

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

PEDRO BISPO DA SILVA SANTOS

**Aperfeiçoamento de Métricas de  
Similaridade Semântica entre Serviços Web**

Dissertação apresentada como requisito parcial  
para a obtenção do grau de  
Mestre em Ciência da Computação

Prof. Dr. Leandro Krug Wives  
Orientador

Prof. Dr. José Palazzo Moreira de Oliveira  
Co-orientador

Porto Alegre, agosto de 2012

## CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Santos, Pedro Bispo da Silva

Aperfeiçoamento de Métricas de Similaridade Semântica entre Serviços Web / Pedro Bispo da Silva Santos. – Porto Alegre: PPGC da UFRGS, 2012.

63 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2012. Orientador: Leandro Krug Wives; Coorientador: José Palazzo Moreira de Oliveira.

1. Serviços web. 2. Algoritmos de similaridade. 3. Web semântica. I. Wives, Leandro Krug. II. Oliveira, José Palazzo Moreira de. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Pró-Reitor de Coordenação Acadêmica: Prof. Rui Vicente Oppermann

Pró-Reitora de Pós-Graduação: Prof. Aldo Bolten Lucion

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do PPGC: Prof. Álvaro Freitas Moreira

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*KISS*  
— KEEP IT SIMPLE, SIR

## **AGRADECIMENTOS**

Gostaria de agradecer primeiramente aos meus pais, por todo o apoio, ajuda, amor, carinho e inspiração. Dois exemplos de vida e motivos de muito orgulho desse filho ingrato (hahahahaha) que resolveu continuar seus estudos em uma terra tão distante de Maceió, quando teve opções de fazer o mestrado em outras instituições mais próximas. Obrigado por tudo, vocês são minha fonte de inspiração.

Também quero agradecer meus orientadores, os professores doutores Leandro Krug Wives e José Palazzo Moreira de Oliveira, pela orientação, paciência, revisões e apoios dados durante o desenvolvimento desse trabalho. Também agradeço por terem me aceitado como orientando e por sempre se mostrarem prestativos e interessados no meu trabalho.

Agradeço à minha namorada, Amanda Schneider, pela paciência, amor e carinho que recebi durante esses 5 meses de namoro. Obrigado meu amor por estar ao meu lado nessa reta final e por ter sido uma companheira tão boa e compreensível. Eu te amo muito.

Quero agradecer aos meus amigos distantes que através de trocas de emails e ligações telefônicas, fazem com que a amizade se mantenha forte, apesar da distância de 3572 km entre Maceió e Porto Alegre, valeu por me fazer dar boas risadas e pela companhia a distância: Baltazar, Chefe, Deko, Eliel, Fábio, Fuinha, Mago e Marcelo.

Aos companheiros da UFRGS, camaradas gaudérios e uma imigrante carioca (hahahaha), obrigado pelos vários chimas e pelas ajudas durante esse tempo que estive trabalhando no laboratório: Alencar, Alessandro, Ana, Daniel, Fahad, Giseli, Jonas, Leila, Lucinéia.

E agradeço ao CNPq pela ajuda financeira concedida através da bolsa de mestrado.

# SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS</b> . . . . .	7
<b>LISTA DE FIGURAS</b> . . . . .	8
<b>LISTA DE QUADROS</b> . . . . .	9
<b>RESUMO</b> . . . . .	10
<b>ABSTRACT</b> . . . . .	11
<b>1 INTRODUÇÃO</b> . . . . .	12
<b>1.1 Objetivo</b> . . . . .	13
1.1.1 Objetivos Secundários . . . . .	13
<b>1.2 Hipótese</b> . . . . .	13
<b>1.3 Contribuição do Trabalho</b> . . . . .	14
<b>1.4 Organização do Texto</b> . . . . .	14
<b>2 SERVIÇOS WEB</b> . . . . .	16
<b>2.1 WSDL</b> . . . . .	16
<b>2.2 UDDI</b> . . . . .	18
<b>2.3 Orquestração e Coreografia</b> . . . . .	21
<b>2.4 Web Semântica</b> . . . . .	22
2.4.1 Ontologia . . . . .	22
2.4.2 SWRL . . . . .	23
<b>2.5 Serviços Web Semânticos</b> . . . . .	23
2.5.1 OWL-S . . . . .	25
<b>3 ESTADO DA ARTE</b> . . . . .	27
<b>3.1 Métricas de Similaridade entre Serviços Web Tradicionais</b> . . . . .	27
3.1.1 WSCOLAB . . . . .	27
3.1.2 Similaridade baseada em assinaturas . . . . .	30
3.1.3 Análise das Abordagens . . . . .	31
<b>3.2 Métricas de Similaridade entre Serviços Web Semânticos</b> . . . . .	31
3.2.1 Similaridade através de extração de conhecimento . . . . .	32
3.2.2 OWLS-MX . . . . .	33
3.2.3 Filtragem de Registros de Serviços Web Semânticos . . . . .	34
3.2.4 Similaridade baseada em ontologia . . . . .	35
3.2.5 Análise das Abordagens . . . . .	36
<b>3.3 Resumo do Capítulo</b> . . . . .	36

<b>4</b>	<b>TRABALHO DESENVOLVIDO</b>	37
4.1	Similaridade entre Serviços Web Semânticos	39
4.1.1	Limiar	39
4.2	Limitações do Algoritmo de Liu	39
4.3	Hipótese Levantada	40
4.4	Aperfeiçoamento do Algoritmo	41
4.5	Definição de Densidade Semântica	41
4.6	Resumo do Capítulo	43
<b>5</b>	<b>EXPERIMENTOS</b>	44
5.1	ANOVA	47
5.2	Primeira Configuração, $\alpha = 0,2, \beta = 0,6$	48
5.3	Segunda Configuração, $\alpha = 0,25$	49
5.4	Terceira Configuração, $\alpha = 0,2, \beta = 0,6, \lambda = 0,4$	51
5.5	Resumo do Capítulo	54
<b>6</b>	<b>CONCLUSÕES</b>	59
6.1	Publicações Resultantes	59
6.2	Trabalhos Futuros	60
	<b>REFERÊNCIAS</b>	61

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
ERP	Enterprise Resourcing Planning
HTTP	HyperText Transfer Protocol
OWL	Ontology Web Language
OWL-S	Ontology Web Language for Services
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
UBR	Universal Business Registry
UDDI	Universal Description Discovery and Integration
URL	Uniform Resource Locator
UUID	Universally Unique Identifier
WSDL	Web Service Description Language
WSMO	Web Service Modeling Ontology
XML	eXtensible Markup Language
XML-RPC	eXtensible Markup Language - Remote Procedure Call
XSD	XML Schema Definition

## LISTA DE FIGURAS

Figura 1.1:	Pareamento entre conceitos de duas categorias . . . . .	14
Figura 2.1:	Modelo de uso de UDDI (PAPAZOGLU, 2008) . . . . .	19
Figura 2.2:	Orquestração de uma Ordem de Compra (OC) da perspectiva do fabricante (PAPAZOGLU, 2008) . . . . .	22
Figura 2.3:	Coreografia de uma Ordem de Compra (OC) (PAPAZOGLU, 2008)	22
Figura 2.4:	Arquitetura em camadas da Web Semântica atualizada (KIFER et al., 2005) . . . . .	23
Figura 2.5:	Exemplo de Ontologia (LIU et al., 2009) . . . . .	24
Figura 2.6:	OWL-S (MARTIN et al., 2004) . . . . .	25
Figura 2.7:	<i>ServiceProfile</i> (MARTIN et al., 2004) . . . . .	26
Figura 3.1:	Nuvem de tags para o comportamento do serviço . . . . .	27
Figura 3.2:	Nuvem de tags para as entradas do serviço . . . . .	28
Figura 3.3:	Nuvem de tags para as saídas do serviço . . . . .	28
Figura 3.4:	Exemplo de uma nuvem de tags . . . . .	28
Figura 3.5:	Comparando serviços Web como grafos bipartidos . . . . .	31
Figura 3.6:	Grafo de ligações semânticas . . . . .	32
Figura 3.7:	Extração das ligações semânticas (WEI et al., 2008) . . . . .	33
Figura 4.1:	Base semântica hierárquica de palavras . . . . .	37
Figura 4.2:	Pareamento entre conceitos de duas categorias . . . . .	40
Figura 4.3:	Pareamento de conceitos . . . . .	41
Figura 4.4:	Taxonomia de uma Ontologia . . . . .	42
Figura 5.1:	Gráfico da precisão para $\alpha = 0, 2\beta = 0, 6$ . . . . .	48
Figura 5.2:	Gráfico da revocação para $\alpha = 0, 2\beta = 0, 6$ . . . . .	50
Figura 5.3:	Gráfico da <i>f-measure</i> para $\alpha = 0, 2\beta = 0, 6$ . . . . .	50
Figura 5.4:	Função Exponencial, com $\alpha = 0, 2$ . . . . .	51
Figura 5.5:	Função monotônica, com $\beta = 0, 6$ . . . . .	51
Figura 5.6:	Gráfico da precisão para $\alpha = 0, 25$ . . . . .	53
Figura 5.7:	Gráfico da revocação para $\alpha = 0, 25$ . . . . .	53
Figura 5.8:	Gráfico da <i>f-measure</i> para $\alpha = 0, 25$ . . . . .	54
Figura 5.9:	Função Exponencial, com $\alpha = 0, 25$ . . . . .	54
Figura 5.10:	Gráfico da precisão para $\alpha = 0, 2\beta = 0, 6, \lambda = 0, 4$ . . . . .	56
Figura 5.11:	Gráfico da revocação para $\alpha = 0, 2\beta = 0, 6, \lambda = 0, 4$ . . . . .	56
Figura 5.12:	Gráfico da <i>f-measure</i> para $\alpha = 0, 2\beta = 0, 6, \lambda = 0, 4$ . . . . .	57
Figura 5.13:	Função Monotônica, com $\lambda = 0, 4$ . . . . .	57

## LISTA DE QUADROS

Quadro 3.1: Regras de decomposição de nomes . . . . .	30
Quadro 4.1: Lista de serviços Web semânticos Indexados . . . . .	43
Quadro 5.1: Consultas . . . . .	46
Quadro 5.2: Experimentos - $\alpha = 0,2, \beta = 0,6$ . . . . .	49
Quadro 5.3: Experimentos - ANOVA - Precisão, $\alpha = 0,2, \beta = 0,6$ . . . . .	49
Quadro 5.4: Experimentos - ANOVA - Revocação, $\alpha = 0,2, \beta = 0,6$ . . . . .	49
Quadro 5.5: Experimentos - ANOVA - <i>F-Measure</i> , $\alpha = 0,2, \beta = 0,6$ . . . . .	50
Quadro 5.6: Experimentos - $\alpha = 0,25$ . . . . .	52
Quadro 5.7: Experimentos - ANOVA - Precisão, $\alpha = 0,25$ . . . . .	52
Quadro 5.8: Experimentos - ANOVA - Revocação, $\alpha = 0,25$ . . . . .	52
Quadro 5.9: Experimentos - ANOVA - <i>F-measure</i> , $\alpha = 0,25$ . . . . .	53
Quadro 5.10: Experimentos - $\alpha = 0,2, \beta = 0,6, \lambda = 0,4$ . . . . .	55
Quadro 5.11: Experimentos - ANOVA - Precisão, $\alpha = 0,25, \beta = 0,6, \lambda = 0,4$ . . . . .	55
Quadro 5.12: Experimentos - ANOVA - Revocação, $\alpha = 0,25, \beta = 0,6, \lambda = 0,4$ . . . . .	55
Quadro 5.13: Experimentos - ANOVA - <i>f-measure</i> , $\alpha = 0,25, \beta = 0,6, \lambda = 0,4$ . . . . .	56
Quadro 5.14: Experimentos - ANOVA - Comparação da precisão entre a primeira e terceira configuração . . . . .	57
Quadro 5.15: Experimentos - ANOVA - Comparação da revocação entre a primeira e terceira configuração . . . . .	57
Quadro 5.16: Experimentos - ANOVA - Comparação da <i>F-measure</i> entre a primeira e terceira configuração . . . . .	58

## RESUMO

O presente trabalho apresenta uma abordagem aperfeiçoada para medir similaridade entre Serviços Web Semânticos através de um algoritmo baseado em ontologia. O trabalho se trata de um aperfeiçoamento por pegar como base o algoritmo de Liu (LIU et al., 2009) e nele fazer uma pequena mudança que resulta em melhorias significativas em certos cenários. Este algoritmo utiliza informação das categorias ESPE (Entradas, Saídas, Pré-condições, Efeitos), procurando por informações presentes nessas categorias, analisando como os conceitos estão relacionados na taxonomia da ontologia.

A escolha do trabalho de Liu ocorreu devido a experimentos realizados em trabalhos anteriores ao desta dissertação (MAAMAR et al., 2011)(MAAMAR et al., 2011). Nesses experimentos, foi constatado que o algoritmo não atingia um desempenho satisfatório, foi então que se levantou uma hipótese de aperfeiçoamento deste algoritmo.

Experimentos realizados utilizando um conjunto de dados contendo 1083 Serviços Web Semânticos em OWL-S mostram que essa abordagem aperfeiçoada aumenta a precisão do algoritmo, diminuindo desta forma o número de falsos positivos recuperados nos resultados, mantendo uma boa revocação. Nestes experimentos, foram implementados o algoritmo original de Liu e a versão modificada apresentada nesta dissertação. Além disso, este trabalho apresenta quais parâmetros utilizar para configurar o algoritmo e atingir melhor precisão, revocação e *f-measure*.

**Palavras-chave:** Serviços web, algoritmos de similaridade, web semântica.

## Enhancement of Semantic Web Services Matchmaking Metrics

### ABSTRACT

The current work presents an improved approach for an ontology-based semantic web service matchmaking assessment algorithm. This work is based on Liu's algorithm LIU et al. (2009) providing a small change which results in a significant enhancement at some scenarios. The algorithm uses information from IOPE (Inputs, Outputs, Preconditions, Effects) categories, searching for information about the concepts located in these categories, analyzing how they are related in an ontology taxonomy.

Liu's work was chosen due to experiments that were done in works previous to this dissertation (MAAMAR et al., 2011)(MAAMAR et al., 2011). During those experiments, it was realized that Liu's algorithm did not achieve good results. Then, an hypothesis was raised for improving this algorithm.

Experiments performed using a data set containing 1083 OWL-S semantic web services show that the improved approach increases the algorithm precision, diminishing the number of false positives in the retrieved results, and still having a good recall. For doing these experiments, Liu's algorithm and this dissertation version of it were implemented. Furthermore, this work presents the parameters that were used to achieve better precision, recall and f-measure.

**Keywords:** web services, matchmaking algorithms, semantic web.

# 1 INTRODUÇÃO

Com o crescente número de tecnologias, a heterogeneidade de plataformas é um fato consolidado nas empresas de TI. Isso gera um problema quando se deseja realizar a integração entre esses sistemas devido as diferenças encontradas em suas implementações. Logo a interoperabilidade entre aplicações se torna um desafio constante no dia a dia de muitas empresas.

Serviços Web provêm uma solução interessante para interoperabilidade de aplicações de software devido aos seus padrões baseados em XML: SOAP para troca de mensagens (GUDGIN et al., 2003), WSDL para descrição da interface dos serviços web (CHRISTENSEN et al., 2001), e UDDI para o registro de serviços web e armazenamento de informações acerca dos provedores de serviço.

No contexto de serviços Web e interoperabilidade, uma atividade importante é a descoberta de serviços, para que eles possam ser selecionados, utilizados e integrados. Apesar de ser um padrão, UDDI possui vários problemas. Um deles é o fato de não prover um método sofisticado para realização de consultas sobre seus registros. As consultas geralmente consistem de palavras-chaves simples. Tais consultas requerem algum conhecimento prévio acerca dos registros, como *business entity key* (chave de entidade de registro) ou *business entity name* (nome de entidade de registro).

Outro problema é o ranqueamento dos resultados, já que UDDI não estabelece um ranking para os registros recuperados. Isso pode ser um enorme problema em registros públicos, dado o crescimento constante da *Service Web*, que é o nome dado ao conjunto de serviços web disponíveis na Web (YU et al., 2008). Logo, recuperar serviços web utilizando UDDI não é uma opção ótima, dado que os registros recuperados não são rankeados.

O problema de descoberta é principalmente relacionado à falta de expressividade oferecida pela WSDL. WSDL é uma linguagem de descrição inteiramente sintática que define uma interface de um serviço Web através da listagem de suas operações. Nesta listagem, encontram-se tipos de dados e tipos definidos pelo usuário presentes nas entradas e saídas das operações, além das informações de acesso.

Trabalhos apontam que as técnicas de descoberta baseadas apenas na sintaxe apresentam resultados com baixa precisão e alta revocação (KRITIKOS; PLEXOUSAKIS, 2006; LIU et al., 2010). Isso é ruim dado que apesar de diversos serviços relevantes serem recuperados, vários serviços não relevantes também podem vir nessa lista, gerando muito "lixo" na lista de serviços obtidos.

Uma linguagem mais rica se faz necessária para aperfeiçoar o processo de descoberta de serviços web (PETRIE, 2009), e esse é o objetivo dos Serviços Web Semânticos (MCILRAITH; SON; ZENG, 2001). Essa linguagem mais rica deve ser legível por máquinas e por humanos, deve ter uma boa expressividade, onde tal expressividade não

implique em perda de decidibilidade. Como exemplos de linguagens desse tipo, têm-se: WSMO (BRUIJN et al., 2005), OWL-S (MARTIN et al., 2004) e SAWSDL (KOPECKY et al., 2007), sendo WSMO e OWL-S, submissões de membros da W3C.

A idéia da Web Semântica (SHADBOLT; BERNERS-LEE; HALL, 2006) é a de que agentes de software automatizem a maioria das tarefas realizadas por agentes humanos. Logo, a utilização dessas linguagens de descrição semântica facilitaria o processo de descoberta de serviços web para esses agentes de software.

## 1.1 Objetivo

O objetivo deste trabalho é identificar a similaridade entre Serviços Web Semânticos utilizando suas anotações semânticas. A abordagem adotada utiliza o algoritmo proposto por Liu et al(2009) melhorando-o em precisão e *f-measure*. A solução apresentada aqui muda a maneira de se calcular esses conjuntos, o que melhora a qualidade dos resultados em algumas configurações, como pode ser observado nos experimentos realizados.

Liu et al(2009) não apresentam nenhum experimento mostrando o desempenho de seu algoritmo. Esta dissertação apresenta experimentos mostrando como funciona tanto o algoritmo de Liu et al(2009) quanto o da melhoria apresentada nesta dissertação.

### 1.1.1 Objetivos Secundários

Como objetivos secundários, este trabalho tem a apresentação de algumas das possíveis configurações que podem ser utilizadas para os parâmetros que envolvem o algoritmo e qual delas possui o melhor resultado. Para isso, experimentos foram realizados utilizando uma base de 1083 Serviços Web Semânticos.

Um limiar também é apresentado nesta dissertação, dado que nenhum valor é apresentado no trabalho de Liu et al(2009), por fim, esta dissertação apresenta uma definição de densidade semântica para serviços Web semânticos, dado que isso é um dos parâmetros do algoritmo.

## 1.2 Hipótese

Cada serviço Web semântico possui, em sua assinatura, 4 categorias: Entradas, Saídas, Pré-Condições e Efeitos. Para calcular a similaridade entre Serviços Web Semânticos, Liu et al(2009) calculam a similaridade entre tais categorias. Por exemplo, a categoria Entrada do Serviço Web Semântico 1 é pareada com a categoria Entrada do Serviço Web Semântico 2, a categoria Saída do Serviço Web Semântico 1 é pareada com a categoria Saída do Serviço Web Semântico 2, e assim por diante.

Essas categorias, por sua vez, são nada mais do que conjuntos de conceitos. O cálculo da similaridade pode ser feito, portanto, através do cálculo da similaridade entre os conceitos presentes nessas categorias. Assim, para cada conceito presente em uma categoria, sua similaridade é calculada com cada conceito presente em outra categoria, e a partir daí é realizada uma média ponderada dessas similaridades. O valor final é dado como a similaridade entre as categorias.

Isso pode ser visto como um grafo bipartido, onde os nós são os conceitos e os dois conjuntos disjuntos são as duas categorias. Um exemplo dessa visualização pode ser observado na Figura 1.1. Percebe-se claramente que o valor da similaridade entre as categorias será diminuído por causa da similaridade do conceito A com o conceito D e da similaridade do conceito B com o conceito C.

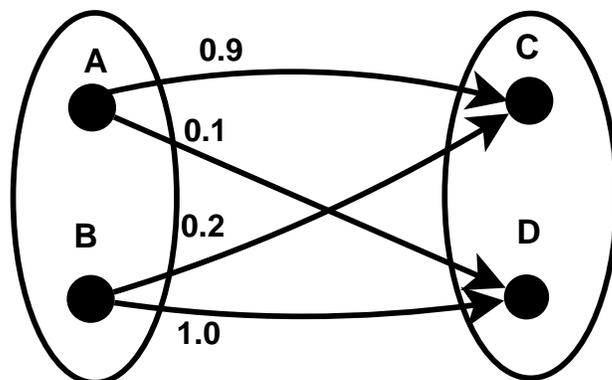


Figura 1.1: Pareamento entre conceitos de duas categorias

Este trabalho levanta uma hipótese para que o cálculo de similaridade entre duas categorias não seja tão prejudicado como é ilustrado na Figura 1.1. Cada conceito de uma categoria terá sua similaridade calculada com cada conceito da outra categoria. Porém, apenas o maior valor será levado em consideração ao fazer a média ponderada. Isso prejudicaria menos o resultado final ao calcular a similaridade entre duas categorias, aperfeiçoando o algoritmo de Liu et al(2009) para calcular a similaridade semântica entre dois Serviços Web Semânticos.

Nesse caso, utilizando a hipótese dessa dissertação, tanto a similaridade entre os conceitos A e D, quanto a similaridade entre os conceitos B e C, não seriam levada em consideração, não prejudicando o resultado final da similaridade entre as categorias.

### 1.3 Contribuição do Trabalho

A contribuição deste trabalho é propor um algoritmo refinado de análise da similaridade entre serviços web semânticos, usando como base o algoritmo de Liu et al(2009). Vários experimentos foram realizados nesse trabalho que comprovam que a hipótese levantada nessa dissertação contribui para a eficácia do algoritmo.

### 1.4 Organização do Texto

O restante do documento está estruturado conforme descrito a seguir. No capítulo 2 é dado um embasamento teórico acerca de Serviços Web explicando brevemente os principais conceitos da área. Este capítulo visa tornar o conteúdo da dissertação mais claro para o leitor caso ele não seja um conhecedor da área.

No capítulo 3 é apresentado o estado da arte destacando os trabalhos relacionados ao tema desta dissertação. São apresentados outros algoritmos de cálculo de similaridade, seus pontos fracos, pontos fortes e as semelhanças e diferenças com o presente trabalho.

As técnicas apresentadas nesse capítulo mostram tanto abordagens que utilizam apenas as informações presentes nos documentos WSDL, ou seja, abordagens puramente sintáticas, quanto abordagens que levam em conta anotações semânticas.

No capítulo 4 apresenta-se em detalhes o trabalho de Liu et al, sua contribuição, e suas limitações. Em seguida, é mostrado o aperfeiçoamento proposto por esta dissertação para conseguir obter uma melhor precisão no cálculo de similaridade entre os Serviços Web Semânticos. Além disso, são detalhados os conceitos de Serviços Web, Web Semântica, junto com a explicação das linguagens WSDL e OWL-S, fundamentais para a definição da interface de um serviço Web.

No capítulo 5 são exibidos os resultados obtidos com os experimentos, juntamente com uma análise destes resultados, que indicam o aperfeiçoamento da técnica de Liu et al. Para os experimentos, é utilizada uma base de dados chamada OWLS-TC (*OWL-S Test Collection* - Coleção de Teste OWL-S) que contém 1083 Serviços Web e 43 consultas, especificando quais Serviços Web Semânticos são relevantes ou não para cada consulta. Várias configurações são experimentadas, além da definição de densidade semântica, dado que nem as configurações, nem a definição de densidade semântica, foram feitas no trabalho de Liu et al.

Por fim, no capítulo 6, são feitas as conclusões sobre o presente trabalho. Direções sobre trabalhos futuros acerca desta dissertação também são apresentadas.

## 2 SERVIÇOS WEB

Serviços Web são definidos por (PAPAZOGLU, 2008) como um módulo de software auto-descritivo acessível via uma rede, como a Internet, e que realiza tarefas, resolve problemas e conduz transações em nome de um usuário ou de uma aplicação. Através dessas características, Serviços Web permitem que aplicações sejam construídas de maneira distribuída facilmente.

Isso é possível pelo fato de que Serviços Web são implementados utilizando padrões da Internet, o que diminui problemas com heterogeneidade de plataformas. Assim, Serviços Web hospedados em diferentes ambientes e implementados em diferentes tecnologias podem se comunicar graças ao uso de tais padrões. É afirmado por (PAPAZOGLU; RIBBERS, 2006) que a modularidade e flexibilidade dos Serviços Web os tornam ideais para integração de aplicações de *e-business*.

Os autores dão o seguinte exemplo: imagina-se um Serviço Web que checa de maneira geral a quantidade de algum produto no estoque de alguma empresa, e tal Serviço Web é provido por uma empresa que oferece soluções de TI. Tal Serviço Web pode ser acessado e integrado desta maneira por uma aplicação que realiza gerenciamento do estoque de alguma loja. Isso possibilita a integração de aplicações de empresas diferentes de uma maneira distribuída, e com um baixíssimo acoplamento.

Uma definição mais detalhada de um Serviço Web é a seguinte: Um Serviço Web é aplicação web independente de plataforma, com baixo acoplamento e programável. Essa aplicação pode ser descrita, publicada, descoberta, coordenada e configurada utilizando artefatos XML com o propósito de desenvolver aplicações distribuídas interoperáveis.

### 2.1 WSDL

WSDL é uma descrição em XML de um serviço web (PAPAZOGLU, 2008). Tal descrição é padronizada e detalha a interface provida pelo serviço web, e funciona como uma espécie de contrato entre o provedor do serviço web e seus clientes. Os detalhes que são encontrados acerca do serviço web em seu WSDL são as operações oferecidas, os protocolos utilizados, os tipos de dados, endereços para localização do serviço web e alguns requisitos não funcionais (Qualidade de Serviço). Detalhes de implementação, como a linguagem de programação em que o procedimento foi implementado não são fornecidos, tais informações são transparentes ao cliente.

Um exemplo de um documento WSDL pode ser encontrado na Listagem 2.1. Em um documento WSDL, os seguintes elementos são encontrados (CHRISTENSEN et al., 2001):

- **Types** - um container para as definições de tipos de dados usando algum sistema

de tipagem (i.e. XSD).

- **Message** - um definição abstrata e tipada de algum dado sendo comunicado.
- **Operation** - uma descrição abstrata de uma ação suportada por um Serviço.
- **PortType** - um conjunto abstrato de operações suportado por um ou mais *endpoints*.
- **Binding** - um protocolo concreto e uma especificação de um formato de dado para um *portType* específico.
- **Port** - um *endpoint* único definido como uma combinação de um *binding* com um endereço de rede.
- **Service** - uma coleção de *endpoints* relacionados.

### Listagem 2.1: Exemplo de um documento WSDL (CHRISTENSEN et al., 2001)

```

1 <?xml version="1.0" ?>
2 <definitions name="StockQuote"
3
4 targetNamespace="http://example.com/stockquote.wsdl"
5     xmlns:tns="http://example.com/stockquote.wsdl"
6     xmlns:xsd="http://example.com/stockquote.xsd"
7     xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
8     xmlns="http://schemas.xmlsoap.org/wsdl/">
9
10 <types>
11     <schema targetNamespace="http://example.com/stockquote.xsd"
12         xmlns="http://www.w3.org/2000/10/XMLSchema">
13         <element name="TradePriceRequest">
14             <complexType>
15                 <all>
16                     <element name="tickerSymbol" type="string"/>
17                 </all>
18             </complexType>
19         </element>
20         <element name="TradePrice">
21             <complexType>
22                 <all>
23                     <element name="price" type="float"/>
24                 </all>
25             </complexType>
26         </element>
27     </schema>
28 </types>
29
30 <message name="GetLastTradePriceInput">
31     <part name="body" element="xsd:TradePriceRequest"/>
32 </message>
33
34 <message name="GetLastTradePriceOutput">
35     <part name="body" element="xsd:TradePrice"/>
36 </message>
37
38 <portType name="StockQuotePortType">

```

```

39     <operation name="GetLastTradePrice">
40         <input message="tns:GetLastTradePriceInput"/>
41         <output message="tns:GetLastTradePriceOutput"/>
42     </operation>
43 </portType>
44
45 <binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType"
46 >
47     <soap:binding style="document" transport="http://schemas.
48         xmlsoap.org/soap/http"/>
49     <operation name="GetLastTradePrice">
50         <soap:operation soapAction="http://example.com/
51             GetLastTradePrice"/>
52         <input>
53             <soap:body use="literal"/>
54         </input>
55         <output>
56             <soap:body use="literal"/>
57         </output>
58     </operation>
59 </binding>
60
61 <service name="StockQuoteService">
62     <documentation>My first service</documentation>
63     <port name="StockQuotePort" binding="tns:StockQuoteBinding">
64         <soap:address location="http://example.com/stockquote"/>
65     </port>
66 </service>
67
68 </definitions>

```

---

## 2.2 UDDI

UDDI foi criado com o propósito de facilitar a etapa de registro e descoberta de serviços (PAPAZOGLU, 2008). A especificação de UDDI propõe uma maneira padronizada de descrever serviços web, localizar negócios e integrar serviços web.

A Figura 2.1 mostra o modelo de uso de UDDI. Primeiramente são criadas definições de tipos de serviços pelos consórcios industriais e provedores de serviços. A partir destes tipos definidos, os serviços implementados são publicados nos registros de acordo com os tipos definidos, padronizando assim a maneira em que os serviços são registrados. A partir dessa base de registros, os serviços são encontrados pelos clientes, os clientes analisam os detalhes das definições dos tipos dos serviços, e, ao encontrar os serviços desejados, realizam as invocações em si.

Uma iniciativa tomada pela IBM, Microsoft, SAP e NTT foi a criação do UBR, uma espécie de coleção de registros acessíveis publicamente. Nesses registros, existem três componentes interrelacionados: *páginas brancas*, que contêm informações de contato das empresas provedoras de serviço, como endereço e telefone; *páginas amarelas* que fornecem informações acerca das classificações dos negócios baseadas em taxonomias industriais padronizadas; e as *páginas verdes* que fornecem descrições sobre os serviços oferecidos e suas funcionalidades, assim como URLs apontando para os artefatos de descrição.

Entretanto, as descrições dos Serviços Web são textuais, indicando no máximo uma URL com maiores detalhes acerca do Serviço. Devido à natureza das descrições em

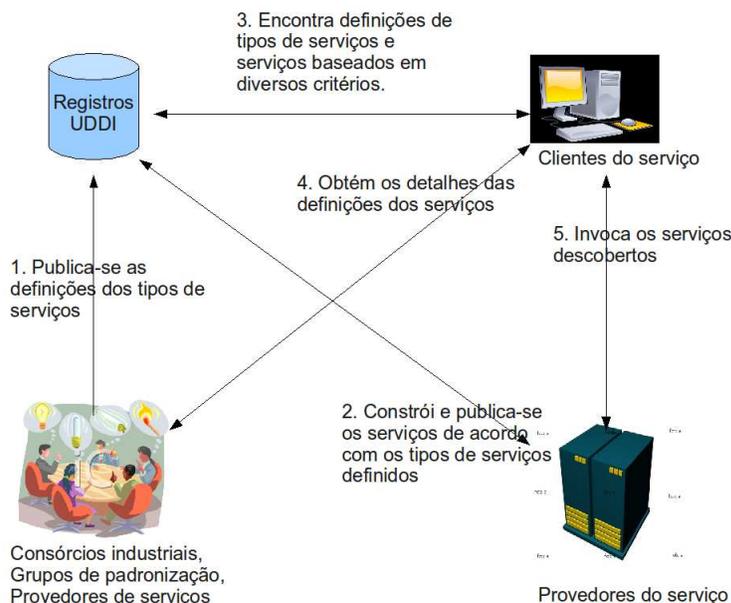


Figura 2.1: Modelo de uso de UDDI (PAPAZOGLU, 2008)

UDDI, as consultas não permitem que detalhes técnicos sejam utilizados, o que resulta em consultas gerais. Além disso, os resultados de consultas não recebem um ranqueamento em relação à relevância, tornando UDDI uma solução não muito prática para descoberta de Serviços Web.

CURBERA et al. (2002) afirma que existem dois elementos básicos que definem um registro UDDI: *businessEntity* e *businessService*. Um exemplo de um *businessEntity* simplificado pode ser visto na Listagem 2.2 e um exemplo de um *businessService* simplificado pode ser visto na Listagem 2.3. O elemento *businessEntity* define mais informações de páginas brancas, como a descrição do negócio em si, endereço, informações de contato em geral.

#### Listagem 2.2: Exemplo de um *businessEntity* simplificado (CURBERA et al., 2002)

```

1 <businessEntity businessKey=
2     "A687FG00-56NM-EFT1-3456-098765432124">
3     <name>Acme Travel Incorporated</name>
4     <description xml:lang="en">
5         Acme is a world leader in online travel services
6     </description>
7     <contacts>
8         <contact useType="US general">
9             <personName>Acme Inc.</personName>
10            <phone>1 800 CALL ACME</phone>
11            <email useType="">acme@acme-travel.com</email>
12            <address>
13                <addressLine>Acme</addressLine>
14                <addressLine>12 Maple Avenue</
15                    addressLine>
16                <addressLine>Springfield , CT 06785</
17                    addressLine>
18            </address>
19        </contact>
20    </contacts>

```

```

19     <businessServices> ...
20   </businessServices>
21   <identifierBag> ...
22 </identifierBag>
23   <categoryBag> ...
24     <keyedReference tModelKey=
25       "UUID:DB77450D-9FA8-45D4-A7BC-04411D14E384"
26       keyName="Electronic check-in"
27       keyValue="84121801" />
28   </categoryBag>
29 </businessEntity>

```

---

### Listagem 2.3: Exemplo de um *businessService* simplificado (CURBERA et al., 2002)

---

```

1 <businessService serviceKey=
2   "894B5100-3AAF-11D5-80DC-002035229C64"
3   businessKey=
4   "D2033110-3AAF-11D5-80DC-002035229C64">
5   <name>ElectronicTravelService</name>
6   <description xml:lang="en">Electronic Travel
7     Service</description>
8   <bindingTemplates>
9     <bindingTemplate bindingKey=
10       "6D665B10-3AAF-11D5-80DC
11         -002035229C64"
12       serviceKey=
13         "89470B40-3AAF-11D5-80DC
14           -002035229C64">
15       <description>
16         SOAP-based e-checkin and flight info
17       </description>
18       <accessPoint URLType="http">
19         http://www.acme-travel.com/
20         travelservice
21       </accessPoint>
22       <tModelInstanceDetails>
23         <tModelInstanceInfo tModelKey=
24           "D2033110-3BGF-1KJH-234C
25             -09873909802">
26           ...
27         </tModelInstanceInfo>
28       </tModelInstanceDetails>
29     </bindingTemplate>
30   </bindingTemplates>
31   <categoryBag> ...
32 </categoryBag>
33 </businessService>

```

---

Associado a um *businessEntity* estará associado um ou mais *businessService*. O elemento *businessService* descreve o serviço em si. Tanto um *businessEntity* quanto um *businessService* pode ter um ou mais elementos do tipo *categoryBag* para definir quais categorias eles pertencem. O elemento *categoryBag* corresponde às páginas amarelas dos registros UDDI. Além disso, cada entidade possui um UUID, um identificador único universal que pode ser referenciado por diferentes documentos para identificar entidades específicas.

Esses UUIDs podem ser vistos nas Listagens 2.2 e 2.3, onde um *businessKey* identifica uma *businessEntity* e uma *serviceKey* identifica uma *businessService*. Cada *business-*

*sService* possui um ou vários *bindingTemplate*, que fornecerá um *accessPoint* (ponto de acesso) à implementação do serviço. Este ponto de acesso pode ser uma URL, um email ou até mesmo um número de telefone. O que se supõe é que um determinado serviço pode ter várias implementações, por isso que um *businessService* pode ter um ou mais *bindingTemplate*.

CURBERA et al. (2002) afirma que a parte mais interessante em um elemento do tipo *bindingTemplate* está no elemento *tModelInstanceDetails*. O elemento *tModelInstanceDetails* referencia algo que corresponde às páginas verdes dos registros UDDI. A especificação técnica da interface de um serviço é definida no elemento chamado *tModel*, que pode ser visto na Listagem 2.4.

Listagem 2.4: Exemplo de um *tModel* simplificado (CURBERA et al., 2002)

---

```

1 <tModel tModelKey="">
2   <name>http://www.travel.org/e-checkin-interface</name>
3   <description xml:lang="en">
4     Standard service interface definition for travel
5     services
6   </description>
7   <overviewDoc>
8     <description xml:lang="en">
9       WSDL Service Interface Document
10    </description>
11    <overviewURL>
12      http://www.travel.org/services/e-checkin.wsdl
13    </overviewURL>
14  </overviewDoc>
15  <categoryBag> ...
16  </categoryBag>
17 </tModel>

```

---

A Listagem 2.4 apresenta um exemplo de um *tModel* que ilustra um caso fictício no qual a indústria de viagens resolve criar um padrão de interface WSDL para os serviços de *check-in* eletrônico. Logo, qualquer serviço que implemente essa interface pode referenciar esse *tModel* a partir de seu *bindingTemplate*.

## 2.3 Orquestração e Coreografia

Orquestração e coreografia se referem a questões de integração e organização de serviços web complexos. Apesar de aparentemente significarem a mesma coisa, são conceitos diferentes, que (PELTZ, 2003) define da seguinte forma:

- **Orquestração** - Descreve como Serviços Web interagem entre si, através de troca de mensagens, especificando a ordem da execução e a lógica de negócios a partir da perspectiva e do controle de um único ponto. Um exemplo disto é a Figura 2.2 onde a perspectiva é do fornecedor. Com orquestração, as interações do processo de negócios são sempre controladas a partir de uma determinada perspectiva (privada) de um dos grupos de negócios envolvidos no processo.
- **Coreografia** - é tipicamente a visão global das trocas de mensagens, regras de interação, e acordos realizados entre os pontos de processo de negócios envolvendo vários grupos de negócios, ao invés de apenas um grupo (que é o que ocorre na orquestração). Na coreografia, a sequência de mensagens trocadas são armazenadas,

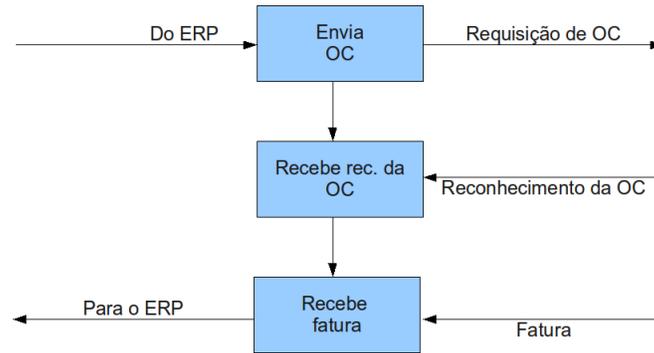


Figura 2.2: Orquestração de uma Ordem de Compra (OC) da perspectiva do fabricante (PAPAZOGLU, 2008)

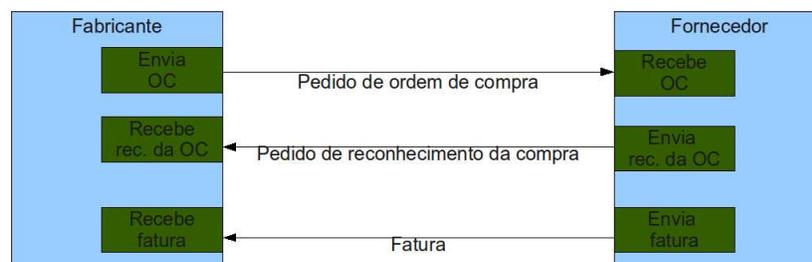


Figura 2.3: Coreografia de uma Ordem de Compra (OC) (PAPAZOGLU, 2008)

onde cada grupo (cliente, fornecedor, parceiros) descrevem as mensagens que são trocadas, não existindo um foco em cima da perspectiva de um determinado grupo. Um exemplo é mostrado na Figura 2.3.

## 2.4 Web Semântica

A Web Semântica proposta por (BERNERS-LEE; HENDLER; LASSILA, 2001) e revisada por (SHADBOLT; BERNERS-LEE; HALL, 2006) visa estender a Web atual para uma web legível por máquinas. Nessa Web estendida, as páginas possuem anotações semânticas com descrições que permitem o uso de algoritmos de inferências facilitando o processo de automação de várias atividades (STUDER; GRIMM, 2007). Sua arquitetura em camadas é mostrada na Figura 2.4.

### 2.4.1 Ontologia

Ontologia é uma representação formal explícita do conhecimento de alguns conceito em um certo domínio, descrevendo suas características e atributos, e especificando restrições (LIU et al., 2009). Uma definição mais formal é a seguinte: Uma ontologia  $O$  consiste em seis elementos  $\{C, A^C, R, A^R, H, X\}$ , onde  $C$  representa o conjunto de conceitos;  $A^C$  representa o conjunto de atributos, um para cada conceito;  $R$  representa o conjunto de relacionamentos;  $A^R$  representa o conjunto de atributos, um para cada relacionamento;  $H$  representa a hierarquia de conceitos;  $X$  representa o conjunto de axiomas. Um exemplo de ontologia é mostrado na Figura 2.5.

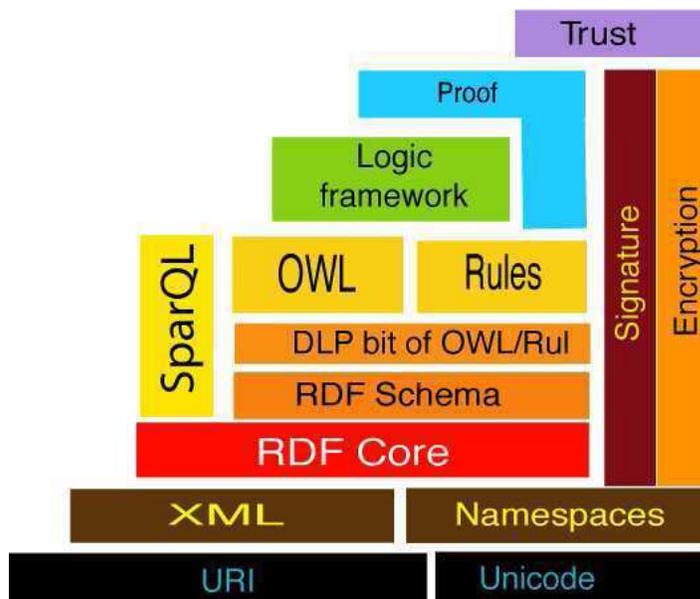


Figura 2.4: Arquitetura em camadas da Web Semântica atualizada (KIFER et al., 2005)

## 2.4.2 SWRL

SWRL é um acrônimo para *Semantic Web Rule Language* (Linguagem de Regras da Web Semântica). Ela é baseada na combinação de OWL com Rule Markup Language, uma linguagem de marcação para descrição de regras (HORROCKS et al., 2004).

As regras possuem uma forma de implicação entre antecedente (corpo) e conseqüente (cabeça). Isso significa que, toda vez que as condições presentes no antecedente forem verdade, as condições presentes no conseqüente também terão de ser verdade. Cada átomo nessas regras pode ser da forma  $C(x)$ ,  $P(x, y)$ ,  $sameAs(x, y)$  ou  $differentFrom(x, y)$ , onde  $C$  é um conceito em OWL,  $P$  é uma propriedade em OWL,  $x$  e  $y$  são instâncias, variáveis ou valores de dados.

Abaixo, um exemplo de uma regra em SWRL:

$$pai(?x, ?y) \wedge irmao(?y, ?z) \Rightarrow tio(?x, ?z)$$

## 2.5 Serviços Web Semânticos

Serviços Web Semânticos empregam tecnologias da Web Semântica na área de Serviços Web: as funcionalidades, as entradas e saídas, pré-condições e efeitos, tudo é representado em ontologias (MCILRAITH; SON; ZENG, 2001; SYCARA et al., 2003; CABRAL et al., 2004; VERMA et al., 2005).

Ao usar tais tecnologias, os Serviços Web Semânticos podem informar, em anotações legíveis por máquinas, as suas funcionalidades e as sequências em que mensagens devem ser trocadas. Anotando as interfaces dos Serviços Web com ontologias, especificando o "que" o serviço web faz e "como" ele faz, as publicações de registros de serviços web ficam bem mais elaboradas, permitindo desta maneira, formas de descobertas mais sofisticadas que UDDI.

$$O_{univ} = \{C_{univ}, A_{univ}^C, R_{univ}, A_{univ}^R, H_{univ}, X_{univ}\}$$

where

$$C_{univ} = \{Student, PhDStudent, AcademicStaff, Professor, Department, Course, Project, \dots\}$$

$$A_{univ}^C = \{A_{univ}^C(Student), A_{univ}^C(PhDStudent), A_{univ}^C(AcademicStaff), A_{univ}^C(Professor), A_{univ}^C(Department), A_{univ}^C(Course), A_{univ}^C(Project), \dots\}$$

$$A_{univ}^C(Student) = \{name, matricnum, email, \dots\}$$

$$A_{univ}^C(PhDStudent) = \{name, matricnum, project, \dots\}$$

$$A_{univ}^C(AcademicStaff) = \{name, staffid, \dots\}$$

$$A_{univ}^C(Professor) = \{name, staffid, email, \dots\}$$

$$A_{univ}^C(Department) = \{name, researcharea, \dots\}$$

$$A_{univ}^C(Course) = \{name, title, period, \dots\}$$

$$A_{univ}^C(Project) = \{name, title, period, \dots\}$$

.....

$$R_{univ} = \{Super\ vise(Professor, PhDStudent), Teach(Professor, Course), TaughtBy(Course, Professor), Faculty(AcademicStaff, Department), TakeCourse(Student, Course), \dots\}$$

$$A_{univ}^R = \{A_{univ}^R(Super\ vise), A_{univ}^R(WorkIn), A_{univ}^R(Major), A_{univ}^R(Teach), A_{univ}^R(Take), \dots\}$$

$$A_{univ}^R(Super\ vise) = \{startdate, enddate, \dots\}$$

$$A_{univ}^R(WorkIn) = \{apptdate, \dots\}$$

$$A_{univ}^R(Major) = \{academicyear, \dots\}$$

$$A_{univ}^R(Teach) = \{semester, year, \dots\}$$

$$A_{univ}^R(Take) = \{semester, year, \dots\}$$

.....

$$H_{univ} = \{(Student, PhDStudent), (AcademicStaff, Professor), \dots\}$$

$$X_{univ} = \{TaughtBy(X, Y) \leftarrow Teach(Y, X), \dots\}$$

Figura 2.5: Exemplo de Ontologia (LIU et al., 2009)

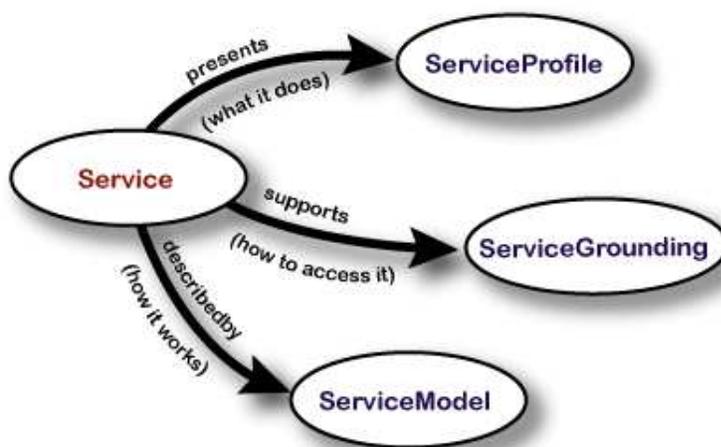


Figura 2.6: OWL-S (MARTIN et al., 2004)

### 2.5.1 OWL-S

OWL-S consiste em uma ontologia descrita em OWL para descrever Serviços Web Semânticos (MARTIN et al., 2004). A Figura 2.6 mostra seu modelo. Cada Serviço Web Semântico é representado pelo conceito *Service*. Cada Serviço Web Semântico apresenta um perfil representado pelo conceito *ServiceProfile* que mostra o "que"o Serviço Web Semântico faz. A Figura 2.7 apresenta a principal classe que representa o perfil de um Serviço Web Semântico e suas propriedades.

Como foi mencionado anteriormente, o perfil de um Serviço Web Semântico especifica o "que"o Serviço Web Semântico faz. Isso engloba especificar suas entradas, saídas, pré-condições e efeitos. Além disso, outras características podem ser especificadas no perfil de um Serviço Web Semântico, características como categoria do Serviço Web Semântico, tempo de resposta, qualidade de serviço, disponibilidade geográfica, etc.

O conceito *ServiceModel* especifica "como"o Serviço Web Semântico trabalha. No caso, esse conceito detalha se o Serviço faz parte de algum processo que envolve orquestração. O conceito *ServiceGrounding* detalha "como"acessar o Serviço Web Semântico, detalhando quais protocolos de rede são utilizados e quais WSDL descrevem a interface dos serviços, fazendo um mapeamento dos conceitos presentes no perfil do Serviço Web Semântico para os tipos de dados encontrados na WSDL do Serviço Web.

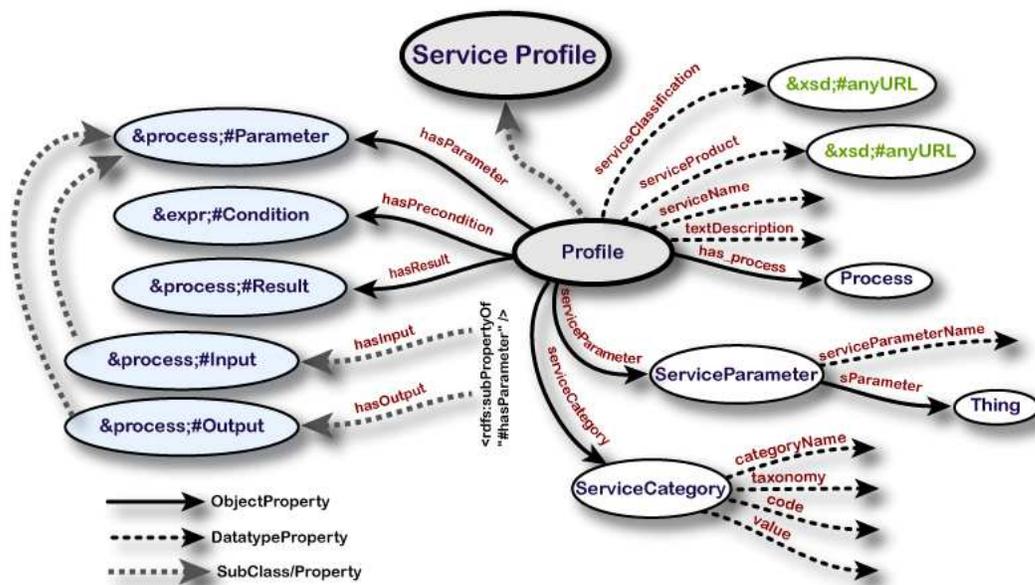


Figura 2.7: *ServiceProfile* (MARTIN et al., 2004)

## 3 ESTADO DA ARTE

Este capítulo tem como objetivo apresentar trabalhos relacionados a esta dissertação publicados nos últimos anos nos principais veículos relacionados a área de serviços Web e Web Semântica. Foram encontradas duas vertentes na literatura: métricas entre serviços Web Tradicionais e métricas entre serviços Web semânticos. Assim sendo, este capítulo possui três seções, duas apresentando os principais trabalhos encontrados em cada uma dessas vertentes e uma apontando a deficiência de cada um dos trabalhos apresentados neste capítulo.

### 3.1 Métricas de Similaridade entre Serviços Web Tradicionais

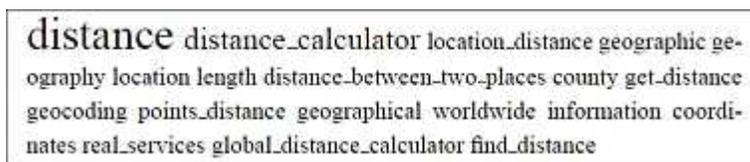
Esta seção apresenta dois trabalhos encontrados na literatura que calculam a similaridade de certos aspectos de serviços Web sem o uso de anotações semânticas.

#### 3.1.1 WSCOLAB

O uso de tags para realização de *matchmakings* (termo utilizado na literatura para o processo de similaridade entre serviços Web) é proposto por (GAWINECKI et al., 2010), onde os usuários classificam de maneira colaborativa (através de tags) três facetas: entrada, saída e comportamento. Um exemplo é mostrado nas Figuras 3.1, 3.2 e 3.3, fonte: (WSCOLAB, 2010).

As figuras apresentam nuvens de tags, uma nuvem de tag nada mais do que a visualização de um determinado conjunto de tags, onde as tags mais utilizadas são destacadas. Um exemplo de nuvem de tag pode ser visto na Figura 3.4, fonte: (CROFT; METZLER; STROHMAN, 2009).

O algoritmo de matchmaking de (GAWINECKI et al., 2010) retorna serviços web que são tanto compatíveis com a interface (as tags de entrada e saída casam no mínimo com uma tag de entrada da consulta e com uma tag de saída da consulta) ou compatíveis com o comportamento (possibilita que serviços Web com diferentes interfaces sejam recuperados).



```
distance distance_calculator location_distance geographic ge-
ography location length distance_between_two_places county get_distance
geocoding points_distance geographical worldwide information coordi-
nates real_services global_distance_calculator find_distance
```

Figura 3.1: Nuvem de tags para o comportamento do serviço



No WSCOLAB, cada usuário faz suas consultas utilizando tags, assim, atribuindo tags de entrada, tags de saída e tags de comportamento para a sua consulta. Sua consulta é representada da seguinte forma:

$$q = (q_i, q_o, q_b)$$

onde  $q_i$ ,  $q_o$  e  $q_b$  são conjuntos de tags, correspondendo respectivamente à entrada, saída e comportamento do serviço Web que o usuário está procurando.

Como é necessário que as tags utilizadas pelo usuário existam de fato na folksonomia<sup>1</sup> da coleção, WSCOLAB sugere tags a medida que o usuário digita as tags de sua consulta, onde a sugestão é baseada nas 15 tags mais utilizadas para a faceta em questão.

WSCOLAB oferece dois tipos de busca: uma é a busca binária e outra é a busca ranqueada. A busca binária é bem simples e não fornece nenhum ordenamento, o que pode ser um problema caso o número de serviços Web recuperados for grande. A busca ranqueada por sua vez é mais elaborada e oferece um ordenamento dos serviços Web recuperados.

### 3.1.1.1 Busca Binária

A definição formal da busca binária utiliza conceitos de álgebra relacional, como Projeção ( $\pi_p$ ) e Seleção ( $\sigma_p$ ). Projeção equivale à cláusula SQL SELECT DISTINCT e Seleção equivale à cláusula SQL WHERE. Assim, para uma consulta  $q = (q_i, q_o, q_b)$ , os resultados serão  $r(q, (F_i, F_o, F_b))$ , onde

$$r(q, (F_i, F_o, F_b)) = r(q_i, F_i) \cap r(q_o, F_o) \cup r(q_b, F_b)$$

onde

$$r(q_b, F_b) = \pi_S(\sigma_{t \in q_b}(F_b))$$

$$r(q_i, F_i) = \pi_S(\sigma_{t \in q_i}(F_i))$$

$$r(q_o, F_o) = \pi_S(\sigma_{t \in q_o}(F_o))$$

lembrando que  $S$  é o conjunto de Serviços Web nos registros.

### 3.1.1.2 Busca ranqueada

Para a realização da busca ranqueada cada consulta terá um valor de similaridade associado com cada serviço Web da base de serviços Web. Essa similaridade é calculada da seguinte forma:

$$\text{sim}(q, s) = w_b * \text{sim}(q_b, \sigma_s(F_b)) + w_i * \text{sim}(q_i, \sigma_s(F_i)) + w_o * \text{sim}(q_o, \sigma_s(F_o))$$

onde  $w_i = w_o = 0,4$  e  $w_b = 0,2$ . Esses valores foram obtidos através de experimentos realizados (GAWINECKI et al., 2010) onde foi concluído que tais valores apresentam melhores resultados. Para cada faceta, a similaridade é calculada utilizando o Modelo de Espaço Vetorial (SALTON; BUCKLEY, 1988).

Cada consulta será representada por um vetor de  $m$  dimensões, onde  $m$  é igual ao número de tags distintas que aquela faceta possui,  $m = |T|$ . Os pesos atribuídos para cada tag são dados utilizando a abordagem TF/IDF (SALTON; BUCKLEY, 1988). A similaridade é dada pelo cosseno formado entre os dois vetores. Assim, por exemplo, dada a consulta  $q_i$  e as tags de entrada que um serviço Web oferece,  $\sigma_s(F_i)$ , a similaridade entre esses dois vetores é definida da seguinte maneira:

<sup>1</sup>Folksonomia é o conjunto de tags de um corpus (CROFT; METZLER; STROHMAN, 2009)

$$\text{sim}(q_i, \sigma_s(F_i)) = \frac{\sum_{t \in q_i} w_{s,t}}{W_s},$$

onde

$$w_{s,t} = tf_{s,t} * idf_t, W_s = \sqrt{\sum_{t \in q_i} w_{s,t}^2}$$

$$tf_{s,t} = \frac{n_{s,t}}{N_s}, idf_t = \log \frac{|S|}{1+|S_t|}$$

sendo  $n_{s,t}$  o número de usuários que anotaram a entrada do serviço Web  $s$  e  $N_s$  o número de anotações que todos os usuários fizeram para o serviço Web  $s$ .  $|S|$  é o número de todos os serviços Web da coleção, e  $|S_t|$  é o número de todos os serviços Web que têm a entrada anotada com a tag  $t$ . O processo para cálculo de similaridade com as outras facetas é semelhante ao processo da faceta de entrada.

### 3.1.2 Similaridade baseada em assinaturas

Esse trabalho (LIU et al., 2010) apresenta uma abordagem baseada na assinatura das operações de cada serviço Web. Normalmente, as operações possuem nomes que correspondem à semântica de sua implementação. Esses nomes são concatenados em um único termo, onde esse termo começa com a letra da primeira palavra, seguida pelas outras palavras concatenadas, onde cada palavra é identificada pela presença de uma letra maiúscula.

Por exemplo, *getMapOfUSAddress* é uma concatenação dos termos *get*, *Map*, *Of*, *US*, *Address*. Isso é uma convenção de código em várias linguagens orientadas a objeto, como na linguagem Java por exemplo<sup>2</sup>. Nesse trabalho (LIU et al., 2010), cada serviço Web tem sua similaridade calculada com a de outro serviço Web com base nas nomenclaturas de suas operações.

Primeiramente, é realizado um pré-processamento dos nomes das operações de cada serviço Web, e, em seguida, são calculadas as similaridades semânticas entre os termos com base em outro trabalho (SAHAMI; HEILMAN, 2006). O pré-processamento ocorre utilizando as quatro regras mostradas na Quadro 3.1, fonte: (SAHAMI; HEILMAN, 2006).

Quadro 3.1: Regras de decomposição de nomes

Regra	Nome	Palavra
Mudança de Caixa	filmeAção	filme, ação
Mudança de Caixa	ComédiaFilme	comédia, filme
Eliminação de Sufixos com Número	filme1	filme
Separação de <i>underlines</i>	filme_comédia	filme, comédia

Dado um termo  $k$ ,  $N(k)$  é o número de páginas que o Google retorna como resultado das páginas Web que contêm o termo  $k$ . Sendo  $N(k_1 \cap k_2)$  o número de páginas que contêm os termos  $k_1$  e  $k_2$ , a similaridade entre os termos é calculada da seguinte maneira:

$$s = \frac{\log\left(\frac{N(k_1 \cap k_2) \times N}{N(k_1) \times N(k_2)}\right)}{\log N}$$

<sup>2</sup><http://www.oracle.com/technetwork/java/javase/documentation/codeconventions-135099.html#367>

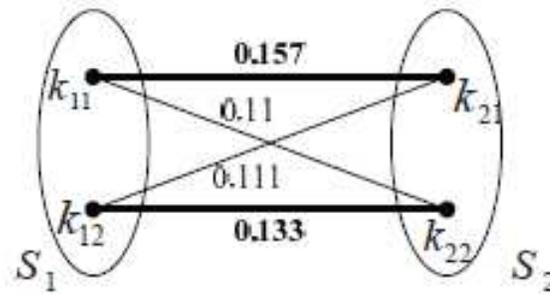


Figura 3.5: Comparando serviços Web como grafos bipartidos

sendo  $N$  o número total de páginas na Web, onde usualmente geralmente o valor  $10^{11}$  é atribuído a esse parâmetro.

Esse trabalho representa um serviço Web como um grafo bipartido como pode ser visto na Figura 3.5, fonte: (LIU et al., 2010). Esse grafo é formalmente definido como  $G = (S_1, S_2, E)$  onde  $S_1$  é o serviço Web 1 e  $S_2$  é o serviço Web 2.

Cada serviço Web é representado pelos termos que são extraídos de suas assinaturas através das regras mostradas no Quadro 3.1.  $E$  na definição do grafo bipartite corresponde à similaridade calculada, e  $E$  ligará cada elemento de  $S_1$  com os elementos de  $S_2$ , lembrando que  $|S_1| \leq |S_2|$ .

Assim sendo,  $M \subset E$ , onde cada vértice de destino e de origem das arestas do conjunto  $M$  é distinto,  $M$  é o somatório máximo das arestas de  $E$ , onde esse valor será denominado de  $max\_value_{1,2}$ . Logo, a similaridade entre dois serviços Web será:

$$similaridade_{1,2} = \frac{max\_value_{1,2}}{|M|}$$

### 3.1.3 Análise das Abordagens

O trabalho de LI et al. (2011) tem a desvantagem de necessitar de um histórico contendo a troca de mensagens entre os serviços Web, o que será algo um pouco difícil de encontrar na *Service Web*. Porém aparenta ser uma abordagem razoável em coleções fechadas, como em registros de serviços Web internos de empresas.

A contribuição de (WSCOLAB, 2010) aparenta ser bastante eficiente pelos resultados apontados em seus experimentos. Porém, ela depende que usuários analisem os Serviços Web em si, e forneçam tags razoáveis e adequadas para eles, assim, dependendo muito da contribuição da comunidade.

LIU et al. (2010) também obtiveram bons resultados em seus experimentos, entretanto, seu trabalho é muito dependente de idioma e encontra sérios problemas em casos de sinonímia e homonímia. Nos casos de sinonímia, têm-se palavras com significados iguais e grafias diferentes, e nos casos de homonímia têm-se uma palavra com mais de um significado. Logo, a similaridade entre palavras que se encontram nesses casos só funciona utilizando o contexto em que elas são aplicadas.

## 3.2 Métricas de Similaridade entre Serviços Web Semânticos

Esta seção apresenta quatro trabalhos encontrados na literatura que calculam a similaridade de certos aspectos de serviços Web com o uso de anotações semânticas.

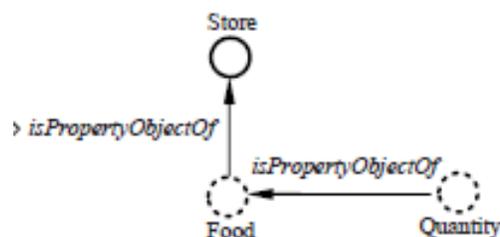


Figura 3.6: Grafo de ligações semânticas

### 3.2.1 Similaridade através de extração de conhecimento

Esse trabalho (WEI et al., 2008) apresenta uma métrica que é baseada nas ligações semânticas entre os conceitos envolvidos nas assinaturas dos serviços Web. A principal motivação dessa técnica é a de que, muitas vezes, só considerar a assinatura (entrada/saída) não é suficiente.

O exemplo mostrado pelos autores é um em que dois serviços Web semânticos que têm como entrada *Mercearia* e como saída *Comida*. Apesar da assinatura dos dois ser igual, um retorna a quantidade de uma determinada comida no estoque e o outro retorna o preço.

Para resolver esse problema, esse trabalho propõe o uso de um grafo de ligações semânticas entre os conceitos presentes na assinatura do Serviço Web Semântico. Essas ligações semânticas nada mais são do que ligações entre os conceitos, onde essas ligações são representadas por propriedades.

Cada ligação pode ser representada por uma sentença  $(SC, CT, OC)$ , onde  $SC$  representa o conceito que é sujeito da sentença,  $OC$  representa o conceito que é objeto da sentença e  $CT$  representa o predicado da sentença. Um exemplo é a sentença  $(Livro, "publicado pela", "Springer")$ . Um grafo de ligações semânticas é mostrado na Figura 3.6, fonte: (WEI et al., 2008).

Essas restrições são extraídas a partir das descrições textuais em linguagem natural do serviço Web. Cada conceito corresponde a uma palavra-chave na descrição textual, e suas possíveis relações são obtidas a partir da estrutura sintática do texto. A Figura 3.7 mostra o processo de extração, onde, primeiramente, o texto é pré-processado, sendo feita a tokenização e a identificação de conceitos e entidades, logo após é realizada a geração da *parse-tree*, e, a partir da *parse-tree*, é feita a extração para obtenção do grafo de ligações semânticas.

A similaridade é então calculada a partir dos grafos de ligações semânticas. Sendo  $C_r$  o grafo de ligações semânticas do serviço web  $r$  e  $C_s$  o grafo de ligações semânticas do serviço web  $s$ , a similaridade entre os dois serviços é calculada pela seguinte fórmula:

$$ConstraintGraphMatch(C_r, C_s) = \sum_{i=1}^P \max_{j \in P'} (TripleMatch(RT_i, ST_j)) / P$$

onde  $P$  é o número de triplas contidas no grafo  $C_r$ ,  $P'$  é o número de triplas contidas no grafo  $C_s$ , e a função  $TripleMatch(RT_i, ST_j)$  é usada para estimar a similaridade entre duas triplas,  $RT_i \in C_r$  e  $ST_j \in C_s$ .

Para calcular a similaridade entre as duas triplas, são necessários três passos:

- **Passo 1:** computar o grau de similaridade entre os conceitos que são sujeitos da tripla;

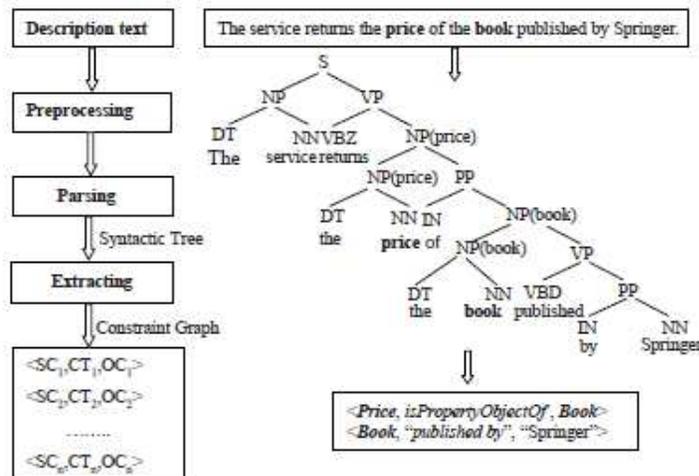


Figura 3.7: Extração das ligações semânticas (WEI et al., 2008)

- **Passo 2:** computar o grau de similaridade entre os conceitos que são objetos da tripla se as restrições da tripla forem do mesmo tipo e os conceitos forem similares;
- **Passo 3:** computar a soma ponderada dos valores obtidos nos passos anteriores se os conceitos forem similares.

Para verificar se existe algum grau de similaridade entre dois conceitos, a taxonomia da ontologia é analisada. Existem cinco níveis de similaridade: *Exact*, *Plug-in*, *Subsumed-by*, *Intersect*, e *Fail*. Sendo  $r$  o conceito requisitado e o  $s$  o conceito sendo comparado, os detalhes são mostrados a seguir:

- **Exact match:** os conceitos são exatamente iguais.
- **Plugin-match:**  $r \in Pais(s) \vee s \in Filhos(r)$
- **Subsumed-match:**  $s \in Pais(r) \vee r \in Filhos(s)$
- **Intersect match:**  $\frac{||Pais(r) \wedge Pais(s)||}{\max(||Pais(r)||, ||Pais(s)||)} \geq 0.5$
- **Fail:** se o conceito  $r$  não se encaixa em nenhuma das situações acima.

### 3.2.2 OWLS-MX

A abordagem de (KLUSCH; FRIES; SYCARA, 2009) utiliza uma técnica híbrida, dado que ela utiliza informações semânticas e sintáticas. Os autores dividem as técnicas em dois tipos: as baseadas em inferência lógica e as baseadas em recuperação de informação. Esse algoritmo não utiliza informações presentes nas pré-condições e efeitos da assinatura semântica de um serviço Web semântico.

A OWLS-MX utiliza vários filtros de similaridade. Existem cinco tipos de filtros: *EXACT*, *PLUG IN*, *SUBSUMES*, *SUBSUMED-BY* e *NEAREST-NEIGHBOR*. Os três primeiros são semânticos e os dois últimos são considerados híbridos por utilizar informações consideradas sintáticas.

Uma consulta  $Q$  vai ser composta por dois conjuntos de conceitos descritos utilizando Lógica de Descrição, onde esses conjuntos serão  $I_Q$  e  $O_Q$ , representando respectivamente

as entradas e saídas da consulta  $Q$ . Da mesma forma o serviço Web Semântico  $S$  será descrito por dois conjuntos,  $I_S$  e  $O_S$ , representando respectivamente as entradas e saídas do serviço Web semântico  $S$ . Cada conceito  $C$  presente nesses conjuntos deve estar presente na terminologia  $T$  (TBox), especificada em uma ontologia descrita em Lógica de Descrição.  $CT_T$  é a hierarquia dos conceitos nessa terminologia  $T$ .

$LSC(C)$  é a lista de conceitos específicos diretos de  $C$ , ou seja, que são subconceitos diretos de  $C$ , sem nenhum outro conceito no caminho, e  $LSG(C)$  é a lista de conceitos genéricos diretos de  $C$ .  $SynSim(Q, S)$  é a similaridade textual entre a consulta  $Q$  e o Serviço Web Semântico  $S$ . Sendo  $\alpha$  um limiar de similaridade presente no intervalo  $[0,1]$ ,  $\sqsubseteq$  o símbolo que representa que um conceito é especificação de outro e  $\equiv$  o símbolo que representa que dois conceitos são equivalentes, tem-se que os seguintes graus de similaridade calculados por OWLS-MX<sup>3</sup>:

- **Exact match:** Cada conceito presente na entrada da consulta  $Q$  é equivalente a algum conceito na entrada do Serviço Web Semântico  $S$ , e vice-versa. O mesmo ocorre para os conceitos presentes na saída de  $Q$  e  $S$ . Formalmente, esse grau é definido da seguinte forma:  $\forall C \in I_S \exists C' \in I_Q: C \equiv C' \wedge \forall D \in O_Q \exists D' \in O_S: D \equiv D'$ .
- **Plug-in match:**  $\forall C \in I_S \exists C' \in I_Q: C' \sqsubseteq C \wedge \forall D \in O_Q \exists D' \in O_S: D' \in LSC(D)$ .
- **Subsumes match:**  $\forall C \in I_S \exists C' \in I_Q: C' \sqsubseteq C \wedge \forall D \in O_Q \exists D' \in O_S: D' \sqsubseteq D$ .
- **Subsumed-by match:**  $\forall C \in I_S \exists C' \in I_Q: C' \sqsubseteq C \wedge \forall D \in O_Q \exists D' \in O_S: D' \equiv D \vee D' \in LGC(D)$ .
- **Logical Fail:** OWLS-MX retorna esse grau se todos os graus acima falharem.
- **Hybrid subsumed-by match:**  $\forall C \in I_S \exists C' \in I_Q: C' \sqsubseteq C \wedge \forall D \in O_Q \exists D' \in O_S: (D' \equiv D \vee D' \in LGC(D)) \wedge SynSim(Q, S) \geq \alpha$ . Esse filtro híbrido combina técnicas de similaridade semânticas e sintáticas.
- **Nearest-neighbor match:** Um serviço Web semântico  $S$  caíra nesse filtro se  $SynSim(Q, S) \geq \alpha$ . Este filtro é utilizado caso todos os filtros acima falhem.
- **Fail:** OWLS-MS retorna uma falha total se nenhum filtro acima satisfaz as condições entre o serviço Web semântico  $S$  e a consulta  $Q$ .

Esses serviços Web semânticos serão ordenados de acordo com o filtro que OWLS-MX atribuir, e a ordem de relevância é a seguinte: EXACT > PLUG-IN > SUBSUMES > SUBSUMED-BY > LOGICAL FAIL > HYBRID SUBSUMED-BY > NEAREST-NEIGHBOR > FAIL.

### 3.2.3 Filtragem de Registros de Serviços Web Semânticos

Esse método (KHDOUR; FASLI, 2010) propõe um filtro de serviços Web semânticos para uma consulta, diminuindo a quantidade necessária de tempo para calcular a similaridade entre serviços Web semânticos relevantes e a consulta. Entretanto, este mecanismo

<sup>3</sup>Alguns graus de similaridade aqui definidos também se encontram no trabalho de Wei et al(2008)

determina apenas se, a priori, um serviço Web semântico é relevante ou não para a consulta de uma forma binária. Após isso, outro algoritmo de similaridade será utilizado e, assim, provê um ordenamento dos serviços Web semânticos por ordem de relevância.

### 3.2.4 Similaridade baseada em ontologia

Liu et al (2009) apresenta um algoritmo para medir a similaridade semântica entre Serviços Web Semânticos analisando a relação entre os conceitos presentes na taxonomia de uma determinada ontologia. Esse algoritmo é baseado outro trabalho de Li, Bandar e McLean(2003) para calcular a similaridade semântica entre palavras utilizando a hierarquia de uma base semântica, a qual leva em consideração a maneira em que essas palavras estão localizadas nessa taxonomia.

Cada palavra nessa base hierárquica é a especificação de outra palavra, logo, duas palavras nessa base terão uma ou mais palavras mais genéricas em comum. Li, sugere que, além de usar a distância entre as duas palavras na hierarquia, deve-se utilizar a profundidade da palavra mais específica que ambas têm em comum. Isso significa que, dentre as palavras em comum que ambas possuem, deve-se utilizar aquela com uma maior profundidade na hierarquia. Outra coisa a ser levada em conta é a densidade semântica entre essas palavras. Assim, dado duas palavras  $w_1$  e  $w_2$ , sua similaridade semântica será calculada da seguinte forma:

$$s(w_1, w_2) = f(f_1(l), f_2(h), f_3(d)) \quad (1)$$

sendo  $l$  o caminho mais curto entre  $w_1$  e  $w_2$ ,  $h$  a profundidade da palavra em comum mais específica, e  $d$  a densidade semântica entre as palavras. Para calcular  $f_1(l)$ , a seguinte equação é utilizada:

$$f_1(l) = e^{-\alpha l}$$

sendo  $\alpha$  um fator de suavização. Sua forma exponencial assegura que o seu valor fique no intervalo  $[0,1]$ . Para  $f_2(h)$ :

$$f_2(h) = \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}$$

sendo  $\beta$  um outro fator de suavização, onde  $\beta > 0$ , dado que à medida que  $\beta \rightarrow \infty$ , o parâmetro de profundidade não é mais considerado para medir a similaridade entre as palavras.

Para medir a similaridade semântica entre duas palavras  $w_1$  e  $w_w$ , é necessário a presença de um corpus<sup>4</sup>, dado que cada palavra fornece um ganho de informação, e esse valor é baseado na probabilidade de se achar aquela palavra entre outras em um determinado corpus (baseado na frequência dessa palavra no corpus). Logo, a densidade semântica é calculada da seguinte forma:

$$d = \max_{c \in \text{sub}(w_1, w_2)} [-\log p(c)]$$

onde  $\text{sub}(w_1, w_2)$  é o conjunto de palavras que são generalizações comuns de  $w_1$  e  $w_2$ ,  $d$  é o valor máximo que se pode obter a partir do ganho de informação em cada uma das palavras genéricas que ambas possuem em comum. Assim sendo, para calcular  $f_3(d)$ :

$$f_3(d) = \frac{e^{\lambda d} - e^{-\lambda d}}{e^{\lambda d} + e^{-\lambda d}}$$

sendo  $\lambda$  um fator de suavização semelhante à  $\beta$ .

<sup>4</sup>Corpus consiste em um conjunto de textos escritos ou falados em uma língua que serve como base de análise

### **3.2.5 Análise das Abordagens**

A abordagem de (KLUSCH; FRIES; SYCARA, 2009) utiliza uma técnica híbrida, dado que ela utiliza informações semânticas e sintáticas. Essa técnica usa o mesmo mecanismo de (WEI et al., 2008) para computar a similaridade entre conceitos. Entretanto, seus algoritmos não utilizam informações presentes nas pré-condições e efeitos dos serviços Web semânticos, diferente do trabalho apresentado aqui nesta dissertação.

O método de (KHDOUR; FASLI, 2010) não passa de um filtro, ele não faz de fato o cálculo de similaridade entre dois serviços Web semânticos. O trabalho de (WEI et al., 2008) baseia-se muito na descrição textual em linguagem natural dos serviços Web Semânticos, o que é tem grandes desvantagens dado que essas descrições não são obrigatórias e dependentes de idioma.

## **3.3 Resumo do Capítulo**

Este capítulo apresentou os principais trabalhos encontrados na literatura relacionados ao trabalho apresentado por esta dissertação. Os trabalhos encontrados foram divididos em dois grupos, métricas entre serviços Web Tradicionais e métricas entre serviços Web semânticos. Para cada trabalho encontrado foi feita uma breve análise.

## 4 TRABALHO DESENVOLVIDO

Liu et al (2009) apresenta um algoritmo para medir a similaridade semântica entre Serviços Web Semânticos analisando a relação entre os conceitos presentes na taxonomia de uma determinada ontologia. Esse algoritmo é baseado outro trabalho de Li, Bandar e McLean(2003) para calcular a similaridade semântica entre palavras utilizando a hierarquia de uma base semântica, a qual leva em consideração a maneira em que essas palavras estão localizadas nessa taxonomia. Um exemplo de hierarquia de uma base semântica de palavras é descrita na Figura 4.1, fonte: (LI; BANDAR; MCLEAN, 2003).

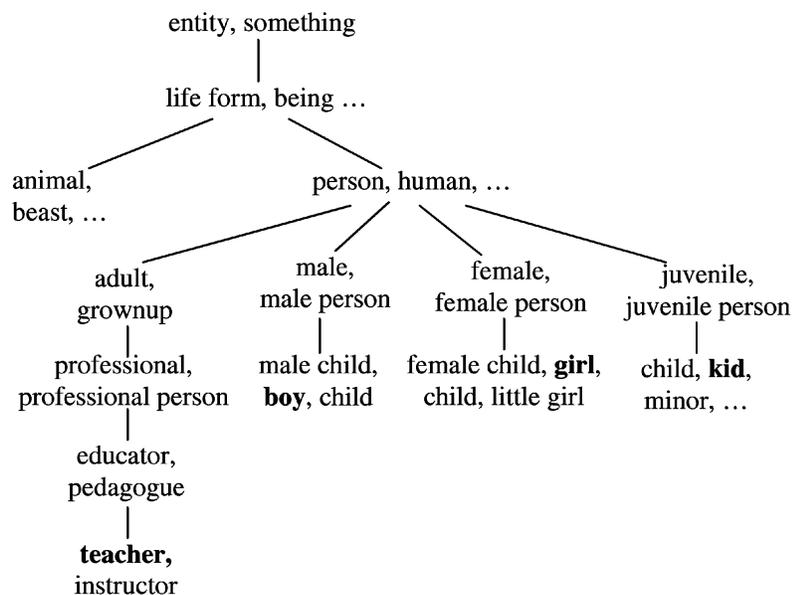


Figura 4.1: Base semântica hierárquica de palavras

Uma maneira intuitiva de calcular a similaridade entre duas palavras consiste em avaliar a distância entre elas dentro dessa hierarquia. Por exemplo, considerando a Figura 4.1, a palavra *boy* é mais similar à palavra *girl* do que a palavra *teacher*, dado que a distância entre *boy* e *girl*, *boy-male-person-female-girl*, é menor do que a distância entre *boy* e *teacher*, *boy-male-person-adult-professional-educator-teacher*.

Entretanto, essa maneira não aparenta ser a melhor forma de medir a similaridade semântica entre duas palavras. Um contraexemplo é que, através dessa metodologia, *boy* é mais similar a *animal* do que *teacher*. Logo, para medir a similaridade semântica entre duas palavras utilizando a hierarquia de uma base semântica de palavras, é preciso levar em conta algo além da distância entre as palavras. Algo a ser utilizado são as palavras que ambas têm em comum.

Cada palavra nessa base hierárquica é a especificação de outra palavra, logo, duas palavras nessa base terão uma ou mais palavras mais genéricas em comum. Li, sugere que, além de usar a distância entre as duas palavras na hierarquia, deve-se utilizar a profundidade da palavra mais específica que ambas têm em comum. Isso significa que, dentre as palavras em comum que ambas possuem, deve-se utilizar aquela com uma maior profundidade na hierarquia. Outra coisa a ser levada em conta é a densidade semântica entre essas palavras. Assim, dado duas palavras  $w_1$  e  $w_2$ , sua similaridade semântica será calculada da seguinte forma:

$$s(w_1, w_2) = f(f_1(l), f_2(h), f_3(d)) \quad (1)$$

sendo  $l$  o caminho mais curto entre  $w_1$  e  $w_2$ ,  $h$  a profundidade da palavra em comum mais específica, e  $d$  a densidade semântica entre as palavras. Para calcular  $f_1(l)$ , a seguinte equação é utilizada:

$$f_1(l) = e^{-\alpha l}$$

sendo  $\alpha$  um fator de suavização. Sua forma exponencial assegura que o seu valor fique no intervalo  $[0,1]$ . Para  $f_2(h)$ :

$$f_2(h) = \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}$$

sendo  $\beta$  um outro fator de suavização, onde  $\beta > 0$ , dado que à medida que  $\beta \rightarrow \infty$ , o parâmetro de profundidade não é mais considerado para medir a similaridade entre as palavras.

Para medir a similaridade semântica entre duas palavras  $w_1$  e  $w_w$ , é necessário a presença de um corpus<sup>1</sup>, dado que cada palavra fornece um ganho de informação, e esse valor é baseado na probabilidade de se achar aquela palavra entre outras em um determinado corpus (baseado na frequência dessa palavra no corpus). Logo, a densidade semântica é calculada da seguinte forma:

$$d = \max_{c \in \text{sub}(w_1, w_2)} [-\log p(c)]$$

onde  $\text{sub}(w_1, w_2)$  é o conjunto de palavras que são generalizações comuns de  $w_1$  e  $w_2$ ,  $d$  é o valor máximo que se pode obter a partir do ganho de informação em cada uma das palavras genéricas que ambas possuem em comum. Assim sendo, para calcular  $f_3(d)$ :

$$f_3(d) = \frac{e^{\lambda d} - e^{-\lambda d}}{e^{\lambda d} + e^{-\lambda d}}$$

sendo  $\lambda$  um fator de suavização semelhante à  $\beta$ .

---

<sup>1</sup>Corpus consiste em um conjunto de textos escritos ou falados em uma língua que serve como base de análise

## 4.1 Similaridade entre Serviços Web Semânticos

A ideia de utilizar anotações semânticas para descrever as interfaces de serviços Web vem do trabalho de (MCILRAITH; SON; ZENG, 2001). Seu principal objetivo é permitir que agentes de software sejam capazes de automatizar o processo de descoberta, composição e invocação de serviços Web. O algoritmo apresentado por Liu et al usa quatro categorias presentes no perfil de um serviço Web semântico: Pré-condições, Entradas, Saídas, e Efeitos.

Cada categoria é considerada como um conjunto de conceitos, onde, dadas duas categorias, Liu et al, definem que a similaridade de tais categorias é calculada da seguinte forma:

$$D_s(C_1, C_2) = \sum_{c_1 \in C_1, c_2 \in C_2} ws(c_1, c_2) \quad (2)$$

sendo  $c_1$  um conceito da categoria  $C_1$ ,  $c_2$  um conceito da categoria  $C_2$ , e  $w$  o peso para o  $i$ -ésimo par de conceitos. Essa equação utiliza todos os possíveis pares de conceitos entre as duas categorias, e se, em algum caso, a similaridade calculada por (1) for igual a zero, então o peso  $w$  para esse par também será igual a zero. Além disso,  $\sum w = 1$ . Então, dados dois serviços Web semânticos, sua similaridade é medida a partir da seguinte equação:

$$S = \sum_i W_i D_s(C_{i1}, C_{i2})$$

sendo  $W_i$  o peso para o par da  $i$ -ésima categoria, onde  $\sum W_i = 1$ . Cada par de categorias tem que ser uma das quatro categorias presentes no perfil de um serviço Web semântico: Entradas, Saídas, Pré-condições e Efeitos. Diferentemente dos pares de conceitos, aqui as categorias fazem par somente com as categorias do seu próprio tipo.

### 4.1.1 Limiar

Liu et al (2009) não informam qual limiar utilizar para definir se dois serviços Web semânticos serão similares ou não. Esta dissertação sugere que seja utilizado um limiar de 0,75 para definir quando há similaridade entre dois serviços Web semânticos, e os resultados mostram que essa sugestão aparenta ser adequada.

## 4.2 Limitações do Algoritmo de Liu

No cálculo da similaridade entre as categorias, o algoritmo de Liu faz um produto cartesiano entre elas (que são vistas como um conjunto de conceitos), calcula a similaridade entre cada par de conceitos e faz uma média levando em conta todas as similaridades calculadas entre esses pares. O problema é que, dado dois conjuntos equivalentes, o mapeamento é 1:1 (1 para 1). Então, mesmo tendo duas categorias similares (com um grande número de conceitos equivalentes), esse algoritmo prejudica o cálculo de sua similaridade, dado que os pares não equivalentes diminuirão o valor da média.

Experimentos realizados nessa dissertação e em trabalhos anteriores (MAAMAR et al., 2011) (MAAMAR et al., 2011) indicam que esse algoritmo não alcança resultados satisfatórios, tendo na maioria dos casos uma alta revocação e uma baixa precisão e *f-measure*. Ou seja, esse algoritmo traz muitos serviços irrelevantes na lista de serviços recuperados.

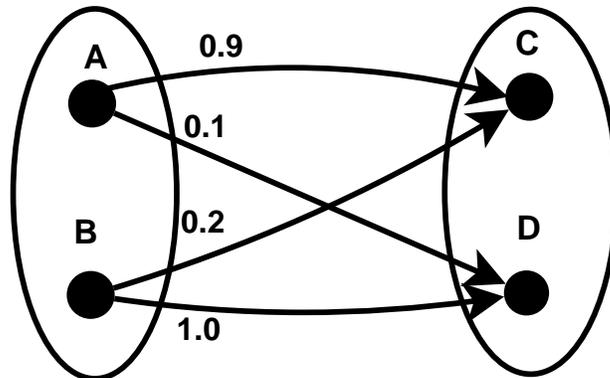


Figura 4.2: Pareamento entre conceitos de duas categorias

### 4.3 Hipótese Levantada

Cada serviço Web semântico possui, em sua assinatura, 4 categorias: Entradas, Saídas, Pré-Condições e Efeitos. Para calcular a similaridade entre Serviços Web Semânticos, Liu et al(2009) calculam a similaridade entre tais categorias. Por exemplo, a categoria Entrada do Serviço Web Semântico 1 é pareada com a categoria Entrada do Serviço Web Semântico 2, a categoria Saída do Serviço Web Semântico 1 é pareado com a categoria Saída do Serviço Web Semântico 2, e assim por diante.

Essas categorias, por sua vez, são nada mais do que conjuntos de conceitos. O cálculo da similaridade pode ser feito, portanto, através do cálculo da similaridade entre os conceitos presentes nessas categorias. Assim, para cada conceito presente em uma categoria, sua similaridade é calculada com cada conceito presente em outra categoria, e a partir daí é realizada uma média ponderada dessas similaridades. O valor final é dado como a similaridade entre as categorias.

Isso pode ser visto como um grafo bipartido, onde os nós são os conceitos e os dois conjuntos disjuntos são as duas categorias. Um exemplo dessa visualização pode ser observado na Figura 4.2. Percebe-se claramente que o valor da similaridade entre as categorias será diminuído por causa da similaridade do conceito A com o conceito D e da similaridade do conceito B com o conceito C.

Este trabalho levanta uma hipótese para que o cálculo de similaridade entre duas categorias não seja tão prejudicado como é ilustrado na Figura 4.2. Cada conceito de uma categoria terá sua similaridade calculada com cada conceito da outra categoria. Porém, apenas o maior valor será levado em consideração ao fazer a média ponderada. Isso prejudicaria menos o resultado final ao calcular a similaridade entre duas categorias, aperfeiçoando o algoritmo de Liu et al(2009) para calcular a similaridade semântica entre dois Serviços Web Semânticos.

Nesse caso, utilizando a hipótese dessa dissertação, tanto a similaridade entre os conceitos A e D, quanto a similaridade entre os conceitos B e C, não seriam levada em con-

sideração, não prejudicando o resultado final da similaridade entre as categorias.

#### 4.4 Aperfeiçoamento do Algoritmo

O problema do algoritmo proposto por Liu et al(2009) é mostrado na Figura 4.3. Assumindo que os conceitos  $A$  e  $C$  sejam de uma categoria, e os conceitos  $B$  e  $D$  sejam de outra, o algoritmo de Liu et al(2009) calcula as similaridades entre os conceitos e os usa como um saco de conceitos para atribuir a similaridade entre as categorias.

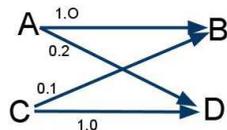


Figura 4.3: Pareamento de conceitos

No caso ilustrado na Figura 4.3, percebe-se claramente que o valor da similaridade será prejudicado por causa da baixa similaridade do conceito  $A$  com o conceito  $B$ , e do conceito  $B$  com o conceito  $C$ . O valor da similaridade entre as categorias deveria ser 1 nesse caso, mas o algoritmo de Liu et al(2009) não atribuirá esse valor.

Assim sendo, o aperfeiçoamento aqui apresentado considera apenas a maior similaridade entre cada par de conceitos de uma categoria com os conceitos da outra categoria, ao invés de utilizar todos os valores de similaridades obtidos entre cada possível par de conceitos. Em outras palavras, para cada conceito apresentado em uma categoria, esse conceito fará par com um conceito da outra categoria, tendo cada par formado um grau de similaridade, e somente o maior grau de similaridade será levado em consideração.

Logo, ao invés de utilizar (2) para calcular a similaridade entre as categorias, essa métrica será aperfeiçoada da seguinte maneira:

$$D_s(C_1, C_2) = \sum_{c_1 \in C_1} w_{c_1} \max_{c_2 \in C_2} (s(c_1, c_2))$$

Os experimentos realizados no trabalho apresentado nesta dissertação mostram que a modificação melhora a precisão para algumas configurações (mais que o dobro em duas configurações), dado que o algoritmo de Liu et al(2009) traz muito falsos positivos, tendo altas taxas de revocação e baixas taxas de precisão. Embora esta abordagem tenha uma revocação um pouco menor, a taxa de revocação é aceitável. Esses detalhes serão mostrados no capítulo de experimentos.

#### 4.5 Definição de Densidade Semântica

Quando Li, Bandar e McLean(2003) mencionam a densidade semântica de duas palavras em um corpus, eles definem densidade semântica da seguinte forma:

$$d = \max_{c \in \text{sub}(w_1, w_2)} [-\log p(c)]$$

onde  $\text{sub}(w_1, w_2)$  é o conjunto de palavras que são generalizações comuns de  $w_1$  e  $w_2$ , e  $d$  é o valor máximo que se pode obter a partir do ganho de informação em cada uma das palavras genéricas que ambas possuem em comum.

Quanto menor for a probabilidade de se encontrar uma determinada palavra no corpus, maior será o ganho de informação que essa palavra fornecerá. Por exemplo, uma

conjunção é um tipo de palavra que possui uma alta frequência dentro de qualquer texto (lembrando que a probabilidade nesse caso é obtida a partir da frequência relativa), logo, as conjunções possuem uma alta probabilidade de serem encontradas. Logo, uma conjunção dificilmente irá agregar alguma informação relevante. Isso é uma das razões pela qual as conjunções são removidas durante o pré-processamento de textos.

No caso de serviços Web semânticos, Liu et al(2009) não explicam explicitamente como calcular a densidade semântica dos conceitos envolvidos nas categorias de cada serviço Web semântico. Esta seção explicita como fazer tal cálculo.

Para calcular a densidade semântica entre dois conceitos presentes na assinatura de dois serviços Web semânticos, é necessário possuir a taxonomia da ontologia que envolve esses conceitos, uma taxonomia como a apresentada na Figura 4.4<sup>2</sup>. Para ilustração, supõe-se que se queira calcular a densidade semântica entre os conceitos *Magazine* e *Encyclopedia*. Para isso, recupera-se todos os superconceitos que esses dois conceitos possuem em comum, que no caso são *Publication* e *PrintedMaterial*.

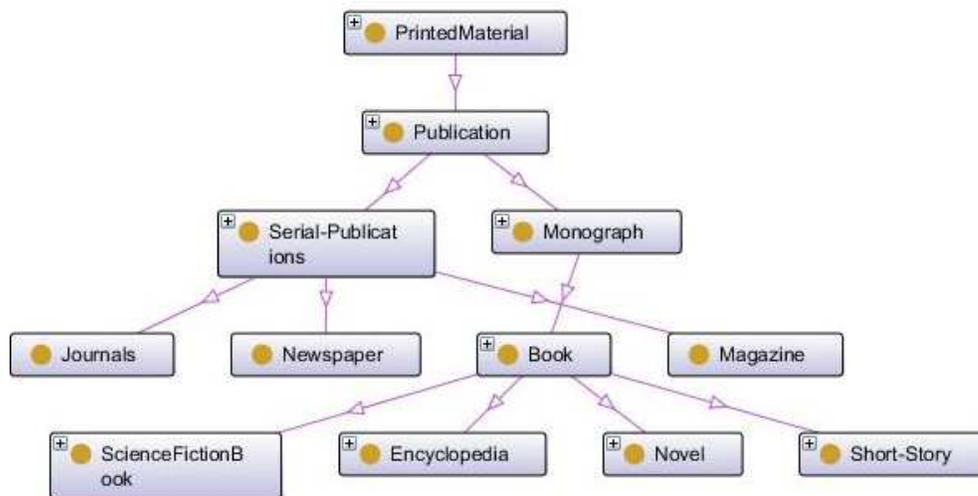


Figura 4.4: Taxonomia de uma Ontologia

A questão agora é como calcular a densidade semântica de um conceito. No caso de Serviços Web Semânticos, esta dissertação apresenta a seguinte maneira de calcular a densidade semântica de um conceito. Primeiramente, como aqui não se trata de textos, não existe a presença de um corpus.

Porém, existe uma base de serviços Web semânticos, e cada serviço Web semântico possui uma assinatura. Em sua assinatura, existem conceitos e para cada conceito existente na ontologia, calcula-se a probabilidade de se achar esse conceito na assinatura de um serviço Web semântico. Essa probabilidade é calculada através da seguinte equação:

$$p(c) = \frac{N_c}{N}$$

sendo  $N_c$  o número de Serviços Web Semânticos que possuem o conceito  $c$  em sua assinatura, e  $N$  o número total de Serviços Web Semânticos indexados na coleção. Assim sendo, o ganho de informação que um conceito fornecerá será:

<sup>2</sup>Essa imagem corresponde a uma parte da taxonomia da ontologia book.owl presente na coleção OWLS-TC 4

$$ganho\_de\_informacao = -\log \frac{N_c}{N}$$

assim, a densidade semântica entre dois conceitos será

$$d = \max_{c \in super(c_1, c_2)} -\log \frac{N_c}{N}$$

sendo  $super(c_1, c_2)$  o conjunto de superconceitos em comum que os conceitos  $c_1$  e  $c_2$  possuem.

No exemplo dado para calcular a densidade semântica entre os conceitos *Magazine* e *Encyclopedia*, supõe-se que a Quadro 4.1 seja uma lista de serviços Web semânticos indexados. O ganho de informação de *PrintedMaterial* é 0,602 enquanto o ganho de informação de *Publication* é 0,301. Logo, a densidade semântica entre os conceitos *Magazine* e *Encyclopedia* será 0,602.

Quadro 4.1: Lista de serviços Web semânticos Indexados

Serviço	Entrada	Saída	Pré-condições	Efeitos
Serviço 1	<i>PrintedMaterial</i>	<i>Book</i>	-	-
Serviço 2	<i>Magazine</i>	<i>Magazine</i>	<i>Magazine</i>	-
Serviço 3	<i>Publication</i>	<i>Magazine</i>	-	Magazine
Serviço 4	<i>Newspaper</i>	<i>Publication</i>	-	-

## 4.6 Resumo do Capítulo

Esse capítulo apresentou o algoritmo de Liu et al(2009) e seu aperfeiçoamento. Aqui é identificado o ponto-chave que prejudica o desempenho do algoritmo, que é, no caso, a maneira como a similaridade entre as categorias é calculada. Um limiar é definido para saber quando um serviço Web semântico é similar a outro, dado que Liu et al(2009) não apresentam esse valor em seu trabalho. Além disso, é definido o conceito de densidade semântica no contexto de conceitos utilizados em assinatura de serviços Web semânticos.

## 5 EXPERIMENTOS

O presente capítulo apresenta os experimentos realizados nessa dissertação. O objetivo desses experimentos é o de verificar se a hipótese dessa dissertação de fato melhora o desempenho do algoritmo de Liu. Para analisar se há ou não uma melhoria do algoritmo, foram implementadas duas versões, uma de acordo com o artigo original de Liu e outra de acordo com a hipótese deste trabalho.

Lembrando que a hipótese levantada trata de não considerar todos os valores calculados entre os pares de conceitos no cálculo da média entre duas categorias. A hipótese é de que, para cada conceito da primeira categoria, calcule-se a similaridade desse conceito com cada conceito da outra categoria, e levar em consideração apenas o maior valor calculado.

Três métricas de desempenho foram usadas: precisão, revocação e *f-measure*. Para cada consulta, foram utilizadas essas três métricas. Para cada par (implementação, configuração), foi calculada a média das precisões, revocações e *f-measure*. Em seguida, foi utilizado o teste estatístico ANOVA para saber se havia uma diferença significativa entre os dados, pois, havendo uma média maior e uma diferença significativa entre os dados, conclui-se que há uma melhoria naquela métrica de desempenho.

A coleção OWLS-TC4 (OWL-S Test Collection 4) foi usada para realizar os experimentos. Ela está disponível para download em <http://www.semWebcentral.org/projects/owlstc/>. Este conjunto de dados é composto por 1083 serviços descritos em OWL-S 1.1 de nove domínios diferentes: educação, saúde, gastronomia, viagem, comunicação, economia, armamento bélico, geografia e simulação.

Parte desses serviços Web foram obtidos a partir de UBR (*Universal Business Records* - Registros de Negócios Universais), os quais estão extintos<sup>1</sup>. Existem 42 consultas, e um arquivo em XML nomeado *owls-tc4.xml* que define quais são os serviços relevantes para cada consulta. Uma consulta é descrita como um serviço, através de suas entradas, saídas, pré-condições e efeitos.

Existem dois tipos de relevância atribuídos às consultas: relevância binária e relevância classificada. A estrutura do arquivo XML pode ser vista nas Listagens 5.1 e 5.2. A relevância binária informa apenas se o serviço Web semântico é ou não é relevante para a consulta. A relevância classificada informa quão relevante aquele serviço Web semântico é para a consulta. Os experimentos realizados neste trabalho utilizaram o conjunto de relevância binária.

### Listagem 5.1: Estrutura do arquivo XML para relevância binária

```
1 <binaryrelevanceset> // para o conjunto de relevancia binaria
2   <request ...>
```

<sup>1</sup><http://soa.sys-con.com/node/164624/>

```

3         <name ... />
4         <uri .../>
5         <ratings>
6             <offer ...> //cada requisicao (consulta)
                oferece varios resultados
7                 <name ... />
8                 <uri .../>
9                 <relevant>valor</relevant> // aqui o
                    valor pode ser 0 ou 1
10            </offer>
11            ...
12        </ratings>
13    </request>
14    ...
15 </binaryrelevanceset>

```

---

### Listagem 5.2: Estrutura do arquivo XML para relevância classificada

---

```

1 <relevancesgrades> // para o conjunto de relevancia classificada
2   <services>
3     <request ...>
4       <name ... />
5       <uri .../>
6       <ratings>
7         <offer ...>
8           <name ... />
9           <uri .../>
10          <grade ...>
11            <name>NomeDaClassificacao</name> // highly
                relevant (valor 3); relevant (valor 2);
12            <value>ValorDaClassificacao</value> //
                potentially relevant (valor 1); nonrelevant
                (valor 0)
13          </grade>
14        </offer>
15        ...
16      </ratings>
17    </request>
18    ...
19  </services>
20 </relevancesgrades>

```

---

As ontologias que descrevem os conceitos presentes nas entradas e saídas dos serviços Web foram obtidas de fontes públicas na Web. Mais informação acerca da base dados encontram-se no manual que vem junto com a base de dados.

A implementação do algoritmo foi feita utilizando a API Jena<sup>2</sup> para leitura das ontologias em OWL, e a API OWL-S<sup>3</sup> para leitura das interfaces de Serviços Web Semânticos. Embora existam outros conjuntos de dados disponíveis, OWLS-TC foi escolhido dado que atualmente OWL-S é atualmente a linguagem mais utilizada na Web para descrição de serviços web semânticos (KLUSCH; ZHING, 2008). As pré-condições e efeitos estão em SWRL.

<sup>2</sup>disponível para download em <http://jena.sourceforge.net/>

<sup>3</sup>disponível para download em <http://on.cs.unibas.ch/owls-api/>

Quadro 5.1: Consultas

Índice	Nome da Consulta
1	1personbicyclecar_price_service.owl
2	book_price_service.owl
3	bookpersoncreditcardaccount_service.owl
4	bookpersoncreditcardaccount_price_service.owl
5	car_price_service.owl
6	citycountry_hotel_service.owl
7	country_skilledoccupation_service.owl
8	dvdplayermp3player_price_service.owl
9	EBookOrder1.owl
10	fall_down_pill.owl
11	geocodeUSAAddress.owl
12	geographical-regiongeographical-region_map_service.owl
13	geopolitical-entity_weatherprocess_service.owl
14	getAltitudeAboveSeaLevelOfLocation.owl
15	getDistanceBetweenCitiesWorldwide.owl
16	getLocationOfCityState.owl
17	getLocationOfUSCity.owl
18	getLocationOfUSZipcode.owl
19	getMapOfUSAAddress.owl
20	getSunsetSunriseTimeOfLocation.owl
21	getZipcodeForUSCity.owl
22	governmentdegree_scholarship_service.owl
23	governmentmissile_funding_service.owl
24	grocerystore_food_service.owl
25	hospital_investigating_service.owl
26	lock_door.owl
27	maxprice_cola_service.owl
28	mileToKilometerConverter.owl
29	novel_author_service.owl
30	open_door.owl
31	preparedfood_price_service.owl
32	publication-number_publication_service.owl
33	recommendedprice_coffeewhiskey_service.owl
34	researcher-in-academia_address_service.owl
35	shoppingmall_cameraprice_service.owl
36	surfing_destination_service.owl
37	surfinghiking_destination_service.owl
38	surfingorganization_destination_service.owl
39	title_comedyfilm_service.owl
40	title_videomedia_service.owl
41	university_lecturer-in-academia_service.owl
42	userscience-fiction-novel_price_service.owl

Entretanto, de acordo com (KLUSCH; FRIES; SYCARA, 2009), a maioria dos serviços Web semânticos disponíveis em OWL-S não especificam pré-condições e efeitos. Então, essas duas categorias foram desconsideradas nos experimentos aqui realizados. Os experimentos foram executados em duas máquinas, uma com Sistema Operacional Windows 7 Home Basic, com um processador Intel(R) Core(TM) i3 2,53GHz com 4,00 GB de memória, e outra com Sistema Operacional Windows XP, processador Pentium 4 3,4 GHz e 2GB de memória. Três medidas de desempenho de recuperação de informação foram utilizadas:

$$precisao = \frac{|{\text{documentos\_relevantes}} \cap {\text{documentos\_recuperados}}|}{|{\text{documentos\_recuperados}}|}$$

$$revocacao = \frac{|{\text{documentos\_relevantes}} \cap {\text{documentos\_recuperados}}|}{|{\text{documentos\_relevantes}}|}$$

$$f - measure = 2 * \frac{precisao * revocacao}{precisao + revocacao}$$

Implementação A corresponde à implementação do algoritmo apresentado neste trabalho, e a implementação B foi feita de acordo com o trabalho de Liu et al (2009). Liu

et al (2009) não apresentaram em seu trabalho publicado qual era a melhor configuração para os parâmetros envolvidos no cálculo da similaridade, sem falar que não há nenhum experimento realizado que ilustre o desempenho de sua abordagem.

Li, Bandar e McLean (2003) apresentam três configurações relevantes para medir a similaridade entre palavras: considerar apenas o tamanho da distância entre as palavras e a profundidade da palavra mais específica que ambas possuem em comum, com  $\alpha = 0, 2$  e  $\beta = 0,6$  ( $f_1, f_2$ ); considerar apenas o tamanho da distância entre as palavras, com  $\alpha = 0, 25$  ( $f_1$ ); e considerar o tamanho da distância entre as palavras, a profundidade da palavra mais específica que ambas possuem em comum, e a densidade semântica entre elas, com  $\alpha = 0, 25$ ,  $\beta = 0, 6$  e  $\lambda = 0, 4$  ( $f_1, f_2, f_3$ ).

## 5.1 ANOVA

Esta seção explica como funciona o ANOVA. O ANOVA (*Analysis of Variance* - Análise de Variância) consiste em um teste estatístico para verificar se há diferença significativa entre as médias de  $g$  grupos distintos (FISHER, 1921; ARMSTRONG; SLADE; EPERJESI, 2000). Dado  $g$  grupos, cada grupo com  $n$  amostras, calcula-se as somas dos quadrados entre grupos e dentro dos grupos.  $SQE$  é a soma de quadrados entre grupos, e ela é calculada através da seguinte maneira:

$$SQE = n * (\bar{x}_1 - \bar{x})^2 + \dots + n * (\bar{x}_g - \bar{x})^2$$

sendo  $\bar{x}_i$  a média do grupo  $i$ , e  $\bar{x}$  a média entre todas as amostras. A soma de quadrados dentro dos grupos é  $SQD$ , e calculada da seguinte maneira:

$$SQD = \sum_{grupo1}^n (x_i - \bar{x}_1)^2 + \dots + \sum_{grupog}^n (x_i - \bar{x}_g)^2$$

A média dos quadrados é calculada também, sendo  $MQE$  a média dos quadrados entre os grupos:

$$MQE = \frac{SQE}{g-1}$$

e a média dos quadrados dentro dos grupos é:

$$MQD = \frac{SQD}{N-g}$$

onde  $N$  é o número total de amostras. Assim, a estatística de teste  $F$  será:

$$F = \frac{MQE}{MQD}$$

Se  $F$  for maior que  $F$ -crítico, então pode-se afirmar que existe diferença significativa ao nível de 5% entre as médias dos grupos.

## 5.2 Primeira Configuração, $\alpha = 0,2, \beta = 0,6$

O quadro 5.2 apresenta os resultados para a primeira configuração: a que considera o tamanho da distância entre os conceitos na taxonomia da ontologia e profundidade do superconceito comum mais específico dentro da taxonomia, utilizando  $\alpha = 0,2$  e  $\beta = 0,6$ . A Figura 5.4 ilustra como a função exponencial com  $\alpha = 0,2$  se comporta. A Figura 5.5 mostra como a função monotônica com  $\beta = 0,6$  se comporta.

Percebe-se a partir dos gráficos, que quanto maior a distância menor é a similaridade, e quanto maior a profundidade, maior a similaridade entre os conceitos. Nesse experimento, a densidade semântica não foi levada em consideração. A função exponencial é utilizada para a distância pela sua natureza decrescente.

O gráfico na Figura 5.4 mostra que quando a distância é maior ou igual a 11, a similaridade é nula. O gráfico na Figura 5.5 mostra que quando a profundidade é zero, a similaridade é nula. A similaridade ganha força para um valor maior ou igual a 2.

Os resultados mostram que a implementação A tem uma precisão muito melhor que a implementação B, 85,26% contra 34,96%. O Quadro 5.3 aponta que existe uma diferença significativa entre os dados, o que reforça o argumento de que a implementação de acordo com o trabalho proposto nesta dissertação é mais precisa que a implementação de Liu et al (2009) para essa configuração.

A média da revocação da implementação A é um pouco menor que a média da implementação B, 69,41% contra 71,72%. No entanto, o Quadro 5.4 mostra que não existe uma diferença significativa entre os dados, o que significa que não se pode afirmar qual implementação apresenta uma melhor revocação. Além disso, isso não seria um problema tão grande dado que o objetivo é reduzir o número de falsos positivos.

Já para a *f-measure*, novamente, a implementação desta dissertação possui uma média muito superior à da implementação B, 71,28% contra 41,72%. O Quadro 5.5 mostra que existe uma diferença significativa entre os dados, o que, mais uma vez, reforça o argumento da melhoria que esta dissertação fornece para essa configuração dos parâmetros.

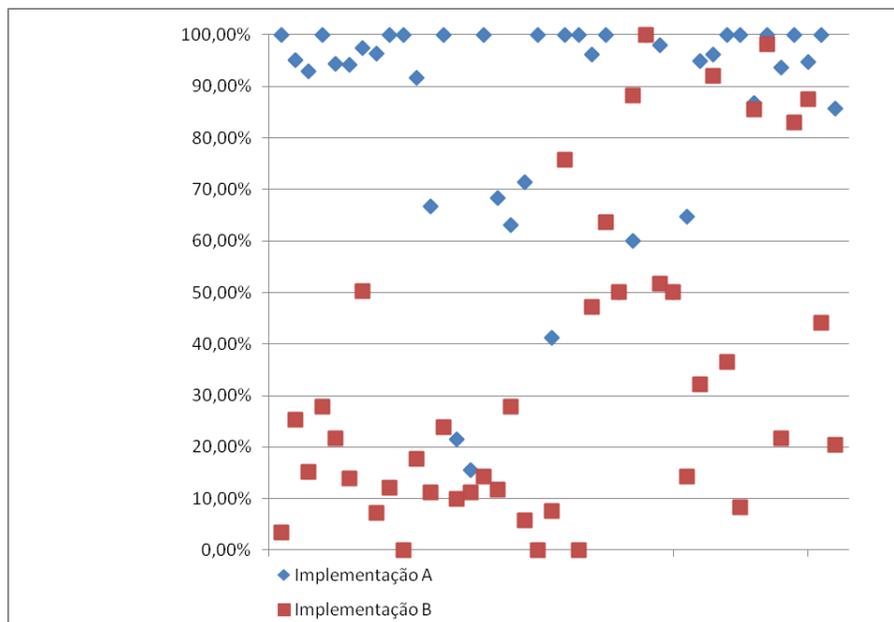


Figura 5.1: Gráfico da precisão para  $\alpha = 0,2, \beta = 0,6$

Quadro 5.2: Experimentos -  $\alpha = 0, 2, \beta = 0, 6$ .

consulta	Implementação A			Implementação B		
	precisão	revocação	f-measure	precisão	revocação	f-measure
1	100,00%	42,39%	59,54%	3,43%	7,61%	4,73%
2	95,08%	73,42%	82,86%	25,24%	100,00%	40,31%
3	92,86%	50,00%	65,00%	15,20%	100,00%	26,40%
4	100,00%	30,53%	46,77%	27,86%	100,00%	43,58%
5	94,44%	91,40%	92,90%	21,76%	61,29%	32,11%
6	94,29%	84,62%	89,19%	13,90%	66,67%	23,01%
7	97,53%	87,78%	92,40%	50,31%	91,11%	64,82%
8	96,30%	96,30%	96,30%	7,29%	92,59%	13,51%
9	100,00%	20,00%	33,33%	12,04%	86,67%	21,14%
10	100,00%	100,00%	100,00%	0,00%	0,00%	0,00%
11	91,67%	47,83%	62,86%	17,65%	26,09%	21,05%
12	66,67%	80,00%	72,73%	11,28%	100,00%	20,27%
13	100,00%	89,19%	94,29%	23,87%	100,00%	38,54%
14	21,43%	100,00%	35,29%	10,00%	100,00%	18,18%
15	15,63%	26,32%	19,61%	11,11%	42,11%	17,58%
16	100,00%	16,67%	28,57%	14,29%	27,78%	18,87%
17	68,42%	61,90%	65,00%	11,76%	19,05%	14,55%
18	63,16%	60,00%	61,54%	27,78%	50,00%	35,71%
19	71,43%	31,25%	43,48%	5,77%	18,75%	8,82%
20	100,00%	100,00%	100,00%	0,00%	0,00%	0,00%
21	41,18%	70,00%	51,85%	7,55%	40,00%	12,70%
22	100,00%	38,46%	55,56%	75,76%	96,15%	84,75%
23	100,00%	72,50%	84,06%	0,00%	0,00%	0,00%
24	96,15%	83,33%	89,29%	47,17%	83,33%	60,24%
25	100,00%	92,50%	96,10%	63,64%	87,50%	73,68%
26	50,00%	100,00%	66,67%	50,00%	100,00%	66,67%
27	60,00%	83,33%	69,77%	88,24%	83,33%	85,71%
28	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
29	98,04%	72,46%	83,33%	51,79%	84,06%	64,09%
30	50,00%	100,00%	66,67%	50,00%	100,00%	66,67%
31	64,71%	91,67%	75,86%	14,20%	95,83%	24,73%
32	95,00%	100,00%	97,44%	32,20%	100,00%	48,72%
33	96,15%	78,13%	86,21%	92,00%	71,88%	80,70%
34	100,00%	95,83%	97,87%	36,51%	95,83%	52,87%
35	100,00%	24,00%	38,71%	8,39%	100,00%	15,48%
36	86,79%	92,00%	89,32%	85,45%	94,00%	89,52%
37	100,00%	94,64%	97,25%	98,21%	98,21%	98,21%
38	93,75%	30,61%	46,15%	21,68%	100,00%	35,64%
39	100,00%	67,31%	80,46%	82,98%	75,00%	78,79%
40	94,74%	36,73%	52,94%	87,50%	42,86%	57,53%
41	100,00%	76,19%	86,49%	44,19%	90,48%	59,38%
42	85,71%	26,09%	40,00%	20,35%	84,06%	32,77%
Média	85,26%	69,41%	71,28%	34,96%	71,72%	41,72%

Quadro 5.3: Experimentos - ANOVA - Precisão,  $\alpha = 0, 2, \beta = 0, 6$ 

Fonte da Variação	SQ	gl	MQ	F	valor-P	F crítico
Entre grupos	5,314125636	1	5,314125636	73,16143927	$5,5664 * 10^{-13}$	3,957388177
Dentro dos grupos	5,95611987	82	0,072635608			
Total	11,27024551	83				

Quadro 5.4: Experimentos - ANOVA - Revocação,  $\alpha = 0, 2, \beta = 0, 6$ 

Fonte da Variação	SQ	gl	MQ	F	valor-P	F crítico
Entre grupos	0,011169428	1	0,011169428	0,116347027	0,733903165	3,957388177
Dentro dos grupos	7,872079918	82	0,096000975			
Total	7,883249346	83				

### 5.3 Segunda Configuração, $\alpha = 0, 25$

Apesar da primeira configuração ter apresentado bons resultados, o Quadro 5.6 mostra que a segunda configuração não teve o mesmo sucesso. Essa segunda configuração consi-

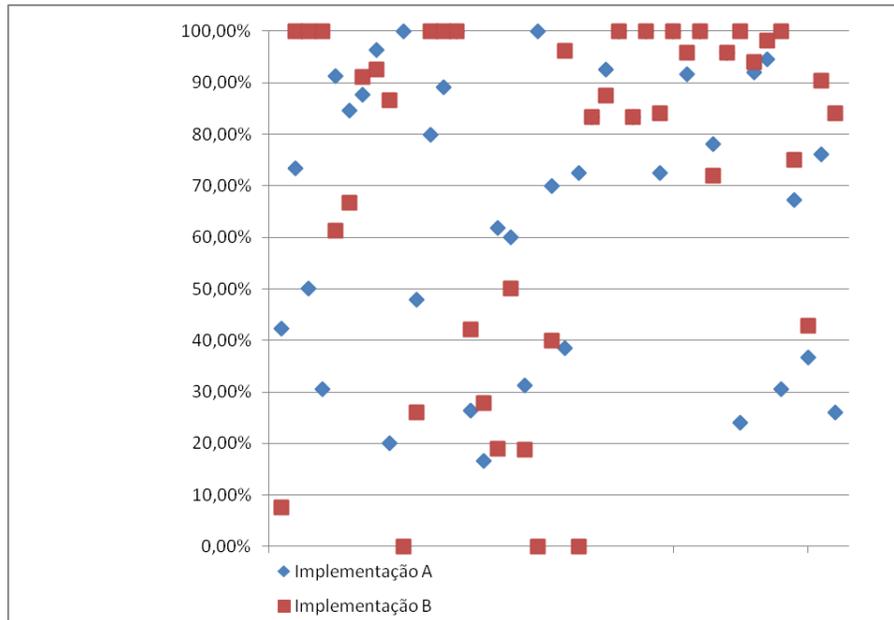


Figura 5.2: Gráfico da revocação para  $\alpha = 0, 2\beta = 0, 6$

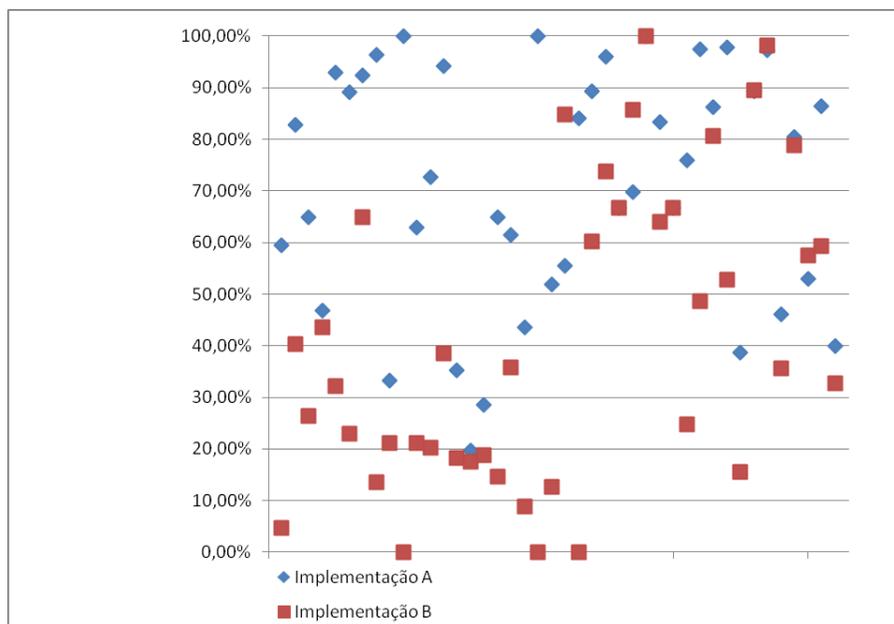


Figura 5.3: Gráfico da  $f$ -measure para  $\alpha = 0, 2\beta = 0, 6$

Quadro 5.5: Experimentos - ANOVA -  $F$ -Measure,  $\alpha = 0, 2\beta = 0, 6$

Fonte da Variação	SQ	gl	MQ	F	valor-P	F crítico
Entre grupos	1,835193515	1	1,835193515	26,57160679	$1,72437 * 10^{-6}$	3,957388177
Dentro dos grupos	5,663408668	82	0,069065959			
Total	7,498602183	83				

dera apenas a distância entre os conceitos para calcular a sua similaridade, com  $\alpha = 0.25$ . A Figura 5.9 mostra como a função exponencial se comporta para esse valor. O gráfico na Figura 5.9 mostra que quando a distância é maior ou igual a 10, a similaridade é nula.

A Implementação A conseguiu obter uma média maior de precisão, revocação e  $f$ -

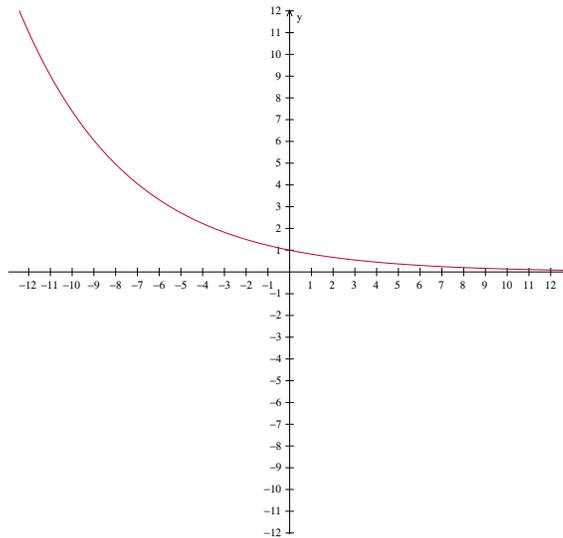


Figura 5.4: Função Exponencial, com  $\alpha = 0,2$

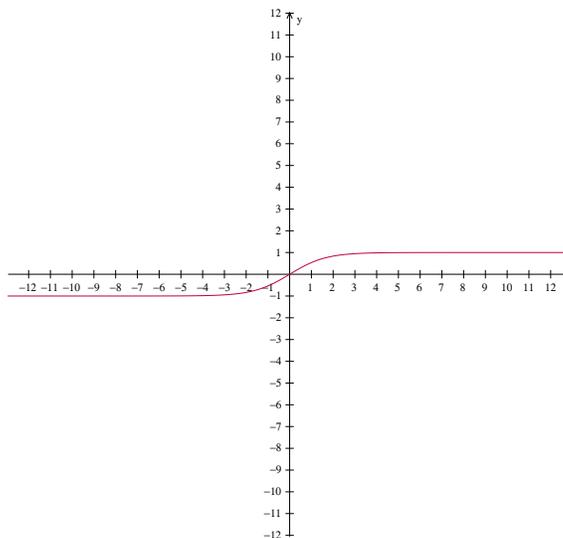


Figura 5.5: Função monotônica, com  $\beta = 0,6$

measure, porém como os Quadros 5.7, 5.8 e 5.9 mostram, apenas a revocação obteve uma diferença significativa entre os dados. Não se pode afirmar que há uma diferença significativa entre os dados de precisão e *f-measure*. Logo, só se pode afirmar que, para essa configuração, houve uma melhoria na revocação do algoritmo.

#### 5.4 Terceira Configuração, $\alpha = 0,2, \beta = 0,6, \lambda = 0,4$

O Quadro 5.10 aponta os resultados obtidos para a configuração que utiliza a densidade semântica. Os valores de  $\alpha$  e  $\beta$  são iguais ao da primeira configuração, o valor para  $\lambda$  é 0,4. O comportamento da função monotônica com  $\lambda$  igual a 0,4 é exibido na Figura 5.13.

O gráfico na Figura 5.5 mostra que quando a profundidade é zero, a similaridade é nula. A similaridade ganha força para um valor maior ou igual a 3. A implementação do trabalho apresentado nessa dissertação aponta uma precisão muito maior que a implementação de (LIU et al., 2009), e o Quadro 5.11 mostra que existe uma diferença significativa

Quadro 5.6: Experimentos -  $\alpha = 0,25$ .

consulta	Implementação A			Implementação B		
	precisão	revocação	f-measure	precisão	revocação	f-measure
1	6,70%	64,13%	12,14%	4,98%	50,00%	9,06%
2	8,44%	97,47%	15,54%	8,00%	100,00%	14,82%
3	2,74%	100,00%	5,34%	2,75%	100,00%	5,36%
4	9,30%	100,00%	17,01%	8,77%	100,00%	16,13%
5	9,50%	93,55%	17,24%	7,70%	79,57%	14,04%
6	3,59%	87,18%	6,90%	2,57%	66,67%	4,96%
7	8,41%	87,78%	15,35%	7,46%	84,44%	13,71%
8	2,73%	100,00%	5,31%	2,41%	96,30%	4,71%
9	1,41%	93,33%	2,78%	1,30%	93,33%	2,57%
10	0,19%	100,00%	0,38%	0,00%	0,00%	0,00%
11	3,00%	91,30%	5,80%	1,34%	43,48%	2,61%
12	1,45%	80,00%	2,85%	1,63%	100,00%	3,20%
13	4,04%	89,19%	7,73%	4,40%	97,30%	8,42%
14	0,38%	100,00%	0,75%	0,36%	100,00%	0,71%
15	1,87%	78,95%	3,65%	1,73%	73,68%	3,38%
16	2,28%	88,89%	4,45%	1,12%	50,00%	2,19%
17	2,71%	90,48%	5,27%	1,40%	47,62%	2,72%
18	2,24%	95,00%	4,38%	1,72%	70,00%	3,35%
19	1,75%	87,50%	3,42%	0,98%	50,00%	1,92%
20	0,52%	100,00%	1,03%	0,24%	50,00%	0,48%
21	0,99%	80,00%	1,96%	0,95%	80,00%	1,88%
22	6,53%	100,00%	12,26%	6,12%	100,00%	11,53%
23	3,93%	72,50%	7,46%	0,00%	0,00%	0%
24	3,36%	80,00%	6,45%	3,21%	80,00%	6,18%
25	5,15%	92,50%	9,75%	4,70%	87,50%	8,93%
26	0,20%	100,00%	0,40%	0,20%	100,00%	0,40%
27	1,82%	100,00%	3,58%	1,69%	100,00%	3,32%
28	0,10%	100,00%	0,20%	0,10%	100,00%	0,19%
29	7,48%	98,55%	13,91%	6,91%	98,55%	12,92%
30	0,20%	100,00%	0,40%	0,20%	100,00%	0,40%
31	5,39%	93,75%	10,19%	5,20%	95,83%	9,87%
32	1,94%	100,00%	3,82%	1,88%	100,00%	3,69%
33	3,05%	93,75%	5,90%	2,80%	93,75%	5,43%
34	2,80%	95,83%	5,44%	2,60%	95,83%	5,06%
35	2,36%	84,00%	4,59%	2,59%	100,00%	5,06%
36	5,05%	100,00%	9,62%	4,64%	100,00%	8,87%
37	5,56%	98,21%	10,52%	5,10%	98,21%	9,70%
38	4,98%	100,00%	9,50%	4,60%	100,00%	8,80%
39	4,15%	78,85%	7,88%	4,19%	86,54%	8,00%
40	4,06%	81,63%	7,74%	4,14%	89,80%	7,92%
41	1,43%	57,14%	2,79%	1,76%	71,43%	3,44%
42	7,57%	100,00%	14,07%	6,74%	100,00%	12,63%
Média	3,60%	91,23%	6,80%	3,12%	81,66%	5,92%

Quadro 5.7: Experimentos - ANOVA - Precisão,  $\alpha = 0,25$ 

Fonte da Variação	SQ	gl	MQ	F	valor-P	F crítico
Entre grupos	0,000483768	1	0,000483768	0,737112718	0,393090076	3,957388177
Dentro dos grupos	0,053816685	82	0,000656301			
Total	0,054300453	83				

Quadro 5.8: Experimentos - ANOVA - Revocação,  $\alpha = 0,25$ 

Fonte da Variação	SQ	gl	MQ	F	valor-P	F crítico
Entre grupos	0,192031437	1	0,192031437	4,942087182	0,028962251	3,957388177
Dentro dos grupos	3,186220168	82	0,038856344			
Total	3,378251606	83				

entre os dados, o que reforça a melhoria na precisão do algoritmo.

A revocação por sua vez foi maior utilizando o algoritmo de Liu et al (2009), porém, o Quadro 5.11 mostra que não existe uma diferença significativa entre os dados, logo não se pode afirmar que a revocação foi prejudicada, pois apesar da média da revocação da

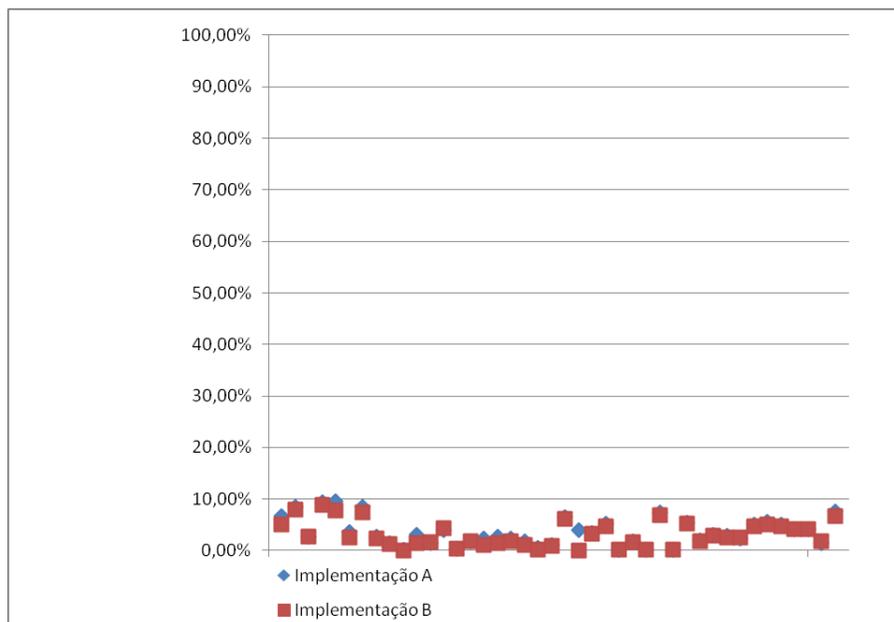


Figura 5.6: Gráfico da precisão para  $\alpha = 0,25$

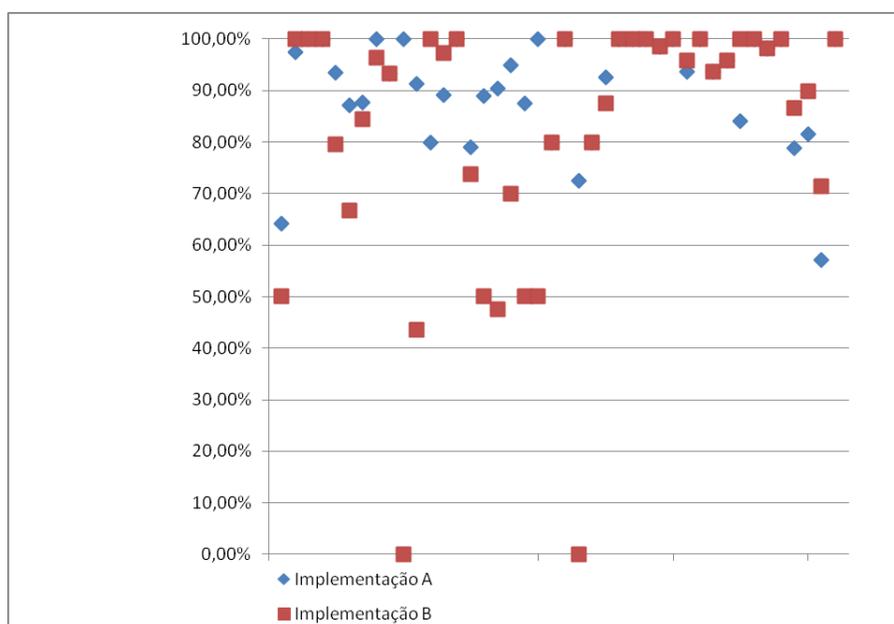


Figura 5.7: Gráfico da revocação para  $\alpha = 0,25$

Quadro 5.9: Experimentos - ANOVA - *F-measure*,  $\alpha = 0,25$

Fonte da Variação	SQ	gl	MQ	F	valor-P	F crítico
Entre grupos	0,001650102	1	0,001650102	0,747387882	0,389826493	3,957388177
Dentro dos grupos	0,181041735	82	0,002207826			
Total	0,182691837	83				

implementação B ser maior que a média da revocação da implementação A, não existe uma diferença significativa entre os dados como é apontado pela Quadro 5.12.

Por outro lado, a *f-measure* apresenta as mesmas características que a precisão, uma média muito maior e uma diferença significativa entre os dados, o que leva à conclusão

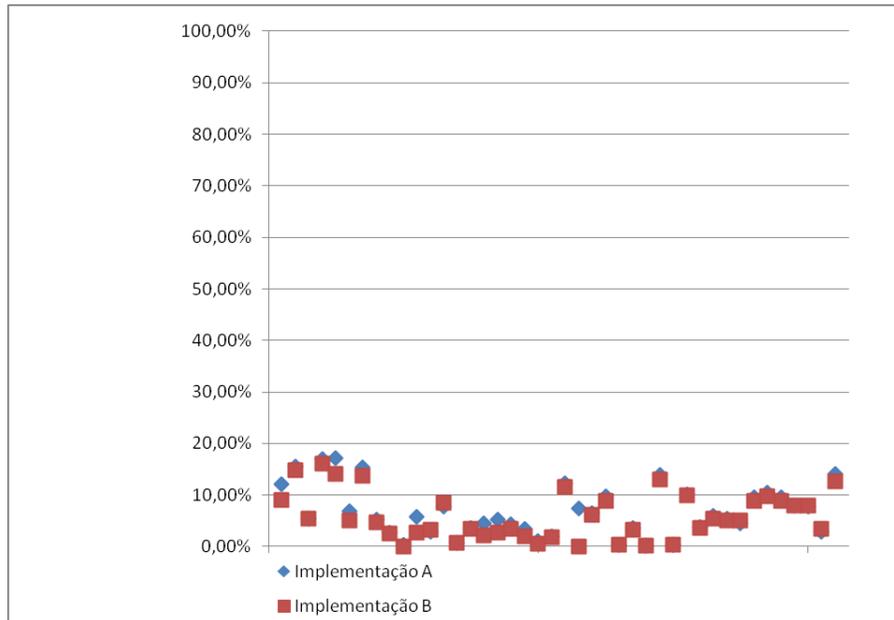


Figura 5.8: Gráfico da  $f$ -measure para  $\alpha = 0,25$

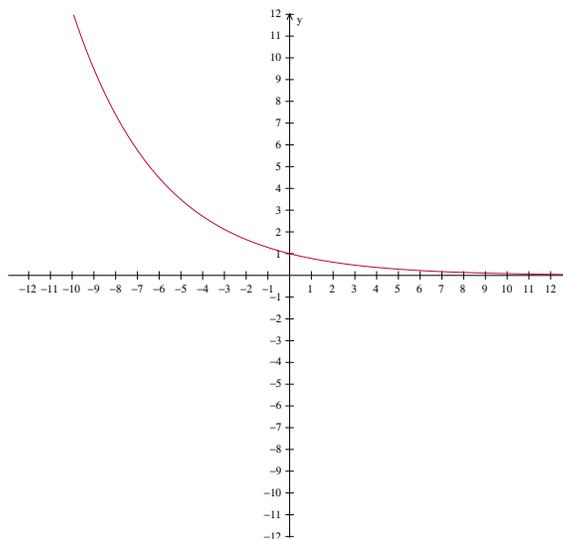


Figura 5.9: Função Exponencial, com  $\alpha = 0,25$

de que, por ter uma melhoria na precisão e na  $f$ -measure, e não saber se a revocação foi prejudicada de fato, o algoritmo em si também foi melhorado para essa configuração.

Os Quadros 5.14, 5.15 e 5.16 mostram respectivamente o resultado do ANOVA para os resultados de precisão, revocação e  $f$ -measure utilizando o aperfeiçoamento apresentado nessa dissertação. Observa-se pelos quadros que o ANOVA não aponta uma diferença significativa entre os dados, logo não é possível afirmar qual configuração dentre as duas trará um melhor resultado.

## 5.5 Resumo do Capítulo

Este capítulo apresentou os experimentos realizados no trabalho desta dissertação. Os resultados obtidos mostram que o algoritmo de Liu et al(2009) apresenta resultados

Quadro 5.10: Experimentos -  $\alpha = 0,2, \beta = 0,6, \lambda = 0,4$ .

consulta	Implementação A			Implementação B		
	precisão	revocação	f-measure	precisão	revocação	f-measure
1	100,00%	42,39%	59,54%	3,43%	7,61%	4,73%
2	95,08%	73,42%	82,86%	25,24%	100,00%	40,31%
3	92,86%	50,00%	65,00%	15,20%	100,00%	26,40%
4	100,00%	30,53%	46,77%	27,86%	100,00%	43,58%
5	94,44%	91,40%	92,90%	21,76%	61,29%	32,11%
6	94,29%	84,62%	89,19%	22,01%	89,74%	35,35%
7	97,50%	86,67%	91,76%	48,52%	91,11%	63,32%
8	96,30%	96,30%	96,30%	7,29%	92,59%	13,51%
9	100,00%	20,00%	33,33%	12,04%	86,67%	21,14%
10	100,00%	100,00%	100,00%	16,67%	100,00%	28,57%
11	90,00%	39,13%	54,55%	20,93%	39,13%	27,27%
12	66,67%	80,00%	72,73%	8,43%	100,00%	15,54%
13	100,00%	86,49%	92,75%	20,11%	100,00%	33,48%
14	21,43%	100,00%	35,29%	7,14%	100,00%	13,33%
15	21,74%	26,32%	23,81%	9,88%	42,11%	16,00%
16	100,00%	16,67%	28,57%	13,16%	27,78%	17,86%
17	70,59%	57,14%	63,16%	16,67%	28,57%	21,05%
18	63,16%	60,00%	61,54%	27,03%	50,00%	35,09%
19	71,43%	31,25%	43,48%	13,85%	56,25%	22,22%
20	100,00%	100,00%	100,00%	7,32%	75,00%	13,33%
21	43,75%	70,00%	53,85%	7,41%	40,00%	12,50%
22	100,00%	38,46%	55,56%	68,25%	82,69%	74,78%
23	100,00%	72,50%	84,06%	0,00%	0,00%	0%
24	96,00%	80,00%	87,27%	43,86%	83,33%	57,47%
25	100,00%	92,50%	96,10%	65,52%	95,00%	77,55%
26	50,00%	100,00%	66,67%	50,00%	100,00%	66,67%
27	87,50%	77,78%	82,35%	87,50%	77,78%	82,35%
28	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
29	98,04%	72,46%	83,33%	51,35%	82,61%	63,33%
30	50,00%	100,00%	66,67%	50,00%	100,00%	66,67%
31	64,71%	91,67%	75,86%	14,20%	95,83%	24,73%
32	95,00%	100,00%	97,44%	32,20%	100,00%	48,72%
33	96,15%	78,13%	86,21%	92,00%	71,88%	80,70%
34	100,00%	91,67%	95,65%	35,48%	91,67%	51,16%
35	100,00%	24,00%	38,71%	8,45%	100,00%	15,58%
36	97,87%	92,00%	94,85%	95,92%	94,00%	94,95%
37	100,00%	94,64%	97,25%	98,18%	96,43%	97,30%
38	93,75%	30,61%	46,15%	22,17%	100,00%	36,30%
39	100,00%	46,15%	63,16%	76,47%	50,00%	60,47%
40	94,74%	36,73%	52,94%	87,50%	42,86%	57,53%
41	100,00%	57,14%	72,73%	38,46%	71,43%	50,00%
42	85,71%	26,09%	40,00%	19,29%	78,26%	30,95%
Média	86,40%	67,73%	70,72%	35,45%	76,23%	42,24%

Quadro 5.11: Experimentos - ANOVA - Precisão,  $\alpha = 0,25, \beta = 0,6, \lambda = 0,4$ 

Fonte da Variação	SQ	gl	MQ	F	valor-P	F crítico
Entre grupos	5,451753727	1	5,451753727	79,91079139	$9,50492 * 10^{-14}$	3,957388177
Dentro dos grupos	5,594285801	82	0,068222998			
Total	11,04603953	83				

Quadro 5.12: Experimentos - ANOVA - Revocação,  $\alpha = 0,25, \beta = 0,6, \lambda = 0,4$ 

Fonte da Variação	SQ	gl	MQ	F	valor-P	F crítico
Entre grupos	0,151530842	1	0,151530842	1,937861842	0,167664991	3,957388177
Dentro dos grupos	6,411978798	82	0,078194863			
Total	6,56350964	83				

não razoáveis e que a melhoria que essa dissertação apresenta é bastante significativa, aumentando mais que o dobro da precisão que o algoritmo de Liu et al(2009) possui. Aqui foram apresentadas algumas das possíveis configurações para se utilizar no algoritmo e quão eficazes essas configurações são.

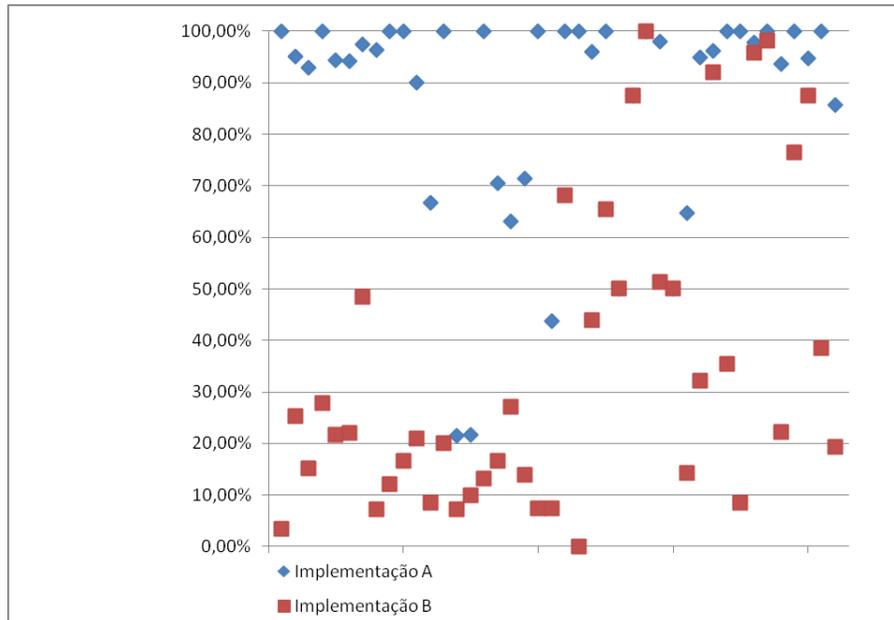


Figura 5.10: Gráfico da precisão para  $\alpha = 0,2$ ,  $\beta = 0,6$ ,  $\lambda = 0,4$

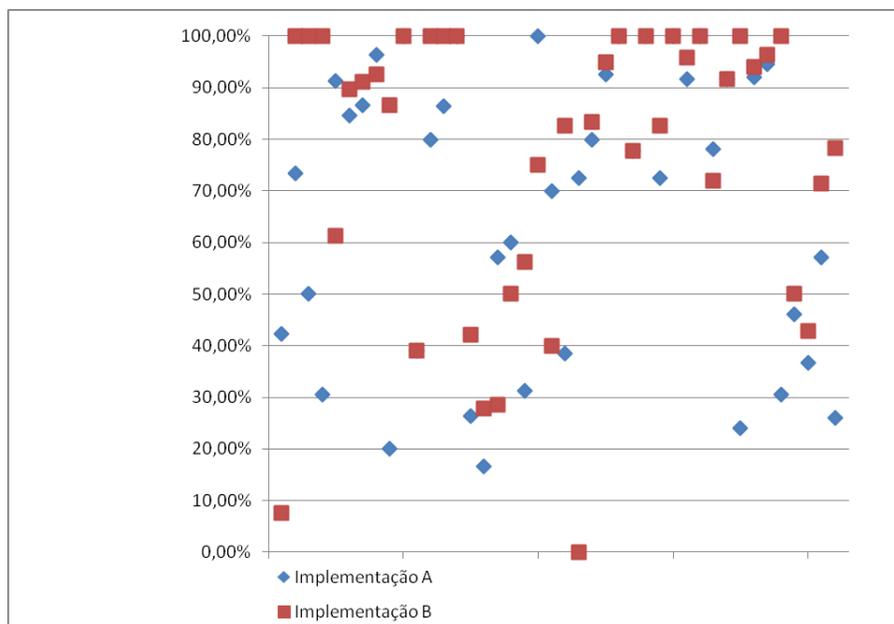


Figura 5.11: Gráfico da revocação para  $\alpha = 0,2$ ,  $\beta = 0,6$ ,  $\lambda = 0,4$

Quadro 5.13: Experimentos - ANOVA - *f-measure*,  $\alpha = 0,25$ ,  $\beta = 0,6$ ,  $\lambda = 0,4$

Fonte da Variação	SQ	gl	MQ	F	valor-P	F crítico
Entre grupos	1,704059622	1	1,704059622	27,39295588	$1,25277 \cdot 10^{-6}$	3,957388177
Dentro dos grupos	5,101051877	82	0,06220795			
Total	6,8051115	83				

Duas configurações apresentaram ótimos resultados, além de explicitar o aperfeiçoamento apresentado neste trabalho. Uma configuração não obteve êxito e apontou resultados nada favoráveis. As configurações foram escolhidas de acordo com o trabalho de Li, Bandar e McLean(2003). Para verificar a existência de uma diferença significativa entre

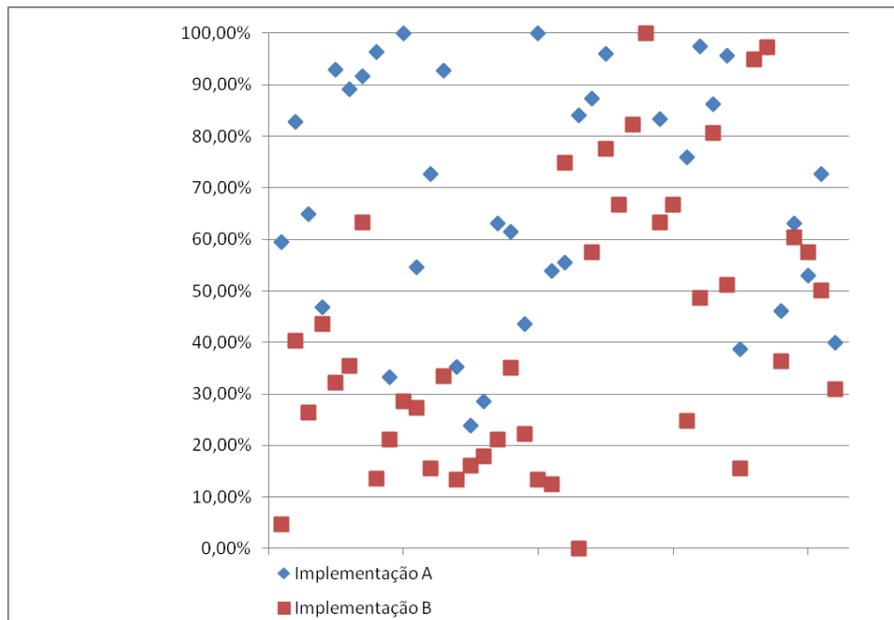


Figura 5.12: Gráfico da  $f$ -measure para  $\alpha = 0, 2\beta = 0, 6, \lambda = 0, 4$

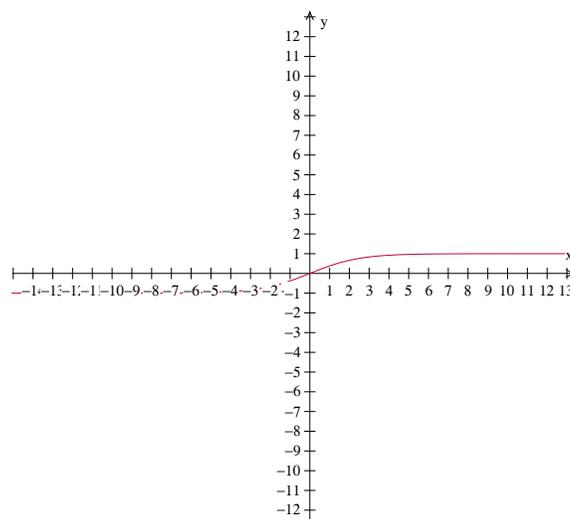


Figura 5.13: Função Monotônica, com  $\lambda = 0, 4$

Quadro 5.14: Experimentos - ANOVA - Comparação da precisão entre a primeira e terceira configuração

Fonte da Variação	SQ	gl	MQ	F	valor-P	F crítico
Entre grupos	002695447	1	0,002695447	0,05619898	0,813198982	3,957388177
Dentro dos grupos	3,932930026	82	0,047962561			
Total	3,935625473	83				

Quadro 5.15: Experimentos - ANOVA - Comparação da revocação entre a primeira e terceira configuração

Fonte da Variação	SQ	gl	MQ	F	valor-P	F crítico
Entre grupos	0,005921731	1	0,005921731	0,076475621	0,782827453	3,957388177
Dentro dos grupos	6,349499754	82	0,077432924			
Total	6,355421485	83				

Quadro 5.16: Experimentos - ANOVA - Comparação da *F-measure* entre a primeira e terceira configuração

Fonte da Variação	SQ	gl	MQ	F	valor-P	F crítico
Entre grupos	0,000646665	1	0,000646665	0,012251967	0,912133753	3,957388177
Dentro dos grupos	4,328003767	82	0,052780534			
Total	4,328650432	83				

os dados, foi utilizado o ANOVA.

## 6 CONCLUSÕES

O algoritmo de Liu et al (2009) mede a similaridade semântica entre serviços Web semânticos. Seu algoritmo é baseado em ontologia e calcula a similaridade entre dois conceitos com base na estrutura da taxonomia que envolve a ontologia que descreve estes conceitos. Cada serviço Web semântico possui quatro categorias: Entradas, Saídas, Pré-condições e Efeitos. No entanto, o algoritmo calcula de maneira não-ótima a similaridade entre as categorias.

Este trabalho apresentou um aperfeiçoamento deste algoritmo. O ponto-chave deste aperfeiçoamento é na maneira que a similaridade entre as categorias é calculada. Este trabalho ilustra como o algoritmo de Liu et al (2009) funcionaria com um desempenho fraco e apresenta uma solução para esse ponto-chave. Experimentos foram realizados e comprovaram tal melhoria. Lembrando que o trabalho de Liu et al (2009) sequer realizou algum experimento para medir o desempenho de sua abordagem.

A base de dados utilizada possui 1083 Serviços Web Semânticos e 41 consultas. Três configurações foram sugeridas e testadas, duas delas obtendo excelentes resultados, dobrando a precisão do algoritmo original. Fora isso, este trabalho apresentou o conceito de densidade semântica contextualizado para serviços Web semânticos e definiu um limiar de similaridade para o algoritmo aqui apresentado.

### 6.1 Publicações Resultantes

A realização desta pesquisa deixa como contribuições o aperfeiçoamento do algoritmo de Liu et al (2009), junto com as definições de um limiar para atribuir a similaridade e a contextualização de densidade semântica para Serviços Web Semânticos. Até o presente momento foram geradas três publicações.

A primeira publicação trata-se de um artigo completo publicado em revista (MAAMAR et al., 2011), na *IEEE Internet Computing*, conceito A2 pela CAPES. Esse trabalho foca em uma técnica de descoberta de serviços Web utilizando redes sociais. Aqui a rede social inicial é construída a partir do uso do algoritmo de Liu et al (2009). Alguns experimentos foram realizados utilizando uma base muito menor, e foi nesse momento que se percebeu falhas no algoritmo e que ele necessitava ser otimizado.

A segunda publicação trata-se de um artigo completo publicado nos anais da conferência *ACM WIMS (Web Intelligence, Mining and Semantics) 2011* (MAAMAR et al., 2011). Esse artigo foca mais no framework para construção das redes sociais de recomendação de Serviços Web, e os experimentos utilizados foram os mesmos utilizados no primeiro artigo publicado.

Em seguida, foi submetido o artigo *An Improved Approach for Measuring Similarity among Semantic Web Services*, que trata do trabalho apresentado aqui nesta dissertação,

para a conferência *NWeSP (Next Generation Web Services and Practices) 2011*, conceito B3 pela CAPES e com anais indexados pela *IEEE*, e o artigo foi aceito para publicação como artigo completo. No entanto, não tivemos recurso e infelizmente perdemos a publicação.

Por último, submetemos o mesmo artigo para o *WEBIST (Web Information Systems and Technologies) 2012* e o artigo foi aceito para publicação, e desta vez tivemos recurso e o artigo será publicado.

## 6.2 Trabalhos Futuros

Como trabalhos futuros, têm-se o uso de outras bases de dados disponíveis para uso público e citadas na literatura, como por exemplo, a SAWSDL. O uso de outras configurações para avaliar seu desempenho também será realizado mais para frente, além da comparação de seu desempenho com outras técnicas encontradas e a submissão para o evento de análise de desempenho SWS Challenge.

Também se planeja utilizar técnicas de lógica de descrição e analisar se ocorre uma melhoria para o cálculo de similaridade entre conceitos. Nessa parte de similaridade entre conceitos, planeja-se utilizar técnicas recentes na literatura para similaridade entre palavras e verificar se elas atingem alguma melhoria. Além disso é necessário fazer uma análise mais detalhada e comparativa dos parâmetros e configurações utilizadas.

Os periódicos *IEEE Internet Computing*, *IEEE Transactions on Service Computing* e *IEEE Transactions on Software Engineering* são visados para a próxima submissão contendo os resultados obtidos dos trabalhos futuros.

## REFERÊNCIAS

- ARMSTRONG, R. A.; SLADE, S. V.; EPERJESI, F. An introduction to analysis of variance (ANOVA) with special reference to data from clinical experiments in optometry. **Ophthalmic physiological optics the journal of the British College of Ophthalmic Opticians Optometrists**, [S.l.], v.20, n.3, p.235–241, 2000.
- BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The semantic web. **Scientific American**, [S.l.], May 2001.
- BRUIJN, J. de et al. Web Service Modeling Ontology (WSMO). In: W3C RECOMMENDATION. **Anais...** [S.l.: s.n.], 2005.
- CABRAL, L. et al. Approaches to Semantic Web Services: an overview and comparisons. In: EUROPEAN SEMANTIC WEB CONFERENCE. **Anais...** [S.l.: s.n.], 2004. p.225–239. (Lecture Notes in Computer Science).
- CHRISTENSEN, E. et al. Web Services Description Language (WSDL). In: W3C RECOMMENDATION. **Anais...** [S.l.: s.n.], 2001.
- CROFT, B.; METZLER, D.; STROHMAN, T. **Search Engines: information retrieval in practice**. 1.ed. [S.l.]: Addison Wesley, 2009.
- CURBERA, F. et al. Unraveling the Web services web: an introduction to soap, wsdl, and uddi. **Internet Computing, IEEE**, [S.l.], v.6, n.2, p.86–93, march-april 2002.
- FISHER, R. A. On the probable error of a coefficient of correlation deduced from a small sample. **Metron**, [S.l.], v.1, n.4, p.3–32, 1921.
- GAWINECKI, M. et al. WSCOLAB: structured collaborative tagging for web service matchmaking. In: INTERNATIONAL CONFERENCE ON WEB INFORMATION SYSTEMS AND TECHNOLOGIES. **Anais...** Springer, 2010. (Lecture Notes in Business Information Processing).
- GUDGIN, M. et al. SOAP Version 1.2 Part 1: messaging framework. In: W3C RECOMMENDATION. **Anais...** [S.l.: s.n.], 2003.
- HORROCKS, I. et al. SWRL: a semantic web rule language combining owl and ruleml. In: W3C RECOMMENDATION. **Anais...** [S.l.: s.n.], 2004.
- KHDOUR, T.; FASLI, M. A Semantic-Based Web Service Registry Filtering Mechanism. In: ADVANCED INFORMATION NETWORKING AND APPLICATIONS WORKSHOPS (WAINA), 2010 IEEE 24TH INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2010. p.373–378.

- KIFER, M. et al. A Realistic Architecture for the Semantic Web. In: RULEML. **Anais...** [S.l.: s.n.], 2005. p.17–29.
- KLUSCH, M.; FRIES, B.; SYCARA, K. OWLS-MX: a hybrid semantic web service matchmaker for owl-s services. **Web Semantics: Science, Services and Agents on the World Wide Web**, [S.l.], v.7, n.2, p.121 – 133, 2009.
- KLUSCH, M.; ZHING, X. Deployed Semantic Services for the Common User of the Web: a reality check. In: SEMANTIC COMPUTING, 2008 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2008. p.347 –353.
- KOPECKY, J. et al. SAWSDL: semantic annotations for wsdl and xml schema. **IEEE Internet Computing**, [S.l.], v.11, n.6, p.60 –67, nov.-dec. 2007.
- KRITIKOS, K.; PLEXOUSAKIS, D. Semantic QoS Metric Matching. In: WEB SERVICES, 2006. ECOWS '06. 4TH EUROPEAN CONFERENCE ON. **Anais...** [S.l.: s.n.], 2006. p.265 –274.
- LI, X. et al. A Petri Net Approach to Analyzing Behavioral Compatibility and Similarity of Web Services. **Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on**, [S.l.], v.41, n.3, p.510 –521, may 2011.
- LI, Y.; BANDAR, Z. A.; MCLEAN, D. An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. **IEEE Transactions on Knowledge and Data Engineering**, Piscataway, NJ, USA, v.15, p.871–882, July 2003.
- LIU, F. et al. Measuring Similarity of Web Services Based on WSDL. In: WEB SERVICES (ICWS), 2010 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2010. p.155 –162.
- LIU, M. et al. An weighted ontology-based semantic similarity algorithm for web service. **Expert Systems with Applications**, [S.l.], v.36, n.10, p.12480–12490, 2009.
- MAAMAR, Z. et al. Using Social Networks for Web Services Discovery. **Internet Computing, IEEE**, [S.l.], v.15, n.4, p.48 –54, july-aug. 2011.
- MAAMAR, Z. et al. Towards a framework for weaving social networks principles into web services discovery. In: INTERNATIONAL CONFERENCE ON WEB INTELLIGENCE, MINING AND SEMANTICS, New York, NY, USA. **Proceedings...** ACM, 2011. p.51:1–51:11. (WIMS '11).
- MARTIN, D. et al. OWL-S: semantic markup for web services. In: W3C RECOMMENDATION. **Anais...** [S.l.: s.n.], 2004.
- MCILRAITH, S.; SON, T.; ZENG, H. Semantic Web services. **IEEE Intelligent Systems**, [S.l.], v.16, n.2, p.46 – 53, mar-apr 2001.
- PAPAZOGLU, M. P. **Web Services: Principles and Technology**. [S.l.]: Pearson, Prentice Hall, 2008.
- PAPAZOGLU, M.; RIBBERS, P. **e-Business - Organisational And Technical Foundations**. [S.l.]: John Wiley & Sons, 2006.

PELTZ, C. Web services orchestration and choreography. **Computer**, [S.l.], v.36, n.10, p.46–52, 2003.

PETRIE, C. Practical Web Services. **Internet Computing, IEEE**, [S.l.], v.13, n.6, p.93–96, nov.-dec. 2009.

SAHAMI, M.; HEILMAN, T. D. A web-based kernel function for measuring the similarity of short text snippets. In: **WORLD WIDE WEB**, 15., New York, NY, USA. **Proceedings...** ACM, 2006. p.377–386. (WWW '06).

SALTON, G.; BUCKLEY, C. Term-weighting approaches in automatic text retrieval. In: **INFORMATION PROCESSING AND MANAGEMENT. Anais...** [S.l.: s.n.], 1988. p.513–523.

SHADBOLT, N.; BERNERS-LEE, T.; HALL, W. The Semantic Web Revisited. **IEEE Intelligent Systems**, [S.l.], v.21, n.3, p.96–101, 2006.

STUDER, R.; GRIMM, S. **Semantic Web Services, Concepts, Technologies, and Applications**. Berlin, Heidelberg: Springer-Verlag, 2007.

SYCARA, K. et al. Automated discovery, interaction and composition of Semantic Web Services. **Journal of Web Semantics**, [S.l.], v.1, p.27–46, 2003.

VERMA, K. et al. **The METEOR-S approach for configuring and executing dynamic web processes**. [S.l.: s.n.], 2005.

WEI, D. et al. Extracting Semantic Constraint from Description Text for Semantic Web Service Discovery. In: **INTERNATIONAL CONFERENCE ON THE SEMANTIC WEB**, Berlin, Heidelberg. **Anais...** Springer-Verlag, 2008. p.146–161. (Lecture Notes In Computer Science).

WSCOLAB. **Maciej Gawinecki** – **WScolab**. Disponível em: <<http://www.ibspan.waw.pl/gawinec/wss/wscolab.html>>. Acesso em: julho 2010.

YU, Q. et al. Deploying and managing Web services: issues, solutions, and directions. **The VLDB Journal**, Secaucus, NJ, USA, v.17, p.537–572, May 2008.