Construção de um Processador de Linguagem Natural

Iohan Gonçalves Vargas[1], **Vaston Gonçalves da Costa**[1]

[1]Universidade Federal de Goiás - Campus Avançado de Catalão (UFG - CAC)
Catalão – GO – Brasil – Departamento de Ciência da Computação

{iohanufg,vaston}@gmail.com

Abstract. *Natural language processors emerge as keys tools for formal verification of systems. The use of this type of translator supplies a lack in proper validation formulas and their application cost is reasonable. Through a Natural Language Processor, which helps a user to process a text in natural language into logical formulas in DL, whose syntax is formal, tough learning and mastery, can construct a knowledge base and then make inferences about the information in order to explore the database through a logical reasoner. Thus, this paper presents how one can use translators with logical reasoners in the presentation of knowledge in a specific domain and generate and explain structured on the same evidence.*

Resumo. *Processadores de linguagem natural surgem como principais ferramentas para verificação formal de sistemas. A utilização desse tipo de tradutor supre a carência na validação correta de fórmulas e seu custo de aplicação é acessível. Dado um Processador de Linguagem Natural, que auxilia a processar um texto, em linguagem natural, para fórmulas lógicas em DL (Description Logic), cuja sintaxe é formal, de difícil aprendizado e domínio, pode-se construir uma base de conhecimento e, posteriormente, realizar inferências sobre as informações. Assim, este trabalho visa apresentar como se podem utilizar tradutores lógicos na representação do conhecimento em um domínio específico de forma que possibilite raciocinadores tomarem decisões.*

1. Introdução

Com os avanços teóricos, representação do conhecimento surge como um dos conceitos centrais e importantes em Inteligência Artificial (IA). Segundo Davis [1993], é um ambiente pelo qual se modela dados, de forma a orientar o raciocínio sobre um determinado domínio, objetivando facilitar a tomada de decisões, a partir de inferências, tendo como base a representatividade real. A definição de *Description Logic* - DL (Lógica Descritiva) tem sido amplamente utilizada em Bases de Conhecimento, é um conceito capaz de compreender diferentes formas, na qual pode se representar o conhecimento e obter conclusões. Em meados da década de 70, surgiu o conceito de DL, que se define basicamente em um nome, que se refere a qualquer uma das várias linguagens lógicas comumente utilizadas em Representação do Conhecimento. Assim, define-se um conjunto de conceitos, tidos como afirmações individuais (ABOX) e um conjunto de relações binárias definidas sobre os mesmos (TBOX) [Davis *et al.*, 1993].

Representação do Conhecimento é o estudo do pensamento como um processo computacional, o uso de DL e seus mecanismos de inferências são propícios para este

trabalho, e deseja-se usá-los para representar e explicar bases de conhecimentos em domínios específicos. No desenvolvimento deste trabalho, são apresentados os estudos realizados na utilização de DL na formalização da semântica de textos em linguagem natural, para, posteriormente, empregar-se provadores de teoremas para raciocinar sobre a informação contida nos textos originais.

2. Fundamentação Teórica

Processamento de linguagem natural (PLN) é uma sub-área da inteligência artificial e da linguística que estuda os problemas da geração e compreensão automática de línguas humanas naturais, no qual através deste trabalho, estabelece-se uma relação com a lógica descritiva, para representação formal de conhecimento.

Sistemas de geração de linguagem natural convertem informação de base de dados de computadores em linguagem normalmente compreensível ao ser humano, de forma lógica e coerente, contudo, convertem ocorrências de linguagem humana em representações mais formais, mais facilmente manipuláveis por programas de computador, ou como mencionado no título deste trabalho, processador de linguagem natural. Processamento de linguagem natural é um método atrativo para interação homem-máquina, que pode ser aplicado a problemas de maiores níveis de ambiguidade e complexidade das inserções lógicas. Com base nesta afirmação, pode-se relatar que o uso de uma linguagem lógica para formalização de um texto normativo é indispensável e importante em todo o processo, bem como, no auxílio da construção da base de conhecimento.

Precisamos inicialmente estudar o que significa compreender, conceito-base de PLN. A compreensão é o ato de transformar uma forma de representação em outra, que seja significativa para o ambiente em questão e que permita o mapeamento para um conjunto de ações apropriadas, tanto para fins de armazenamento da informação, quanto para a tomada de algum tipo de decisão.

Para nosso trabalho, *Description Logic* - DL foi utilizada, tem sido amplamente utilizada em Bases de Conhecimento, emergindo como um conceito capaz de compreender e representar o conhecimento. DL não é um conceito novo, os primeiros estudos foram em meados da década de 70, cuja época, foi ponto marcante para estudiosos da área, pois surgiu o conceito de DL, que se define basicamente em um nome, que se refere a qualquer uma das várias linguagens lógicas, comumente utilizadas em Representação do Conhecimento [Davis *et al.*, 1993].

Portanto, DL são conjuntos de formalismos de representação do conhecimento de um domínio. Na Figura 1, é mostrado a estrutura de um sistema em DL, primeiramente, define-se os conceitos relevantes ao domínio, e utilizam estes conceitos para especificar as propriedades de objetos e indivíduos do domínio, criando a descrição do domínio. Uma lógica descritiva é formada por uma linguagem descritiva, que é utilizada para definir como os conceitos e como os papéis são formados.

3. Estado da Arte

Como mencionado anteriormente, processamento de linguagem natural é um método atrativo para interação homem-máquina, sistemas mais antigos como SHRDLU, trabalhava com *'blocks worlds"* restritos, com vocabulários restritos, levando pesquisadores a um excessivo otimismo, que mais tarde foi superado quando o sistema foi aplicado a problemas

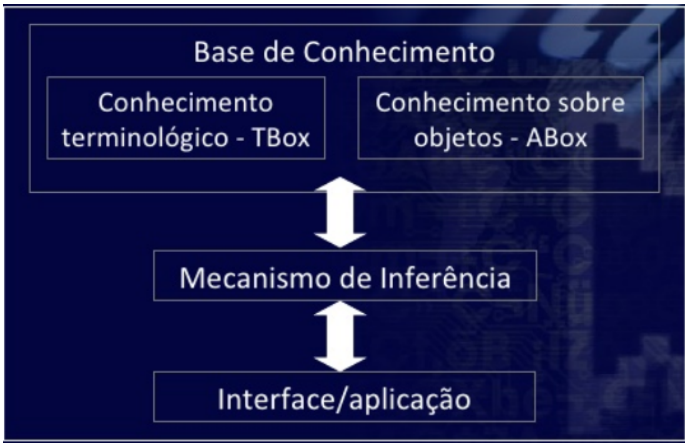

Figura 1. Estrutura de um sistema em DL

mais realistas, envolvendo ambiguidade e complexidade. No que se trata de processadores de linguagem natural, nosso trabalho se destaca e apresenta inovação, com utilização de DL e Ontologia OWL, que será detalhada nas páginas seguintes.

4. Desenvolvimento

O processador de linguagem natural consiste em representar o conhecimento de um texto em linguagem descritiva (DL) , assim estabelecendo um relacionamento com a semântica do domínio pré-definido. Dessa forma, a área de Processamento de Linguagem Natural (PLN) possui ligações significativas com o desenvolvimento deste trabalho, assim como a definição de ontologias, que são usadas para capturar conhecimento sobre um domínio de interesse, descrevendo conceitos e relações do domínio. Um dos mais importantes conceitos em ontologias, é a Ontologia OWL *(Web Ontology Language)*, este é um padrão definido mais recentemente e monitorado pela W3C *(World Wide Web Consortium)*.

Sabendo-se que, para muitas organizações a formalização de textos baseados em informação é muito importante, é comum encontrar documentos de texto, para serem avaliados por um agente, para processamento e tomada de decisões. Assim sendo, apresenta-se um exemplo de formalização de textos normativos no domínio da Segurança da Informação (SI) e de extração de conhecimento destes textos de linguagem natural. Para isso, primeiro definem-se os conceitos relevantes de um domínio/terminologia e então, usando estes conceitos, especificam-se as propriedades dos objetos e indivíduos deste domínio. Porém, é necessário que haja uma validação dos controles de segurança, assim, a terminologia relacionada a SI, de forma breve, é: controle de segurança, políticas de segurança e padrões de segurança, ambos são formalizados como conceitos na ontologia.

Neste sentido, queremos verificar se um controle de segurança é a implementação de uma ação específica do ponto de vista lógico. Para tanto, se o Controle01 possuir a seguinte descrição da política de segurança:

O tráfico de rede para a administração remota do servidor Netware deve ser criptografado usando SSL. E se a ação ‘Configurar todo o sistema para criptografar conexões usadas para o acesso remoto ao sistema”fizer parte de uma política de segurança de uma organização, nomeada como Ação02. Pode-se inferir que Controle01 de fato implementa

Ação02. Isto é feito provando que o conceito representando o controle é subsumido pelo conceito representando a ação, isto é, que a fórmula DL Controle 01 $\subseteq Ação02 é provável$.

Para formalizar as declarações ou as afirmações Controle01 e Ação02 na ontologia em formas lógicas, as seguintes considerações são feitas: frases na voz ativa ou passiva, no modo declarativo ou imperativo, e sentenças que contém os mesmos termos relacionados, devem se resumir a uma mesma forma lógica. Observe suas respectivas representações formais:

Controle01 $\equiv$
1. $\ni Verbo.(Criptografar \cap$
2. $\ni Tema.TráficoDeRede \cap$
3. $\ni Instrumento.SSL \cap$
4. $\ni Objetivo.(AdministraçãoRemota \cap$
5. $\ni Tema.ServidorNetware))$

$Ação02 \equiv$
1. $\ni Verbo.(Configurar \cap$
2. $\ni Tema.Sistema \cap$
3. $\ni Objetivo.(Criptografar \cap$
4. $\ni Tema.(ConexãoDeRede \cap$
5. $\ni éInstrumentoDe.(AcessoRemoto \cap$
6. $\ni Tema.Sistema))))$

No qual o Verbo é uma propriedade fornecida a fim de relacionar o conceito de ação com os conceitos verbais apropriados; Tema é uma propriedade específica para representar o tema do verbo; Objetivo e Instrumento são outras propriedades que representam papéis temáticos na ontologia; e éInstrumentoDe é uma propriedade inversa de Instrumento. Observe que a sentença de controle foi convertida para a voz ativa, e esta atitude é tida como um artefato desejável de formalização. O exemplo acima, de formalização, foi retirado de [Amaral *et al.*, 2006]. A prova, pode ser obtida em [Rademaker *et al.*, 2007], de que Controle01 $\subseteq Ação02$.

Para seguir os objetivos específicos deste projeto visando a obtenção de resultados palpáveis e bem concretizados teoricamente, a ferramenta Protege OWL foi utilizada para análise da estrutura de DL, no que se refere ao uso prático da mesma.

A ferramenta Protege OWL é dividida em 3 categorias principais: Indivíduos (*Individuals*), propriedades (*Properties*) e classes (*Classes*), no qual é possível definir os indivíduos do domínio, as propriedades às quais estes indivíduos pertencem e fazem relação com um ou mais indivíduos, e por fim é determinada as classes que envolvem o domínio, estabelecendo uma estrutura hierárquica do sistema e evitando inconsistência dos dados envolvidos. A Figura 2, tem-se um exemplo de hierarquia de um sistema de Pizza, demonstrando a usabilidade da ferramenta Protege e a contribuição para o desenvolvimento do protótipo de processador de linguagem natural.

Contudo, podemos afirmar que é baseado em um modelo lógico que torna possível definir os conceitos de forma como são escritos. Possibilitando definir conceitos mais complexos a partir de conceitos simples. Para validação e evitar inconsistência na estru-

Figura 2. Hierarquia de Classes

tura hierárquica da ontologia desenvolvida.

Ontologia OWL é classificada em 3 espécies:

1. OWL Lite (sintaticamente mais simples)
2. OWL DL (baseia-se em Lógica Descritiva, passível de raciocínio Lógico)
3. OWL Full (Destinada a situações com alta expressividade, não é possível efetuar inferências)

Com o intuito de desenvolver o protótipo de um processador de linguagem natural foi feito um estudo das diversas linguagens de programação existente, a priori a linguagem de programação Python foi objeto de estudo deste trabalho. Através do *framework* EasyGUI, disponibilizado gratuitamente na internet, é possível criar ambientes gráficos na linguagem Python, porém com o intuito de encontrar a linguagem de programação mais apropriada para desenvolver o protótipo do processador de linguagem natural, este *framework* foi apenas estudado e ponderado seu pontos positivos.

Como forma de ampliar os conhecimentos em linguagens de programação e diversas ferramentas de desenvolvimento gráfico, a ferramenta QTCreator foi estudada e colocada em prática durante o desenvolvimento do protótipo. Disponibilizada gratuitamente na internet, permite o desenvolvimento mais rápido do sistema quando comparado com o *framework* EasyGUI. Porém, apresenta maior incompatibilidade ao gerar o código fonte em Python da interface gráfica previamente desenvolvida, assim a ferramenta QTCreator foi apenas objeto de estudo/análise.

A linguagem JAVA, foi tida como apropriada para o desenvolvimento do protótipo, tal afirmação se baseia fortemente no conhecimento prévio adquirido durante o curso de graduação de Ciências da Computação na Universidade Federal de Goiás, a IDE (*Integrated Development Environment*) NetBeans na versão 7.2 foi um instrumento de trabalho para utilização linguagem JAVA. Na Figura 3, tem-se a representação da interface do Tradutor desenvolvido.

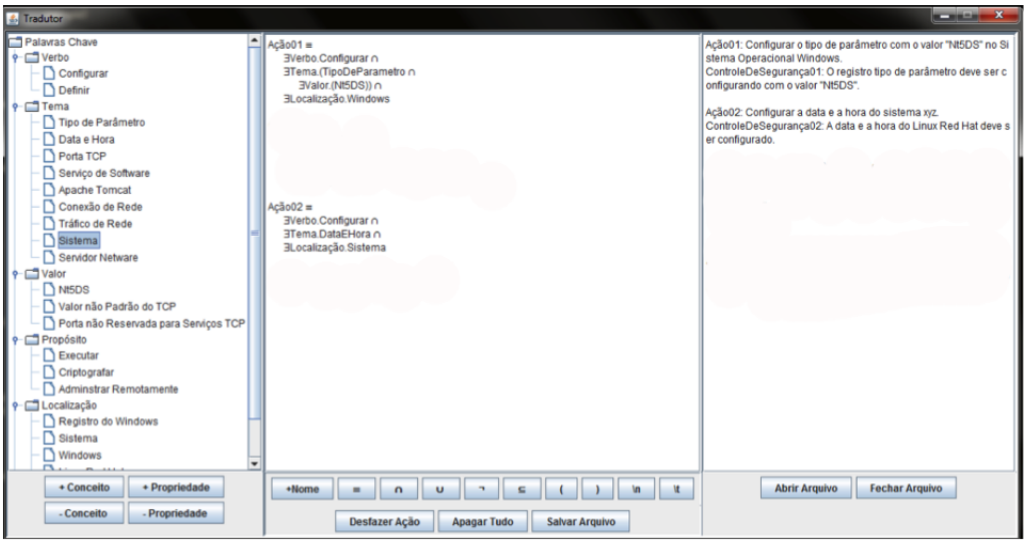

Figura 3. Protótipo do Processador em JAVA

4.1. Funcionalidades

O Processador de Linguagem Natural é dividido em 3 partes, a Figura 4 refere-se a Parte 1 do processador de linguagem natural, como mostrado abaixo:

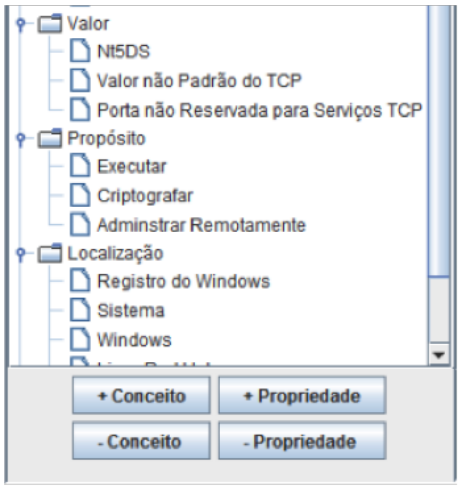

Figura 4. Parte 1 do Protótipo

Na Parte 1 do sistema, é possível adicionar com base no domínio de interesse, Conceitos e Propriedades. Temos: o Conceito Valor, para adicionar um Conceito basta clicar no botão +Conceito, e as propriedades que este conceito apresenta no domínio são, Nt5DS, Valor não Padrão de TCP, Porta não Reservada para Serviços TCP, para adicionar propriedades basta clicar no botão +Propriedade. A Parte 2 do sistema permite ao usuário inserir no processador de linguagem natural o problema descrito, salvo em arquivo texto. Para isto, basta clicar no botão Abrir Arquivo (Figura 4) e em seguida localizar no HD do computador o arquivo texto que contém a descrição do problema. A Parte 3, é destinada a formalização em DL, do problema inserido anteriormente, existe uma barra de símbolos lógicos na parte inferior da tela, tais como: +Nome (para adicionar nome ao problema), = (igualdade), $\cap(interseção), \cup(união), \neg(negação), \subset (estácontido)$.

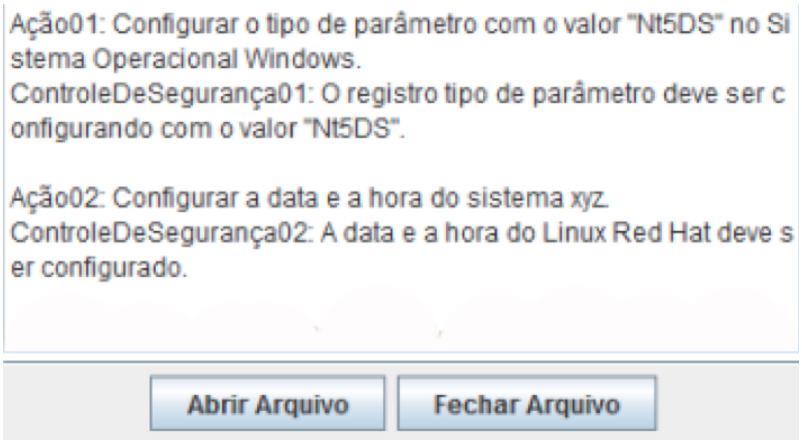

Figura 5. Parte 2 do Protótipo

Figura 6. Parte 3 do Protótipo

5. Resultados

As bases de desenvolvimento da área de Inteligência Artificial se encontram principalmente na noção teórica de "máquina de Turing' e na ideia de que "Pensar é computar', proposta pelo matemático, lógico e cientista da Computação Alan Turing. Os estudos de Turing contribuíram para o desenvolvimento da parte da Lógica relacionada com a análise simbólica do raciocínio [Tassinari, 2011].

Tendo-se um sistema de Base de Conhecimento, que armazena informações que descrevem o domínio através do formalismo de uma lógica, podem-se utilizar algoritmos de inferência que permitam a extração de conhecimentos implícitos e não percebidos pelo ser humano. Os raciocinadores lógicos podem incorporar mecanismos de inferência e utilizar diversos algoritmos, e são eles que garantem que o novo conceito incorporado à base de conhecimento, faça sentido e não cause nenhuma contradição nos conceitos já definidos no domínio [Mertins, 2011]. Dessa forma, processador de linguagem natural pode ser utilizado no processo de validação e garantir que elas não afetem de forma negativa, produzindo inconsistência e ou contradições na base de conhecimento.

Portanto, o uso de uma linguagem lógica para formalização de um texto normativo é indispensável, e nada mais justo que a utilização das lógicas de descrição que, em contraste com outros sistemas de representação, são equipadas com uma semântica formal e bem definida. Assim sendo, um Processador de Linguagem Natural se mostra, na prática, importante em todo esse processo e no auxílio da construção da base de conhecimento.

6. Conclusões

A partir da lógica descritiva, pode-se modelar o raciocínio humano, partindo-se de frases declarativas (ou proposições), caracterizando-se como o elo na automatização do processamento da informação. Na área de IA, por exemplo, a representação formal garante o correto raciocínio, expressando-se, assim, o alto poder de usabilidade de um raciocinador lógico. Logo, é de grande importância o estudo da lógica, pois o cérebro humano ter seu comportamento regido por relações lógicas. A lógica formal, mantém o rigor matemático necessário para modelar situações e analisá-las formalmente, desde então é aplicada e utilizada na validação sintática e semântica de softwares, na geração e otimização de códigos e na área de segurança em sistemas de informação.

Em suma, a lógica é uma área de grande êxito e se faz necessário a implementação de estratégias mais sofisticadas que envolvam raciocinadores, cujo objetivo é gerar formalização lógica em diferentes domínios de conhecimento. Há a interação com pesquisadores e alunos de pós-graduação do Laboratório de Técnicas e Métodos Formais (TecMF) do Departamento de Informática da PUC-Rio, que pesquisam tópicos relacionados a Representação do Conhecimento e Lógica de Descrição (DL), contando com a interação interinstitucional, que contribui para uma maior sinergia nos resultados.

Contudo, foi construído um Processador de Linguagem Natural, que mesmo classificado como protótipo, caracteriza-se como um refinamento do estudo sobre o tema, sendo possível obter resultados que, embora pareçam simples, são considerados um salto no estudo da automatização do conhecimento e do raciocínio lógico.

Referências

Davis, Randall. Shrobe, Howard. Szolovits Peter. (1993). *What is a Knowledge Representation?* AI Magazine, Volume 14, Number I.

Amaral, F. N., Bazílio, C., Silva, G. M. H., Rademaker, A., Haeusler, E. H. (2006). *Ontology-based Approach to the Formalization of Information Security Policies*. Em: 10th IEEE *International Enterprise Distributed Object Computing Conference Workshops* (EDOCW06).

Rademaker, A. R., Amaral, F. N., Haeusler, E. H. (2004). *A Sequent Calculus for ALC*. Monografias em Ciência da Computação. Em: 25/07. Pontifícia Universidade Católica do Rio de Janeiro. Setembro.

Tassinari, R. P., Gutierre, J. H. B. (2011). Lógica como Cálculo Raciocinador. São Paulo, 2011

Mertins, L. E. (2011). Extensão Virtual do Mundo Real: Integração Semântica e Inferência. Dissertação de Mestrado em Ciência da Computação. Universidade Católica de Pelotas. Pelotas.