Regras SWRL: Análise de similaridade e detecção de erros

Adriano Rivolli[1, 2], João Paulo Orlando[1], Cláudio H. Yamamoto[2], Dilvan A. Moreira[1]

Instituto de Ciências Matemáticas e de Computação[1]
Universidade de São Paulo – USP
Av. Trabalhador Sancarlense, 400 – São Carlos – SP

IFSP Salto[2]
Departamento de Informática
Rua Rio Branco, 1780 - Vila Teixeira - Salto – SP

rivolli@icmc.usp.br, orlando@icmc.usp.br, haruo@cefetsp.br, dilvan@gmail.com

RESUMO

A Web Semântica renovou o interesse em sistemas baseado em regras. *Semantic Web Rule Language* (SWRL) é uma linguagem que permite combinar regras com ontologias definidas em *Web Ontology Language* (OWL) e assim aumentar sua expressividade. Todavia, desenvolvedores encontram dificuldades em gerenciar adequadamente grandes quantidades de regras. Um sistema com muitas regras é difícil de entender e propício a erros, principalmente quando usado e mantido colaborativamente. Para minimizar este problema, técnicas e ferramentas são necessárias para organizar, visualizar e criar conjuntos de regras SWRL. Este trabalho discute e apresenta duas estratégias nesta direção: a análise de similaridade entre regras e a detecção de erros léxicos e sintáticos. Elas nos permitem compreender, visualizar e corrigir regras SWRL em grandes bases de conhecimentos.

ABSTRACT

The Semantic Web renewed the interest in rule based software systems. Semantic Web Rule Language (SWRL) is a rule language that allows rules to be combined with Web Ontology Language (OWL) knowledge bases to improve its expressivity. However, developers face difficulties on managing large rule sets. A large rule set is difficult to understand and error prone, especially when used and maintained collaboratively. To minimize this problem, techniques and tools are needed to organize, view and create large rule sets in SWRL. This paper discusses and presents two strategies in this direction: the similarity analysis between rules and the detection of syntactic and lexicon errors. They can help us to understand, view and correct SWRL rules in large knowledge bases.

Categories and Subject Descriptors

H.3.4 [**Information storage and retrieval**]: Systems and Software – *Semantic Web.*

General Terms

Algorithms, Management.

Keywords

Rule; Rule SWRL; Rule Similarity; Rule Errors; Semantic Web. SWRL;

WebMedia'11: Proceedings of the 17th Brazilian Symposium on Multimedia and the Web. Short Papers.
October 3 -6, 2011, Florianópolis, SC, Brazil.
ISSN 2175-9650.
SBC - Brazilian Computer Society

1. INTRODUÇÃO

O crescimento do uso de regras na Web Semântica contribuiu para renovar e aumentar o interesse em sistemas de regras e seu desenvolvimento [1]. *Semantic Web Rule Language* (SWRL) aumenta a expressividade da *Web Ontology Language* (OWL), que é uma poderosa linguagem que especifica descrições de alto nível para conteúdo Web e conjuntos de dados [2].

Usuários e desenvolvedores de regras têm se deparado com alguns problemas, principalmente quando o conjunto de regras se torna grande ou possui regras complexas [3]. Assim, eles precisam de ferramentas para criar, visualizar e gerenciar regras, que possibilitem principalmente: A aquisição do conhecimento sem inconsistências, ambiguidade e regras duplicadas; e a visualização de regras e conjunto de regras de forma a facilitar o entendimento e conhecimento das mesmas.

Neste trabalho são apresentadas duas técnicas para regras SWRL. A primeira consiste em determinar a similaridade entre regras, enquanto a segunda consiste na detecção de erros em regras. A partir da análise de similaridade de regras, foram desenvolvidos o agrupamento de regras e a sugestão de termos na criação de novas regras. Uma *Application Programming Interface* (API) realiza a abstração lógica das regras e integra estas técnicas em uma ferramenta para criação e visualização de regras, em desenvolvimento.

2. SWRL

SWRL é uma expressiva linguagem de regras que combina cláusulas *Horn* com conceitos definidos em OWL e pode ser usada para aumentar a capacidade de inferência sobre os indivíduos em uma base de conhecimento em OWL [2]. Regras em SWRL são compostas de duas partes: o antecedente (*body*) e o consequente (*head*). Cada regra é uma implicação entre o antecedente e o consequente, que pode ser entendida como: quando as condições do antecedente são verdadeiras, então as condições do consequente também são verdadeiras. Ambas as partes consistem em uma conjunção de zero ou mais átomos, não permitindo disjunções ou negação.

Os átomos, por sua vez, são formados por um predicado e um ou mais argumentos. A especificação W3C define seis tipos de átomos [3]: *Class; Object property; Data valued property; Data range; Same/different; Built-in*. Além disso, os átomos se referem a: 1) indivíduos; 2) valores; 3) variáveis para indivíduos; e, 4) variáveis para valores, sendo que variáveis são tratadas como quantificadores universais e possuem o escopo limitado à regra à qual pertencem.

Embora as regras SWRL possam ser representadas em mais de um formato, o formato de leitura humano é adotado neste trabalho. A seta (→) é usada para separar antecedente e consequente, o acento

circunflexo (^) representa a conjunção entre os átomos e o sinal de interrogação (?) distingue as variáveis dos nomes de indivíduos.

2.1 Conjuntos de regras

Para o desenvolvimento deste trabalho foram utilizados dois conjuntos significativos de regras, aplicados em diferentes domínios e com características distintas.

A ontologia *Autism Phenolog Rules* [4] representa um grande e complexo conjunto de regras, estendendo algumas ontologias disponíveis na *Open Biomedical Ontologies* (OBO) *Foundry* e utiliza regras para categorizar fenótipos de autismo. A ontologia possui 156 regras, que correspondem a um total de 2137 átomos. Estes átomos utilizam 155 predicados distintos, sendo que, apenas 9 predicados são utilizados no consequente. O número médio de átomos por regra é 13, a menor regra do conjunto possui 10 átomos e a maior possui 20 átomos.

A ontologia *Family History* [5] define possíveis relações entre membros de uma família e representa um grande conjunto de regras, porém, não muito complexas. A ontologia possui 146 regras, que correspondem a um total de 470 átomos. Estes átomos utilizam 181 predicados distintos, sendo que, a distribuição destes entre o antecedente e o consequente é mais proporcional. O número médio de átomos por regra é 3, a menor regra do conjunto possui 3 átomos e a maior possui 7 átomos.

3. TRABALHOS RELACIONADOS

3.1.1 SWRL Tab [6]

O SWRL Tab é um *plug-in* para o editor de ontologias Protégé, sua principal finalidade é permitir a edição e gerenciamento de regras SWRL.

As regras são apresentadas no formato tabular, sendo possível editá-las diretamente onde são apresentadas ou em um editor a parte. A ferramenta realiza a identificação dos erros sintáticos conforme a especificação SWRL [2], e valida o correto uso dos termos da ontologia. Além disso, um conjunto de APIs é disponibilizado, permitindo, por exemplo, a inferência das regras e realização de testes. Por outro lado, a ferramenta não possui nenhum tipo de recurso relacionado a organização das regras.

3.1.2 Axiomé [3]

O Axiomé também é um *plug-in* para o Protégé e faz uso das APIs disponibilizadas pela ferramenta SWRL Tab. Sob a perspectiva do usuário o Axiomé amplia a quantidade de recursos voltados a organização e visualização das regras, além de disponibilizar a edição das regras por meio de *templates* gerados com base na estrutura das regras.

As principais técnicas que a ferramenta agrega são: Agrupamento automático das regras; Gráficos de visualização e dependências das regras; Paráfrases em regras (uma explicação textual da regra gerada dinamicamente); e *Template* para edição e aquisição de regras.

3.1.3 Regras de negócio

Existe um grande número de técnicas e ferramentas relacionadas à criação de regras de negócio, que podem apontar a direção para o aprimoramento e criação de novas técnicas relacionadas a regras SWRL [7]. Embora os avanços das ferramentas e os estudos relacionados à criação de regras de negócio, ainda não estejam consolidados [1], o estágio atual de seu desenvolvimento está muito à frente se comparado a regras semânticas.

A análise destas ferramentas mostra as distintas formas de interação do usuário com as regras, como por exemplo: tabelas e árvores de decisão, *templates* para edição de regras, linguagem natural, diagramas para visualização e editores integrados que utilizam técnicas de agrupamento, sugestão de termos, detecção e correção de erros.

4. DESENVOLVIMENTO

4.1 Representação lógica das regras

As regras SWRL são armazenadas no formato OWL como parte da própria ontologia em que são criadas. As regras e suas partes, são representadas internamente pela ferramenta Protégé como indivíduos que pertencem a uma ou mais classes e possuem propriedades que os relacionam entre si, como pode ser visto na especificação W3C [2], nas seções *XML Concrete Syntax* e *RDF Concrete Syntax*. Além disso, as ferramentas SWRL Tab e Axiomé são muito acopladas ao Protégé. Com isso, uma API para regras SWRL passou a ser desenvolvida para ser independente de uma ferramenta específica.

Nessa API, as regras são representadas de um modo mais próximo ao modelo lógico utilizado pelos desenvolvedores de regras. Na Figura 1 são apresentadas as principais classes e seus relacionamentos utilizados na representação das regras em SWRL.

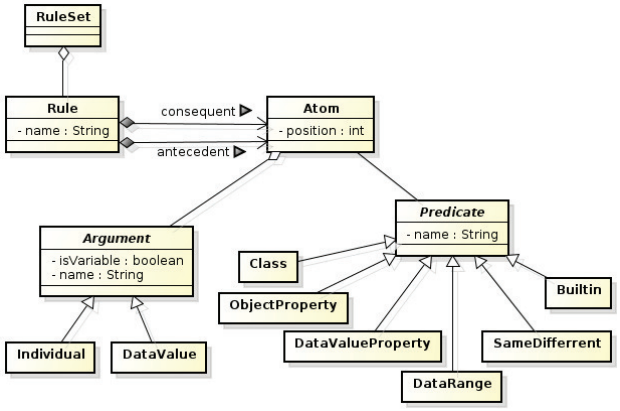

Figura 1 - Principais classes do modelo lógico desenvolvido

A partir deste modelo foi desenvolvido um procedimento automático para extração de informações e características do conjunto de regras que oferece uma visão completa das mesmas. As principais características extraídas foram: quantidade; média; mínimo e máximo; distribuição e frequência referentes aos átomos, tipos de átomo, predicados, argumentos e tipos de argumentos. É possível analisar estas informações observando apenas o antecedente, o consequente, ou ambos.

Além disso, a API possui um módulo de integração do modelo com as APIs do Protégé versões 3 (OWL versão 1) e 4 (OWL versão 2) para obtenção dos conceitos definidos na ontologia e suas regras. Isso permitiu o desenvolvimento de um *plug-in* para esta ferramenta.

4.2 Similaridade entre regras

Mensurar um fator que determina o quanto duas regras são similares ou diferentes se mostrou uma opção capaz de auxiliar na visualização, organização e no processo de composição de regras. Ao organizar as regras em listas, grupos ou categorias o grau de similaridade das regras pode ser considerado, fazendo com que as

regras sejam apresentadas de maneira significativa para os usuários. Na composição, é possível sugerir novos termos baseando-se em regras similares e utilizar a similaridade para identificação de erros, como o de regras duplicadas. Além disso, a maneira como foi desenvolvido o algoritmo de similaridade permite identificar padrões nas regras.

O primeiro passo do algoritmo consiste na elaboração de uma matriz de características, com as linhas correspondendo às regras e as colunas correspondendo aos átomos. Para cara linha/coluna é atribuído o número de ocorrências do átomo na regra.

Essa matriz de características permite algumas variações. Ao invés de utilizar a regra como um todo, pode ser utilizado apenas o antecedente ou consequente. Além disso, os argumentos podem ser descartados utilizando apenas os predicados. A partir da matriz de características obtida, é possível determinar a distância entre duas regras P e Q quaisquer pela expressão:

$$\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

Onde: P = [p_1, p_2, ..., p_n] e Q = [q_1, q_2, ..., q_n] são os vetores de características das regras P e Q, respectivamente.

A fórmula da distância Euclidiana, utilizada para o cálculo da similaridade, fornece a distância entre dois pontos em um plano n-dimensional [8]. Com isso, quando duas regras possuem exatamente os mesmos átomos o resultado obtido pela aplicação da fórmula é zero, e as regras são consideradas idênticas. Desta forma, quanto maior a distância obtida, maior é a diferença entre as regras e consequentemente menor a similaridade entre elas. Para ilustrar a aplicação do algoritmo, considere as regras:

R1: `has_grandparent(?a, ?b) ^ Person_Female(?a) -> has_granddaughter(?b, ?a)`
R2: `has_grandparent(?a, ?b) ^ Person_Male(?a) -> has_grandson(?b, ?a)`
R3: `has_natural_father(?a, ?b) ^ Person_Male(?a) -> has_natural_son(?b, ?a)`

Cujos átomos são:

A1: `has_grandparent(?a, ?b)`
A2: `Person_Female(?a)`
A3: `has_granddaughter(?b, ?a)`
A4: `Person_Male(?a)`
A5: `has_grandson(?b, ?a)`
A6: `has_natural_father(?a, ?b)`
A7: `has_natural_son(?b, ?a)`

A partir deste conjunto de regras e átomos a matriz de característica é representada na Tabela 1.

Tabela 1 - Matriz de características de exemplo

	A1	A2	A3	A4	A5	A6	A7
R1	1	1	1	0	0	0	0
R2	1	0	0	1	1	0	0
R3	0	0	0	1	0	1	1

Aplicando a fórmula ao comparar as regras entre si, é possível determinar, por exemplo, que a regra R1 é mais similar à regra R2 – d(R1,R2) = 2, do que à regra R3 – d(R1,R3) = 2,44. Isso porque R1 e R2 compartilham o átomo `has_grandparent(?a, ?b)` enquanto que R1 e R3 não compartilham nenhum termo.

4.3 Detecção de erros

Ferramentas capazes de identificar e corrigir erros são um dos mais importantes requisitos apontados pelos desenvolvedores de regras [1]. A partir da especificação SWRL, foi definida uma lista com os possíveis erros que os desenvolvedores de regras podem cometer durante o processo de composição de regras. Além disso, é proposta uma classificação de escopo e tipo para os erros. A classificação de escopo compreende: (1) Regra; (2) Átomo; (3) Argumento. A classificação de tipo compreende: (1) Aviso[1]; (2) Erro léxico; (3) Erro sintático; (4) Erro semântico.

A listagem conta ainda com as informações de descrição para correção e mensagem formatada para o usuário. A detecção de erros semânticos não foi alvo desse trabalho. Esse é um dos trabalhos futuros, sugeridos neste artigo.

5. AVALIAÇÃO

Foram realizados alguns testes para avaliar a API juntamente com as técnicas de análise de similaridade e detecção de erros, propostas neste trabalho. Além disso, existe um *plug-in* para edição de regras SWRL para o Web Protégé que faz uso da API, construído em colaboração com *o BMIR Stanford University* e que está em fase final de desenvolvimento.

Além do *plug-in*, foram realizados testes específicos para as técnicas de análise de similaridade e detecção de erros.

5.1 Técnica de Similaridade

Foram conduzidos dois tipos de testes para a avaliação do algoritmo de similaridade proposto neste trabalho.

O primeiro teste consistiu na verificação da similaridade entre todas as regras dos conjuntos apresentados na Seção 2.1. Cada regra foi combinada com todas as demais regras do conjunto e para cada combinação o algoritmo de similaridade foi aplicado.

O segundo teste foi realizado com o objetivo de verificar a eficácia da análise de similaridade entre as regras. Para tanto, os seguintes cenários foram simulados: (1) Regras exatamente iguais; (2) Regras com os mesmos predicados e significados, mas com argumentos nomeados diferentes; (3) Regras com os mesmos predicados, mas com argumentos nomeados diferentes e com significados diferentes; (4) Regras que utilizam 6 predicados comuns, sendo 3 no antecedente e 3 no consequente; (5) Regras que utilizam 6 átomos comuns, sendo 3 no antecedente e 3 no consequente; (6) Regras que utilizam 10 átomos comuns, sendo 5 no antecedente e 5 no consequente; (7) Regras com antecedentes idênticos e consequentes diferentes; (8) Regras com consequentes idênticos e antecedentes diferentes; (9) Regras com átomos diferentes, porém com significados relacionados por meio da ontologia; (10) Regras com átomos diferentes e significados diferentes;

5.2 Detecção de Erros

A avaliação do sistema de detecção de erros ocorreu de duas maneiras. Primeiramente, foi aplicado o algoritmo de detecção de erros para todas as regras dos conjuntos apresentados na Seção 2.1. A finalidade deste teste foi identificar possíveis regras com erros. No segundo, foram simulados todos os possíveis erros na API proposta e também nas ferramentas SWRL Tab e Axiomé.

[1] Os avisos (w*arnings*) não são efetivamente erros, mas foram incluídos na lista pelo fato de sua ocorrência possivelmente não ser desejada ou esperada.

6. RESULTADOS

A API desenvolvida é desacoplada de uma ferramenta específica, abstrai as regras para um nível mais alto e fornece informações relevantes sobre o conjunto de regras, possibilitando ao desenvolvedor ter uma visão geral deste.

A similaridade das regras é utilizada pelo algoritmo de detecção de erros na busca de regras idênticas. Outro cenário, não explorado aqui, mas possível, é o agrupamento automático de regras utilizando o fator de similaridade para apresentar ao desenvolvedor, grupos que possuam regras relacionadas. Além disso, regras similares podem ser utilizadas para sugestão de novos termos durante o processo de composição.

Uma análise dos resultados gerados pela combinação de todas as regras nos conjuntos de regras *Autism Rule Phenolog* e *Family History* evidencia:

- Regras com graus de similaridade menor ou igual a dois (utilizando o método Euclidiano) estão muito próximas e, portanto, podem ser assumidas como regras similares sob a perspectiva do usuário;
- Fatores como amplitude (diferença entre o maior e menor grau), número de graus distintos e moda podem ser utilizados para classificar os conjuntos de regras em homogêneo/heterogêneo e simétrico/assimétrico;
- Enquanto o primeiro conjunto possui um pequeno número de predicados no consequente, o segundo conjunto possui consequentes com apenas um átomo sempre com predicados distintos. Com isso, erroneamente todos os consequentes do primeiro caso são considerados similares e no segundo caso, distintos.

Sob a perspectiva da detecção de erros, foram identificados e organizados os principais erros relacionados com SWRL. Todas as regras da ontologia *Autism Rule Phenolog* apresentaram um erro sintático relacionado aos argumentos. É muito provável que este erro não tenha causado problemas para a interpretação da regra, uma vez que no antecedente as variáveis são do tipo individuo e no consequente a mesma variável assume um tipo de dado primitivo, ou vice versa. Além disso, foram identificadas duas regras repetidas, porém com nomes diferentes.

Ao comparar o sistema de identificação de erros das ferramentas SWRL Tab e Axiomé com a API desenvolvida, constatou-se que as duas primeiras permitem que os usuários cometam alguns equívocos, por outro lado, a identificação dos principais erros léxicos e sintáticos está disponível nestas ferramentas. Por fim, nenhuma delas possui suporte a detecção de erros semânticos.

Para agregar recursos semânticos às técnicas propostas, é necessário analisar as ontologias associadas as regras considerando, por exemplo, a hierarquia das classes e características das propriedades como transitividade, propriedades inversas, domínios e faixa de valores.

7. CONSIDERAÇÕES FINAIS

Este trabalho apresentou duas técnicas para aprimorar ferramentas de regras SWRL que permitem analisar a similaridade entre regras e detectar erros léxicos e sintáticos. Enquanto a primeira pode ser utilizada para apoiar a visualização e a composição de regras, a segunda é mais apropriada para o processo de composição. Tais técnicas foram integradas em uma API que oferece um conjunto de recursos para gerenciamento e manipulação de regras SWRL.

A estratégia usada para obter o grau de similaridade entre duas regras pode utilizar qualquer métrica para o cálculo de distância entre regras, como a Euclidiana utilizada neste trabalho. Nos testes iniciais, foi possível observar que a abordagem é adequada para uma grande quantidade de situações. Em uma das avaliações realizadas, a abordagem teve resultados satisfatórios em 4/10 dos cenários e parcialmente em 3/10 dos cenários propostos.

Quanto à detecção de erros, foram organizados os principais erros relacionados à linguagem SWRL. O algoritmo desenvolvido identifica três tipos de erros: avisos, léxicos e sintáticos. A avaliação indica que a abordagem conseguiu detectar uma grande quantidade de erros.

Encontra-se em estágio avançado o desenvolvimento de um *plug-in* para a ferramenta Web Protégé, que faz uso da API desenvolvida juntamente com as técnicas de análise de similaridade e detecção de erros, e que permitirá o gerenciamento, visualização e composição de regras SWRL. A partir desta ferramenta, novas técnicas e funcionalidades que fazem o uso das que foram propostas neste trabalho serão desenvolvidas.

Como trabalhos futuros, serão desenvolvidas melhorias nas duas técnicas agregando a estas recursos semânticos. Por fim, espera-se desenvolver técnicas para agrupamento automático de regras com base na similaridade, gerando grupos de regras significativos sob a perspectiva do usuário.

8. AGRADECIMENTOS

Esse trabalho contou com o financiamento do CNPq.

9. REFERÊNCIAS

[1] Zacharias, V. 2008. Development and verification of rule based systems – a survey of developers. In *Rule Representation, Interchange and Reasoning on the Web: Int'l Symposium*, 6-16.

[2] W3C. 2004. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. http://www.w3.org/Submission/SWRL/.

[3] Hassanpour, S., O'Connor, M. J. and Das, A. K. 2009. Exploration of SWRL Rule Bases through Visualization, Paraphrasing, and Categorization of Rules, RuleML 2009, 246–261.

[4] Tu, S., Tennakoon, L., O'Connor, M. J., Shankar, R. and Das, A. K. 2008. Using an integrated ontology and information model for querying and reasoning about phenotypes: the case of autism. In *Proc. of the American Medical Informatics Association*, 727–731.

[5] Peace, J. and Brennan, P. F. 2008. Instance testing of the family history ontology. In *Proc. of the American Medical Informatics Association (AMIA) Annual Symposium*.

[6] SWRLTab. 2011. SWRL Tab. http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTab

[7] Rivolli, A., Orlando, J. P. and Moreira, D. A. 2011. An Analysis of Rules-Based Systems to Improve SWRL Tools. ICEIS 2011, Beijing, China, *in press*.

[8] Salzberg, S. 1991. Distance Metrics for Instance-Based Learning, In *Proc. of ISMIS'916th International Symposium*, Methodologies for Intelligent Systems, 399-408.