

Similaridade entre Estruturas Ontológicas

Marcirio Silveira Chaves*, Vera Lúcia Strube de Lima

Programa de Pós-Graduação em Ciência da Computação
Faculdade de Informática
Pontifícia Universidade Católica do Rio Grande do Sul
Av. Ipiranga, 6681
90619-900 Porto Alegre, RS

{mchaves, vera}@inf.pucrs.br

Abstract. *The automatic mapping among Ontological Structures has been a continuous concern as a task of integration and reuse of knowledge. This article describes an ongoing work that considers the application of the similarity measure proposed in [Maedche e Staab, 2002] to compare terms in distinct hierarchies. This measure helps the mapping among Ontological Structures allowing to find terms with lexical similarity. We propose two heuristics to deal with the terms in semantic-structural level and develop an interface to help knowledge engineers work.*

Resumo. *O mapeamento automático entre Estruturas Ontológicas (EOs) tem sido uma preocupação constante como uma tarefa de integração e reuso de conhecimento. Este artigo descreve um trabalho em desenvolvimento que aplica, em primeiro lugar, a medida de similaridade proposta em [Maedche e Staab, 2002] para comparar termos em hierarquias distintas. Esta medida auxilia o mapeamento entre EOs na medida que permite encontrar termos lexicalmente similares. Em segundo lugar, propomos duas heurísticas para tratar os termos em nível semântico-estrutural e desenvolvemos uma interface para auxiliar o trabalho do engenheiro do conhecimento.*

1. Introdução

Se duas pessoas desejam comunicar-se, ambas utilizam uma linguagem comum. Entretanto, se uma dessas pessoas desconhece a(s) linguagem(ns) falada(s) pela outra, faz-se necessária a intervenção de um mediador externo que possibilite um vocabulário comum, de forma que a comunicação possa ocorrer com um mínimo de entendimento. Este mesmo princípio pode ser utilizado para estruturas ontológicas.

Entende-se por Estrutura Ontológica (EO) um conjunto de termos previamente definidos, associados de forma explícita por relações semânticas, em formato legível por humanos e por máquinas, aí incluindo-se coleções de vocabulários ou de conceitos.

Atualmente, as pesquisas que envolvem o mapeamento entre EOs ainda prevêem uma grande quantidade de trabalho manual, sendo as propostas mais avançadas

*Financiado pelo projeto de pesquisa HP-CPAD (Centro de Processamento em Alto Desempenho), convênio PUCRS/HP.

[Doan et al., 2002, Noy e Musen, 1999] caracterizadas como semi-automáticas, pois ainda não se conta com técnicas que permitam automatizar completamente esse processo.

Noy e Musen [Noy e Musen, 1999] afirmam que o trabalho de mapear, unir ou alinhar EOs é realizado, na maior parte das vezes, à mão, sem qualquer ferramenta para automatização total ou parcial. Esse mapeamento manual é lento [Uschold, 2001], tedioso e suscetível a erro [Doan et al., 2002, Noy e Musen, 1999, Noy e Musen, 2001]. Noy e Musen [Noy e Musen, 1999] ainda acrescentam que é um processo difícil de repetir e simplesmente não é prático, para certas aplicações, como por exemplo em bibliotecas digitais. Para Doan [Doan et al., 2002] o mapeamento manual não é escalável no contexto da *web*. Uschold [Uschold, 2001] comenta que o mapeamento automático é difícil e constitui um problema de pesquisa desafiador. Conforme Ding e Foo [Ding e Foo, 2002], “a necessidade de intervenção manual nos aspectos de *geração, mapeamento e evolução* de EOs atestam a natureza complexa da pesquisa nesta área e os problemas associados ainda não solucionados”. A afirmação de Ding e Foo [Ding e Foo, 2002] explicita a natureza abstrata dos três aspectos de pesquisa relacionados às EOs, a saber, *geração, mapeamento e evolução*, explicada pelo fato de as EOs serem desenvolvidas para sistemas distintos, por seres humanos com visões de mundo diferentes.

Com o grande número de EOs disponíveis na *web* nos últimos anos, surgiram preocupações como estendê-las, adaptá-las e compará-las. Usuários de EOs frequentemente possuem uma EO-base, na qual navegam ou sobre a qual realizam consulta. Para auxiliar estas tarefas é importante que os usuários tenham conhecimento da similaridade entre a EO-base e as EOs recuperadas em outras aplicações. Neste contexto surge a necessidade do mapeamento entre EOs, que tem sido feito, principalmente, por meio de união e/ou alinhamento.

Para estudar o problema do mapeamento entre EOs, utilizamos medidas de similaridade entre termos em nível lexical, além de duas heurísticas que permitem melhorar o mapeamento gerado automaticamente. Optamos por trabalhar, num primeiro momento, com EOs da língua inglesa, pois ainda não se tem uma quantidade suficiente de EOs de domínios similares na língua portuguesa.

Este artigo está organizado em 5 seções. A seção 2 apresenta os trabalhos correlatos. A tarefa de mapeamento entre EOs e os conceitos de união e alinhamento são apresentados na seção 3. A seção 4 descreve um esforço de aprimoramento do mapeamento automático entre EOs, as medidas de similaridade e as heurísticas utilizadas. Finalmente, na seção 5, são feitas as considerações e incluídas etapas para continuidade do trabalho.

2. Trabalhos Correlatos

Nesta seção são descritos alguns dos trabalhos encontrados na literatura que demonstram preocupação no sentido de automatizar a tarefa de mapeamento entre EOs.

2.1. *Anchor-Prompt*

Noy e Musen [Noy e Musen, 2001] desenvolveram o algoritmo *Anchor-Prompt*, que utiliza como entrada um conjunto de combinações âncoras¹ identificadas previamente (de

¹Pares de termos relacionados.

modo automático ou manual). As EOs utilizadas são provenientes da biblioteca do programa DAML².

Inicialmente, o algoritmo recebe como entrada os termos-âncora. Conhecido o comprimento do caminho entre esses termos, é atribuído um grau à similaridade entre os mesmos. Por exemplo, em uma EO_1 composta dos termos A-B-C-D o comprimento do caminho do nodo A até o nodo D é 3; em uma outra EO_2 formada pelos termos A-M-N-D o comprimento do caminho entre A e D também é 3. Neste caso, os termos-âncora são A e D e o grau de similaridade entre B e M e entre C e N será mais elevado, pois esses termos estão nas mesmas posições relativas no caminho que vai de A até D.

Apesar de prover mapeamentos consistentes, a abordagem baseada em âncoras possui uma forte limitação quando as EOs são de profundidades diferentes, ou seja, quando uma EO é profunda (possuindo muitos níveis na hierarquia) e a outra EO é rasa (contendo poucos níveis na hierarquia). Neste caso, Noy e Musen [Noy e Musen, 2000] afirmam que o algoritmo não funciona bem.

O presente trabalho não utiliza a abordagem de termos-âncora, pelos motivos já identificados em [Noy e Musen, 2001]. Não raras vezes temos encontrado EOs com profundidades distintas e, conseqüentemente, optamos por utilizar uma abordagem que não leva em consideração somente a profundidade das hierarquias.

2.2. Heurística e Probabilidade

Prasad, Peng e Finin [Prasad et al., 2002] desenvolveram um mapeador de EOs que apresenta as EOs ao usuário, e este último indica os pontos (conceitos) equivalentes para mapeamento. Esses pontos são funcionalmente similares aos termos-âncora apresentados em [Noy e Musen, 2001].

Após a intervenção inicial do usuário, o mapeamento automático pode ser realizado por meio de duas abordagens: heurística e de probabilidade. A abordagem heurística considera a porcentagem de filhos de um conceito em uma hierarquia A que podem ser mapeados para outro conceito em uma hierarquia B. Por exemplo, se o conceito C_A possui dez filhos e seis desses filhos ([Prasad et al., 2002] operam sobre uma taxa mínima de similaridade de 60%) são similares a um conceito C_B , pode-se concluir que C_A é similar a C_B . Já a abordagem de probabilidade é baseada nos resultados obtidos por um classificador de documentos que permite a identificação da similaridade entre conceitos.

Este trabalho utilizou somente EOs no formato DAML-OIL³ e não considerou as propriedades relacionadas a cada conceito. O algoritmo de probabilidade que apresentou, segundo os autores, melhores resultados é fortemente dependente do conjunto de documentos disponíveis.

O presente trabalho não utiliza documentos que instanciem conceitos, mas faz uso de heurísticas próprias para realizar o mapeamento entre conceitos similares.

²DARPA Agent Markup Language - <http://www.daml.org/ontologies/>

³DARPA Agent Markup Language - *Ontology Interchange Language ou Ontology Inference Layer* - <http://www.w3.org/TR/daml+oil-reference>

2.3. Medidas de Similaridade

Em [Maedche e Staab, 2002] é apresentada uma visão em duas camadas (lexical e conceitual) para medir a similaridade entre EOs.

Em nível lexical, os autores basearam-se na Distância de Edição (DE) de Levenshtein [Levenshtein, 1966]. Essa medida considera o número de mudanças que devem ocorrer para transformar uma cadeia de caracteres em outra. Por exemplo, a $DE(\text{automóvel}, \text{automóveis})$ é 2, pois uma operação de alteração e uma operação de inserção mudam a cadeia de caracteres `automóvel` para `automóveis`. A contribuição de Maedche e Staab consiste na medida de Combinação de Caracteres (CC), dada pela seguinte fórmula:

$$CC(T_i, T_j) := \max \left(0, \frac{\min(|T_i|, |T_j|) - DE(T_i, T_j)}{\min(|T_i|, |T_j|)} \right) \in [0, 1]. \quad (1)$$

A medida CC calcula a similaridade entre dois termos T_i e T_j . O comprimento do termo de menor comprimento é representado pelo $\min(|T_i|, |T_j|)$. Por exemplo, ao processar a similaridade dos termos (`automóvel`, `automóveis`) o menor comprimento é 9 e o valor da $DE(T_i, T_j)$ é 2. Logo, tem-se os seguintes valores para $CC(\text{automóvel}, \text{automóveis})$:

$$CC(\text{automóvel}, \text{automóveis}) := \max \left(0, \frac{9 - 2}{9} \right) = \max \left(0, \frac{7}{9} \right) \in [0, 1].$$

O menor comprimento é considerado tanto no numerador quanto no denominador da fórmula, o que permite ponderar o número de alterações obtidas com a DE em relação ao termo com menor comprimento. No exemplo, tem-se como resultado o valor 0,778 que corresponde à similaridade entre os termos (`automóvel`, `automóveis`). A medida CC sempre retorna um grau de similaridade entre 0 e 1, onde 1 indica uma combinação perfeita e zero indica uma ausência de similaridade.

Em nível conceitual, são comparadas as estruturas semânticas das EOs com a utilização de um léxico. Este trabalho utilizou EOs da língua alemã pertencentes ao domínio do turismo. O presente trabalho não faz uso de léxico, mas utiliza a medida de similaridade CC aplicando-a para EOs da língua inglesa.

3. Mapeamento entre EOs

A tarefa de mapeamento entre EOs reflete uma preocupação com a reutilização de EOs existentes, expandindo-as e combinando-as por algum meio.

O mapeamento entre EOs pode-se dar de duas formas. Em primeiro lugar, por meio da conexão entre as EOs e as fontes de informação. Neste caso, as fontes de informação geralmente utilizadas são bases de dados. Por outro lado, o mapeamento pode ser realizado diretamente entre EOs na tentativa de identificar quais termos são iguais ou similares semanticamente. O presente trabalho apresenta-se nesta última forma.

Quando realizado diretamente entre as EOs, o mapeamento tem sido tratado na literatura por duas abordagens principais: união e alinhamento. Essas abordagens serão descritas na próxima subseção.

3.1. União e Alinhamento entre EOs

A primeira abordagem para unir EOs encontrada na literatura está em [Hovy, 1998] e nela são descritas várias heurísticas para identificar conceitos correspondentes entre EOs.

União consiste na criação de uma EO coerente que inclui informação de todas (pelo menos duas) as EOs-base [Noy e Musen, 2000]. Porém, na união entre EOs os termos que não combinam não são detectados nem tratados [Hakimpour e Geppert, 2001].

Já o processo de alinhamento entre EOs é definido em [Sowa, 2001] como sendo o mapeamento de conceitos e relações, entre duas EOs A e B, que preserva em ambas a ordem parcial por subtipos. Dois conceitos ou relações são ditos equivalentes se um alinhamento mapeia um conceito ou relação x em uma EO A para um conceito ou relação y em uma EO B. Caso não exista correspondência entre todos os conceitos ou relações, o mapeamento é dito parcial. A única alteração permitida antes de as EOs serem alinhadas é a inclusão de novos subtipos ou supertipos de conceitos ou relações em ambas as EOs, para prover alvos adequados para o alinhamento [Sowa, 2001]. Entretanto, para realizar estas alterações faz-se necessário um conhecimento prévio, por parte do engenheiro do conhecimento, sobre as EOs sendo alinhadas. O presente trabalho busca evitar este tipo de intervenção manual antes da realização do mapeamento, fazendo com que o alinhamento ocorra de forma automática com os termos existentes nas EOs. Apenas a etapa final, de verificação da consistência do mapeamento automático, requer intervenção manual.

No processo de união entre EOs, duas EOs A e B passam a ser representadas por uma EO C, ao passo que, no processo de alinhamento, as EOs A e B permanecem, e são criadas ligações entre os termos e relações presentes nas mesmas.

O presente trabalho é voltado ao alinhamento entre EOs, sendo criadas ligações entre os termos. O mapeamento realizado será descrito nas próximas seções, que apresentam a aplicação e o aprimoramento do mapeamento automático, as medidas de similaridade e as heurísticas utilizadas.

4. Aplicação e Aprimoramento do Mapeamento Automático entre EOs

O presente trabalho, assim como [Noy e Musen, 2001], utiliza EOs da biblioteca de EOs do programa DAML. O algoritmo desenvolvido trata os formatos OWL⁴, RDF⁵, OIL⁶ e DAML-OIL. Para o engenheiro do conhecimento, quando mapeando duas EOs, as diferentes sintaxes utilizadas pelos padrões OWL, RDF, OIL e DAML-OIL são um fator que dificulta a comparação entre as EOs. Neste sentido, o presente algoritmo abstrai para o usuário a sintaxe, apresentando a EO através de uma interface na forma de uma hierarquia com níveis e subníveis, tal como mostrado na tabela 1. As EOs apresentadas nessa tabela modelam os domínios das comunidades de pesquisa em *Web Semântica*⁷ e dos departamentos acadêmicos de universidades⁸, respectivamente.

⁴*Ontology Web Language* - <http://www.w3.org/TR/owl-ref/>

⁵*Resource Description Framework* - <http://www.w3.org/RDF/>

⁶*Ontology Interchange Language ou Ontology Inference Layer* - <http://www.ontoknowledge.org/oil/>

⁷<http://www.daml.org/ontologies/4>

⁸<http://www.daml.org/ontologies/64>

Tabela 1: Exemplo de extratos de EOs na forma hierárquica

EO ₁	EO ₂
Person	Person
Employee	Student
AcademicStaff	GraduateStudent
FacultyMember	UndergraduateStudent
AssistantProfessor	Worker
AssociateProfessor	AdministrativeStaff
FullProfessor	Chair
Lecturer	SystemStaff
AdministrativeStaff	Faculty
Manager	Lecturer
TechnicalStaff	PostDoct
Student	Professor
Graduate	AssistantProfessor
PhDStudent	AssociateProfessor
Undergraduate	FullProfessor
Event	VisitingProfessor
Lecture	

As principais abordagens encontradas na literatura para unir ou alinhar EOs incluem a comparação de termos ou conceitos em nível lexical e semântico-estrutural. Seguindo estas abordagens elaboramos um algoritmo para promover o mapeamento.

4.1. Comparação Lexical

Inicialmente, para detectar se dois termos são equivalentes ou similares, é realizada uma comparação em nível lexical utilizando a DE de Levenshtein. Se dois termos possuem o mesmo comprimento e seus caracteres estão nas mesmas posições, então bem provavelmente esses termos serão equivalentes, exceto em caso de polissemia, que pode ser tratado por meio da abordagem semântico-estrutural (próxima seção). Alguns resultados das combinações entre duas EOs geradas por esta medida são apresentados na tabela 2.

Tabela 2: Exemplos utilizando medidas de similaridade

EO ₁	EO ₂	Distância de Edição (Levenshtein)	Combinação de Caracteres (Maedche & Staab)
book	book	0	1
masterThesis	mastersThesis	1	0,92
book	work	2	0,5
book	booklet	3	0,25
employee	worker	7	0
academicStaff	faculty	11	0
facultyMember	professor	11	0

Se considerarmos $DE \leq 2$ como limite aceitável para dois termos serem considerados similares, observa-se que serão gerados resultados inconsistentes para termos não

equivalentes, tais como `book` e `work`. Este limite pode ser modificado pelo engenheiro do conhecimento conforme o grau de consistência no resultado do mapeamento.

Na presente abordagem, adotamos a proposta de [Maedche e Staab, 2002] e implementamos a medida CC. Esta medida parece ser melhor do que a DE de Levenshtein uma vez que leva em consideração o comprimento das palavras sendo combinadas.

O presente algoritmo permite ao usuário estabelecer um grau mínimo de similaridade entre termos calculado pela CC, sendo que somente serão gerados mapeamentos acima deste mínimo. Quando este valor for igual a 0,8 por exemplo, combinações como `book` e `work` não serão aceitas (ver tabela 2).

A medida CC diminui a influência de pseudo-diferenças⁹ entre conjuntos de caracteres em EOs distintas. Apesar de esta medida apresentar alguns resultados que induzem ao erro, como por exemplo `lecture` e `lecturer`, a quantidade de combinações corretas contribuiu com o experimento realizado sendo utilizada também em nível semântico-estrutural.

No experimento realizado, ao mesmo tempo que a medida proposta em [Maedche e Staab, 2002] parecia gerar melhores resultados, algumas combinações bastante similares ou equivalentes ainda não estavam sendo mapeadas devido à não similaridade lexical. Para completar o mapeamento entre termos não similares, a comparação foi realizada em nível semântico estrutural, quando utilizamos duas heurísticas que consideram a posição dos termos na hierarquia.

4.2. Comparação Semântico-Estrutural

Após verificar se dois termos sendo combinados são equivalentes em nível lexical, verificam-se suas posições na hierarquia. Caso o termo mais genérico ou algum dos termos mais específicos dos termos sendo combinados sejam equivalentes, então sugere-se que os termos sendo combinados são equivalentes. Caso contrário, é bastante provável a ocorrência de polissemia.

Para realizar a comparação semântico-estrutural, nos apoiamos na medida de similaridade CC, pelo fato de a mesma ter produzido melhores resultados do que a medida de Levenshtein em testes preliminares. Inicialmente, é sugerido o valor mínimo de similaridade 0,8, mas esse valor pode ser alterado pelo usuário a qualquer momento. As heurísticas apresentadas a seguir são aplicadas a todas as EOs não existindo condições especiais para sua aplicação.

4.2.1. Heurística 1: Normalização de Vocabulário

Levando em consideração os exemplos de EOs que estamos trabalhando e o fato de tratar-se da língua inglesa em alguns casos a terminação no texto de um conceito é igual àquela no texto do conceito pai. Por exemplo, os conceitos `GraduateStudent` e `UndergraduateStudent` encontram-se nessa situação.

⁹Por exemplo, casos como uso ou não uso de *underscore* ou hifens, uso de singular e plural ou uso de caracteres com marcação adicional.

Pelo fato de as EOs serem projetadas por pessoas diferentes, com visões de mundo distintas, o texto dos conceitos pode ser diferente, mas sua semântica na EO é a mesma. Para tratar este tipo de situação, o presente algoritmo verifica se um conceito possui na terminação o texto de seu conceito pai e elimina essa terminação para realizar a comparação com os termos de outra EO. No exemplo do parágrafo anterior, `GraduateStudent` e `UndergraduateStudent` passam a ser tratados como `Graduate` e `Undergraduate` na comparação com outra EO.

4.2.2. Heurística 2: Ancestral e Descendentes

Nos casos em que um conceito X em uma EO_A não possui equivalente lexical em uma EO_B verifica-se se o pai do conceito X na EO_A é lexicalmente equivalente ao pai de um conceito Y na EO_B . Se existir uma equivalência lexical, então verifica-se se existe similaridade entre algum dos filhos dos conceitos X e Y . Se pelo menos um filho de X é lexicalmente equivalente a um dos filhos de Y então gera-se um mapeamento entre X e Y . Por exemplo, na tabela 1, o termo `employee` pertencente à EO_1 tem como pai o termo `person`, tal como o termo `worker` na EO_2 . Seguindo na EO_1 verifica-se que `employee` possui o termo `administrativeStaff` como filho, o que também ocorre na EO_2 para o termo `worker`. Assim, é gerado um mapeamento entre `employee` e `worker`.

Após detectada a equivalência entre termos não similares lexicalmente, esses termos passam a ser tratados como equivalentes para as combinações restantes. No exemplo anterior, os termos `employee` e `worker` auxiliam o algoritmo na geração de mais dois mapeamentos. O primeiro mapeamento ocorre entre os termos `academicStaff` e `faculty`, pois ambos possuem pais equivalentes em nível imediatamente superior, e pelo menos um filho (`lecturer`) lexicalmente similar, e o segundo mapeamento ocorre entre os termos `facultyMember` e `professor`.

Finalizado o processamento automático, o especialista do domínio, utilizando o protótipo desenvolvido, pode modificar o grau de similaridade lexical e processar o algoritmo novamente. É possível, também, descartar mapeamentos, caso não se concorde com as combinações geradas, ou ainda acrescentar novas combinações que sejam detectadas manualmente.

4.3. Experimento e Resultados

Um total de 25 EOs nos formatos RDF, OWL, OIL e DAML-OIL foram processadas e exibidas na forma de uma estrutura hierárquica para o usuário. Entretanto, devido a estas EOs serem de domínios muito distintos, poucos mapeamentos puderam ser gerados. Para este artigo, elencamos duas EOs, para as quais alguns conceitos selecionados são apresentados na tabela 1. Essas EOs produziram os dados mostrados na tabela 3 após o processamento automático.

A EO_1 , pertencente ao domínio das comunidades de pesquisa em *Web Semântica*, possui 54 conceitos. Após o processamento utilizando a medida CC, o algoritmo encontrou 16 equivalências com os 44 conceitos pertencentes à EO_2 , cujo domínio são os departamentos acadêmicos de universidades. O valor mínimo de similaridade para gerar um

Tabela 3: Dados sobre as EOs da língua inglesa processadas

	Valor Absoluto		Valor Percentual	
	EO _A	EO _B	EO _A	EO _B
Quantidade Total de Conceitos	54	44	100%	
Quantidade de Mapeamentos (CC)	16		29,62%	36,36%
Quantidade de Mapeamentos (CC + Heurísticas)	22		40,74%	50%
Ganho obtido com as Heurísticas	6		11,12%	13,64%

mapeamento foi igual a 0,75, pois valores menores geraram mapeamentos incorretos, tal como entre os termos *AssistantProfessor* e *VisitingProfessor*. Após utilizar a medida CC, realizamos o mesmo mapeamento utilizando as heurísticas propostas. Essas heurísticas permitiram a geração de mais 6 mapeamentos, o que corresponde a um ganho de 27,27% sobre o resultado obtido somente com a medida CC.

Os percentuais de conceitos mapeados utilizando as heurísticas (40,74% dos conceitos da EO₁ e 50% dos conceitos da EO₂) parecem ser significativos na medida em que uma verificação manual, sob o ponto de vista do usuário, não encontrou mais equivalências entre os conceitos sendo mapeados. Assim, os termos não mapeados possivelmente não possuem equivalência.

Por meio dos experimentos foram constatados casos em que o algoritmo encontrou mais de uma equivalência para um termo. Por exemplo, o termo *lecturer* (EO₂) possui alta similaridade com os termos *lecture* e *lecturer* (EO₁). Neste caso, apenas o resultado com maior valor é gerado para o usuário.

Ao final do processamento, os mapeamentos encontrados pelo algoritmo são apresentados ao usuário juntamente com o valor de similaridade entre os termos mapeados. Além disso, o usuário pode conferir as posições dos termos mapeados na hierarquia, uma vez que seus ícones são marcados diferentemente.

Os resultados apresentados na tabela 3 permitem ao engenheiro do conhecimento estimar o grau de sobreposição existente entre as EOs. No caso de o engenheiro do conhecimento necessitar desenvolver uma EO para um determinado domínio ele pode verificar as EOs já existentes, compará-las e decidir se é viável ou não o reuso das EOs.

5. Considerações e Trabalho Futuro

Este artigo apresenta uma pesquisa em andamento que está distante de ser completa. A solução está em fase inicial de desenvolvimento e novas heurísticas podem surgir ao passo que o trabalho evolui. O artigo descreveu, principalmente, a necessidade de investigar formas alternativas para o mapeamento entre EOs de modo que o usuário perceba o maior número de equivalências existentes entre as EOs sendo comparadas. Aplicamos duas medidas de similaridade em nível lexical e duas heurísticas para tratar o mapeamento em nível semântico-estrutural.

Apesar dos poucos resultados obtidos, os mesmos são encorajadores. Até o mo-

mento foram tratados apenas os conceitos nas EOs. Como trabalho futuro destacamos:

- a experimentação junto a EOs em língua portuguesa;
- uma etapa de validação do algoritmo, na qual serão utilizadas outras EOs além das descritas neste artigo;
- o tratamento do formato de saída do algoritmo (atualmente, o formato é apenas legível por humanos e não por máquinas).

Referências

- Ding, Y. e Foo, S. (2002). Ontology Research and Development Part 2 - A Review of Ontology Mapping and Evolving. *Journal of Information Science*, 28(5):375–388.
- Doan, A., Madhavan, J., Domingos, P., e Halevy, A. (2002). Learning to Map between Ontologies on the Semantic Web. In *Proceedings of the World-Wide Web Conf. (WWW-2002)*, Honolulu, Hawaii, USA.
- Hakimpour, F. e Geppert, A. (2001). Resolving Semantic Heterogeneity in Schema Integration: an Ontology Based Approach. In *Proceedings of the International Conference on Formal Ontology in Information Systems FOIS-2001*.
- Hovy, E. (1998). Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses. In *Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC)*, Granada, Spain.
- Levenshtein, I. V. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Cybernetics and Control Theory*, 10(8):707–710.
- Maedche, A. e Staab, S. (2002). Measuring Similarity between Ontologies. In *Proceedings Of the European Conference on Knowledge Acquisition and Management - EKAW-2002. Madrid, Spain, October 1-4*, pages 251–263.
- Noy, N. F. e Musen, M. (1999). SMART: Automated Support for Ontology Merging and Alignment. In *Twelfth Banff Workshop on Knowledge Acquisition, Modeling, and Management - Banff, Alberta, Canada*.
- Noy, N. F. e Musen, M. (2000). PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *Proceedings of the 17th National Conference on Artificial Intelligence, (AAAI'00)*.
- Noy, N. F. e Musen, M. A. (2001). Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In *Proceedings of the Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*, Seattle, WA.
- Prasad, S., Peng, Y., e Finin, T. (2002). Using Explicit Information To Map Between Two Ontologies. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems - Workshop on Ontologies in Agent Systems (OAS) - Bologna, Italy. 15-19 July*.
- Sowa, J. F. (2001). Building, Sharing, and Merging Ontologies. <http://www.jfsowa.com/ontology/ontoshar.htm>. Acessado em Janeiro de 2003.
- Uschold, M. (2001). Where is the Semantics in the Semantic Web? . In *Workshop on Ontologies in Agent Systems*, Montreal, Canada.