Classificador de alinhamento de ontologias utilizando técnicas de aprendizado de máquina

Alex Alves, Anselmo Guedes, Kate Revoredo, Fernanda Baião

Departamento de Informática Aplicada
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

{alex.alves,anselmo.guedes,katerevoredo,fernanda.baiao}@uniriotec.br

Abstract. *The ontology alignment process is a necessary step to reduce the semantic heterogeneity among ontologies. This paper presents a machine learning approach to generate ontologies alignment classifiers, based on data alignments found through different similarity functions.*

Resumo. *O processo de alinhamento de ontologias é uma das etapas necessárias para que se possa reduzir a heterogeneidade semântica entre ontologias existentes. Este trabalho apresenta uma abordagem baseada em técnicas de aprendizado de máquina para gerar modelos classificadores de alinhamento de ontologias, tendo como base de dados os alinhamentos encontrados através de diferentes funções de similaridade.*

1. Introdução

Ontologia é a representação do conhecimento de um domínio, onde um conjunto de objetos e seus relacionamentos são descritos por um vocabulário [Gruber 1993]. É comum que uma ontologia represente a taxonomia de um domínio, que é representada por classes de objetos e relações entre eles [Berners-Lee *et al*. 2001]. Com a crescente e diversificada utilização de ontologias na tecnologia da informação, é inevitável que haja uma heterogeneidade semântica dos domínios representados por cada ontologia.

Para superar o problema da heterogeneidade semântica, uma das etapas necessárias é realizar a combinação das entidades para a determinação de um alinhamento, isto é, um conjunto de correspondências entre os elementos das ontologias [Shvaiko e Euzenat 2011]. O processo de alinhamento de ontologias encontra correspondências entre os elementos de cada ontologia e denota a força desta correspondência com um valor entre 0 e 1. Diversas soluções de alinhamento de ontologias vêm sendo propostas nos últimos anos, dentre elas técnicas que utilizam aprendizado de máquina [Mitchell 1997].

Este trabalho tem como objetivo a geração de um classificador que seja capaz de indicar se existe uma correspondência entre elementos de 2 ontologias.

O artigo está estruturado da seguinte forma: Na Seção 2 é apresentada a fundamentação teórica; na Seção 3 a abordagem para geração do classificador é apresentada; na Seção 4, os experimentos executados e os resultados obtidos são descritos; na Seção 5 são abordados alguns trabalhos relacionados; e na Seção 6 é concluído o trabalho.

2. Fundamentação Teórica

Nesta seção, os conceitos utilizados no desenvolvimento desta proposta são apresentados.

2.1 Alinhamento de Ontologias

Em sistemas abertos e distribuídos, como a web semântica, a heterogeneidade dos dados não pode ser evitada. Diferentes pessoas têm interesses e hábitos diferentes, utilizam ferramentas diferentes e possuem conhecimento, na maioria das vezes, em diferentes níveis de detalhe [Alex Alves et al. 2012]. Estas várias razões levam a diversas formas de heterogeneidade, e, portanto, devem ser cuidadosamente levadas em consideração.

A questão da heterogeneidade também pode ser percebida quando considerando a modelagem de um domínio através de Ontologias [Gruber 1993]. Uma ontologia muitas vezes pode ser utilizada para definir o vocabulário utilizado por alguma aplicação em particular.

O objetivo do alinhamento de duas ou mais ontologias é reduzir a heterogeneidade existente entre elas. A heterogeneidade não reside exclusivamente nas diferenças entre os objetivos das aplicações de acordo com o fim para o qual foram concebidas ou nos formalismos expressos nas ontologias na qual foram codificadas. O alinhamento de ontologias ocorre no sentido de identificar as correspondências entre os elementos individuais de múltiplas ontologias, e é uma condição necessária para estabelecer a interoperabilidade entre elas [Erigh 2007].

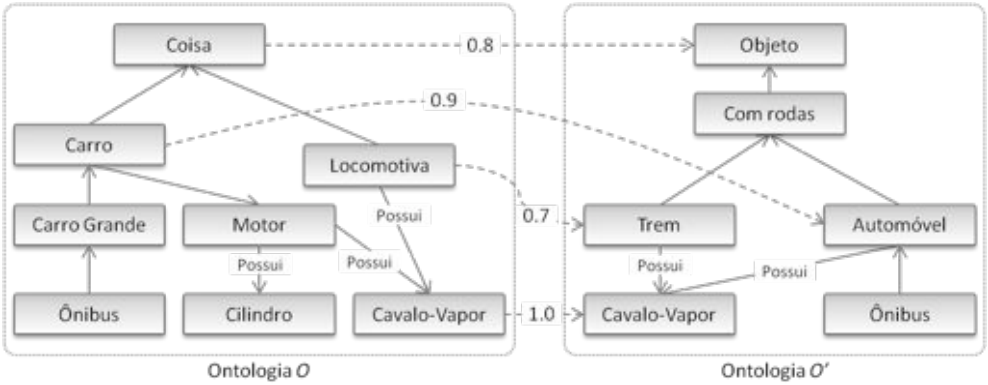


Figura 1. Exemplo de alinhamento entre duas ontologias. Adaptado de [Abolhassani, *et. al.* 2006].

Por exemplo, considere as duas ontologias (O e O') ilustradas na Figura 1, onde as arestas pontilhadas indicam correspondências entre elementos de O e O'. Por exemplo, existe uma correspondência entre os elementos Carro e Automóvel das ontologias O e O' respectivamente. Além disso, é possível indicar a confiança, a força, desse alinhamento através da atribuição de peso às arestas. A confiança da correspondência entre Carro e Automóvel é de 0.9.

2.2. Métricas de qualidade

A natureza dos conjuntos de dados que serão utilizados em um projeto de avaliação de alinhamento de ontologias deve atender dois requisitos: (i) cobertura de aspectos relevantes e (ii) imparcialidade da avaliação. Este conjunto de dados consiste

tipicamente de pelo menos duas ontologias e um alinhamento esperado entre essas duas ontologias, chamado de alinhamento de referência [Ehrig 2007]. A seleção das métricas para a avaliação é uma tarefa que deve levar em consideração o critério de não favorecer qualquer abordagem de alinhamento. Para a tarefa de alinhamento de ontologias, tipicamente têm sido utilizadas as seguintes métricas:

- Verdadeiros Positivos (VP): É o conjunto de correspondências encontradas, que fazem parte do conjunto de correspondências do alinhamento de referência;
- Falsos Positivos (FP): É o conjunto de correspondências encontradas, que não fazem parte do conjunto correspondências do alinhamento de referência;
- Falso Negativos (FN): É o conjunto de pares de elementos que não foram identificados como correspondências possíveis e que estão presentes no conjunto de correspondências do alinhamento de referência;
- Precisão (*Precision*): A precisão mede a proporção de correspondências corretas que foram encontradas, ou seja, dentre as correspondências encontradas quantas realmente são? Dado um alinhamento de referência *R*, a precisão de um alinhamento *A* pode ser calculada da seguinte forma:

$$P(A,R) = \frac{[\mathrm{VP}]}{[\mathrm{VP+FP}]}$$

- Cobertura (*Recall)*: A cobertura mede a proporção de correspondências corretas encontradas dentre todas as possíveis. Dado um alinhamento de referência *R*, a cobertura de um alinhamento *A* pode ser calculada da seguinte forma:

$$R(A,R) = \frac{[\mathrm{VP}]}{[\mathrm{VP+FN}]}$$

- Medida-F (*F-Measure*): A Medida-F representa a harmonização entre a precisão e a cobertura. Pode ser a medida principal para avaliar a qualidade de um alinhamento. Dado um alinhamento de referência *R*, a precisão e a cobertura, a medida-F pode ser calculada da seguinte forma:

$$F(A,R) = \frac{(b^2+1).P(A,R).R(A,R)}{b^2.\mathrm{P(A,R)+R(A,R)}}$$

Com *b*=1 sendo um fator de peso padrão, chega-se a:

$$F_1(A,R) = \frac{2.P(A,R).R(A,R)}{P(A,R)+R(A,R)}$$

2.3. Descoberta de Conhecimento em Base de Dados

O processo de Descoberta de Conhecimento em Bases de Dados (Knowledge Discovory in Databases (KDD)) envolve fases e tarefas [Marcos *et al*. 2012]. Segundo Fayyad *et al*. [1996], KDD é um processo, de várias etapas, não trivial, interativo e iterativo, para identificação de padrões compreensíveis, válidos, novos e potencialmente úteis a partir de grandes conjuntos de dados.

Técnicas de KDD serão utilizadas, neste trabalho, para a geração de um classificador capaz de indicar se existe correspondência entre pares de elementos de duas ontologias. A ferramenta de KDD, Weka[1], será utilizada para apoiar os experimentos. A escolha da Ferramenta Weka se deve ao fato de estar disponível para uso (licença GPL) e ser amplamente utilizada em trabalhos científicos [Ian *et al*. 2011], [Thornton *et al*. 2012]. O software foi escrito na linguagem Java e contém uma GUI (*Graphical User Interface*) para interagir com arquivos de dados e produzir resultados visuais. Os algoritmos de aprendizado de máquina, que tem por objetivo aprender um classificador a partir de uma base de dados composta por atributos e uma classe, considerados neste artigo estão descritos nas Seções a seguir.

2.3.1 Regressão Linear

Quando os atributos são numéricos, a regressão linear é uma técnica natural a considerar. Este é um método básico na estatística. A ideia é expressar a classe como uma combinação linear dos atributos com pesos pré-determinados [Ian *et al*. 2011].

$$X=w_0+w_1a_1+w_2a_2+ \ldots +w_ka_k$$

Onde: x é a classe; $a_1, a_2, \ldots, a_k$ são os valores de atributos, e $w_0, w_1, \ldots, w_k$ são pesos. Os pesos são calculados a partir dos dados de treinamento. A notação fica um pouco pesada, porque é necessária uma forma de expressar os valores de atributos para cada instância de treinamento. A primeira instância terá uma classe, por exemplo, $x^{(1)}$, e valores de atributo $a_1^{(1)}, a_2^{(1)}, \ldots, a_k^{(1)}$, onde o expoente indica que se trata do primeiro exemplo. Uma vez que os cálculos tenham sido realizados, o resultado é um conjunto de pesos numéricos, com base nos dados de treino, o qual pode ser usado para predizer a classe de novos casos.

2.3.2 Máquina de vetores de suporte

Máquina de vetores de suporte (*Support Vector Machine (SVM)*) possui seus fundamentos na teoria de aprendizagem estatística e tem mostrado resultados empíricos promissores em muitas aplicações práticas [Witten *et al*. 2011]. Consiste em um método de aprendizado que tenta encontrar um hiperplano ótimo de modo que ele possa separar diferentes classes de dados com a maior margem possível [Pang-Ning *et al*. 2009], chamada *Soft Margin* como mostra a Figura 2.

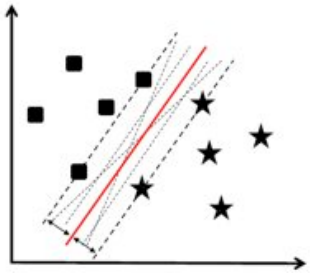


Figura 2 – Hiperplano Ótimo. Fonte: [Lima 2013]

[1] http://www.cs.waikato.ac.nz/ml/weka/.

O SVM foi originalmente concebido para lidar com classificações binárias, entretanto a maior parte dos problemas reais requer múltiplas classes. Para se utilizar uma SVM para classificar múltiplas classes é necessário transformar o problema multiclasse em vários problemas de classes binárias. Outro fator importante a considerar é que em muitos problemas reais as classes não são linearmente separáveis mesmo utilizando a margem de folga. Nestes casos, SVM mapeia os dados para um espaço de dimensão maior para então procurar pelo hiperplano ótimo.

3. Abordagem Proposta

A abordagem proposta no presente trabalho se baseia na criação de um modelo classificador que explicita o quanto cada função de similaridade avaliada é determinante para a obtenção de uma correspondência correta. Este trabalho não se propõe a avaliar nenhuma ferramenta de alinhamento ou função de similaridade individualmente, mas sim servir como um recurso externo a essas ferramentas para aumentar a qualidade do alinhamento por elas gerado, tomando como base o classificador obtido.

Para a geração do conjunto de dados que será utilizado para obtenção do classificador, é necessário que alguns passos sejam executados de forma sistemática, para tal propõe-se que sejam executados os passos ilustrados na Figura 3.

Figura 3 – Proposta de processo para geração de conjunto de dados para obtenção do classificador

1. Geração de produto cartesiano: É comum que as ferramentas de alinhamento de ontologias retornem apenas as correspondências encontradas entre as ontologias alinhadas. Para que o processo de mineração de dados, que irá gerar o classificador, seja mais eficiente, propõe-se que seja gerado o produto cartesiano dos elementos das duas ontologias, a fim de gerar uma base com todas as associações possíveis entre os elementos das ontologias O e O'.

2. Identificação das correspondências: O preenchimento do atributo classe é feito atribuindo 1 para os pares de elementos incluídos no alinhamento de referência e o resto dos elementos que não estão no alinhamento de referência são preenchido com 0.

3. Criação de Função como atributo No conjunto de dados que será utilizado para o processo de mineração dados, cada função deve virar um atributo contendo os valores encontrados em cada correspondência. A ideia é gerar um classificador que combina diversas funções de similaridade, aplicando mineração de dados. Dessa forma o conjunto de dados a ser treinado para geração do classificador pode ser exemplificado conforme a Tabela 1.

Tabela 1. Exemplo de conjunto de dados para mineração de dados

Função 1	Função 2	Função 3	Atributo de Classe
1,00	0,84	0,94	1
0,00	0,00	0,00	1

1,00	0,22	0,41	0
0,00	0,00	0,00	0

Após a geração do *dataset* de acordo com os passos de 1 a 3 é aplicado um algoritmo de classificação sobre os dados gerados para o aprendizado supervisionado do modelo.

4. Execução do Experimento

Os conjuntos de dados disponibilizados pela OAEI[2] [Shvaiko e Euzenat 2011] estão em observância com os critérios necessários para a realização de um projeto de avaliação de alinhamento de ontologias, e possuem as características necessárias para a execução do processo apresentado na Seção 3. Diante disso, o conjunto de dados *benchmark*, disponibilizado pela campanha da OAEI de 2012, foi selecionado para a execução do experimento. Este conjunto de dados é composto de 51 ontologias, sendo uma base e as restantes variações desta para a execução do alinhamento entre elas. Para cada alinhamento é fornecido um alinhamento de referência, através da qual é possível calcular as métricas de qualidade e executar o processo de aprendizado supervisionado. OAEI é uma iniciativa internacional coordenada cujo objetivo é avaliar os pontos fortes e fracos das ferramentas de alinhamento, comparar o desempenho de técnicas, melhorar as técnicas de avaliação e etc...

4.1 Geração dos alinhamentos

Para o experimento foram utilizadas funções de similaridade baseadas em *String*, que comparam os termos e atribuem uma força de alinhamento entre [0,1] levando em consideração a estrutura em sequências de texto, que são vistos como sequências de caracteres. Estas técnicas consideram que a semelhança entre dois termos aumenta quando a semelhança entre as suas cadeias de texto correspondentes também aumenta, mas sem ter em conta a semântica subjacente nos termos. Tais funções, de acordo com a classificação das abordagens de alinhamento proposta por Euzenat [2007], são técnicas no nível sintático e/ou terminológico. As funções de similaridade utilizadas estão na Tabela 2:

Tabela 2 - Funções de similaridade

Função de Similaridade	Descrição
1. StringDistAlignment	Função baseada em nomes e distância entre os nomes. Pode combinar livremente nomes de propriedades com nomes de classes.
2. NameEqAlignment	Função que realiza o alinhamento de duas ontologias baseado na igualdade do nome de suas entidades, por exemplo, associando 1 quando os nomes forem iguais e 0 caso contrário.
3. EditDistNameAlignment	Usa uma edição de distância (ou *Levenstein*) entre nomes de entidade. Assim, uma matriz de distâncias é construída e o alinhamento a partir das distâncias é escolhido.
4. NameAndPropertyAlignment	Esta função leva em consideração o nome da classe e suas propriedades para realizar o alinhamento.
5. SMOANameAlignment	Essa função utiliza a técnica SMOA (*String Metric for Ontology Alignmemt*) (Métrica de *Strings* para alinhamento de ontologias) [Stoilos *et al*. 2005] para encontrar correspondências entre os elementos.
6. SubsDistNameAlignment	Função que realiza o alinhamento de duas ontologias baseado na distância das *substrings* geradas para os nomes dos elementos. Por padrão utiliza a distância *Levenstein*.
7. Trigram	Essa função é um caso especial do N-Gram [Kondrak 2005] onde N=3. N-Gram é um

[2] http://oaei.ontologymatching.org/

	modelo probabilístico de predição do próximo item em uma sequência.
8. Levenshtein [1966]	Função baseada em string que compara os elementos por seus nomes locais. Examina o número mínimo de operações (inserções, exclusões e / ou substituições) que são necessárias para transformar uma sequencia de texto em outro.
9. SmithWaterman [1981]	Função para a realização de alinhamento de sequências local, elaborado para determinar as regiões similares entre duas sequências. Em vez de olhar a sequência total, compara segmentos de todos os comprimentos possíveis e otimiza a medida de similaridade.
10. QGramsDistance 11. [Kondrak 2005]	Função baseada em *string* que compara os elementos por seus nomes locais. A *string* é dividida em *tokens* com comprimento igual 2. Um q-gram, neste contexto, refere-se a uma sequência de letras, a partir de uma dada palavra.
12. JaroWinkler [1989]	Função baseada em string que analisa o número e a ordem dos dois caracteres comuns em duas cadeias de texto.
13. MongeElkan [1996]	A função de distância Monge & Elkan (1996), é uma variante afim da função de distância de Smith-Waterman, com parâmetros de custo em particular, e dimensionado para o intervalo [0,1].

Para a geração das correspondências utilizando as funções de 1 a 6, foi utilizada a *Alignment API and Server 4.4*[3], que é uma plataforma extensível de alinhamento de ontologias desenvolvida pelos organizadores do *OAEI*, caracterizada por um conjunto de abstrações para expressar, acessar e compartilhar alinhamentos de ontologias. A função 7 foi utiliza com a ferramenta *COMA Community Edition 3.0*[4], desenvolvida pela *University of Leipzig*, Alemanha, que é uma ferramenta baseada em workflow com a possibilidade de combinação de funções. Por fim, as funções 8 a 12 estão disponíveis no projeto *Simmetrics*[5], que é uma biblioteca de similaridade métrica de edição de distância, fornecida pela Universidade de *Sheffield*, no Reino Unido.

4.2 Processo de Mineração

Para a criação do modelo de classificação, foram selecionados os algoritmos de Regressão Linear e Support Vector Machines – SVM. Ambos os algoritmos foram executados utilizando a ferramenta de mineração de dados Weka.

No processo de experimentação, para cada algoritmo foram geradas algumas variações do modelo de classificação, utilizando os seguintes critérios: (i) modelo classificador utilizando todas as funções de correspondência, (ii) modelo classificador utilizando as três funções que apresentaram melhores resultados da métrica de precisão, (iii) modelo classificador utilizando as três funções que apresentaram melhores resultados da métrica de cobertura e (iv) modelo classificador utilizando as três funções que apresentaram melhores resultados da métrica Medida-F. Cada variação mencionada foi aplicada sobre o conjunto de dados completo e sobre o conjunto de dados somente de conceitos utilizando as funções das ferramentas de alinhamento de 1 a 7 que estão apresentadas na Seção 4.1 e depois foi gerado outro processo para geração do classificador utilizando as funções de alinhamento de 8 a 12 onde foram considerados todos os elementos de todos os conjuntos de *dataset* do Benchmark 2012 do *OAEI*.

[3] http://alignapi.gforge.inria.fr/

[4] http://dbs.uni-leipzig.de/Research/coma.html

[5] http://sourceforge.net/projects/simmetrics/

4.3 Resultados do Experimento

O primeiro processo para geração do classificador utilizou os critérios descritos na Seção anterior e os cenários descritos na Tabela 3.

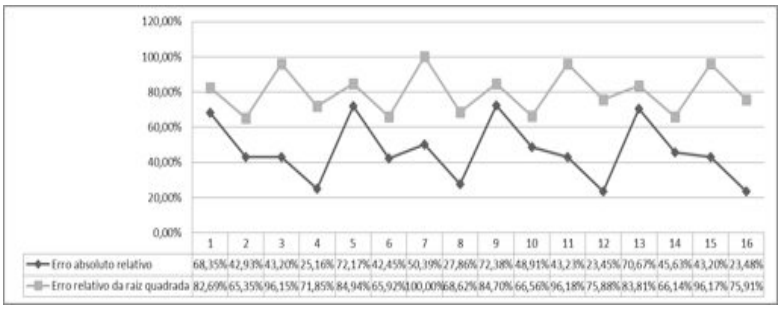
Tabela 3. Cenários dos conjuntos de dados para mineração

Cenário	Descrição
1	Regressão Linear considerando todas as funções de similaridade utilizando todos os elementos das ontologias
2	Regressão Linear considerando todas as funções de similaridade utilizando somente os conceitos das ontologias
3	SVM considerando todas as funções de similaridade utilizando todos os elementos das ontologias
4	SVM considerando todas as funções de similaridade utilizando somente os conceitos das ontologias
5	Regressão Linear considerando as três funções que apresentaram melhores resultados de cobertura utilizando todos os elementos das ontologias
6	Regressão Linear considerando as três funções que apresentaram melhores resultados de cobertura utilizando somente os conceitos das ontologias
7	SVM considerando as três funções que apresentaram melhores resultados de cobertura utilizando todos os elementos das ontologias
8	SVM considerando as três funções que apresentaram melhores resultados de cobertura utilizando somente os conceitos das ontologias
9	Regressão Linear considerando as três funções que apresentaram melhores resultados de precisão utilizando todos os elementos das ontologias
10	Regressão Linear considerando as três funções que apresentaram melhores resultados de precisão utilizando somente os conceitos das ontologias
11	SVM considerando as três funções que apresentaram melhores resultados de precisão utilizando todos os elementos das ontologias
12	SVM considerando as três funções que apresentaram melhores resultados de precisão utilizando somente os conceitos das ontologias
13	Regressão Linear considerando as três funções que apresentaram melhores resultados de Medida-F utilizando todos os elementos das ontologias
14	Regressão Linear considerando as três funções que apresentaram melhores resultados de Medida-F utilizando somente os conceitos das ontologias
15	SVM considerando as três funções que apresentaram melhores resultados de Medida-F utilizado todos os elementos das ontologias
16	SVM considerando as três funções que apresentaram melhores resultados de Medida-F utilizado somente os conceitos das ontologias

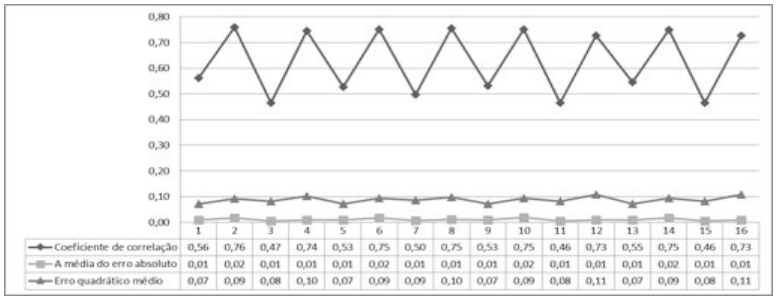
Para avaliar os classificadores consideramos as seguintes métricas: erro médio absoluto (*Mean absolute error*), raiz do erro médio quadrático (*Root mean squared error*), raiz do erro relativo ao quadrado (*Root relative squared error*), erro relativo absoluto (*Relative absolute error*) e o coeficiente de correlação (*Correlation coefficient*).

Todos os modelos gerados apresentaram bons resultados (medida-F) no teste de validação. É interessante observar que os modelos que utilizaram o conjunto de dados relacionando todas os elementos das ontologias entre si (incluindo conceitos, relações e tipos) apresentaram melhores resultados sobre os modelos que utilizaram o conjunto de dados relacionando somente com conceitos (ver Figuras 4a e 4b), o que seria a princípio mais intuitivo. Esse fenômeno pode ter ocorrido pelo tamanho do conjunto de dados, onde o conjunto de dados considerando pares formados por todos os elementos possui 34.225 instâncias, ao passo que o conjunto de dados formado apenas por pares considerando conceitos possui 2.618 instâncias.

Ao comparar os algoritmos utilizados, SVM apresentou, em todos os casos, menor taxa de erros que Regressão Linear, dando destaque à uniformidade da taxa de erros, que pouco varia em função da variação do conjunto de dados utilizados.



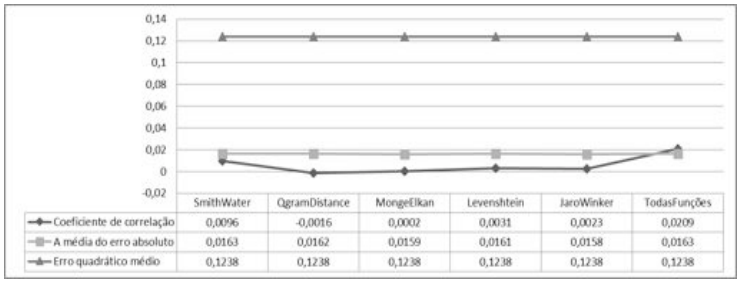
(a)



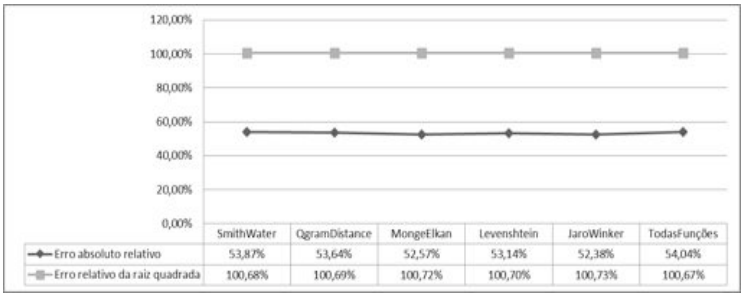
(b)

Figura 4 – Resultados do processo 1 nos conjuntos de dados para obtenção do classificador

Já no segundo processo para a geração do classificador foi desenvolvido um protótipo e utilizadas as funções de similaridade de 8 a 12, onde foram considerados todos os elementos, de acordo com os resultados do processo anterior.



(a)



(b)

Figura 5 – Resultados do processo 2 nos conjuntos de dados para obtenção do classificador

Todos os classificadores gerados apresentaram bons resultados no teste de validação como no processo anterior, e também o algoritmo SVM teve os melhores resultados em relação à Regressão Linear. Portanto, o gráfico das Figuras 5a e 5b só mostram os resultados do algoritmo SVM por função de similaridade utilizada e resultado com a combinação de todas as funções. O conjunto de dados gerado considerando todos os pares de elementos possui 130.905 instâncias, quase 100.000 instâncias a mais que o processo anterior o que fez com que o tempo para processar cada função no algoritmo SVM levasse em média 2:30 horas.

5. Trabalhos Relacionados

YAM ++ [Duyhoa *et al*. 2011] é uma ferramenta de alinhamento de ontologia, que suporta a descoberta de alinhamento de ontologias por abordagens de aprendizado de máquina. O processo de alinhamento pode ser decomposto em três passos: (1) na fase de aprendizagem, o usuário irá selecionar um conjunto de métricas de similaridade, um conjunto de dados *gold standard* e um modelo de aprendizagem de máquina; (2) YAM + + cria dados de treinamento e realiza um processo de formação; e (3) a fase de classificação. YAM + + gera uma classificação, que é usada para classificar e produzir os mapeamentos entre ontologias de entrada. Diferente da nossa abordagem que tem o propósito de criar um modelo de classificador para auxiliar na verificação do alinhamento se ele é ou não um alinhamento valido, já o YAM++ utiliza um classificador para produzir o alinhamento.

Erigh [2007] propõe um processo para otimizar um alinhamento de forma parametrizável, que foi chamado de APFEL (*Alignment Process Feature Estimation and Learning*). Este processo consiste basicamente no treinamento supervisionado de uma base de alinhamentos para a classificação do alinhamento, com a entrada de dados parametrizada e o alinhamento obtido validado pelo usuário. No presente trabalho, o processo ocorre de forma automática, uma vez que é fornecido o alinhamento de referência, dado que o processo APFEL não considera, sendo assim dispensável a validação pelo usuário do alinhamento obtido.

GLUE [Doan *et al*. 2003] é um sistema de alinhamento de ontologias que explora a aprendizagem de máquina para calcular mapeamentos semânticos entre os conceitos. Considerando duas ontologias distintas, o processo de descoberta do mapeamento entre os conceitos é baseado em medida de similaridade, a qual é definida através da distribuição de probabilidade conjunta calculada por meio de duas técnicas de aprendizagem base aplicados às instâncias ontologia. Neste trabalho há a necessidade da intervenção do usuário no processo de alinhamento que é semiautomático.

6. Conclusão

O alinhamento de ontologias tem sido amplamente utilizado para tratar a heterogeneidade semântica nos mais diversos cenários. No entanto, as abordagens e ferramentas atuais de alinhamento se baseiam em funções de similaridade que ainda resultam em baixa precisão e cobertura. Neste sentido, este trabalho propôs uma abordagem para o aprendizado de um modelo classificador que automaticamente descobre qual a melhor combinação de funções de similaridade que resulta em um alinhamento de melhor qualidade. A abordagem proposta não tem o intuito de avaliar as ferramentas de alinhamento ou as funções de similaridade individualmente, mas o

modelo gerado pode ser utilizado como um recurso externo para melhorar os resultados de ferramentas atuais.

Uma limitação da proposta é que são aplicáveis somente às funções que foram utilizadas para a geração do classificador. Um trabalho futuro é procurar gerar uma função genérica para avaliação e classificação de alinhamento de ontologias e outro seria fazer a processo para geração do classificador utilizando todas as funções de similaridades em um único processo utilizando o algoritmo SVM.

7. Referências

Abolhassani, H., Hariri, B., Haeri, S., On Ontology Alignment Experiments. Webology. 2006. Disponível em: http://www.webology.org/2006/v3n3/a28.html

Alex Alves, Natalia Padilha, Sean Siqueira, Fernanda Baião and Kate Revoredo. (2012), "Using Concept Maps and Ontology Alignment for Learning Assessment", IEEE Technology and Engineering Education (ITEE), 1558-7908 © 2012 IEEE Education Society Students Activities Committee (EdSocSAC).

Berners-Lee, T., Hendler, J. E Lassila, O. (2001) The semantic web, Scientific American, p. 28-37

C. Thornton, F. Hutter, H.H. Hoos, and K. Leyton-Brown, (2012) Auto-WEKA: Automated Selection and Hyper-Parameter Optimization of Classification Algorithms. ;In Proceedings of CoRR.

Doan A., Madhavan J., Domingos P., Halevy A., (2003) Ontology matching: a machine learning approach. In: Handbook on ontologies in information systems. New York: Information Science Reference, pp 397–416.

Duyhoa N., Zohra B., E Remi C. (2011) "A Flexible system for ontology matching". Proceedings In Caise 2011 Forum.

Ehrig, M. (2007) Ontology Alignment: Bridging the Semantic Gap, Springer.

Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., Trojahn, C., (2011) Ontology Alignment Evaluation Initiative: six years of experience, Journal on data semantics XV.

Euzenat, J., Shvaiko, P. (2007) Ontology Matching, Springer-Verlag Berlin Heidelberg.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., (1996) "From Data Mining to Knowledge Discovery in Databases", AI Magazine, American Association for Artificial Intelligence, pp. 37-54.

G. Kondrak. (2005) "N-gram similarity and distance". Proceedings of the Twelfth International Conference on String Processing and Information Retrieval (SPIRE 2005), pp. 115-126, Buenos Aires, Argentina.

Gruber, T., R. (1993) "A translation approach to portable ontologies. Knowledge Acquisition", pp. 199-220. Disponível em http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html.

Ian H. W., Eibe F., Mark A. H., Geoffrey H. (2011) “Data Mining: Practical Machine Learning Tools and Techniques” (The Morgan Kaufmann Series in Data Management Systems) (3rd Edition)

Jaro, M. A. (1989). “Advances in record-linkage methodology as applied to matching the 1985 census of Tampa”, Florida. Journal of the American Statistical Association 84:414–420.

Levenstein, I. (1966) “Binary codes capable of correcting deletions, insertions and reversals”. Cybernetics and Control Theory.

Lima, E.S., Pozzer, C.T., D'ornellas, M., Ciarlini, A.E.M., Feijo, B., Furtado, A.L., 2009. Support Vector Machines for Cinematography Real-Time Camera Control in Storytelling Environments. In: VIII Brazilian Symposium on Games and Digital Entertainment, Rio de Janeiro, Brazil, pp. 44-51. [DOI: http://doi.ieeecomputersociety.org/10.1109/SBGAMES.2009.14].

Mitchell, T. M. (1997) Machine Learning. McGraw-Hill.

Monge, A., Elkan, C. (1996) “The field-matching problem: algorithm and applications”. In: Proceedings of the second international Conference on Knowledge Discovery and Data Mining.

Pang-Ning T., M. Steinbach, V. Kumar: (2009) Introdução ao “Data Mining” Mineração de Dados. Rio de Janeiro: Editora Ciência Moderna Ltda.

Smith, T. F.; Waterman, Michael S. (1981). "Identification of Common Molecular Subsequences". Journal of Molecular Biology 147 pp. 195–197. DOI:10.1016/0022-2836(81)90087-5.

Marcos A., Kate R., Leila A., (2012) Avaliando uma Oportunidade Exploratória de Petróleo através de Mineração de Dados: VIII Simpósio Brasileiro de Sistemas de Informação, SP. São Paulo, 2012. v. 3. p. 666-671.