UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA DE PRODUÇÃO

ANDRE BORTOLON

UM MODELO PARA A EXTRAÇÃO DE CONCEITOS E ESTABELECIMENTO DE CONTEXTOS EM SISTEMAS BASEADOS EM CONHECIMENTO

Tese de Doutorado

Florianópolis
2006

ANDRE BORTOLON

UM MODELO PARA A EXTRAÇÃO DE CONCEITOS E ESTABELECIMENTO DE CONTEXTOS EM SISTEMAS BASEADOS EM CONHECIMENTO

Tese apresentada ao
Programa de Pós-Graduação
em Engenharia de Produção
da Universidade Federal de
Santa Catarina como
requisito para obtenção do
grau de Doutor em
Engenharia de Produção

Florianópolis
2006

Andre Bortolon

Um Modelo para a Extração de Conceitos e Estabelecimento de Contextos em Sistemas Baseados em Conhecimento

Esta tese foi julgada e aprovada para a obtenção do grau de **Doutor em Engenharia de Produção** no **Programa de Pós-Graduação em Engenharia de Produção** da Universidade Federal de Santa Catarina

Florianópolis, 24 de fevereiro de 2006.

Prof. Édson Pacheco Paladini,
Coordenador do Programa de Pós Graduação em Engenharia de Produção

Banca Examinadora:

Hugo Cesar Hoeschl, Post Doc, Orientador	Carlos Augusto M. Remor, Dr.
Christianne Coelho de Souza Reinisch Coelho, Dra.	Paulo de Tarso Mendes Luna, Dr.
Tânia Cristina D'Agostini Bueno, Dra.	

Agradecimentos

A Deus

À minha família

Às equipes da WBSA Sistemas Inteligentes S.A. e do Ijuris, pela ajuda nos trabalhos

Ao prof. Hugo Cesar Hoeschl, pela orientação

À profa. Christianne Coelho, pela ajuda na estruturação do trabalho

Aos prof. Carlos Augusto Remor, Paulo de Tarso Mendes Luna e Tania Cristina D'Agostini Bueno, pela atenção dispendida na leitura e avaliação do trabalho

À Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior (CAPES) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela bolsa de pesquisa concedida durante o curso

À Kátia, pela compreensão e pelo carinho

Resumo

Sistemas de Recuperação de Informação normalmente trabalham com tecnologias baseadas em palavras-chave. Embora, tais sistemas atinjam resultados satisfatórios, eles não são aptos a responder consultas mais complexas elaboradas por usuários. Para isto, existem os Sistemas Baseados em Conhecimento, os quais utilizam-se de ontologias para a representação do conhecimento embutido nos textos. As técnicas mais avançadas de construção de ontologias atualmente baseiam-se na participação de três atores: o engenheiro de conhecimento, o especialista do domínio e o analista de sistemas. Este trabalho dispende tempo, haja vista os numerosos estudos que devem ser feitos para determinar quais elementos devem participar da base de conhecimento e como eles se inter-relacionam. Desta forma, utilizar sistemas computacionais que, ao menos, agilizem este trabalho é fundamental para a criação de sistemas para o mercado. Este trabalho apresenta um modelo que permite que a representação do conhecimento seja feita diretamente pelo computador, necessitando de intervenção mínima, ou até nenhuma, do usuário humano, ampliando a abrangência de domínios que um sistema pode manter, tornando-o mais eficiente e de fácil utilização.

Palavras-chave: Inteligência Artificial, Recuperação de Informação, Sistemas Baseados em Conhecimento.

Abstract

Information Retrieval Systems normally deal with keyword-based technologies. Although those systems reach satisfactory results, they aren't able to answer more complex queries done by users. To do that, there are the Knowledge-Based Systems, which use ontologies to represent the knowledge embedded in texts. Currently, the most advanced techniques to build ontologies are based on the participation of three components: the knowledge engineer, the domain specialist, and the system analyst. This work demands time due to the various studies that should be made do determine which elements must participate of the knowledge base and how these elements are interrelated. In this way, using computational systems that, at least, accelerate this work is fundamental to create systems to the market. A model that allows a computer directly represents the knowledge, just needing a minimal human intervention, or even no one, enlarges the range of domains a system can maintain, becoming it more efficient and user-friendly.

Keywords: Artificial Intelligence, Information Retrieval, Knowledge-Based Systems.

SUMÁRIO

1 INTRODUÇÃO

1.1 Contextualização

Pesquisadores afirmam que atualmente vive-se a era do excesso de informações. Soares et. al. (2004) calculam que a Internet possuía, em junho de 2003, aproximadamente 8,25 bilhões de páginas. Segundo o mesmo trabalho, o Google, considerado o maior mecanismo de busca da Internet, possuia indexado apenas 37% deste conteúdo (aproximadamente 3,1 bilhões de documentos). Arnold (2005) possui dados mais recentes, informando que, no início de 2005, o Google já possuía mais de 8 bilhões de páginas indexadas em sua base de dados. Seguindo os cálculos de Soares et. al. (2004) pode-se estimar que a Internet pública e indexável possua em torno de 21,6 bilhões de páginas. Considere-se também todo o conteúdo disponível em sítios fechados ou base de dados que não podem ser indexadas, não é exagero afirmar que o conteúdo da Internet pode chegar aos 100 bilhões de páginas.

Nestas páginas pode-se encontrar todo o tipo de conteúdo, com opiniões diversas e antagônicas sobre o mesmo assunto. Então, como pode-se determinar qual é a verdadeira informação, ou qual documento contém algo realmente relevante ao assunto que estamos consultando? As ferramentas atuais baseiam-se, na sua grande maioria, em busca por palavras-chave e conectores lógicos. Mas esta não é uma interface natural. Afinal, quando queremos obter alguma informação em uma biblioteca, por exemplo, nós fazemos uma pergunta a um bibliotecário e ele nos oferece com corretude quais são os livros lá existentes que correspondem ao assunto. Da mesma forma, se fizermos qualquer pergunta a um especialista humano, pergunta esta sobre o assunto dominado por ele, ele vai nos dar a resposta sem que nós precisemos consultar livros ou material extra sobre o assunto. Apenas sua resposta já é suficiente para resolver nossa dúvida.

Com o volume de informação que se pressupõe existente na Internet, é impossível para uma pessoa organizá-lo e armazená-lo para que ela possa responder qualquer pergunta sobre qualquer assunto. Para que este processo

seja informatizado, torna-se necessário que os computadores entendam o conteúdo dos documentos e possam responder adequadamente às consultas feitas através dos softwares em linguagem natural.

A organização e o armazenamento das informações é um problema que vem sendo abordado de muitas maneiras na história da tecnologia da informação. Os SGBDs (Sistemas Gerenciadores de Banco de Dados) são as formas mais comuns de armazenamento de dados encontradas atualmente. Entretanto, eles trabalham com a chamada informação estruturada, ou seja, o conteúdo a ser inserido nele precisa ser previamente disposto em uma estrutura para que eles sejam capazes de fornecer as informações necessárias quando requisitado.

As informações são armazenadas na forma de tabelas, onde cada linha da tabela é um registro de dados novo e as colunas definem quais são os dados a ser guardados. Cada célula possui um dado que pode ser uma relação com outra tabela, mas seu conteúdo só pode ser acessado inteiramente. Isto implica que estes sistemas não forneçam interfaces em linguagem natural para que sejam feitas as pesquisas. Eles utilizam linguagens algébricas e altamente rígidas. A ausência de um parâmetro, por exemplo, pode implicar no não funcionamento da consulta.

Uma tecnologia que tenta aliar a capacidade de manipulação de documentos não estruturados aos SGBDs são os bancos de dados textuais. Eles possuem a estrutura de funcionamento dos SGBDs tradicionais, mas com ajustes visando maximizar a performance da manipulação de textos. Entretanto, eles padecem do mesmo problema que os SGBDs tradicionais, que é o fato da linguagem de procura ser rígida, prejudicando a recuperação em alguns casos.

Para os documentos não-estruturados, a tecnologia tradicionalmente aplicada para sua manipulação é a Recuperação de Informação (RIJSBERGEN, 1979; SALTON e MCGILL, 1983). Embora ela não tenha a intenção de recuperar a informação propriamente dita, no sentido de modificar o conhecimento do usuário sobre o assunto solicitado, ela consegue fazer a recuperação dos documentos que possuem o conteúdo solicitado pelo usuário

na consulta. RI permite ao usuário realizar buscas em linguagem natural, digitando termos e procurando por estes nos documentos. Normalmente, os mecanismos de busca oferecem opções baseadas em operadores lógicos, permitindo que o usuário procure por um termo mais outro, um termo menos outro, etc.

Outro problema que os sistemas de Recuperação de Informação enfrentam e tentam resolver é a identificação da relevância das palavras que aparecem em um documento. Afinal, nem todas as palavras escritas num documento são para representar o assunto tratado. Praticamente a maioria são palavras de ligação e de composição de um texto. Desta forma, mecanismos sofisticados de identificação da relevância das palavras são vitais para que os mecanismos possam classificar adequadamente os documentos no retorno para o usuário. Além disto, eles tentam encontrar outras maneiras de hierarquizar os documentos, a fim de evitar que apenas as palavras sejam determinantes do resultado. Um dos critérios mais conhecidos atualmente é o *fan-in/fan-out*, no qual o número de referências para uma determinada página indica que aquela página é mais confiável, dando a ela uma preferência. O Google, por exemplo, utiliza este como um de seus critérios para a recuperação.

Mas, os sistemas de Recuperação de Informação possuem dificuldade para retornar documentos quando o usuário consulta por um assunto, ao invés de utilizar as palavras que o representam. Para isto, desenvolveu-se a Representação do Conhecimento, visando a utilização de técnicas para a relacionar as palavras de forma a permitir que os documentos sejam classificados por assuntos.

A técnica mais usada para a Representação do Conhecimento são as redes semânticas. Nelas, os elementos são relacionados entre si e as ligações entre eles mostram o significado do relacionamento entre os elementos.

Raciocínio Baseado em Casos (RBC) também é uma tecnologia muito utilizada para a recuperação de informações, sejam elas estruturadas ou não-estruturadas. A maioria dos autores concorda que o RBC é um método de raciocínio baseado na proposta de utilizar experiências passadas encapsuladas

em estruturas de dados como a base para lidar com novas situações similares. A abordagem parece ser intuitiva: quando uma nova situação acontece, deve-se tentar alguma coisa que já foi utilizada com sucesso. O RBC trabalha com a estruturação dos documentos na forma de casos, extraindo os principais atributos de um caso e disponibilizando-os para a entrada do usuário. O processo de funcionamento do RBC é demonstrado na Figura 1, denominada ciclo do RBC (AAMODT e PLAZA, 1994).

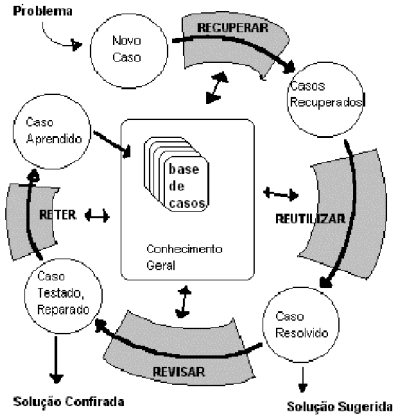


Figura 1. Ciclo do RBC

O ciclo divide-se em quatro passos:

1. Recuperação: O problema apresentado pelo usuário é transformado em um novo caso, o qual é comparado com todos os casos da base para que sejam recuperados os mais similares;
2. Reutilização: O conjunto de casos recuperados é analisado e um deles é sugerido como sendo a solução para o problema apresentado pelo usuário;
3. Revisão: A solução sugerida é testada e, caso necessário, modificada para tornar-se a solução do problema;
4. Retenção: o novo caso, juntamente com sua solução revisada, é inserido na base de casos para sua utilização futura.

Existe uma subcategoria de RBC que visa a manipulação de casos que contenham textos como um de seus atributos. Chamada RBC Textual (LENZ e BURKHARD, 1997; LENZ, 1998), esta técnica trabalha com a extração de

atributos do texto para auxiliar na representação do mesmo. Assim, além de texto livre, o usuário pode utilizar um destes atributos para a recuperação. Além disto, como os sistemas de RBC são baseados em conhecimento extraído dos casos, este conhecimento precisa ser ampliado para compreender o texto existente nos casos. Isto permite uma expansão do texto, permitindo a classificação destes por assuntos e a melhoria da recuperação.

Exemplos de aplicações utilizando RBC Textual são o FAQFinder (BURKE et. al., 1997), o CBR-Answers (LENZ et. al., 1998), o Jurisconsulto (BUENO et. al., 1999) e o AlphaThemis (BUENO et. al., 2003).

O objetivo do sistema FAQFinder é responder perguntas formuladas em linguagem natural (em inglês) através da recuperação de arquivos FAQ dos *newsgroups* da USENET. O FAQFinder integra uma combinação de técnicas de Raciocínio Baseado em Casos com técnicas de Recuperação de informação e Processamento de Linguagem Natural para a recuperação de documentos. O FAQFinder utiliza um tesauro como base de raciocínio em uma base de conhecimento semântico. Ele é projetado como um sistema genérico para recuperar FAQs sem focar em algum domínio específico. Para isto, ele executa um processo em dois passos. No primeiro, ele realiza uma análise superficial sobre as palavras-chave contidas na consulta para determinar quais são os *newsgroups* que possuem uma maior relação com ela. Em seguida, ele realiza uma análise mais aprofundada nestes *newsgroups* selecionados para determinar quais são as perguntas que serão retornadas ao usuário como resposta.

O CBR-Answers utiliza uma abordagem semelhante à do FAQFinder. O diferencial consiste em que o CBR-Answers enfoca especificamente os domínios técnicos. Baseado em técnicas de engenharia de conhecimento, o conhecimento específico do domínio é inserido no sistema e é empregado para a avaliação da similaridade. Ele também utiliza a Wordnet como base de raciocínio. Para recuperar os documentos, ele utiliza as palavras-chave significativas específicas ao domínio, utilizando como métrica de similaridade a quantidade de teremos similares que existem na consulta feita pelo usuário e nos documentos recuperados.

O Jurisconsulto realiza buscas em acórdãos emitidos por tribunais. Estes acórdãos são modelados de forma que, além do texto, eles possuem outros atributos. Tais atributos são: data, relator, local, resultado, tipo do recurso e tipificação, a qual se subdividia em tipo geral, flagrante, qualificação, tentativa e co-autoria. Destes atributos, apenas o último entrava na métrica de similaridade, juntamente com o texto do acórdão. O texto do acórdão é representado por um vocabulário controlado, contendo as principais expressões sobre os assuntos tratados nos acórdãos.

O AlphaThemis recupera súmulas dos tribunais brasileiros. Também possui a representação do documento baseada no texto e em mais alguns atributos: tribunal, ano, ramo do direito, ramo secundário, indicador temático central, indicador temático subsidiário e número da súmula. A principal diferença entre o AlphaThemis e o Jurisconsulto é que o vocabulário controlado passa a ser dividido em três categorias: os termos existentes na súmula; os termos fortemente conexos, os quais representam termos que estão altamente relacionados com o assunto tratado na súmula; e os termos relativamente conexos, representando assuntos relacionados à súmula.

Outra diferença é a presença de uma maior quantidade de itens na métrica de similaridade. Todos os atributos, exceto o número da súmula, fazem parte da métrica. Cada um destes itens possui um peso específico, com o texto sendo o mais significativo. Além disto, existem os chamados pesos dinâmicos, onde o usuário pode determinar que um atributo participe com mais ou menos significância da métrica. Por exemplo, supondo que o usuário deseje que o tribunal tenha o dobro de importância dos demais atributos, ele deve marcar todos os atributos o peso dinâmico de 50% e o atributo tribunal com o peso 100%.

A Representação do Conhecimento Contextualizado Dinamicamente (RC^2D) é uma evolução na maneira de extrair o conhecimento dos documentos para facilitar o processo de recuperação. Unida com a Pesquisa Contextual Estruturada (PCE), forma um poderoso mecanismo de indexação, armazenamento e recuperação de documentos não estruturados. Estas duas tecnologias são melhor abordadas no capítulo dois.

O que estas aplicações têm em comum é o fato de elas necessitarem da criação manual dos índices do vocabulário controlado. Este processo é custoso, principalmente devido ao tempo gasto pelos especialistas para avaliar os documentos e extrair os termos iniciais. RC^2D acelera o processo de desenvolvimento da base de conhecimento. Ela possui alguns mecanismos que automatizam o processo, mas os levantamentos iniciais ainda precisam ser feitos manualmente. Isto é um processo custoso e demorado, despendendo uma grande quantidade de recursos humanos e tempo. Então, torna-se necessário criar um modelo que permita a construção de ferramentas para, no mínimo, auxiliar a extração de termos e seus relacionamentos.

1.2 Problema de Pesquisa

A recuperação de informações pode ser baseada na indexação de todo o documento ou apenas na indexação de partes dele, reconhecendo os elementos importantes através de um vocabulário controlado. Entretanto, a construção de um vocabulário controlado é um processo dispendioso, devido ao tempo demandado para o estudo e elaboração do mesmo. Além disto, a ausência de um especialista no domínio trabalhado pode acarretar na construção de um vocabulário incompleto ou, até mesmo, com erros para este domínio.

Este problema agrava-se com a extensão de um vocabulário controlado para ser uma ontologia dentro de uma base de conhecimento. As relações entre os termos feitas sem especialistas podem não ser válidas e leva o sistema a recuperar informações que não estejam devidamente relacionadas.

Imaginando-se uma ontologia que possa ser usada independentemente de domínio, este problema toma uma dimensão ainda maior, haja vista que não há disponibilidade de especialistas em todos os domínios e, muito menos, pode-se planejar a quantidade de tempo necessária para a construção da mesma.

Partindo-se disto, podemos considerar como sendo a pergunta de pesquisa: Quais são os elementos que um software de linguagem natural deve

ter para extrair automaticamente os conceitos e estabelecer o contexto de um texto qualquer?

1.3 Objetivos

1.3.1 Objetivos Gerais

O objetivo deste trabalho é desenvolver um modelo computacional para a extração automática de termos a partir de um conjunto aleatório de textos, visando a identificação dos conceitos e do contexto ao qual o texto pertence.

1.3.2 Objetivos Específicos

Os objetivos específicos são:

- Construir um modelo de representação de documento;
- Desenvolver um modelo de representação de textos voltado para a Língua Portuguesa;
- Desenvolver um modelo de relacionamento entre palavras para a criação de termos para um dicionário de ontologias;
- Desenvolver um modelo de relacionamento entre termos para o estabelecimento de contextos.

1.4 Metodologia

O modelo apresentado neste trabalho foi desenvolvido para a aplicação em qualquer conjunto de textos, desde que escritos em Português.

O trabalho iniciou-se com a observação do processo de construção de ontologias utilizado pela WBSA/IJURIS no sistema KMAI, ou seja, como os termos são formados e quais são as suas principais características. Procurou-se, então, verificar quais são as formações gramaticais existentes nos termos para que se determinar a possibilidade de criação de um mecanismo baseado em linguagem natural para o reconhecimento automático.

Como o estudo dos termos confirmou a existência dos padrões, o próximo passo foi verificar tecnologias que realizassem a representação dos textos em linguagem natural para uma representação formal. A tecnologia escolhida foi a UNL (Universal Networking Language), a qual possui uma estrutura formal bem determinada e minimizando consideravelmente as ambigüidades.

Entretanto, a UNL sofre de uma carência na conversão de textos em português. Desta forma, optou-se pela elaboração de um modelo de representação alternativo baseado na UNL. Este modelo baseia-se na estrutura das sentenças da própria UNL, porém mantém um conjunto de relações que permite a formação de regras mais simples para a conversão a partir do português.

Criada a estrutura de representação dos textos, partiu-se para a análise dela para permitir a extração dos termos. A própria representação já permite o reconhecimento dos termos, mas é necessário verificar quais deles são mais significativos. Para isto, estudou-se alternativas de cálculo de pesos. Escolheu-se as métricas de freqüência de termos e freqüência inversa de documentos utilizadas em Recuperação de Informação. A modificação feita é que tais métricas foram calculadas sobre as relações, ao invés de apenas com as palavras.

Além do cálculo dos pesos dos termos, verificou-se o estudo para calcular a freqüência conjunta das relações nos documentos para determinar quais relações podem ser consideradas como pertencentes a um mesmo contexto.

Ao final de todo este processo, tem-se o modelo descrito neste trabalho.

1.5 Justificativa

A construção de uma ontologia através de um processo manual é custosa, tanto em termos de tempo quanto em termos de dinheiro. Além disto, este processo pode possuir falhas no caso de equipes que não estejam

devidamente preparadas para realizar o processo de engenharia do conhecimento. Em sistemas comerciais, os clientes podem não ter a disponibilidade de tempo necessária para a realização de todo o processo de engenharia do conhecimento.

É necessário algum modelo que permita, pelo menos, que o sistema seja capaz de identificar quais são os conceitos-chave que existem no conjunto de documentos iniciais do usuário. Estes conceitos podem servir como subsídios para um futuro processo de engenharia do conhecimento ou até mesmo servir como a base de conhecimento do sistema, funcionando para o processo de recuperação de informações.

O modelo proposto reduz o tempo gasto pela engenharia do conhecimento, pois tenciona conseguir identificar automaticamente os conceitos existentes no conjunto de documentos, além de relacionar estes conceitos para que os documentos possam ser classificados em categorias.

Tendo em vista que o modelo apresentado neste trabalho visa a construção de ferramentas que permitam ampliar a capacidade dos computadores de manipular de documentos textuais, melhorando a interação entre homens e computadores, o trabalho encaixa-se no contexto da Engenharia de Produção e Sistemas.

1.6 Descrição dos Capítulos

O capítulo um apresenta a contextualização do tema, pergunta de pesquisa, objetivos, metodologia e justificativa para o desenvolvimento deste trabalho.

O capítulo dois descreve os referenciais teóricos utilizados neste trabalho. Nele encontram-se conceitos de Recuperação de Informação, Ontologias, RC^2D e PCE, Processamento de Linguagem Natural, UNL e Gramática.

O capítulo três descreve a evolução do processo de construção de ontologias que é a base do modelo descrito.

O capítulo quatro demonstra o modelo teórico proposto.

O capítulo cinco apresenta as conclusões obtidas no trabalho.

2 REFERENCIAIS TEÓRICOS

A construção de um modelo automático de extração de conceitos envolve diversas áreas de conhecimento e tecnologias. Este capítulo analisa as tecnologias e elementos teóricos que serviram como base para a elaboração deste modelo.

Como o modelo propõe-se a recuperar documentos textuais a partir de consulta de usuários, ele deve basear-se em teorias de Recuperação de Informação, que é a primeira tecnologia abordada. Entretanto, existem diversas outras tecnologias que são utilizadas para melhorar o processo de recuperação, aprimorando a resposta dada para o usuário. Algumas destas tecnologias foram trabalhadas para a construção deste modelo, citando-se Recuperação do Conhecimento Contextualizado Dinamicamente (RC^2D), Processamento de Linguagem Natural e UNL.

Além destas, o capítulo também mostra conceitos de ontologias voltados à Tecnologia da Informação, bem como alguns conceitos de gramática da Língua Portuguesa que são abordados no Processamento de Linguagem Natural.

2.1 Recuperação de Informação

Recuperação de Informação (RI) é a técnica tradicionalmente aplicada quanto se tenta recuperar documentos textuais para um determinado problema (RIJSBERGEN, 1979; SALTON e MCGILL, 1983). Entretanto, ao contrário do que o nome sugere, os sistemas de Recuperação de Informação não recuperam realmente a informação no sentido que eles entregam fatos satisfazendo uma informação necessária. Particularmente, eles procuram por documentos os quais possuem alguma coisa relacionada à informação necessária expressada em uma consulta. Como define Rijsbergen (1979), “Um sistema de recuperação de informação não informa (isto é, não modifica o conhecimento do) o usuário sobre o assunto de sua investigação. Ele

meramente informa sobre a existência (ou não existência) de documentos relacionados a sua requisição".

Há diferenças entre sistemas de recuperação de informação e sistemas de recuperação de dados. A principal diferença vem do fato que os sistemas de recuperação de dados retornam apenas a comparação exata com a consulta. Os parâmetros fornecidos pelo usuário são diretrizes cujos objetos no conjunto de resposta devem seguir. Já os sistemas de RI retornam as melhores respostas para o usuário, utilizando-se de comparação parcial. A consulta do usuário é um conjunto de palavras que representa a semântica da informação necessária.

Outra diferença relevante entre os dois tipos de sistemas é a natureza da linguagem utilizada pelo usuário para fazer a sua consulta. Nos sistemas de recuperação de dados são utilizadas linguagens artificiais, nas quais o usuário deve fazer uma descrição completa daquilo que deve ser recuperado. Por sua vez, os sistemas de recuperação de informação trabalham com linguagens formais que utilizam elementos de linguagem natural, permitindo ao usuário a descrição incompleta do que ele necessita. Mesmo assim, através da recuperação pela comparação parcial anteriormente mencionada, o sistema consegue recuperar os documentos para o usuário.

O último ponto que merece ser destacado é o fato de que um sistema de recuperação de dados não é tolerante a erros. Um erro na comparação implica na falha total do sistema. Já em um sistema de recuperação de informação, pequenos erros não afetam significantemente o sistema.

O fato é que a recuperação de informação é a principal tecnologia utilizada para a recuperação de documentos não estruturados. Embora, ela também possa ser utilizada para a recuperação de dados estruturados, a grande vantagem está na manipulação dos primeiros.

Os sistemas de RI trabalham com a representação de um documento. Esta representação pode ser de dois tipos: lógica ou completa. Na representação lógica, o sistema armazena uma lista de palavras-chaves ou termos de índice dos documentos. Esta lista pode ser produzida automaticamente ou por um especialista humano. Na representação completa,

o sistema armazena todo o conjunto de palavras do documento para a recuperação.

Ambas as representações sofrem uma seqüência de operações textuais, as quais reduzem a complexidade da representação do documento e permitem a mudança da visão lógica ou do texto completo para um conjunto de termos de índice. Este conjunto é composto por três passos. O primeiro é a remoção de *stop words*. *Stop words* são palavras que possuem uma alta freqüência de ocorrência no texto ou são palavras cuja utilização é auxiliar aos elementos principais da frase, não possuindo real significância e não interferindo no processo de recuperação. Nesta categoria enquadram-se os artigos, numerais, preposições, conjunções, interjeições e alguns pronomes. A remoção de *stop words* reduz o tamanho do texto a ser guardado entre 30% e 50%.

O segundo passo é a redução das palavras a um radical comum, num processo conhecido como *stemming*. Este processo consiste na comparação das palavras com uma lista de sufixos, removendo estes das palavras e mantendo apenas o radical. Algumas regras devem ser utilizadas para minimizar os casos ambíguos, tais como “o tamanho mínimo da palavra restante ser maior que 2” e/ou “a palavra não pode finalizar com 'q'”.

O último passo compreende a identificação dos grupos de substantivos, eliminando assim os adjetivos, advérbios e verbos.

Assim, no processo de indexação padrão de um sistema de Recuperação de Informação, os termos indexados compreendem apenas os radicais dos substantivos existentes nos textos.

As consultas devem sofrer o mesmo tratamento, para que seja possível fazer a correta comparação entre ela e a lista dos documentos.

A discussão sobre a eficácia destes procedimentos pode ser longa. A necessidade de adjetivos para a qualificação dos substantivos como elementos de busca é clara. A presença deles pode mudar totalmente o sentido de um substantivo, o que produz um resultado não desejado. Assim, torna-se necessário manter o adjetivo e, se possível, agrupá-lo ao substantivo,

formando um único índice de procura. Por exemplo, a palavra “guerra” possui um significado próprio. Já “guerra civil” e “guerra fria” possuem outros significados mais peculiares, demonstrando a influência que a presença dos adjetivos possui.

Já a influência das *stop words* no significado dos termos é menor, mas decisiva em alguns casos. Mas, é muito mais fácil para o usuário pensar nas expressões de consulta utilizando as *stop words* do que pensar sem elas. Mas, seu uso pode ser restrito a apenas o necessário, armazenando apenas o necessário.

Outro fator que influencia na representação do documento é a forma do vocabulário de indexação. O vocabulário pode ser controlado ou não controlado. Vocabulários controlados são listas de termos que o indexador deve utilizar para encontrar os índices dos documentos da base. Estes termos podem estar agrupados entre si, classificados hierarquicamente ou organizados de outra forma. Rijsbergen (1979) afirma que “Não há limite para o tipo de controle sintático que uma linguagem (de indexação) pode ter”. Os vocabulários controlados que possuem tais características organizacionais assemelham-se muito a ontologias, podendo até ser classificados como tais em alguns casos.

Já os vocabulários não controlados correspondem exatamente ao fato de não existirem palavras ou expressões a ser procuradas. Todas os elementos são considerados na indexação, podendo ou não sofrer o processamento anteriormente citado (*stemming* e localização de substantivos).

Todavia, sabe-se que uma tarefa que os sistemas de Recuperação de Informação tradicionalmente não executam é encontrar documentos cujo conteúdo esteja relacionado à consulta do usuário, mas as palavras por ele utilizadas não estão presentes nestes documentos. Este refinamento é conseguido através do uso de ontologias, que serão analisadas posteriormente.

Existem algumas métricas utilizadas para medir o valor dos termos indexados. As principais métricas são a freqüência de termos e a freqüência inversa de documentos.

A freqüência de termos mede a importância de um termo dentro de um documento em particular. Ela pode ser calculada através da seguinte fórmula (Rijsbergen, 1979):

$tf_i = \frac{n_i}{\sum_k n_k}$, onde n_i é o número de vezes que o termo *i* aparece no documento e n_k o número de vezes que cada um dos *k* termos aparecem no documento (ou seja, o número de termos do documento).

A freqüência inversa do documento mede a importância geral de um termo com relação a toda a coleção de documentos. Ela é calculada através da seguinte fórmula (Rijsbergen, 1979):

$idf_i = \log_2(\frac{N}{n_i})$, onde *N* é o número de documentos da coleção e n_i é o número de documentos em que o termo *i* aparece.

A importância de um termo para um documento é calculada através da fórmula *tf*idf*. Esta fórmula destaca que um termo que aparece mais vezes em um documento e aparece em poucos documentos da coleção é o termo mais representativo para aquele documento. Ela é muito utilizada pelos motores de busca para auxiliar na recuperação dos documentos.

Por exemplo, considerando-se que o termo “narcotráfico” apareça 2 vezes em um documento de 100 palavras. A freqüência do termo será 0,02. Considerando-se, então, que a coleção possua 1.000.000 de documentos e “narcotráfico” apareça em 700 deles. A freqüência inversa do documento será 10,4804. O valor de *tf*idf* é igual a 0,2096.

A utilização de pesos é muito importante para os motores de busca. Yu e Salton (1977) e Jones (1972), por exemplo, apresentam provas formais que a utilização de pesos relacionados à freqüência inversa de documentos, mais especificamente o valor de (*idf* + 1), torna a recuperação muito mais efetiva que sem a utilização de qualquer peso.

Assim, conclui-se que qualquer sistema que trabalhe com Recuperação de Informação deve considerar seriamente a utilização de pesos na sua estrutura, não importando qual é a abordagem utilizada.

Outro ponto importante a ser considerado nos sistemas de RI é a forma de avaliação deles. As métricas mais utilizadas para a medição destes sistemas são a precisão e o *recall*. A precisão é o número de documentos relevantes recuperados dividido pelo número de documentos recuperados. O *recall* (também chamado revocação) é o número de documentos relevantes recuperados dividido pelo número de documentos relevantes da base. Além destes, Cleverdon et. al. (1966) cita outras quatro métricas:

1. A cobertura da coleção, isto é, a extensão que o sistema inclui a matéria relevante;
2. O tempo de resposta, isto é, o tempo que o sistema leva entre o início da requisição e a apresentação da resposta;
3. A forma da apresentação da saída;
4. O esforço envolvido por parte do usuário para obter as respostas.

Embora estas métricas sejam razoavelmente fáceis de determinar, eles podem sofrer por subjetividade dos avaliadores ou por outros elementos externos, tais como a capacidade da máquina que está processando, etc.

Em outro trabalho, Cleverdon (1967) afirma que os valores de *recall* e precisão são inversamente proporcionais, ou seja, um sistema para obter um alto valor de *recall* deve sacrificar sua precisão e vice-versa. O grande desafio é encontrar um valor de intersecção que maximize ambos os valores. Afinal, não existe um valor pré-determinado para esta intersecção.

Existe uma métrica que realiza uma média harmônica com pesos entre a precisão e o *recall*, chamada métrica F (tradução livre de *F-measure*). A métrica F tradicional é calculada por:

$$F = \frac{2 \times precisão \times recall}{precisão + recall}$$

Esta é conhecida como métrica F_1, porque a precisão e o *recall* possuem o mesmo peso. A fórmula geral é:

$$F_N = \frac{(1 + N^2) \times precisão \times recall}{(N^2 \times precisão) + recall}$$

Além da F_1, outras métricas comumente utilizadas são a $F_{0.5}$, que significa que a precisão tem o dobro de peso do *recall*, e a F_2, em que o *recall* tem o dobro do peso da precisão.

Os valores de precisão e *recall* podem ser calculados para qualquer tipo de sistema de recuperação envolvendo documentos não estruturados. Lenz et. al. (1998), por exemplo, mostra a aplicação destas métricas na avaliação de sistemas de Raciocínio Baseado em Casos textuais. Da mesma forma, Bueno et. al. (2003) mostra a avaliação de um sistema que utiliza RC^2D com as mesmas métricas.

2.2 Ontologias

Na filosofia, a ontologia é a parte que estuda o ser enquanto ser, isto é, o ser considerado independente de suas determinações particulares, e naquilo que constitui sua inteligibilidade própria. Pode-se definir ontologia como teoria do ser em geral, da essência do real.

Russel e Norvig (1995) definem as ontologias, no campo da Inteligência Artificial, como sendo o vocabulário de predicados, funções e constantes que representam os conceitos do domínio no nível lógico, entretanto, sem determinar suas propriedades específicas e suas relações.

O uso de ontologias em sistemas de Recuperação de Informação adiciona a eles a capacidade de recuperar documentos que estejam relacionados ao assunto que está sendo pesquisado sem que eles contenham as palavras utilizadas na consulta.

Bueno (2005) mostra a Suíte de Engenharia do Conhecimento, uma ferramenta para a construção de ontologias que estende o conceito para englobar também as relações entre os elementos que as compõem. As ontologias demonstradas em Bueno (2005) baseiam-se na representação de documentos não-estruturados, ou seja, na determinação de termos que são relevantes para a futura recuperação de tais documentos. Os termos, além da função citada, também são relacionados entre si, de forma a permitir que os

documentos possam ser classificados dentro de domínios. Existem quatro tipos de relação entre os termos:

- **Sinônimos**: representam termos que podem ser intercambiáveis sem alterar o significado da frase em que estão inseridos. Ex.: “narcotráfico” e “tráfico de drogas”;
- **Conexos**: representam termos que são relacionados fortemente dentro do contexto que está sendo trabalhado. Ex.: no contexto de futebol, pode-se dizer que “juiz” é conexo de “bandeira”.
- **Tipo de**: representam as relações de hiperonímia e hiponímia. Estas relações indicam que um termo possui um sentido mais genérico e outro mais específico. Por exemplo, “animal” possui relação de hiperonímia com “leão”, enquanto “leão” possui relação de hiponímia com “animal”.
- **Parte de**: representa a relação de meronímia, ou seja, que um termo é parte de outro termo. Exemplo: “roda” é parte de “carro”. Esta relação também possui uma representação contrária, permitindo que se possa descrever que “carro” é todo de “roda”.

Entretanto, a construção de ontologias é um processo que demanda uma grande quantidade de tempo e recursos humanos. Um especialista no domínio que está sendo mapeado deve estar presente praticamente durante todo o período de construção para garantir que o resultado do processo de construção tenha qualidade.

Além disto, uma ontologia manualmente construída sempre corre o risco de estar incompleta, tanto pela ausência de termos quanto pela ausência de relacionamentos entre os termos. Também, a inclusão de novos termos pode ser falha, devido a fatores externos à execução do sistema. Por exemplo, o excesso de trabalho do especialista do domínio ou dos engenheiros do conhecimento.

Tudo isto leva à necessidade de um sistema que consiga identificar conceitos automaticamente, ou, pelo menos, faça um levantamento dos

conceitos que aparecem nos documentos para sugeri-los ao especialista e que ele apenas necessite confirmar (ou rejeitar) os novos termos.

2.3 RC²D e PCE

A Representação do Conhecimento Contextualizado Dinamicamente (RC²D), apresentada por Hoeschl (2001), é uma metodologia para a representação do conhecimento baseada na presença de dicionários de termos normativos e avaliação de freqüência de palavras. A RC²D permite a representação automática de casos em sistemas baseados em conhecimento. A Figura 2 mostra o processo de construção de uma base de conhecimento utilizando a RC²D.

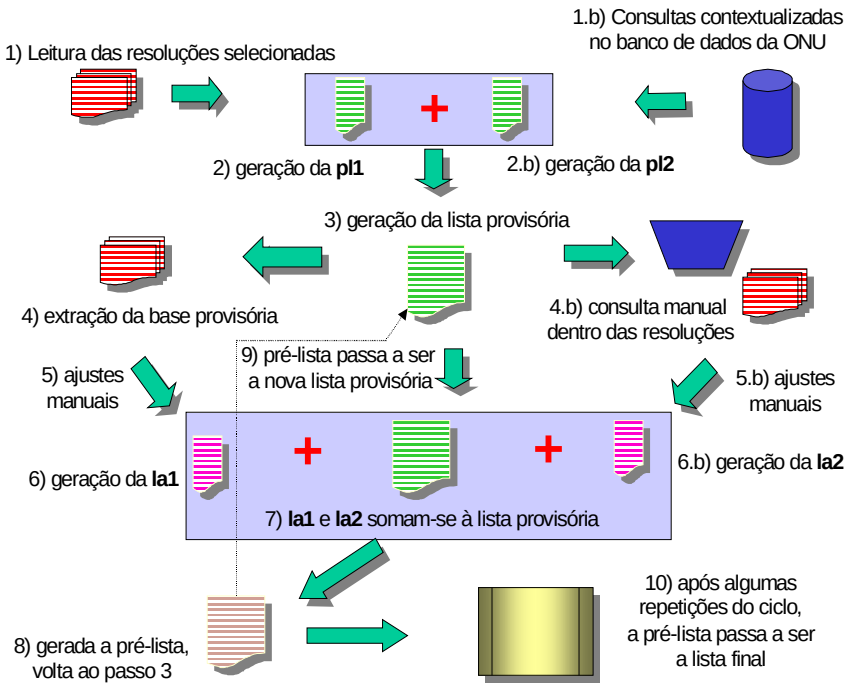


Figura 2. RC²D

A construção da base é iniciada através do levantamento de dois conjuntos de documentos. O primeiro é uma amostra do conjunto que será utilizado posteriormente no sistema, enquanto o segundo são outros documentos relacionados ao contexto desejado. A análise manual de cada um destes conjuntos gera uma pré-lista de termos, as quais são unidas para a

formação da primeira lista provisória. O próximo passo corresponde à análise estatística dos termos na base de documentos provisórios. Esta base corresponde a um conjunto de treinamento, com documentos que são exatamente da mesma categoria que os documentos que serão utilizados no sistema. A análise estatística é feita tanto manualmente quanto automaticamente, através de softwares estatísticos. O resultado das análises passam por ajustes manuais, gerando novas listas que são somadas à lista provisória. Ao final, a pré-lista é analisada e retorna-se ao passo 3, considerando a pré-lista como uma nova lista provisória, até que o resultado seja satisfatório e a lista de termos seja a lista final.

Hoeschl (2001) classifica o processo RC^2D em uma etapa teórica e outra empírica:

1. Etapa teórica: diz respeito à análise de documentos e textos referentes ao assunto enfocado, e posterior processo dialético envolvendo especialistas na área de recuperação documental e/ou no assunto abordado pelo sistema, a fim de identificar quais expressões são relevantes e caracterizadoras dos assuntos tratados nos documentos.
2. Etapa empírica: consiste na experimentação feita com as expressões extraídas do processo teórico sobre os documentos que farão parte do sistema, bem como análise numérica sobre a ocorrência das expressões. Também foram levadas em consideração as estatísticas sobre incidência das expressões nos documentos, dado utilizado para inclusão/ampliação de índices, ou supressão de alguns deles, ou decomposição ou, ainda, agrupamento.

A Pesquisa Contextual Estruturada (PCE) é uma metodologia para a realização de pesquisas em linguagem natural, abstraindo-se conceitos como a utilização de conectores lógicos e palavras-chave. Segundo Hoeschl (2001), “A pesquisa é considerada ‘contextual’ e ‘estruturada’ pelas seguintes razões: 1. É levado em consideração o contexto dos documentos armazenados quando da formação de estrutura retórica do sistema; 2. Este contexto norteia o processo de ajuste da entrada bem como da comparação e seleção dos documentos; 3. Quando da elaboração da consulta, a entrada não está limitada

a um conjunto de palavras, ou à indicação de atributos, podendo assumir o formato de uma questão estruturada pelo conjunto de um longo texto somado à possibilidade de acionamento de pesos dinâmicos sobre atributos específicos, que funcionam como 'filtros' e fazem uma seleção preliminar dos documentos a serem analisados".

A Figura 3 mostra o processo de consulta na PCE, utilizando um de seus recursos, que são os filtros de nível. A consulta do usuário é mapeada para representar a mesma estrutura dos documentos que estão armazenadas na base de documentos. Esta consulta derivada é enviada para o processo de recuperação. O primeiro passo é a aplicação do filtro de nível, onde todos os documentos que tiverem um número de termos na consulta inferior ao estabelecido no filtro são descartados. Os documentos restantes são comparados à consulta através de uma métrica de similaridade e são apresentados para o usuário ordenados pelo valor de similaridade obtido.

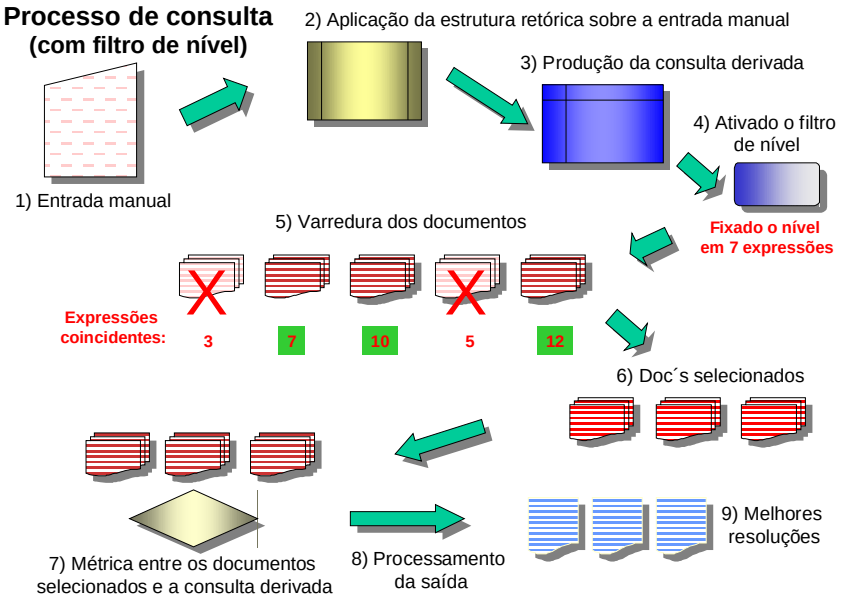


Figura 3. PCE

Embora os resultados obtidos pelos sistemas que utilizam as tecnologias RC^{2}D e PCE sejam excelentes (HOESCHL, 2001; HOESCHL et. al. 2003; HOESCHL et. al. 2004; RIBEIRO, 2004), a construção da base de conhecimento pode ser melhorada, principalmente nos passos iniciais, onde

muitos processos manuais podem ser automatizados. O levantamento das listas iniciais, por exemplo, pode ser feito por um sistema que sugira quais termos estão presentes nos documentos e que podem ser utilizados. Posteriormente, a qualificação dos termos também pode ser auxiliada por programas que calculem pesos indicativos do valor que eles possuem para a coleção de documentos.

2.4 Processamento de Linguagem Natural

O uso de técnicas de Processamento de Linguagem Natural (PLN) em mecanismos que trabalham com informação tornou-se necessário a medida que os sistemas começaram a manipular grandes volumes de informação não-estruturada (textos), bem como com o aumento da sofisticação desejada pelo usuário para atender a suas consultas.

As técnicas de PLN começaram a ser estudadas no início da década de 60. Por esta época, acreditava-se que os computadores facilmente poderiam traduzir textos de um idioma para outro. Já na metade desta década, percebeu-se que a tarefa não seria fácil como se pensava. Atualmente, os sistemas de PLN tiveram progressos reais, conseguindo bons resultados em domínios restritos. Para domínios amplos, os resultados não são satisfatórios. Isto acontece porque as linguagens naturais possuem uma variedade enorme de possibilidades de formação de frases. Também, uma mesma idéia pode ser expressa de mais de uma maneira. Estes problemas ainda não foram devidamente resolvidos, o que acaba prejudicando o desempenho dos sistemas em domínios abertos.

As principais aplicações de PLN são programas de consultas a banco de dados, extrair informações de textos, recuperar documentos relevantes de uma coleção, traduzir textos entre idiomas e reconhecer palavras faladas.

Os sistemas de PLN seguem um processo de quatro etapas para o reconhecimento de palavras e o processamento do texto que é fornecido como entrada para eles. Como os sistemas não sabem quais são as palavras que são digitadas, ele precisa reconhecê-las, processando a entrada caractere a

caractere. As quatro etapas são: *tokenização*, análise morfológica[1], procura no dicionário e recuperação de erros.

A *tokenização* é o processo de dividir a entrada em *tokens* distintos, ou seja, as palavras e os sinais de pontuação. Este processo é trivial para linguagens ocidentais, baseadas nos caracteres latinos. Entretanto, para idiomas como o japonês, a tarefa é mais árdua, haja vista que não há espaços em branco entre as palavras.

A análise morfológica, em PLN, é o processo de descrever uma palavra em termos de seus prefixos, sufixos e radicais. As palavras podem ser compostas de três maneiras diferentes:

- **Flexão**: Reflete as modificações nas palavras para a formação dos contextos gramaticais. Por exemplo: o sufixo “s” no plural.
- **Derivação**: Deriva uma palavra a partir de outra, usualmente de outra categoria. Por exemplo, “corrida” que deriva de “correr”.
- **Composição**: Une duas palavras. As palavras podem estar unidas com ou sem hífen. Na Língua Portuguesa, esta diferenciação denomina-se palavras compostas por justaposição, quando os radicais não sofrem alteração (Exemplo: primeira-dama, beija-flor). Quando os radicais sofrem alteração, diz-se que as palavras são compostas por aglutinação (Exemplo: petróleo = pedra + óleo, aguardente = água + ardente).

Há muitas ambigüidades morfológicas envolvidas no processo. Um exemplo é a própria palavra “processo”. Ela pode ser tanto um substantivo, como na frase anterior, quanto um verbo, a primeira pessoa do singular do presente do indicativo (eu processo).

A procura no dicionário é encontrar a palavra em um dicionário e retornar sua definição. Este dicionário pode conter também as entradas para palavras, cuja derivação pela análise morfológica seja muito complexa, além das entradas comuns dos dicionários. Por exemplo, um dicionário normalmente

[1]Alguns conceitos descritos aqui neste processo podem ter suas definições diferentes da sua descrição na gramática da Língua Portuguesa, tais como análise morfológica e formação das palavras.

contém as palavras no masculino e no singular. Entretanto, não compensa implementar uma regra morfológica que defina que “princesa” é o feminino de “príncipe” e insira apenas este último no dicionário. Assim, colocam-se ambas as formas para o mecanismo de busca.

A recuperação de erros é o último passo do processo. Ela pretende reconhecer uma palavra que não está presente no dicionário. Há quatro tipos de recuperação de erros. A primeira é criar regras morfológicas que tentem adivinhar a classe gramatical baseada em partes das palavras. Exemplo: uma palavra que termina com o sufixo “mente” tem boa probabilidade de ser um advérbio. A segunda é a presença de letras maiúsculas, as quais são normalmente utilizadas para denotar nomes próprios, que são substantivos. A terceira é representar formatos especializados que identifiquem datas, horas, cifras monetárias, etc. A última é implementar rotinas de correção de erros para encontrar palavras que são próximas às palavras de entrada, identificando um erro de digitação ou gramatical. Os dois modelos mais populares para esta tarefa são os baseados em letras e os baseados em som. O modelo baseado em letras consiste na inserção ou remoção de uma letra na palavra, a transposição de duas letras da palavra ou na substituição de uma letra por outra. É um método custoso, haja vista que uma palavra de 10 letras transforma-se em 555 palavras na tentativa de correção de apenas um erro. O modelo baseado em som utiliza a pronúncia da palavra para a comparação com as demais palavras do dicionário, na tentativa de encontrar uma que tenha a pronúncia igual ou parecida com a entrada.

O ideal de um sistema de PLN é que ele possa comunicar-se com um usuário humano com a mesma fluência que dois humanos comunicam-se entre si. Ou seja, disponibilizar uma interface de tal forma que um usuário pudesse perguntar para o computador da mesma forma que pergunta para outra pessoa. Da mesma forma, o computador deveria ser capaz de responder para o usuário a informação precisa, e não uma lista de documentos que contém o assunto, como fazem os sistemas de Recuperação de Informação.

2.5 UNL

A Universal Networking Language (UNL) pode ser definida como uma metalinguagem digital para a descrição, armazenamento e disseminação de informação em uma forma independente de máquina e de linguagem natural (DAVE e BHATTACHARYYA, 2001; UNESCO, 2003). Ela foi desenvolvida pela Universidade das Nações Unidas na metade da década de 90 e, posteriormente, distribuída para vários centros ao redor do mundo. Seu funcionamento é o mesmo de uma interlíngua, ou seja, todas as linguagens naturais são convertidas para UNL e ela pode ser convertida para qualquer outra linguagem.

A UNL trabalha com a premissa de que a informação mais importante de uma sentença é o conceito que ela carrega. Este conceito é representado pela combinação das Palavras Universais e das Relações que também são universais, representadas para todos os idiomas.

A informação ou o conhecimento é expressada na forma de uma rede semântica com hipernodos, conforme descrito por UNDL (2004). Nesta rede semântica, os nós representam os conceitos e as arestas representam as relações entre os conceitos. Estas relações não possuem ambigüidade, tentando eliminar este problema que aparece nas linguagens naturais. A Figura 4 representa a rede semântica da UNL para a sentença "*There is a boy and two girls, the boy is naughty and the girls are cute*." (Há um menino e duas meninas, o menino é malcriado e as meninas são graciosas.)

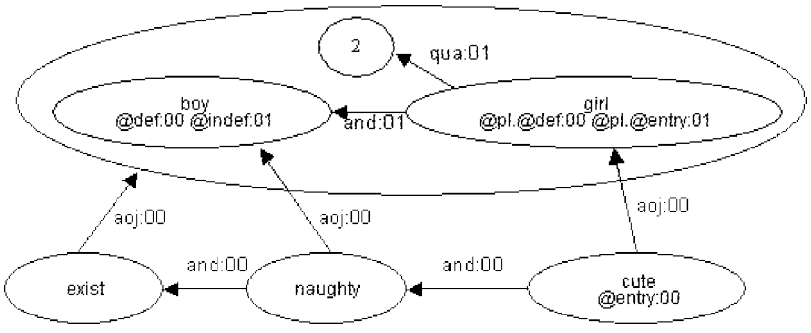


Figura 4. Rede semântica representando uma sentença UNL

Como pode-se observar na Figura 4, o nó “exist” liga-se com um nó que contém outros três nós. Este nó maior é chamado escopo, indicando que os nós dentro deles relacionam-se entre si de uma maneira particular e que o conjunto deles é que pertence ao significado da sentença completa.

Os três elementos que compõem e são representados na estrutura da UNL são os seguintes:

Palavras Universais: Constituem o vocabulário da UNL, representando um conceito relacionado a uma palavra. É dividida em duas partes. A primeira, chamada *headword*, é uma palavra em inglês que corresponde ao conceito que está sendo expresso. A segunda, chamada *constraint list*, representa a interpretação da Palavra Universal para o subconjunto ou o conceito especificado pela *headword*. Por exemplo, imaginemos a Palavra Universal “drink(agt>thing,obj>liquid)”. A *headword* expressa o conceito de “beber”, enquanto a lista de restrições especifica o conceito “beber” para aquele que é feito por um agente qualquer e cujo objeto é um líquido. As Palavras Universais são unificadas em uma Base de Conhecimento. Nela, as Palavras Universais são relacionadas entre si, de forma hierárquica.

Relações: Relacionam duas Palavras Universais através de seu comportamento sintático. Elas pretendem descrever a objetividade da sentença. As relações são acrônimos de três letras representando algum dos comportamentos mapeados. Todas as relações na UNL são binárias, seguindo a forma: *rel(UW1, UW2)*. Exemplos de relações são: “agt (agente)”, “obj (objeto)”, “and (e)”, “or (ou)”.

Atributos: Ao contrário das Relações, os Atributos descrevem a subjetividade da sentença. Elas mostram o que quer ser dito a partir do ponto de vista do orador. Os Atributos dividem-se em sete tipos: o tempo com relação ao orador (presente, passado, futuro), aspectos, visão de referência do orador, ênfase ou foco do orador, convenções, atitudes do orador e sentimentos e pontos de vista do orador.

Além destes mecanismos, a UNL trabalha com algumas ferramentas que são as responsáveis pelo processo de conversão. As mais importantes são

o *Enconverter* e o *Deconverter*. O primeiro é responsável pelo processo de conversão da linguagem natural para a UNL. O segundo faz o processo inverso.

Tanto o *Enconverter* quanto o *Deconverter* utilizam duas estruturas básicas para a realização de seu processo de conversão: o dicionário de palavras e o dicionário de regras. O dicionário de palavras contém o relacionamento entre as palavras do idioma e as palavras universais. O dicionário de regras contém as regras de conversão. É baseado nas regras de formação das frases do idioma.

O *Enconverter* utiliza também uma estrutura chamada Base de Conhecimento. A Base de Conhecimento é uma estrutura em forma de árvore, onde todas as palavras universais estão relacionadas hierarquicamente. Ela baseia-se na divisão das palavras em quatro tipos de conceitos: nominais, verbais, adjetivais e adverbiais.

O *Deconverter* utiliza, além dos elementos já citados, o dicionário de co-ocorrências, utilizado para selecionar as palavras mais apropriadas para um determinado contexto na linguagem para a qual está sendo convertida a sentença.

Por exemplo, consideremos a sentença em português “A menina lê a revista.” Esta sentença, em UNL, seria representada por:

agt(read(agt>thing,obj>thing).@entry, girl(icl>concrete thing).@def)
obj(read(agt>thing,obj>thing).@entry, magazine(icl>concrete thing) .@def)

Os elementos da UNL encontrados na sentença anterior são:

Palavras Universais: *read(agt>thing,obj>thing)*, *girl(icl>concrete thing)*, *magazine(icl>concrete thing)*. Cada uma destas palavras universais representa um dos conceitos utilizados na frase: ler, menina e revista, respectivamente. Nestas palavras universais pode-se observar as restrições, as quais seguem regras para a montagem da base de conhecimento (as restrições estão simplificadas a título de exemplo). As restrições para os conceitos “menina” e “revista” indicam que eles são conceitos nominais. Já a do conceito “ler”, indica que ele é um conceito verbal.

Relações: agt, obj. Representam o relacionamento entre os conceitos. O primeiro é a relação de agente do verbo, onde o conceito “ler” recebe como agente o conceito “menina”. O segundo é a relação de objeto, onde “ler” é relacionado ao objeto “lido”, ou seja, “revista”.

Atributos: *@def*, *@entry*. Representam características das palavras universais de acordo com o ponto de vista do orador. A primeira indica que o conceito é definido, ou seja, é uma menina específica que está sendo mencionada. O segundo é um atributo especial. Serve para mostrar ao *Deconverter* qual é o ponto de partida para o processo de conversão de UNL para o idioma.

O dicionário de entradas para a sentença UNL anterior seria o seguinte:

[a] {} (ART, FEM, ^NBR) <P,0,0>;
[ê] {} (SFX, P45, 3PS, ET1, IND) <P,0,0>;
[l] {} “read(agt>thing,obj>thing)” (VER, STM, P45, ACT, VD2) <P,0,0>;
[menina] {} “girl(icl>concrete thing)” (NOU, FEM, ^NBR) <P,0,0>;
[revista] {} “magazine(icl>concrete thing)” (NOU, FEM, ^NBR) <P,0,0>;

Entre colchetes está a *headword*, que é a palavra na língua original. Em alguns casos, como na entrada [l], encontra-se apenas a parte da palavra que é realmente invariável, ou seja, que não vai ser alterada em nenhuma das variações das palavras.

A segunda coluna é o ID da palavra, representado entre chaves. Este ID é opcional, não sendo utilizado neste exemplo.

A palavra universal é representada entre aspas. Algumas entradas não possuem palavras universais porque elas não representam conceitos, mas são complementos das palavras normais, tais como os sufixos, artigos, preposições, entre outras.

Após a palavra universal, entre parênteses, encontra-se a lista de atributos. Estes atributos, que não são os mesmos atributos da especificação da UNL, representam as características morfológicas, sintáticas e até semânticas da entrada na língua de origem.

A última informação é referente ao idioma, freqüência e prioridade das entradas. A freqüência indica quantas vezes a entrada aparece no dicionário e a prioridade indica qual a entrada que terá preferência para ser testada pelo *Enconverter* ou pelo *Deconverter*.

As regras da UNL montam a sentença UNL, no caso do *Enconverter*, ou a sentença no idioma nativo, no caso do *Deconverter*. Um exemplo de regra que seria utilizada para a conversão da sentença em português anterior para UNL seria (a regra é muito simples, tendo que ser melhor trabalhada para sua utilização):

>{NOU::agt:}{VER,ACT:::}P100;

O sinal de > representa o tipo de regra. Existem 13 tipos de regras para o *Enconverter*, as quais podem ser melhor estudadas na especificação do *Enconverter*. Esta indica que o nó da esquerda será ligado ao nó da direita através de uma relação.

As chaves representam os nós que estão sendo analisados na regra. Cada nó possui quatro seções, divididas pelo sinal de dois pontos (:). Na primeira seção, encontram-se as condições, ou seja, quais são os atributos que o nó deve ter para a regra ser aplicada. Na segunda seção, quais são as ações que serão aplicadas no nó caso a regra seja aplicada. Na terceira seção, encontram-se as relações que o nó deve possuir para a aplicação da regra. A última seção contém os papéis que o nó deve ter no dicionário de co-ocorrência (*Deconverter*) ou na base de conhecimento (*Enconverter*).

Embora o projeto UNL ainda não tenha sido finalizado e os dicionários de conversão de e para o português ainda não estejam disponíveis, os conceitos que fundamentam a UNL são apropriados para a representação de conhecimento em sistemas. Dave e Bhattacharyya (2001) apresentam um modelo de agrupamento de documentos baseado em UNL, ressaltando as características de desambigüização das Palavras Universais e da capacidade de representação que as relações possuem. O modelo apresentado por Dave e Bhattacharyya (2001) utiliza os já existentes mecanismos de conversão para inglês e indiano. A inexistência dos dicionários em português prejudica

iniciativas de utilização da UNL para a representação dos documentos neste idioma.

2.6 Gramática

As linguagens naturais possuem uma representação formal de sua estrutura. Esta estrutura formal é a gramática, através da qual podem-se determinar todas as funções que as palavras desempenham nos textos, sejam eles escritos ou falados. Embora a gramática de um idioma sofra as modificações implicadas pelas mudanças nos hábitos lingüísticos dos falantes dele, ela tem condições de ser a melhor aproximação de uma ciência exata que pode haver para um idioma.

O intuito deste trabalho é a construção de um sistema de computador que consiga fazer a representação formal de um texto, na tentativa de extrair seus conceitos e determinar o contexto ao qual o documento pode ser encaixado.

Desta forma, o modelo deve pressupor a utilização de mecanismos e técnicas de gramática, tais como as análises morfológica e sintática. Assim, esta seção descreve alguns conceitos relacionados à gramática e que são utilizados no decorrer do trabalho.

A análise morfológica é o processo de determinação da classe gramatical das palavras que compõem uma frase ou um texto. Todas as palavras da Língua Portuguesa pertencem a uma das dez seguintes classes gramaticais, de acordo com Faraco & Moura (1995a):

- **Substantivo**: Palavra que dá nome aos seres;
- **Verbo**: Palavra que indica ação, estado ou fenômeno da natureza;
- **Adjetivo**: Palavra que caracteriza os seres;
- **Artigo**: Palavra que acompanha o substantivo, determinando ou indeterminando-o;
- **Pronome**: Palavra que representa ou acompanha o substantivo, considerando-o apenas como pessoa do discurso;

- **Numeral**: Palavra que indica quantidade ou ordem dos seres;
- **Advérbio**: Palavra que indica circunstância. Pode acompanhar um verbo, um adjetivo ou mesmo um advérbio;
- **Preposição**: Palavra que serve para ligar dois termos de uma mesma oração;
- **Conjunção**: Palavra que serve para relacionar duas orações ou dois termos semelhantes de uma mesma oração;
- **Interjeição**: Palavra que expressa sentimento ou emoção.

As seis primeiras classes são conhecidas como variáveis, porque apresentam mudança de forma. As demais são invariáveis, porque não apresentam esta mudança.

As combinações e as relações entre as palavras são objetos de estudo da sintaxe. Conforme Faraco & Moura (1995b), à sintaxe interessam:

a) a função que as palavras exercem na frase (função sintática).

b) a ordem das palavras na frase (sintaxe de colocação).

c) a concordância das palavras na frase (sintaxe de concordância).

d) a dependência das palavras na frase (sintaxe de regência).

A análise sintática é o processo de exame da estrutura de um período e das orações que compõem um período. Neste processo é determinada a função sintática das palavras. Para uma melhor compreensão do processo, inicia-se pela definição dos conceitos de frase, período e oração.

Faraco e Moura (1995b) definem que uma frase “é qualquer enunciado lingüístico que tem sentido completo. Pode ser formada por uma só palavra ou por várias, pode ter verbo ou não.” Exemplos: “Oi.” “Atenção!” “A casa é amarela.”

Um período é a frase que tem verbo. Ela encerra-se sempre com um dos seguintes sinais de pontuação: ponto final, ponto de interrogação, ponto de exclamação, reticências, dois pontos (em alguns casos). O período pode ser

simples, quando possui apenas um verbo, ou composto, quando tem mais de um verbo.

Uma oração é a parte do período que se organiza em torno de um verbo ou de uma locução verbal.

Na gramática, um termo é a palavra considerada de acordo com a função sintática que exerce na oração. Os termos da oração podem ser:

1. **Essenciais**: Existem em todas as orações.

- **Sujeito:** É o termo que denota o ser a respeito de quem ou de que faz uma declaração. Pode ser simples, composto, indeterminado ou inexistente.
- **Predicado:** É o que se declara a respeito do sujeito. Pode ser verbal, nominal ou verbo-nominal.

2. **Integrantes**: Completam o sentido dos verbos e dos nomes

- **Complemento verbal:** Há dois complementos verbais, o objeto direto e o objeto indireto. O objeto direto completa o sentido de um verbo sem a necessidade de uma preposição obrigatória. O objeto indireto completa o sentido de um verbo com a necessidade de uma preposição obrigatória.
- **Complemento nominal:** É o termo que, precedido de preposição, completa o sentido de um substantivo, adjetivo ou advérbio.
- **Agente da passiva:** É o termo que indica o ser que pratica a ação quando o verbo está na voz passiva. Corresponde ao sujeito da voz ativa.

3. **Acessórios**: Desempenham função secundária, especificando o substantivo ou expressando circunstância.

- **Adjunto adnominal:** É o termo que acompanha sempre o substantivo, determinando-o ou qualificando-o.
- **Adjunto adverbial:** É o termo da oração que indica uma circunstância para o verbo ou intensifica o adjetivo ou o próprio advérbio.

- **Aposto:** É o termo que se junta o a outro de valor substantivo ou pronominal para explicá-lo ou especificá-lo melhor. Normalmente, vem separado da oração por vírgula, dois pontos ou travessão.

Além destes, existe o Vocativo, que é um termo à parte, não pertencendo à estrutura da oração. Serve para chamar ou interpelar alguém.

A análise sintática compreende também a análise do período. Um período pode ser composto por coordenação, subordinação ou coordenação e subordinação. Períodos compostos por coordenação são formados por orações cujos significados são independentes, ou seja, você não precisa de uma oração para entender o que a outra diz. Nos períodos compostos por subordinação, o entendimento de uma oração depende da outra. Nos períodos compostos por coordenação e subordinação, existem orações independentes e dependentes nele. Entretanto, tal análise não é objeto de estudo deste trabalho, haja vista que ela não interfere nos objetivos do trabalho.

O estudo da gramática auxilia na maneira que os textos podem ser representados. Ou, pelo menos, é necessário para a construção de regras que faça a conversão da linguagem natural para alguma linguagem que o computador representa, como, por exemplo, a UNL.

3 Evolução e Avaliação do Modelo de Construção de Ontologias

O modelo de construção de ontologias abordado neste trabalho é utilizado no desenvolvimento da base de conhecimento do sistema KMAI, desenvolvido pela empresa WBSA Sistemas Inteligentes S.A. em parceria com o Instituto de Governo Eletrônico, Inteligência Jurídica e Sistemas, o Ijuris.

Este modelo começou a ser desenvolvido quando da construção do software Jurisconsulto. O Jurisconsulto é um software para a recuperação de acórdãos de tribunais. Acórdãos são as decisões dos processos, publicadas pelos juízes. São textos semi-estruturados, com um cabeçalho definido, contendo as informações introdutórias, e um texto corrido, podendo ter várias páginas. Para recuperá-los, o Jurisconsulto utiliza a técnica de Raciocínio Baseado em Casos (RBC). Cada acórdão é representado como um caso, contendo uma seqüência de atributos para a recuperação. Os atributos de um caso são: Número, Relator, Local, Data da Publicação, Tipo do Recurso, Resultado, Tipificação e um conjunto de Expressões Indicativas. A Tipificação, por sua vez, era subdividida em cinco partes: Tipo Geral, Modalidade, Qualificação, Tentativa e Co-autoria.

O processo de recuperação é executado quando a consulta do usuário é transformada em um caso no mesmo formato do acórdão. São retornados como resposta os acórdãos mais similares à consulta. O grau de similaridade é calculado utilizando as expressões indicativas e a tipificação. Os demais atributos são utilizados como filtro.

As expressões indicativas, responsáveis pela representação do texto, consistiam de uma lista de expressões extraídas dos textos, baseadas nas leis do Código Penal Brasileiro. Tais expressões eram criadas de acordo com a necessidade, permitindo a recuperação do acórdão. Estas expressões não possuíam ligação entre si e era necessário entrar com a expressão corretamente para recuperar o acórdão desejado. Para determinar quais expressões indicativas estavam relacionadas ao acórdão, o sistema verificava

o texto do acórdão, procurando quais expressões da lista estão ali presentes. Este processo era feito durante a indexação do acórdão. Na indexação, também são identificados automaticamente todos os demais atributos, através de critérios de localização, através da posição dos atributos no texto e através de regras, para identificar a localização, onde é procurado o crime do qual o acórdão trata.

A evolução do Jurisconsulto é o Olimpo, um sistema para a recuperação das resoluções do Conselho de Segurança da ONU. Da mesma forma que um acórdão, uma resolução é um documento semi-estruturado, contendo um cabeçalho com algumas informações e um corpo textual que alcança várias páginas. Para executar sua tarefa, o Olimpo utiliza a Pesquisa Contextual Estruturada (PCE), onde a resolução é representada por um conjunto de atributos, com alguns sendo utilizados apenas como filtros e outros sendo parte da métrica de similaridade.

A métrica de similaridade do Olimpo baseia-se na comparação das expressões indicativas da entrada do usuário e das expressões existentes na resolução. As resoluções que contiverem mais expressões indicativas inseridas pelo usuário, são as mais similares. Os atributos utilizados como filtros são: Ano, País, Sigla, Assunto, Número e Número da Reunião.

A representação da base de conhecimento do Olimpo é feita utilizando-se a metodologia de RC^2D. Embora o resultado final seja uma lista de expressões similar à existente no Jurisconsulto, a qualidade do resultado é superior, devido aos múltiplos ciclos de avaliação aos quais a lista de expressões é submetida. Da mesma forma que no Jurisconsulto, as expressões são independentes, não possuindo relação entre si. A indexação também é feita nos mesmos moldes do Jurisconsulto.

O primeira evolução dos sistemas que utiliza diversas categorias para os termos é o AlphaThemis. O AlphaThemis é um sistema que permite a recuperação de súmulas dos tribunais brasileiros. Da mesma forma que o Olimpo, o sistema é construído utilizando as técnicas de PCE e RC^2D. As súmulas são resumos de decisões de vários tribunais, determinando que casos sobre o mesmo assunto sejam julgados desta maneira.

Ao contrário dos acórdãos, que podem conter várias páginas, as súmulas possuem poucas linhas. Isto obriga que seja feita uma expansão do conhecimento das súmulas, enquanto nos acórdãos são extraídos apenas os principais elementos dele. A conseqüência desta expansão é que vários atributos da súmula acabam participando do processo de recuperação.

A estrutura da súmula é composta pelos seguintes atributos: Número da Súmula, Tribunal, Data, Ramo do Direito (a súmula pode ter mais de um), Indicador Temático Central (tema central da súmula), Indicador Temático Secundário (temas secundários da súmula), Termos Fortemente Conexos (termos diretamente relacionados ao texto da súmula) e Termos Relativamente Conexos (termos que têm uma relação mais distante com o texto da súmula).

Destes atributos, apenas o número da súmula, o tribunal e a data não fazem parte da métrica de similaridade. A consulta do usuário é transformada num caso com o mesmo formato da súmula. O texto da consulta é comparado com o texto da súmula, com o indicador temático central e secundário, e com os termos conexos. Caso um termo seja encontrado no texto da súmula ou no indicador temático central, seu valor é contado por inteiro. Caso seja encontrado no indicador temático secundário, considera-se apenas 90% do valor. Caso seja um termo fortemente conexo, o valor é 70% e caso seja termo relativamente conexo, o valor é 40%.

O retorno do AlphaThemis são as súmulas ordenadas de acordo com a métrica de similaridade. Todas as súmulas que tiverem alguma similaridade com a consulta são retornadas, em grupos de 10 súmulas. Cada um dos parâmetros pode receber um valor percentual entre 0 e 100 indicando qual deve ser o peso de sua participação na métrica de similaridade, mecanismo conhecido por pesos dinâmicos. Desta forma, se o usuário deseja que o texto da súmula tenha o dobro do peso dos demais parâmetros, por exemplo, o usuário estabelece o valor 100 para o texto e 50 para os demais. O recurso dos pesos dinâmicos é útil principalmente nos casos das súmulas que tratam de mais de um tema. Também, as súmulas podem ser acessadas diretamente, através de seu número e do tribunal, sem a utilização da métrica de similaridade.

Além da recuperação, o AlphaThemis realiza a mineração das súmulas, extraindo estatísticas que são acessadas através de perguntas pré-definidas. Por exemplo, pode-se fazer a pergunta “Quantas súmulas foram publicadas por ramo do direito em um determinado tribunal?” e o sistema gera um gráfico com a distribuição das súmulas de acordo com o ramo do direito no tribunal selecionado.

A grande modificação na construção da base de conhecimento do AlphaThemis é a inserção dos termos conexos. Tais termos são relacionados diretamente ao documento, expandindo o conteúdo da súmula para que possa ser representado os assuntos da súmula. Embora o assunto seja definido pelo indicador temático, é necessário colocar os termos que representam este assunto para que o sistema possa comparar com a consulta feita pelo usuário. A inserção dos pesos para os termos conexos permite que eles possam ser relacionados com mais de um documento, com a métrica de similaridade sendo a responsável pela classificação das súmulas para retorno do sistema.

Já a indexação dos documentos era essencialmente manual, já que as expressões indicativas do documento, bem como os termos conexos, eram inseridos manualmente pelos engenheiros do conhecimento. Esta inserção era feita diretamente na interface de construção do conhecimento, onde todos os atributos de cada súmula eram determinados.

O desenvolvimento e a tecnologia desenvolvida tanto para o Olimpo quanto para o AlphaThemis culminaram no desenvolvimento do KMAI. O KMAI (Knowledge Management with Artificial Intelligence – Gestão do Conhecimento com Inteligência Artificial) é um software para gestão do conhecimento que trabalha com conceitos de inteligência competitiva, inteligência militar e inteligência empresarial e estratégica. O grande diferencial do sistema é o uso de ontologias para classificação dos documentos e suporte para a análise.

Dividido em vários módulos, o KMAI permite a coleta de informações nas mais variadas fontes, desde bancos de dados até sítios na Internet. As informações coletadas são indexadas para que o processo de análise possa ser feito pelo usuário. A análise pode ser feita através de três módulos.

O primeiro é o módulo de análise textual. Nele, o usuário insere um texto em linguagem natural, sem limitação de números de palavras, e o sistema recupera todos os documentos que estão relacionados ao contexto digitado pelo usuário. Este contexto é determinado pelas ontologias, através da utilização dos termos que estão relacionados aos termos do usuário.

O segundo módulo é a análise gráfica, onde os documentos da base são visualizados através de gráficos, utilizando-se critérios como data e fonte. Também, os documentos podem ser visualizados de acordo com sua classificação pelas ontologias, separados em domínios do conhecimento.

O terceiro módulo no qual pode ser feita análise é o de monitoramentos. Nele, o sistema utiliza os argumentos de pesquisa salvos pelo usuário em uma análise textual para a realização de buscas periodicamente. O sistema calcula a variação do retorno de documentos, indicando se o assunto que está sendo analisado está aparecendo mais ou menos nos documentos que vão entrando na base.

Além da coleta automática, o sistema recebe documentos também através de entradas do usuário, via o módulo de notas informativas. Estas permitem que documentos produzidos pelos usuários sejam inseridos no sistema para futuras análises. Na teoria de inteligência, as notas informativas podem conter qualquer tipo de relato que seja útil para a organização e cujas informações devem ser armazenadas para futura utilização.

A representação do conhecimento contido no KMAI é feita através do editor de ontologias. Este módulo permite que o conhecimento seja estruturado através de termos e relações entre os termos. Como termos entende-se uma ou mais palavras que formam um conceito. Por exemplo, “narcotráfico”, “tráfico de drogas”. Estes termos estão relacionados entre si através de seis tipos diferentes de relações:

Sinônimo: Indica que os dois termos possuem o mesmo significado no domínio do conhecimento que está sendo trabalhado. Exemplo: “narcotráfico” é sinônimo de “tráfico de drogas”;

Conexos: Indica que os termos estão relacionados semanticamente dentro do domínio. Exemplo: “narcotráfico” é conexo a “tráfico de armas”;

Tipo de: Indica que um termo representa uma especialização do outro termo. Exemplo: “cocaína” é um tipo de “droga”;

Tipo disso: Indica que um termo representa a generalização do outro termo. Exemplo: um tipo de “droga” é “cocaína”;

Parte de: Indica que um termo representa uma parte do outro termo que está relacionado. Exemplo: “roda” é parte do “carro”;

Parte disso: Indica que um termo representa o todo onde está contido o outro termo. Exemplo: “carro” é o todo da “roda”.

Dois termos possuem apenas um tipo de relação dentro de um domínio, mas podem estar relacionados de maneira diferente em domínios diferentes.

Uma grande diferença entre os processos de representação do conhecimento do AlphaThemis para o KMAI é que o primeiro relacionava os termos diretamente aos documentos. Ou seja, os termos eram conexos aos documentos. Para as súmulas, esta representação era fácil, já que o volume de súmulas compensava o estudo das mesmas para poder fazer este relacionamento direto.

Entretanto, o KMAI não podia utilizar este artifício, haja vista que os documentos vinham de fontes diferenciadas, representando domínios que não são conhecidos a priori. Desta forma, no KMAI os termos são relacionados diretamente entre si. A indexação indica quais são os termos da ontologia que existem no documento. Quando o usuário faz a recuperação destes documentos, o sistema se encarrega de procurar por todos os termos que estão relacionados ao termo que o usuário digitou e recuperar os documentos que os contêm. Neste caso, as relações representam pesos diferentes para a métrica de similaridade e esta é responsável por ordenar os documentos para a resposta. Em resumo, a expansão do conhecimento no AlphaThemis era feita no momento da indexação, enquanto no KMAI este processo é feito no momento da recuperação.

Na primeira versão do KMAI, os termos eram organizados dentro de uma árvore de temas e subtemas. Não era possível a criação de outros níveis na hierarquia, além dos termos e das relações só poderem pertencer aos subtemas. Também, não era permitida a existência de termos na base de conhecimento que não tivessem relações com outros termos.

A subdivisão em apenas dois níveis não era suficiente para a representação do conhecimento em organizações complexas, que trabalham com diversos ramos do conhecimento. Assim, na segunda versão do KMAI, implantou-se uma nova representação do conhecimento, formada por uma árvore de domínios. Nesta árvore, podem-se criar quantos níveis forem necessários para a organização, dispondo os mesmos de maneira hierárquica para o usuário.

Esta nova forma de organização possibilita ao usuário organizar os termos de maneira mais ampla, sem a necessidade de prender todos eles a apenas dois níveis de hierarquia. Além disto, a nova versão também permite que o usuário insira termos nos domínios sem a necessidade de relacionar estes termos com outros termos. Os tipos das relações permanecem os mesmos da versão anterior.

Embora existam mecanismos poderosos para a representação do conhecimento, a identificação de quais termos fazem partes dos domínios ainda é um processo que exige uma quantia razoável de trabalho manual, principalmente no início do processo, quando é feita a identificação dos domínios e de quais são os documentos que representam a base para a extração do conhecimento do sistema. Estes processos manuais tornam a construção de uma nova ontologia um processo demorado e custoso, necessitando de um grande número de pessoas para a construção das ontologias.

Na tentativa de determinar o que são termos e como os domínios podem ser formados, realizou-se um estudo para verificar qual é a composição dos termos de um dicionário de ontologias padrão e como estes termos são formados. Este estudo compreendeu a análise de 1.042 termos cadastrados no sistema KMAI (WBSA, 2006), termos estes pertencentes a 17 domínios de

conhecimento, visando descobrir o número de palavras que compõem um termo e qual a classe gramatical a que pertencem as palavras do termo.

Um termo é qualquer entrada do dicionário de ontologias. Ele pode ser composto de uma ou mais palavras. Exemplos de termos existentes no dicionário analisado, com e sem *stop words* podem ser vistos na Tabela 1.

N° Palavras	Termos com stop words	Termos sem stop words
1	Biocombustível Ecoturismo	Biocombustível Ecoturismo
2	Combustível natural Crescimento populacional	Combustível natural Crescimento populacional
3	Deterioração dos rios Poluição do ar	Deterioração rios Poluição ar
4	Abastecimento de água potável Aumento populacional das cidades	Abastecimento água potável Aumento populacional cidades
5	Controle da degradação do solo Valor econômico dos recursos naturais	Controle degradação solo Valor econômico recursos naturais
6	Concentração de gases poluentes na atmosfera Universalização do acesso à energia elétrica	Concentração gases poluentes atmosfera Universalização acesso energia elétrica
7	Concentração de dióxido de carbono na atmosfera Sistema de coleta e tratamento do lixo	Concentração dióxido carbono atmosfera Sistema coleta tratamento lixo
8	Ordenação e controle do uso do solo urbano	Ordenação controle uso solo urbano

Tabela 1. Exemplos de termos por número de palavras

O dicionário, entretanto, não possui uma divisão igualitária entre o número de termos e o número de palavras por termo. O gráfico da Figura 5 mostra a distribuição do número de termos de acordo com o número de palavras.

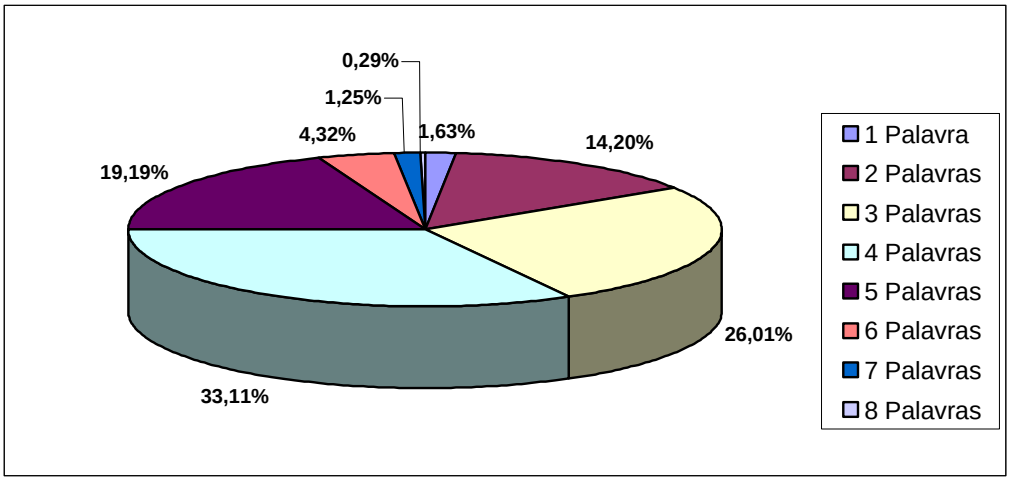


Figura 5. Gráfico com o número de palavras por termo (com *stop words*)

O teste demonstra que os termos possuem entre duas e cinco palavras, sendo que, removidas as *stop words*, o número de palavras estabiliza-se em duas ou três, com a maioria ficando neste último valor, como pode ser visto na Figura 6. Esta remoção pode ser considerada uma

transformação significativa, já que apenas 20,5% dos termos não as possuem e, quando o número de palavras é maior que dois, apenas 4,6%.

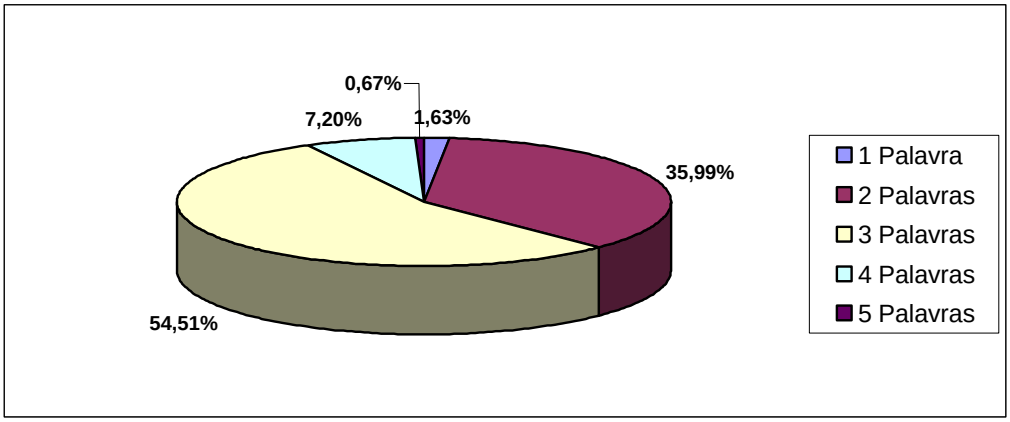


Figura 6. Gráfico com o número de palavras por termo (sem *stop words*)

A intenção deste experimento é demonstrar que os termos são compostos por poucas palavras realmente úteis e as *stop words* são elementos de composição altamente presentes no texto.

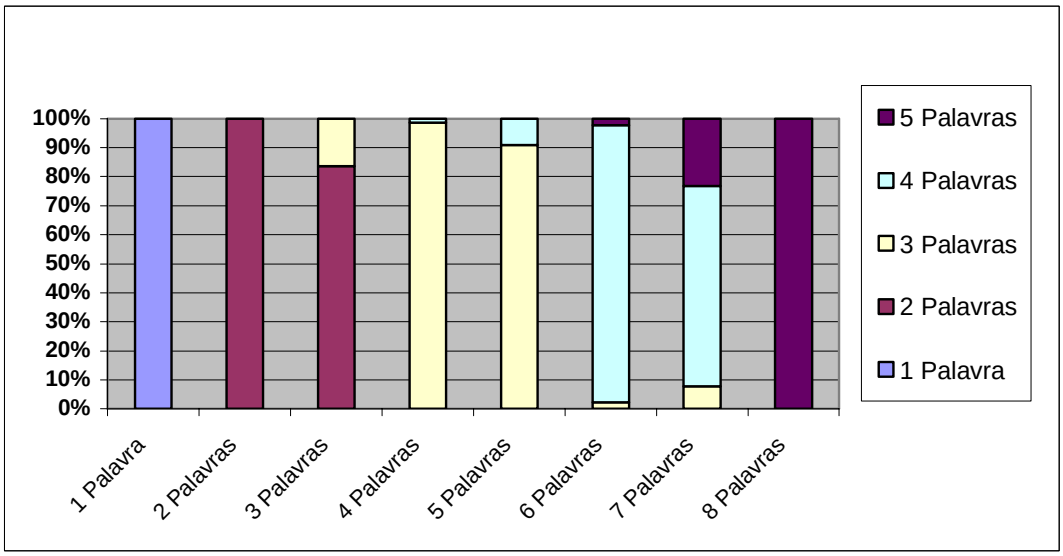


Figura 7. Número de palavras que não são *stop words* nos termos

O segundo teste realizado foi a verificação da classe gramatical das palavras que compõem os termos. A intenção deste teste é verificar quais elementos deveriam ser procurados na sentença e como eles estão relacionados. No total, os 1.042 termos contém 3.891 palavras, com uma

média de 3,73 palavras por termo. A Figura 8 demonstra a divisão dos termos por classe gramatical e a Tabela 2 demonstra os dados levantados no teste.

	1 Palavra	2 Palavras	3 Palavras	4 Palavras	5 Palavras	6 Palavras	7 Palavras	8 Palavras	Total
Substantivos	17	150	505	692	580	133	47	12	2136
Preposições	0	0	225	338	372	85	30	8	1058
Adjetivos	0	145	76	347	36	44	5	3	656
Verbos	0	1	5	0	2	3	1	0	12
Conjunções	0	0	1	2	7	4	3	1	18
Artigos	0	0	1	0	3	1	1	0	6
Numeral	0	0	0	1	0	0	0	0	1
Pronomes	0	0	0	0	0	0	1	0	1
Locução	0	0	0	0	0	0	3	0	3
Total	17	296	813	1380	1000	270	91	24	3891

Tabela 2. Dados da análise da classe gramatical de palavras por termo e número de palavras

Pode-se observar tanto na Figura 8 quanto na Tabela 2 que os termos são basicamente formados por substantivos. De fato, todos os termos contém pelo menos um substantivo, como visto na Figura 9. A medida que vai aumentando o número de palavras no termo, observa-se o aumento da participação dos adjetivos e das preposições. Destaca-se que, no estudo, não foi feita a análise considerando locuções adjetivas, ou seja, os substantivos que, ligados a outros substantivos por preposição, fazem a função de adjetivo. O item Locução que aparece nos gráficos e na tabela correspondem a uma locução adverbial.

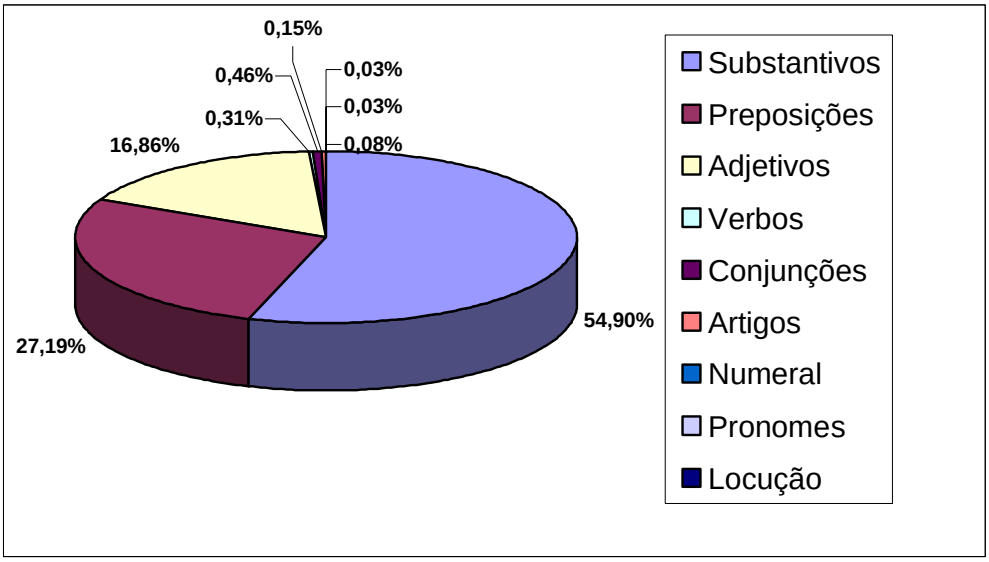


Figura 8. Percentual de palavras por classe gramatical

A Figura 9 mostra claramente a importância de determinar quais são os substantivos nas sentenças e mantê-los como índices do documento. Afinal, todos os termos possuem como elemento principal um substantivo. Mas, ela também mostra que as preposições têm um papel fundamental na formação dos termos. Afinal, nos maiores grupos de termos, que são os de 3 e 4 palavras, as preposições participam na ampla maioria deles, participando de todos os termos com mais de 4 palavras. O fato de não haver preposições em termos com menos de 2 palavras é esperado, haja vista que a função delas na língua é de ligação entre palavras. A Tabela 3 mostra a participação das classes gramaticais por termo.

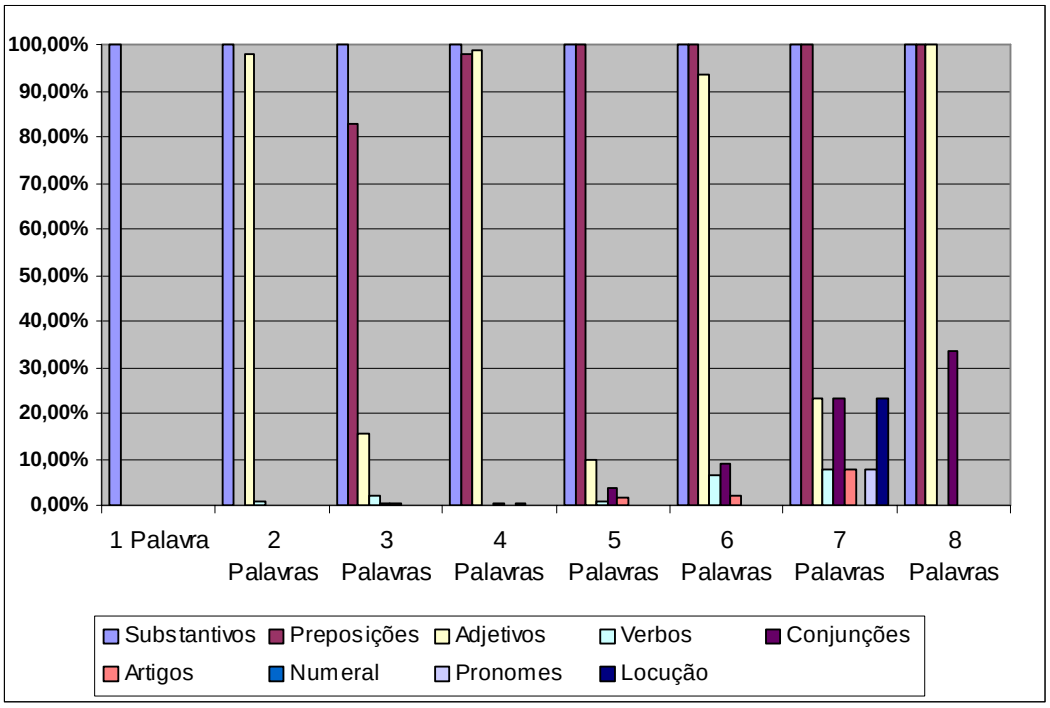


Figura 9. Classes gramaticais de acordo com o número de palavras do termo

	1 Palavra	2 Palavras	3 Palavras	4 Palavras	5 Palavras	6 Palavras	7 Palavras	8 Palavras
Substantivos	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
Preposições	0,00%	0,00%	83,03%	97,97%	100,00%	100,00%	100,00%	100,00%
Adjetivos	0,00%	97,97%	15,50%	98,84%	10,00%	93,33%	23,08%	100,00%
Verbos	0,00%	0,68%	1,85%	0,00%	1,00%	6,67%	7,69%	0,00%
Conjunções	0,00%	0,00%	0,37%	0,58%	3,50%	8,89%	23,08%	33,33%
Artigos	0,00%	0,00%	0,37%	0,00%	1,50%	2,22%	7,69%	0,00%
Numeral	0,00%	0,00%	0,00%	0,29%	0,00%	0,00%	0,00%	0,00%
Pronomes	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	7,69%	0,00%
Locução	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	23,08%	0,00%

Tabela 3. Participação das classes gramatical nos termos conforme o número de palavras

Com este estudo, pode-se concluir que a extração dos termos a partir dos documentos pode ser feita através da criação de relacionamentos entre as palavras. Tais relacionamentos são feitos de acordo com a função das palavras na frase, seguindo o conceito da função sintática mas não a utilizando como elemento de referência.

4 MODELO PROPOSTO

A extração de conceitos é um processo que ultrapassa as características tradicionais das ferramentas de Recuperação de Informação. Tais ferramentas utilizam-se de técnicas de indexação que consideram as palavras isoladamente. Algumas mais sofisticadas mantém consigo a distância entre as palavras, permitindo o reconhecimento de consultas envolvendo mais de uma palavra, correspondendo a expressões do idioma. Por exemplo, a expressão “banco de dados” precisa de um mecanismo de aproximação de palavras para que documentos que contenham unicamente as palavras “banco” ou “dados” não retornem como resultados.

Além disto, a maioria dos mecanismos possuem seus algoritmos de tratamento de linguagem natural, utilizados para a identificação das palavras, voltados para o inglês, haja vista que ele é o idioma predominante na Internet. A Língua Portuguesa (e as línguas latinas em geral, como espanhol, italiano, francês, etc.) possui uma característica altamente flexional, gerando diversas variações para a mesma palavra, de acordo com o significado. Algoritmos de tratamento voltados para o inglês tendem a não obter bons resultados, devido às suas características de sintaxe rígida e morfologia fraca.

A construção de um mecanismo voltado para o português pode parecer estranha, devido à pequena participação do idioma na Internet. Dados de novembro de 2004, fornecidos pelo sítio Global Reach (2004), dizem que 3,5% dos usuários da Internet falam este idioma. Entretanto, 82% dos usuários brasileiros preferem acessar páginas no seu idioma nativo, de acordo com o sítio Today Translations (2005), citando pesquisa Nielsen Associados. Desta forma, pode-se utilizar o português como base de pesquisa para um sistema de linguagem natural.

O modelo proposto pretende estender a indexação não controlada por palavras para compreender a indexação que utiliza vocabulário controlado, principalmente a que utiliza ontologias para a representação do documento. Para isto, ele armazena os relacionamentos entre as palavras que compõem o texto do documento, além das palavras isoladamente. Fazendo isto, o sistema

forma termos com as palavras, possibilitando a contextualização dos documentos. Os termos são então relacionados entre si, a fim de agrupar os termos que possuem relação em algum contexto.

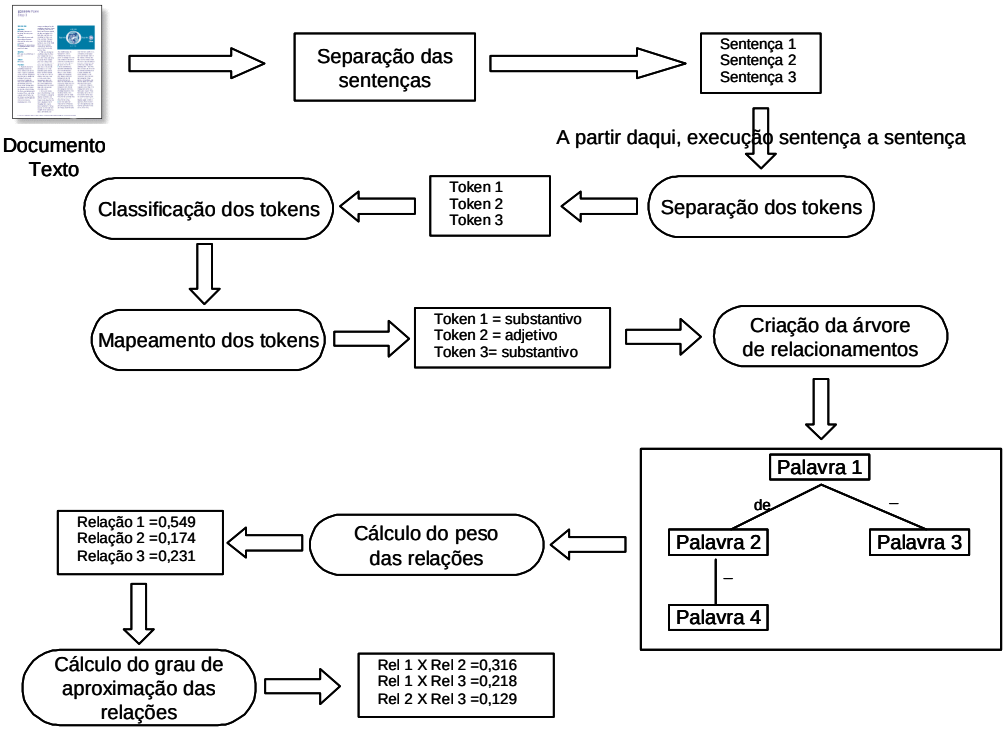


Figura 10. Processo de representação dos documentos

O processo de representação dos documentos é feito de acordo com os seguintes passos, representados na Figura 10:

1) Separação em sentenças;

2) Separação em *tokens*;

3) Classificação dos *tokens*;

4) Mapeamento dos *tokens*;

5) Construção da árvore de relacionamentos;

6) Cálculo dos pesos das relações;

7) Cálculo do grau de aproximação das relações.

4.1 Separação em sentenças

O primeiro passo é a separação do texto em sentenças, identificando as frases que o compõem. Uma frase é qualquer enunciado completo do idioma, contendo uma idéia expressa. A frase pode ter uma ou mais palavras, com ou sem verbo. Elas podem ser identificadas no texto através dos sinais de pontuação que indicam a finalização do pensamento. Tais sinais são o ponto final (.), o ponto de exclamação (!) e o ponto de interrogação (?). Em alguns casos especiais, outros sinais também representam o final da frase, destacando-se o ponto-e-vírgula (;) e os dois pontos (:). Entretanto, como são exceções, os poucos casos podem ser ignorados e tratados como sendo uma única frase. Para manter a conformidade com as tecnologias que baseiam o modelo, as frases passam a ser doravante denominadas sentenças.

A identificação das palavras e a construção da árvore são processos semelhantes à análise sintática. Desta forma, os passos têm que ser executados para cada sentença em separado.

4.2 Separação dos *tokens*

A análise da sentença inicia-se pela separação dos seus *tokens*. De acordo com a definição da Wikipedia (2006a), um *token* é qualquer elemento atômico dentro de uma seqüência de caracteres. Eles podem ou não ter um significado para os seres humanos, mas devem ser úteis para o processamento destinado.

O critério básico para a localização de um *token* é a existência de um caractere não-alfanumérico na seqüência. Assim, a frase “Onde fica a rua 6?” possui os *tokens* “Onde”, “fica”, “a”, “rua”, “6” e “?”. Espaços em branco normalmente são excluídos, mas não existem impeditivos de sua inclusão na lista de *tokens*.

Entretanto, critérios extras normalmente precisam ser adicionados para a determinação dos *tokens*, seguindo necessidades da aplicação. Um exemplo bastante comum é a utilização dos separadores em numerais. O número

“6.438,96” tem que ser interpretado como sendo um *token* único. Assim, é necessário um critério que indique que os sinais de pontuação “.” e “,” não interrompem uma seqüência de caracteres numéricos desde que exista outro caractere numérico depois deles. Outro exemplo são palavras compostas, que possuem hífen, tais como “guarda-chuva”, que também devem possuir uma regra que permita que a seqüência seja reconhecida como um *token* único.

Considerando tais critérios, podemos melhor separar os *tokens*, permitindo que da frase “O guarda-chuva custa R$ 10,00.” sejam extraídos os *tokens* “O”, “guarda-chuva”, “custa”, “R$”, “10,00” e “.”, e não “O”, “guarda”, “-”, “chuva”, “custa”, “R$”, “10”, “,”, “00” e “.”.

4.3 Classificação dos *tokens*

Uma vez separados os *tokens*, deve ser feita sua classificação. Eles podem ser classificados em quatro categorias diferentes:

- **Números**: toda a seqüência de caracteres numéricos, com ou sem separadores de milhar e/ou decimais. Ex.: “4”, “36”, “11,5”, “1.086”. Como não pretende-se fazer a interpretação dos valores, não serão observadas variações dos separadores, como as existentes entre o português e o inglês (onde o ponto é o separador decimal ao invés da vírgula).
- **Datas**: toda a seqüência de caracteres numéricos com separadores de data e sem espaços em branco na seqüência. São considerados separadores de data os caracteres “/” e “-”. Exemplos.: “25/12/2005”, “25/12/05”, “25-12-2005”. Destaca-se que não é função do modelo identificar se a data possui um valor válido. Portanto, a seqüência “45/14/64” também seria considerada uma data.
- **Sinais de pontuação**: todos os caracteres que representam sinais de pontuação na Língua Portuguesa. Ex.: ponto final (.), vírgula (,), parênteses (()), etc.
- **Palavras**: todas as demais seqüências de caracteres. Assim, permite-se o reconhecimento e a indexação de todos os elementos que

aparecerem no texto, não importando se eles são realmente palavras. Neste passo, não será feita a distinção entre palavras que fazem parte do dicionário e são conhecidas e palavras que são seqüências aleatórias, cujo significado não é compreendido ou possui um contexto específico.

Ao final da classificação, tem-se uma lista de *tokens* com a informação do seu tipo, para que seja feito seu mapeamento.

Por exemplo, a frase “Comprei um guarda-chuva dia 10/03 por R$ 10,00.”, geraria os seguintes *tokens*, com sua classificação:

Comprei : **palavra**
Um : **palavra**
guarda-chuva : **palavra**
dia : **palavra**
10/03 : **data**
por : **palavra**
R$: **palavra**
10,00 : **número**
. : **sinal de pontuação**

4.4 Mapeamento dos *tokens*

O mapeamento dos *tokens* serve para que todos eles sejam identificados de acordo com suas características gramaticais. Isto é feito localizando-os em um dicionário da Língua Portuguesa, obtendo a partir deste a classificação gramatical de cada um. Todas as dez categorias do português são consideradas: substantivos, adjetivos, artigos, pronomes, verbos, numerais, advérbios, preposições, conjunções e interjeições. Os números, as datas e os sinais de pontuação serão mapeados como tais.

Quanto mais completo for o dicionário, mais palavras ele terá e também, mais acepções para cada palavra. Isto gera um problema relacionado a múltiplas classificações que uma palavra pode ter. Um exemplo é a palavra “meio”. Ela pode ser classificada em cinco categorias gramaticais: numeral (0,5=meio); substantivo masculino singular (o meio da roda); substantivo masculino plural (os fins justificam os meios); adjetivo (meio caminho); e advérbio (dia meio quente).

Este problema só pode ser relacionado através da contextualização da palavra, ou seja, verificando quais elementos estão ligados a ela para que possa ser feita a correta classificação. Como a etapa de criação da árvore de relacionamentos executa esta verificação, cada palavra pode ser mapeada neste estágio como pertencente a mais de uma categoria gramatical. Desta forma, “meio” seria mapeada como “numeral”, “substantivo”, “adjetivo” e “advérbio”.

Outro problema que precisa ser contornado são as palavras que não estão no dicionário. Isto pode acontecer devido a vários fatores. O primeiro caso são os nomes próprios. Normalmente, os dicionários não contém os nomes de pessoas, obrigando a criação de um dicionário específico para armazenamento de nomes próprios de maior freqüência na população, o que facilita a identificação. Entretanto, é conhecida a criatividade do povo para nomear seus filhos, o que praticamente impossibilita a criação de um dicionário de nomes próprios completo. Na escrita formal e até mesmo coloquial, é padrão que os nomes próprios apareçam com letras maiúsculas, fato que pode ser utilizado para a identificação dos mesmos. Todos os nomes próprios são substantivos por natureza, recebendo assim esta classificação.

O segundo caso em que as palavras não encontram-se no dicionário acontece quando as palavras possuem um erro de digitação ou de ortografia. Os erros de digitação mais freqüentes observados são:

- **Caractere faltante**: um caractere não é digitado em qualquer parte da palavra. Ex.: “envir” ao invés de “enviar”;
- **Caractere extra**: um caractere é acidentalmente digitado na palavra. Ex.: “envioar”, ao invés de “enviar”;
- **Caractere errado**: um caractere digitado no lugar de outro. Ex.: “enbiar” ao invés de “enviar”;
- **Caracteres trocados**: dois caracteres que são invertidos durante a digitação. Ex.: “envira” ao invés de “enviar”;
- **Ausência de acentos**: o digitador ignora os acentos e a cedilha. Ex.: “colecao” ao invés de coleção. Este erro é menos comum quando

consideramos textos publicados, tais como artigos, notícias de jornal e páginas que contém conteúdo explicativo.

A solução deste problema é aplicar um algoritmo de correção. Diversas técnicas de correção estão disponíveis. A mais simples é a força bruta, que cria todas as palavras possíveis considerando os erros anteriormente citados e verifica quais dos resultados são palavras corretas. Esta técnica tem uma boa taxa de acerto, haja vista que a palavra correta sempre está no conjunto de resultados. Entretanto, seu desempenho é questionável, haja vista que uma simples palavra de cinco letras gera 295 palavras candidatas para a correção (sem considerar as vogais que podem ser acentuadas e o ç). Outras técnicas utilizam heurísticas, inserindo regras que consideram proximidade de teclas ou erros comuns do idioma (ex.: ausência de um “r” no dígrafo “rr”), entre outras. Existem também regras probabilísticas, que calculam a similaridade entre palavras e estabelecem que aquelas com similaridade maior que um limite mínimo são consideradas corretas.

O terceiro caso de palavras ausentes são as palavras flexionadas morfologicamente. Tais palavras são as formas plurais, verbos conjugados, variações de gênero, aumentativos e diminutivos. Estas formas não são consideradas nos dicionários, exceto quando seu significado é diferente da forma singular. Duas abordagens podem ser adotadas neste caso. A primeira é inserir todas as formas possíveis da palavra no dicionário, fazendo as devidas associações para que a palavra seja reconhecida como sendo uma palavra correta. A outra é incluir regras de substituição de sufixos, permitindo o reconhecimento das formas de plural, terminações de conjugações verbais, etc. Exemplo de regra: “as palavras terminadas em ‘is’ correspondem ao plural das palavras terminadas em ‘l’, salvo as exceções.”

O último caso são as seqüências que não podem ser classificadas como nenhuma das anteriores. Classificar tais seqüências depende de um estudo experimental, verificando casos e elaborando regras para seu tratamento. Para o escopo do trabalho, todas as seqüências que não puderem ser identificadas serão consideradas substantivos.

Esta solução decorre do ponto tratado anteriormente relacionado aos nomes próprios. Como é praticamente impossível determinar todos os nomes de pessoas, entidades, etc. que possam aparecer em um documento, é perigoso optar pelo descarte das seqüências não classificadas, porque os nomes próprios podem ser descartados.

A própria correção de palavras é afetada pelo caso dos nomes próprios. Toma-se, por exemplo, o nome “Félix”. Considerando que ele não esteja no dicionário de nomes próprios (ou não exista um), qualquer algoritmo de correção transformaria este nome na palavra “feliz”.

Um teste feito foi extrair 15 frases de textos jornalísticos para avaliação. Estas frases totalizaram 476 *tokens*, dos quais 78 eram sinais de pontuação, 3 números e 395 palavras. Das palavras, 241 eram palavras que estavam no dicionário, 121 eram palavras flexionadas morfologicamente, 32 eram nomes próprios e 1 palavra era um termo estrangeiro, que foi considerada como desconhecida. Não foram encontrados erros de digitação ou ortografia. O gráfico da Figura 11 mostra a distribuição percentual.

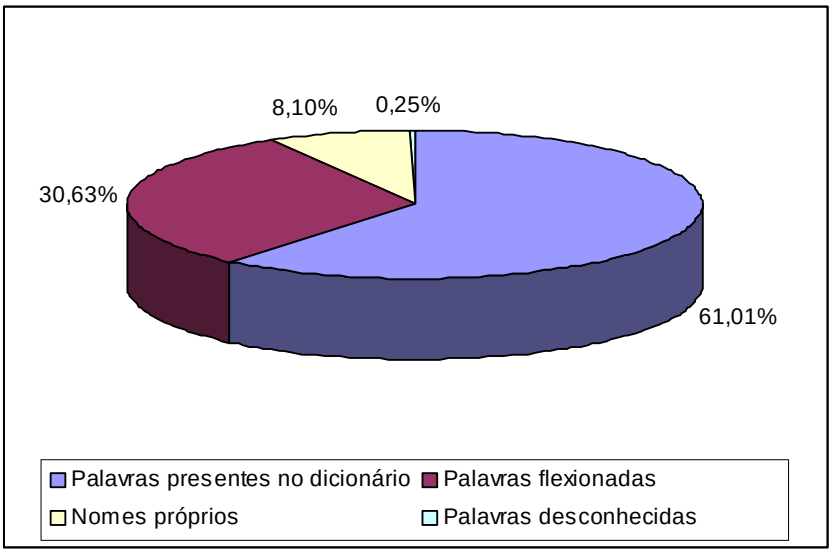


Figura 11. Percentual de palavras presentes no dicionário e causa

Assim, o mapeamento dos *tokens* para que se transformem em palavras é uma tarefa mais árdua para o computador do que aparenta. Os critérios para tratamento da língua são complexos e possuem muitas exceções, o que dificulta encontrar a regra correta.

No modelo, a estratégia adotada para os problemas apresentados baseia-se no fato de que a base de documentos possui um grau de corretude elevado, ou seja, a quantidade de erros de digitação e/ou ortografia pode ser considerada desprezível (como observado no teste realizado). As principais ferramentas de busca adotam estratégias similares. O Google, por exemplo, indexa as palavras mesmo que elas contenham tais erros, permitindo que sejam feitas buscas usando essas palavras. Tais ferramentas sugerem para o usuário a palavra correta, caso seja detectado um erro.

Desta forma, o único tratamento feito nas palavras corresponde ao reconhecimento das formas plurais, conjugação de verbos, etc. num processo denominado normalização. As palavras no plural são reduzidas a sua forma no singular, as palavras no feminino são passadas para o masculino (quando cabível) e os verbos são reduzidos a seu infinitivo. Isto permite a identificação da grande maioria dos casos de palavras não encontradas. O segundo grupo mais observado de palavras não encontradas corresponde ao de nomes próprios. Considerando a baixa quantidade de erros de ortografia, todos as palavras que não foram encontradas após a normalização são consideradas candidatas a nomes próprios, recebendo assim, o mapeamento como substantivos.

O resultado do mapeamento dos *tokens* é uma lista com o *token* e as possíveis palavras candidatas para a sua representação, bem como a classificação gramatical de cada uma.

Por exemplo, considerando a frase: “A casa de Roberval pegou fogo dia 25”, o resultado será a seguinte lista de *tokens*:

a = {(a; artigo); (a; pronome); (a; substantivo)}
casa = {(casa; substantivo); (casar; verbo)}
de = {(de; preposição)}
Roberval = {(Roberval; substantivo)}
pegou = {(pegar; verbo)}
fogo = {(fogo; substantivo)}
dia = {(dia; substantivo)}
25 = {(25; número)}

4.5 Construção da árvore de relacionamentos

Embora a lista montada no mapeamento dos *tokens* já possa ser usada para a recuperação dos documentos, o processo de extração de conceitos ainda não pode ser adequadamente realizado apenas com ela. Assim, optou-se por relacionar as palavras entre si de forma a encontrar os conceitos que estão expressos na sentença.

O relacionamento entre as palavras é um processo que pode ser feito através de análise sintática. Vários modelos de análise sintática estão disponíveis e podem ser utilizados. Existe também uma tecnologia que além da análise sintática, realiza também a análise semântica das sentenças, que é a UNL. Entretanto, a UNL não possui, atualmente, um conjunto de regras que permita a geração das sentenças a partir do português. A construção das regras é um trabalho a parte, devido ao volume de variações apresentados no idioma, os quais deveriam ser tratados para a geração das sentenças UNL. Também, muitas das relações que a UNL disponibiliza ainda não foram devidamente mapeadas para seus correspondentes em português, o que dificulta o trabalho de reconhecimento.

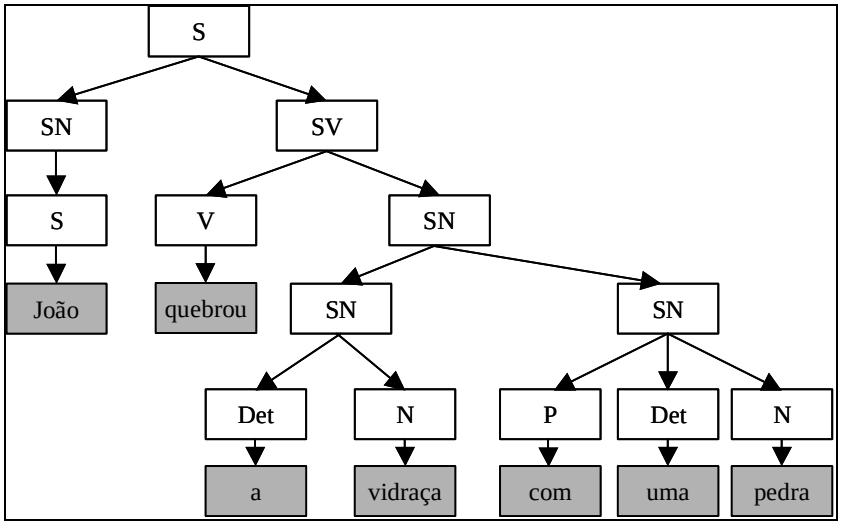


Figura 12. Árvore sintática

Desta forma, a solução encontrada foi a construção de um modelo de relacionamentos que permitisse fazer um mapeamento nas sentenças e a construção de uma árvore de relacionamentos similar a uma árvore sintática.

Dentro de uma árvore sintática, as palavras estão relacionadas de acordo com sua função sintática, ou seja, representa-se quem é o sujeito da frase, o predicado, etc. Por exemplo, a frase “João quebrou a vidraça com uma pedra” teria a árvore sintática mostrada na Figura 12. O processo selecionado segue também a teoria da gramática de Chomsky.

Entretanto, construir um mecanismo que realize a análise sintática e encontre as funções de cada palavra é uma tarefa extremamente complexa, existindo poucos sistemas que geram resultados satisfatórios.

O estudo apresentado no capítulo três permite concluir que a extração dos termos a partir dos documentos pode ser feita através da criação de relacionamentos entre as palavras. Tais relacionamentos são feitos de acordo com a função das palavras na frase, seguindo o conceito da função sintática mas não a utilizando como elemento de referência.

Desta forma, optou-se pela utilização das preposições como os elementos de ligação das palavras nos relacionamentos que podem gerar termos. As palavras que estiverem ligadas diretamente, sem preposição, será utilizado um caractere especial para identificar o relacionamento.

Mesmo sem a intenção de determinar a função sintática das palavras, é necessária a existência de algumas regras para identificar a qual palavra uma outra está relacionada. Por exemplo, na frase “A casa grande amarela”, ambos os adjetivos (“grande” e “amarela”) estão relacionados a “casa”. Este conjunto de regras não realiza a análise sintática propriamente dita. Ele apenas atende as necessidades do modelo.

A forma de representação dos relacionamentos é baseada na sintaxe da UNL. Ela é composta por um rótulo indicando o tipo da relação e duas palavras universais, cada qual com seus respectivos atributos. Um exemplo de relação UNL é “agt(run(icl>do).@entry,car(icl>thing).@pl)”, onde “agt” é o rótulo da relação; “run(icl>do)” e “car(icl>thing)” são as palavras universais; e @entry e @pl são os atributos das palavras universais.

No modelo, a relação possui a mesma sintaxe. Entretanto, os elementos que compõem a relação não foram utilizados.

Os rótulos das relações foram substituídos pelas preposições que ligam as palavras. Eles não são utilizados porque, na UNL, representam características morfossintáticas da linguagem. Em outras palavras, para determinar qual relação UNL está presente na sentença é necessário fazer a análise sintática da mesma, tarefa a qual já foi demonstrada como complexa e cujo custo é maior que os benefícios que ela traria para o modelo.

Da mesma forma, as palavras universais da UNL não são utilizadas. Elas são substituídas pelas palavras normalizadas do dicionário de português. Primeiro, porque as palavras universais possuem como premissa a existência de um cabeçalho em inglês, o que implicaria na tradução e utilização de um dicionário de outro idioma juntamente com o dicionário de português. Segundo, porque as restrições das palavras universais correspondem a sua hierarquia na base de conhecimento da UNL, cuja estrutura é complexa e, até a versão do *Enconverter* testada, não era suportada. Assim, optou-se pela utilização direta das palavras em português, simplificando a construção do dicionário de entradas.

Os atributos das palavras universais da UNL têm como principal função a descrição da subjetividade da sentença a partir do ponto de vista do emissor. (UNDL, 2004). Além destes, existem alguns atributos que são utilizados para indicar variações nas sentenças, utilizados como simples convenções, tais como plural, voz passiva, etc. Assim, os atributos indicadores de plural (@pl), de abrangência de conceito (definido = @def, indefinido = @indef, negação = @not) e de tempo (passado = @past, presente = @present, futuro = @future) foram usados. Também, foi criado um atributo que é necessário para o português e, devido a sua natureza genérica, a UNL não possui, que é o atributo de gênero (masculino = @masc, feminino = @fem). Este atributo é utilizado principalmente em adjetivos cuja forma varia de acordo com a concordância.

Assim, no modelo, a estrutura da relação é a seguinte:

relação(*palavra1. atributos, palavra2.atributos*),

onde *relação* é a preposição que liga as duas palavras ou o caractere "_", indicando que as duas palavras estão diretamente ligadas, sem

preposição; *palavra1* e *palavra2* são as duas palavras do dicionário que estão sendo relacionadas; e *atributos* é a lista com os atributos daquela palavra na sentença.

O documento fica então estruturado como sendo uma seqüência de sentenças, onde cada sentença é formada por um conjunto de relações. Para descrever a estrutura, utiliza-se o padrão XML (eXtensible Markup Language). Um exemplo de documento é o seguinte:

Texto do documento original: “O presidente da Autoridade Palestina, Yasser Arafat, morreu ontem em Paris. Seu corpo será transportado para o Cairo, para as homenagens fúnebres. O enterro será em Hamalah, na Cisjordânia.”

```
<documento id=346>
        <sentenca id=1>
                de(presidente.@def, Autoridade.@def)
                _(Autoridade.@def, Palestina)
                _(presidente.@def, morrer.@past)
                _(morrer.@past, ontem)
                em(morrer.@past, Paris)
                _(Yasser,Arafat)
        </sentenca>
        <sentenca id=2>
                _(corpo, seu)
                _(corpo, ser.@future)
                _(ser.@future, transportado)
                para(transportado, Cairo.@def)
                para(transportado, homenagem.@pl.@def)
                _(homenagem.@pl.@def, fúnebre.@pl)
        </sentenca>
        <sentenca id=3>
                _(enterro.@def, ser.@future)
                em(ser.@future, Hamalah)
                em(Hamalah, Cisjordânia.@def)
        </sentenca>
</documento>
```

4.6 Cálculo dos pesos das relações

Os sistemas tradicionais de Recuperação de Informação calculam pesos para os índices a fim de utilizá-los posteriormente como parâmetros para a recuperação. Os principais valores calculados são a freqüência de termos (*tf*) e a freqüência inversa de documentos (*idf*), bem como o peso do termo no documento, calculado pela fórmula *tf*idf*.

O modelo proposto também prevê o cálculo de métricas para determinar quais relações são realmente conceitos e quais relações são

elementos do idioma, aparecendo nos documentos apenas para fazer a concordância entre idéias ou para melhorar o estilo da escrita.

Os mesmos valores serão calculados para cada relação. Desta forma, as relações possuem sua freqüência para cada documento e possuirão uma freqüência inversa de documentos. Estes pesos são calculados porque, no modelo, as relações passam a desempenhar o papel de índices do documento, além das palavras. Como as fórmulas são para determinar o peso dos índices, serão feitas as devidas substituições nas fórmulas tradicionais de Recuperação de Informação. Desta forma, a freqüência da relação (*rf*) é o número de vezes que a relação aparece no documento, dividido pelo número de relações do documento, de acordo com a fórmula:

$rf_i = \frac{n_i}{\sum_k n_k}$, onde n_i é o número de vezes que a relação *i* aparece no documento e n_k o número de vezes que cada uma das *k* relações aparecem no documento.

Já a freqüência inversa do documento da relação é calculada pela fórmula:

$idf_i = \log_2(\frac{N}{n_i})$, onde *N* é o número de documentos da coleção e n_i é o número de documentos em que a relação *i* aparece.

Por exemplo, um documento contém 150 palavras, estas formando 40 relações. Uma das relações é *de(tráfico, droga.@pl)*, que aparece duas vezes. O *rf* desta relação é igual a 2/40 = 0,05. Supondo que a base de documentos tem 100.000 documentos, e a relação *de(tráfico, droga@pl)* aparece em 100 documentos, o *idf* da relação é 2,5. O valor de *rf*idf* é igual a 0,125.

4.7 Cálculo do grau de aproximação das relações

O último estágio do modelo é calcular o grau de proximidade entre as relações, a fim de agrupar as possíveis relações que formam um contexto. Na construção de ontologias, um dos fatores considerados mais importantes para

estabelecer os relacionamentos entre os termos é exatamente a presença dos termos em um mesmo documento. Conforme vai aumentando o número de documentos com ambos os termos, significa que maior é a relação entre estes termos.

Salton e McGill (1983) apresentam a co-ocorrência estatística de palavras onde os coeficientes de similaridade são obtidos entre pares de termos distintos baseados em coincidências nas associações de termos para os documentos da coleção. Analogamente, pode-se fazer a transposição para as relações. Substituindo os termos na matriz de associações do modelo de espaço temporal por relações, constrói-se uma nova matriz, como mostrado na Tabela 4. As linhas da matriz passam a representar os vetores individuais dos documentos e as colunas identificam as associações das relações aos documentos.

	R_1	R_2	...	R_k	...	R_m
D_1	rf_{11}	rf_{12}	...	rf_{1k}	...	rf_{1m}
...	...	...	...	...	...	...
D_n	rf_{n1}	rf_{n2}	...	rf_{nk}	...	rf_{nm}

Tabela 4. Matriz de associação de relações

Assim, da mesma forma que no cálculo do termo, a similaridade entre a relação *k* e qualquer relação *l* pode ser medida baseada nos respectivos pares de colunas da matriz:

$$sim(\mathrm{REL}_k, \mathrm{REL}_l) = \frac{\sum_{i=1}^{n} rf_{ik} \cdot rf_{il}}{\sum_{i=1}^{n} rf_{ik}^{\ 2} + \sum_{i=1}^{n} rf_{il}^{\ 2} - \sum_{i=1}^{n} rf_{ik} \cdot rf_{il}}$$

dados os vetores de termos na forma de REL_k = (rf_{1k}, ...,) onde rf_{ik} indica a freqüência de REL_k no documento *i*, assumindo *n* documentos na base. Como resultado, é computado um vetor de associação rel_k-rel_{Tk}, expressando a similaridade da relação *k* com cada relação *l* através de $sim(REL_k,REL_l)$.

Assim, as relações que aparecerem juntas nos documentos serão relacionadas como mais prováveis de pertencerem ao mesmo contexto. Relações que porventura não pertençam a nenhum contexto, ou pertençam a

muitos, terão valores de associação com muitas outras relações, mas todas com valores proporcionalmente baixos. Os grupos de conceitos que tiverem um grau de proximidade alto são agrupados como um contexto.

Por exemplo, considere-se as relações dispostas nas colunas abaixo com os seus respectivos *rf* para cada documento, dispostos nas linhas (Tabela 5).

	(1)	(2)	(3)	(5)	(6)	(10)
D1	1	1	1	1	1	1
D2	0	1	1	0	0	0
D3	0	1	1	0	0	0
D4	0	0	1	0	0	1
D5	0	0	0	0	1	0

Tabela 5. Matriz de exemplo para relações e documentos

A Tabela 6 mostra o grau de similaridade entre as relações baseado nos dados da Tabela 5 e calculado de acordo com a co-ocorrência estatística das relações.

	(1)	(2)	(3)	(5)	(6)	(10)
(1)	X	0.33	0.25	1	0.5	0.5
(2)	0.33	X	0.75	0.33	0.25	0.25
(3)	0.25	0.75	X	0.25	0.2	0.5
(5)	1	0.33	0.25	X	0.5	0.5
(6)	0.5	0.25	0.2	0.5	X	0.33
(10)	0.5	0.25	0.5	0.5	0.33	X

Tabela 6. Similaridade entre as relações

5 CONCLUSÕES

O modelo proposto neste trabalho vem de uma abordagem híbrida, utilizando diversas tecnologias para sua formação. Mesmo contendo uma forte base de Recuperação de Informação, a participação das demais tecnologias, principalmente Processamento de Linguagem Natural e RC^2D, são fundamentais para que ele possa ser base de um sistema que interprete os documentos textuais e extraia seus conceitos.

A solução adotada para a representação de um documento prima pela simplicidade. A adoção de um documento XML para o fazê-lo vai de encontro à tendência atual dos softwares existentes, onde tal padrão é adotado com sucesso e em larga escala.

A representação dos textos foi voltada para aqueles escritos em português. Embora a tecnologia adotada, a UNL, permita a representação de textos escritos em qualquer linguagem natural, a ausência de mecanismos adequados para a conversão para nosso idioma obrigou à utilização de uma abordagem paralela, voltada para o português. Mesmo assim, a representação mostra-se extensível, ao menos para as demais línguas latinas, como o espanhol, italiano, etc.

A estrutura derivada da UNL apresenta-se adequada para o português. A utilização das preposições como elementos de ligação entre as palavras torna o mecanismo de conversão mais simples que se fosse necessária a utilização da análise sintática, já que esta é um processo custoso e complicado.

Outro ponto favorável à estrutura adotada no trabalho é que ela permitiu o fácil reconhecimento de termos. O formato de relações binárias permite a visualização rápida de quais palavras estão ligadas entre si, o que auxilia na construção de termos.

Por último, a viabilidade da utilização das métricas de Recuperação de Informação permitiu que fosse criado um conjunto de pesos para determinar quais termos são mais relevantes dentro da base de documentos. Este cálculo de relevância é estendido para permitir a comparação da freqüência conjunta

de documentos, ou seja, quantas vezes cada par de termos aparece juntos. Isto permite criar associações entre os termos, de uma maneira similar à rede semântica utilizada para a formação de ontologias.

Outra função do cálculo entre estes termos é encontrar os contextos existentes na base de documentos. Determinar qual é o termo que representa o contexto (algo como o “nome” do contexto) não foi possível até o momento. Entretanto, o conjunto de termos que formam o contexto pode ser verificado.

Pelos resultados obtidos em simulações, o modelo está apto para ser utilizado em um sistema de recuperação de informação, com boas chances de aumentar o desempenho comparado aos sistemas atuais.

5.1 Trabalhos Futuros

A principal necessidade existente no modelo atual é um protótipo que teste todos os passos em conjunto. Além do teste em conjunto, é necessária uma melhor avaliação dos relacionamentos gerados pelo processo de relacionamento de termos visando a elaboração de contextos.

Neste mesmo ponto, pode ser realizado um trabalho para a avaliação dos termos visando a procura do termo determinante do contexto, ou seja, qual dos termos pode ser considerado o mais importante e o indicador do contexto.

Quando os mecanismos de conversão da UNL estiverem todos completamente finalizados, pode-se estudar a substituição do modelo de representação proposto pelas sentenças UNL. Isto permitiria a utilização do modelo para qualquer idioma.

Mesmo com o modelo atual, deve-se estudar a aplicação desta representação para as demais línguas latinas e também para línguas não latinas, tais como o inglês e o alemão.

6 REFERÊNCIAS BIBLIOGRÁFICAS

AAMODT, A.; PLAZA, E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and Systems Approaches. Artificial Intelligence Communications, Vol. 7, No. 1, 1994.

ARNOLD, Stephen A. The Google Legacy. Capítulo 3. Infonortics. 2005. Disponível em: http://www.infonortics.com/publications/google/ technology.pdf

BORTOLON, Andre ; WANGENHEIM, C. G. v. ; DOMINGOS, Marlon. Uma Abordagem Híbrida para o Gerenciamento de Documentos FAQ em Português. In: 1º Congresso Brasileiro de Computação, 2001, Itajaí. Anais do 1º Congresso Brasileiro de Computação, 2001. v. 1.

BORTOLON, Andre ; WANGENHEIM, C. G. v. ; WANGENHEIM, A. v. A Hybrid Approach for the Management of FAQ Documents in Latin Languages. In: Proceedings of the International Conference on Case-Based Reasoning, 2001. v. 1. Vancouver, 2001.

BUENO, Tânia Cristina D' Agostini; Engenharia da Mente: Uma Metodologia de Representação do Conhecimento para a Construção de Ontologias em Sistemas Baseados em Conhecimento. Tese de Doutorado (Engenharia de Produção). Universidade Federal de Santa Catarina. Florianópolis, 2005.

BUENO, Tânia C. D.; BORTOLON, Andre; HOESCHL, Hugo C.; MATTOS, Eduardo S.; RIBEIRO, Marcelo S. Analyzing the use of Dynamic Weights in Legal Case Based System. In: Proceedings of Ninth International Conference on Artificial Intelligence and Law. Springer Verlag, 2003.

BUENO, Tânia Cristina D' Agostini; HOESCHL, H. C.; BORTOLON, Andre; MATTOS, E. S.; SANTOS, C. S. Knowledge engineering suite: a tool to create ontologies for automatic knowledge representation in knowledge-based systems. In: DEXA - 4th International Conference on Electronic Government - E-GOV2005, 2005, Copenhagen, 2005.

BUENO, Tânia Cristina D' Agostini ; WANGENHEIM, C. G. V. ; HOESCHL, Hugo Cesar ; MATTOS, Eduardo da Silva ; BARCIA, Ricardo Miranda . Retrieval in Jurisprudencial Text Bases using Juridical Terminology. In: International Conference in Inteligence Artificial and Law - ICAIL, 1999, Oslo, 1999.

BURKE, R.; HAMMOND, K.; KULYUKIN, V.; LYTINEN, S.; TOMURO, N; SCHOENBERG, S. Question Answering from Frequently Asked Question Files. AI Magazine, 18(2), 1997.

CLEVERDON, C.W. The Cranfield tests on index language devices. In: Aslib Proceedings, 19, 6, 173-94. Cranfield, 1967.

CLEVERDON, C.W.; MILLS, J.; KEEN, M. Factors Determining the Performance of Indexing Systems. Vol. 1, Design. Vol. 2, *Test Results*. ASLIB Cranfield Project. Cranfield, 1966.

DAVE, Shachi; BHATTACHARYYA, Pushpak. Knowledge Extraction from Hindi Texts. Journal of Institution of Electronic and Telecommunication Engineers, vol. 18, no. 4, Julho, 2001.

FARACO, Carlos Emílio; MOURA, Francisco Marto de; Língua e Literatura. 23ª Edição. v. 2. Ática. São Paulo, 1995a.

FARACO, Carlos Emílio; MOURA, Francisco Marto de; Língua e Literatura. 23ª Edição. v. 3. Ática. São Paulo, 1995b.

Global Reach. Global Internet Statistics (By Language). Disponível em: http://global-reach.biz/globstats/index.php3. Acessado em: 19/01/2006.

HOESCHL, H. C. Sistema Olimpo: tecnologia da informação jurídica para o Conselho de Segurança da ONU. Tese de Doutorado (Engenharia de Produção). Universidade Federal de Santa Catarina. Florianópolis, 2001.

HOESCHL, H. C.; BUENO, T. C. D.; BORTOLON, Andre; BARCELLOS, Vânia; MATTOS, E. S. . Olimpo System Web-Tecnology For Electronic Government

And World Peace. In: 6th International Conference on Enterprise Information Systems, 2004, Porto, 2004.

JONES, K. Sparck. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation, n. 28. pág. 111-121. 1972.

KEEN, E.M.; DIGGER, J.A. Report of an Information Science Index Languages Test. Aberystwyth College of Librarianship, País de Gales, 1972.

KMAI, versão 3.0. Software de Gestão do Conhecimento. [S.I.]: WBSA Sistemas Inteligentes S.A., 2006.

LENZ, Mario. Textual CBR and Information Retrieval -- A Comparison. In: Proceedings of 6th German Workshop on CBR, 1998.

LENZ, Mario; BURKHARD, Hans-Dieter. CBR for Document Retrieval: The FALLQ Project. In: Leake e Plaza. Case-Based Reasoning Research and Development ({ICCBR}-97), Lecture Notes in Artificial Intelligence No. 1266. pág. 84-93. Springer-Verlag. Berlim, 1997.

LENZ, M.; HÜBNER, A.; KUNZE, M. Question Answering with Textual CBR. In: Proceedings of the International Conference on Flexible Query Answering Systems. Denmark, 1998.

LENZ, M.; HÜBNER, A.; KUNZE, M. Textual CBR. In: M. Lenz et al (eds.), Case-Based Reasoning Technology. Lecture Notes in Artificial Intelligence 1400. Springer Verlag, 1998.

RIBEIRO, Marcelo Stopanovski. KMAI, da RC²D à PCE. Gestão do conhecimento com inteligência artificial, da representação do conhecimento contextualizado dinamicamente à pesquisa contextual estruturada. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis, 2003.

RUSSELL, Stuart; NORVIG, Peter. Artificial Intelligence: A Modern Approach. Prentice-Hall. New Jersey, 1995.

SALTON, C.; MCGILL, M. Introduction to Modern Information Retrieval. McGraw-Hill, New York, 1983.

SOARES, António; BARROSO, João; BULAS-CRUZ, José. Estimativa da PIW através de Motores de Pesquisa de Grande Escala. In: Conferência IADIS Ibero-Americana WWW/Internet 2004. Madrid, 2004.

Today Translations. Language usage on Internet. Disponível em: http://www.todaytranslations.com/index.asp-Q-Page-E-Language-Usage-on-the-Internet—99144709. Acessado em: 19/01/2006.

UNDL Foundation. Enconverter Specifications. Version 3.3. 2002.

UNDL Foundation. The Universal Networking Language (UNL) Specifications. Version 3. Edition 3. December, 2004. Disponível em: http://www.undl.org/unlsys/unl/UNLSpecs33.pdf. Acessado em: 02/05/2005.

UNESCO. Cultural and Linguistic Diversity in the Information Society. Unesco, Paris, 2003. Disponível em: http://portal.unesco.org/ci/en/file_download.php/3325a1350642eecaf1a3e417b530f897cultural_diversity.pdf. Acessado em: 31/10/2005.

VAN RIJSBERGEN, C. J. Information Retrieval. Second Edition. Butterworths. London, 1979.

Wikipedia, a Enciclopédia Livre. Enciclopédia On-line. Disponível em: http://www.wikipedia.org.

Wikipedia, a Enciclopédia Livre. Tesauro. Disponível em: http://pt.wikipedia.org/w/index.php?title=Tesauro&oldid=1242901. Acessado em: 10/01/2006a.

Wikipedia, a Enciclopédia Livre. Token. Disponível em: http://en.wikipedia.org/w/index.php?title=Token&oldid=35180260. Acessado em: 11/01/2006b.

YU, C.T.; SALTON, G. Effective information retrieval using term accuracy. Communications of ACM, n. 20. pág. 135-142. 1977.

ANEXO I – ARTIGO PUBLICADO NO ICEIS 2004

O artigo a seguir foi publicado no ICEIS – International Conference on Enterprise Information Systems, no ano de 2004.

Olimpo System WEB-Tecnology for Electronic Government and World peace

Hugo C. Hoeschl, Tânia Cristina D. Bueno, Vânia Barcellos, Andre Bortolon, Eduardo S. Mattos

E-Gov, Juridical Intelligence and Systems Institute – Ijuris
Email: digesto@digesto.net, tania@ijuris.org, bortolon@eps.ufsc.br, mattos@ijuris.org , vania@ijuris.org

Keywords: UN Security Council´s Resolutions, Dynamically Contextualised Knowledge Representation (DCKR), Structured Contextual Search – SCS, Information of Technology, Data retrieve, JAVA

Abstract: The paper describes the Olimpo System, a knowledge-based system that enables the user to access textual files and to retrieve information that is similar to the search context described by the user in natural language. The paper is focused on the innovation recently implemented on the system and its new features. A detailed description is presented about the search level and the similarity metrics used by the system. The methodology applied to the Olimpo system emphasises the use of information retrieval methods combined with the Artificial Intelligence technique named SCS (Structured Contextual Search).

1 INTRODUCTION

Some complex and specific domains require an information retrieval system that is more than just a great technology to search for documents in large text databases. A good knowledge representation is also required.

The present approach enables to retrieve textual information that is similar to the search text described by the user using natural language. Through the extraction of relevant information using DCKR technology (Dynamically Contextualised Knowledge Representation) [8] [9], new documents are automatically included in the knowledge database. Concepts of Case-Based Reasoning (CBR) [1] [2] and information retrieval techniques were applied to obtain a better performance of the system, leading to the technology named Structured Contextual Search – SCS.

The following item 2 of this paper addresses the UN Security Council and the Resolution document; in items 3 and 4 the knowledge representation methodology is presented and Olimpo system is described; in items 5 and 6, describe how the web-technology will be applied to Olimpo system and the impacts that influence not only in the technological field, but also in relation to the citizenship, the knowledge and the research and item 7 is the conclusion of the paper.

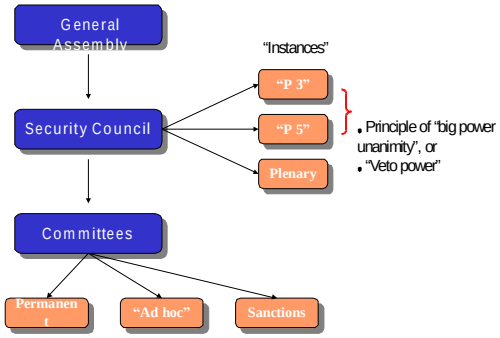
2 ABOUT THE UN SECURITY COUNCIL

The importance of the UN body becomes noticeable when one follows the main global means of communication and no further arguments are required. Being the source of the documents handled by the Olimpo system, it is useful to give more details about the Security Council and its document base.

According to its Charter (Article 7-1), the United Nations Organization (UNO) is comprised of six special bodies. All of them issue relevant documentation and it is highly important to have an adequate tool to retrieve those documents.

Given its characteristics and aspects related to the Resolutions, the Security Council was chosen as application field of the Olimpo system.

The Security Council is described by Article 7-1 of UNO's Charter, where it is referred to as a special body of the United Nations. The Security Council is specifically addressed in Chapter V, Articles 23 to 32. As per Article 24-1, its central function is to assume "the main responsibility in maintaining international peace and security."



Source: www.un.org

Figure 1 - Position of the Security Council

It should be emphasized that the Security Council has a juridical and an executive profile. According to Kelsen (apud Steinfus [11]), it is juridical because it holds the monopoly of legitimate violence at the international scope and judges the existence of facts, determines sanctions on them and who will enforce these sanctions. That turns it a juridical body. And this profile enables a good application of the technology of juridical information, especially SCS and its particular method of rhetoric structure analysis of a given jurisdictional context, based on the knowledge structure involving the body, which maximizes the task of intelligent retrieval of documents when adequate modelling is used.

The Security Council has also political characteristics and it has discritionary power to establish violations, according to Steinfus [11]; therefore the Security Council holds an executive characteristic, turning it a juridical-executive body.

The Security Council presents some peculiarities. One of them is to be currently the most powerful jurisdictional body on the planet. Another one is the existence of internal, informal instances, named "P 3" (Western permanent member countries) and "P 5" (all permanent member countries), according to Steinfus [11]. Another peculiarity is the existence of internal bodies with specific power delegation to perform certain tasks, on a permanent or "ad hoc" level, like the sanctions committee, as shown on Figure 1.

Among the documents issued by the Security Council, six of them have greater relevance. Based on their structure and relevance, the Resolutions were chosen for the application of the Olimpo system.

As per the structure of the document, the Resolutions have some characteristics that make it easier to apply the technology referred to herein.

3 DYNAMICALLY CONTEXTUALISED KNOWLEDGE REPRESENTATION

Olimpo´s performance is centred about the combination of aspects derived from CBR and text information retrieval, in addition to an adequate organization of the knowledge related to the subject the system is focused on (in the present case, the UN Security Council´s Resolutions). The aforementioned knowledge organization is what enables the DCKR technology, which is a methodology that provides the possibility of comparing the contexts described in the documents and not only a comparison between words or attributes.

3.1 ANALYSIS OF THE RHETORIC STRUCTURE

The rhetoric structure of the system is comprised of indicative expressions used for comparison means and it was, first time ever, dynamically prepared. Up to then it was usual to choose a list of index pointers from a source external to the research group (for example, Court library indexes). Little work was done on the list of index pointers and its selection was based on its similarity with the context of the system under development. For the Olimpo system it was decided to build a particular and specific list, which should be aligned with the issues effectively treated by the Resolutions and, on the other hand, should be coherent with the documentation context of the managing entity of the database. In this view, in order to collect a list of expressions a detailed reading of the Resolutions was performed, searching onto UNO´s database on the Internet was done and debating with research groups was used. Those expressions were then tested and subject to

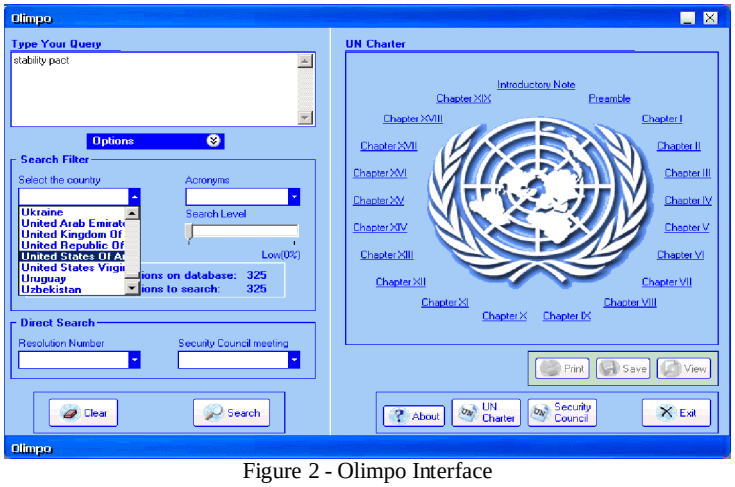


Figure 2 - Olimpo Interface

statistic analysis in order to evaluate their function as reference elements for the indexation and retrieval of documents. A set of expressions with high significance was selected, eliminating those ones with very high or very low frequency of occurrence because they were not very much helpful for establishing a context.

This process had a dynamic characteristic because it was done several times and expressions were included or excluded according to their statistic performance. The routine described shows how it worked to obtain a final list containing a set of expressions that could efficiently reflect the generic, rhetoric structure of the Resolutions, which gave the material form to the dynamically contextualised knowledge representation.

3.2 STRUCTURED CONTEXTUAL SEARCH – SCS

The searching process being described is said to be "contextual" and "structured" due to the following reasons:

a) For building the rhetoric structure of the system, it is taken into consideration the context of the stored documents;

b) This context is the basis for the input adjustment process, as well as for the comparison and selection of documents;

c) When writing the search text, the input is not limited to a set of words or attributes, but it can take the format of a long text, including the possibility of setting specific attributes, which work as filters and function as a preliminary selection of documents to be searched.

Information contained in the documents is represented in the form of a case, consisting of the original document and a set of eight indexes in the form of pairs of attribute-value: subject, date, number of the Resolution, meeting, country, acronyms, decisions, and indicative expressions. These indexes are part of the system interface (see Figure2).

In general, the system works in a way similar to other case-based systems [3] [5] (see Figure 3), where a manual entry passes through an adjustment and is then submitted to a comparison with the documents contained in the database, from which the most suitable ones are selected based on similarity calculations

After a refined modelling of the database, the Resolutions are stored by Olimpo system, according to their characteristics and central attributes (main topics, related subjects, countries involved); peripheral attributes (other related Resolutions, other UNO´s organisms referred to); and superficial attributes (dates, numbers and names). This kind of structure allows to give (variable) weights to attributes, enabling a more precise, contextualised search.

Furthermore, the control of depth of search enables a selection of documents according to a higher or lower occurrence of indicative expressions within the text of the Resolution, before starting to compare the documents. This process provides a more efficient way of reducing the search field; it is not a mere pre-selection of documents based on their superficial characteristics, but a preliminary comparison oriented by the context related to the search input.

After completing the process, the result is a list of indicative expressions referring to the Resolutions, producing an individual record of the occurrence of each one of the expressions within the text of each Resolution. These records allow the system to make the comparison and to apply the global similarity metrics.

In addition to the indicative expressions, the process of automatic extraction of attributes was prepared to detect and extract the subject, date, number of the Resolution, acronyms, country names, and parts of the text that contain the expressions with higher occurrence.

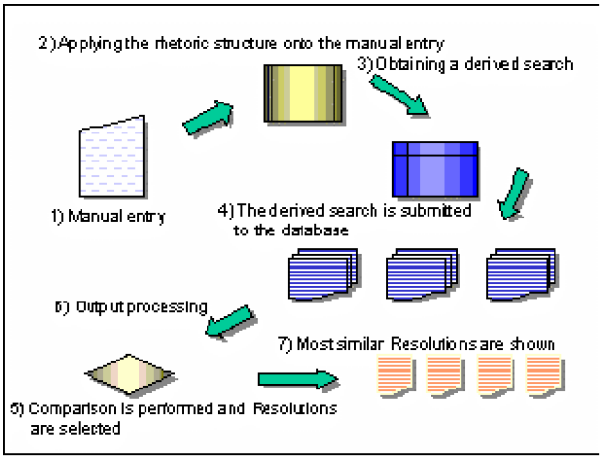


Figure 3 – Searching Process

The main features of the Olimpo system are the simultaneous use of textual information retrieving techniques based on CBR and the possibility of an extensive textual input. That makes the system to reach a differentiated performance in terms of information retrieval.

However, considering that the comparisons are based on a rhetoric structure previously prepared, the better working of the system is linked to a description of the search entry closer to that rhetoric structure. So, the system performance becomes gradually more consistent as the search entry language gets closer to the structure identified in the documents that generated the knowledge base of the system.

It has to be mentioned that all the Resolutions were monitored all the time with respect to the number of indicative expressions they presented during the structuring phase of the Resolutions knowledge base.

4 SIMILARITY METRICS

The similarity metrics was structured to consider the indicative expressions present in the case and in the search, after applying the rhetoric structure on the textual entry and producing the derived search. This derived search is actually the reference to work out the similarity metrics.

Taking as an example the case where a derived search with a total of 50 expressions is obtained after applying the rhetoric structure on a given search text: this set of expressions is compared to the records in the database and the similarity percentage is calculated based on the number of similar expressions found within each individual record. If 43 expressions are found, for instance, then the similarity will be 86%; it will be 72%, if 36 expressions are found, or 56% in the case of finding just 28 expressions, and so on.

This type of metrics is quite simple, one of the most simple that could be used in this situation, but it works in a quite stable way and can be improved in the future by incorporating new mechanisms like trigrams or applying internal weights to the most frequent words found in the text of the Resolution.

In fact, what is the strong feature of the system is not the similarity metrics, but the way how the indicative expressions are organized so that the metrics provides a better performance.

A clear example of this particularity of the system is an expression formed by two words like “United Nations”. A simple similarity based on counting individual words will show a 100% index when both words are found within the text, regardless their position, or 50% in the case just one of the words is found. However, if a differentiated indexation is used, by which “United Nations” (the two exact words appearing together) is one expression, “United” is another expression, and “Nations” is a third one (all with the same weight, for the time being), this configures a different situation. In this case, it is not enough finding the two words within the text, even when separated; they should appear together and having the exact meaning. Based on these criteria, the similarity index will be 33.33% when only one of the two words is found, or 66.66%, when both words are found in separate location, and it will reach 100% only when both words are present and appear together.

5 THE FUTURE OF OLIMPO SYSTEM

Internet was developed more than three decades ago, financed by the Department of the Defense of the U.S.A. Originally projected to connect the main systems of computers about a dozen of universities and organizations of research, the internet is currently accessible in hundreds of millions of computers in the world.

With the introduction of the World Wide Web - which allows personal computers to visualize documents based on multimedia about

almost all the subjects - the internet literally explodes for what it certainly seems to become, the main mechanism of communication of the world.

Most computer applications were executed in computers which couldn´t communicate between themselves. Currently, applications that are communicated with hundreds of millions of computers of the world can be written. The Internet establishes technology of computation and communications.

It turns our work easier. The information is accessible worldwide in an instantaneous and convenient form. It becomes possible that individuals and local small companies have a worldwide exposition. The people can find the better prices in products or services. Communities that have similar interests can stay in contact with each other. Researchers can be informed instantaneously about de last advances around the globe.

JAVA applications can be written in any computer platform, it means, any Java application needs a 32 bits version, such as Windows 95, Windows NT, other premium versions of Microsoft operational systems, MAC and UNIX, what results in a important economy of time and costs of development of systems for enterprises.

Java is a language completely object-oriented with hard support for proper techniques of engineering softwares. The programming object-oriented shapes objects of the real world with corresponding softwares. It takes advantage of the class relationships in which objects of a certain type - as a type of vehicles - they have the same characteristics. It also takes advantage of inheritance relationships where just-brought objects of a certain type inherit characteristics of existing ranks, but still keeping exclusive characteristics. An object of the convertible rank certainly has the characteristics of the automobile class, but the ceiling of a convertible opens and closes.

The object-oriented programming (OOP) supplies us a more natural and intuitive way to see the process of programming – as follows, shaping objects of the real world, its attributes and its behaviors. The OOP also shapes the communication between objects. In the same ways as people change messages between themselves, the objects also communicate by messages.

The OOP encapsulates data (attributes) and methods (behaviour) in packages called objects, data and methods of an object are intimately united. The objects have the property of hiding the information. This means that even though the objects can communicate with each other through friendly interfaces, normally it is not allowed for objects to know how the other objects are implemented - the implementation details are occulted inside of the objects themselves. Certainly it is possible to direct a car without knowing well the details of how the engine, the transmission and the system of exhaust pipe work internally.

For the use on the WEB, the interfaces of Olimpo system will be adapted in JAVA, because of the advantages shown above. The data base will be extended, so as to enclose all the Resolutions of the Security Council, making it possible for the users of the Web to carry through consultations to the complete base and thus to take off more advantage of the system. The site will be of free access to the public, or either, without no cadastre form, and moreover it will possess a system of automatic updating to facilitate maintenance of information.

The system will have basically two distinct modules: the first one will be of the consultation that, based in the methodology of Context Structured Search – CSS will allow a similar retrieval to the existing one in the current system, containing filters and fields to open-search concept-based, and the administration module, who will be responsible for the inclusion of new resolutions and maintenance of the database. The administration module is necessary, because Olimpo system has a base of knowledge especially developed to retrieve in an intelligent way the resolutions of Security Council. Knowledge base is a referential structure of representation in that domain specially studied, built so that the algorithms to retrieve information, can be semantic references in the search.

The knowledge base of Olimpo system will be opened after its implementation to insert news data. Those can be given through the insertion of new resolutions, as well as more words and references in the specialized dictionary, which is meant for the mining and retrieval of information contained in the system database.

The updating of the data will be made in a dynamic form, that is, all the process will be made with a reduced effort, being up to the administrator of the system the function of including of new data in the base and the managing of the dynamics of the working of the system, keeping this up to date and compatible with the new referring events the Security Council of the ONU.

As follows below, there is a sketch of how the structure of the net will be. (Figure 4):

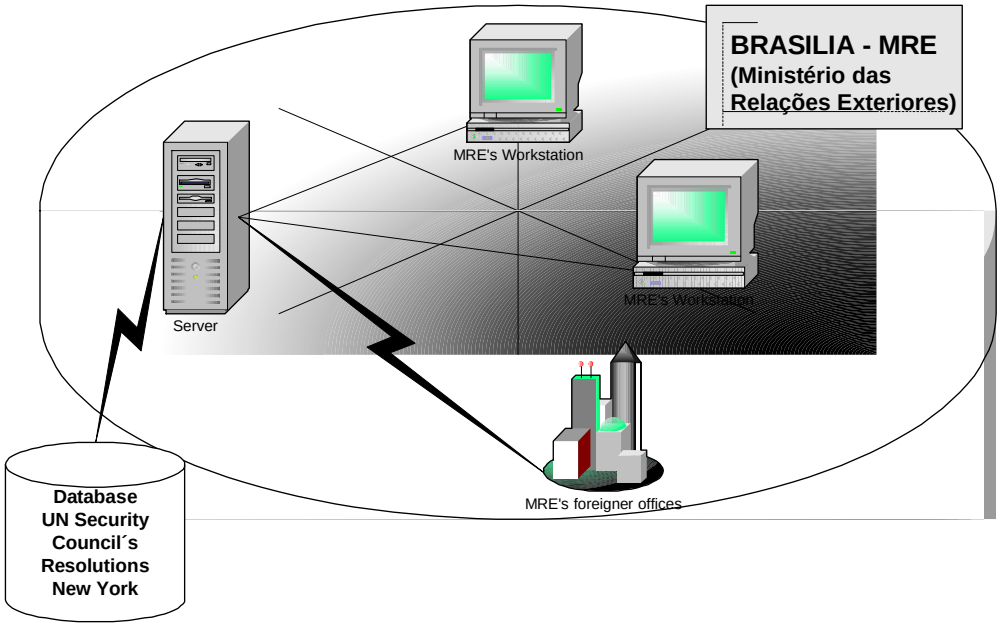


Figure 4

6 FORESEEN IMPACTS

Application of Olimpo system will in such a way bring a series of impacts at national and international level, influencing not only in the technological field, but also in relation to citizenship, the knowledge and the research.

See some relevant impacts bellow:

Scientific Impact

· Accomplishment of seminaries and workshops on the applicability of the Technology of the Legal Information for the Security Council of the ONU;

· Publications in newspapers of the Institutions involved;

· Stimulus to the perfectioning of the techniques applied;

· Academic works about this theme.

Technological Impact

· Consolidation of the technique "Dynamically Contextualised Knowledge Representation" (RCKR), which increases performance of systems structured in knowledge;

· Perfectioning of the techniques of use of textual data bases with RCKD, that propitiate the application of PCE, for construction of Olimpo system;

· Projection of Brazil as a technological reference.

Economic Impact

· Promotion of the companies of technology of the Information and Management of Knowledge;

· Dissemination of the applicability of the technologies used in OlimpoÒ by areas, that present the same difficulty in document research;

· Dissemination of the techniques of applied Management of the Knowledge to Olimpo.

Social Impact

· Facing the efficiency and the agility presented with the use of Olimpo in the processes of search of information, increasing the search of the users to the available services.

7. CONCLUSION

Olimpo system is a clear example of an innovative approach to the issue of information retrieval from complex text databases. Based on CSS technology (Context Structured Search, the system reaches a higher performance using DCKR technique for knowledge representation.

The innovation and new features implemented represent an upgrade of Olimpo system, improving its overall performance and usability.

Olimpo® System will be an important Brazilian contribution for the unanimity between nations, in view of the fact that it democratizes the access to knowledge of the select circle of the Security Council of the ONU, what makes it an important technological contribution for the World-wide Peace. Faster, necessary and perfect decisions will have, surely, greater and better international legitimation, favoring the creation of a fairer scene. As a consequence, it will be an instrument for the demonstration of the technological capacity withheld by Brazil, in the area of softwares for the management of knowledge.

REFERENCES

Amondt, A.; Plaza, E. “Case-Based Reasoning: Fundamental Issues, Methodological Variations, and System Approaches”. AI Communications 17(1), 1994.

Bench-Capon, T. J. M. Some observations on modelling case based reasoning with formal argument models. In: Proceedings of the Seventh International Conference on Artificial Intelligence and Law, p. 36-42, Oslo: Norway, June 14-18, 1999. 220 p.

Bruninghaus, Stefanie; ASHLEY, Kevin D. Toward adding knowledge to learning algorithms for indexing legal cases. In: Proceedings of the Seventh International Conference on Artificial Intelligence and Law, p. 9-17, Oslo: Norway, June 14-18, 1999. 220 p.

Bueno, Tânia Cristina D´Agostini. The use of juridical theory for retrieval from large juridical textual databases. Master Dissertation, PPGEP/UFSC. Florianópolis (Brazil): 1999. Original title: O uso da teoria jurídica para recuperação em amplas bases de textos jurídicos.

Bueno, Tania Cristina D'Agostini; Hoeschl, Hugo Cesar; Mattos, Eduardo da Silva; Barcia, Ricardo Miranda; Wangenheim, Christiane Gresse Von. JurisConsulto: Retrieval in Jurisprudencial Text Bases using Juridical Terminology. In: The Seventh International Conference on Artificial Intelligence And Law, 1999, Oslo. Proceedings of the Conference. New York: ACM, 1999. v.1. p.147-155.

Bueno, Tania Cristina D'Agostini; Hoeschl, Hugo Cesar; Mattos, Eduardo da Silva; Wangenheim, Christiane Gresse Von; Barcia, Ricardo Miranda. The use of juridical theory for retrieval from large juridical textual databases. In: Encontro Nacional de Inteligência Artificial, 1999, Rio de Janeiro. Anais do XIX Congresso Nacional da Sociedade Brasileira de Computação. Rio de Janeiro: Edições EntreLugar, 1999. v.4. p.107-120. Original title: Uso da teoria jurídica para recuperação em amplas bases de textos jurídicos.

Bueno, Tania Cristina D'Agostini; Hoeschl, Hugo Cesar; Mattos, Eduardo da Silva; Barcia, Ricardo Miranda; Bortolon, André; Wangenheim, Christiane Gresse Von. JurisConsulto. Florianópolis (Brazil): 1999. Software rights registered.

Hoeschl, Hugo Cesar. Olimpo System: Juridical Information Technology for UNO´s Security Council. Florianópolis (Brazil): UFSC, 2002. Doctorate Thesis. Original title: Sistema Olimpo: Tecnologia da Informação Jurídica para o Conselho de Segurança da ONU.

Hoeschl, Hugo Cesar; Barcia, Ricardo Miranda; Bueno, Tânia Cristina D´Agostini; Mattos, Eduardo da Silva; Bortolon, Andre; Donatti, Fabrício Tadeu. Olimpo System. Florianópolis (Brazil), 2000. Software rights registered.

Hoeschl, Hugo Cesar; BUENO, Tania Cristina D'agostini; MATTOS, Eduardo da Silva; BORTOLON, André; RIBEIRO, Marcelo Stopanowski; THEISS, Irineu; BARCIA, Ricardo Miranda. STRUCTURED CONTEXTUAL SEARCH FOR THE UN SECURITY COUNCIL. In: ICEIS - 5TH INTERNATIONAL CONFERENCE ON ENTERPRISE INFORMATION SYSTEMS, 2003, Angers. Proceedings of the fifth International Conference On Enterprise Information Systems. Setúbal: School of Technology of Setúbal, 2003. v. 2, p. 100-107. Referências adicionais: Classificação do evento: Internacional; França/Inglês; Meio de divulgação: Impresso; Homepage: http://www.iceis.org/papers.htm; ISSN/ISBN: 9729881618.

Hoeschl, Hugo Cesar; BUENO, Tania Cristina D'agostini; BORTOLON, André; RIBEIRO, Marcelo Stopanowski; MATTOS, Eduardo da Silva; THEISS, Irineu. Dynamically contextualized knowledge representation of the United Nations Security Council Resolutions. In: NINTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW, 2003, Edimburgo. ICAIL 2003 Proceedings. New York: ACM, 2003. v. 1, p. 95-96. Referências adicionais: Classificação do evento: Internacional; Escócia/Inglês; Meio de divulgação: Impresso; Homepage: http://www.cirfid.unibo.it/~agsw/icail03/; ISSN/ISBN: 1581137478.

Hoeschl, Hugo Cesar; Bueno, Tânia Cristina D´Agostini; Mattos, Eduardo da Silva; Bortolon, Andre; Barcia, Ricardo Miranda. Olimpo: Contextual Structured Search to improve the representation of UN Security Council Resolutions with information extraction methods. In: The 8th International Conference on Artificial Intelligence and Law, 2001, St.Louis, MO, USA. Proceedings of the Conference. New York: ACM, 2001. v.1. p. 271-218.

Steinfus, Ricardo. Handbook of International Organizations. Porto Alegre (Brazil), 1997, 352p. Original title: Manual de organizações internacionais

ANEXO II – ARTIGO PUBLICADO NO DEXA 2005

O artigo a seguir foi publicado no DEXA – International Conference on Database and Expert Systems Applications, no ano de 2005.

Knowledge Engineering Suite: a Tool to Create Ontologies for Automatic Knowledge Representation in Knowledge-based Systems

Tania Cristina D'Agostini Bueno[1], Hugo Cesar Hoeschl[2], Andre Bortolon[2], Eduardo Mattos[1], Cristina Souza Santos[1]

[1] WBSA Sistemas Inteligentes SA, Parque Tecnológico Alfa, Centro de Tecnologia IlhaSoft , SC 401 Km 1 - Módulo 10 - Térreo B - João Paulo - 88030-000 - Florianópolis, SC – Brasil
{tania,mattos,cristina}@wbsa.com.br
http://www.wbsa.com.br
[2] Instituto de Governo Eletrônico, Inteligência Jurídica e Sistemas – IJURIS, Rua Lauro Linhares, 728 – sala 105 – Trindade - 88036-0002 - Florianópolis – SC – Brasil
{hugo, andre}@ijuris.org
http://www.ijuris.org

Abstract. This paper is focused on the process of systematic knowledge acquisition to be used in knowledge-based systems. The result is a computational structure that can be used inside the organization (Intranet) as well as outside (Internet). This structure is the Knowledge Engineering Suite, an ontological engineering tool to support the construction of ontologies in a collaborative environment and is based on observations from the Semantic Web, UNL (Universal Networking Language) and WordNet. We use both a knowledge representation technique called DCKR to organize knowledge, and psychoanalytic studies, focused mainly on Lacan and his language theory to develop a methodology called Mind Engineering to improve the synchronicity between knowledge engineers and specialists on a particular domain.

1 Introduction

The importance of knowledge-based systems is that they provide some particular characteristics of human intelligence to the computer, such as the capacity to understanding natural language and simulate reasoning under uncertainty conditions. Definition of the relevant information to be inserted into a knowledge-based system is a major problem in the construction of such systems, mainly because the process is basically experimental and depends mostly on the ability of the knowledge engineer. In particular, there is a high difficulty related to the definition of the terminology used to nominate the concepts and the relations. [1] Despite the high number of methods to perform the knowledge acquisition process, there is no one that deals with the understanding and learning of people involved in the process, both specialists and knowledge engineers.

More recently, the notion of ontology has become popular in fields such as intelligent information integration, information retrieval on the Internet, and knowledge management. The reason is partly due to what they promise: a shared and common understanding of some domain that can be communicated through people and computers [2]. Cooperative work has been used by different development teams worldwide, with reference to WordNet, Semantic Web and UNL (Universal Networking Language) through the construction of ontologies using collaborative tools. The use of ontological engineering tools, or metatools, to support the Knowledge Engineering process enables the process of organizing a knowledge base established on the relationship between relevant expressions within a context. Ontologies, as a basis for automatic generation of knowledge acquisition tools, simplify the system specification phase by taking advantage of ontologies defined during the Knowledge Engineering process [3]. Nevertheless, experience shows that often the bottleneck of building sharable ontologies lies more in

the social process than in the technology itself [4]. Therefore, a methodology for the process of knowledge acquisition was developed, so that the specialist and the knowledge engineer can work in synchronicity, in cooperative networked organizations. We call this methodology Mind Engineering. This synchronization process begins with the understanding of human intelligence, its unconscious manifestations and its relationship with words, since, according to Lacan, every human investigation is linked irreversibly to the inner space created by language.

In the present development, a tool was created to support the Knowledge Engineering process by assisting developers in the design and implementation of ontologies on a specific domain.

In earlier works, we used a methodology called DCKR (Dynamically Contextualized Knowledge Representation) [5]. DCKR allows to build a knowledge base, improving the construction of the ontology of the domain and the automatic representation of cases in knowledge-based systems, either in the legal area [6] or any other knowledge management domain [7].

It follows a description of the methodology for knowledge synchronization. This methodology allowed an exceptional coherence among the semantic relations of what are called 'indicative expressions', mainly by the support of all this computational structure during the process. This allowed the knowledge engineer and the specialist to develop, more than the knowledge representation of the domain, abilities such as an inherent conscience, discipline, persistence, and empathy.

2 Knowledge Representation in Knowledge-based Systems

We use a special process to extract and represent knowledge in the process of developing knowledge-based systems. The main purpose is to allow an automatic process of text indexing, on the basis of a controlled vocabulary and a dictionary of normative terms, constructed persuasively through the relevance of pre-defined terms, called key-normative terms [8]. Given the need to turn the acquisition process faster, it was necessary to evolve the process using IR (Information Retrieval) techniques to associate the relevance of the terms with the frequency of the words added to the controlled vocabulary and the dictionary of normative terms; this approach resulted in a methodology of knowledge representation called DCKR - Dynamically Contextualized Knowledge Representation [9]. DCKR is a methodology of knowledge representation centered on a dynamic process of acquisition of knowledge from texts, defined through the elaboration of a controlled vocabulary and a dictionary of terms, associated to an analysis of frequency of the words and indicative expressions of the specific context (see figure 1).

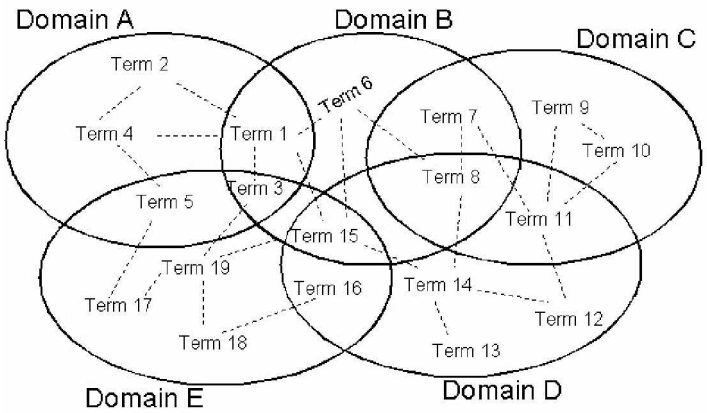


Fig. 1. The semantic relations of indicative expressions

2.1 UNL, Semantic Web and WordNet

In the process of knowledge acquisition for the preparation of a knowledge base of intelligent systems, methodologies that use web environments and cooperative development have to be used. Nowadays, there are three main solutions worldwide that use the Internet for the development of ontologies: UNL, Semantic Web and WordNet.

UNL (Universal Networking Language) [10] is a language for computers to share information through a network. It is meant for representing the natural language, so that computers can process the text and represent it in different languages.

WordNet [11] is a lexicon reference system inspired in psycholinguistic theories on the human lexical memory. The nouns, verbs, adjectives and adverbs of the English language are, organized in sets of synonyms, each one representing a lexical concept. Different semantic relations link the sets to each other.

The Semantic Web [12] is an extension of the current Web, in which the information has a very well defined meaning, allowing computers to process the information contained in web pages and to understand it, performing operations that facilitate the users' work.

The three initiatives are meant to facilitate the automatic processing of information contained in documents, allowing computers to perform more intelligent operations and to retrieve information in a more efficient way.

2.2 The Use of Ontologies in the System

The ontologies structure is the heart of a knowledge-based system that uses DCKR methodology. The reason for that is because all processing and storage of gathered information and knowledge base organization is done using this structure. It also plays an important role in the quality of the results presented to the user.

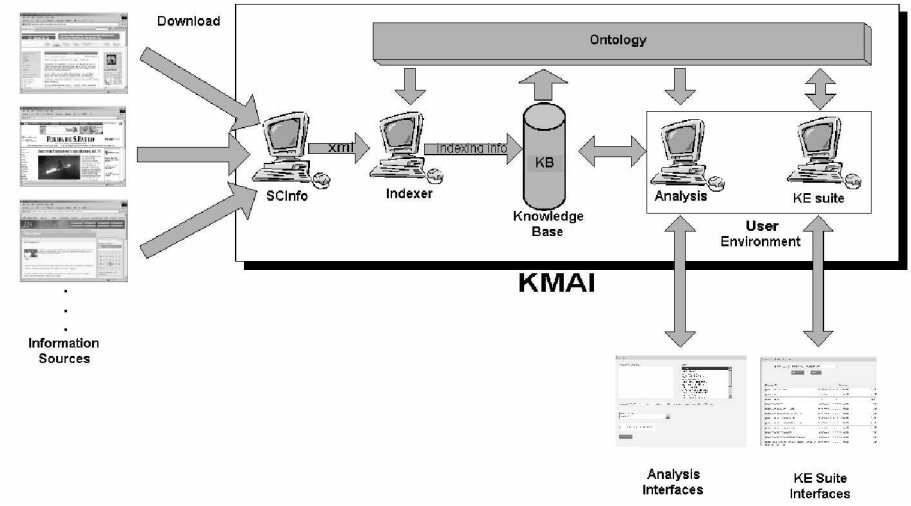


Fig. 2. Ontologies in the system

The participation of the ontology structure in the system occurs in three moments (see figure 2). At the first moment, the system extracts information from different previously selected sources. Each one of these documents is indexed based in the ontologies defined by the specialists and knowledge engineers during the knowledge engineering process. It means that the system will mark the documents with all indicative expressions found in the text, storing them in an organized way in the knowledge base. Thus, it is possible to make a pre-classification of the cases in the base according to what was defined in the knowledge organization promoted by the ontologies.

In a second moment, the ontologies are important in the analysis interface available to the user. The process begins at the moment in which the user types the input text for the search. At this point, the indicative expressions defined by the user that coincide with the ones presented in the ontology are identified. These expressions identified in the entry case determine the stream of relations that will be used by the system. It means that there is a dynamic relation between the way the user enters the indicative expression in the analysis interface and the way the relations in the Knowledge Engineering Suite are defined for this expression.

The first versions of the Knowledge Engineering Suite worked with key expressions, an approach that resulted in some rigidity in the ontology organization, for the weight of the information that was typed by the user in the search text was not considered. This rigidity is acceptable in cases in which the content of the documents stored in the system is standardized, with a small degree of variation. But in cases with broader domains and with different sources of information with no control over its contents, this approach was not efficient.

For this reason, it was decided to apply an approach that turns the use of ontology more dynamic in the analysis process. In this new approach, the importance of the indicative expressions to be considered is defined by the user. The system gives priority to the expressions and search for the corresponding

derivations for each case, according to the knowledge base. A priori, there is no hierarchy in the organization of the ontology in knowledge base. The weight of the relations will be based only in what is required by the search, where the context intended by the user is defined.

The third moment where the ontology takes part is in the Knowledge Engineering Suite, available in the system and integrated in its architecture. Through the Knowledge Engineering Suite the user is able to update the knowledge base with new expressions. At each new update in the ontology, the system re-indexes all the texts stored in the knowledge base, so the users may use this new ontology organization to search for documents previously indexed. It allows the verification of old documents that are related to a context that is important at the present moment. This way, it is possible to define a dateline about a subject, locating its start point.

2.3 The Knowledge Engineering Suite

The Knowledge Engineering Suite is an Ontological Engineering Tool for collaborative work on the Web, aiming to ease the sharing of knowledge between the Knowledge Engineering team and the specialist team. The Suite allows to build relationships between complex terms, considering its concept in the specific domain of application. These relationships are based on AI (Artificial Intelligence) techniques [13], theories of language, Semantic Web, WordNet, and UNL.

The creation of an infrastructure for the acquisition of knowledge for cooperative work on the Web is an efficient and effective tool of knowledge acquisition for intelligent systems. Many different techniques of Knowledge Acquisition exist; but Knowledge Engineering Suite (see figure 3) is integrated as part of DCKR methodology. Here, tools such as the Frequency Extractor, the Semantic Extractor and the Knowledge Engineering Suite have been associated with the methodology to help in the task of knowledge acquisition.

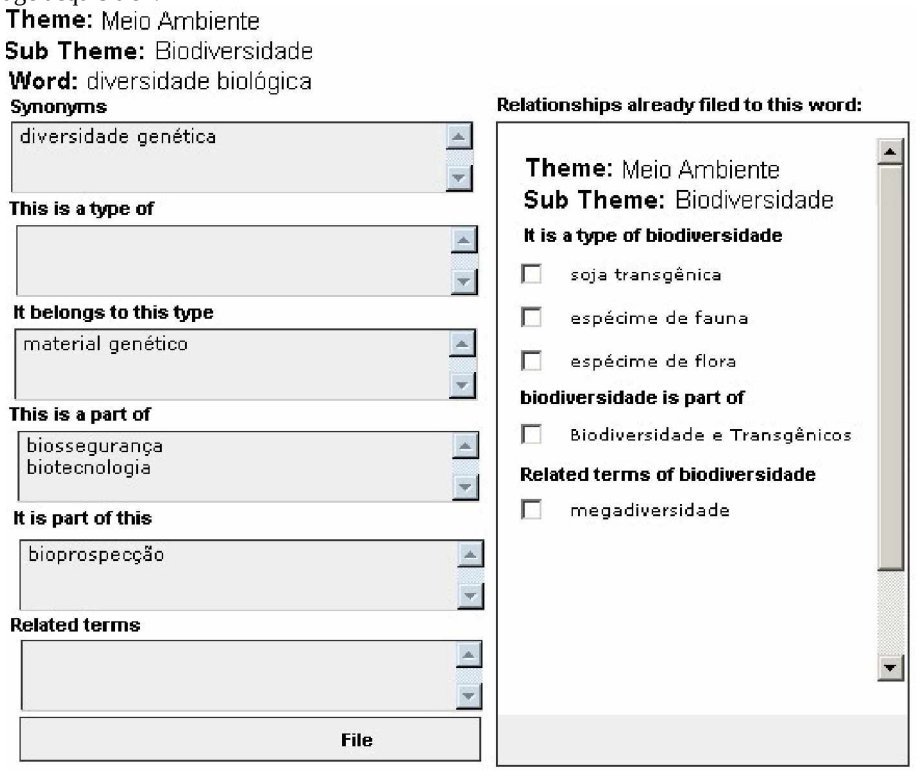


Fig. 3. Editing Module - Ontology construction (insertion and consistency checking)

This application works with extractors of automatic standards in conjunction with knowledge engineers and domain specialists as according to specifications found in the methodology DCKR, which consists of a dynamic process of analysis of the general context involving the theme to be focused on. The Suite is an editor of ontologies structured in a form to allow a cooperative work on the Web between the Knowledge Engineering team and the specialists team.

This computational environment of shared access has two main objectives: organization and representation of knowledge, and updating of the knowledge base. It is basically composed by four

modules, which are:
1. Filing. It allows to set up a contextualized dictionary, by selecting themes and sub-themes for the classification of indicative expressions. In this environment the user defines the theme and sub-theme under which new indicative expressions will be inserted. A domain can be categorized in various themes and sub-themes;
2. Search. It informs about other terms already filed on the base, which have some phonetic similarity with the term typed. This tool allows the verification of possible typing errors, besides preventing duplicated filing of the same term. It is a search system based on phonetic similarity. It supplies the user with a list of similar indicative expressions found in the knowledge base, in alphabetical order, when a query is typed by the user. The search module is used in the filing, edition and administration modules;
3. Relationship Editor. Allows the building of the relationship tree, always considering the similarity between all the terms filed and the ones already existing on the base. These relationships allow the system to expand the search context. The organization of the tree allows the dynamic definition of the weights of the indicative expressions according to the query of the user. The fields with all available relationships are presented. They are the following: -synonyms; -related terms; "this is a type of"; "it belongs to this type"; "this is a part of"; "it is part of this". The editor presents the existing relationships and allows to include them (see figure 3). Each relationship has a weight related to the defined indicative expression in the query by the user.
4. Administration Environment. The knowledge integration and the validation between words are made in accordance with the context of themes and sub-themes. The environment is organized in three levels: High Level - allows to insert themes and sub-themes, to validate exclusions, to include and to exclude users, to check productivity of each user and to check descriptions of the dictionaries, themes, sub-themes and indicative expressions; - Medium level- allows to check productivity and historical data; and, Low level- allows to check descriptions.

The definition of related concepts implies research work or help from a knowledge specialist on the matter. They are terms that can be considered as synonyms of themes and secondary themes, as well as close to the application context. An identifiable limit does not exist for the number of related concepts. Therefore it is important to observe the application of the terms in real cases. The specialists are helped in this task by a technological structure.

The module of related concepts is used by the domain specialists. They can work in their office, and then the contents are integrated into the knowledge base through the knowledge acquisition module (see figure 3). In order to enable the specialists work, a methodology based on the Theory of Juridical Argumentation [2] and Extensive Interpretation is used.

All the concepts, linked each other, generate a semantic-like network. This network improves the system capacity to recognize concepts, independently of finding it or not in the text. The network is organized into levels, indicating the "distance" between two concepts. These levels are used later on in the similarity measure.

However, all this structure and methodology was not enough to turn the cooperative work efficient and effective. A more holistic approach was necessary, which allows a greater coherence between the relations of the expressions, mainly in the definition of the related terms where the participation of the specialist is almost exclusive. It is important to highlight that this structure of contextualized ontologies allows automatic information indexing by the system and a knowledge acquisition that gives more qualitative answers in the retrieval process.

3 Enabling the Synchronicity in a Collaborative Networked Organization

The different unfolding of the human inventivity, although it is so diversified, has the same origin, the unconscious mind and the human perceptions. This is because distinct constructions eventually lead the mind to the same reference. Therefore we created a methodology that allows the immediate perception of the specialist to arise, without the pretension to reach all the knowledge, but with clear objectives, for example, to eliminate the common resistance of people to technological innovations, standing out the importance of management of human capital. [14].

During the development of tasks of Knowledge Engineering, it was observed that the efficiency of the acquisition process had a direct relation with good relationship between the knowledge engineer and the domain specialist, no matter what the quality or content of the interviews were, or the efficient application of the support tools. Thus, keeping this relationship in perfect synchrony is a key factor for the success of the system and a challenge for which the stages defined in the present work serve as a model of relative

success.

Common sense tells us that immediate perception (intuition) has greater effectiveness on the best solution for a problem than the application of rules of the propositional logic. However, the most accepted proposal is people trying to solve deductive problems applying rules such as those of the propositional logic. According to Lacan [15], if we consider that the unconscious is structured as a language, it is possible to reconstruct the unconscious associations between the words, thus disclosing a context.

There are elements, like the cognitive complexity and the capacity to learn, that supply the underlying individual traces on which the specialized knowledge and abilities are based, and similarly, sociability and confidence supply the anchors to develop and to keep a net of relationships. Thus, identifying that non-cognitive knowledge is also important knowledge of the institutions and, for this reason, they must be part of the capital of these organizations, it is necessary to look for a way to identify it and to represent it in the knowledge based systems. Therefore, this complex net of communications between the diverse areas of talent will provide the necessary flexibility, versatility and adaptability intelligences.

All the languages are structured as an articulating system. But their character and coherence is a unique articulated system. Thus the cognitive point of view concerning the symbolic acquisitions has as foundation the meanings generally supported by natural language or specialized languages such as the formal ones. To have these elementary meanings present in the work of a team requires synchronous thinking.

This synchronization process starts with the understanding of human intelligence, its unconscious manifestations and its relationship with words. Therefore, in accordance with Lacan [16], every human investigation is tied irreversibly in the interior of the space created by the language. But, for the success of this dynamics of cerebral exercise, it is essential the person to be in a positive attitude. The brain registers, learns and builds ramifications only when it is open to what is new.

3.1 Mind Engineering Methodology

There are many different techniques of Knowledge Acquisition. We created Mind Engineering (see figure 3) to help developing the following process (DCKR methodology): (1) Inventory of the entire domain (classification of all sources of digital information that will be in the system database); (2) Application of the word frequency extractor based on the database inventoried; (3) Comparison between extractor results with the specialist needs; (4) Construction of a representative vocabulary of the domain by the specialist and knowledge engineers; (5) Application of the semantic extractor on the database using the representative vocabulary (indicative expressions); (6) Definition of a list of words based on the evaluation of the results of the frequency of the indicative expressions found in the inventory (7) Construction of the ontologies in the Knowledge Engineering Suite based on this controlled vocabulary (8) Definition of synonyms, related terms, homonyms, hyponyms, hypernyms and meronyms.

The acquisition of knowledge carried out by the team of Knowledge Engineers had a bigger effectiveness in the area of its specialization [5] [6] than the acquisition performed by the same team in domains different from its specialization [7], where some obstacle of communication caused the need of a new acquisition process to be implemented.

Not having synchronization problems, the deep knowledge of the specialists on the AI technique applied in the system modeling (e.g., Case-Based Reasoning) allowed the transference of knowledge into the computational language in a very positive way for the final target of the system.

It was observing the elements presented in the two processes that we were able to systematize a series of questions, improving the speed and quality of knowledge represented in the system.

Additionally, uncommon procedures of knowledge acquisition were adopted, such as neurolinguistics and meditation techniques, to defragment the emotional memory of the specialist and to facilitate the learning process (see figure 4). This happened due to the following problems: (1) Resistance against the system; (2) Difficulty to reproduce the process of decision-making; (3) Low quality of the knowledge handled.

However, the focus object is not the area of application of the system (domain), but the work of the specialist and the knowledge engineer to define the target of the system and create the knowledge base of this system. To identify and to classify knowledge levels is essential, therefore both (specialists and engineers) have to be trained on the learning process; that requires them to overcome the comfort zone. Knowledge Engineering is mostly a process of knowledge exchange.

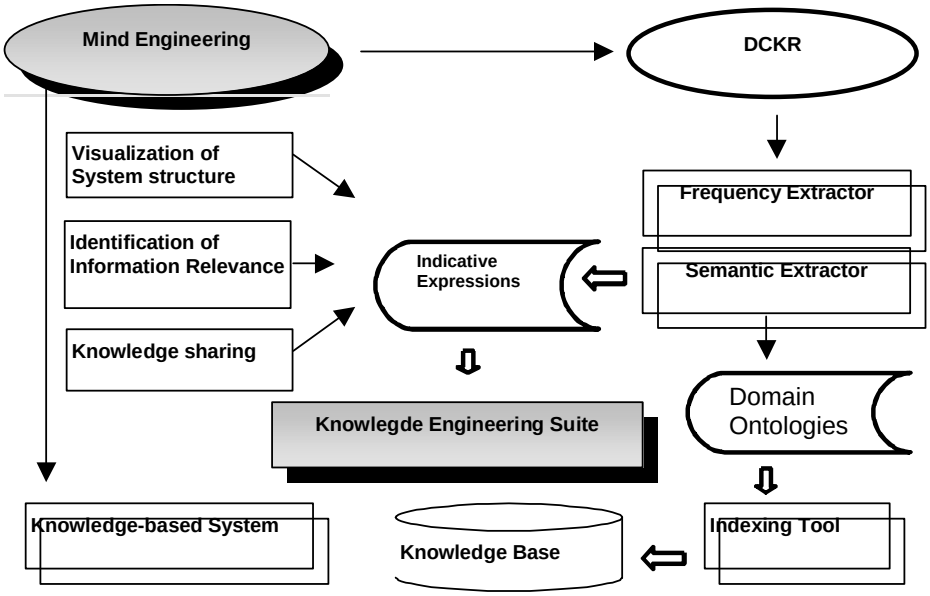


Fig. 4. Mind Engineering Methodology applied to the construction of ontologies in Knowledge-based Systems

The importance of existing knowledge for new acquisitions comes from the basic role they play inside the construction of the representation and from the idea given by that acquisition process to those representations. The importance of this phase is the exchange of knowledge; the specialist gets to know how his knowledge could be organized, that is, the basic concepts of the technique of Artificial Intelligence used in the representation of knowledge. Thus he will be able to contribute with more effectiveness and will have a greater interest in participating in the process. For the knowledge engineer, the exchange will lead to a more immediate perception of the target of the system and will increase the interest in going deeper in the study of the domain. Both will be prepared to deal with this overload and to obtain the ability necessary to plan or to choose a perspective that determines which elements of the situation must be treated as important elements and which can be ignored. By perceiving that the vast information or knowledge is reduced to a number of characteristics and relevant aspects, the decision making process becomes easier.

The continuous sharing of the established visions makes the specialists and engineers to work in better cooperation in the construction of the ontologies of the domain. This productive process is continuous and can lead to changes in the system implementation phase.

4 Conclusion

The systematization and organization of domain specialist teams together with the team of Knowledge Engineering became a big challenge in the development of knowledge management systems. The cooperative work between the teams does not only require the deep knowledge on the application domain, but also on the organization of its knowledge base. The creation of a computational environment on the web allowed a greater sharing of information and better results of the teams in the construction of knowledge-based systems.

The Knowledge Engineering Suite enables a cooperative work among people in different places, structuring a continuous knowledge base and easy visualization (knowledge tree) through relationship networks and supplies an exceptional coherence among the semantic relations of the indicative expressions, mainly by the support of all this computational structure during the process. This allowed the knowledge engineer and the specialist to develop much more than the knowledge of the domain, but abilities such as conscience itself, discipline, persistence, and empathy.

References

1. Resende, Solange Oliveira. Sistemas Inteligentes: fundamentos e aplicações. Barueri, SP: Manole,2003.
2. Duineveld, A. J. et al, 1999. WonderTools? A comparative study of ontological engineering tools. Twelfth Workshop on Knowledge Acquisition, Modeling and Management.Voyager Inn, Banff, Alberta, Canada.
3. Eriksson, H. et al, 1999. Automatic Generation of Ontology Editors. Twelfth Workshop on Knowledge Acquisition, Modeling and Management.Voyager Inn, Banff, Alberta, Canada.
4. Benjamins, V.R., 1998. The ontological engineering initiative (KA)2, Formal Ontology in Information systems. IOS Press, Amsterdam.
5. Hoeschl, Hugo. C. Bueno, Tania. C. D., Barcia, Ricardo. M., Bortolon, Andre., Mattos, Eduardo Da Silva. Olimpo: Contextual structured search you improve the representation council of UN security with information extraction methods In: å. Artificial International conference on inteligence and law, 2001, St. Louis. ICAIL 2001 Proceedings. New York: ACM SIGART, 2001, p.217 – 218.
6. Bueno, Tânia Cristina D'Agostini. O Uso da Teoria Jurídica para Recuperação em Amplas Bases de Textos Jurídicos. 1999. 94 f. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Catarina, Florianópolis, 1999.
7. Ribeiro, Marcelo Stopanovski. KMAI, da RC²D à PCE. Gestão do conhecimento com inteligência artificial, da representação do conhecimento contextualizado dinamicamente à pesquisa contextual estruturada. [2004]. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis, 2003.
8. Bueno, Tânia C. D. et al, 1999. JurisConsulto: Retrieval in Jurisprudencial Text Bases using Juridical Terminology. Proceedings of the Seventh International Conference On Artificial Intelligence And Law. ACM, New York.
9. Hoeschl, Hugo. C. et al, 2003. Structured Contextual Search For The Un Security Council. Proceedings of the fifth International Conference On Enterprise Information Systems. Anger, France, v.2. p.100 – 107.
10. UNL. Universal Networking Language. Available at: http://www.unl.ias.unu.edu/unlsys/index.html. Access on: 19 jan. 2004.
11. WORDNET. Available at: http://www.cogsci.princeton.edu/~wn/. Access on: 19 jan. 2004.
12. Semantic Web. Available at: http://www.w3.org/2001/sw/. Access on: 19 jan. 2004.
13. Kolodner, J. Case-Based Reasoning. Morgan Kaufmann, Los High, CA. 1993.
14. Gratton, Lynda, Ghoshal, Sumantra. Managing Personal Capital Human: new ethos will be the "Volunteer" Employee, The European Management Journal, vol 21, n° 1 pp1-10, February, 2003.
15. Lacan, Jacques. Os seminários de Lacan. Disponível em CD Rom, 2000.
16. Miller Jacques-Alain, 1988. Percurso de Lacan: uma introdução. Jorge Zahar Editor Ltda, 2a edição, Rio de Janeiro.

ANEXO III – ARTIGO PUBLICADO NO SIMPÓSIO DO WCC 2006

O artigo a seguir foi escrito baseado na tese apresentada e foi apresentado no Symposium on Professional Practice in AI 2006, ocorrido dentro do WCC (World Computer Congress) no ano de 2006.

A Model for Concepts Extraction and Context Identification in Knowledge Based Systems

Andre Bortolon, Hugo Cesar Hoeschl, Christianne C.S.R. Coelho, Tania Cristina D'Agostini Bueno
IJURIS – E-Gov, Juridical Intelligence and Systems Institute
Lauro Linhares, St. 728. 105. Trindade. 88036-002
Florianopolis, SC, Brazil
bortolon@ijuris.org, metajur@digesto.net, ccsrcoelho@aol.com, tania@ijuris.org
Home Page: http://www.ijuris.org

Abstract. Information Retrieval Systems normally deal with keyword-based technologies. Although those systems reach satisfactory results, they aren't able to answer more complex queries done by users, especially those directly in natural language. To do that, there are the Knowledge-Based Systems, which use ontologies to represent the knowledge embedded in texts. Currently, the construction of ontologies is based on the participation of three components: the knowledge engineer, the domain specialist, and the system analyst. This work demands time due to the various studies that should be made do determine which elements must participate of the knowledge base and how these elements are interrelated. In this way, using computational systems that, at least, accelerate this work is fundamental to create systems to the market. A model, that allows a computer directly represents the knowledge, just needing a minimal human intervention, or even no one, enlarges the range of domains a system can maintain, becoming it more efficient and user-friendly.

Keywords: Artificial Intelligence, Information Retrieval, Knowledge-Based Systems.

1 Introduction

The huge amount of documents on Internet has become a challenge to everyone that tries to find any information on any subject. [1] developed a method that allows us to estimate the size of Internet in 8.25 billions of pages on June 2003. The same work says the most known search engine, Google, has indexed only 37% of those pages. In 2005, [2] says Google has more than 8 billion pages indexed. Assuming that Google has maintained the rate of indexed documents on the size of Internet, it is not exaggerated to affirm that Internet has around 21.6 billion of documents. These documents deal with a wide variety of subjects, giving different, or even opposite, points of view on them.

In general, Information Retrieval (IR) systems work with indexes to represent the documents. These indexes can be built with or without a controlled vocabulary. Controlled vocabularies are lists of all important words in a specific domain. However, to build a controlled vocabulary is an expensive task, demanding much time and people. Besides, the absence of a domain specialist can produce low quality vocabularies, lowering the system's performance.

Ontologies are used to extend the controlled vocabularies, allowing knowledge engineers to relate terms among them. But, considering the Internet, it is almost impossible to build ontologies that represent all domains, besides all the time that is necessary to execute the representation.

The objective of this work is to develop a computer model to automatic extraction of terms from a random set of text documents, aiming at finding concepts and identifying the context that the document belongs. Or, at least, to provide information to a domain specialist and/or a knowledge engineer to build an ontology on one or more domains, simultaneously. In this case, the time spent in the construction of the ontology can be highly reduced.

This paper presents the preliminary results of the work. Its studies aren't completed yet, needing more detailed research to finish it.

Section 2 shows the technologies that base the model, section 3 shows the proposed model, and section 4 presents the conclusions obtained up to now and future works.

2 Involved Technologies

2.1 Information Retrieval

Information Retrieval (IR) [3, 4] is the traditionally applied technique to retrieve textual documents to a specific problem. However, unlike its name suggests, IR do not retrieve the information in the sense it delivers the facts that satisfy some necessary information. According to [3], “An information retrieval system does not inform (that is, change the knowledge of) the user on the subject of his inquiry. It merely informs on the existence (or non-existence) and whereabouts of documents relating to his request.”

To represent the documents, IR systems can work with either a logical or a complete representation. The former uses a keyword list, which is previously built by a human. The latter uses all words from the document. But, both have problems. The first one needs well-trained specialists to represent adequately the domain that the document belongs. The lack of keywords ends by degrading the performance, since the system cannot answer some queries, even they are related to the domain. The second representation maintains all the words, which can retrieve documents that are not related to the context. So, it is necessary to find more refined techniques to represent the documents aiming at contextualized searches.

2.2 Knowledge Engineering

According to [5], Knowledge Engineering (KE) is the process of construction of a knowledge base. In a simplified way, a knowledge base can be understood as a set of representations of facts on the world.

The construction of a knowledge base is done through a process called knowledge acquisition, where the knowledge engineers work together with domain specialists on the study and gathering of rules and concepts that are relevant to that domain. In the knowledge representation, a domain is some part of the world on which it is desired to express some knowledge. Normally, the knowledge engineer is not a domain specialist. He/she just needs one to support the study of knowledge acquisition.

There are many processes of knowledge engineering. Here, it is focused on the process used in Dynamically Contextualized Knowledge Representation (DCKR) methodology [6]. This process is well described in [7]. This work focuses on the problems that a knowledge engineering team has to create a knowledge base if both KE team and the domain specialists are not synchronized and shows a methodology to obtain this synchronization.

Briefly, the process can be described as a sequence of steps [7]. First one is to divide the domain in subdomains, as much as necessary. After that, knowledge engineers elaborate a conceptual map of the subdomain that will be worked on. Then, they identify the domain's usual vocabulary, visualize the results and identify the relevance of these results to the domain. The last step is to insert the terms and relations in the system. [7] says, “Although the Intelligent Systems has been demonstrated their importance and maturity, their use still has some challenges for their wide dissemination and implantation. The knowledge acquisition stage still is manual and subjective.”

One alternative to accelerate the time is to build an automatic system that can extract the terms of the documents and relate them. So, the KE team would only have to examine the results of the system and approve or decline them.

2.3 UNL

The Universal Networking Language (UNL) [8] can be defined as a digital metalanguage for the description, storage and dissemination of information independently of machine or natural language. It has been developed by United Nations University (UNU) since 90s and, later, distributed to various research centers around the world. UNL works in the same way as an interlingua, that is, all the natural languages can be converted to UNL and UNL can be converted in any natural language.

UNL works with the premise that the most important information in a sentence is the concept in it. This concept is represented through Universal Words (UW) and Relations. Both UWs and Relations are universal, that is, all languages can represent them. So, the sentence “the dog runs” is represented by the UNL sentence:

agt(run(icl>do).@entry, dog(icl>animal).@def),

where “run(icl>do)” and “dog(icl>animal)” are UWs that represent the concepts “run” and “dog”, respectively, and “agt” represents the relation between those UWs, indicating the second concept is the agent of the first one.

UNL's structure is based on three basic elements:

Universal Words: are the UNL vocabulary, representing a concept related to a word. Divided in two parts: headword, corresponding the concept; and, constraint list, representing the interpretation of the UW, that is, in which domain the UW is inserted. All the UWs are unified in a Knowledge Base. This Knowledge Base is hierarchical, allowing a better classification of the words.

Relations: relate two Universal Words through their syntactic behavior. They intent to describe the objectivity of the sentence. Examples: “agent”, “object”, “and”.

Attributes: on the contrary of relations, the Attributes describe the subjectivity of the sentence. They show what is been said from the speaker's point of view.

Beside these mechanisms, UNL has some tools that are responsible for the translation process. The most important are the Enconverter and the Deconverter. The former is responsible for translation the natural language to UNL. The latter does the inverse.

Due to its organization, UNL has a great potential to represent element that can be used in both Information Retrieval and Knowledge Based Systems.

3 A Model to Automatic Extraction of Terms and Contexts

The model described here intends to identify terms automatically and join these terms trying to determine contexts. To do this task, the model follows these steps:

1. Separation of text in sentences;
2. Separation of sentences in tokens;
3. Classification of tokens;
4. Mapping the tokens;
5. Building of the relationship tree;
6. Calculation of relationship's weights;
7. Calculation of relationship's proximity value.

3.1 Separation of text in sentences

In first step, all the sentences are separated to facilitate next steps. According to Brazilian grammar, described in [9], a sentence “is any linguistic enunciation that has complete sense. It can have one or more words, with or without a verb.” Examples: “Hi.” “Attention!” “The house is yellow.”

So, it is necessary to find all punctuation marks that indicate the end of a sentence: periods (.), exclamation points (!), and question marks (?). In some cases, colons (:) and semi-colons (;) can finish sentences, but they are not considered in this model. Finding all those punctuation marks, it is available a list of sentences.

For instance, considering the text: “The Palestinian president, Yasser Arafat, died in Paris. His body will first be carried to Cairo, Egypt, to the funeral reverences. The burial will be in Hamalah, Cisjordania.” It should be divided in three sentences: (1) The Palestinian …Paris; (2) His body … reverences; and (3) The burial … Cisjordania.

The next six steps are applied sentence by sentence, since the process is very similar to syntactic analysis.

3.2 Separation of sentences in tokens

Separation in tokens is a common process in IR indexing and Natural Language Processing systems. Here, it is used in same way.

3.3 Classification of tokens

Each token is classified in four categories:

Number: every sequence of number characters, with or without separators. Ex.: “87”, “1.439,26[2]”;

Dates: every sequence of number characters with date separators (both “-” and “/”). Ex.: “25/04/2006”;

Punctuation marks: all Brazilian Portuguese punctuation marks. Ex.: periods (.), commas (,), parentheses (());

Words: every sequence that cannot be classified in the previous ones. So, it is possible to classify all elements that appear in the text.

3.4 Mapping the tokens

Fourth step is to map each token to one of the ten morphological categories from Brazilian Portuguese grammar, such as: Nouns, Adjectives, Articles, Pronouns, Verbs, Numerals, Adverbs, Preposition, Conjunctions, and Interjections. This map is done comparing each token with a dictionary.

If a token has more than one entry in the dictionary, all the entries become candidates. For instance, the word “meio[3]” can belong to four different categories: Numeral, Noun, Adjective, and Adverb. To solve this problem, it is necessary to verify the words that are related to the ambiguous word in the sentence, establishing a context. This process is executed in the next step.

Other problem that appears in the mapping process is when the token doesn't have an entry in the dictionary. This situation occurs due to four main reasons. The first is the proper nouns. Normally, names of people, places, etc. don't appear in dictionaries. There are special dictionaries to deal with people names, places, and so on, but they cannot be complete, specially related to people, since parents are very creative people when naming their children. Considering an elegant and well-done texts, proper nouns start with a capital letter. So, this can be considered as a rule to identify them in the texts.

The second reason for tokens without entries is those that correspond to words with morphologic inflections, such as, plural, verbs (in Portuguese, all persons have a specific inflection), and gender (there are different suffixes to masculine and feminine). Two approaches can be used to solve the problem: insert all inflections of the word in the dictionary, indicating the correspondence with the main word; or, build a set of rules to replace suffixes and find the main form of the word.

Tokens that correspond to words incorrectly spelled or with typing errors are the third reason. The most common causes for typing errors are:

Missing character: a missing character in any part of the word. Ex.: “snd” instead of “send”;

Extra character: an extra character in any part of the word. Ex.: “sensd” instead of “send”;

Wrong character: a wrong character typed. Ex.: “semd” instead of “send”;

Changed character: two characters typed inversely. Ex.: “sned” instead of “send”;

Absence of accent marks: the word doesn't have the accent marks. Ex.: “coleção” is typed as “colecao”. This error is very common in search engine queries. Although, in regular documents, its occurrence is low.

The most common way to solve this problem is using an algorithm that generates all the words that should be the actual word, considering all the possible errors. But, this approach is questionable, since a

---

[2] Brazilian number format.

[3] In English, “half”.

simple five-letter word has 295 possible words (except absence of accent marks) as candidates. Other technique uses heuristics to find the most common errors (for instance, the absence of one “s” in words with “ss”). One other technique is to use probabilistic rules to determine the most similar entries to the given token.

The last reason is the tokens that aren't a word, so no entry can be related to them. For instance “abcde”. In these cases, the word is mapped to a noun.

The option of mapping all the unknown tokens as nouns comes from the fact that it is almost impossible to have a complete dictionary of proper nouns. So, if a token isn't an incorrect or inflected word, very probably it is a proper noun. The documents used as a test base confirm it, since they don't have words without any meaning in their body.

Therefore, the result of the third step is a list of tokens and their corresponded grammatical category. For instance, the sentence “Peter broke the window with a rock.” has the following list[4]:

Peter={(Peter; noun)}
broke={(brake; verb), (broke; adjective)}
the={(the; article), (the; adverb), (the; preposition)}
window={(window; noun)}
with={(with; preposition)}
a={(a; noun), (a; article), (a; preposition), (a; verb)}
rock={(rock; verb), (rock; noun)}

3.5 Building of the relationship tree

The relationship between words is a process that can be done through syntactic analysis. Among all the current available technologies, the chosen one is UNL, because it can perform both syntactic and semantic analysis. But, UNL rules for Portuguese aren't completely ready, yet. So, the solution was build a structure based in UNL, but with simpler representation and mechanism.

To build this structure, a study was performed to find which are the elements in language that can be put together to form terms. It was analyzed 1,042 terms of the KMAI System's ([11]) ontology. As expected, 100% of terms have a noun as part. These nouns were mainly related to other nouns and adjectives. More than 90% of terms with more than two words have a preposition relation the two other words. The conclusion was prepositions could be used as the element of relation between two words. The words that don't have a preposition between them are related with a underscore (_).

This simpler structure was improved using the UNL attributes that indicates number, time, concept (definite or indefinite nouns and negation), and a special one necessary to Portuguese, gender.

Therefore, the relation's structure is described as:

relation(word1.attributes, word2.attributes),

where *relation* is the preposition between the words or the character “_”, *word1* and *word2* are the words and *attributes* are the attributes of each word. For instance, the previous sentence “Peter broke the window with a rock.”, becomes[5]:

_(Peter, brake.@past)
_(brake.@past, window.@def)
with(window.@def, rock.@indef)

3.6 Calculation of relationship's weights

To determine which relationships correspond to terms, it is necessary to calculate some weights to the relations. In this model, it is used the same weights that are used in regular IR systems, such as, term frequency (tf) and inverse document frequency (idf). Since the weights are based in the relations, they become relation frequency (rf) and relation's inverse document frequency (ridf). They are calculated by the same formula:

---

[4] The English categories for the words come from [10]. The examples are illustrative. The model has just been tested in Brazilian Portuguese.

[5] Again, the example is illustrative. The model has just been tested in Brazilian Portuguese.

$$rf_i = \frac{n_i}{\sum_k n_k}$$

where n_i is the number of times the relation appears in the document and n_k is the number of relations of the document.

$$idf_i = \log_2\left(\frac{N}{n_i}\right)$$

where N is the number of documents and n_i is the number of documents with the relation.

3.7 Calculation of relationship's proximity value

The last step is to establish a proximity value between the relations, aiming at the creation of contexts. To do this, it is used a model based in the statistic co-occurrence of words, described in [4]. There, the similarity coefficients between two terms are based on coincidences in the term associations in the documents from the collection. The documents are represented by a matrix based in the vector-space model, where the rows are the documents' individual vectors and the columns identify the associations of terms and documents. In the model described in the paper, relations between two words are represented in the columns rather than terms. So, the matrix becomes as shown in Table 1.

Table 1. Matrix of association of terms

	R_1	R_2	…	R_k	...	R_m
D_1	rf_{11}	rf_{12}	…	rf_{1k}	…	rf_{1m}
…	…	…	…	…	…	…
D_n	rf_{n1}	rf_{n2}	…	rf_{nk}	…	rf_{nm}

The similarity between two relations can be calculated by the formula:

$$sim(REL_k, REL_l) = \frac{\sum_{i=1}^{n} rf_{ik} \cdot rf_{il}}{\sum_{i=1}^{n} rf_{ik}^2 + \sum_{i=1}^{n} rf_{il}^2 - \sum_{i=1}^{n} rf_{ik} \cdot rf_{il}}$$

where rf_{ik} indicates the frequency that relation i appears in document k and n is the number of documents in the base.

The similarity value indicates the probability that the two relations have to be related each other, since it indicates how many times one relation appeared and other also did. Doing the calculus for many relations can create cohesion between the relations, generating groups of relations that might be contexts.

Considering the matrix showed in Table 2:

Table 2. Example Matrix

	(1)	(2)	(3)	(5)	(6)	(10)
D_1	1	1	1	1	1	1
D_2	0	1	1	0	0	0
D_3	0	1	1	0	0	0
D_4	0	0	1	0	0	1
D_5	0	0	0	0	1	0

The similarities between the relations are disposed in Table 3.

Table 3. Similarities between relations

	(1)	(2)	(3)	(5)	(6)	(10)
(1)	X	0.33	0.25	1	0.5	0.5
(2)	0.33	X	0.75	0.33	0.25	0.25
(3)	0.25	0.75	X	0.25	0.2	0.5
(5)	1	0.33	0.25	X	0.5	0.5
(6)	0.5	0.25	0.2	0.5	X	0.33
(10)	0.5	0.25	0.5	0.5	0.33	X

To really find a context, it is necessary to put a minimum threshold to initiate the grouping. Tests have been done trying to determine this value, but no result was obtained yet. Basically, the test is to get one context and select a number of documents on it. So, extract the relations and calculate the similarity between that. Also, it is necessary to find other relations that should be considered in other contexts. After that, get other context related to the first and calculate the frequency of the relations from the first one appears and compare the values. Last, get a third context that does not have any relation with the first and do the same process. So, we can get some average from the three values and test in a generic set of documents.

4 Conclusions

Since the research is not finished yet, there are not so many results achieved up to now. But, the reducing of time to build a initial set of terms to analyze and build an ontology has already been evidenced. The main reason for it is that the simple structure of the model create documents that allow a fast recognizing of related words, creating a lot of candidate terms. Usage of weights also highlight the terms that happens with more frequency and which are normally related.

The model is also ready to be used in a Information Retrieval System, improving the representation and the results to the users.

Also, the model can be used in any IR system that uses UNL to structure the documents in the base, since the representation is strongly based in UNL.

References

[1] SOARES, António; BARROSO, João; BULAS-CRUZ, José. Estimativa da PIW através de Motores de Pesquisa de Grande Escala. In: Conferência IADIS Ibero-Americana WWW/Internet 2004. Madrid, 2004.

[2] ARNOLD, Stephen A. The Google Legacy. Chapter 3. Infonortics. 2005. Available at: http://www.infonortics.com/publications/google/technology.pdf.

[3] VAN RIJSBERGEN, C. J. Information Retrieval. Second Edition. Butterworths. London, 1979.

[4] SALTON, C.; MCGILL, M. Introduction to Modern Information Retrieval. McGraw-Hill, New York, 1983.

[5] RUSSELL, Stuart; NORVIG, Peter. Artificial Intelligence: A Modern Approach. Prentice-Hall. New Jersey, 1995.

[6] HOESCHL, H. C. Sistema Olimpo: tecnologia da informação jurídica para o Conselho de Segurança da ONU. Tese de Doutorado (Engenharia de Produção). Universidade Federal de Santa Catarina. Florianópolis, 2001.

[7] BUENO, Tânia Cristina D' Agostini; Engenharia da Mente: Uma Metodologia de Representação do Conhecimento para a Construção de Ontologias em Sistemas Baseados em Conhecimento. Tese de Doutorado (Engenharia de Produção). Universidade Federal de Santa Catarina. Florianópolis, 2005.

[8] UCHIDA, Hiroshi; ZHU, Meiying; DELLA SENTA, Tarcisio; The UNL, A Gift for a Millennium. UNU Institute of Advanced Studies. Tokyo, 1999.

[9] FARACO, Carlos Emílio; MOURA, Francisco Marto de; Língua e Literatura. 23ª Edição. Ática. São Paulo, 1995. v. 3.

[10] Merriam-Webster Online Dictionary. http://www.m-w.com. Accessed at: 26/04/2006.

[11] KMAI. Knowledge Management with Artificial Intelligence. Software. http://www.kmai.com.br.