Information Retrieval and Learning</topic>
        <topic>Hybrid Learning Algorithms</topic>
        <topic>Planning and Learning</topic>
        <topic>Convex Optimization</topic>
        <topic>Stochastic Methods</topic>
        <topic>Combinatorial Optimization</topic>
        <topic>Multiclassifier Fusion</topic>
        <topic>Natural Language Processing</topic>
        <topic>Information Retrieval</topic>
        <topic>Ranking</topic>
        <topic>Web Applications</topic>
        <topic>Economics, Business and Forecasting Applications</topic>
        <topic>Bioinformatics and Systems Biology</topic>
        <topic>Audio and Speech Processing</topic>
```

```
        <topic>Signal Processing</topic>
        <topic>Image Understanding</topic>
        <topic>Sensors and Early Vision</topic>
        <topic>Motion and Tracking</topic>
        <topic>Image-based Modelling</topic>
        <topic>Shape Representation</topic>
        <topic>Object Recognition</topic>
        <topic>Video Analysis</topic>
        <topic>Medical Imaging</topic>
        <topic>Learning and Adaptive Control</topic>
        <topic>Perception</topic>
        <topic>Learning in Process Automation</topic>
        <topic>Learning of Action Patterns</topic>
        <topic>Virtual Environments</topic>
        <topic>Robotics</topic>
        <topic>Biometrics</topic>
</entity>
<entity name="PHOTOPTICS">
        <topic>Adaptative Optics</topic>
        <topic>Optometry</topic>
        <topic>Fiber Optics Technology</topic>
        <topic>Biomedical Optics</topic>
        <topic>Computational Optical Sensing and Imaging</topic>
        <topic>Optical Instrumentation</topic>
        <topic>Optics in Astronomy and Astrophysics</topic>
        <topic>Spectroscopy, Imaging and Metrology</topic>
        <topic>Optical Communications and Networking</topic>
        <topic>Vision and Color</topic>
        <topic>Photorefractive Effects, Materials and Devices</topic>
        <topic>Optical Materials and Devices</topic>
        <topic>Light-matter Interaction</topic>
        <topic>Opto-Mechatronics</topic>
        <topic>Applied Industrial Optics</topic>
        <topic>Display Technology and Holography</topic>
        <topic>Organic and Bio-photonics</topic>
        <topic>Optogenetics</topic>
        <topic>MOEMS-MEMS and Nanophotonics</topic>
        <topic>Microwave Photonics</topic>
        <topic>Photonics for Energy</topic>
        <topic>Power Photonics</topic>
        <topic>Photonic and Optoelectronic Materials and Devices</topic>
        <topic>Communications and Switching Photonics</topic>
        <topic>Ultrafast Electronics, Photonics and Optoelectronics</topic>
        <topic>Green Photonics</topic>
        <topic>Photodetectors, Sensors and Imaging</topic>
        <topic>Fiber Optics devices</topic>
        <topic>Nonlinear Optics</topic>
        <topic>Plasmonic Structures and Quantum dots</topic>
        <topic>Metamaterials</topic>
        <topic>High Intensity Lasers and High Field Phenomena</topic>
        <topic>Quantum Electronics and Laser Science</topic>
        <topic>Quantum Information and Measurement</topic>
        <topic>Biomedical and Therapeutic Laser Applications</topic>
        <topic>Laser Microscopy</topic>
        <topic>Laser Spectroscopy</topic>
        <topic>Fiber Lasers and Applications</topic>
        <topic>Waveguide Lasers</topic>
```

```
            <topic>Semiconductor Lasers and LEDs</topic>
            <topic>Weapons and Military Technology</topic>
            <topic>Plasma Technologies</topic>
        </entity>
</entities>
```

Anexo IV

Propriedades de revisores

Neste anexo são apresentados os tópicos associados às respetivas áreas de conhecimento,

```
<entities>
	<entity name="MODELSWARD_1 - Cesar Gonzalez-Perez">
		<topic>Domain-specific Modeling</topic>
		<topic>General-purpose Modeling Languages and Standards</topic>
		<topic>Syntax and Semantics of Modeling Languages</topic>
		<topic>Meta-modeling: Foundations and Tools</topic>
		<topic>Reasoning about Models</topic>
		<topic>Software Process Modeling, Enactment and
	Execution</topic>
	</entity>
	<entity name="MODELSWARD_2 - Cinzia Cappiello">
		<topic>Workflow Management Systems</topic>
		<topic>Business Process Modeling</topic>
		<topic>Service Oriented Architectures</topic>
		<topic>Modeling for the Cloud</topic>
		<topic>Model Quality Assurance Techniques</topic>
	</entity>
	<entity name="MODELSWARD_3 - Damiano Distante">
		<topic>Business Process Modeling</topic>
		<topic>Model Transformation</topic>
		<topic>Model-Driven Architecture</topic>
		<topic>Frameworks for Model-Driven Development</topic>
		<topic>Software Process Modeling, Enactment and
	Execution</topic>
		<topic>Model-Driven Project Management</topic>
	</entity>
	<entity name="MODELSWARD_4 - Fergal Mc Caffery">
		<topic>Business Process Modeling</topic>
		<topic>Systems Engineering</topic>
		<topic>Software Process Modeling, Enactment and
	Execution</topic>
		<topic>Agile Model-Driven Development</topic>
	</entity>
	<entity name="MODELSWARD_5 - Guglielmo de Angelis">
		<topic>Business Process Modeling</topic>
		<topic>Model Transformation</topic>
		<topic>Model-Driven Architecture</topic>
		<topic>Service Oriented Architectures</topic>
		<topic>Model-based Testing and Validation</topic>
		<topic>Generative Programming</topic>
	</entity>
	<entity name="ICSPORTS_1 - Abdülkerim Kasim Baltaci">
		<topic>Sports Biomechanics</topic>
		<topic>Applied Physiology and Exercise</topic>
		<topic>Neuromuscular Physiology</topic>
		<topic>Health and Fitness</topic>
	</entity>
	<entity name="ICSPORTS_2 - Amir Ali Mohagheghi">
		<topic>Gait and Posture</topic>
```

```
        <topic>Neuromuscular Physiology</topic>
        <topic>Motor Control and Coordination</topic>
</entity>
<entity name="ICSPORTS_3 - Andrey  Koptyug">
        <topic>Sports Biomechanics</topic>
        <topic>Simulation and Mathematical Modeling</topic>
        <topic>Physical Education and Sports for Handicapped</topic>
        <topic>Injury Prevention</topic>
</entity>
<entity name="ICSPORTS_4 - Anthony Leicht">
        <topic>Applied Physiology and Exercise</topic>
        <topic>Health and Fitness</topic>
        <topic>Training and Testing</topic>
        <topic>Sport Statistics and Analyses </topic>
</entity>
<entity name="ICSPORTS_5 - Jan  Cabri">
        <topic>Sports Biomechanics</topic>
        <topic>Applied Physiology and Exercise</topic>
        <topic>Neuromuscular Physiology</topic>
        <topic>Training and Testing</topic>
</entity>
<entity name="ICPRAM_1 - Daniel  Boullosa ">
        <topic>Information retrieval</topic>
        <topic>Feature Selection and Extraction</topic>
        <topic>Spectral Methods</topic>
        <topic>Clustering</topic>
        <topic>Ranking</topic>
        <topic>Web Applications</topic>
</entity>
<entity name="ICPRAM_2 - Bjoern Schuller">
        <topic>Motion and Tracking</topic>
        <topic>Clustering</topic>
        <topic>Signal Processing</topic>
</entity>
<entity name="ICPRAM_3 - Elena Marchiori">
        <topic>Matrix Factorization</topic>
        <topic>Clustering</topic>
        <topic>Instance-based Learning</topic>
        <topic>Evolutionary Computation</topic>
</entity>
<entity name="ICPRAM_4 - Fabio Gonzalez">
        <topic>Information retrieval</topic>
        <topic>Matrix Factorization</topic>
        <topic>On-line Learning</topic>
        <topic>Information Retrieval and Learning</topic>
</entity>
<entity name="ICPRAM_5 - Francisco Martínez Álvarez">
        <topic>Neural Networks</topic>
        <topic>Ensemble Methods</topic>
        <topic>Clustering</topic>
        <topic>Evolutionary Computation</topic>
        <topic>Signal Processing</topic>
</entity>
<entity name="PHOTOPTICS_1 -  Javier PELAYO ZUECO">
        <topic>Fiber Optics</topic>
        <topic>Optical Instrumentation</topic>
        <topic>Spectroscopy, Imaging and Metrology</topic>
```

```
                <topic>Optical Communications and Networking</topic>
                <topic>Photonic in Communications and Switching</topic>
                <topic>Fiber Lasers and Applications</topic>
                <topic>Nonlinear Optics</topic>
        </entity>
        <entity name="PHOTOPTICS_2 - Alexander  Argyros">
<topic>Fiber Optics</topic>
                <topic>Biomedical Optics</topic>
                <topic>Optical Instrumentation</topic>
                <topic>Optical Communications and Networking</topic>
        </entity>
        <entity name="PHOTOPTICS_3 - André  Nicolet">
                <topic>Fiber Optics</topic>
                <topic>MEMS and Nanophotonics</topic>
                <topic>Microwave Photonics</topic>
                <topic>Quantum Information and Measurement</topic>
        </entity>
        <entity name="PHOTOPTICS_4 - Carlos Saavedra">
                <topic>Fiber Optics</topic>
                <topic>Biomedical Optics</topic>
                <topic>Optical Instrumentation</topic>
                <topic>Optical Communications and Networking</topic>
                <topic>Photonic and Optoelectronic Materials and Devices</topic>
                <topic>Photonic in Communications and Switching</topic>
                <topic>Quantum Information and Measurement</topic>
        </entity>
        <entity name="PHOTOPTICS_5 - David Plant">
                <topic>Fiber Optics</topic>
                <topic>Optical Communications and Networking</topic>
        </entity>
</entities>
```

Anexo V

Propriedades de Artigos

Neste anexo são apresentados os tópicos associados às respetivas áreas de conhecimento, para cada um dos artigos estudados.

```
<entity name="MODELSWARD_1 - Variability Management supporting the Model-Driven Design of User Interfaces">
		<topic>Model Transformations and Generative Approaches</topic>
		<topic>Software Factories and Software Product Lines</topic>
	</entity>
	<entity name="MODELSWARD_2 - Using Software Categories for the Development of Generative Software">
		<topic>Model Transformations and Generative Approaches</topic>
		<topic>Agile Model-Driven Development</topic>
		<topic>Generative Programming</topic>
	</entity>
	<entity name="MODELSWARD_3 - Characterizing Workload Pattern">
		<topic>Software Process Modeling, Enactment and Execution</topic>
		<topic>Model-based Testing and Validation</topic>
	</entity>
	<entity name="MODELSWARD_4 - A Modular Method for Global System Behaviour Specification">
		<topic>Business Process Modeling</topic>
		<topic>Systems Engineering</topic>
		<topic>Frameworks for Model-Driven Development</topic>
	</entity>
	<entity name="MODELSWARD_5 - Towards Product Lining Model-Driven Development Code Generators">
		<topic>Component-based software engineering</topic>
		<topic>Software Factories and Software Product Lines</topic>
		<topic>Generative Programming</topic>
	</entity>
	<entity name="ICSPORTS_1 - Low Aerodynamic Drag Suit for Cycling - Design and Testing">
		<topic>Sports Biomechanics</topic>
		<topic>Training and Testing</topic>
	</entity>
	<entity name="ICSPORTS_2 - AT Training Zones for Different Pace Plans of 200 M Swimmers">
		<topic>Applied Physiology and Exercise</topic>
		<topic>Training and Testing</topic>
		<topic>Coaching</topic>
	</entity>
	<entity name="ICSPORTS_3 - Links Between Mood States, Time Of Practise And Game Results In Brazilian Voleibol Players ">
		<topic>Sports Psychology</topic>
		<topic>Coaching</topic>
	</entity>
	<entity name="ICSPORTS_4 - Flexible studs in prevention of football injuries: A preliminary laboratory study: A preliminary laboratory study">
		<topic>Sports Biomechanics</topic>
		<topic>Injury Prevention</topic>
```

```
        </entity>
        <entity name="ICSPORTS_5 - Relationship Between Initiation of Gaze
Stabilization and Angle of Head and Trunk Movement During a Jump with Full Turn">
                <topic>Sports Psychology</topic>
                <topic>Coaching</topic>
        </entity>
        <entity name="ICPRAM_1 - Cogn-Evo - Document Clustering Based on a Cognitive-
Evolutionary System">
                <topic>Information retrieval</topic>
                <topic>Feature Selection and Extraction</topic>
                <topic>Clustering</topic>
                <topic>Evolutionary Computation</topic>
        </entity>
        <entity name="ICPRAM_2 - HyperSAX: Fast Approximate Search of Multidimensional
Data">
                <topic>Information retrieval</topic>
                <topic>Clustering</topic>
        </entity>
        <entity name="ICPRAM_3 Automatic Image Annotation using Convex Deep Learning
Models">
                <topic>Neural Networks</topic>
                <topic>Kernel Methods</topic>
                <topic>Classification</topic>
                <topic>Object Recognition</topic>
        </entity>
        <entity name="ICPRAM_4 -  ZiZo: Arabic Digit Recognition by Adaptive Network
Based Fuzzy Inference System">
                <topic>Neural Networks</topic>
                <topic>Classification</topic>
                <topic>Clustering</topic>
                <topic>Fuzzy Logic</topic>
        </entity>
        <entity name="ICPRAM_5 - Binary Quantization for Selecting Central Points of
Dimension Reduction Projection Simple-Map">
                <topic>Information retrieval</topic>
                <topic>Feature Selection and Extraction</topic>
                <topic>Information Retrieval and Learning</topic>
        </entity>
        <entity name="PHOTOPTICS_1 - Automatic Waveguide-fiber Alignment Algorithm
based on Coupling Model">
                <topic>Photonic and Optoelectronic Materials and Devices</topic>
                <topic>Optoelectronics</topic>
        </entity>
        <entity name="PHOTOPTICS_2 - Mechanical Characterisation of the Four Most Used
Coating Materials for Optical Fibres">
                <topic>Fiber Optics</topic>
                <topic>Optical Instrumentation</topic>
                <topic>Optics in Astronomy and Astrophysics</topic>
                <topic>Optical Materials</topic>
        </entity>
        <entity name="PHOTOPTICS_3 - Bragg grating solitons in semilinear dual-core
system with cubic-quintic nonlinearity">
                <topic>Fiber Optics</topic>
                <topic>Photonic and Optoelectronic Materials and Devices</topic>
                <topic>Photonic in Communications and Switching</topic>
                <topic>Ultrafast Electronics, Photonics and Optoelectronics</topic>
                <topic>Nonlinear Optics</topic>
```

```
        </entity>
        <entity name="PHOTOPTICS_4 - Design of an optimized distal optic for non
linear endomicroscopy">
                <topic>Fiber Optics</topic>
                <topic>Biomedical Optics</topic>
                <topic>Optical Instrumentation</topic>
                <topic>Nonlinear Optics</topic>
        </entity>
        <entity name="PHOTOPTICS_5 - The Sensors Are Innovative in Internet of
Things">
                <topic>Quantum Electronics and Laser Science</topic>
                <topic>Optical Materials</topic>
        </entity>
```