



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

Dissertação de Mestrado

**MeMO – Uma Estratégia para *Merging*
Múltiplo de Ontologias baseada em
*clustering***

FABIANA FREIRE DE ARAÚJO

FORTALEZA/CE
2009



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

MeMO – Uma Estratégia para *Merging* Múltiplo de Ontologias baseada em *clustering*

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal do Ceará, como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientadora: Prof^ª. Dr^ª. Bernadette Farias Lóscio

FABIANA FREIRE DE ARAÚJO



**UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO**

MeMO – Uma Estratégia para *Merging* Múltiplo de Ontologias
baseada em *clustering*

Fabiana Freire de Araújo
Orientadora: Prof^a. Dr^a. Bernadette Farias Lóscio

Aprovada em __/__/____

BANCA EXAMINADORA

Prof^a. Dr^a. Bernadette Farias Lóscio (Orientadora)
Universidade Federal do Ceará – UFC

Prof^a. Dr^a Diana Magalhães de Oliveira
Universidade Estadual do Ceará – UECE

Prof^a Dr^a Vânia Maria Ponte Vidal
Universidade Federal do Ceará – UFC

Prof^a Dr^a Ana Carolina Brandão Salgado
Universidade Federal de Pernambuco– UFPE

*Dedico esse trabalho a
toda minha família e amigos*

AGRADECIMENTOS

Agradeço a Deus por ter me possibilitado conhecer as pessoas certas para conseguir todas as minhas conquistas até este momento e que continuarei conquistando, colhendo os frutos de um trabalho dedicado e que trouxe muitas gratificações.

Agradeço a Deus por me indicar os caminhos da minha monografia (orientada pela professora Diana Magalhães e pelo professor Marcos Negreiros) que ajudou nos primeiros passos do meu mestrado.

Agradeço a Deus pela minha orientadora, professora Bernadette (que sofreu bastante comigo), a ter aceitado o desafio de me orientar. Tenho certeza que metade dos pecados ela já pagou nesses últimos dois anos [: D].

Agradeço a Deus por ter me disponibilizado ótimas pessoas para me acompanharem nesses últimos anos: as pessoas do NUGEN e as pessoas do grupo ARIDA, principalmente um subgrupo muito importante que me ajudou fortemente no desfecho do trabalho. Esse subgrupo, o RTL, é formado por Hélio, Danusa e Lígia.

Agradeço a Deus por todos os meus familiares que direta ou indiretamente me ajudaram não só nesse momento, mas em todos os momentos anteriores que culminaram no mestrado, principalmente meus pais, “Vasquim” e “Ledinha”; minhas tias, Márcia (Mãe 2) e Mika, que me acolheram em momentos decisivos de minha vida profissional e pessoal; e meu irmão Dudu, que é todo “fresquim” e que conversou bastante comigo pelo *skype* em vários momentos do mestrado, me ajudando a prosseguir nos momentos de fraqueza, me fazendo rir nos momentos de angústia, e me deixando “xingá-lo” nos momentos de desabafo [;-)].

Agradeço a Deus por todas as demais oportunidades (pessoas e instituições) que me foram dadas (UFC, UECE, funcionários e Funcap) para o início e término deste trabalho.

RESUMO

Uma vez que diferentes ontologias estão disponíveis na Web, sendo muitas do mesmo domínio e representadas de formas diferentes, a necessidade de combiná-las (*merging*) para obter uma visão integrada e, conseqüentemente, o enriquecimento de um domínio, consiste em um problema relevante para diversas aplicações. Várias soluções foram propostas, mas todas abordando uma combinação de apenas duas ontologias. Em casos onde seja necessário combinar mais de duas ontologias, de forma automática, não há garantias a respeito da qualidade da ontologia final gerada. Visando contornar esta limitação, este trabalho propõe uma estratégia baseada na técnica de *clustering* hierárquico para o *Merging* Múltiplo de Ontologias (MeMO) onde é realizada uma análise prévia da similaridade entre as ontologias, a fim de agrupá-las de acordo com suas semelhanças e então combiná-las progressivamente. Este agrupamento é realizado com o intuito de determinar a ordem na qual as combinações devem ser realizadas e, dessa forma, obter melhores resultados. Este trabalho mostra que o uso da semelhança entre as ontologias durante a combinação automática pode causar impactos positivos sobre o resultado final se comparado a uma combinação aleatória das ontologias.

Palavras-chave: *Merging* de Ontologias, Técnica de *Clustering*, Integração Semântica de Dados

ABSTRACT

Since numerous ontologies are available on the web, the requirement for merging them remains a pertinent issue for several applications. A few solutions have been proposed, although attempts to find entirely automated techniques to integrate more than two ontologies into a single one have given no warranty over the quality of the resulting merged ontology. To address such challenge, we propose a clustering-based approach for Merging Multiple Ontologies (MeMO). It calculates the similarity among the ontologies in order to define the order in which the ontologies will be merged to obtain better results. Through this approach, the assistance of the similarity among ontologies during the automatic merging will improve the final result.

Keyword: Ontology Merging, *Clustering Analysis*, Semantic Integration

SUMÁRIO

CAPÍTULO 1 - Introdução.....	13
1.1 Motivação	15
1.2 Objetivos.....	18
1.3 Contribuições.....	18
1.4 Organização do Trabalho.....	19
CAPÍTULO 2 - Fundamentação Teórica.....	20
2.1 Técnicas de <i>Clustering</i>	20
2.1.1 Etapa I: Definição da técnica de agrupamento	21
2.1.2 Etapa II: Definição de uma medida de semelhança (dessemelhança) entre objetos.....	22
2.2.3 Etapa III: Definição do método de agrupamento ou de construção dos clusters.....	23
2.2 Ontologias.....	25
2.2.1 Técnicas de <i>Matching</i> de Ontologias	26
2.2.2 Técnicas de <i>Merging</i> de Ontologias.....	32
2.2.3 Soluções para <i>Matching</i> e <i>Merging</i> de Ontologias	37
2.3 Avaliação de Ontologias.....	42
2.3.1 Medidas para Avaliação de Ontologias.....	43
2.4 Considerações finais.....	48
CAPÍTULO 3 - <i>Merging</i> Múltiplo de Ontologias.....	50
3.1 Visão Geral	50
3.2 MeMO – Estratégia para <i>Merging</i> Múltiplo de Ontologias	52
3.2.1 Etapa I: Construção da matriz de similaridades.....	53
3.2.2 Etapa II: Combinação Progressiva.....	56
3.3 Considerações Finais.....	69
CAPÍTULO 4 - Avaliação da estratégia MeMO.....	70
4.1 Descrição dos Experimentos.....	70
4.2 O Protótipo MeMOTOol	74
4.3 Resultados dos experimentos.....	75
4.4 Análise dos Resultados dos Experimentos	83
4.5 Considerações finais.....	86

CAPÍTULO 5 - Conclusões.....	88
5.1 Contribuições.....	88
5.2 Publicações.....	89
5.3 Trabalhos Futuros.....	90
5.4 Considerações Finais.....	92
REFERÊNCIAS.....	94
APÊNDICES	91
APÊNDICE A – Alinhamento entre ontologias.....	99
APÊNDICE B – Ontologias Resultantes de um Merging Múltiplo.....	105
APÊNDICE C – Ontologias Utilizadas nos Experimentos.....	109

LISTA DE FIGURAS

Figura 1. Arquitetura do Sistema FlagellLink	16
Figura 2. Exemplo de formação de três grupos de dados correspondentes	20
Figura 3. Exemplo de um dendrograma onde foram identificados dez grupos.....	22
Figura 4. Cálculo da Matriz de Distância.....	23
Figura 5. Abordagens hierárquicas	24
Figura 6. Exemplo de duas Ontologias de entrada.....	27
Figura 7. Exemplo de uma matriz de correspondências.....	28
Figura 8. Exemplo de duas possíveis matrizes de correspondências	31
Figura 9. Exemplo de alinhamento	32
Figura 10. Processo de Merging de Ontologias.....	33
Figura 11. Abordagens de <i>merging</i> de ontologias. Retirada de [Davies et al 2006].....	34
Figura 12. Exemplo de duas ontologias do domínio University	34
Figura 13. Exemplos de merging de ontologias: A) Merge Completo; B) Ontologia Ponte	35
Figura 14. Exemplo de ontologia referência (ORef, à esquerda) e computada (OComp, à direita). Figura adaptada de [Maedche et al 2002] e [Dellschaft e Staab, 2006].....	45
Figura 15. Ilustração da Estratégia MeMO	51
Figura 16. Exemplo de uma matriz de similaridade.....	53
Figura 17. Árvore binária resultante do merging múltiplo de quatro ontologias.....	57
Figura 18. Exemplo de identificação de duas ontologias mais semelhantes	58
Figura 19. Exemplo de duas ontologias O_i e O_j e seu alinhamento $A(O_i, O_j)$	61
Figura 20. Exemplo da ontologia resultante do merging das duas ontologias O_i e O_j	61
Figura 21. Exemplo de Construção de novos alinhamentos	63
Figura 22. Reconstrução da matriz da Figura 3.2.....	63
Figura 23. Exemplo de três ontologias do domínio Universidade	65
Figura 24. Exemplo de junção de dois alinhamentos	66
Figura 25. Demonstração da validade do processo de Junção de Alinhamentos	67
Figura 26. Funcionalidade da MeMOTool.....	74
Figura 27. Interface gráfica da ferramenta MeMO.....	75
Figura 28. Quadro comparativo entre as medidas <i>F-measure</i> encontradas nos experimentos	85

LISTA DE ALGORITMOS

Algoritmo 1. Algoritmo Genérico para Métodos Aglomerativos	25
Algoritmo 2. Construção da matriz de similaridade inicial.....	56
Algoritmo 3. Combinação Progressiva	58
Algoritmo 4. Junção de Alinhamentos	68

LISTA DE TABELAS

Tabela 1. Resumo de técnicas de comparação de elementos	29
Tabela 2. Técnicas de comparação estrutural	30
Tabela 3. Quadro Comparativo das Soluções Estudadas	41
Tabela 4. Extração de características das ontologias da Figura 14.....	47
Tabela 5. Exemplo das medidas globais.....	48
Tabela 6. Experimentos realizados para validação da estratégia MeMO	72
Tabela 7. Resultados do primeiro experimento	77
Tabela 8. Resultados do segundo experimento	78
Tabela 9. Resultados do terceiro experimento	79
Tabela 10. Resultados do quarto experimento.....	80
Tabela 11. Resultados do quinto experimento.....	81
Tabela 12. Resultados do quinto experimento.....	82
Tabela 13. Resultados dos experimentos	83

CAPÍTULO 1

Introdução

Uma ontologia caracteriza uma área ou um domínio de discurso, identificando conceitos e relacionamentos entre eles, geralmente definidos em uma linguagem formal, e disponibilizando metadados e informações explícitas quanto à semântica dos dados. De uma maneira geral, as ontologias são úteis para garantir uma comunicação livre de ambigüidades. No contexto da *web* semântica, o termo ontologia é usado como “uma especificação formal e explícita de uma conceitualização compartilhada” [Gruber, 1993]. As ontologias formam, portanto, a base da integração semântica e são consideradas fundamentais para o compartilhamento de informações.

Idealmente, esperava-se que cada domínio tivesse uma única ontologia, capaz de representar adequadamente seus conceitos e suas relações. Entretanto, especificar tais ontologias é uma tarefa extremamente complicada. Dessa forma, surgem diferentes ontologias sobre um mesmo domínio, as quais, na maioria das vezes, utilizam diferentes terminologias e estruturas hierárquicas distintas para a definição dos conceitos do domínio. Diante dessa grande diversidade de ontologias, surge a necessidade de combiná-las (*merging*), a fim de possibilitar a troca de informações.

A combinação de ontologias consiste em criar uma nova ontologia a partir de duas ou mais ontologias existentes. Esta combinação pode seguir diferentes abordagens como, por exemplo, a *união*; onde a nova ontologia conterá todos os elementos das duas ontologias iniciais. É importante destacar que as soluções atualmente disponíveis na literatura contemplam o *merging* de apenas duas ontologias, sendo algumas automáticas [Rasgado e Guzman, 2006] e outras semi-automáticas [Aumueller et al 2005], [Noy e Musen, 2002].

Entretanto, há situações em que é preciso realizar a combinação de mais de duas ontologias de forma automática, como em aplicações voltadas para integração de bancos de dados biológicos [Araújo et al 2008], onde se tem ontologias locais representando bancos de dados e é preciso ter uma visão integrada (global) de todas essas ontologias. Neste caso, apesar das soluções disponíveis na literatura serem extensíveis para a combinação de múltiplas ontologias, não há garantias a respeito da qualidade da ontologia final gerada, denominada ontologia global. A ontologia global será mais ou menos adequada, ou até mesmo ideal, dependendo dos objetivos iniciais de sua criação e de como ela será aplicada. Para que a ontologia global atenda seus objetivos, faz-se necessário o

envolvimento de um especialista do domínio. Na sua ausência, entretanto, será preciso contar com algum artifício que aproxime o resultado final dos objetivos inicialmente previstos.

Outro fator importante a ser considerado com relação às soluções para *merging* de ontologias diz respeito ao custo de execução da operação de *matching* [Euzenat e Valtchev, 2003], que deve ser realizada no início de cada processo de *merging* a fim de identificar as correspondências (alinhamentos) entre os elementos das ontologias a serem combinadas. Vale ressaltar que de acordo com a quantidade de ontologias e com o grau de complexidade das mesmas, a atividade de *matching* pode tornar-se bastante dispendiosa.

Neste contexto, o presente trabalho propõe uma estratégia para *Merging* Múltiplo de Ontologias (MeMO). Esta estratégia tem como base a técnica de *clustering* [Everit, 1980] e realiza combinações progressivas automáticas entre ontologias, de acordo com as semelhanças existentes entre elas. Deste modo, o uso da semelhança entre as ontologias, durante a combinação automática, irá causar um impacto positivo (favorecendo uma maior adequação aos objetivos) sobre o resultado final.

Dentre os diferenciais da estratégia MeMO destacam-se: i) o processo de combinação automática de múltiplas ontologias; ii) o processo de junção dos alinhamentos de ontologias, a fim de evitar novas chamadas ao processo de *matching*; iii) a análise prévia da similaridade entre as ontologias, resultando em uma ontologia global mais próxima da ontologia ideal; e iv) a disponibilização de ontologias intermediárias (visões parciais) e dos alinhamentos gerados durante o processo, possibilitando a reutilização dos mesmos. É importante destacar que este trabalho não pretende substituir as soluções existentes na atualidade, mas propõe contribuir onde tais soluções não são suficientes.

Para validação da estratégia proposta foi desenvolvido um protótipo (*MeMOTool*¹) e foram realizados diversos experimentos com o intuito de demonstrar que a ontologia global obtida retrata uma visão global mais próxima da visão global ideal. Através dos experimentos mostrou-se que a ordem de execução do *merging* entre as ontologias pode afetar a ontologia global resultante, de maneira que melhores resultados podem ser obtidos quando ontologias mais semelhantes são combinadas em primeiro lugar.

O restante deste capítulo está organizado da seguinte forma: a seção 1.2 explica as motivações que levaram ao desenvolvimento deste trabalho; a seção 1.3 descreve pontualmente os objetivos a

¹ <http://www.lia.ufc.br/~fabiana/memo.html>

serem alcançados; a seção 1.4 mostra objetivamente as contribuições advindas deste trabalho; e finalmente, na seção 1.5 é apresentada a organização desta dissertação.

1.1 Motivação

A principal motivação para este trabalho surgiu da necessidade de integrar bancos de dados biológicos de forma automática. Esta necessidade foi identificada durante o desenvolvimento de um projeto em parceria com o NUGEN (Núcleo de Genômica e Bioinformática, da UECE – <http://nugen.uece.br>).

O NUGEN tem se especializado em sequenciar genomas e analisar os dados sobre organismos flagelados (como o parasito causador da zoonose leishmaniose visceral, a *Leishmania chagasi* e a alga verde *Chlamydomonas reinhardtii*). Nesse sentido, genes e proteínas dos organismos são estudados na perspectiva de se conhecer, por exemplo, os componentes estruturais e funcionais do flagelo, uma organela-chave na motilidade e na virulência destas espécies [Oliveira et al., 2005]. Como consequência desses estudos surgiu o FlagelLink [Araújo *et al.* 2008], um sistema para integração de dados biológicos relacionados ao domínio flagelo e que estão distribuídos em diversas fontes de dados públicas disponíveis na Web.

O sistema FlagelLink utiliza uma abordagem centralizada onde os dados são extraídos das fontes de dados e armazenados em um banco de dados local, o FlagelLinkDB, com o auxílio de uma ontologia global, para garantir que os dados adicionados não sejam redundantes.

Como pode ser visto na Figura 1, o FlagelLink é um sistema baseado em um modelo de três camadas: a primeira é a *Camada de Bancos de Dados*, onde os esquemas das fontes de dados são representados por ontologias locais que serão armazenadas em um repositório local (ROn); a segunda camada é a *Camada de Integração* responsável pela integração semântica dos dados; a terceira e última camada é a *Camada de Usuário* responsável pela interação com o usuário, onde se encontram as ferramentas de análise e busca sobre os bancos de dados do FlagelLink (FlagelLinkDB, BlastDB e HMMDB).

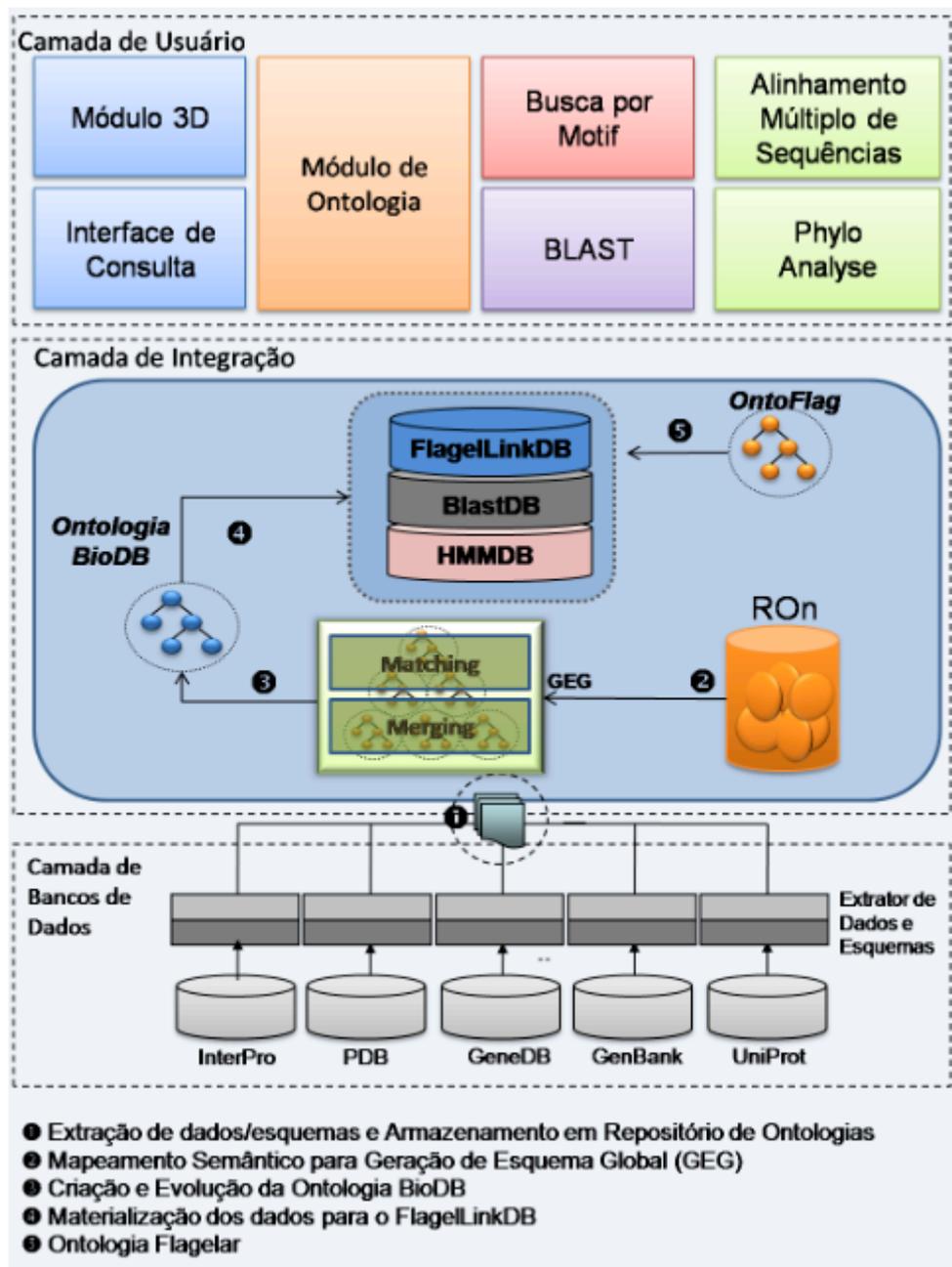


Figura 1. Arquitetura do Sistema FlagelLink

A estratégia proposta neste trabalho para *merging* múltiplo de ontologias pode ser aplicada na segunda camada do FlagelLink. Com o uso da estratégia MeMO, todas as ontologias armazenadas em ROn podem ser combinadas (Passo ‘2’ na Figura 1) e uma nova ontologia global (Ontologia BioDB) será gerada (Passo ‘3’), representando uma visão global de todas as ontologias de ROn. Com essa ontologia global, podem-se combinar os próprios dados extraídos de suas fontes originais para serem materializados no *data warehouse* (DW) FlagelLinkDB sem redundâncias. Uma segunda ontologia (OntoFlag) também está presente nessa camada e tem como o objetivo enriquecer as consultas do usuário. A estratégia MeMO também pode ser aplicada na construção

dessa segunda ontologia

É importante destacar que, no contexto de dados biológicos, a atividade de integração dos dados é rotineira, visto que os conjuntos de bancos de dados biológicos disponíveis podem ser genéricos (de grandes proporções, onde os dados são de diferentes domínios) ou específicos para um determinado domínio (exclusivos como, por exemplo, um banco de dados sobre um único organismo, o ChlamyDB - <http://www.chlamy.org/chlamydb.html>). Dependendo dos objetivos das análises, os pesquisadores podem utilizar um determinado conjunto de bancos de dados para gerar repositórios de dados específicos de acordo com o que desejam analisar. Em outras situações pode ser necessário integrar diferentes fontes ou subconjuntos das fontes de dados. Como a estratégia MeMO disponibiliza ontologias intermediárias, estas podem ser reutilizadas para evitar reprocessamento.

Um sistema semelhante ao FlagelLink é o ONTOFUSION [Pérez-Rey et al 2006], no qual as fontes de dados são representadas por ontologias e estas são combinadas de acordo com os conjuntos de bancos de dados necessários para uma determinada análise. A principal diferença em relação ao FlagelLink está na abordagem de integração, pois o ONTOFUSION utiliza uma abordagem distribuída, não sendo necessário fundir os dados. A ontologia global nesse sistema, que pode ser gerada através da utilização da MeMO, é utilizada para consultar diretamente as diferentes bases de dados.

Além do ambiente de bancos de dados biológicos, a estratégia MeMO também poderá ser aplicada em outras áreas em que haja necessidade da combinação de múltiplas ontologias e que pouca (ou nenhuma) expertise humana esteja disponível, ou então para os casos em que as soluções existentes sejam insuficientes.

Estudos sobre alinhamento múltiplo de seqüências biológicas [Feng e Doolittle, 1987] ajudaram a formalizar a estratégia proposta. Da mesma forma que no alinhamento múltiplo de seqüências, a estratégia MeMO também é baseada na técnica de *clustering*, com o diferencial de que, além de executar o alinhamento de ontologias, ela também executa a atividade de combinação (*merge*) de ontologias (combinação progressiva). Como no alinhamento progressivo, a combinação progressiva também não garante encontrar o melhor resultado, mas busca um resultado com qualidade aceitável.

1.2 Objetivos

O objetivo principal deste trabalho é mostrar que, com uma análise prévia das similaridades entre as ontologias, é possível combiná-las de forma a garantir um resultado mais consistente e reproduzível (pode ser novamente gerado) do que se esse fator não fosse levado em consideração (ou seja, se a combinação fosse executada de forma aleatória). Assim, os objetivos específicos deste trabalho são:

- Especificação da estratégia para *Merging* Múltiplo de Ontologias (MeMO) baseada na técnica de *clustering*;
- Especificação de uma estratégia para junção de alinhamentos de ontologias para evitar chamadas desnecessárias à operação de *matching* de ontologias, visto que é uma operação custosa e que pode ser inviável para um processo automático;
- Criação de um protótipo (MeMOTool) para validação da estratégia proposta;
- Criação de ontologias ideais de forma semi-automática para serem comparadas com as ontologias obtidas com ou sem a utilização da estratégia;
- Realização de experimentos de teste e validação;
- Comparação dos resultados obtidos nos experimentos e das ontologias ideais.

1.3 Contribuições

Com os objetivos atingidos, este trabalho proporcionará as seguintes contribuições:

- Estratégia para *Merging* Múltiplo de Ontologias baseada em *clustering*;
- Estratégia para junção de dois alinhamentos para evitar a chamada de uma operação de *matching*. Essa estratégia é importante em processo automático de *merging* múltiplo;
- Ferramenta para *merging* de ontologias onde será possível, a partir de um alinhamento prévio confiável: (i) combinar duas ontologias; (ii) combinar múltiplas ontologias

utilizando a estratégia proposta neste trabalho ou um *merging* múltiplo aleatório; (iii) fazer a junção de dois alinhamentos; (iv) visualizar ontologias; (v) visualizar alinhamentos; (vi) visualizar a árvore binária que representa o *merging*, onde cada nó representa uma ontologia dada como entrada no início do processo e o nó raiz representa a ontologia global; e (vii) analisar a qualidade de ontologias através de uma comparação entre uma ontologia de referência (*gold standard*) e a ontologia computada ou resultante (ontologia global);

- Todas as ontologias e alinhamentos gerados durante este trabalho ficarão disponíveis para reuso.

1.4 Organização do Trabalho

Esse trabalho está dividido em cinco capítulos, incluindo a introdução:

- O capítulo 2 aborda a fundamentação teórica, onde se encontra a base necessária para o desenvolvimento deste trabalho e inclui uma visão geral sobre a técnica de *cluster*, conceitos básicos sobre ontologia e os processos que a envolvem, *matching* e *merging* de ontologias, além das técnicas e ferramentas mais difundidas na literatura, junto com um quadro comparativo sobre as mesmas.
- O capítulo 3 descreve a estratégia proposta através da apresentação das técnicas e ferramentas utilizadas, além dos algoritmos desenvolvidos, envolvendo toda a fundamentação teórica.
- O capítulo 4 descreve os experimentos realizados, o protótipo desenvolvido, as ontologias utilizadas para a realização dos experimentos, bem como os resultados obtidos com os experimentos. Além disso, é realizada uma análise detalhada sobre os resultados.
- O capítulo 5 apresenta a conclusão do trabalho, considerações finais e perspectivas de trabalhos futuros. Nesse capítulo serão listadas todas as contribuições e melhorias.

CAPÍTULO 2

Fundamentação Teórica

Neste capítulo serão abordados os conceitos fundamentais que serviram de alicerce para o desenvolvimento do trabalho proposto, incluindo a técnica de *clustering* e a interoperabilidade de ontologias, que envolve os conceitos de *matching* e *merging* de ontologias. Na seção 2.1 será apresentada a técnica de *clustering* com seus principais passos; na seção 2.2 serão apresentados os principais conceitos relacionados aos processos de *matching* e *merging* de ontologias, além das principais ferramentas pertinentes encontradas na literatura; e a seção 2.3 apresentará conceitos relacionados à qualidade de ontologias, úteis na avaliação dos resultados obtidos neste trabalho.

2.1 Técnicas de *Clustering*

Para tratar dados em grandes quantidades de forma a se obter resultados que possam ser aproveitados como informação/conhecimento, é preciso, em geral, formar grupos (*clusters*) de dados que tenham “certa” correspondência entre si (Figura 2). Tais grupos são caracterizados por padrões [Theodoridis e Koutroumbas, 2003] encontrados nos dados.

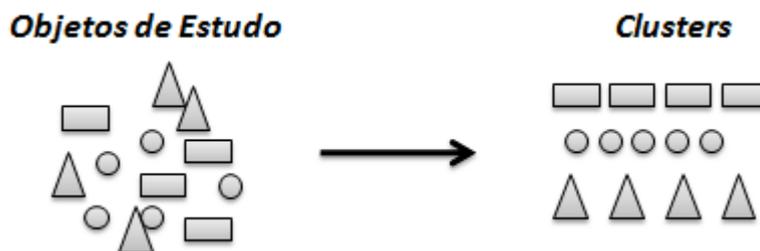


Figura 2. Exemplo de formação de três grupos de dados correspondentes

Reconhecimento de padrões é uma sub-área de aprendizagem de máquinas em que se busca identificar um determinado padrão em um objeto de estudo para uma possível classificação do mesmo [Mrozek e Cyran, 2001]. Essa sub-área tem por objetivo a classificação dos dados em um número de categorias ou classes, existindo duas abordagens principais: sistemas supervisionados e

não-supervisionados. A diferença entre os dois sistemas é que no primeiro as características dos objetos de estudo são conhecidas, incluindo a categoria a que pertencem, sendo esse sistema usado para classificação de um grande conjunto de dados; já no segundo, a categoria a que pertence a coleção de dados não é conhecida, podendo esse sistema ser utilizado para encontrar agrupamentos naturais de dados. Este trabalho utilizará a segunda abordagem visto que trabalhará com um conjunto de dados (ontologias) onde não se conhece as categorias a que pertencem e se tem por objetivo agrupar objetos de acordo com suas características semelhantes.

Uma técnica muito importante e bem difundida na área de reconhecimento de padrões é a técnica de *clustering*, uma abordagem não-supervisionada que utiliza algoritmos para descobrir padrões a partir de alguma característica regular entre os objetos. O intuito é agrupar objetos semelhantes de acordo com alguma medida de similaridade, de modo que os elementos de um mesmo grupo (*cluster*) sejam mais semelhantes entre si e menos semelhantes entre objetos de outros grupos. Essa técnica classificatória pode ser utilizada quando se deseja explorar similaridades entre indivíduos ou entre variáveis, definindo-os em grupos [Everitt, 1980].

Para identificar todos os grupos, a partir de um conjunto de dados, é preciso tomar algumas decisões que podem influenciar diretamente no resultado final e que devem ser feitas de acordo com o tipo de dado tratado. Essas decisões são feitas em cada etapa da técnica de *clustering*. Aqui, serão definidas três etapas gerais para a técnica de *clustering* baseadas em [Metz e Monard, 2008]: (i) definição da técnica de agrupamento; (ii) definição da medida de similaridade; e (iii) definição do método de agrupamento a ser aplicado. Estas etapas serão descritas nas próximas seções.

2.1.1 Etapa I: Definição da técnica de agrupamento

O primeiro passo a ser realizado visando identificar os grupos, é definir qual técnica de agrupamento utilizar, se hierárquica ou não [Jain et al., 1999; Sander et al., 2003], cada uma utilizando uma maneira diferente para a identificação e representação dos *clusters*:

1. Técnicas não-hierárquicas: agrupam dados semelhantes onde os grupos não possuem relação entre si. O algoritmo k-means [Hamerly e Elkan, 2002] é um dos mais utilizados.
2. Técnica hierárquica: o resultado é obtido a partir de uma hierarquia que descreve um grupo diferente a cada nível de acordo com o grau de semelhança, possibilitando a análise dos *clusters* em diferentes níveis de granularidade [Metz e Monard, 2008].

Considerando a natureza dos dados utilizados nesse trabalho, cujas características não podem ser determinadas em termos de densidade de ocupação ou grupos previamente definidos, o **método adotado nesse trabalho utiliza uma técnica hierárquica**, que funciona da seguinte maneira [Kaufman, 1990]: (i) a partir da matriz inicial de dados, onde estão relacionados os dados e suas propriedades, obtém-se uma matriz de similaridade; (ii) em seguida, inicia-se a detecção de pares de casos com a mais alta similaridade (ou menor distância) para essa combinação, segundo níveis hierárquicos de similaridade; e (iii) escolhe-se um dentre os diversos procedimentos (aglomerativos ou divisivos), de tal modo que cada ciclo de agrupamento obedeça a uma ordem sucessiva no sentido do decréscimo de similaridade. O resultado final é um dendrograma (Figura 3), que consiste de um tipo especial de árvore, na qual os nós pais agrupam os objetos representados pelos nós filhos.

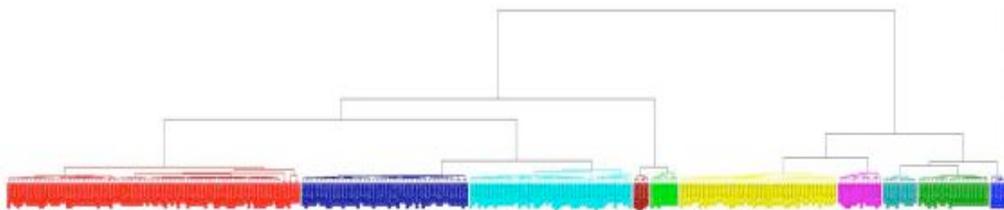


Figura 3. Exemplo de um dendrograma onde foram identificados dez grupos

Como exemplos de aplicações da técnica de *clustering* hierárquica estão as ferramentas ClustalW [Thompson et al., 1994] e XClust [Lee et al., 2002]. A primeira está voltada para o alinhamento de seqüências biológicas, um problema NP-completo que se utiliza dessa técnica para melhorar a qualidade do alinhamento final do conjunto de seqüências, e que, por se tratar de uma heurística, nada garante que o resultado final seja o ótimo. Nesse contexto, a abordagem hierárquica também é conhecida como alinhamento progressivo [Feng e Doolittle, 1987]. A segunda ferramenta trabalha na mesma idéia da primeira, mas, ao invés de seqüências biológicas, os dados tratados são DTDs (*Document Type Definition*) e o cálculo da similaridade é realizado através de métodos de *match* descritos para esse tipo de dado [Doan et al., 2001; Su et al., 2001].

Tendo em vista a técnica adotada neste trabalho, as etapas a seguir serão descritas com base exclusivamente nas técnicas e abordagens hierárquicas.

2.1.2 Etapa II: Definição de uma medida de semelhança (dessemelhança) entre objetos

Com a técnica de agrupamento definida, é preciso escolher a forma de calcular a distância ou semelhança entre os objetos. Esses objetos podem ser considerados como pontos no espaço onde, para cada um, são definidas as propriedades (dimensões) relevantes a serem utilizadas. Essas propriedades são representadas em uma matriz inicial que é utilizada para obter a distância entre tais objetos. Um exemplo de matriz inicial pode ser vista na Figura 4.A onde “Gene A”, “Gene B” e “Gene C” são os objetos de estudo e “Exp 1”, “Exp 2”, “Exp 3” e “Exp 4” são as propriedades selecionadas para calcular suas distâncias.

	Exp 1	Exp 2	Exp 3	Exp 4
Gene A	1.0	2.0	3.0	4.0
Gene B	5.0	4.0	1.0	8.0
Gene C	6.0	5.0	4.0	2.0

➔

	Gene A	Gene B	Gene C
Gene A	0	6.3	6.2
Gene B	6.3	0	6.8
Gene C	6.2	6.8	0

Figura 4. Cálculo da Matriz de Distância

Existem várias formas de se calcular a distância entre esses pontos, sendo a mais comum a distância Euclidiana. No entanto, tal escolha dependerá dos tipos de objetos estudados. A Figura 4.B ilustra um exemplo da criação de uma matriz de similaridade a partir do cálculo das distâncias entre os objetos em função das propriedades (Figura 4.A) utilizando a distância Euclidiana.

No caso deste trabalho, onde os dados são ontologias, podem-se utilizar soluções de *matching* existentes na literatura como [Euzenat, 2004] e [Castano et al.,2003] para identificar a similaridade entre as ontologias e então criar a matriz de similaridade. Nas ontologias, as propriedades comparadas serão os nomes dos elementos que as compõem e a posição dos mesmos na estrutura.

2.2.3 Etapa III: Definição do método de agrupamento ou de construção dos clusters

Na terceira etapa deve-se decidir qual método utilizar para o agrupamento. Nesse sentido, os métodos podem ser classificados de duas maneiras como mostrado na Figura 5 e descritos a seguir [Anderberg et al., 1993]:

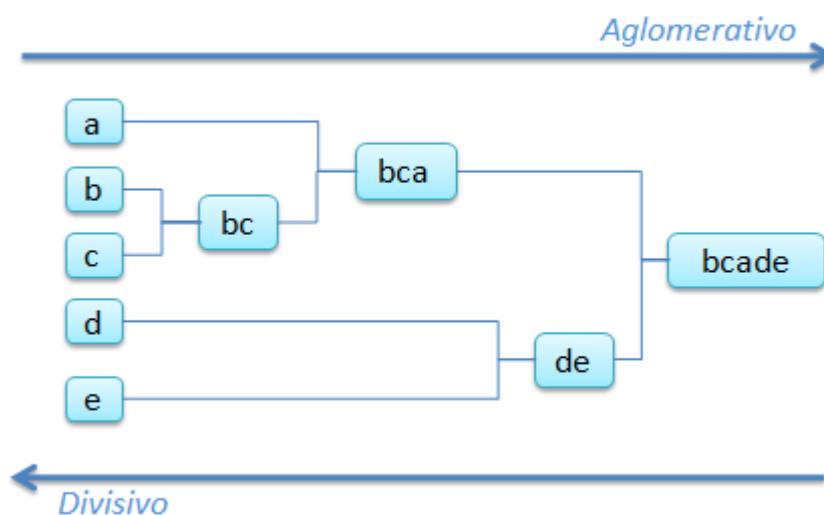


Figura 5. Abordagens hierárquicas

- i. Aglomerativos: inicia considerando cada objeto como um único grupo. Assim, no primeiro momento, para N objetos existirão N grupos. Esses grupos vão sendo unidos de acordo com a medida de similaridade encontrada, formando novos grupos até chegar a um único grupo. Na primeira iteração, os métodos precisam computar semelhanças de todos os pares para n objetos: $O(n^2)$. Em cada uma das subsequentes iterações de uniões ($n-2$), deve-se computar a distância entre o último *cluster* criado e todos os outros *clusters* existentes.
- ii. Divisivos: de forma oposta ao anterior, começa com um grande grupo composto por todos os objetos, subdividindo-se em grupos menores até chegar a um grupo com apenas um único objeto. Os métodos divisivos são pouco mencionados na literatura, pois exigem uma maior capacidade computacional que os métodos aglomerativos [Kaufman, 1990]. A complexidade de tempo é da ordem de $O(2^n)$.

Devido à complexidade das duas abordagens, que é $O(n^2 \log n)$ para os aglomerativos e $O(2^n)$ [Jain et al.1999] para os divisivos, o método mais utilizado e que será adotado nesse trabalho é o aglomerativo, o qual segue, de forma geral, o Algoritmo 1.

Algoritmo Aglomerativo

Entrada: N elementos

Saída: Um conjunto de grupos

1. Iniciar com N grupos, contendo um elemento em cada grupo
2. Iniciar uma matriz de similaridade $\mathbf{M}_{N \times N}$
3. Repetir
4. Localizar menor distância d_{IJ} entre dois elementos I e J
5. Atualizar matriz de distância \mathbf{M} retirando os elementos \mathbf{I} e \mathbf{J}
6. Atualizar a matriz \mathbf{M} adicionando as novas distâncias do grupo (\mathbf{I}, \mathbf{J})
7. Até N-1, quando todos os elementos estarão em um único grupo

Algoritmo 1. Algoritmo Genérico para Métodos Aglomerativos

O passo (5) é onde se diferencia cada algoritmo para cálculo de distância entre os grupos, que pode ser utilizado para o método aglomerativo [Johnson e Wichern, 1992]. Nesse sentido, várias soluções foram propostas como, por exemplo, o *Single Link* (vizinhos mais próximos), que determina a distância entre os grupos pela menor distância entre os objetos que compõem esses grupos; o *Complete Link* (vizinhos mais distantes), que calcula as distâncias entre os grupos pela maior distância encontrada entre os objetos dos grupos; e o *Average link* (média ponderada), que calcula a distância entre os dois grupos pela distância média entre todos os pares dos objetos nos dois grupos, adicionando pesos aos objetos. Uma comparação entre esses e outros métodos pode ser vista em [Kaufman, 1990].

2.2 Ontologias

As ontologias são recursos fundamentais quando se faz necessário o conhecimento explícito da semântica dos dados, a qual pode ser usada, por exemplo, para facilitar a comunicação e o compartilhamento de informações entre diferentes sistemas.

As ontologias podem ser classificadas de diferentes formas na literatura. Entre estas classificações, encontra-se a proposta por [Heijst et al., 1997], que enumera os seguintes tipos de ontologias: i) ontologia de topo, descrevendo conceitos mais gerais, aplicados a diversas áreas de conhecimento; ii) ontologia de domínio, descrevendo conceitos específicos de um domínio ou área de conhecimento; e iii) ontologia de tarefa, descrevendo conceitos utilizados na solução de um determinado problema ou tarefa. As ontologias utilizadas nesse trabalho serão ontologias de domínio.

Uma ontologia é formada por diferentes componentes, utilizados para representação de um determinado domínio de interesse [Staab e Studer, 2004]. Uma ontologia \mathbf{O} pode ser definida como

uma 4-tupla: $O = (C, P, I, K)$ onde C é o conjunto de conceitos ou classes; P são as propriedades (relações binárias) que podem ser propriedades de objetos (*ObjectProperty*), propriedades de tipos de dados (*DataTypeProperty*) ou propriedades de anotações (*AnnotationProperty*); I é o conjunto de indivíduos ou instâncias, que representa os objetos de uma classe; e K é o conjunto de axiomas, que podem ser utilizados para inferir conhecimentos.

As ontologias podem ser usadas para minimizar o problema da heterogeneidade semântica dos dados encontrados na Web. Assim, uma ontologia pode ser utilizada, dentre outros propósitos, para o suporte à interoperabilidade de fontes de dados distribuídas e heterogêneas. Essa heterogeneidade semântica pode ocorrer tanto devido ao fato dos itens da informação possuírem o mesmo significado aparente, mas realidades diferentes (homônimos); como quando os nomes diferenciam sintaticamente, mas possuem o mesmo significado (sinônimos) [Goh, 1997].

Atualmente, são desenvolvidos e publicados muitos estudos com ontologias, em particular, as ontologias biológicas (ou bio-ontologias). Um dos problemas existentes na atualidade, e que é um ponto relevante neste trabalho, é a resolução dos problemas de heterogeneidade. A seguir serão apresentados os processos que envolvem o uso de ontologias rumo à redução da heterogeneidade: o *matching* e o *merging* de ontologias.

2.2.1 Técnicas de *Matching* de Ontologias

O processo de *matching* de ontologias consiste em comparar duas ontologias à procura de um conjunto de relações ou correspondências satisfatórias entre os seus elementos. Essas correspondências, por sua vez, são identificadas a partir de uma medida de similaridade entre cada elemento.

Formalmente, de acordo com [Euzenat e Shvaiko, 2007], o processo de *matching* pode ser definido como: *Sejam dois conjuntos de elementos O_1 e O_2 e uma função de similaridade $f: O_1 \times O_2 \rightarrow [0,1]$. O processo de *matching* consiste em achar um conjunto de correspondências (relações válidas) A onde $A \subseteq O_1 \times O_2$. Esse conjunto de correspondências é a saída do processo definido como alinhamento.*

Pela definição de *matching*, vê-se, então, que o alinhamento é um subconjunto do produto cartesiano das ontologias, ou seja, dentre todas as correspondências possíveis entre os elementos das duas ontologias, o alinhamento final possui apenas correspondências consideradas válidas.

De forma geral, o processo de *matching* é realizado em três etapas: (i) **importação de duas**

ontologias; (ii) identificação de similaridade, e (iii) especificação das correspondências válidas. Cada etapa será brevemente descrita a seguir.

Etapa i – Importação de duas ontologias

Esta primeira etapa consiste em receber e reconhecer duas ontologias (O_1 e O_2) como uma entrada válida, conforme o seu formato que pode ser, por exemplo, OWL² ou RDF³. O formato aceito neste trabalho será OWL, uma linguagem de maior expressão que a RDF, recomendação da W3C, e que tem como objetivo facilitar o processamento de conteúdo semântico dos dados, sendo uma evolução da XML, RDF e RDFS. Caso as ontologias de entrada se adequem ao formato aceito pela solução, a segunda etapa é iniciada. A Figura 6 ilustra um exemplo de duas ontologias, onde os círculos representam as classes e os retângulos representam as propriedades das classes.

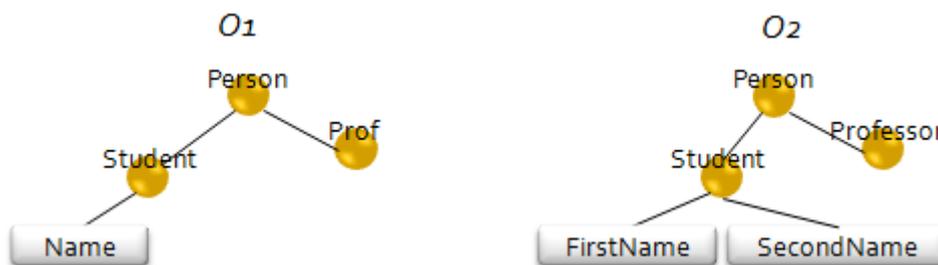


Figura 6. Exemplo de duas Ontologias de entrada

Etapa ii – Identificação de Similaridade

A segunda etapa calcula os valores que representam as semelhanças entre os elementos das duas ontologias ($O_1 \times O_2$), em conformidade com a função de similaridade adotada (*Definição 1*), que pode levar em consideração diferentes níveis da ontologia (nível de elemento e nível conceitual [Euzenat e Shvaiko, 2007]). Ao final é gerada uma estrutura, normalmente uma matriz, que representa todas as correspondências possíveis entre os elementos das ontologias e suas similaridades. A Figura 7 ilustra um exemplo de uma matriz de correspondências, com valores hipotéticos, em que a primeira coluna indica os elementos de uma ontologia O_1 e a primeira linha os elementos de uma ontologia O_2 , a ser comparada com a primeira. Os valores identificados são os valores de similaridade calculados a partir de uma função de similaridade qualquer.

² <http://www.w3.org/TR/owl-ref/>

³ <http://www.w3.org/TR/rdf-schema/>

$O_1 \setminus O_2$	Person	Professor	Student	FirstName	SecondName
Person	1.0	0.6	0.65	0.45	0.4
Prof	0.6	0.9	0.2	0.4	0.3
Student	0.65	0.2	1.0	0.4	0.3
Name	0.33	0.2	0.3	0.8	0.7

Figura 7. Exemplo de uma matriz de correspondências

Técnicas de análise em nível de elementos comparam os elementos isoladamente, ignorando sua relação com outros elementos. A comparação é feita baseada em textos, podendo ser computada sintaticamente ou semanticamente. Na comparação sintática, a computação da similaridade pode ser realizada de acordo com a string através do *match* exato ou utilizando algoritmos que verifiquem quão próximas são as strings. A técnica mais conhecida para isso é a de edição de distância (*EditDistance*), na qual são verificadas as operações (inserção, deleção, substituição) necessárias para transformar um elemento em outro. Como exemplos de métodos de edição de distância estão *Levenstein* [Levenstein, 1966], *Needleman-Wunsch* [Needleman & Wunsch, 1970], *Smith-Waterman* [Smith & Waterman, 1981] e *SMOA* [Stoilos et al 2005]. Já a comparação semântica é baseada nas relações terminológicas entre os elementos (sinônimos, hiperônimos e hipônimos) através do auxílio de um *thesaurus* como o *WordNet*⁴.

A Tabela 1 fornece um resumo das categorias existentes para comparação de ontologias em nível de elementos. As categorias são: baseadas em strings, lingüísticas e restrições.

⁴ <http://wordnet.princeton.edu/doc>

Tabela 1. Resumo de técnicas de comparação de elementos

<i>Técnicas baseadas em String</i>		
Descrição	Descrição	Exemplo
<i>Prefix</i>	Recebe duas strings de entrada e verifica se a primeira começa com a segunda.	String1: <u>Int</u> String2: <u>Integer</u>
<i>Suffix</i>	Recebe duas strings de entrada e verifica se a primeira termina com a segunda.	String1: <u>Fone</u> String2: <u>Telefone</u>
<i>Edit Distance</i>	Recebe duas strings de entrada e calcula quantas operações de inserção, remoção ou substituição são requeridas para transformar uma string na outra, normalizados pela tamanho da maior.	String1: NKN String2: NIKON operações:insere(I),insere(O) Distância: 2/5
<i>N-gram</i>	Pega o conjunto de substrings de uma string e compara com a outra string. Normalmente as substrings tem tamanho 3, sendo 3-gram .	String1: NIKON String2: NKON substrings de 1: nik, iko, kon distância: 1/3
<i>Técnicas baseadas em lingüística</i>		
<i>Tokenization</i>	Baseadas em tokens. As palavras são colocadas em seqüências de tokens.	String: hands_free_kits tokens: <hands,free,kits>
<i>Lemmatization</i>	Baseada em análise morfológica para identificar as formas possíveis de uma string.	String: Kits Possibilidades: Kits, Kit
<i>Elimination</i>	Os tokens que são artigos, preposições, conjunções e assim por diante, são marcados para serem descartados da string.	(um, a, por, de)
<i>Técnicas baseadas em restrições</i>		
<i>Comparação de datatypes</i>	Compara os tipos de dados.	Integer < Real

Técnicas de análise em nível de estruturas comparam cada elemento de acordo com sua posição na estrutura da ontologia e sua relação com os demais elementos. Na abordagem estrutural, a comparação entre os elementos é realizada pela análise de como estes elementos estão dispostos na estrutura (p.ex: posição dentro da estrutura hierárquica). Para tal fim, foram propostos diversos algoritmos baseados na taxonomia, onde são levadas em consideração apenas as relações de especialização; e baseados em grafos, onde é levada em consideração a posição no grafo. Como exemplo de soluções podem ser citadas a AnchorPrompt [Noy, 2001] e a OLA (OWL Lite Aligner) [Euzenat, 2003].

A Tabela 2 fornece um resumo de algumas soluções para comparação estrutural de ontologias de acordo com as duas categorias citadas. Essas soluções podem ser combinadas com as soluções do nível de elementos para obter um resultado mais confiável.

Tabela 2. Técnicas de comparação estrutural

<i>Técnicas baseadas em taxonomia</i>	
<i>Matching</i> de caminho delimitado	A obtenção de dois caminhos que ligam duas classes definidas pela relação hierárquica é seguida da comparação dos termos e de suas posições ao longo do caminho e depois da identificação dos termos similares.
Regras de super e sub-classe	Se os super (ou sub) conceitos são os mesmos, os conceitos atuais são similares um com o outro.
<i>Técnicas baseadas em grafos</i>	
<i>Children</i>	Dois elementos não folhas são estruturalmente similares se seus filhos (<i>Children</i>) imediatos são similares.
<i>Leaves</i>	Dois elementos não folhas são estruturalmente semelhantes se seus nós folhas (<i>Leaves</i>) são similares, caso seus filhos imediatos não são

Uma visão geral de todas essas medidas de ambos os níveis podem ser encontradas em [Shavaiko e Euzenat, 2005].

Etapa iii – Especificação das correspondências válidas

A terceira etapa do processo de *matching* consiste na definição de quais das correspondências identificadas na segunda etapa são realmente válidas para, a partir daí, gerar o alinhamento, ou seja, o conjunto de correspondências válidas.

Para identificar as correspondências válidas a partir da matriz de correspondências, é aplicado um filtro, como por exemplo, o *threshold*, visando limitar o valor de similaridade entre os elementos para considerar uma correspondência válida. Essa similaridade também é definida em [Euzenat, 2004] como um valor de confiança entre os elementos. Além do filtro, outro fator que determina o alinhamento resultante é a direção adotada para o alinhamento, que pode gerar um conjunto de correspondências diferentes. A direção de um alinhamento está relacionada à definição de qual ontologia é a preferencial ou usada como referência na busca por correspondências.

Considerando a matriz exemplo da Figura 7, suponha que o *threshold* adotado seja 0.5. Dessa forma, as correspondências válidas serão as que possuem similaridade maior que esse valor. A Figura 8 mostra os dois possíveis resultados do *matching* de acordo com esse valor de *threshold* e com a direção adotada para alinhamento. O alinhamento ‘O1 → O2’ indica que a ontologia O1 foi definida como preferencial, então a busca por correspondências parte dos elementos dessa ontologia. Já o alinhamento ‘O2 → O1’ indica que a ontologia O2 é a ontologia preferencial.

O1 → O2			O2 → O1		
Person	Person	1.0	Person	Person	1.0
Prof	Professor	0.9	Professor	Prof	0.9
Student	Student	1.0	Student	Student	1.0
Name	FirstName	0.8	FirstName	Name	0.8
			SecondName	Name	0.7

Figura 8. Exemplo de duas possíveis matrizes de correspondências

Note que, apesar de o elemento “*Person*” de O_1 ter um valor maior que ‘0,5’ para mais de um elemento, a correspondência válida é apenas com o elemento “*Person*” de O_2 . Isso ocorre porque o alinhamento considerado tem cardinalidade 1:1, ou seja, para cada elemento existirá apenas uma correspondência. Assim, ao identificar que existe mais de uma correspondência válida para um elemento, escolhe-se apenas o elemento com maior valor relacionado. Normalmente um alinhamento de cardinalidade 1:n, ou seja, um alinhamento onde para cada elemento de uma ontologia pode existir mais de um elemento correspondente em outra ontologia, ocorre quando há interferência humana.

A Figura 9 ilustra um exemplo simplificado de um alinhamento utilizando o formato descrito em [Euzenat, 2004] onde as *tags* ‘*onto1*’ e ‘*onto2*’ dão informações sobre as ontologias envolvidas (localização e formato) e as *tags* ‘*map*’ dão informações sobre as correspondências como elementos relacionados (<entity/>), relação entre os elementos (<relation/>) e valor de similaridade ou confiança (<measure/>).

```

<Alignment>
  <onto1>
    <Ontology rdf:about="O1.owl">
      <location>file:/ontologies/O1.owl</location>
      <formalism>
        <Formalism align:name="OWL1.0" align:uri="http://www.w3.org/owl#"/>
      </formalism>
    </Ontology>
  </onto1>
  <onto2>
    <Ontology rdf:about="http://www.owl-ontologies.com/O2.owl">
      <location>file:/ontologies/O2.owl</location>
      <formalism>
        <Formalism align:name="OWL1.0" align:uri="http://www.w3.org/owl#"/>
      </formalism>
    </Ontology>
  </onto2>
  <map>
    <Cell>
      <entity1 rdf:resource='O1.owl#Person'/>
      <entity2 rdf:resource='O2.owl#Person'/>
      <relation></relation>
      <measure rdf:datatype='http://www.w3.org/XMLSchema#float'>1.0</measure>
    </Cell>
  </map>

```

```

<map>
  <Cell>
    <entity1 rdf:resource='01.owl#Student' />
    <entity2 rdf:resource='02.owl#Student' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/XMLSchema#float'>1.0</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='01#Name' />
    <entity2 rdf:resource='02.owl#FirstName' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/XMLSchema#float'>0.8</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='01.owl#Prof' />
    <entity2 rdf:resource='02.owl#Professor' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/XMLSchema#float'>0.9</measure>
  </Cell>
</map>
</Alignment>
</rdf:RDF>

```

Figura 9. Exemplo de alinhamento

O resultado da terceira etapa, então, é o conjunto de correspondências válidas que dependerá do limite de similaridade adotado e da direção do alinhamento.

2.2.2 Técnicas de *Merging* de Ontologias

Existe uma grande quantidade de ontologias disponíveis que podem ser compartilhadas entre diferentes sistemas, sendo muitas delas representações de um mesmo domínio. Dependendo da aplicação, pode se tornar necessário ter uma ontologia que represente uma visão global (integrada) dessas diversas ontologias. Para a construção dessa visão pode-se utilizar o processo de *merging*, que gera uma nova ontologia a partir da combinação de duas ontologias iniciais, as quais permanecem inalteradas; sendo que a nova ontologia contém os conhecimentos de ambas ontologias [Euzenat e Shvaiko, 2007]. Além de proporcionar uma visão global, o processo de *merging* ajuda no enriquecimento de um determinado domínio.

Para a obtenção da ontologia global, o processo de *merging* segue algumas etapas que, neste trabalho, serão definidas a seguir: (i) importação das ontologias iniciais ou ontologias fonte, (ii) a busca por similaridade, (iii) especificação de correspondências válidas e (iv) combinação das ontologias fontes, gerando uma nova ontologia (visão global). Como pode ser observado, as etapas (i), (ii) e (iii) representam o processo de *matching* já discutido, que é, então, o início do processo de *merging*. Logo, o processo de *merging* é diretamente influenciado pela qualidade do resultado do

processo de *matching*. Se este é ineficiente, o *merging* poderá gerar um resultado incoerente. A Figura 10 ilustra o processo de *merging* onde, inicialmente, ocorre o *matching* e ao final as ontologias são combinadas, gerando uma visão global das mesmas.

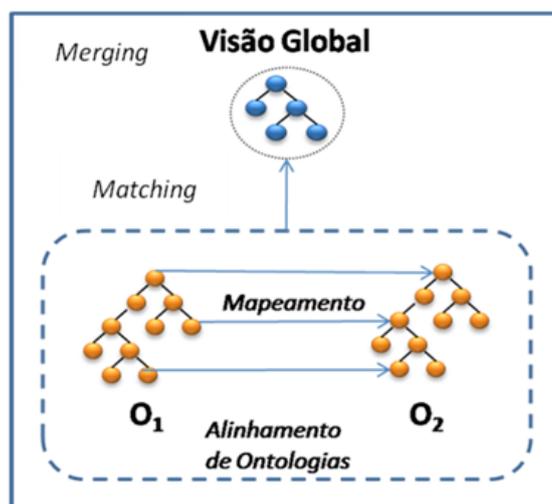


Figura 10. Processo de Merging de Ontologias

Existem duas abordagens para o processo de *merging* [Davies et al 2006]. Na primeira (Figura 11.A), conhecida como *Merge Completo*, as ontologias fonte são combinadas e uma nova ontologia é gerada. Nessa abordagem, não é possível identificar a origem de cada elemento na ontologia resultante, uma vez que as suas referências são perdidas durante o processo de *merging*. Como exemplos de soluções que utilizam essa abordagem destacam-se: iPROMPT [Noy e Musen, 2002], FCA-Merge [Stumme e Madche, 2001], OMAAlgorithm [Rasgado e Guzman, 2006], COMA++ [Aumüller et al 2005] e ILIADS [Udrea et al., 2005]. Na segunda abordagem (Figura 11.B) a nova ontologia é composta por um conjunto de axiomas associados que especificam os elementos relacionados entre as ontologias, e as duas ontologias iniciais. Como exemplos de soluções que utilizam essa segunda abordagem estão: Alignment API[Euzenat, 2004], HCONE-merge[Kotis et. Al. 2006] e OntoMerge[Dou et al., 2005]. A abordagem aplicada neste trabalho é a de união, visto que não existe o interesse em preservar as referências para os elementos originais.

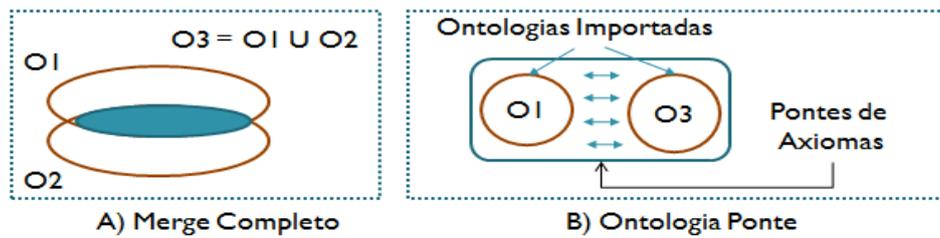


Figura 11. Abordagens de *merging* de ontologias. Retirada de [Davies et al 2006]

A Figura 12 ilustra um exemplo de duas ontologias do domínio *University*. Pelo processo de *merging*, inicialmente são identificados os elementos correspondentes (*matching*) para em seguida estes serem combinados. Os elementos comuns nesse exemplo são ligados pelas setas que identificam as correspondências.

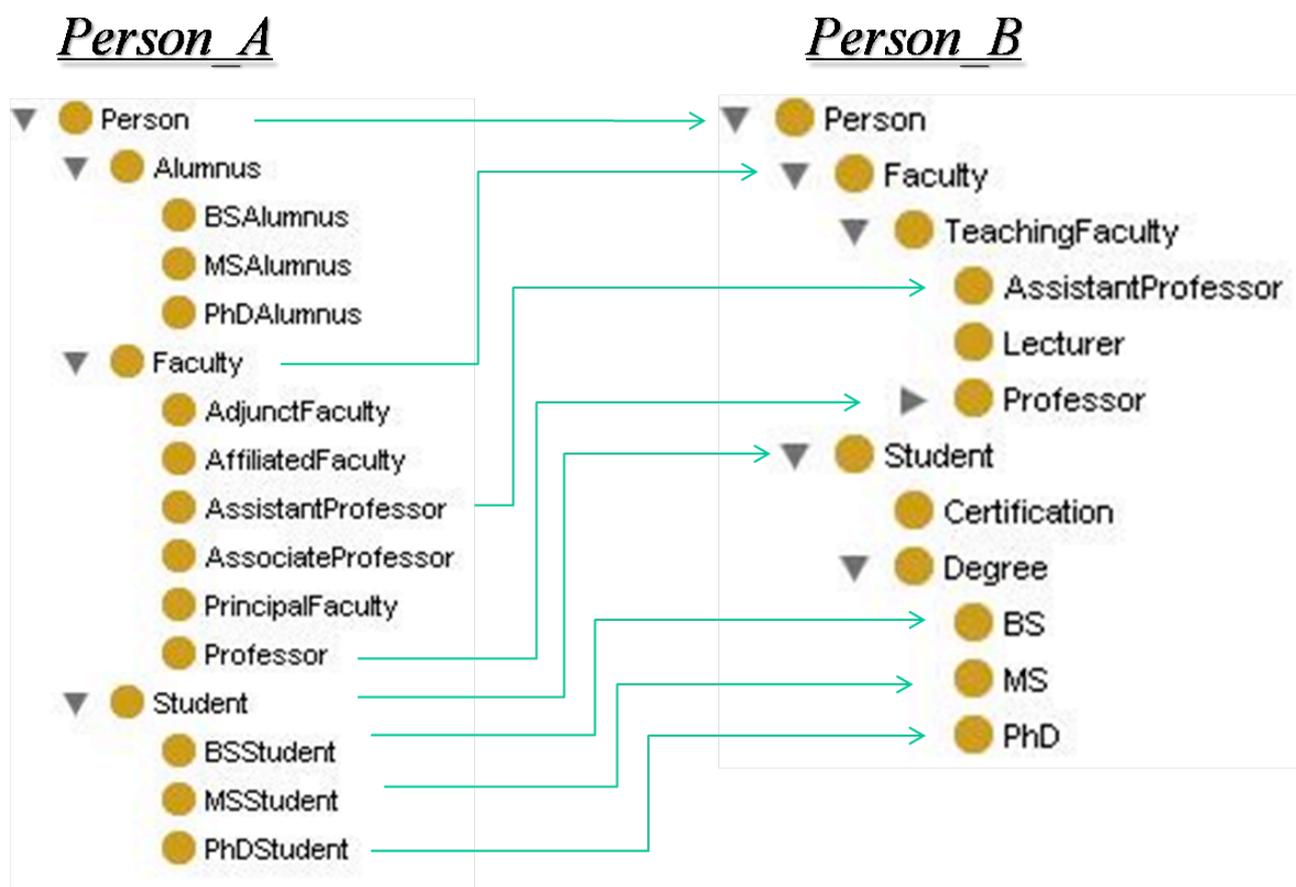


Figura 12. Exemplo de duas ontologias do domínio *University*

A Figura 13.A mostra como poderia ser o resultado da combinação das duas ontologias da Figura 12 caso fosse aplicada a primeira abordagem (*Merge Completo*). Na ontologia resultante os

elementos são unificados, não existindo uma forma palpável de identificar a origem de cada um deles. A ferramenta utilizada para gerar esse exemplo foi iPROMPT (*plugging* do Protégé⁵), o qual utiliza o artifício de colocar como nome dos elementos da nova ontologia a concatenação do nome dos elementos da ontologia de origem juntamente com o nome da ontologia (novo_nome = nome_original + nome_da_ontologia_de_origem), como pode ser visto, por exemplo, no elemento *Alumnus--Person_A* e *Degree--Person_B*. No primeiro exemplo, *Alumnus* é o nome do elemento utilizado na ontologia de origem e *Person_A* é o nome da ontologia de origem. Já no segundo exemplo, o elemento é a concatenação do elemento *Degree* e o nome da ontologia de origem *Person_B*. Note que sem esse artifício não seria possível identificar a origem de cada elemento. Na Figura 13.B a ontologia ilustrada representa o resultado do *merge* das duas ontologias da Figura 12 utilizando a segunda abordagem (Ontologia Ponte). A ferramenta utilizada para gerar esse resultado foi a Alignment API [Euzenat, 2004]. Pode ser observado que é possível identificar a origem de cada elemento através do *namespace* 'p1'; e a qual o elemento da segunda ontologia ('p2') está relacionado.

A) Merge Completo



B) Ponte de Axiomas



Figura 13. Exemplos de merging de ontologias: A) Merge Completo; B) Ontologia Ponte

⁵ <http://protege.stanford.edu/>

É possível que, durante o processo de *merging*, na etapa de combinação (após o *matching*) alguns conflitos apareçam. Vale ressaltar que a solução para tais conflitos pode variar de acordo com a abordagem adotada. Em alguns casos, esses problemas são passados para o usuário para que ele decida o que fazer como no iPROMPT [Noy e Musen, 2002]. Em outros casos, são definidos métodos para resolver esses conflitos, como o que ocorre na solução proposta em [Rasgado and Guzman, 2006] onde, em casos de dúvidas, o sistema recorre a auxílios (ex: WordNet) para a solução de cada conflito. Dentre os vários conflitos podem ser destacados os seguintes tipos:

- Conflito de Nomes: dois elementos a serem combinados são equivalentes (têm mesmo significado), mas têm nomes diferentes;
- Conflito de Referência: um elemento de uma das ontologias fonte é adicionado à nova ontologia e faz referência a outro elemento que ainda não foi adicionado. Essa referência pode ser em termos de herança (um filho apontando para um pai que ainda não foi adicionado) ou relação de propriedade (uma propriedade tem como domínio uma classe que ainda não foi adicionada);
- Conflito de Redundância: um elemento de uma das ontologias fonte é adicionado à nova ontologia, mas esta já possui um elemento idêntico;
- Violação de restrições:
 - Duas propriedades são equivalentes e possuem tipos de dados diferentes (ex: uma propriedade é do tipo real e outra é do tipo inteiro);
 - Restrição com definições diferentes (ex: diferentes limites para valores máximos ou mínimos);
- Restrição de representação: dois elementos a serem combinados são heterogêneos, isto é, diferentes representações de conceitos (um é classe e outro é propriedade, ou um é propriedade de objeto e outro é propriedade de tipo de dado, por exemplo);

Independente da abordagem de *merging*, de *matching* ou das soluções adotadas para os conflitos, uma ontologia O_c gerada a partir da combinação de duas outras (O_i , O_j) deve obedecer a alguns requisitos para que o resultado seja considerado de boa qualidade [Pottinger and Bernstein, 2003]:

1. Preservação de elementos: cada elemento e_i de uma ontologia O_i ou um elemento e_j de uma ontologia O_j corresponde a exatamente um elemento e_c de uma ontologia O_c , gerada a partir da junção das ontologias fonte O_i e O_j ;

2. Preservação de igualdade: um elementos e_i de uma ontologia O_i e um elemento e_j de uma ontologia O_j são ditos equivalentes se existir um único elemento e_c de uma ontologia O_c , tal que e_c é resultado da combinação de e_i e e_j ;
3. Preservação da relação: cada relação encontrada entre dois elementos relacionados deve ser a mesma relação entre os elementos correspondentes em O_c ;
4. Preservação da propriedade: todo elemento e_c de uma ontologia O_c , gerada a partir do *merging* das ontologias fonte O_i e O_j , tem uma determinada propriedade se seus correspondentes em O_i ou O_j também possuírem essa propriedade;
5. Satisfação de restrições: O_c satisfaz todas as restrições existentes em O_i e O_j .

Como pôde ser visto, as etapas mais custosas e que são determinantes para o processo de *merging* estão relacionadas às etapas do processo de *matching*. Nenhum desses dois processos são problemas resolvidos. Muitos estudos ainda vêm sendo realizados no sentido de aperfeiçoá-los. A seguir serão apresentadas, de forma resumida, algumas das soluções disponíveis na literatura que foram analisadas para esse trabalho e direcionadas a esses processos.

2.2.3 Soluções para *Matching* e *Merging* de Ontologias

Muitas ferramentas vêm sendo desenvolvidas buscando facilitar o processo de *matching* e *merging* de ontologias. De forma geral, as soluções utilizam métodos léxicos combinados com estruturais para obter um resultado mais satisfatório. Para o desenvolvimento deste trabalho foram estudadas algumas das principais soluções disponíveis na literatura relacionadas a *matching* e *merging* de ontologias. Essas soluções serão sumarizadas no restante desta seção e um quadro comparativo será apresentado em seguida.

I - PROMPT [Noy and Musen, 2000]:

O PROMPT é formado por um conjunto de ferramentas com o objetivo de manipular ontologias, envolvendo operações de alinhamento, mapeamento e combinação de ontologias.

O **AnchorPROMPT** [Noy and Musen, 2001] é a ferramenta responsável pelo alinhamento utilizando técnicas léxicas e estruturais, sendo baseada em grafos cujos caminhos representam os relacionamentos entre os elementos (*anchor*). Inicialmente é feita uma análise léxica entre os nomes das ontologias comparadas, apresentando para o usuário os mapeamentos candidatos que podem ser alterados ou confirmados. Com os mapeamentos candidatos, faz-se a análise estrutural entre os

vizinhos podendo passar para o usuário novos mapeamentos mais refinados.

O **iPROMPT** [Noy and Musen, 2002] é responsável pela atividade de *merging*. Essa ferramenta recebe duas ontologias de entrada e realiza uma comparação léxica entre as mesmas. Cada possível combinação é passada para o usuário decidir qual operação será realizada. Nos casos de conflito, é passado para o usuário o conflito em questão e sugestões para resolvê-lo. Os conflitos que podem aparecer são: de nome, de referência, de redundância, de violação das restrições dos valores de propriedades. Assim, como exemplo de ações, o usuário pode executar *copy*, onde ele copia um elemento de uma ontologia para a ontologia a ser criada, ou *merge*, onde ele combina dois elementos semelhantes de cada ontologia fonte para a ontologia resultante.

Além dessas duas ferramentas, esta também existe o PromptDiff, responsável pela identificação de diferenças entre duas versões da mesma ontologia, e o PromptFactor que extrai porções de uma ontologia.

II - HMatch [Castano et al 2003; Castano et al., 2007]

Esta é uma ferramenta voltada para *matching* de ontologias que permite combinar diversas técnicas para melhorar seu resultado final. Faz busca por correspondências de forma automatizada e permite a descoberta de conhecimentos. Ela recebe como entrada duas ontologias e tem como saída as correspondências entre os conceitos da ontologia que tem o mesmo significado ou significados próximos. Utiliza a comparação lingüística e semântica, combinada com técnicas baseadas em grafos para refinar o alinhamento. Além disso, esta ferramenta é capaz de comparar os indivíduos de uma ontologia e combinar alinhamentos.

III – AUTOMS [Kotis et al 2006]

Esta é uma API desenvolvida para facilitar a criação de novas ferramentas que manipulem ontologias. O objetivo é realizar o alinhamento automático de ontologias, onde são utilizadas tanto técnicas léxicas (por exemplo, a COCLU – explora nomes, rótulos e comentários) juntamente com o WordNet, como técnicas estruturais; podendo ainda fazer a combinação de várias técnicas para melhorar a qualidade do alinhamento. Essa ferramenta compara tanto classes como propriedades e instâncias, sendo a similaridade entre propriedades de objetos baseada em seus domínios e *ranges*. Ao final, com todas as correspondências candidatas identificadas, realiza-se uma nova iteração para refinar o resultado.

IV – HCONE-merge [Kotis et al 2006]

Essa é uma ferramenta voltada para *merging* de ontologias, onde são utilizadas abordagens

lingüísticas e estruturais que são exploradas pelo método LSI (*Latent Semantics Indexing*) para associar a interpretação informal de acordo com a interpretação humana realizada com o WordNet. Esse método transforma definições de conceitos formais para um vocabulário comum e explora essas definições através de serviços de raciocínio da lógica descritiva. O objetivo é validar os mapeamentos entre ontologias e achar um conjunto mínimo de axiomas para o *merge* de ontologias. São construídas duas ontologias intermediárias relativas às ontologias de entrada, com elementos correspondentes da ontologia original com o sentido encontrado no WordNet e com axiomas traduzidos da ontologia original. A partir dessas duas ontologias intermediárias são encontrados os mapeamentos e axiomas, podendo ambas as ontologias serem combinadas.

V – ILIADS [Udrea et al 2007]

Ferramenta voltada para a integração de ontologias, combinando aprendizado estatístico e inferência lógica. Utiliza técnica de *cluster* para agrupar elementos semelhantes e descobrir a relação entre os elementos e indivíduos relacionados. É uma ferramenta ideal quando se trata de ontologias que possuem indivíduos, conseguindo fazer um raciocínio entre os elementos e descobrir novos relacionamentos e ter um resultado final, uma ontologia integrada, de melhor qualidade. Como é baseado em lógica descritiva tem um processamento mais demorado.

VI - Alignment API [Elzenat, 2004]

É uma API que visa atender às necessidades relacionadas ao processo de *matching* automático de ontologias. Esta API gera um alinhamento em RDF em um formato específico, podendo combinar comparações léxicas com estruturais (OLA - OWL Lite Aligner [Euzenat, 2003]), usar o WordNet e alinhamentos pré-computados para melhorar o resultado final do alinhamento. Essa API permite a inserção de novos métodos de alinhamento que se queira adicionar. Com essa API é possível visualizar a matriz de correspondências, além de gerar axiomas a serem utilizados no *merging* de ontologias.

VII - FCA-Merge [Stumme and Madche, 2001]

Este é um método para *merge* de ontologias baseado em técnicas de análise de conceito formal [Wille, 1982; Ganter and Wille, 1999] onde são usadas instâncias específicas da aplicação de duas ontologias que precisam ser combinadas. Para as duas ontologias iniciais são extraídos um conjunto de instâncias a partir de documentos texto de um domínio específico e gerado uma estrutura de conceitos, a partir da qual é possível gerar uma nova ontologia. Desse modo, o processo de *merge* consiste em três passos: i) extração de instâncias a partir de documentos de

linguagem natural, onde é feita a análise morfológica das palavras, reconhecimento dos nomes das entidades e dos tipos de palavras, e recuperação específica do domínio, tendo como resultado contextos formais relativos às instâncias; ii) em seguida é computado o conjunto de conceitos que tem grau parecido de detalhes das duas ontologias; (iii) no último passo, a ontologia combinada é derivada a partir dos conceitos, ocorrendo de forma semi-automática e precisando da interação com o usuário que define o conjunto de relações e ajustes necessários para a conclusão.

VIII - OM Algorithm [Rasgado and Guzman, 2006]

Este é um algoritmo de *merge* que faz a operação de combinação de ontologias de forma automática. Este algoritmo procura o elemento mais similar da primeira ontologia em relação à segunda ontologia. Ao achar esse elemento é criado, então, um terceiro elemento enriquecido com sinônimos que representará os dois elementos das ontologias de entrada. Nesse momento são detectadas todas as inconsistências. Quando há ambigüidade nos nomes, ou seja, nomes diferentes nas ontologias e mesmos valores, o algoritmo procura por sinônimos e verifica a similaridade; caso o resultado dê entre 0.8 e 1.0 eles são sinônimos e a cópia ocorre normalmente. No caso dos elementos serem iguais ou sinônimos, possuem mesma faixa de valores, mas valores diferentes, o algoritmo reconhece essa diferença e adiciona os dois valores, o que enriquece a ontologia resultante. Essa algoritmo utiliza um formato próprio para as ontologias, a OMNotation, e é nesse formato que a ontologia resultante é gerada.

IX - COMA [Do, H. e Rahm, 2002; Aumueller et al 2005]

É um sistema cujo principal foco é o processo de *matching* tanto para ontologias como para esquemas (Relacionais ou XMLSchemas). É composto por diferentes algoritmos de *matching* e possui flexibilidade para que novos algoritmos sejam adicionados à sua biblioteca. Possui uma interface de usuário, um repositório de esquemas e um modelo uniforme de dados para dar suporte a esquemas e ontologias escritas em linguagens diferentes. Além disso, esse sistema possui uma abordagem de *match* baseado em fragmentos para decompor problemas de *match* grandes em problemas menores, ou para a execução de *merging* de ontologias ou comparação de diferentes mapeamentos. Sua abordagem para *match* é semi-automática, onde a interferência do usuário é necessária, e iterativa, e onde são reutilizados os resultados de *match* previamente computados. Adicionado a tudo isso, essa ferramenta possui uma plataforma para combinar e enriquecer os resultados obtidos.

X – ONTOMERGE [Dou et al., 2005]

Esse é um sistema para tradução de ontologias, em que é realizada a tradução de conceitos

expressos em uma ontologia para outra ontologia. A idéia principal é fazer a tradução de uma ontologia pelo *merging* de ontologias e raciocínio automático. As ontologias de entrada são traduzidas do seu formato original (ex: OWL) para um formato interno. O *merging*, então, é realizado pela união dos axiomas das mesmas. As pontes de axiomas são adicionadas para relacionar o termo de uma ontologia para outra. Com as ontologias combinadas, inicia-se a atividade de tradução que ocorre de forma automática pelo mecanismo de raciocínio.

A Tabela 3, baseada em [Euzenat e Shvaiko, 2007], faz uma comparação entre as soluções discutidas. A coluna 'Entrada' indica o formato de entrada aceito por cada solução e a coluna 'Saída' indica o resultado fornecido ao usuário após a execução da solução, podendo ter um dos seguintes tipos: 'Alinhamento' (conjunto de correspondências); 'Ontologias' (normalmente resultado do processo de *merge*); e 'Axiomas'. A coluna 'Necessidade' indica o que o sistema precisa para a sua execução, podendo ser o usuário para interferir nos resultados (interativo), instâncias do que se deseja analisar (p.ex: documentos relacionados às ontologias) ou é automático (executa de forma independente). A coluna '*Matching*' indica as técnicas utilizadas pelas soluções para esse processo. A maioria das soluções de *matching* combina a técnica léxica com a estrutural para obter um resultado mais confiável. Algumas soluções utilizam recursos externos como WordNet, ou algum alinhamento pré-computado para melhorar a qualidade do alinhamento final. A coluna '*Merging*' indica a abordagem de *merge* adotada por cada solução.

Tabela 3. Quadro Comparativo das Soluções Estudadas

Ferramentas	Entrada	Saída	Necessidade	Matching		Merging
				Técnicas Léxica	Técnicas Estrutural	Abordagem
iPROMPT	OWL	Ontologia	Usuário	String	-	Merge Completo
Alignment API	OWL, RDF	Alinhamento, Ontologia	Automático	String, Lingüística, Restrições, Recursos externos	Pontos Fixos	Ontologia Ponte
FCA-Merge	OWL	Ontologia	Usuário, instâncias	Lingüística	Análise de Conceito Formal	Merge Completo
OMAlgorithm	OM -Notation	Ontologia	Automático	Lingüística	Taxonomia,	Merge Completo
COMA++	Esquema Relacional, XML Schema, OWL, XSD, XDR, SQL	Alinhamento	Usuário	String, Lingüística, Restrições, Recursos externos	Grafo, Taxonomia, Ponto Fixo	Merge Completo
AnchorPrompt	OWL	Axiomas (OWL, RDF)	Usuário	String, Recursos Externos	Taxonomia, Grafo	-
ILIADS	OWL	Ontologia	Automático	String, Lingüística, Recursos externos	Taxonomia	Merge Completo
HMatch	OWL	Alinhamento	Automático	String, Lingüística, Recursos externos	Grafo	-
AUTOMS	OWL	Mapeamentos	Automático	COCLU	Taxonomia	-
HCONE-merge	OWL	Ontologia	Auto, usuário	Lingüística	-	Ontologia Ponte
OntoMerge	OWL	Ontologia	Alinhamento	Alinhamento Externo	-	Ontologia Ponte

2.3 Avaliação de Ontologias

Um dos pontos importantes da Web Semântica é o compartilhamento de informações que, por sua vez, estão disponíveis em diversas fontes independentes na web. Como a base da web semântica são as ontologias estas precisam ser confiáveis para que possam ser compartilhadas e reutilizadas por sistemas dirigidos à ontologia.

O desenvolvimento de aplicações baseadas em ontologia é um desafio no que diz respeito ao reuso de ontologias. Normalmente, esses sistemas precisam selecionar dentre as várias ontologias disponíveis a mais adequada para seus propósitos. Definir qual ontologia é mais ideal para uma determinada aplicação é uma atividade importante, visto que a qualidade da ontologia influencia diretamente do desempenho da aplicação [Sabou et al., 2006]. É preciso, então, comparar as características das ontologias para selecionar a ideal para o sistema.

O processo para avaliar uma ontologia consiste em estabelecer e aplicar critérios e métricas de modo a garantir a qualidade da ontologia. Atualmente não existe nenhum mecanismo apropriado para medir essa qualidade. De modo geral, a avaliação de ontologias pode ser categorizada como: (i) baseadas na comparação de algum modelo ideal (*gold standard*) como em [Maedche and Staab, 2002], e (ii) baseada em uma avaliação manual como em [Brewster et al 2004]. A segunda forma de avaliação analisa a ontologia em si, investigando, por exemplo, o tipo de conhecimento que pode ser extraído da ontologia; se é o esperado ou se são conhecimentos irrelevantes ou incoerentes, verificando possíveis erros taxonômicos [Gomez-Perez, 1999]. São exemplos de trabalhos nesse sentido: o OntoQA [Tartit et al 2005] e o OntoCAT [Cross and Pal, 2006; Cross and Pal, 2008], ferramentas que buscam avaliar a qualidade da ontologia de acordo com sua estrutura (distribuição das classes, relações, propriedades e heranças) e distribuição das instâncias da ontologia. No entanto, o trabalho aqui apresentado, não se propõe a analisar quão boa é uma ontologia (cabe ao usuário essa análise) e sim analisar o quão próxima está a ontologia resultante ou computada (O_{Comp}) de uma ontologia ideal ou de referência (O_{Ref}). Devido a isso, a abordagem de avaliação de ontologias adotada neste trabalho é a primeira, a qual utiliza uma ontologia modelo, *gold standard*, para validar a ontologia computada.

Nos últimos anos diversas medidas para avaliação de ontologias utilizando a primeira abordagem (*gold standard*) foram propostas, levando em consideração as diferentes camadas ontológicas (léxica ou estrutural). Na camada léxica as medidas binárias (certo ou errado) são

freqüentemente usadas para comparar termos das ontologias (O_{Ref} e O_{Comp}) considerando o *match* exato de strings. A comparação estrutural (conceito hierárquico) busca comparar as posições dos conceitos nas hierarquias das ontologias. Essas medidas que avaliam as ontologias ajudam a determinar a similaridade entre ontologias. Estudos têm sido realizados no intuito de definir essas medidas como, por exemplo, [Bisson, 1992], [Weinstein e Birmingham, 1999], [Maedche et al 2002] e [Brank et al 2006]. Apenas em [Dellschaft e Staab, 2006] é feita uma primeira tentativa para estabelecer uma medida padrão para calcular a similaridade entre ontologias, tendo sido criada uma ferramenta para facilitar o uso dessas medidas, a *OntEval*⁶. Em [Hjelm e Buitelaar, 2008] é proposta uma melhoria para esse padrão, no entanto, para facilitar a análise da estratégia MeMO através da utilização da ferramenta citada, essa melhoria não será levada em consideração.

2.3.1 Medidas para Avaliação de Ontologias

As medidas utilizadas neste trabalho para poder avaliar as ontologias resultantes serão aplicadas de acordo com [Dellschaft e Staab, 2006], onde as ontologias são comparadas por nível utilizando as principais medidas utilizadas para a avaliação da qualidade de ontologias, devido a sua simplicidade e facilidade de compreensão, são a *Precision*, *Recall* e *F-measure*.

A medida *Precision*, calculada por (1), mede o número de itens identificados corretamente como uma porcentagem do número de itens identificados, ou seja, quantos dos itens identificados são realmente corretos. Quanto maior a precisão maior a garantia de que o que tem sido identificado é correto.

$$P(O_{Ref}, O_{Comp}) = \frac{|O_{Ref} \cap O_{Comp}|}{|O_{Comp}|} \quad (1)$$

A medida *Recall*, calculada por (2), mede o número de itens identificados corretamente como uma porcentagem do número total de itens corretos. Em outras palavras, mede quanto dos itens que deveriam ter sido realmente identificados foram de fato identificados. Quanto mais alto o *Recall*, maior a garantia de que os itens corretos serão identificados.

$$R(O_{Ref}, O_{Comp}) = \frac{|O_{Ref} \cap O_{Comp}|}{|O_{Ref}|} \quad (2)$$

De fato, nem *Precision* e nem *Recall* podem, sozinhas, medir a qualidade de uma ontologia. A medida *Recall* pode ser facilmente maximizada à custa de uma medida ruim de *Precision*, ou seja o

⁶ <http://nlp.shef.ac.uk/abraxas/index.html>

retorno de todas as correspondências possíveis, porém não necessariamente corretas. Da mesma forma, um valor alto para *Precision* pode ser alcançado à custa de um valor ruim de *Recall* pelo retorno de apenas algumas das correspondências corretas [Do e Rahm, 2002]. Por isso é preciso considerar as duas medidas ou combiná-las.

A *F-measure* é uma medida harmônica entre *Precision* e *Recall*, usando-as como uma média ponderada dos dois. Se a média ponderada é 0.5 (que é normalmente o caso), ambas tem a mesma importância. Quanto maior essa medida maior a qualidade da ontologia. De forma geral essa medida é definida como (3):

$$F - measure = \frac{(\beta^2 + 1)P * R}{(\beta^2 R) + P} \quad (3)$$

Em (3) β reflete o peso de P (*Precision*) e R (*Recall*). Se P e R tem o mesmo peso, então pode ser definida:

$$F - measure = \frac{P * R}{0.5 * (R + P)} \quad (4)$$

Dessa forma, *F-measure* pode ser calculada em (5) como:

$$F(O_{Ref}, O_{Comp}) = \frac{2 * P(O_{Ref}, O_{Comp}) * R(O_{Ref}, O_{Comp})}{P(O_{Ref}, O_{Comp}) + R(O_{Ref}, O_{Comp})} \quad (5)$$

Ao se tratar de ontologias a relação binária ‘certo’ ou ‘errado’ não é suficiente. É preciso saber sobre a qualidade de uma ontologia (o quão certa está), visto que existem relações hierárquicas que não podem ser ignoradas. Por isso, ao comparar duas ontologias é preciso investigar as diferentes camadas da ontologia, léxica e estrutural. A seguir é dado como essas três medidas são usadas nas diferentes camadas.

A - Camada Léxica

A medida *Lexical Precision* (LP) e *Lexical Recall* (LR) são usadas para comparar termos da ontologia de referência e da ontologia computada podendo estar de acordo com algoritmos de *matching* citados anteriormente ou através do *matching* exato de *strings*, sendo definidas como:

- *LP-Lexical Precision*: mede quantas classes de uma ontologia global (computada) estão corretas com relação à ontologia ideal (referência), ou seja, quantas classes presentes na ontologia global possuem o mesmo nome que uma classe presente na ontologia ideal;

- *LR – Lexical Recall*: mede quantas classes da ontologia ideal (referência) estão, de fato, presentes na ontologia global (computada), dentre as que deveriam estar presentes.

Para ilustrar como essas medidas são computadas considere o exemplo da Figura 14 onde O_{Ref} é a ontologia de referência com um conjunto de classes C_{Ref} e O_{Comp} a ontologia computada com conjunto de classes C_{Comp} . Vê-se que existem cinco classes comuns entre as duas ontologias, um total de sete classes em O_{Comp} , e um total de oito em O_{Ref} . A medida *LP* é calculada, então, pela razão entre o número de classes identificadas corretamente pelo número de classes identificadas ($LP = 5/7 = 0.71$). Assim, nesse exemplo, 71% dos das classes identificadas são realmente corretas. A medida *LR* é calculada pela razão entre o número de classes identificadas pelo número de classes que realmente deveriam ter sido identificadas ($LR = 5/8 = 0.65$). Nesse exemplo, 65% dos itens que deveriam ter sido identificados foram de fato identificados. Por (5) tem-se que $LF = 0.67$, logo a O_{Comp} é 67% semelhante lexicamente a O_{Ref} .

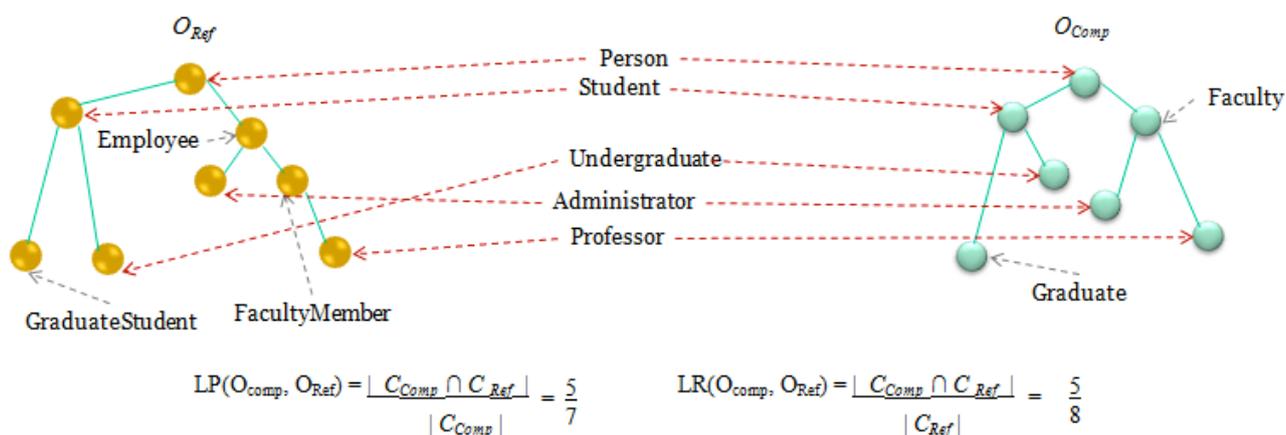


Figura 14. Exemplo de ontologia referência (ORef, à esquerda) e computada (OComp, à direita). Figura adaptada de [Maedche et al 2002] e [Dellschaft e Staab, 2006].

B - Camada Taxonômica

Visando uma comparação entre ontologias, uma análise apenas da camada léxica não é suficiente. É preciso analisar a hierarquia dos conceitos tendo, assim, medidas taxonômicas de *Presicion* (TP) e *Recall* (TR). Essas medidas são definidas aqui como:

- *TP – Taxonomical Precision*: mede quantos elementos presentes na hierarquia de um conceito em uma ontologia global (computada) estão corretos em relação aos elementos que estão presentes na hierarquia do conceito correspondente na ontologia ideal (referência).

- *TR – Taxonomical Recall*: mede quantos elementos presentes na hierarquia de um conceito na ontologia ideal estão, de fato, presentes na ontologia global (computada), dentre aqueles que deveriam estar.

A comparação hierárquica de conceitos é dividida em local e global. Na local compara-se a posição hierárquica de um elemento da O_{Comp} com um elemento da O_{Ref} . A comparação global é computada pela média das medidas locais para os pares de conceitos de O_{Ref} e de O_{Comp} .

A medida local deve ser computada através da caracterização da posição de um elemento da ontologia (características extraídas da hierarquia - *ce* - *characteristic extract*). Essa caracterização significa identificar o conjunto de elementos que está no caminho do elemento em questão que vai desde a raiz da ontologia até o nó folha. Caso os elementos comparados tenham posições similares na hierarquia, existirão muitos conceitos em comum. A partir da extração de características, as medidas taxonômicas *Precision* (tp_{ce}) e *Recall* (tr_{ce}) de dois elementos c_1 da ontologia O_{Comp} e c_2 da ontologia O_{Ref} são dadas por (6) e (7), respectivamente.

$$tp_{ce}(c_1, c_2, O_{Comp}, O_{Ref}) = \frac{|ce(c_1, O_{Comp}) \cap ce(c_2, O_{Ref})|}{|ce(c_1, O_{Comp})|} \quad (6)$$

$$tr_{ce}(c_1, c_2, O_{Comp}, O_{Ref}) = \frac{|ce(c_1, O_{Comp}) \cap ce(c_2, O_{Ref})|}{|ce(c_2, O_{Ref})|} \quad (7)$$

As características dos elementos são dadas pelo conjunto de todos os seus super e sub-elementos (classes) que é definido como *semantic cotopy* (sc). Formalizando sc , dado um conceito $c \in C_i$ de uma ontologia O_i , sc é definido por:

$$sc(c, O_i) = \{c_k \mid c_k \in C_i \wedge (c_k \leq c \vee c \leq c_k)\} \quad (8)$$

A Tabela 4 (adaptada de [Dellschaft e Staab, 2006]) ilustra como são realizados esses cálculos taxonômicos de acordo com a Figura 2.14, onde c é a coluna dos elementos da ontologia de referência e da computada; $sc(c, O_{Ref})$ é a coluna das características extraídas da ontologia de referência; $sc(c, O_{Comp})$ é a coluna das características extraídas da ontologia computada; e tp e tr são as medidas taxonômicas calculadas a partir das medidas obtidas das colunas referentes à sc . Considerando a primeira linha da tabela como exemplo, para o elemento *Person*, foram

identificados cinco itens corretos (*Person, Student, Undergraduate, Administrator, Professor*) dos sete identificados. Esses itens corretos são identificados pela interseção entre os elementos comuns encontrados nos caminhos dos elementos em suas respectivas ontologias.

Tabela 4. Extração de características das ontologias da Figura 14.

c	$sc(c, O_{Ref})$	$sc(c, O_{Comp})$	tp_{sc}	tr_{sc}
Person	{Person, Student, Graduate Student, Undergraduate, Employee, Administrator, FacultyMember, Professor}	{Person, Student, Graduate, Undergraduate, Faculty, Administrator, Professor}	5/7	5/8
Student	{Person, Student, Graduate Student, Undergraduate}	{Person, Student, Graduate, Undergraduate}	3/4	3/4
Graduate Student	{Person, Student, Graduate Student}	-	0	0
Undergraduate	{Person, Student, Undergraduate}	{Person, Student, Undergraduate}	3/3	3/3
Employee	{Person, Employee, Administrator, FacultyMember, Professor}	-	0	0
Administrator	{Person, Employee, Administrator}	{Person, Faculty Administrator}	2/3	2/3
FacultyMember	{Person, Employee, FacultyMember, Professor}	-	0	0
Professor	{Person, Employee, FacultyMember, Professor}	{Person, Faculty, Professor}	2/3	2/4
Graduate	-	{Person, Student, Graduate}	0	0
Faculty	{Person, Faculty Administrator, Professor}	-	0	0

Com as características extraídas e com as medidas locais computadas é possível, agora, fazer o cálculo da medida global. As medidas taxonômicas globais nada mais são que as médias das medidas locais adquiridas que são dadas por:

$$TP_{sc}(O_{Comp}, O_{Ref}) = \frac{1}{|C_{Comp}|} \sum_{c \in C_{Comp}} \begin{cases} tp_{sc}(c, c, O_{Comp}, O_{Ref}) & \text{se } c \in C_{Comp} \\ 0 & \text{caso contrário} \end{cases} \quad (9)$$

$$TR_{sc}(O_{Comp}, O_{Ref}) = \frac{1}{|C_{Ref}|} \sum_{c \in C_{Ref}} \begin{cases} tp_{sc}(c, c, O_{Comp}, O_{Ref}) & \text{se } c \in C_{Ref} \\ 0 & \text{caso contrário} \end{cases} \quad (10)$$

Seguindo o exemplo dado na Figuras 2.14 e com as medidas taxonômicas obtidas na Tabela 4, a Tabela 5 ilustra as medidas globais relativas às ontologias exemplo onde cada coluna representa uma parte das fórmulas (9) e (10). Assim, pela tabela têm-se que 54% das classes identificadas estão hierarquicamente corretas e 44% das classes que deveriam ser identificadas foram de fato identificadas de forma hierarquicamente corretas.

Tabela 5. Exemplo das medidas globais

TP			TR		
$\sum_{c \in C_{Comp}} \begin{cases} tp_{ac}(c, c, O_{Comp}, O_{Ref}) & \text{se } c \in C_{Ref} \\ 0 & \text{caso contrário} \end{cases}$	$ C_{Comp} $	Resultado	$\sum_{c \in C_{Ref}} \begin{cases} tp_{ac}(c, c, O_{Comp}, O_{Ref}) & \text{se } c \in C_{Ref} \\ 0 & \text{caso contrário} \end{cases}$	$ C_{Ref} $	Resultado
3,79	7	0,54	3,54	8	0,44

Para que as medidas *Precision* e *Recall* sejam balanceadas é preciso combinar essas medidas, o que é feito através da medida taxonômica *F-measure* (TF) tendo sua medida global em (11). Quanto mais alto o valor dessa medida melhor é a qualidade da hierarquia de conceitos.

$$TF(O_{Comp}, O_{Ref}) = \frac{2 * TP(O_{Comp}, O_{Ref}) * TR(O_{Comp}, O_{Ref})}{TP(O_{Comp}, O_{Ref}) + TR(O_{Comp}, O_{Ref})} \quad (11)$$

Continuando o exemplo da Tabela 5 essa medida é calculada da seguinte maneira: $TF = \{2 * 0,54 * 0,44 / (0,54 + 0,44)\} = 0,45$. Isso significa que a ontologia computada é 45% semelhante à ontologia ideal.

Pode-se notar que a medida TF, medida harmônica para a camada léxica não foi citada, apesar de ser fácil de ver como é calculada. Essa medida não foi destacada propositalmente porque não será levada em consideração para análise que é realizada neste trabalho, visto que se quer avaliar a qualidade taxonômica da ontologia computada, que é uma medida mais sensível às combinações.

Pretende-se, então, utilizar essas medidas para validar a estratégia de *merging* múltiplo de ontologias proposta neste trabalho. Para isto serão criadas ontologias ideais ou de referência (*gold standard*) para serem comparadas com as ontologias geradas como resultado do uso da estratégia MeMO, de maneira que a comparação das ontologias será feita de acordo com os valores obtidos para os critérios de qualidade destacados nesta seção. Mais detalhes sobre essas medidas podem ser encontrados em [Dellschaft e Staab, 2006].

2.4 Considerações finais

Este capítulo apresentou os principais temas de estudo para a realização deste trabalho: técnica de *clustering*; *matching* e *merging* de ontologias; e medidas de qualidade para avaliação de ontologias.

A técnica de *clustering* é uma técnica bastante utilizada quando se tem uma grande

quantidade de dados e se deseja extrair informações desses dados. Essa técnica agrupa os dados de acordo com suas semelhanças e, a partir daí, podem ser feitas análises ou realizar alguma outra atividade em cima de cada grupo. Essa técnica foi de fundamental importância para a elaboração da estratégia, sendo as ontologias analisadas e agrupadas de acordo com suas semelhanças. Embora com a utilização dessa técnica não seja garantido um resultado final ideal, por ser uma heurística, é possível obter um resultado final com alguma garantia de qualidade.

O *matching* de ontologias é o processo que busca por elementos correspondentes entre duas ontologias. Esse processo é importante para que as correspondências sejam identificadas e a partir daí seja realizada alguma outra atividade com essas informações, como, por exemplo, a atividade de *merging*. Com as correspondências identificadas é possível combinar duas ontologias onde uma nova ontologia é gerada representando todos os conceitos presentes nas duas ontologias originais envolvidas no processo. Muitas soluções já foram propostas para ambos os processos utilizando diferentes abordagens.

Como um dos objetivos deste trabalho consiste em gerar ontologias que representem uma visão global de um conjunto de ontologias, foi necessário realizar estudos sobre medidas que pudessem avaliar a qualidade das ontologias geradas. De maneira mais específica, surgiu a necessidade de avaliar a qualidade das ontologias geradas com o uso da estratégia MeMO com relação a a uma ontologia ideal ou uma ontologia gerada a partir de execuções aleatórias de múltiplos processos de *merging*. Nesses estudos foi identificada uma ferramenta, a OntEval, que calcula valores para as medidas (*Precision*, *Recall* e *F-measure*) entre ontologias, sendo bastante útil para a avaliação do trabalho proposto.

O próximo capítulo discutirá a aplicação desses conceitos fundamentais na estratégia MeMO, descrevendo as técnicas e soluções adotadas neste trabalho de acordo com os estudos preliminares realizados.

CAPÍTULO 3

Merging Múltiplo de Ontologias

A estratégia MeMO, apresentada neste capítulo, pode ser utilizada para a combinação de múltiplas ontologias relacionadas a um mesmo domínio. Um dos diferenciais da estratégia proposta é o uso da técnica de *clustering* com o intuito de agrupar ontologias mais similares e assim obter melhores resultados ao final do processo de *merging* múltiplo. Na seção 3.1 é dada uma visão geral sobre a estratégia. Na seção 3.2 é dada uma visão mais aprofundada detalhando as etapas e as principais atividades: cálculo da similaridade global de um par de ontologias, combinação de duas ontologias e junção de dois alinhamentos. Na seção 3.3 são feitas as considerações finais sobre este capítulo.

3.1 Visão Geral

O objetivo da estratégia MeMO é possibilitar a combinação (*merging*) automática de um conjunto de ontologias (O_1, \dots, O_n), com o intuito de se obter uma nova ontologia (O_g). Tal ontologia deve representar uma visão global de todas as outras envolvidas no processo de *merging* múltiplo. Além disso, como mencionado anteriormente, um aspecto fundamental da estratégia consiste em garantir a qualidade da ontologia final gerada, de forma que esta esteja sempre próxima à ontologia que seria obtida a partir de um processo manual ou semi-automático, realizado por especialistas do domínio.

Para tanto, a abordagem aqui proposta é inspirada no princípio utilizado por diversas soluções existentes para o problema de alinhamento múltiplo de seqüências biológicas [Feng & Doolittle, 1987; Thompson et al., 1994]. Nestas soluções, a partir de duas ou mais seqüências dadas como entrada, são realizados os alinhamentos entre as seqüências, ou seja, são identificados os seus trechos coincidentes, onde a cada passo, o par de seqüências mais semelhantes é alinhado. A determinação de tal par é feita através da utilização de técnicas de *clustering* hierárquico. É importante salientar que estas técnicas correspondem a soluções heurísticas e, como em toda heurística, nada garante que o ótimo será encontrado. Porém, ao final do processo, o alinhamento

completo entre todas as seqüências da entrada será melhor que o alinhamento obtido caso as seqüências fossem combinadas de maneira aleatória.

Na estratégia MeMO, a técnica de *clustering* hierárquico também é utilizada onde o ponto de partida é uma matriz de similaridade que auxilia na busca por ontologias mais próximas (semelhantes), as quais são combinadas duas a duas de forma progressiva. A Figura 15 apresenta uma visão geral da estratégia, ilustrando suas duas etapas (*Construção da Matriz de Similaridade* e *Combinação Progressiva*), bem como as atividades vinculadas a cada uma delas.

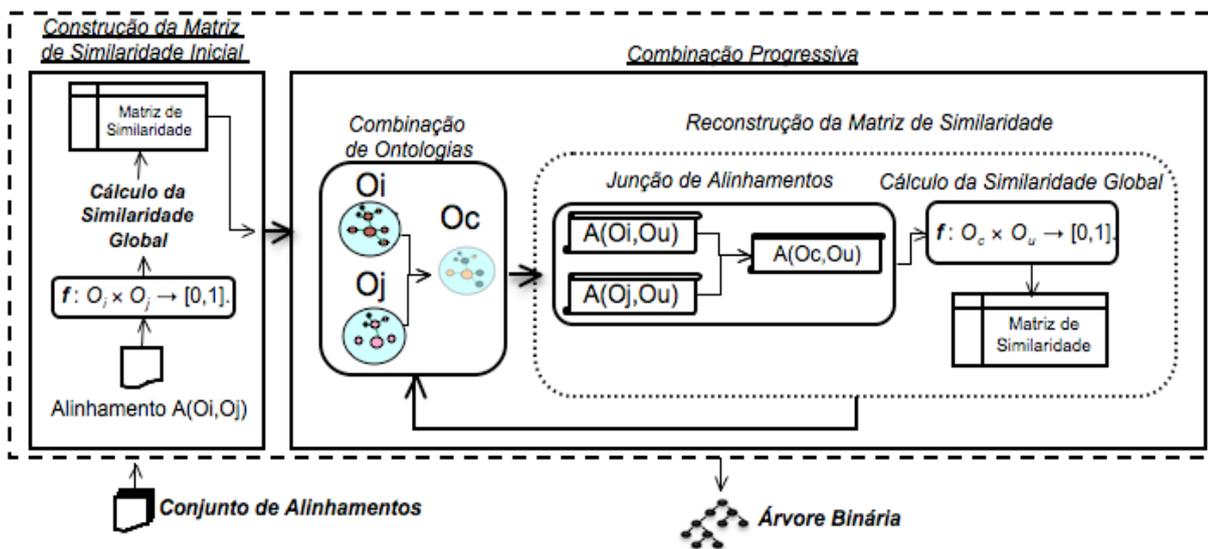


Figura 15. Ilustração da Estratégia MeMO

De maneira geral, a estratégia recebe como entrada um conjunto de alinhamentos, os quais representam as correspondências entre os elementos das ontologias. A primeira etapa se caracteriza pela construção da matriz de similaridade inicial. Para cada alinhamento é calculado o valor de similaridade global entre o par de ontologias. No final dessa etapa a matriz preenchida é passada para a segunda etapa, a *Combinação Progressiva*, onde as ontologias passam a ser combinadas progressivamente, de acordo com as seguintes atividades: *Combinação de Ontologias* e *Reconstrução da Matriz de Similaridade*. A primeira atividade realiza o *merging* entre as duas ontologias mais semelhantes, obtidas com base na matriz de similaridade. Por sua vez, a *Reconstrução da Matriz de Similaridade* tem como objetivo gerar uma nova matriz de similaridade envolvendo a ontologia gerada na atividade anterior. Para isso um novo alinhamento é criado (junção de alinhamentos) entre a nova ontologia gerada (O_c) e as demais ontologias envolvidas no processo para então se calcular a similaridade entre as ontologias e adicionar esse valor a uma nova matriz de similaridade. Com a nova matriz construída essa segunda etapa pode ser recomeçada.

Essa etapa termina quando a matriz reconstruída possuir apenas um elemento. Como saída do processo de *merging* múltiplo, tem-se uma árvore binária, onde as folhas correspondem ao conjunto das ontologias iniciais e a raiz representa a ontologia final resultante do processo de *merging* múltiplo. Além disso, ao longo dos níveis da árvore, é possível visualizar as ontologias combinadas, obtidas no decorrer do processo. Nas subseções seguintes, será detalhada cada uma destas etapas e atividades.

Dentre os diferenciais da estratégia proposta destacam-se: i) o processo de combinação automática de múltiplas ontologias; ii) o processo de junção de alinhamentos de ontologias para evitar novas chamadas ao processo de *matching*; iii) a análise prévia da similaridade entre as ontologias, resultando em uma ontologia final mais próxima da ontologia ideal; e iv) a disponibilização de ontologias intermediárias (visões parciais) e dos alinhamentos gerados durante o processo, possibilitando reutilização dos mesmos.

Visando facilitar o entendimento, cada ontologia O_i ($1 \leq i \leq n$) do conjunto de ontologias $O = \{O_1, \dots, O_n\}$ que serão combinadas, será definida como se segue:

Definição 1 (Ontologia). Uma ontologia O_i é uma descrição formal de um modelo de dados sendo constituída por um conjunto de elementos, onde:

- O_i é formado por um conjunto de classes (C_i) e um conjunto de propriedades (P_i);
- C_i é o conjunto de classes da ontologia O_i , onde cada classe $c \in C_i$ caracteriza um conjunto de indivíduos;
- P_i é o conjunto das propriedades existentes em O_i que podem ser propriedades de objeto ou propriedades de tipo de dados;
- $E_i = \{e_{i1}, \dots, e_{im} \mid e_{it} \in C_i \cup P_i \text{ e } 1 \leq t \leq m\}$ representa o conjunto de elementos da ontologia O_i , onde cada $e_{it} \in E_i$ pode ser uma classe ou propriedade, sendo representada como $e_{it} \in (C_i \cup P_i)$;
- $|O_i|$ representa o número total de elementos da ontologia O_i .

Neste trabalho não serão levadas em consideração as instâncias e nem as restrições que podem eventualmente constituir uma ontologia.

3.2 MeMO – Estratégia para *Merging* Múltiplo de Ontologias

A estratégia MeMO utiliza técnicas de *clustering* hierárquico a fim de auxiliar a busca por ontologias mais próximas (semelhantes) para serem agrupadas. Seguindo essa técnica, a estratégia foi dividida em duas etapas que serão descritas a seguir: *Construção da matriz de similaridades* e *Combinação Progressiva*.

3.2.1 Etapa I: Construção da matriz de similaridades

Esta etapa tem como objetivo a construção de uma matriz (ponto de partida para a obtenção dos grupos), chamada de matriz de similaridade, na qual cada célula (i, j) representa a medida de similaridade global entre um par de ontologias $(O_i, O_j) \in \mathcal{O}$ onde $1 \leq i, j \leq n$. Conforme ilustrado na Figura 15, a entrada para a construção dessa matriz é o conjunto de todos os alinhamentos \mathcal{A} onde cada alinhamento está relacionado a duas ontologias do conjunto \mathcal{O} . Esses alinhamentos podem ser obtidos através de algoritmos e ferramentas de *matching* já existentes como, por exemplo, o *HMatch* [Castano, 2007], CATO [Breitman et al 2005] e a *Alignment API* [Euzenat, 2004]. A Figura 16 ilustra um exemplo de uma matriz de similaridades, preenchida por valores hipotéticos, com quatro ontologias quaisquer, onde os valores representam as semelhanças entre as ontologias.

	O_1	O_2	O_3	O_4
O_1	1.0	0.6	0.8	0.5
O_2	0.6	1.0	0.7	0.6
O_3	0.8	0.7	1.0	0.7
O_4	0.5	0.6	0.7	1.0

Figura 16. Exemplo de uma matriz de similaridade

O princípio para construção da matriz de similaridade segue a técnica de *clustering* hierárquico, onde os valores que compõem a matriz são calculados a partir das distâncias entre os objetos de um grupo (linha 5 do Algoritmo 2). Segundo abordado em [Su e Gulla, 2006], o cálculo da distância entre dois grupos (*clusters*) pode ser representado pelo cálculo da similaridade entre duas ontologias, se considerarmos que a similaridade corresponde à distância e a ontologia corresponde a um grupo. Dessa forma, podemos construir uma matriz de similaridade aplicando uma das abordagens propostas na literatura [Su e Gulla, 2006; Do e Rahm, 2002] para o cálculo da similaridade entre duas ontologias O_i e O_j . Em geral, tais abordagens podem levar em consideração: (i) a quantidade de correspondências presente no alinhamento; (ii) a quantidade de elementos em cada uma das ontologias O_i e O_j ; e (iii) o valor de confiança/semelhança entre cada par de

elementos a e b (correspondência) presentes no alinhamento.

Desse modo, o cálculo da similaridade entre ontologias é realizado utilizando os alinhamentos do conjunto A onde cada alinhamento deste conjunto identifica duas ontologias e suas correspondências, podendo ser definido como se segue [Euzenat, 2004; Euzenat, 2008]:

Definição 2 (Alinhamento). Um alinhamento $A_w(O_i, O_j)$, ou simplesmente A_w , entre duas ontologias O_i e O_j é formado por um conjunto de correspondências onde:

- $A_w(O_i, O_j) = \{(x_1, \dots, x_z) \mid x_k \text{ é uma correspondência do alinhamento e } 1 \leq k \leq z\}$;
- $x_k = (a, b, r, sim)$, onde $a \in E_i$; $b \in E_j$; r é um relacionamento entre a e b , o qual pode ser de equivalência(\equiv), generalização(\supseteq) ou especialização(\sqsubseteq); e sim é uma medida de confiança no relacionamento, que varia entre $[0,1]$. Essa medida de confiança pode ser computada por algum método, como uma função qualquer que calcula a similaridade entre dois elementos, ou especificada pelo usuário;
- $|A_w|$ representa o número total de correspondências de A_w , ou seja, $|A_w| = z$.

Neste trabalho foram estudadas três fórmulas para o cálculo da similaridade global entre ontologias:

- Fórmula (1) [Su e Gulla, 2005]: dada pela razão entre a soma dos valores de similaridade de todas as correspondências encontradas em um alinhamento qualquer $A(O_i, O_j)$ pelo produto do número de elementos encontrados nas duas ontologias envolvidas. Esse produto representa a quantidade de todas as possibilidades de combinações possíveis entre os elementos das duas ontologias.

$$sim(O_i, O_j) = \frac{\sum_{(a,b) \in A(O_i, O_j)} sim(a, b)}{|O_i| * |O_j|} \quad (1)$$

- Fórmula (2) [Do and Rahm, 2002]: dada pela razão entre a soma dos valores de similaridade de todas as correspondências possíveis e o total de elementos das duas ontologias. Para se conseguir todos os valores de similaridade possível é preciso encontrar todas as possibilidades de correspondências, o que se faz gerando um alinhamento $A(O_i, O_j)$ que é o alinhamento em que O_i é a primeira ontologia, e outro $A(O_j, O_i)$, onde O_j é a primeira ontologia. Essas duas direções de alinhamento são consideradas porque podem ter um conjunto de correspondências diferentes.

$$Average(O_i, O_j) = \frac{\sum_{(a,b)} sim(a, b) + \sum_{(b,a)} sim(b, a)}{|O_i| + |O_j|} \quad (2)$$

- Fórmula (3) [Do and Rahm,2002]: dada pela razão entre o número total de correspondências dos alinhamentos nas duas direções e o número total de conceitos das ontologias. Essa fórmula (*DICE*) não leva em consideração os valores de similaridade.

$$DICE(O_i, O_j) = \frac{|A(O_i, O_j)| + |A(O_j, O_i)|}{|O_i| + |O_j|} \quad (3)$$

Na estratégia MeMO, utiliza-se a função (3) para o cálculo da similaridade entre duas ontologias. Como pode ser observado, esta função não leva em consideração as mediadas de confiança $sim(a,b)$ dos relacionamentos. Com o uso desta função, a similaridade é obtida a partir da razão entre o número total de correspondências dos alinhamentos (em ambas as direções) e o número total de conceitos das ontologias. Esta característica foi fundamental para a escolha da função *DICE*. Isto porque, como será descrito posteriormente, ao longo do processo automático de *merging* múltiplo de ontologias, as informações sobre as medidas de confiança nem sempre estarão disponíveis. Dessa forma, tornou-se necessário a escolha de uma solução que permitisse a construção da matriz de similaridade de maneira independente destes valores.

O Algoritmo 2 ilustra a construção da matriz inicial onde é dado como entrada um conjunto de alinhamentos \mathcal{A} e retornada uma matriz de similaridade $\mathbf{M}_{N \times N}$ onde N é a quantidade de ontologias a serem combinadas. Como é dado como entrada um conjunto de alinhamentos, é preciso descobrir a quantidade de ontologias envolvidas no processo para inicializar a matriz. A partir do conjunto de alinhamentos calcula-se o número N de ontologias envolvidas (linha 1) e então a matriz é inicializada (linha 2) com N colunas e N linhas. Uma variável chamada *maior* (linha 3), que conterà informações sobre o maior valor da matriz e as posições das ontologias relacionadas a esse valor, é inicializada em seguida. Com a matriz e a variável *maior* inicializadas começa o cálculo do valor de similaridade global entre duas ontologias utilizando o respectivo alinhamento para o preenchimento dessa matriz (linhas de 4 a 8). Esse valor de similaridade global (SG) é calculado na linha 5 a partir de um alinhamento $A_w(O_i, O_j) \in \mathcal{A}$. Em seguida identifica-se na matriz a posição, ou seja a célula da matriz, que representa a combinação dessas duas ontologias (linhas 6 e 7). Esse valor é, então, adicionado à matriz de similaridade $\mathbf{M}_{N \times N}$ (linha 8). Nas linhas de 9 a 12 é

feita uma comparação entre o valor de similaridade calculado e o valor armazenado na variável *maior*. Se o valor armazenado for menor que o calculado então o valor armazenado é substituído (linhas de 10 a 12), de tal forma que a variável *maior* sempre armazenará o maior valor de similaridade da matriz. Essa etapa termina quando a matriz estiver completamente preenchida com todos os valores de similaridade entre as ontologias.

Algoritmo ConstruçãoMatriz

Entrada: Conjunto de alinhamentos $A = \{A_1, \dots, A_m\}$

Saída: Matriz de Similaridade $M_{N \times N}$

1. Descubra quantidade de ontologias envolvidas: $N(N-1)/2=m$;
2. Inicializa matriz de similaridade $M_{N \times N}$
3. $maior.sim \leftarrow 0, maior.pos_i \leftarrow -1, maior.pos_j \leftarrow 0$;
4. Para $w \leftarrow 1$ até m faça
5. SG \leftarrow Calcula similaridade global entre O_i e O_j usando o alinhamento $A_w(O_i, O_j)$;
6. $pos_i \leftarrow$ Descubra posição (linha e coluna) de O_i na matriz M ;
7. $pos_j \leftarrow$ Descubra posição (linha e coluna) de O_j na matriz M ;
8. Adiciona SG na matriz em $M[pos_i, pos_j]$;
9. Se $maior.sim < SG$
10. $maior.sim \leftarrow SG$;
11. $maior.pos_i \leftarrow pos_i$;
12. $maior.pos_j \leftarrow pos_j$

Algoritmo 2. Construção da matriz de similaridade inicial

A estratégia prossegue passando a *Matriz de Similaridade* inicial como entrada para a próxima etapa, *Combinação Progressiva*, descrita a seguir.

3.2.2 Etapa II: Combinação Progressiva

O objetivo desta etapa consiste na geração de uma ontologia global O_g , tendo como consequência, a obtenção de uma árvore binária que representa a combinação progressiva de todas as ontologias envolvidas no processo de *merging* múltiplo. Os nós folhas representam as ontologias do conjunto O , os nós intermediários representam as ontologias combinadas durante o processo (visões parciais), e o nó raiz representa O_g . A Figura 17 ilustra um exemplo de uma árvore binária que representa o resultado dessa etapa referente a quatro ontologias quaisquer ($O = \{O_1, O_2, O_3, O_4\}$).

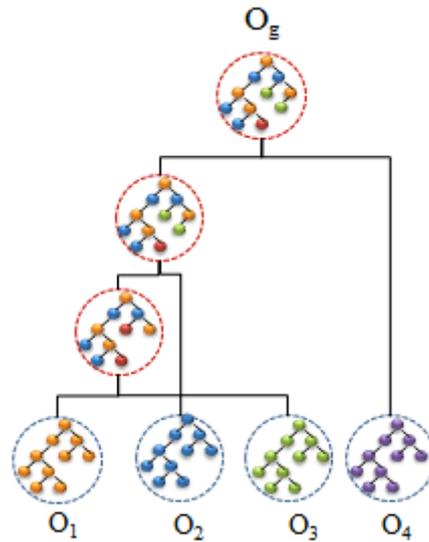


Figura 17. Árvore binária resultante do merging múltiplo de quatro ontologias

Para este fim, esta etapa é composta por duas atividades: (i) *combinação de ontologias* e (ii) *reconstrução da matriz de similaridade*. Estas atividades são repetidas até que, partindo de uma Matriz de Similaridade Inicial $M_{n \times n}$, seja obtida uma matriz $M_{1 \times 1}$ a qual possui apenas um valor relacionado à ontologia final O_g . Assim, a ontologia global será representada pela união de todas as ontologias envolvidas tal que $O_g = O_1 \cup O_2 \cup \dots \cup O_n$, dado de forma geral em (4).

$$O_g = \bigcup_{i=1}^n O_i \quad (4)$$

O Algoritmo 3 ilustra a realização da combinação progressiva de um conjunto de ontologias (O_1, \dots, O_n), onde são informados como entrada a matriz de similaridade $M_{N \times N}$, o conjunto de alinhamentos A e a variável *maior*, a qual identifica o maior valor de similaridade (*maior.sim*) da matriz e as posições na matriz das ontologias relacionadas a esse valor (*maior.pos_i* e *maior.pos_j*). A saída do algoritmo é o conjunto de todas as ontologias encontradas durante o processo.

Algoritmo CombinacaoProgressiva

Entrada: Matriz $M_{N \times N}$, $A = \{A_1, \dots, A_m\}$, Variável *maior*

Saída: Conjunto de ontologias

1. Inicializar uma árvore binária T com N nós folhas
2. Para $x \leftarrow 1$ até $N-1$ faça
3. $O_i \leftarrow$ Identifica a ontologia da posição *maior.pos_i*;
4. $O_j \leftarrow$ Identifica a ontologia da posição *maior.pos_j*;
5. $A_w \leftarrow$ Alinhamento $A(O_i, O_j)$;
6. $O_c \leftarrow$ merge (O_i, O_j, A_w);
7. Adicionar O_c a um nó da árvore T
8. Reconstrói M retirando O_i e O_j e adicionando O_c ;

Algoritmo 3. Combinação Progressiva

Na linha 1 é inicializada a árvore binária onde os nós folhas são constituídos pelas ontologias iniciais. Nas linhas 3 e 4 são identificadas as ontologias mais semelhantes da matriz de similaridade M através da célula (i, j) ; e na linha 5 é identificado o alinhamento entre essas ontologias. Em seguida, essas duas ontologias são combinadas (linha 6) e a ontologia resultante é adicionada a um nó da árvore (linha 7) cujos filhos são as ontologias de entrada da função *merge*. Na linha 8, a matriz é reconstruída de forma que as ontologias das posições relativas ao maior valor são retiradas e a ontologia resultante do *merge* é adicionada na primeira posição da matriz ($M[1,1]$). Cada uma das duas atividades envolvidas nessa etapa será descrita a seguir.

Etapa II - Atividade 1: Combinação de duas ontologias

A atividade de Combinação de Ontologias consiste na realização do *merging* entre as duas ontologias O_i e O_j mais semelhantes, obtidas a partir da identificação da célula (i, j) que contenha o maior valor na matriz de similaridade, em um dado momento do processo de *merging* múltiplo.

A Figura 18 ilustra um exemplo onde é identificado o maior valor na matriz (0.8) na célula $(0, 2)$. As ontologias identificadas foram O_1 e O_3 , e a ontologia gerada pela combinação é O_5 .

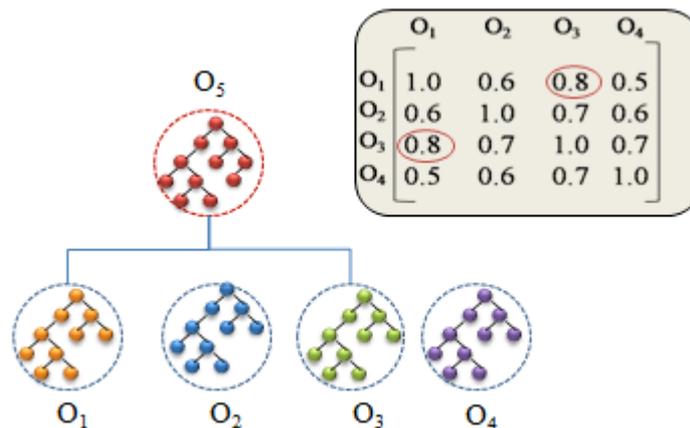


Figura 18. Exemplo de identificação de duas ontologias mais semelhantes

Como discutido anteriormente, existem diversas abordagens para o *merging* de ontologias, todas seguindo as idéias básicas de um processo de *merging* que consiste em identificar e combinar elementos correspondentes, copiar elementos distintos e resolver possíveis conflitos. Como o foco deste trabalho não é o desenvolvimento de um algoritmo para realizar o *merging* de duas ontologias foram escolhidas duas abordagens para serem usadas como base: [Pottinger e Bernstein 2003] e

[Rasgado, 2006]. Essas abordagens foram escolhidas porque as demais soluções estudadas ou não se adequam aos objetivos do trabalho ou não fornecem mais detalhes sobre seus algoritmos. É importante destacar que a estratégia proposta neste trabalho não depende de um processo de *merging* específico. Assim, de forma geral, levando em consideração as abordagens escolhidas, a combinação (*merging*) de duas ontologias segue a seguinte idéia:

- Para cada classe $c \in C_i$ se existir $x_k \in A_w(O_i, O_j)$ tal que $x_k = (c, c', =, sim)$ e $c' \in C_j$ então $C_c = C_c \cup c$;
- Para cada classe $c \in C_i$ se existir $x_k \in A_w(O_i, O_j)$ tal que $x_k = (c, c', \supseteq, sim)$ ou $x_k = (c, c', \subseteq, sim)$ e $c' \in C_j$ então $C_c = C_c \cup c \cup c'$ onde as relações entre as classes são preservadas;
- Para cada classe $c \in C_i$ se existir $x_k \in A_w(O_i, O_j)$ tal que $x_k = (c, p, =, sim)$ com $p \in P_j$ ou $x_k = (p, c, =, sim)$ com $p \in P_j$ e $c \in C_j$ então $C_c = C_c \cup c$;
- Para cada propriedade $p \in P_i$ se existir $x_k \in A_w(O_i, O_j)$ tal que $x_k = (p, p', =, sim)$ com $p' \in P_j$ então $P_c = P_c \cup p$ onde o domínio da propriedade é a união dos domínios;
- Para qualquer elemento $e_i \in E_i$ se NÃO existir uma correspondência em $A_w(O_i, O_j)$ relacionando e_i com um elemento $e_j \in E_j$ então: i) $C_c = C_c \cup e_i$, caso e_i seja uma classe, ou ii) $P_c = P_c \cup e_i$, caso e_i seja uma propriedade, de maneira que são preservadas as relações de e_i com os outros elementos de E_i ;

Como resultado desta atividade, é gerada uma nova ontologia O_c , chamada de *ontologia combinada* que pode formalmente ser definida como:

Definição 3 (Ontologia combinada). Seja $merge(O_i, O_j, A_w)$ uma função que recebe como entrada duas ontologias O_i e O_j e um alinhamento $A_w = A(O_i, O_j)$ entre ambas. O resultado desta função consiste em uma *ontologia combinada* O_c , a qual possui um conjunto de elementos $E_c = E_i \cup E_j$, tal que $C_c \supseteq C_i \cup C_j$ e $P_c \supseteq P_i \cup P_j$.

A decisão tomada neste trabalho de permanecer com as classes na nova ontologia nos casos onde classes e propriedades são correspondentes se deve a seguinte idéia: se um elemento for uma classe com indivíduos, então deve existir uma classe na nova ontologia para que essas instâncias possam ser adicionadas corretamente. Se as classes não fossem preservadas essas instâncias perderiam suas referências. Porém, descartar a propriedade pode também causar problemas nos

casos onde essas propriedades estão relacionadas às classes com indivíduos. Com a propriedade descartada o indivíduo também perderia essa informação relacionada à propriedade. Caso essa ontologia estivesse relacionada a uma aplicação que envolvesse esquemas de banco de dados essa perda da propriedade poderia causar problemas. Uma forma de solucionar seria criar uma nova propriedade para substituir a propriedade perdida onde o domínio dessa propriedade seria a classe adicionada na nova ontologia que foi escolhida ao invés da propriedade em questão. Lembrando que essas são soluções possíveis, não existindo uma solução ideal. Dependerá dos objetivos a que a ontologia a ser criada se propõe. Apesar deste trabalho não tratar com indivíduos no atual momento, essa possibilidade foi levada em consideração pensando em uma possível evolução da estratégia.

O algoritmo utilizado neste trabalho para a combinação de duas ontologias inicialmente combina as classes e depois as propriedades. Essa ordem é obedecida para diminuir alguns tipos de conflitos que podem ocorrer durante o processo [Pottinger e Bernstein, 2003; Noy e Musen 2002]. Esse ganho é obtido devido às propriedades referenciarem as classes (domínios das propriedades são formados por classes).

Para melhor ilustrar como ocorre a combinação de ontologias, considere o exemplo da Figura 19 onde são apresentadas duas ontologias, O_1 e O_2 , não normalizadas, relativas ao domínio “University”, e o alinhamento $A(O_1, O_2)$. Por questões de simplificação, apenas a combinação das classes será ilustrada. Inicialmente, a ontologia O_i é percorrida e, para cada elemento de O_1 , é verificado se existem correspondências em $A(O_1, O_2)$ envolvendo este elemento. Caso existam, as correspondências são tratadas com base na abordagem adotada para o *merging*. Do contrário, o elemento é simplesmente adicionado na nova ontologia O_{12} . Após a conclusão do percurso na ontologia O_1 , a ontologia O_2 também é percorrida, no sentido de garantir que todos os elementos presentes em ambas as ontologias estarão em O_{12} , mesmo os que não possuem correspondências em $A(O_1, O_2)$.

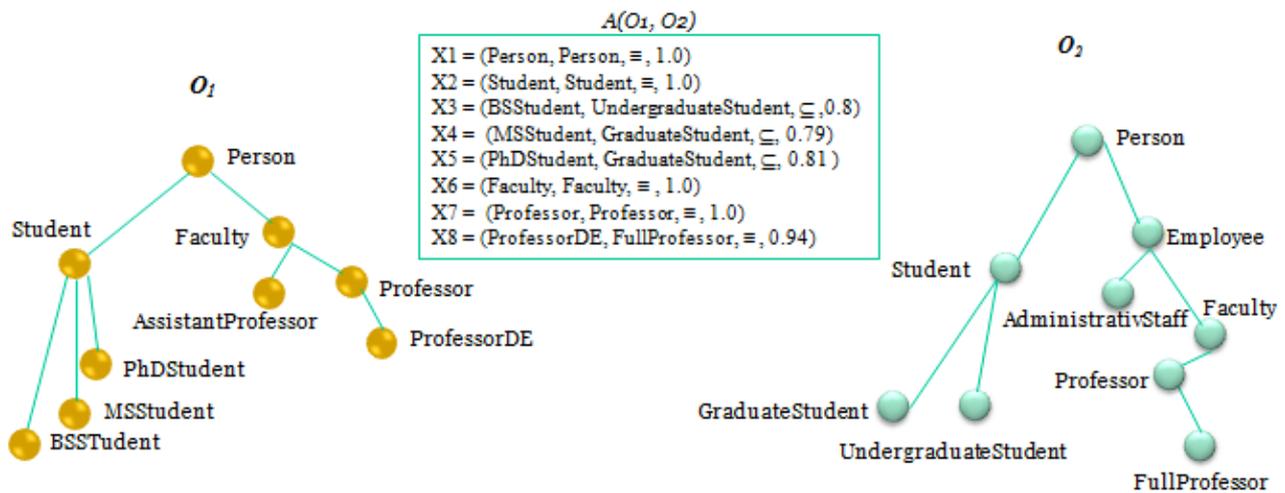


Figura 19. Exemplo de duas ontologias O_i e O_j e seu alinhamento $A(O_i, O_j)$

A Figura 20 mostra como deveria ser gerada uma ontologia O_{12} resultante do *merging* entre as ontologias O_1 e O_2 da Figura 19. Os números em cada nó indicam a ordem na qual os elementos foram adicionados, considerando o percurso da árvore em pré-ordem.

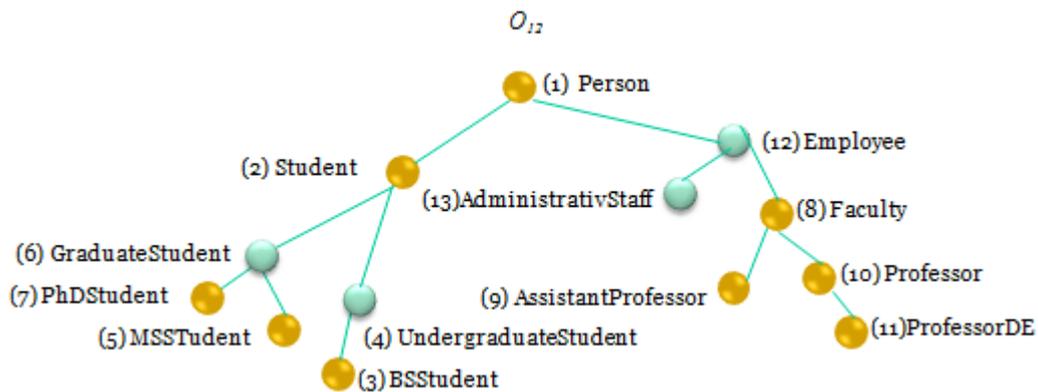


Figura 20. Exemplo da ontologia resultante do merging das duas ontologias O_i e O_j

Inicialmente, seguindo a estrutura de O_1 , são adicionados à nova ontologia os elementos (1) *Person*, (2) *Student* e (3) *BSStudent*. É observado pelo alinhamento $A(O_1, O_2)$ que o elemento *BSStudent* é subclasse de (4) *UndergraduateStudent* da ontologia O_2 . Com a identificação dessa correspondência, o elemento *UndergraduateStudent* é então adicionado como superclasse de *BSStudent*, continuando como subclasse de *Student*. Em seguida é adicionado (5) *MSStudent*, que pelo alinhamento vê-se que é subclasse de (6) *GraduateStudent*, sendo este adicionado em seguida. Seguindo O_1 , (7) *PhDStudent* é adicionado e é verificado no alinhamento que seu correspondente já está adicionado. *PhDStudent* é adicionado como subclasse do seu correspondente. Em seguida são

adicionados (8)*Faculty*, (9)*AssistantProfessor*, (10)*Professor* e (11)*ProfessorDE*. Com a primeira ontologia complementemente percorrida, inicia-se o acesso à segunda ontologia. É identificado que *Person*, *Student*, *GraduateStudent* e *UndergraduateStudent* já foram adicionados. Ao acessar *Employee* são identificadas as posições de sua super e subclasses na nova ontologia para então ser adicionada. A última classe a ser adicionada é a (13)*AdministrativStaff*. As demais classes são acessadas, mas é identificado que já foram adicionadas.

Observando a Figura 20, é possível notar que todas as relações presentes nas duas ontologias originais são mantidas na ontologia O_{12} , indicando que esta preserva as características das ontologias O_1 e O_2 , a partir das quais foi gerada.

Com o fim da atividade de combinação de duas ontologias, começa a atividade de reconstrução da matriz de similaridades onde serão calculados os valores de similaridade entre a nova ontologia e as outras ontologias que ainda não foram consideradas no processo de *merging* múltiplo.

Etapa II – Atividade 2: Reconstrução da Matriz de Similaridade

Uma vez que duas ontologias foram combinadas na atividade anterior, para que a fase de *Combinação Progressiva* continue a ser corretamente executada, é necessário criar uma nova matriz de similaridades, onde as ontologias envolvidas no passo anterior são excluídas e a nova ontologia gerada (O_c) é adicionada. Nesse sentido, a reconstrução desta matriz ocorre com base em duas sub-atividades:

- (i) *criação de novos alinhamentos*: conforme discutido anteriormente, os valores presentes na matriz de similaridade são calculados a partir dos alinhamentos entre cada par de ontologias. Para isso é necessário gerar os novos alinhamentos entre O_c e as demais ontologias (O_1, \dots, O_{n-2}). Este trabalho propõe uma nova abordagem, a *junção de alinhamentos* para evitar chamadas ao processo de *matching*. Essa junção consiste em reutilizar alinhamentos existentes para a criação de um novo alinhamento. A Figura 21 ilustra, de forma geral, os alinhamentos que deveriam ser criados caso duas de quatro ontologias quaisquer fossem combinadas. Cada novo alinhamento a ser criado, ou seja, alinhamento entre a nova ontologia (O_5) e as demais ontologias (O_2, O_4), é gerado pela junção dos alinhamentos relacionados às ontologias envolvidas na combinação (O_1 e O_3) e a ontologia alvo (ontologia que se quer calcular o valor de similaridade). Assim, o novo alinhamento $A(O_5, O_2)$ é gerado por $A(O_1, O_2) \cup A(O_2, O_3)$ e o outro novo alinhamento $A(O_5, O_4)$ é gerado por $A(O_1, O_4) \cup A(O_3, O_4)$;



Figura 21. Exemplo de Construção de novos alinhamentos

(ii) *cálculo de similaridade*: de posse dos novos alinhamentos, o cálculo do valor de similaridade global entre O_c e as demais ontologias da matriz é realizado de maneira semelhante à descrita na Seção 3.2.1. A atividade de reconstrução da matriz termina quando a nova matriz está completamente preenchida com os novos valores de similaridade. A Figura 22 ilustra um exemplo da nova matriz onde foram retiradas as ontologias O_1 e O_3 (ontologias combinadas) e adicionada a ontologia resultante (O_5).

	O_5	O_2	O_4
O_5	1.0	0.7	0.6
O_2	0.7	1.0	0.6
O_4	0.7	0.6	1.0

Figura 22. Reconstrução da matriz da Figura 3.2

A primeira sub-atividade (*criação de novos alinhamentos*) é um diferencial neste trabalho onde uma nova abordagem é proposta para viabilizar o *merging* múltiplo automático através da reutilização de alinhamentos. A forma tradicional de se construir alinhamentos entre ontologias se dá através da realização de um processo de *matching*. No entanto, este é um processo custoso, o qual envolve análise léxica, estrutural e semântica entre cada elemento das ontologias [Euzenat e Valtchev, 2003]. Além disso, as soluções atualmente existentes para o *matching* automático nem

sempre são suficientemente satisfatórias, seja pela qualidade do resultado ou pela complexidade em torno do processo [Castano et al.2003; Aleksovski et AL 2007].

Por outro lado, durante a realização de um processo de *merging* múltiplo de ontologias, há a necessidade de se construir novos alinhamentos para cada combinação executada. Tal necessidade, aliada ao contexto apresentado acima, nos leva a enfrentarmos dois problemas principais: (i) o alto custo para realização de vários *matchings* e (ii) a confiabilidade dos vários alinhamentos gerados a partir destes *matchings*. Mais precisamente, este segundo problema é agravado quando realizamos o *merging* múltiplo de maneira automática, que é o caso deste trabalho. Isto porque a automação do processo inviabiliza a interferência do usuário, no sentido de validar os alinhamentos gerados a cada passo. Na realidade, o único alinhamento sobre o qual podemos ter garantias é o alinhamento da entrada, uma vez que este pode ser validado pelo usuário antes do processo se iniciar.

O processo de junção de alinhamentos é baseado na seguinte idéia: quando uma nova ontologia O_c for gerada a partir da combinação de duas outras ontologias O_i e O_j , ao invés de se realizar um *matching* entre O_c e cada uma das outras ontologias O_u ($O_u \in \{O_1, \dots, O_{n-2}\}$) da matriz, os novos alinhamentos $A(O_c, O_u)$ podem ser obtidos reutilizando-se os alinhamentos fornecidos na primeira etapa da estratégia MeMO. Esta reutilização acontece através da junção de alinhamentos, que pode ser definida como segue:

Definição 4 (Junção de Alinhamentos). Sejam dois alinhamentos $A(O_i, O_u) = \{x_1, \dots, x_t\}$ e $A(O_j, O_u) = \{y_1, \dots, y_z\}$ que compartilham uma ontologia comum O_u . É possível encontrar outro alinhamento $A(O_c, O_u)$, onde $O_c = merge(O_i, O_j, A(O_i, O_j))$, a partir da junção dos alinhamentos $A(O_i, O_u)$ e $A(O_j, O_u)$, de forma que:.

- $A(O_c, O_u) = A(O_i, O_u) \cup A(O_j, O_u)$ onde $O_c = O_i \cup O_j$ e O_u é a ontologia alvo;
- $A(O_c, O_u) = \{v_1, \dots, v_w\} = \{x_1, \dots, x_t\} \cup \{y_1, \dots, y_z\}$ onde $v_i \neq v_j$, $1 \leq i, j \leq w$, e $i \neq j$. Uma correspondência x_k ($1 \leq k \leq t$) é dita diferente de outra correspondência y_h ($1 \leq h \leq z$) quando seus pares de elementos são diferentes. Assim, com $x_k = (a, b, r, sim)$ sendo a elemento de O_i e b elemento de O_u ; e $y_h = (c, d, r', sim')$ sendo c elemento de O_j e d elemento de O_u ; o par (a, b) é igual a (c, d) se $a=c$ e $b=d$, independente da medida de confiança;
- $A(O_i, O_u) \cap A(O_j, O_u)$ é o conjunto de correspondências iguais;
- $|A(O_c, O_u)|$ representa o número de correspondências do alinhamento;
- $|A(O_c, O_u)| = |A(O_i, O_u)| + |A(O_j, O_u)| - |A(O_i, O_u) \cap A(O_j, O_u)|$.

Exemplos de ferramentas que realizam atividade semelhante são: Alignment API⁷ e HMatch⁸. A diferença é que essas duas ferramentas visam enriquecer o alinhamento ou criar um novo alinhamento utilizando a idéia de transitividade. O enriquecimento de um alinhamento é a geração de um novo alinhamento, $A''(O_1, O_2)$, a partir de dois alinhamentos, $A(O_1, O_2)$ e $A'(O_1, O_2)$, relacionados às mesmas ontologias. A segunda forma, conhecida como composição de alinhamentos [Zimmermann e Euzenat, 2006; Euzenat, 2008], gera um novo alinhamento da seguinte maneira: a partir de dois alinhamento conhecidos, $A(O_1, O_2)$ e $A(O_2, O_3)$, pode-se obter o alinhamento $A(O_1, O_3)$. Note que ambas as abordagens são diferentes da adotada aqui. Neste trabalho o intuito é, a partir de dois alinhamentos conhecidos com uma ontologia compartilhada, $A(O_1, O_2)$ e $A(O_3, O_2)$, encontrar o alinhamento $A(O_1, O_3)$.

A Figura 23 apresenta um exemplo de três ontologias do domínio *University* que serão utilizadas para ilustrar como o processo de junção acontece. Considere o conjunto de alinhamentos referente a essas ontologias: $\{A(O_1, O_2), A(O_1, O_3), A(O_2, O_3)\}$. Com a combinação de O_1 e O_3 é preciso gerar os novos alinhamentos relacionados às outras ontologias (O_2).

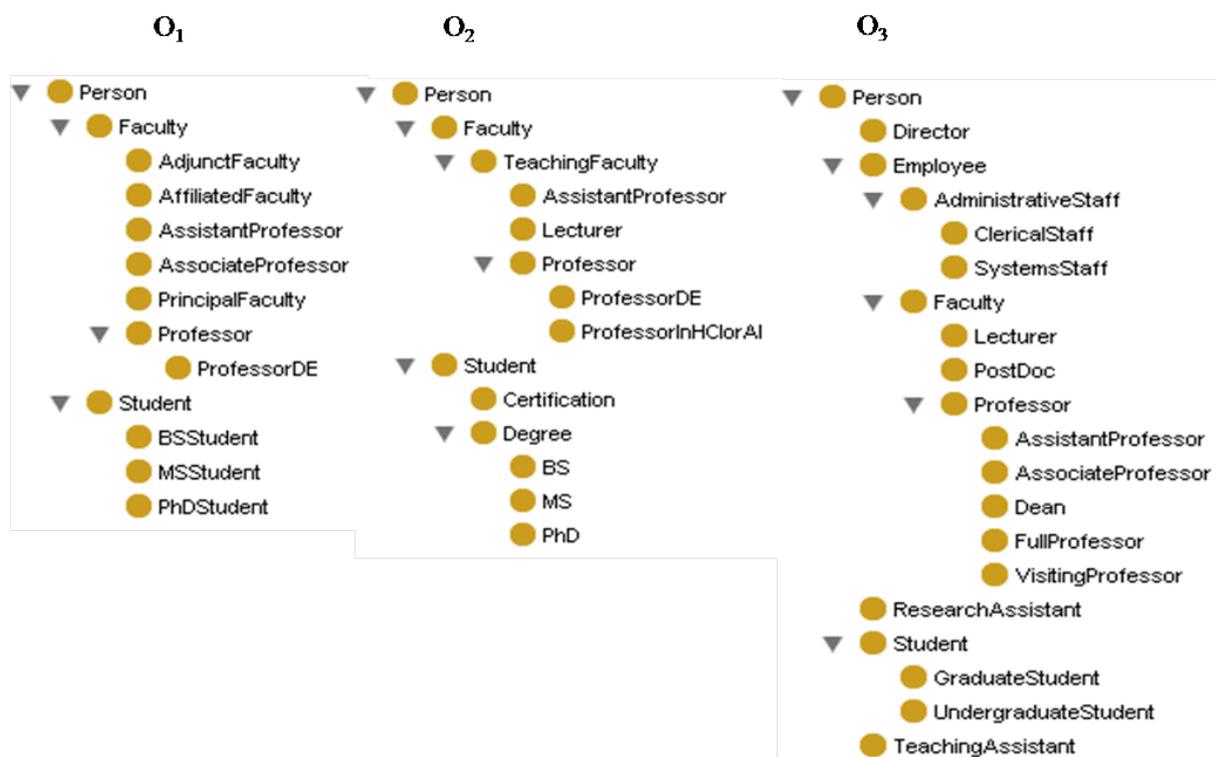


Figura 23. Exemplo de três ontologias do domínio Universidade

Para gerar o novo alinhamento ($A(O_c, O_2)$) o processo de junção utiliza os alinhamentos entre

⁷ <http://alignapi.gforge.inria.fr/>

as ontologias envolvidas no processo de combinação e a ontologia alvo. A Figura 24 ilustra essa junção onde algumas características da *Junção de Alinhamentos* podem ser observadas: somente uma entre duas correspondências iguais (X1 e Y1; X2 e Y2; X7 e Y6; X8 e Y9) dos alinhamentos de entrada são transferidas para o alinhamento resultante da junção (Z1, Z2, Z6, Z8, Z13). Como definido anteriormente, são consideradas “correspondências iguais” aquelas que possuem os mesmos elementos, exceto as medidas de confiança no relacionamento, as quais não estão sendo utilizadas. Todas as demais correspondências são consideradas diferentes e, por isso, todas são transferidas para o alinhamento resultante.

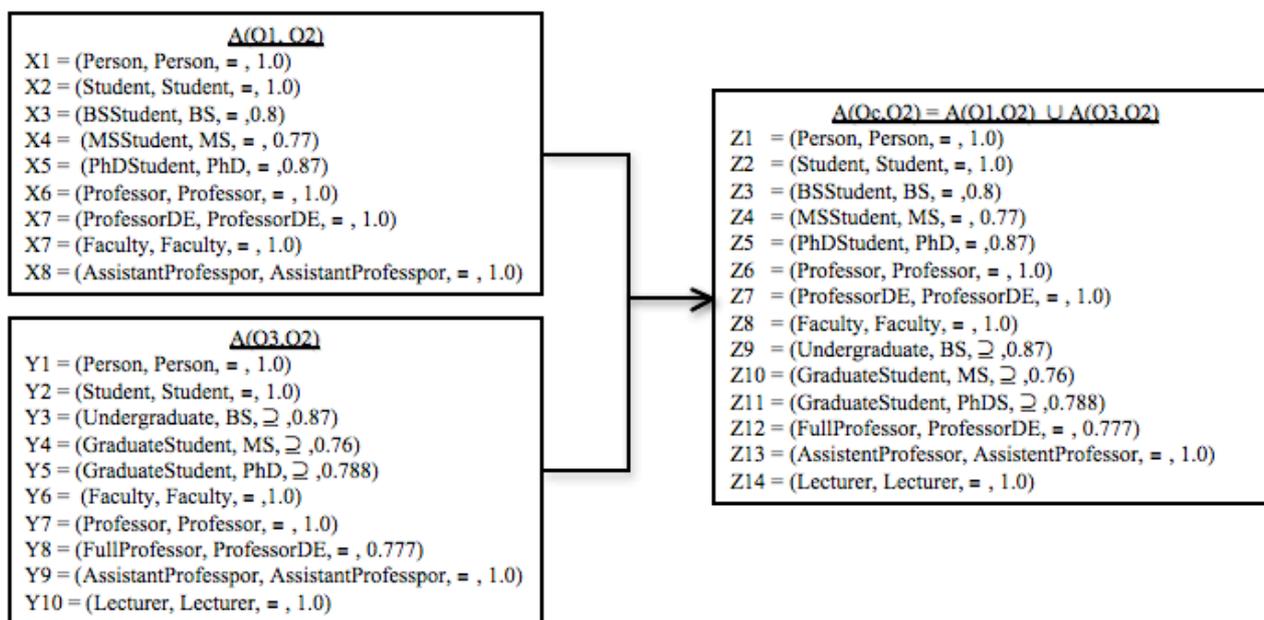


Figura 24. Exemplo de junção de dois alinhamentos

Um conceito importante existente no processo de junção é a idéia de “correspondência acumulada”, como a correspondência Z12 no novo alinhamento. Esta correspondência é dita acumulada porque, considerando que Oc utilize o elemento ‘ProfessorDE’ e o elemento utilizado por O2 também é ‘ProfessorDE’, bastaria existir a correspondência Z7. Caso fosse utilizado um processo de *matching*, ao invés da junção de alinhamentos, apenas a correspondência Z7 seria identificada. No entanto, este acúmulo é importante neste trabalho, já que não é feito o processo de *matching* após as combinações das ontologias. Dessa forma, as correspondências acumuladas podem ser úteis para identificações e combinações posteriores, durante o prosseguimento da execução do *merging* múltiplo.

Para demonstrar que a junção de alinhamentos é válida será mostrado que todas as

⁸ <http://islab.dico.unimi.it/hmatch/news.php>

correspondências que seriam identificadas por um processo de *matching* (A^M) também são identificadas nos alinhamentos gerados por meio da junção de alinhamentos (A^J) (Figura 25). Para essa demonstração será utilizada a idéia da prova de subconjuntos [Scheinerman, 2006].

Demonstração da Junção de Alinhamentos

Seja A^M o conjunto de correspondências obtidas a partir do processo de *matching* e A^J é o conjunto de correspondências obtidas a partir do processo de junção de alinhamentos.

Mostrar que $A^M \subseteq A^J$

Para mostrar que A^M é subconjunto de A^J é preciso mostrar que todos os elementos de A^M também são elementos de A^J , ou seja: $A^M \subseteq A^J \Leftrightarrow \forall v (v \in A^M \rightarrow v \in A^J)$.

Desse modo, suponha três ontologias quaisquer, $O1$, $O2$ e $O3$, e uma ontologia $O4$ onde $O4$ é o resultado da combinação de $O1$ e $O3$, ou seja, $O4 = O1 \cup O3$.

Considere agora os alinhamentos relacionados às ontologias:

1. $A(O1, O2) = \{(x_1, \dots, x_t) \mid 1 \leq k \leq t, x_k = (a, b, r, sim), a \text{ é elemento de } O1 \text{ e } b \text{ é elemento de } O2\}$;
2. $A(O3, O2) = \{(y_1, \dots, y_z) \mid 1 \leq h \leq z, y_h = (c, b, r, sim), c \text{ é elemento de } O3 \text{ e } b \text{ é elemento de } O2\}$;
3. $A(O1 \cup O3, O2) = \{(x_1, \dots, x_t) \cup (y_1, \dots, y_z)\}$, gerado a partir da junção de alinhamentos;
4. $A(O4, O2) = \{(v_1, \dots, v_w) \mid 1 \leq m \leq w, v_m = (d, b, r, sim) \text{ e } d \text{ é elemento de } O4\}$, gerado pelo *matching*.

PROVA

Elementos identificados a partir do processo de *matching*

\Rightarrow Supondo v uma correspondência qualquer de A^M , ou seja, $v \in A(O4, O2)$, onde $v = (d, b, r, sim)$, d é elemento de $O4$ e b é elemento de $O2$.

Como $O4$ é a união de $O1$ e $O3$ os elementos de $O4$ serão originados de $O1$ ou $O3$. Assim, $d = a$ ou $d = c$. Desse modo:

- Se $d = a$ então $v = (a, b, r, sim)$ onde $(a, b, r, sim) \in A(O1, O2)$;
- Se $d = c$ então $v = (c, b, r, sim)$ onde $(c, b, r, sim) \in A(O3, O2)$;

Assim: 1) A^J é formado por $\{(x_1, \dots, x_t) \cup (y_1, \dots, y_z)\}$, 2) $(a, b, r, sim) \in \{(x_1, \dots, x_t)\}$ e 3) $(c, b, r, sim) \in \{(y_1, \dots, y_z)\}$. Logo, $v \in A^J$.

Elementos identificados a partir do processo de junção de alinhamentos

\Leftarrow Supondo que $\{(v_1, \dots, v_w)\} \subseteq A^J$, significando que todo elemento de A^M é também elemento de A^J .

- Seja v uma correspondência qualquer de A^J .
- Seja $v = (d, b, r, sim)$ onde $d = a$ ou $d = c$.
- Se $d \in O4$ (se a for o elemento utilizado em $O4$ então $d = a$, senão $d = c$) então $v \in A^M$.

Assim, toda correspondência será elemento de A^M quando d for um elemento de $O4$.

Logo, $A^J \supseteq A^M$.

Portanto, se $v \in A^M$ então $v \in A^J$. Logo $A^M \subseteq A^J$.

Figura 25. Demonstração da validade do processo de Junção de Alinhamentos

O Algoritmo 4 está relacionado à junção de alinhamentos onde são passados como entrada dois alinhamentos compartilhando uma ontologia comum e retornando como saída um novo alinhamento. Nas linhas de 5 a 10 o algoritmo percorre o primeiro alinhamento ($A(O_i, O_u)$) adicionando cada correspondência ao novo alinhamento ($A(O_c, O_u)$). Em cada uma dessas correspondências é feita uma verificação no segundo alinhamento ($A(O_j, O_u)$) a procura de

correspondências diferentes para um mesmo elemento de O_u (linhas de 7 a 10). Caso encontre correspondências com o mesmo elemento de O_u (linha 8) então é feita uma comparação entre os elementos de O_i e O_j (linha 9). Se esses elementos foram iguais então as correspondências também são iguais e apenas uma das correspondências é adicionada ao novo alinhamento. Caso não sejam, ambas as correspondências são adicionadas ao novo alinhamento. Isso garante que o novo alinhamento não terá repetições. As linhas de 11 a 13 percorrem o segundo alinhamento para adicionar as correspondências ainda não consideradas.

Algoritmo JuncaoAlinhamento

Entrada: Alinhamentos $A(O_i, O_u)$, $A(O_j, O_u)$

Saída: Alinhamento $A(O_c, O_u)$

1. $A(O_i, O_u) = \{x_1 \dots x_t\}$
2. $A(O_j, O_u) = \{y_1 \dots y_z\}$
3. $x_k = (a, b, r, sim)$ onde $1 \leq k \leq t$ onde $a \in O_i$ e $b \in O_u$
4. $y_h = (c, b', r, sim)$ onde $1 \leq h \leq z$ onde $c \in O_j$ e $b' \in O_u$
5. Para $k \leftarrow 1$ até t faça
6. $A(O_c, O_u) \leftarrow A(O_c, O_u) \cup x_k$
7. Para $h \leftarrow 1$ até z faça
8. Se $(x_k = (a, b, r, sim) \wedge y_h = (c, b', r, sim) \wedge b = b')$ então
9. Se $a \neq c$ então
10. $A(O_c, O_u) \leftarrow A(O_c, O_u) \cup y_h$
11. Para $h \leftarrow 1$ até z faça
12. Se $(y_h \notin A(O_j, O_u))$ então
13. $A(O_c, O_u) \leftarrow A(O_c, O_u) \cup y_h$
14. Retorna $A(O_c, O_u)$

Algoritmo 4. Junção de Alinhamentos

Com o novo alinhamento criado, pode-se agora calcular a similaridade entre a nova ontologia gerada a partir da atividade de *Combinação de Ontologias* e a ontologia alvo. O valor calculado é adicionado na matriz. Isso se repete até que toda a matriz tenha sido preenchida, podendo ser então reiniciado o ciclo da segunda etapa para o processo de *merging*. Os valores existentes na matriz que não estejam relacionados às ontologias envolvidas na atividade de combinação são simplesmente copiados para a nova matriz.

Quando o último ciclo da etapa for realizado a matriz conterá apenas um elemento. Esse elemento será o nó raiz da árvore obtido como consequência do *merging* progressivo. Essa árvore é, então, passada para o usuário, que terá acesso a qualquer ontologia que tenha sido gerada durante o processo de *merging* múltiplo.

3.3 Considerações Finais

A estratégia proposta neste trabalho está voltada para uma situação específica onde é necessário realizar o *merging* de múltiplas ontologias, ou seja, é importante deixar claro que o foco principal deste trabalho não é propor um novo método para o *merging* de apenas duas ontologias. Para isso já existem diversas soluções na literatura que podem suprir situações variadas. Todavia, a estratégia MeMO busca dar suporte a situações onde seja necessário realizar múltiplos processos de *merging* de ontologias de forma automática. Dessa forma, diferentes soluções de *merging* encontradas na literatura poderão ser utilizadas em conjunto com a estratégia proposta, a fim de executar a atividade de combinação de duas ontologias. A única restrição é que essas soluções recebam como entrada um alinhamento pré-computado.

Devido à dificuldade em adaptar as soluções existentes para *merging* de duas ontologias, as quais estão vinculadas, de forma automática ou semi-automática, a um processo de *matching*, foi necessário implementar um algoritmo de *merging* de duas ontologias (disponível na literatura) e criar uma nova abordagem, chamada *junção de alinhamentos*, para reutilizar os alinhamentos fornecidos na entrada do processo de *merging* múltiplo e assim obter melhores resultados. O processo de junção utiliza a idéia de “correspondência acumulada” onde é gerado um novo alinhamento em que todas as correspondências presentes nos alinhamentos originais são guardadas para garantir a identificação das correspondências nos processos de *merging* seguintes, já que o processo de *matching* não é executado e não haveria como identificar as correspondências em tempo de processamento. Apesar de os alinhamentos guardarem essas correspondências que podem ser desnecessárias em alguns momentos, o resultado final não é prejudicado. Esse processo de junção de alinhamentos viabilizou o *merging* múltiplo automático de ontologias através do reuso de alinhamentos.

O uso da técnica de *clustering* também foi um fator determinante para a definição da estratégia. Com essa técnica, pode ser realizada uma análise prévia das ontologias e o agrupamento das mesmas, procurando combiná-las de acordo com suas semelhanças, o que possibilitou ontologias finais mais confiáveis.

Para mostrar que a estratégia proposta neste trabalho é válida, foram realizados experimentos que comprovaram seus benefícios. O próximo capítulo mostra os experimentos realizados e apresenta uma análise dos resultados obtidos.

CAPÍTULO 4

Avaliação da estratégia MeMO

Para a validação da estratégia proposta neste trabalho foram realizados experimentos considerando os seguintes tipos de ontologia: ontologias de referência (*gold standard*), ontologias resultantes do uso da estratégia MeMO e ontologias obtidas a partir de um processo de *merging* múltiplo aleatório. Durante os experimentos, as ontologias computadas a partir de processos de *merging* múltiplo (usando MeMO ou uma estratégia aleatória) foram comparadas com as ontologias de referência, a fim de que pudessem ser calculados os valores para as medidas de qualidade *Precision*, *Recall* e *F-Measure*.

Para a geração das ontologias globais (com o uso da estratégia MeMO ou da estratégia aleatória) foi desenvolvido um protótipo, o *MeMOTool*. Através deste protótipo o usuário fornece todos os alinhamentos relacionados às ontologias que se deseja combinar, retornando o resultado do *merging* múltiplo. Além disso, o protótipo possibilita a comparação das ontologias resultantes com uma ontologia de referência através do cálculo das medidas de qualidade.

Para a geração das ontologias foram selecionados diferentes conjuntos de ontologias do domínio acadêmico. Também é importante destacar que a ferramenta semi-automática (iPROMPT) foi utilizada para a geração das ontologias de referência. O restante deste capítulo está organizado como se segue. Na seção 4.1 serão apresentados os tipos de experimentos realizados neste trabalho para validação da estratégia proposta; na seção 4.2 será descrito o protótipo *MeMOTool* desenvolvido para a realização dos experimentos; na seção 4.3 serão mostrados os resultados obtidos a partir dos experimentos realizados; e finalmente na seção 4.4 é feita uma análise a respeito dos resultados obtidos destacando os pontos fortes da estratégia.

4.1 Descrição dos Experimentos

Para ilustrar como a estratégia proposta traz melhorias para o processo de *merging* de múltiplas ontologias foram realizados seis experimentos, onde cada experimento teve dois tipos de testes: um relacionado à estratégia MeMO e outro relacionado a estratégia aleatória. Em cada experimento foi utilizado um número variado de ontologias, tendo como objetivo verificar se cada

ontologia global computada retratava, de fato, uma visão global mais próxima da ideal. Através dos testes, tentou-se mostrar que a ordem de execução do *merging* das ontologias pode afetar a ontologia global resultante, de maneira que melhores resultados podem ser obtidos quando a similaridade entre as ontologias é levada em consideração, ou seja, ontologias mais similares devem ser combinadas primeiro.

De forma pontual, os principais objetivos dos experimentos foram:

- *Geração de uma ontologia modelo ou de referência (gold standard - O_{Ref})*: a ontologia O_{Ref} representa uma “visão global ideal” das ontologias iniciais a serem combinadas.. As ontologias modelo foram criadas por especialistas do domínio através da ferramenta iPROMPT [Noy e Musen, 2002].
- *Geração de uma ontologia global obtida com o uso da estratégia MeMO (O_{Comp}^M)*: a ontologia O_{Comp}^M representa uma ontologia global gerada pelo protótipo *MeMOTool*, ou seja, durante o processo de geração da ontologia O_{Comp}^M foram levadas em consideração as similaridades entre as ontologias originais.
- *Geração de uma ontologia global obtida com o uso de uma estratégia de combinação aleatória (O_{Comp}^A)*: a ontologia O_{Comp}^A representa uma ontologia global obtida como resultado de execuções com ordens aleatórias de múltiplos processos de *merging*. Estas ontologias também foram geradas utilizando o protótipo *MeMOTool*, excluindo-se apenas a etapa de análise da similaridade entre as ontologias.
- *Extração de medidas de qualidade entre O_{Comp}^M e O_{Ref} e entre O_{Comp}^A e O_{Ref}* : as medidas representam a proximidade das ontologias computados usando o *MeMOTool* em relação a ontologia modelo.

Para a realização dos experimentos, foram utilizadas vinte (20) ontologias (ANEXO C) no formato OWL-DL extraídas a partir de sites da Web (*Swoogle*⁹ e *OntoSelect*¹⁰). As ontologias utilizadas descrevem conceitos relacionados ao domínio acadêmico e possuem em média dezenove (19) classes e doze (12) propriedades, com profundidade (maior número de classes entre a raiz e as folhas na ontologia) variando entre um (1) e cinco (5), totalizando trezentas e noventa e cinco (395) classes e duzentas e quarenta e oito (248) propriedades.

Nos experimentos procurou-se identificar se o resultado final poderia ser afetado pelo número de ontologias a serem combinadas. Como ilustrado na Tabela 6, foram realizados

⁹ <http://swoogle.umbc.edu/>

¹⁰ <http://olp.dfki.de/ontoselect/>

experimentos com o número de ontologias iniciais variando entre 3 e 20. É importante observar que a quantidade inicial de alinhamentos utilizada em cada experimento variou de acordo com a quantidade de ontologias. Para geração destes alinhamentos, foi utilizada a ferramenta *Alignment API* [Euzenat, 2004]. Além disso, também foi levada em consideração a experiência dos especialistas do domínio, a fim de garantir que os alinhamentos de entrada sejam confiáveis.

Tabela 6. Experimentos realizados para validação da estratégia MeMO

	Numero de Ontologias	Numero de Alinhamentos
<i>Exp1</i>	3	3
<i>Exp2</i>	4	6
<i>Exp3</i>	8	28
<i>Exp4</i>	12	66
<i>Exp5</i>	16	120
<i>Exp6</i>	20	190

Como pode ser observado na Tabela 6, o número de ontologias iniciais variou para cada experimento. Assim, foi necessário gerar, em cada um dos experimentos: 1(uma) nova ontologia modelo O_{Ref} , 1(uma) ontologia O_{Comp}^M e dez (10) ontologias O_{Comp}^A . No caso das ontologias O_{Comp}^A , a geração de mais de uma ontologia foi realizada para que se pudesse calcular uma média (no caso entre 10), já que na estratégia aleatória os resultados podem variar a cada execução.

Como mencionado anteriormente, os experimentos foram realizados com o objetivo de demonstrar que a ontologia global obtida com o uso da estratégia MeMO seria “mais próxima” da ontologia global ideal (*gold standard*) do que uma ontologia global obtida a partir da execução aleatória de múltiplos processos de *merging*. No entanto, para que tal demonstração seja realizada se faz necessária a utilização de métricas que possam comparar as características das ontologias, no sentido de determinar o quanto uma ontologia “está próxima” de outra. No caso específico deste trabalho, o quanto as ontologias geradas estão próximas à ontologia modelo. Dessa forma, para realização desta análise comparativa entre as ontologias, foram utilizadas as medidas citadas na Seção 2.3: *Precision*, *Recall* e *F-measure* [Dellschaft e Staab, 2006], sendo levadas em consideração as camadas léxica e estrutural das ontologias. A análise da camada léxica consistiu em comparar os termos das ontologias considerando o casamento exato de strings. A análise da camada estrutural buscou comparar as posições dos conceitos nas hierarquias das duas ontologias sendo analisadas.

A seguir, será apresentado um resumo das medidas de qualidade adotadas neste trabalho. É válido ressaltar que todas as medidas são calculadas em termos percentuais. Para as definições abaixo considere que o termo “ontologia global” se refere à ontologia obtida a partir do uso da

estratégia MeMO ou através de combinações aleatórias e que o termo “ontologia ideal” representa a ontologia modelo.

- *LP - Lexical Precision*: mede quantas classes de uma ontologia global computada estão corretas com relação à ontologia ideal, considerando apenas o casamento de strings. Ou seja, quantas classes presentes na ontologia global possuem o mesmo nome que uma classe presente na ontologia ideal.
- *LR – Lexical Recall*: mede quantas classes da ontologia ideal estão, de fato, presentes na ontologia global computada, dentre as que deveriam estar presentes considerando apenas o casamento de strings.
- *TP – Taxonomical Precision*: mede quantos elementos presentes na hierarquia de um conceito em uma ontologia global computada estão corretos em relação aos elementos que estão presentes na hierarquia do conceito correspondente na ontologia ideal.
- *TR – Taxonomical Recall*: mede quantos elementos presentes na hierarquia de um conceito na ontologia ideal estão, de fato, presentes na ontologia global computada, dentre aqueles que deveriam estar.
- *TF (Taxonômica F-medida) e LF (Lexical F-medida)* – medidas harmônicas que equilibram *Precision* e *Recall*, pois estas podem ter valores extremos (maximizados ou minimizados) que podem favorecer interpretações erradas sobre o resultado da diferença entre as ontologias.

Para a obtenção dos valores das medidas acima foi utilizada a ferramenta *OntEval*¹¹, que foi adicionada à *MeMOTool*. *OntEval* recebe como entrada duas ontologias e devolve como resultado um conjunto de valores que especificam as medidas *Precision*, *Recall* e *F-Measure* (léxica e taxonômica), que caracterizam a “relação de proximidade” entre as ontologias analisadas.

Finalmente, uma vez determinada a maneira de se comparar duas ontologias, as seguintes comparações foram realizadas: i) Ontologia O_{Ref} com ontologia O_{Comp}^M e ii) Ontologia O_{Ref} com cada uma das 10 ontologias O_{Comp}^A , sendo calculada uma média para cada uma das medidas consideradas. A partir destas comparações, pudemos analisar qual estratégia gerou a ontologia mais próxima à ideal e, ainda, qual o percentual de aproximação. Estas análises serão descritas nas próximas seções. Todos os experimentos foram realizados em uma máquina HP *Pavilion* com Linux Ubuntu 8.04, processador Intel Core™2Duo e 3GB de memória.

¹¹ <http://nlp.shef.ac.uk/abraxas/index.html>

4.2 O Protótipo MeMOTool

Para a validação da estratégia proposta foi desenvolvido um protótipo, chamado *MeMOTool*, que disponibiliza diversas funções para a realização de atividades envolvendo um conjunto de ontologias e alinhamentos, como apresentado na Figura 26 e descrito a seguir.

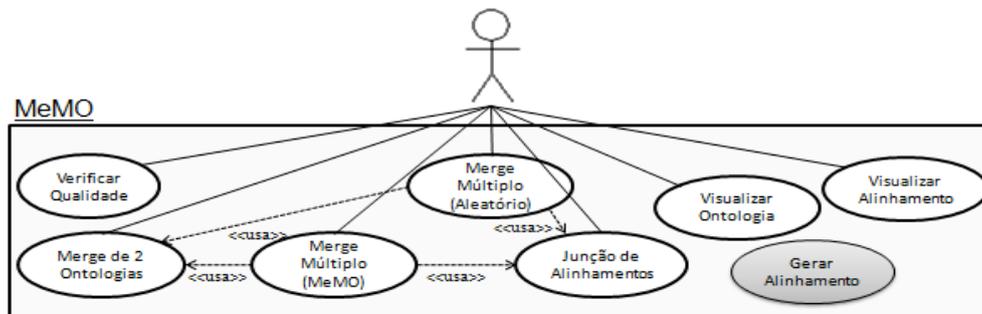


Figura 26. Funcionalidade da MeMOTool

- *Verificar Qualidade*: duas ontologias são comparadas através de medidas qualitativas (*Precision*, *Recall*, *F-Measure*) [Dellschaft e Staab, 2006];
- *Merging de duas ontologias*: duas ontologias são combinadas de acordo com o alinhamento entre elas que é informado pelo usuário, e uma ontologia é retornada;
- *Merging Múltiplo (MeMO)*: um conjunto de ontologias são combinadas de acordo com a estratégia MeMO, onde um conjunto de alinhamentos referentes a essas ontologias são informados como entrada. As combinações são realizadas duas a duas utilizando a função de *merging de duas ontologias*, de acordo com a estratégia MeMO. O resultado final é uma árvore binária onde a raiz é a ontologia global e as folhas são as ontologias iniciais;
- *Merging Múltiplo (Aleatório)*: um conjunto de ontologias é combinado aleatoriamente duas a duas utilizando a função de *merging de duas ontologias*, onde um conjunto de alinhamentos referentes a essas ontologias são passados como entrada. O resultado final é uma árvore binária onde a raiz é a ontologia global e as folhas são as ontologias iniciais;
- *Junção de Alinhamentos*: um novo alinhamento é criado a partir de dois alinhamentos que compartilham uma ontologia comum. Essa função é utilizada para facilitar o processo automático de *merging* múltiplo, evitando chamadas a processo de *matching*;
- *Visualizar Ontologia*: uma ontologia pode ser visualizada através da interface da ferramenta, onde são exibidas suas classes e propriedades;

- *Visualizar Alinhamento*: um alinhamento pode ser visualizado através da interface da ferramenta, permitindo a visualização de todo o seu conteúdo;
- *Gerar Alinhamento*: essa funcionalidade permite que o usuário crie os alinhamentos entre as ontologias diretamente pela MeMOTOol.

A funcionalidade para *Geração de Alinhamentos* ainda não foi desenvolvida, além de também não ser possível manipular os alinhamentos e ontologias no sentido de alterá-los de acordo com a intuição do usuário.

A Figura 27 ilustra a janela principal da ferramenta. É possível observar que o protótipo disponibiliza uma área relacionada aos alinhamentos e outra relacionada às ontologias. A Figura 27.A contém a lista de alinhamentos fornecida pelo usuário, a Figura 27.B contém um arquivo de alinhamento (.rdf) a ser analisado pelo usuário, a Figura 27.C contém a árvore binária resultante do processo de *merging* múltiplo e a Figura 27.D contém uma ontologia selecionada pelo usuário para ser visualizada.

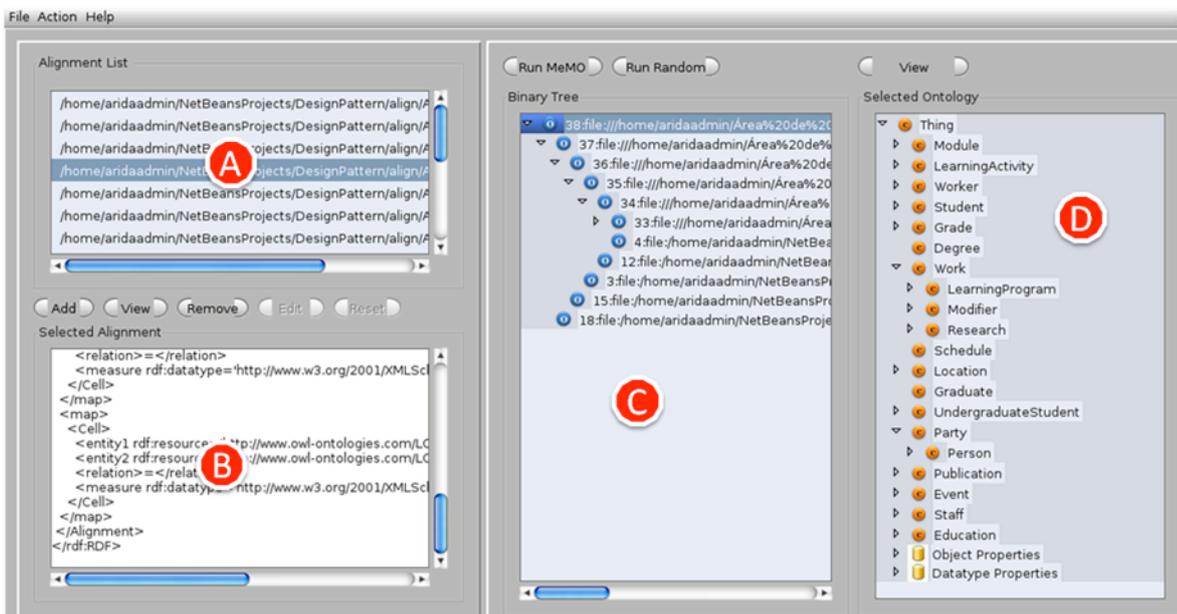


Figura 27. Interface gráfica da ferramenta MeMO

4.3 Resultados dos experimentos

Como dito anteriormente, foram realizados experimentos onde, em cada um, foram gerados três tipos de ontologias: uma ideal (O_{Ref}), uma computada utilizando a estratégia MeMO (O_{Comp}^M) e outra computada utilizando uma estratégia aleatória (O_{Comp}^A). Em geral foram feitas dez execuções

da estratégia aleatória para cada experimento, visto que esta estratégia gera diferentes ontologias. O intuito dos experimentos é mostrar a diferença da estratégia aleatória em relação a MeMO e mostrar como a MeMO é mais estável.

Em alguns casos, a estratégia aleatória obteve resultados iguais ou superiores a estratégia MeMO, porém foram resultados eventuais e sem garantias. Para efeito de demonstração, será apresentada aqui apenas uma das dez ontologias obtidas em cada experimento. A escolha da ontologia aleatória para ilustração nesta seção foi de acordo com a facilidade didática para demonstrar as diferenças entre as ontologias. Juntamente com a ontologia obtida de forma aleatória também serão apresentadas a ontologia modelo (O_{Ref}), gerada de forma semi-automática com o iPROMPT, e a ontologia gerada com o uso da estratégia MeMO, para que se possa visualmente notar as diferenças estruturais entre as ontologias. Outras ontologias aleatórias podem ser visualizadas no APÊNDICE B. Nem todas as dez ontologias obtidas estão presentes no anexo, pois algumas ou se repetiam ao longo das dez execuções; ou eram iguais a estratégia MeMO; ou eram resultados pouco expressivos, sendo colocados aqui apenas os que melhor facilitassem as explicações.

A seguir, são apresentados os seis experimentos que foram realizados.

- ***Experimento 1***

O primeiro experimento utiliza três ontologias iniciais, sendo, neste caso, necessárias apenas seis execuções da estratégia devido às possibilidades de combinações. A Tabela 7 mostra três das ontologias geradas no primeiro experimento: O_{Ref} , O_{Comp}^M e uma das ontologias aleatórias geradas, O_{Comp}^A . Pelo resultado mostrado na tabela, sendo facilmente percebido pela própria ontologia como pelas medidas calculadas (LP , LR , LF , TP , TR , TF), pode-se notar que a O_{Comp}^M é mais próxima da O_{Ref} , enquanto que a O_{Comp}^A é um pouco diferente. Essa diferença pode ser vista, por exemplo, pela posição da classe *Employee* e, conseqüentemente, suas subclasses. Apesar da estratégia aleatória também ter gerado em alguns momentos (duas das seis execuções) um resultado igual a O_{Comp}^M , esse não foi um resultado constante, sendo por isso considerado não confiável.

Tabela 7. Resultados do primeiro experimento

Exp 1 – três ontologias envolvidas		
O_{Ref}	O_{Comp}^M	O_{Comp}^A
	LP = 1.0; LR = 1.0; LF = 1.0 TP = 1.0; TR = 1.0; TF = 1.0	LP = 1.0; LR = 1.0; LF = 1.0 TP = 1.0; TR = 0.854; TF = 0.921
<ul style="list-style-type: none"> ▼ ● Education <ul style="list-style-type: none"> ▶ ● Course ▶ ● Organization ▼ ● Person <ul style="list-style-type: none"> ● Director ▼ ● Employee <ul style="list-style-type: none"> ● Administrator ▶ ● Faculty ● Secretary ▶ ● Student ● Research 	<ul style="list-style-type: none"> ▼ ● Education <ul style="list-style-type: none"> ▶ ● Course ▶ ● Organization ▼ ● Person <ul style="list-style-type: none"> ● Director ▼ ● Employee <ul style="list-style-type: none"> ● Administrator ▶ ● Faculty ● Secretary ▶ ● Student ● Research 	<ul style="list-style-type: none"> ▼ ● Education <ul style="list-style-type: none"> ▶ ● Course ▼ ● Employee <ul style="list-style-type: none"> ● Administrator ▼ ● Faculty <ul style="list-style-type: none"> ● Lecturer ● PostDoc ● Secretary ▶ ● Organization <ul style="list-style-type: none"> ▼ ● Person <ul style="list-style-type: none"> ● Director ▶ ● Professor ▶ ● Student

• **Experimento 2**

A partir do segundo experimento foram realizadas dez execuções da estratégia aleatória. Apesar de ser possível mais de dez possibilidades de combinações, dez foi suficiente para atingir os objetivos dos experimentos que é apenas de validação da estratégia. A Tabela 8 ilustra esse segundo experimento, onde pode ser notada uma diferença da O_{Comp}^M para a O_{Comp}^A na classe *Professor*. Apesar de ser pequena, essa diferença pode causar impacto em aplicações que venham a reutilizar essa ontologia.

Tabela 8. Resultados do segundo experimento

<i>Exp 2 - quatro ontologias envolvidas</i>		
O_{Ref}	O_{Comp}^M	O_{Comp}^A
	LP = 1.0; LR = 1.0; LF = 1.0 TP = 1.0; TR = 1.0; TF = 1.0	LP = 1.0; LR = 1.0; LF = 1.0 TP = 1.0; TR = 0.938; TF = 0.968
<ul style="list-style-type: none"> ▼ Education <ul style="list-style-type: none"> ▶ Course ▶ Organization ▼ Person <ul style="list-style-type: none"> ● Director ▼ Employee <ul style="list-style-type: none"> ● Administrator ▼ Faculty <ul style="list-style-type: none"> ● Lecturer ● PostDoc ▶ Professor ● Secretary ▶ Student ▶ Publication ● Research ● University 	<ul style="list-style-type: none"> ▼ Education <ul style="list-style-type: none"> ▶ Course ▶ Organization ▼ Person <ul style="list-style-type: none"> ● Director ▼ Employee <ul style="list-style-type: none"> ● Administrator ▼ Faculty <ul style="list-style-type: none"> ● Lecturer ● PostDoc ▶ Professor ● Secretary ▶ Student ▶ Publication ● Research ● University 	<ul style="list-style-type: none"> ▼ Education <ul style="list-style-type: none"> ▶ Course ▶ Organization ▼ Person <ul style="list-style-type: none"> ● Director ▼ Employee <ul style="list-style-type: none"> ● Administrator ▼ Faculty <ul style="list-style-type: none"> ● Lecturer ● PostDoc ● Secretary ▶ Professor ▶ Student ▶ Publication ● Research ● University

• *Experimento 3*

O terceiro experimento utiliza oito ontologias iniciais. A Tabela 9 mostra três das ontologias utilizadas no terceiro experimento. Nesse ponto em que as ontologias estão sendo ilustradas nas figuras, pode-se ver a diferença em relação ao elemento *Professor* que na O_{Comp}^M subiu dois níveis na hierarquia, enquanto que a O_{Comp}^A subiu um nível. Vê-se, pelas próprias medidas, uma diferença pequena entre as ontologias, mas ainda assim a MeMO fornece um resultado mais próximo do ideal que a ontologia obtida através da estratégia aleatória.

Tabela 9. Resultados do terceiro experimento

<i>Exp 3 - oito ontologias envolvidas</i>		
O_{Ref}	O_{Comp}^M	O_{Comp}^A
	LP = 0.979; LR = 0.979; LF = 0.979 TP = 0.973; TR = 0.901; TF = 0.935	LP = 0.979; LR = 0.979; LF = 0.979 TP = 0.916; TR = 0.934; TF = 0.925
<ul style="list-style-type: none"> ▼ Education <ul style="list-style-type: none"> ▶ Organization ▼ Person <ul style="list-style-type: none"> ● Director ▼ Employee <ul style="list-style-type: none"> ● Administrator ▼ Faculty <ul style="list-style-type: none"> ● Lecturer ● PostDoc ▶ Professor ● Secretary ▶ Research ▶ Student ▶ Publication ▼ Work <ul style="list-style-type: none"> ▶ Course 	<ul style="list-style-type: none"> ▼ Education <ul style="list-style-type: none"> ▶ Organization ▼ Person <ul style="list-style-type: none"> ● Director ▼ Employee <ul style="list-style-type: none"> ● Administrator ▼ Faculty <ul style="list-style-type: none"> ● Lecturer ● PostDoc ● Secretary ▶ Professor ▶ Student ▶ Publication ▶ Research ▼ Work <ul style="list-style-type: none"> ▶ Course 	<ul style="list-style-type: none"> ▼ Education <ul style="list-style-type: none"> ▶ Organization ▼ Person <ul style="list-style-type: none"> ● Director ▼ Employee <ul style="list-style-type: none"> ● Administrator ▼ Faculty <ul style="list-style-type: none"> ● Lecturer ● PostDoc ▶ Professor ● Secretary ▶ Student ▶ Publication ▶ Research ▶ Work

• *Experimento 4*

A Tabela 10 mostra as três ontologias geradas no quarto experimento, onde foram consideradas doze ontologias. Nesse experimento é possível notar uma diferença maior entre os resultados das ontologias globais tanto da estratégia aleatória quanto da MeMO. Em O_{Comp}^M , por exemplo, os elementos *Person*, *Project*, *AcademicStaff*, *Secretary*, *Professor* e *Course* estão em posições diferentes na hierarquia em relação a O_{Ref} . A O_{Comp}^A obteve uma estrutura semelhante a O_{Comp}^M , porém obteve a posição para *Person* semelhante a O_{Ref} . Em diferentes pontos de ambas as estruturas uma teve uma representação melhor que a outra em relação à O_{Ref} , mas nenhuma foi unânime no total de sua representação.

Tabela 10. Resultados do quarto experimento

Exp 4 - doze ontologias envolvidas		
O_{Ref}	O_{Comp}^M	O_{Comp}^A
	LP = 0.925; LR = 0.986; LF = 0.954 TP = 0.842; TR = 0.845; TF = 0.843	LP = 0.948; LR = 0.973; LF = 0.96 TP = 0.712; TR = 0.9; TF = 0.795

• **Experimento 5**

A Tabela 11 mostra três das ontologias utilizadas no quinto experimento, onde foram consideradas 16 ontologias. Neste caso, vê-se que a diferença em relação a O_{Ref} é ainda maior do que os experimentos anteriores para as duas estratégias, como pode ser notado pelo elemento *Person* que está em posição diferente de O_{Ref} em ambas as ontologias computadas. Isso acontece devido a quantidade de ontologias que é maior, tendo, conseqüentemente, ontologias intermediarias mais complexas (ex.: numero de classes). Essa complexidade dificulta o algoritmo de combinação em suas tomadas de decisões para obter uma ontologia resultante mais correta. O algoritmo analisa

quem são as classes filhas e irmãs da classe a ser adicionada na nova ontologia, além de verificar na nova ontologia se estas já estão adicionadas e se são mais específicas que na ontologia atual. Por isso, quanto mais complexas vão ficando as ontologias, mais o algoritmo fica suscetível a adicionar a nova classe em posições menos corretas e, conseqüentemente, propagar imprecisões.

Tabela 11. Resultados do quinto experimento

<i>Exp 5 - desesseis ontologias envolvidas</i>		
O_{Ref}	O_{Comp}^M	O_{Comp}^A
	LP = 0.907; LR = 0.92; LF = 0.913 TP = 0.595; TR = 0.664; TF = 0.628	LP = 0.905; LR = 0.953; LF = 0.928 TP = 0.546; TR = 0.716; TF = 0.619
<ul style="list-style-type: none"> ▼ ● Modifier <ul style="list-style-type: none"> ▶ ● Education ▶ ● GradingType ▶ ● Position ● Semester ● Sex ▶ ● StudentStatus ▶ ● StudyLevel ▶ ● Module ▼ ● Party <ul style="list-style-type: none"> ▶ ● Organization ▼ ● Person <ul style="list-style-type: none"> ▶ ● Alumnus ● Collaborator ▶ ● Director ● Funcionarios ● GuestSpeaker ● Hermaphrodite ● MasterThesisAdvisor06 ▶ ● PrincipallInvestigator ● Sponsor ▶ ● Student ● Visitor ▼ ● Worker <ul style="list-style-type: none"> ▼ ● Employee <ul style="list-style-type: none"> ● Manager ● Secretary ▶ ● Staff ● StudyProgramLeadPosition ● TechnicalStaff 	<ul style="list-style-type: none"> ▼ ● Education ≡ Education <ul style="list-style-type: none"> ● BachelorEducation ● CollegeEducation ● DoctoralEducation ● MasterEducation ▶ ● Organization ▼ ● Person <ul style="list-style-type: none"> ▶ ● Alumnus ● Collaborator ▶ ● Director ● Funcionarios ● GuestSpeaker ● Hermaphrodite ● MasterThesisAdvisor06 ▶ ● PrincipallInvestigator ▶ ● Professor ● ResearchAssistant ● Sponsor ▶ ● Staff ● TeachingAssistant ● Visitor ▶ ● Worker ▶ ● Publication ● ResearcherOntology ● SecondaryEducation ▶ ● Work <ul style="list-style-type: none"> ▶ ● Event ▶ ● Grade ▶ ● LearningActivity ▶ ● Location ▶ ● Module ▶ ● Party ▶ ● Schedule 	<ul style="list-style-type: none"> ▼ ● Education ≡ Education <ul style="list-style-type: none"> ● BachelorEducation ● CollegeEducation ▶ ● Course ● DoctoralEducation ● MasterEducation ▶ ● Project ▶ ● Publication ▶ ● ResearcherOntology <ul style="list-style-type: none"> ▶ ● Organization ▼ ● Person <ul style="list-style-type: none"> ▶ ● Alumnus ● Collaborator ▶ ● Director ● Funcionarios ● GuestSpeaker ● Hermaphrodite ● MasterThesisAdvisor06 ▶ ● PrincipallInvestigator ▶ ● Research ● Sponsor ▶ ● Staff ▶ ● Student ≡ Student ● TeachingAssistant ● Visitor ▶ ● Worker ▼ ● Work <ul style="list-style-type: none"> ▶ ● LearningProgram ▶ ● Modifier ● SecondaryEducation ▶ ● Event ▶ ● Grade ▶ ● LearningActivity ▶ ● Location ▶ ● Module ▶ ● Party ▶ ● Schedule ▶ ● UndergraduateStudent

• **Experimento 6**

A Tabela 12 mostra o sexto e último experimento, onde foram consideradas 20 ontologias. Nesse experimento a diferença é ainda maior do que os experimentos anteriores, ficando inviável a reutilização das ontologias resultantes de ambas as estratégias sem uma interferência de um

especialista. Essas grandes diferenças se devem tanto pelo algoritmo que precisa ser refinado para tomar decisões mais acertadas, como pelos alinhamentos, que precisariam ser mais enriquecidos. Com melhorias nesses dois pontos, a qualidade do resultado final pode ser aumentada, possibilitando o reuso das ontologias, deixando a estratégia MeMO ainda mais eficiente.

Tabela 12. Resultados do quinto experimento

<i>Exp 6 - vinte ontologias envolvidas</i>		
O_{Ref}	O_{Comp}^M	O_{Comp}^A
	LP = 0.935; LR = 0.96; LF = 0.947 TP = 0.72; TR = 0.713; TF = 0.717	LP = 0.898; LR = 0.94; LF = 0.918 TP = 0.787; TR = 0.76; TF = 0.773
<ul style="list-style-type: none"> ▶ ● Event ▶ ● Grade ▶ ● LearningActivity ▶ ● LearningProgram ▶ ● Location ▼ ● Modifier <ul style="list-style-type: none"> ▶ ● Education ▶ ● GradingType ▼ ● Position <ul style="list-style-type: none"> ● AcademicPosition ● DepartmentHeadPosition ● Semester ● Sex ▶ ● StudentStatus ▶ ● Module ▼ ● Party <ul style="list-style-type: none"> ▶ ● Organization ▼ ● Person <ul style="list-style-type: none"> ▶ ● Alumnus ● Collaborator ▶ ● Director ● GuestSpeaker ● Hermaphrodite ● MasterThesisAdvisor06 ● PrincipallInvestigator ● ResearchAssistant ● Sponsor ▶ ● Student ● Visitor ▼ ● Worker <ul style="list-style-type: none"> ▼ ● Employee <ul style="list-style-type: none"> ● Manager ● Secretary ▼ ● Staff <ul style="list-style-type: none"> ▶ ● AcademicStaff ▶ ● NonAcademicStaff ● StudyProgramLeadPosition ● TechnicalStaff 	<ul style="list-style-type: none"> ▼ ● Education <ul style="list-style-type: none"> ● BachelorEducation ● CollegeEducation ● DoctoralEducation ● MasterEducation ▶ ● Organization ▼ ● Person ≡ Person <ul style="list-style-type: none"> ▶ ● Director ● Funcionarios ● Hermaphrodite ● MasterThesisAdvisor06 ▶ ● Professor ● ResearchAssistant ▶ ● Student ● TeachingAssistant ▼ ● Worker <ul style="list-style-type: none"> ▼ ● Employee <ul style="list-style-type: none"> ▶ ● AcademicStaff ▶ ● Alumnus ● Collaborator ● GuestSpeaker ▶ ● NonAcademicStaff ● PrincipallInvestigator ● Secretary ● Sponsor ● Visitor ▶ ● Publication ● ResearcherOntology ● SecondaryEducation ▼ ● Work <ul style="list-style-type: none"> ▶ ● Course ▶ ● LearningProgram ▶ ● Modifier ▶ ● Project ▶ ● Research ▶ ● Event ▶ ● Grade ▶ ● LearningActivity ▶ ● Location ▶ ● Module ▶ ● Party ● Schedule 	<ul style="list-style-type: none"> ▶ ● Education ▶ ● Event ▶ ● Grade ▶ ● Graduate ▶ ● LearningActivity ▶ ● Location ▶ ● Module ▼ ● Party <ul style="list-style-type: none"> ▶ ● Organization ▼ ● Person <ul style="list-style-type: none"> ▶ ● Alumnus ● Collaborator ▶ ● Director ● Funcionarios ● GuestSpeaker ● Hermaphrodite ● MasterThesisAdvisor06 ● PrincipallInvestigator ● ResearchAssistant ● Sponsor ● TeachingAssistant ● Visitor ▼ ● Worker <ul style="list-style-type: none"> ▼ ● Employee <ul style="list-style-type: none"> ▼ ● Faculty ≡ Faculty <ul style="list-style-type: none"> ● AdjunctFaculty ● AffiliatedFaculty ● AssociateProfessor ● FacultyMember ▶ ● Position ● PostDoc ● PrincipalFaculty ▶ ● Professor ▶ ● Staff ▶ ● TeachingFaculty ● Schedule ▶ ● Student ▶ ● Ungraduate ▼ ● Work <ul style="list-style-type: none"> ▶ ● Course ▶ ● LearningProgram ▶ ● Modifier ▶ ● Research

A seguir serão feitas as análises dos resultados obtidos nos experimentos descritos acima.

4.4 Análise dos Resultados dos Experimentos

Para cada experimento foram obtidos dois conjuntos de valores para as medidas LP , LR , TP , TR , TF : o primeiro conjunto (MeMO) especifica os valores para as medidas de qualidade obtidos a partir da comparação de O_{Ref} com O_{Comp}^M , e o segundo conjunto (Aleatório) especifica as médias dos valores obtidos a partir da comparação da ontologia ideal com as 10 versões de ontologias do tipo O_{Comp}^A . Em outras palavras, o primeiro conjunto denota os valores obtidos com o uso da estratégia MeMO e o segundo conjunto denota os valores obtidos com o uso de uma estratégia aleatória. É importante salientar que, durante esta análise, nos concentraremos nos valores de qualidade taxonômicas (TR , TP e TF), pois estes são os que apresentam os maiores graus de variação em todos os experimentos, para ambas as estratégias. Além disso, são as medidas mais pertinentes com relação à avaliação da qualidade do *merging*, pois o principal ponto a ser comparado é a estrutura da ontologia resultante com relação à ontologia modelo. Os resultados dos experimentos realizados podem ser vistos na Tabela 13, onde são exibidos os valores para as medidas de qualidade obtidas através do uso da ferramenta *OntEval*.

Tabela 13. Resultados dos experimentos

Estratégia	Número de Ontologias	Número de Alinhamentos	LP	LR	TP	TR	TF	Tempo(seg)
Exp1								
MeMO	3	3	1.0	1.0	1.0	0.976	0.988	4.24
Aleatório			1.0	1.0	1.0	0.915	0.954	3.9
Exp2								
MeMO	4	6	1.0	1.0	1.0	1.0	1.0	4.99
Aleatório			1.0	1.0	0.95	0.92	0.97	4.5
Exp3								
MeMO	8	28	0.98	0.98	1.0	0.91	0.954	8.15
Aleatório			0.98	0.98	0.97	0.94	0.931	8.2
Exp4								
MeMO	12	66	0.926	0.987	0.842	0.854	0.843	15.229
Aleatório			0.907	0.964	0.703	0.773	0.735	16.4
Exp5								
MeMO	16	120	0.907	0.92	0.595	0.66	0.63	24.5
Aleatório			0.897	0.952	0.622	0.714	0.66	24.16
Exp6								
MeMO	20	190	0.935	0.96	0.72	0.713	0.717	34.8
Aleatório			0.91	0.94	0.69	0.701	0.697	31.72

Em relação ao tempo de execução a MeMO teve um melhor resultado, apesar de, em alguns momentos, ter um tempo de execução maior, o que se deve aos cálculos de similaridade

necessários. No entanto, essa diferença é muito pequena, não podendo o tempo de execução ser um fator determinante para o uso ou não da estratégia

Como pode ser observado através da Tabela 8, na maioria dos experimentos, a utilização da estratégia MeMO leva a resultados mais próximos a 1.0 (menor distância para a ontologia ideal) do que uso da estratégia aleatória. Além disso, é possível observar que já no primeiro experimento (*Exp1*), o uso da estratégia MeMO apresenta-se melhor que a estratégia aleatória nos valores de TR e TF. Já no segundo experimento (*Exp2*), com quatro ontologias combinadas, a diferença torna-se visível também na medida TP. Tal diferença se acentua no *Exp3* e *Exp4*, com oito e doze ontologias combinadas, respectivamente.

Os resultados das medidas de qualidade estruturais começam a se tornar mais distantes da ontologia ideal (valores *TP* e *TR* abaixo de 0.72) em todos os exemplares de *Exp5* e *Exp6*. Isto reflete uma perda natural e gradual da consistência do *merging* múltiplo por conta da automação do processo. O que estamos chamando de “perda de consistência” poderia ser minimizado, evidentemente, pela interferência de *expertise* humana no domínio de conhecimento da ontologia. No entanto, mesmo com valores um pouco mais distantes de 1.0, a estratégia MeMO ainda se apresenta superior à aleatória. A única exceção fica por conta do experimento 5 (*Exp5*), onde as execuções aleatórias superaram a MeMO em algumas medidas. Isto porque, neste experimento particularmente, as execuções aleatórias chegaram próximas aos seus melhores resultados. No entanto, exatamente por se tratar de uma estratégia aleatória, estes resultados podem se apresentar diferentes para o mesmo número de ontologias em um outro teste. Ao contrário da MeMO, onde o valor apresentado é constante.

Como discutido anteriormente, as medidas *Precision* (TP) e *Recall* (TR) não devem ser consideradas isoladamente. Fez-se necessária, então, a utilização da medida *F-measure* (TF), a qual faz um balanceamento entre as medidas TP e TR. Logo, a TF constitui a principal medida, dentre todas as que foram apresentadas na Tabela 8, para comparar as duas estratégias. Por este motivo, na Figura 28, é exibido um quadro comparativo entre as medidas de TF obtidas nos seis experimentos realizados, apresentando um gráfico em barras para melhor ilustrar as diferenças observadas nos valores tanto da estratégia MeMO quanto da estratégia aleatória. No caso da aleatória, são exibidos 3 valores: a média das 10 execuções, o pior caso e o melhor caso.

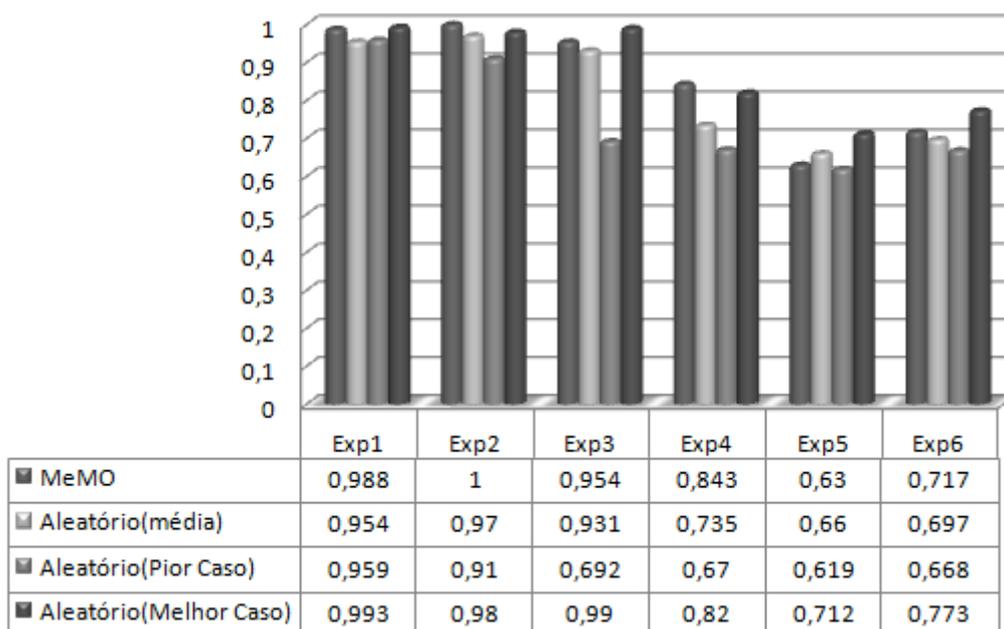


Figura 28. Quadro comparativo entre as medidas *F-measure* encontradas nos experimentos

Com base no quadro comparativo da Figura 28, é possível reafirmar a superioridade da estratégia MeMO, uma vez que os valores obtidos são mais próximos de 1.0 (ou exatamente 1.0), significando que a ontologia global resultante é bem próxima da ontologia ideal. Mais especificamente, pode ser observado que o uso da estratégia MeMO levou a um melhor resultado que a média e que o pior caso da estratégia aleatória em todos os experimentos, exceção feita ao *Exp5*, já comentada anteriormente. Além disso, chama a atenção o fato de que a MeMO conseguiu superar todos os piores casos e três dos seis melhores casos da estratégia aleatória, o que consubstancia a relevância da estratégia proposta. Outrossim, nota-se que os resultados da MeMO foram extremamente confiáveis para a medida *TF* (variando de 0,954 a 1,0) no *merging* de até oito ontologias (*Exp3*), o que representa uma grande vantagem amostral da abordagem proposta, uma vez que oito ontologias podem ser combinadas automaticamente com alta precisão, resguardando as características das ontologias combinadas, o que se constitui em um dos princípios de um bom processo de *merging*.

Ainda na Figura 28, pode-se observar também que, à medida que o número de ontologias aumenta (especialmente a partir do *Exp4*), a qualidade do resultado, independente da estratégia, diminui acentuadamente. Além da quantidade de ontologias, a complexidade delas (quantidade de classes, profundidade) e a qualidade dos alinhamentos usados como entrada para o *merging* múltiplo também influenciam diretamente no resultado final. Pelos experimentos aqui realizados, o uso da estratégia MeMO foi excelente até a combinação de 12 (doze) ontologias (*Exp4*). Acima

disso (*Exp5* e *Exp6*, respectivamente com 16 e 20 ontologias combinadas), já é possível observar uma maior quantidade de resultados inconsistentes para ambas as estratégias.

Apesar de as diferenças nos valores obtidos entre as duas estratégias serem, em alguns casos, aparentemente pequenas, estes valores são bastante significativos, pois uma pequena mudança nos elementos e na estrutura de uma ontologia pode causar impacto em aplicações que as utilizem, propagando equívocos. Por exemplo, em aplicações relacionadas à recuperação de informações baseadas em ontologias, a qualidade da ontologia influencia diretamente no resultado da pesquisa [Strasunskas et al 2008].

Pelo exposto, vê-se que a estratégia MeMO é mais estável e mais confiável que uma estratégia onde a similaridade não é levada em consideração, pois foi melhor em 20 (vinte) cenários dos 24 analisados. Em apenas 04 cenários a estratégia aleatória teve melhor resultado que a MeMO, cuja superioridade se revelou até mesmo na comparação direta com três dos melhores casos da aleatória (*Exp1*, *Exp2* e *Exp4*). Além disso, os melhores casos da estratégia aleatória não são os mais frequentes, de forma que, após muitas execuções os resultados podem ainda ser indesejáveis e muito aquém dos resultados constantes da MeMO. Um exemplo claro pode ser visto no experimento 4 (*Exp4*), onde o pior caso da estratégia aleatória possui o valor de 0,692 ao passo que a estratégia MeMO possui o valor constante de 0,954, o que já representa uma diferença muito considerável. Diante disso, não há como prever até que ponto o melhor dos resultados obtidos por uma estratégia aleatória será realmente bom, enquanto que na estratégia MeMO, o resultado sempre será o mesmo e com qualidade aceitável para reuso na maioria dos experimentos. Adicione a isso o fato de que se a aplicação receber o resultado do *merging* múltiplo como entrada, de maneira automática, não há como verificar se o resultado obtido é ou não satisfatório. Nesse caso, principalmente, se faz necessária uma estratégia que ofereça a garantia de bons resultados constantes.

4.5 Considerações finais

Este capítulo apresentou a validação da estratégia para *merging* múltiplo de ontologias proposta neste trabalho. Foram realizados seis experimentos onde, em cada experimento, foi gerada uma ontologia modelo, uma ontologia global utilizando a MeMO e dez outras computadas através da utilização de uma estratégia aleatória. Para cada ontologia global computada foram calculados os valores das medidas de qualidade (*Precision*, *Recall* e *F-measure*) obtidas através de sua comparação com a ontologia modelo. Como em cada experimento foram geradas dez ontologias aleatórias, uma média foi calculada com os valores dessas medidas. Com isso, foi realizada uma

análise comparativa entre os valores obtidos pela MeMO e os valores obtidos com a estratégia aleatória. Observou-se que a estratégia MeMO foi melhor na maioria dos resultados. Em determinados casos, a estratégia aleatória gerou ontologias mais próximas da ideal, porém, foram resultados esporádicos e sem garantias. Ambas as estratégias tiveram seus resultados bastante comprometidos a partir da combinação de 16 ontologias, tendo a estratégia MeMO obtido resultados mais satisfatórios.

Com a MeMO validada, os próximos passos estão relacionados a sua melhoria, buscando aprimorar a própria codificação do protótipo, os algoritmos e os métodos de *clustering*, além do próprio cálculo de similaridade entre ontologias. Espera-se que, com o aprimoramento da estratégia, esta seja suficiente no futuro para atender as necessidades quando o caso for combinação de múltiplas ontologias de forma automática.

CAPÍTULO 5

Conclusões

Esta dissertação apresentou a estratégia MeMO, uma solução importante para o *merging* de múltiplas ontologias baseada em técnica de *clustering* onde a análise prévia de similaridade é realizada para melhorar a qualidade do resultado final. Apesar de uma solução aleatória para *merging* múltiplo poder prover, em determinados e imprevisíveis momentos, um resultado melhor do que o obtido ao se utilizar a similaridade para uma análise inicial, nada garante que esse resultado será obtido e nem quando. O processo de *merging* múltiplo é um tópico novo no que se refere à ontologias, pois a maioria das soluções propostas até o momento são voltadas para as aplicações onde se faz necessária apenas a combinação de duas ontologias, não suprimindo os casos onde é preciso combinar diversas ontologias de forma automática. Nos casos de aplicações automáticas e na ausência de especialistas do domínio das ontologias, a estratégia aleatória é inviável, ao passo que a estratégia MeMO pode ser uma alternativa razoável. Esse capítulo apresenta as principais contribuições do trabalho, as propostas para trabalhos futuros e as considerações finais.

5.1 Contribuições

Este trabalho teve como principal objetivo propor uma estratégia para *merging* múltiplo de ontologias buscando garantir um resultado final mais confiável. Para atingir esse objetivo principal, outras atividades foram realizadas, gerando as seguintes contribuições:

- Definição de uma estratégia para *Merging* Múltiplo de Ontologias baseada em *Clustering* onde são analisadas as similaridades entre as ontologias e o processo de *merging* ocorre de forma progressiva;
- Definição de uma estratégia para junção de dois alinhamentos, o que evita a chamada de um processo de *matching* e a reutilização de alinhamentos considerados confiáveis. Essa estratégia é importante no processo de *merging* múltiplo automático;

- Construção de uma ferramenta para *merging* de ontologias onde é possível (i) combinar duas ontologias de forma automática, (ii) combinar múltiplas ontologias de forma automática utilizando a estratégia proposta neste trabalho ou um *merging* múltiplo aleatório, (iii) realizar a junção de dois alinhamentos, (iv) visualizar as ontologias, (v) visualizar os alinhamentos, (vi) visualizar a árvore binária resultante do *merging* e (vii) analisar a qualidade dos resultados através de uma ontologia referência (*gold standard*). Todos os processos envolvendo combinação de ontologias requerem como entrada os alinhamentos entre as ontologias;

Todas as ontologias utilizadas nesse trabalho juntamente com seus alinhamentos e ontologias referência (*gold standard*) criadas para teste, além da própria ferramenta, estão disponíveis para download em <http://www.lia.ufc.br/~fabiana/memo.html>.

5.2 Publicações

Durante os estudos e desenvolvimento do trabalho proposto foram conseguidos alguns resultados. Alguns desses resultados já foram publicados e outros estão em análise. Esses trabalhos foram:

- Araujo, F.F. ; Pinheiro, A. M. A. ; Farias, K. M. ; Loscio, B. F. ; Oliveira, D. “FlagellLink A Decision Support System for Distributed Flagellar Data using Data Warehouse”. In: The 23rd Annual ACM Symposium on Applied Computing (SAC) - Bioinformatics Track, 2008. (Publicado)
- Araujo, F. F. ; Loscio, B. F. “MeMO: Uma Estratégia Baseada em Clustering para Merging Múltiplo de Ontologias”. In: Workshop de Teses de Dissertações de Banco de Dados, 2008, Campinas. Workshop de Teses de Dissertações de Banco de Dados, 2008. (Publicado)
- Araujo, F.F.; Lopes, F.L.R.; Oliveira, D.M.; Loscio, B. F. “MeMO: Uma Estratégia baseada em *Clustering* para *Merging* Múltiplo de Ontologias”. In: XXIV Simpósio Brasileiro de Banco de Dados, 2009 (Submetido)
- Araujo, F.F.; Oliveira, H.R; Lopes, F.L.R.; Cunha, D.R.E; Oliveira, D.M.; Loscio, B. F. “MeMOTool: Uma ferramenta para *merging* múltiplo de ontologias”. In: XXII Simpósio

5.3 Trabalhos Futuros

O desenvolvimento desta dissertação deixou caminho aberto para a realização de outros trabalhos, os quais estão relacionados, principalmente, à melhoria dos resultados obtidos. Além disso, alguns trabalhos futuros também contemplam a realização de novos testes e inclusão de novas funções no protótipo desenvolvido. De uma maneira geral os trabalhos futuros podem ser classificados como se segue:

- Realização de melhorias na estratégia MeMO
 - Junção de alinhamentos: melhorar o processo de *Junção de Alinhamentos* no sentido de tornar o algoritmo mais eficiente, conseqüentemente melhorando a qualidade dos alinhamentos resultantes;
 - Processo de *merging*: incrementar o processo de *merging* entre ontologias e, em seguida, analisar o impacto destas melhorias na abordagem de *merging* múltiplo.;
 - Técnica de *cluster*: o alinhamento progressivo ou hierárquico de seqüências biológicas [Feng e Doolittle, 1987] é a *heurística* mais usada, no *entanto*, não garante o ótimo. Outras soluções vêm sendo propostas para melhorar a qualidade do resultado final como o método iterativo e o semi-progressivo. Como exemplo de ferramentas que utilizam essas soluções são o MUSCLE [Edgar, 2004] e a PSAlign [Sze et al 2006], respectivamente. Como este trabalho já verificou que a estratégia é válida, o próximo passo é melhorar a solução. Assim, pretende-se utilizar uma dessas duas novas soluções (iterativo ou semi-progressivo) para *merging* múltiplo de ontologias para ajudar a melhorar a qualidade do resultado final;
 - Cálculo de similaridade: testar novas formas para calcular a similaridade entre as ontologias como os métodos de distância usados na técnica de *cluster* (ex: single link, complete link, average link, Ward, centróide, etc);
 - Manipulação de alinhamentos e ontologias: possibilitar que o usuário possa alterar os

alinhamentos e ontologias diretamente pela ferramenta MeMOTOol;

- Novas funções para o *MeMOTOol*
 - Geração de alinhamentos: adicionar à ferramenta a possibilidade de o usuário realizar o processo de *matching* semi-automático pela própria ferramenta, podendo alterar o resultado de tal forma a deixar o alinhamento ideal;
 - *Merging* de restrições: implementar o *merging* de restrições de ontologias;
 - MeMO API: gerar uma API para que a MeMO possa ser utilizada em qualquer aplicação;
 - *MeMOProtePlug*: adicionar a estratégia MeMO ao *plugin* do Protégé¹², o PROMPT¹³, para disponibilizar o *merging* múltiplo de ontologias aos usuários que utilizam diretamente essa ferramenta de construção de ontologias;
- Realização de novos experimentos utilizando o *MeMOTOol*
 - Aplicar testes com ontologias de diferentes domínios e de complexidades maiores;
 - Aplicar testes para definir a quantidade máxima de ontologias recomendada para a utilização da estratégia;
 - Aplicar testes para definir a complexidade das ontologias (profundidade, quantidade de elementos, etc) recomendada para a utilização da estratégia.

Esses novos experimentos que poderão ser realizados não foram possíveis durante o desenvolvimento deste trabalho devido ao tempo e por não fazerem parte dos objetivos, ou seja, um dos objetivos do trabalho consistia na validação da estratégia e não na sua caracterização. Essa caracterização significa: definir número máximo de ontologias para que a estratégia forneça resultados satisfatórios; definir qual a complexidade da ontologia aceita pela estratégia para dar resultados satisfatórios; identificar o tempo de processamento para cada tipo de ontologia; dentre outras

¹² <http://protege.stanford.edu/>

¹³ <http://protege.cim3.net/cgi-bin/wiki.pl?Prompt>

caracterizações necessárias para que os usuários possam avaliar os benefícios em suas aplicações.

Além de trabalhos relacionados às correções, melhorias, adição de funções e novos testes para caracterizar a estratégia MeMO um outro trabalho importante que deverá ser feito diz respeito a aplicação da estratégia em algum sistema para efetivamente comprovar sua necessidade. Para isso, a estratégia MeMO será utilizada no sistema FlagelLink [Araujo, et al 2008], a fim de realizar as seguintes tarefas: (i) geração de uma ontologia global (*BioDBOntology*) através da combinação de diferentes ontologias que representam os domínios de vários bancos de dados biológicos públicos, onde essa ontologia global será utilizada posteriormente para combinar os próprios dados dos bancos, possibilitando a materialização sem redundância desses dados; e (ii) combinação de diferentes bio-ontologias para gerar uma ontologia de domínio flagelar (OntoFlag), a qual será utilizada posteriormente para o enriquecimento de consultas no banco de dados local.

5.4 Considerações Finais

A solução proposta, estratégia MeMO, não substituirá as soluções existentes e sim suprirá as aplicações onde as soluções atuais não são suficientes. Pode-se dizer que a MeMO e as demais soluções não são concorrentes, mas complementares.

A estratégia MeMO é uma solução importante para o *merging* de múltiplas ontologias garantindo sempre um resultado mais confiável em comparação com uma estratégia de *merging* múltiplo aleatória. Apesar de uma solução aleatória para *merging* múltiplo poder ter, em determinados momentos, um resultado melhor do que se for considerada a similaridade para uma análise inicial, nada garante que esse resultado será obtido. Os testes mostraram que, em uma estratégia aleatória, os piores resultados ocorrem com mais frequência. Nos casos de aplicações automáticas a estratégia aleatória é inviável, no entanto, quando é possível ter uma interação humana onde o usuário pode repetir o processo por várias vezes até encontrar o resultado desejado a estratégia aleatória pode ser considerada viável.

Validada a estratégia MeMO, os próximos passos envolvem o aperfeiçoamento da estratégia no sentido de melhorar a qualidade dos resultados. Para tanto, devem ser realizados novos testes com ontologias de diferentes domínios, aplicando métodos para definir a quantidade máxima de ontologias recomendadas para a utilização da estratégia. Outrossim, devem-se aplicar testes para definir a complexidade recomendada das ontologias (profundidade, quantidade de elementos, etc.)

previamente à sua utilização na estratégia. Além dos testes, também serão realizadas melhorias na ferramenta (disponível em <http://www.lia.ufc.br/~fabiana/memo.html>) para adição de novas funções.

Espera-se que esse trabalho seja um primeiro passo no intuito de se conseguir um processo para *merging* múltiplo completamente automático, iniciando desde o alinhamento até a aquisição de uma ontologia global, onde esse resultado seja confiável a ponto de poder ser dado como entrada para qualquer outra aplicação sem a necessidade da interferência humana durante a execução do processo.

REFERÊNCIAS

- Aleksovski, Z.; Hage, W.V. e Isaac, A. (2007) “A Survey and Categorization of Ontology-Matching Cases”. In: Proceedings of the Workshop on Ontology Matching.
- Araujo, F.F. ; Pinheiro, A. M. A. ; Farias, K. M. ; Loscio, B. F. e Oliveira, D.M. (2008). “FlagellLink A Decision Support System for Distributed Flagellar Data using Data Warehouse”. In: Proceedings of The 23rd Annual ACM Symposium on Applied Computing.
- Anderberg, M. R. (1993). “Cluster analysis for applications”. ISBN: 978-0120576500.
- Aumueller, D.; Do, H.H.; Massmann, S. e Rahm, E. (2005) “Schema and ontology matching with COMA++”. In: SIGMOD Conference.
- Bisson, G. (1992) “Learning in FOL with a similarity measure”. In: Proceedings of AAAI-1992.
- Brank, J.; Mladenic, D. e Grobelnik, M. (2006) “Gold standard based ontology evaluation using instance assignment”. In: Proc. of the EON 2006 Workshop.
- Breitman, K. K. ; Felicissimo, C. H. e Casanova, M. A. (2005). “CATO - A Lightweight Ontology Alignment Tool”. In: 17th Conference on Advanced Information Systems Engineering.
- Brewster, C.; Alani, H.; Dasmahapatra, S. e Wilks, Y. (2004) “Data driven ontology evaluation”. In: Proceedings of Int. Conf. on Language Resources and Evaluation.
- Castano, S.; Ferrara, A. e Montanelli, S. (2003) “H-MATCH: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems”. In: SWDB 2003.
- Castano, S.; Ferrara, A.; Lorusso, D. e Montanelli, S. (2007) “The HMatch 2.0 Suite for Ontology Matchmaking”. In: SWAP 2007
- Cross, V. e Pal, A. (2006) “OntoCAT: An Ontology Consumer Analysis Tool and Its Use on Product Services Categorization Standards”. In: SEBIZ 2006
- Cross, V. e Pal, A. (2008) “An Ontology Analysis Tool”. In: International Journal of General Systems.
- Davies, J.; Studer, R. e Warren, P. (2006) "Semantic Web Technologies: Trends and Research in Ontology-based Systems". ISBN: 978-0470025963
- Dellschaft, K. e Staab, S. (2006) "On how to perform a gold standard based evaluation of ontology learning". In: Proceedings of ISWC-2006 International Semantic Web Conference.
- Do, H. e Rahm, E. (2002) “COMA- A System for Flexible Combination of Schema Matching Approaches”. In: VLDB'02
- Doan, A.; Domingos, P. e Halevy, A. (2001) “Reconciling Schemas of Disparate Data Sources: A

- Machine-Learning Approach”. In: SIGMOD Conference.
- Dou, D.; McDermott, D. e Qi., P. (2005) “Ontology translation on the semantic web”. In: Journal on Data Semantics.
- Euzenat, J. e Valtchev, P. (2003). “An integrative proximity measure for ontology alignment”. In: Proceedings of Semantic Integration workshop at ISWC.
- Euzenat, J. (2004) “An API for Ontology Alignment”. In: International Semantic Web Conference.
- Shvaiko, P. e Euzenat, J. (2005). “Survey of Schema-based Matching Approaches”. In: Journal on Data Semantics.
- Euzenat, J. e Shvaiko, P. (2007). “Ontology Matching”. ISBN:3540496114
- Euzenat, J. (2008) “Algebras of Ontology Alignment Relations”. In: International Semantic Web Conference.
- Everitt, B. (1980). “Cluster analysis”. ISBN: 978-0470220436.
- Feng, D.L e Doolittle, R. F. (1987). “Progressive sequence alignment as a prerequisite to correct phylogenetic trees”. Journal of Molecular Evolution.
- Johnson, R. A. e Wichern, D. W. (1992). “Applied multivariate statistical analysis”. ISBN-13: 978-0130925534.
- Jain, A. K.; Murty, M. N.; e Flynn, P. J. (1999). “Data clustering: a review”. In: ACM Computing Surveys.
- Gomez-Perez, A., (1999). “Evaluation of Taxonomic Knowledge on Ontologies and Knowledge-Based Systems”. In: Workshop on Knowledge Acquisition, Modeling and Management
- Goh, C.H. (1997) “Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources”. Phd, MIT, 1997.
- Ganter, B. e Wille, R. (1999) “Formal Concept Analysis: mathematical foundations”. ISBN: 978-3540627715.
- Gruber, T.R. (1993) “A translation approach to portable ontology specifications, Knowledge Acquisition”. In: Knowledge Acquisition.
- Hamerly, G. e Elkan, C. (2002) “Alternatives to the k-means algorithm that find better clusterings”. In: CIKM 2002.
- Hjelm, H. e Buitelaar, P. (2008) “Multilingual Evidence Improves Clustering-based Taxonomy Extraction”. In: ECAI 2008.
- Kaufman, L. e Rousseeuw, P. J. (1990). “Finding groups in data: an introduction to cluster analysis”. ISBN: 978-0471878766.
- Kotis, K.; Vouros, G. e Stergiou, K. (2006). “Towards automatic merging of domain ontologies: The

- HCONE-merge approach". *Journal of Web Semantics*.
- Kotis, K.; Vouros, G. e Valarakos, A. (2006) "AUTOMS: Automated Ontology Mapping through Synthesis of Methods". In: *Ontology Matching 2006*
- Lee, M.L.; Yang, L.H.; Hsu, W. e Yang, X. (2002) "XClust: clustering XML schemas for effective integration". In: *Proceedings of the 11th international conference on Information and knowledge management*.
- Levenstein, I. (1966) "Binary codes capable of correcting deletions, insertions and reversals". In: *Cybernetics and Control Theory*.
- Maedche, A., e Staab, S. (2002) "Measuring similarity between ontologies". In: *Proceedings of 13th CIKM*.
- Metz, J. e Monard, M. C. (2008). "Interpretação de Clusters Gerados por Algoritmos de Clustering Hierárquico". In: *Workshop de Dissertações e Teses Concluídas do iCMC-USP*.
- Mrozek, A. e Cyran, K. (2001). "Rough Set in Hybrid Methods for Pattern Recognition". In: *International Journal of Intelligence Systems*.
- Needleman, S. e Wunsch, C. (1970) "A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of two Proteins". In: *Molecular Biology*.
- Noy, N. e Musen, M. (2001). "Anchor-prompt: using non-local context for semantic matching". In: *Proceedings of the workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence*.
- Noy, N. F. e Musen, M. A (2002). "The PROMPT suite: Interactive tools for ontology merging and mapping". In: *Technical report, SMI*.
- Oliveira D.M.; Gouveia, J.J.; Diniz, N.B.; Pacheco, A.C.; Vasconcelos, E.J.; Diniz, M.C.; Viana, D.A.; Ferreira, T.D.; Albuquerque, M.C.; Fortier, D.C.; Maia, A.R.; Costa, L.A.; Melo, J.O.; Silva, M.C.; Walter, C.A.; Faria, J.O.; Tome, A.R.; Gomes, M.J.; Oliveira, S.M.; Araujo-Filho, R.; Costa, R.B. e Maggioni, R. (2005) "Pathogenomics analysis of *Leishmania* spp.: flagellar gene families of putative virulence factors". In: *OMICS*.
- Pérez-Rey, D.; Maojo, V.; García-Remesal, M.; Alonso-Calvo, R.; Billhardt, H.; Martín-Sánchez, F. e Sousa, A. (2006) "ONTOFUSION: Ontology-based integration of genomic and clinical databases". In: *Comput Biol Med*.
- Pottinger, R.A. e Bernstein, P.A. (2003) "Merging Models Based on Given Correspondences". In: *Proceedings of the 29th VLDB Conference*.
- Rasgado, A.D.C. e Guzman, A.A. (2006) "A Language and Algorithm for Automatic Merging of Ontologies," In: *15th International Conference on Computing*.
- Sabou, M.; Lopez, V.; Motta, E. e Uren, V. (2006) "Ontology Selection: Evaluation on the Real Semantic Web". In: *4th International Evaluation of Ontologies for the Web Workshop*.

- Sander, J.; Qin, X.; Lu, Z.; Niu, N. e Kovarsky, A. (2003). "Automatic extraction of clusters from hierarchical clustering representations". In: Pacific-Asia Knowledge Discovery and Data Mining.
- Scheinerman, E.R. (2006) "Matemática Discreta - Uma Introdução". ISBN: 8522102910.
- Smith, T.F. E Waterman, M.S. (1981) "Identification of common molecular subsequences". In: Journal of Molecular Biology.
- Staab, S. e Studer, R. (2004) "Handbook on Ontologies in Information Systems". ISBN-13: 978-3540408345.
- Stoilos, G.; Stamou, G. e Kollias, S. (2005) "A String Metric For Ontology Alignment". In: 4th International Semantic Web Conference.
- Su, X. e Gulla, J.A. (2006) "An information retrieval approach to ontology mapping". In: Data Knowledge.
- Stumme, G. e Mädche, A. (2001) "FCA-Merge: bottom-up merging of ontologies". In: Proceedings of the 17th International Joint Conference on Artificial Intelligence.
- Su, H.; Padmanabhan, S. e Lo, M. (2001) "Identification of Syntactically Similar DTD Elements in Schema Matching across DTDs". In: WAIM.
- Tartir S.; Arpinar I.B.; Moore M.; Sheth A. P. e Aleman-Meza B. (2005) "OntoQA: Metric-Based Ontology Quality Analysis". In: Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources.
- Theodoridis, S. e Koutroumbas, K. (2003) "Pattern recognition". ISBN: 0123695317.
- Thompson, J.D.; Higgins, D.G.; Gibson, T.J. (1994). "Clustal W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice". In: Nucleic Acids Resource.
- Udrea, O.; Miller, R.J. e Getoor, L. (2007) "Leveraging data and structure in ontology integration". In: Proceedings of SIGMOD 2007
- Heijst, V.; Schreiber, A. e Wielinga, B.J. (1997) "Using explicit ontologies in KBS development". In: International Journal of Human-Computer Studies.
- Zimmermann, A. e Euzenat, J. (2006) "Three semantics for distributed systems and their relations with alignment composition". in: Proceedings of 5th conference on International semantic web conference.
- Weinstein, P. e Birmingham, W. (1999) "Comparing concepts in differentiated ontologies". In: Proceedings of KAW-99.
- Wille, R. (1982) "Restructuring lattice theory: an approach based on hierarchies of concepts". In: R.I. (Ed.), Ordered Sets, Reidel.

APÊNDICE


```

<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_i.owl#Professor' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_j.owl#Professor' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_i.owl#ProfessorDE' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_j.owl#FullProfessor' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.75</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_i.owl#Faculty' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_j.owl#Faculty' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
  </Cell>
</map>
</Alignment>
</rdf:RDF>

```

A.2 Alinhamento entre uma ontologia O1 e O2

```

<?xml version='1.0' encoding='utf-8' standalone='no'?>
<rdf:RDF xmlns='http://knowledgeweb.semanticweb.org/heterogeneity/alignment#'
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:xsd='http://www.w3.org/2001/XMLSchema#'
  xmlns:align='http://knowledgeweb.semanticweb.org/heterogeneity/alignment#'>
<Alignment>
  <xml>yes</xml>
  <level>0</level>
  <type>**</type>
  <method>fr.inrialpes.exmo.align.impl.method.StrucSubsDistAlignment</method>
  <time>29</time>
  <onto1>
    <Ontology rdf:about="http://www.owl-ontologies.com/Person_i.owl">
      <location>file:/home/ontologies/Person_i.owl</location>
      <formalism>
        <Formalism align:name="OWL1.0" align:uri="http://www.w3.org/2002/07/owl#" />
      </formalism>
    </Ontology>
  </onto1>
  <onto2>
    <Ontology rdf:about="http://www.owl-ontologies.com/Person_u.owl">
      <location>file:/home/ontologies/Person_u.owl</location>
      <formalism>
        <Formalism align:name="OWL1.0" align:uri="http://www.w3.org/2002/07/owl#" />
      </formalism>
    </Ontology>
  </onto2>
  <map>
    <Cell>
      <entity1 rdf:resource='http://www.owl-ontologies.com/Person_i.owl#Person' />
      <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#Person' />
      <relation>=</relation>
      <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.8</measure>
    </Cell>
  </map>
  <map>
    <Cell>
      <entity1 rdf:resource='http://www.owl-ontologies.com/Person_i.owl#Student' />
      <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#Student' />
      <relation>=</relation>
      <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
    </Cell>
  </map>

```

```

<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_i.owl#BSStudent' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#BS' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.8</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_i.owl#MSStudent' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#MS' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.77</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_i.owl#PhDStudent' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#PhD' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.87</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_i.owl#Professor' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#Professor' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_i.owl#ProfessorDE' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#ProfessorDE' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_i.owl#Faculty' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#Faculty' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
  </Cell>
</map>
</Alignment>
</rdf:RDF>

```

A.3 Alinhamento entre uma ontologia O3 e O2

```

<?xml version='1.0' encoding='utf-8' standalone='no'?>
<rdf:RDF xmlns='http://knowledgeweb.semanticweb.org/heterogeneity/alignment#'
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:xsd='http://www.w3.org/2001/XMLSchema#'
  xmlns:align='http://knowledgeweb.semanticweb.org/heterogeneity/alignment#'>
<Alignment>
  <xml>yes</xml>
  <level>0</level>
  <type>**</type>
  <method>fr.inrialpes.exmo.align.impl.method.StrucSubsDistAlignment</method>
  <time>29</time>
  <ontology>
    <Ontology rdf:about="http://www.owl-ontologies.com/Person_j.owl">
      <location>file:/home/ontologies/Person_j.owl</location>
      <formalism>
        <Formalism align:name="OWL1.0" align:uri="http://www.w3.org/2002/07/owl#" />
      </formalism>
    </Ontology>

```

```

</ontol>
<onto2>
  <Ontology rdf:about="http://www.owl-ontologies.com/Person_u.owl">
    <location>file:/home/ontologies/Person_u.owl</location>
    <formalism>
      <Formalism align:name="OWL1.0" align:uri="http://www.w3.org/2002/07/owl#" />
    </formalism>
  </Ontology>
</onto2>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_j.owl#Person' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#Person' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.8</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_j.owl#Student' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#Student' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_j.owl#UndergraduateStudent' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#BS' />
    <relation>'>'</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.87</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_j.owl#GraduateStudent' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#MS' />
    <relation>'>'</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.76</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_j.owl#GraduateStudent' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#PhD' />
    <relation>'>'</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.788</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_j.owl#Professor' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#Professor' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_j.owl#FullProfessor' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#ProfessorDE' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.7777</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_j.owl#Faculty' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#Faculty' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
  </Cell>
</map>
</Alignment>
</rdf:RDF>

```

A.4 Alinhamento entre uma ontologia O5 e O2

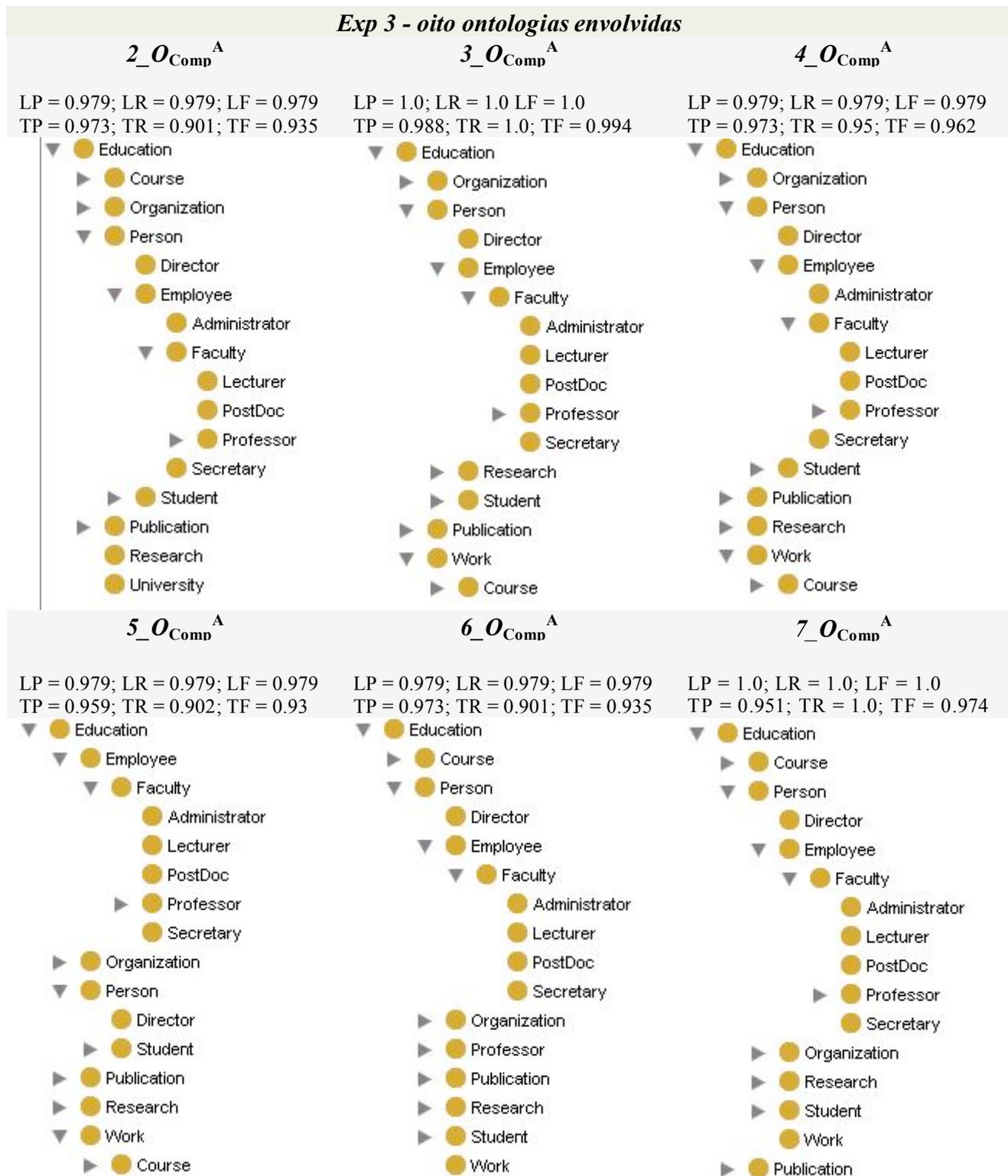
```
<?xml version='1.0' encoding='utf-8' standalone='no'?>
<rdf:RDF xmlns='http://knowledgeweb.semanticweb.org/heterogeneity/alignment#'
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:xsd='http://www.w3.org/2001/XMLSchema#'
  xmlns:align='http://knowledgeweb.semanticweb.org/heterogeneity/alignment# '>
<Alignment>
  <xml>yes</xml>
  <level>0</level>
  <type>**</type>
  <method>fr.inrialpes.exmo.align.impl.method.StrucSubsDistAlignment</method>
  <time>29</time>
  <onto1>
    <Ontology rdf:about="http://www.owl-ontologies.com/Person_c.owl">
      <location>file:/home/ontologies/Person_c.owl</location>
      <formalism>
        <Formalism align:name="OWL1.0" align:uri="http://www.w3.org/2002/07/owl#" />
      </formalism>
    </Ontology>
  </onto1>
  <onto2>
    <Ontology rdf:about="http://www.owl-ontologies.com/Person_u.owl">
      <location>file:/home/ontologies/Person_u.owl</location>
      <formalism>
        <Formalism align:name="OWL1.0" align:uri="http://www.w3.org/2002/07/owl#" />
      </formalism>
    </Ontology>
  </onto2>
  <map>
    <Cell>
      <entity1 rdf:resource='http://www.owl-ontologies.com/Person_c.owl#Person' />
      <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#Person' />
      <relation>=</relation>
      <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.8</measure>
    </Cell>
  </map>
  <map>
    <Cell>
      <entity1 rdf:resource='http://www.owl-ontologies.com/Person_c.owl#Student' />
      <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#Student' />
      <relation>=</relation>
      <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
    </Cell>
  </map>
  <map>
    <Cell>
      <entity1 rdf:resource='http://www.owl-ontologies.com/Person_c.owl#UndergraduateStudent' />
      <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#BS' />
      <relation>'>'</relation>
      <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.87</measure>
    </Cell>
  </map>
  <map>
    <Cell>
      <entity1 rdf:resource='http://www.owl-ontologies.com/Person_c.owl#BSSStudent' />
      <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#BS' />
      <relation>=</relation>
      <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.8</measure>
    </Cell>
  </map>
  <map>
    <Cell>
      <entity1 rdf:resource='http://www.owl-ontologies.com/Person_c.owl#GraduateStudent' />
      <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#MS' />
      <relation>'>'</relation>
      <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.76</measure>
    </Cell>
  </map>
</map>
```

```

<Cell>
  <entity1 rdf:resource='http://www.owl-ontologies.com/Person_c.owl#MSStudent' />
  <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#MS' />
  <relation>=</relation>
  <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.77</measure>
</Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_c.owl#GraduateStudent' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#PhD' />
    <relation>'>'</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.788</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_c.owl#PhDStudent' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#PhD' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.87</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_c.owl#Professor' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#Professor' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_c.owl#FullProfessor' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#ProfessorDE' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.777</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_c.owl#ProfessorDE' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#ProfessorDE' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.owl-ontologies.com/Person_c.owl#Faculty' />
    <entity2 rdf:resource='http://www.owl-ontologies.com/Person_u.owl#Faculty' />
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
  </Cell>
</map>
</Alignment>
</rdf:RDF>

```

APÊNDICE B – Ontologias Resultantes de um Merging Múltiplo



Exp 4 - doze ontologias envolvidas

2_ O_{Comp}^A

3_ O_{Comp}^A

4_ O_{Comp}^A

LP = 0.902; LR = 0.973; LF = 0.936
TP = 0.706; TR = 0.639; TF = 0.671

LP = 0.914; LR = 0.986; LF = 0.937
TP = 0.7; TR = 0.758; TF = 0.728

LP = 0.902; LR = 0.973; LF = 0.936
TP = 0.671; TR = 0.92; TF = 0.776



Exp 5 - desessei ontologias envolvidas

2_0Comp^A

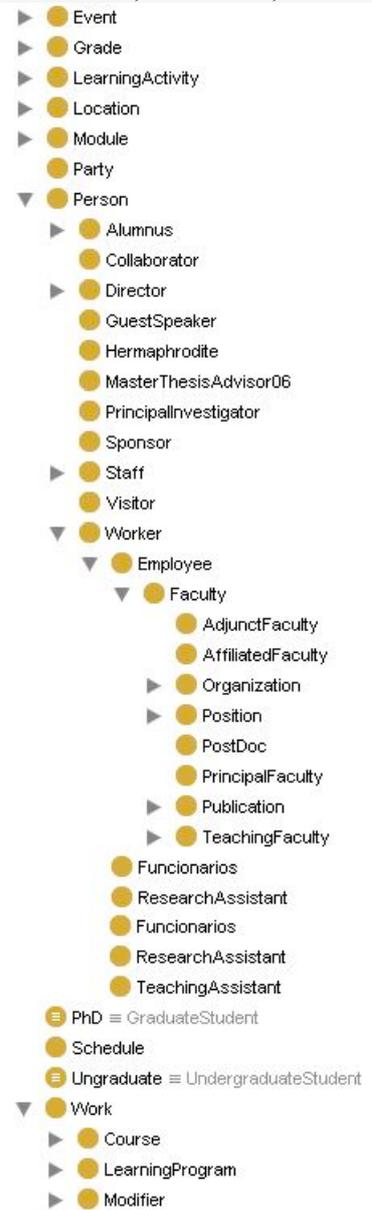
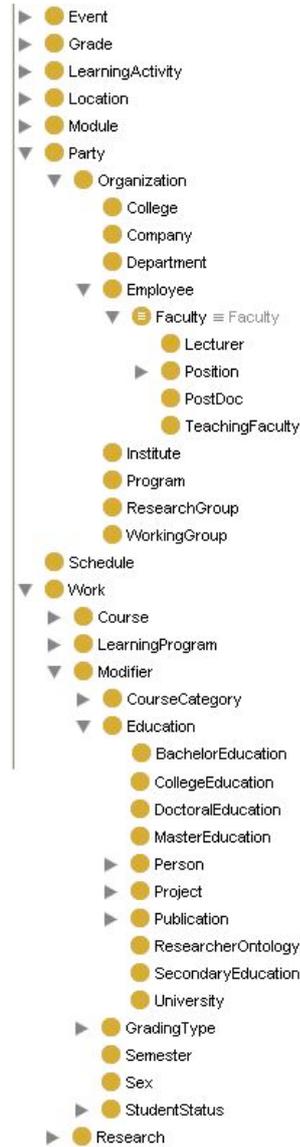
3_0Comp^A

4_0Comp^A

LP = 0.888; LR = 0.953; LF = 0.919
TP = 0.55; TR = 0.74; TF = 0.633

LP = 0.893; LR = 0.946; LF = 0.918
TP = 0.605; TR = 0.746; TF = 0.668

LP = 0.882; LR = 0.953; LF = 0.916
TP = 0.643; TR = 0.699; TF = 0.67



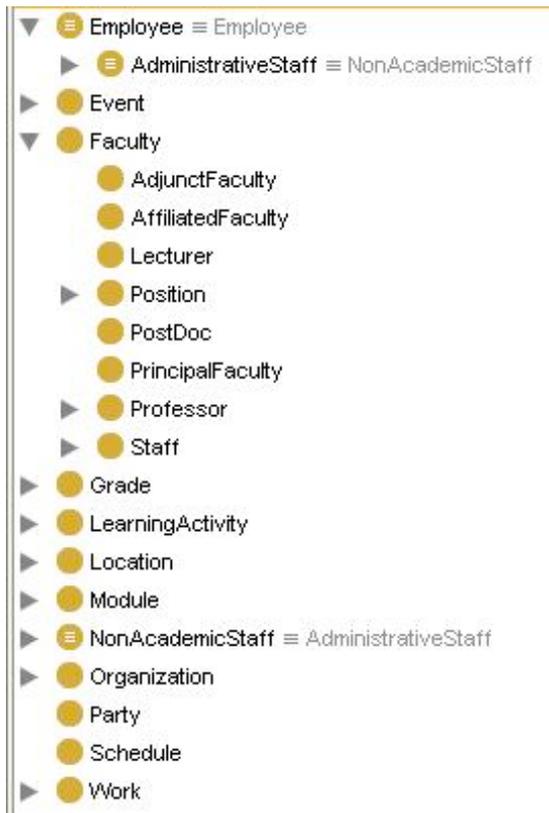
Exp 6 - vinte ontologias envolvidas

$2_{O_{Comp}}^A$

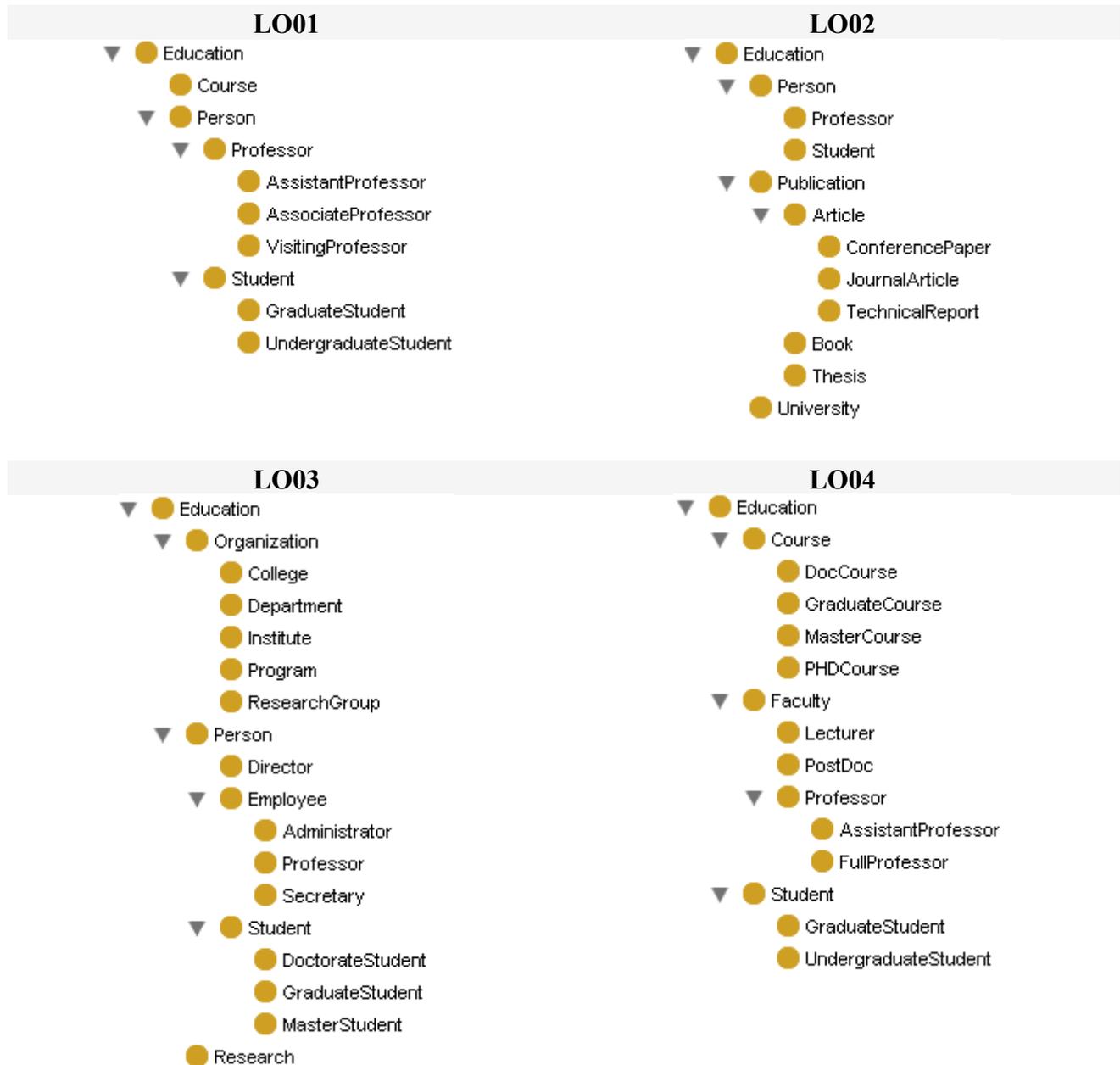
$3_{O_{Comp}}^A$

LP = 0.929; LR = 0.966; LF = 0.947
 TP = 0.659; TR = 0.684; TF = 0.671

LP = 0.903; LR = 0.933; LF = 0.917
 TP = 0.698; TR = 0.673; TF = 0.685



APÊNDICE C – Ontologias Utilizadas nos Experimentos



LO05

- Education
 - Organization
 - Department
 - Institute
 - University
 - Professor
 - Publication
 - Article
 - ConferenceArticle
 - JournalArticle
 - Book
 - Manual
 - Thesis

LO06

- Education
 - Person
 - Faculty
 - Lecturer
 - PostDoc
 - Professor
 - Research
 - AssistantResearch
 - VisitingResearch
 - Student
 - Publication
 - Article
 - Book
 - Thesis
 - DocThesis
 - MasterThesis

LO06

- Education
 - Organization
 - Department
 - ResearchGroup
 - University
 - Person
 - Professor
 - Student
 - GraduateStudent
 - UndergraduateStudent
 - Research
 - AssistantResearch
 - FullResearch
 - Work
 - Course
 - GraduateCourse

LO08

- Education
 - Organization
 - College
 - Program
 - University
 - Publication
 - Article
 - ConferenceArticle
 - JournalArticle
 - Research
 - Book
 - Periodical
 - Proceedings
 - Research

LO09

- ▼ ● Education
 - ▼ ● Course
 - DoctorCourse
 - GraduateCourse
 - MasterCourse
 - PHDCourse
 - GraduateStudent
 - ▼ ● Professor
 - AssistantProfessor
 - AssociateProfessor
 - Dean
 - FullProfessor
 - VisitingProfessor
 - ▼ ● Project
 - Scholarship

LO010

- ▼ ● Education
 - ▼ ● Person
 - ▼ ● Director
 - AdministratorDirector
 - FinancesDirector
 - RHDirector
 - Professor
 - Student
 - ▼ ● Worker
 - ▼ ● Faculty
 - Lecturer
 - PostDoc
 - ▼ ● Work
 - ▼ ● Course
 - GraduateCourse
 - UngraduateCourse
 - ▼ ● Research
 - AssistantResearch
 - FullResearch

LO11

- ▼ ● Person
 - ▼ ● Alumnus
 - BSAlumnus
 - MSAlumnus
 - PhDAlumnus
 - Collaborator
 - ▼ ● Faculty
 - AdjunctFaculty
 - AffiliatedFaculty
 - AssistantProfessor
 - AssociateProfessor
 - PrincipalFaculty
 - Professor
 - GuestSpeaker
 - PrincipalInvestigator
 - Sponsor
 - ▼ ● Student
 - BSStudent
 - MSStudent
 - PhDStudent
 - Visitor

LO12

- ▼ ● ResearcherOntology
 - ▼ ● Organization
 - Company
 - Department
 - Institute
 - University
 - WorkingGroup
 - ▼ ● Person
 - ▼ ● Employee
 - ▼ ● AcademicStaff
 - ▼ ● FacultyMember
 - AssistantProfessor
 - AssociateProfessor
 - FullProfessor
 - Lecturer
 - AdministrativeStaff
 - Manager
 - TechnicalStaff
 - ▼ ● Student
 - ▼ ● Graduate
 - PhDStudent
 - Ungraduate
 - ▼ ● Project
 - DevelopmentProject
 - ResearchProject

LO13

- ▼ ● Event
 - HalfYear
 - Holiday
 - RecurringAppointment
 - StudyYear
- ▼ ● Grade
 - PassFail
 - TenPoint
- ▼ ● LearningActivity
 - Assignment
 - CourseOffering
- ▼ ● LearningProgram
 - ▼ ● CourseProgram
 - ClassroomCourseProgram
 - ThesisProgram
 - LearningItem
 - ModuleProgram
 - StudyProgram
- ▼ ● Location
 - Building
 - Room
- ▼ ● Modifier
 - ▼ ● CourseCategory
 - CategoryA
 - CategoryB
 - CategoryC
 - ▼ ● Education
 - BachelorEducation
 - CollegeEducation
 - DoctoralEducation
 - MasterEducation
 - SecondaryEducation
 - ▼ ● GradingType
 - Grading10
 - GradingPassFail
 - ▼ ● Position
 - ▼ ● AcademicPosition
 - Assistant
 - AssistantProfessor
 - AssociateProfessor
 - Instructor
 - Lector
 - Professor
 - DeanPosition
 - DepartmentHeadPosition
 - SecretaryPosition
 - StudyProgramLeadPosition
 - TemporaryWorkAgreement
 - Semester
 - Sex
 - ▼ ● StudentStatus
 - ▼ ● Active
 - BudgetStudent
 - TuitionStudent
 - Suspended
 - ▼ ● StudyLevel
 - BachelorLevel
 - CollegeLevel
 - DoctoralLevel
 - MastersLevel
- ▼ ● Party
 - ▼ ● Organization
 - Department
 - Faculty
 - University
 - ▼ ● Person
 - Hermaphrodite
 - MasterThesisAdvisor06
 - ▼ ● Staff
 - Administrator
 - FullTimeInstructor
 - PartTimeInstructor
 - Student

LO14

- ▼ ● Organization
 - College
 - Department
 - Institute
 - Program
 - ResearchGroup
 - University
- ▼ ● Person
 - Director
 - ▼ ● Employee
 - ▼ ● AdministrativeStaff
 - ClericalStaff
 - SystemsStaff
 - ▼ ● Faculty
 - Lecturer
 - PostDoc
 - ▼ ● Professor
 - AssistantProfessor
 - AssociateProfessor
 - Dean
 - FullProfessor
 - VisitingProfessor
 - ResearchAssistant
 - ▼ ● Student
 - GraduateStudent
 - UndergraduateStudent
 - TeachingAssistant
 - ▼ ● Publication
 - ▼ ● Article
 - ConferencePaper
 - JournalArticle
 - TechnicalReport
 - Book
 - Manual
 - Software
 - Specification
 - UnofficialPublication
 - Schedule
 - ▼ ● Work
 - ▼ ● Course
 - GraduateCourse
 - Research

LO15

- ▼ ● Module
 - ComputerScienceModule
 - EconomicsModule
 - MathsModule
- ▼ ● Person
 - ▼ ● Staff
 - AcademicStaff
 - NonAcademicStaff
 - ▼ ● Student
 - GraduateStudent
 - UndergraduateStudent

LO16

- ▼ ● Person
 - ▼ ● Faculty
 - ▼ ● TeachingFaculty
 - AssistantProfessor
 - Lecturer
 - ▼ ● Professor
 - ProfessorDE
 - ProfessorInHClorAI
 - Funcionarios
 - ▼ ● Student
 - Certification
 - ▼ ● Degree
 - BS
 - MS
 - PhD

LO17

- ▼ ● Person
 - ▼ ● Alumnus
 - BSAlumnus
 - MSAlumnus
 - PhDAlumnus
 - ▼ ● Faculty
 - AdjunctFaculty
 - AffiliatedFaculty
 - AssistantProfessor
 - AssociateProfessor
 - PrincipalFaculty
 - ▼ ● Professor
 - ProfessorDE
 - ▼ ● Student
 - BSStudent
 - MSStudent
 - PhDStudent

LO18

- ▼ ● Person
 - ▼ ● Employee
 - ▼ ● AcademicStaff
 - ▼ ● FacultyMember
 - AssistantProfessor
 - AssociateProfessor
 - FullProfessor
 - Lecturer
 - AdministrativeStaff
 - ▼ ● Student
 - ▼ ● Graduate
 - PhDStudent
 - Ungraduate
 - ▼ ● Project
 - ResearchProject
 - ▼ ● ResearcherOntology
 - ▼ ● Organization
 - Company
 - Department
 - Institute
 - University
 - WorkingGroup

LO19

LO20

- ▼ ● Organization
 - College
 - Department
- ▼ ● Person
 - Director
 - ResearchAssistant
 - Student
- ▼ ● Publication
 - Article
 - UnofficialPublication
- Schedule
- ▼ ● Work
 - Research

- ▼ ● Person
 - ▼ ● Faculty
 - AdjunctFaculty
 - AffiliatedFaculty
 - Professor
 - PrincipallInvestigator
 - Sponsor
 - Student
 - Visitor