PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

RECURSOS DO PROCESSAMENTO DA LÍNGUA NATURAL APLICADOS NA RECUPERAÇÃO SEMÂNTICA DE DOCUMENTOS DE CASO DE USO

CUSTÓDIO GASTÃO DA SILVA JÚNIOR

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Duncan Dubugras Alcoba Ruiz

Porto Alegre
2012

Dados Internacionais de Catalogação na Publicação (CIP)

S586r Silva Júnior, Custódio Gastão da
Recursos do processamento da língua natural aplicados na recuperação semântica de documentos de caso de uso / Custódio Gastão da Silva Júnior. – Porto Alegre, 2012.
77 f.

Diss. (Mestrado) – Fac. de Informática, PUCRS.

Orientador: Prof. Dr. Duncan Dubugras Alcoba Ruiz.

1. Informática. 2. Processamento da Linguagem Natural. 3. Sistemas de Recuperação da Informação. 4. Engenharia de Requisitos. I. Ruiz, Duncan Dubugras Alcoba. II. Título.

CDD 006.35

Ficha Catalográfica elaborada pelo

Setor de Tratamento da Informação da BC-PUCRS

Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada **"Recursos do Processamento da Língua Natural Aplicados na Recuperação Semântica de Documentos de Caso de Uso"**, apresentada por Custódio Gastão da Silva Junior, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Sistemas de Informação, aprovada em 15/12/09 pela Comissão Examinadora:

Prof. Dr. Duncan Dubugras Alcoba Ruiz– Orientador — PPGCC/PUCRS

Prof. Dr. Marcelo Blois Ribeiro - — PPGCC/PUCRS

Prof. Dr. José Valdeni de Lima – — UFRGS

Homologada em 11/09/12, conforme Ata No. 20 pela Comissão Coordenadora.

Prof. Dr. Fernando Gehm Moraes
Coordenador.

PUCRS
Campus Central
Av. Ipiranga, 6681 – P. 32 – sala 507 – CEP: 90619-900
Fone: (51) 3320-3611 – Fax (51) 3320-3621
E-mail: ppgcc@pucrs.br
www.pucrs.br/facin/pos

DEDICATÓRIA

À Deus que me manteve no prumo em todos os momentos de desarmonização.

À minha família, pelo seu amor e apoio incondicional.

AGRADECIMENTOS

Ao meu orientador Dr. Duncan Dubugras Alcoba Ruiz que com sua maneira peculiar de orientação, me conduziu até a finalização deste trabalho.

Aos meus colegas e amigos do MINTER que estiveram comigo nesta jornada um muito obrigado pelos momentos de estudo e descontração. Todos vocês tem parte neste trabalho.

Aos diretores das empresas Nexo Informática e NBS Informática, por flexibilizar o meu horário de trabalho para que eu pudesse viajar até a PUC/RS.

À minha esposa Giulia Schauffert e a minha filha Allana Anniele por ter me apoiado quando a saudade tirava a minha concentração e pelo incentivo nos momentos de dificuldade.

À Fundação de Amparo à Pesquisa do Estado de Mato Grosso (FAPEMAT) pelo financiamento parcial deste trabalho.

RECURSOS DO PROCESSAMENTO DA LÍNGUA NATURAL APLICADOS NA RECUPERAÇÃO SEMÂNTICA DE DOCUMENTOS DE CASO DE USO

RESUMO

A engenharia de requisitos trata fundamentalmente de como descobrir, analisar, documentar e verificar as funções e restrições que um software deve contemplar. Neste processo o projetista se concentra em entender as necessidades, metas e convicções dos interessados e em como transformá-las em artefatos de software. Isso é conhecido como ciclo de desenvolvimento e é realizado até que o software atenda todos os requisitos dos interessados. Este trabalho descreve como os recursos do processamento da língua natural foram utilizados na construção de uma solução para recuperação semântica de documentos de caso de uso e apresenta os resultados alcançados. Para a construção da solução, foi especificado um método que organiza os trabalhos de preparação e recuperação em duas fases. A primeira descreve a forma como o corpus deve ser preparado e como os termos utilizados na preparação podem ser utilizados na definição das palavras-chave do domínio. A segunda fase explica como a recuperação de documentos é realizada, e mostra como os relacionamentos descritos na ontologia são utilizados para melhorar os resultados da recuperação. Os resultados apresentados mostram que o método descrito neste trabalho é promissor, visto que ele apresentou cobertura de 100% em ambos os testes. Quanto a medida de precisão, que apresentou resultado inferior a 50%, o resultado foi compensado pelo algoritmo de *ranking* que ordenou os documentos de forma similar a classificação manual feita pelos usuários.

Palavras-chave: Recuperação semântica. Informações. Engenharia de requisitos. Conhecimento do domínio. Processamento da língua natural.

RESOURCES OF NATURAL LANGUAGE PROCESSING APPLIED ON SEMANTIC RETRIEVAL OF DOCUMENTS OF USE CASE

ABSTRACT

The Requirements Engineering basically deals with how to discover, analyze, register and verify the functions and restrictions that software must consider. In this process the designer not only concentrates in understanding the necessities, goals and certainties of the interested users but also in changing them into software devices. This process is known as development cycle and it is carried out until the software covers all the requirements of the involved users. This study describes how the resources of the natural language processing were used in the construction for a solution of semantics recovery of use case document and it also presents the reached findings. For the construction of the solution, it is specified a method that organizes the preparation and recovery works in two phases. The first describes the form how the corpus must be prepared and how the terms used in the preparation phase can be used in the definition of the keys concepts of the domain. The second phase explains how the document recovery is carried out and shows how the described relationships in the ontology are used to improve the results of the recovery. The presented findings reveal the described method in this study is efficient, since it presented a covering of 100% in both tests. Related of measure of precision, that presented an inferior result of 50%, it was compensated by the ranking algorithm that sorted the documents of similar form of the manual classification done by the users.

Keywords: Semantics recovery. Information. Requirements Engineering. Domain Knowledge. Natural language processing.

LISTA DE FIGURAS

LISTA DE EQUAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

API	Application Programming Interface
CASE	Computer-Aided Software Engineering
CMMI	Capability Maturity Model Integration
COM	Component Object Model
CRUD	Acrônimo da expressão Create, Retrieve, Update e Delete
EA	*Enterprise Architect*
IA	Inteligência Artificial
ODBC	Open Data Base Connectivity
PLN	Processamento da Língua Natural
PU	Processo Unificado
REQM	Requirements Management
RSLP	Removedor de Sufixos da Língua Portuguesa
TF-IDF	Term Frequency–Inverse Document Frequency
UML	Unified Modeling Language

SUMÁRIO

1. INTRODUÇÃO

Atualmente é vertiginoso o crescimento da quantidade de textos armazenados em formato digital. Isso tem incentivado o aumento de pesquisas que exploram estratégias para recuperação de informações relevantes aos usuários [Yat99]. Diferentes estratégias [Ahn+07, Bai07, Bas07, Wil+06, Chi05, Liu02, Yat99] foram pesquisadas e implementadas em mecanismos de recuperação no sentido de resolver o problema da recuperação de informações: **como disponibilizar e classificar os melhores resultados quando o usuário realizar uma busca?**

A motivação em responder a essa pergunta, provém da constatação de que o usuário ao formular a expressão de busca, o faz sugerindo ao sistema de recuperação o tipo de informação de seu interesse e espera como resposta, documentos que tratem deste e de outros assuntos relacionados à sua pesquisa. Notavelmente as consultas formuladas são ambíguas e, em alguns casos, trazem consigo o uso de jargões específicos a um determinado domínio [Bai07, Yat99].

O uso de jargões específicos e ambigüidade de termos são uma realidade nos documentos de requisitos que utilizam a língua natural como meio para a especificação de projetos de software [Som03]. Mas quando se trata de desenvolvimento de software, normalmente trabalha-se com escopo fechado e termos que tragam ambigüidades não são desejáveis [Som03]. Para resolver este problema, recomenda-se [Coc00] a criação de um dicionário onde todos os termos ambíguos ou jargões são colocados com seus respectivos significados.

Em estudo anterior, disponível em [Sil08], foram avaliadas as ferramentas *CASE IBM Requisite-Pro* e *Borland Caliber RM* e observou-se que, mesmo com a adoção do dicionário de termos, o módulo de busca dessas ferramentas *CASE* para o gerenciamento de requisitos de software não utiliza este dicionário para realizar as pesquisas no corpus. Pior que isso, o referido módulo desconsidera que os termos fornecidos para a consulta, indicam a necessidade de informação desejada, e não a informação em si. O resultado das consultas realizadas apresenta-se então como um subconjunto dos requisitos procurados. Para que o usuário tenha um resultado satisfatório, terá que fazer tantas consultas quantos termos relacionados à consulta original existirem. Mas como saber todos os termos presentes no domínio que se relacionam com a consulta realizada?

A resposta para essa pergunta não é trivial, pois ela trás consigo a complexidade inerente à engenharia de requisitos. Mas pode-se começar a levantar os termos e seus relacionamentos através do modelo de domínio e do dicionário de termos. Esses são artefatos normalmente presentes em uma metodologia de desenvolvimento [Lar07]. Nesta pesquisa nos concentraremos nos artefatos disponíveis no Processo Unificado.

1.1. Objetivos

O objetivo deste estudo é propor um método para recuperação semântica de documentos que se utiliza dos recursos do Processamento da Língua Natural (PLN) e de uma ontologia para representação do seu domínio.

1.2. Justificativa

Em um projeto de software típico são criados inúmeros documentos que representam os requisitos para o desenvolvimento do sistema. Quando se utiliza o Processo Unificado (PU) como metodologia de desenvolvimento, os documentos de requisitos são conhecidos como casos de uso. Após os casos de uso serem implementados, eles são armazenados em um repositório e ficam disponíveis para os envolvidos no projeto.

Em referido estudo anterior, verificou-se que as ferramentas para gestão de projetos de software dispõem de módulos para localização de informações em documentos de seus repositórios, e estes trabalham com pesquisa direta por palavras-chave. As buscas por palavras-chave apenas localizam e recuperam documentos que apresentam em seu conteúdo os termos especificados na consulta, muitas vezes deixando de fora documentos relevantes ou mesmo sobrecarregando o usuário com documentos não relevantes.

A busca por melhores resultados no processo de recuperação tende a agregar recursos semânticos para melhora do sistema de recuperação. Por isso a necessidade de se modelar um ambiente computacional que recupere informações tal qual um especialista humano o faria, ou seja, um ambiente que considere os termos utilizados na pesquisa e o contexto em que esses termos se inserem.

Pesquisas atuais [Li09, Bai07, Ahn+07, Bas07, Par03] mostram que o uso das relações entre os termos e o uso do contexto em que a consulta está sendo realizada é fator

determinante para satisfazer a necessidade de informação do usuário. Estas pesquisas têm concentrado seus esforços na aplicação de ontologias para aquisição e uso das relações entre os termos, e descoberta do contexto. Os resultados alcançados mostram-se promissores.

Este trabalho vai ao encontro dessas pesquisas ao fazer a recuperação semântica utilizando as informações presentes em uma base ontológica, no sentido de alavancar os resultados da recuperação de documentos em um ambiente de desenvolvimento de software. Oferece ainda um protótipo para preparação do corpus e outro para recuperação semântica, ambos integrados na ferramenta *CASE Enterprise Architect (*EA).

O estudo de caso onde são realizados os testes da proposta são documentos de caso de uso que tem como domínio o departamento de pós-graduação de uma universidade. A validação da proposta é realizada de forma empírica, por equipes formadas por analista de sistemas que desempenharão o papel de analistas desenvolvedores e de analistas de negócios deste estudo de caso.

1.3. Projeto de pesquisa

Face ao crescente interesse em recuperação da informação e às suas aplicações em diversas áreas, optou-se por realizar uma pesquisa exploratória. De acordo com Santos [San07], a pesquisa exploratória é utilizada quando se deseja verificar a real importância do problema a ser pesquisado e o estágio em que se encontram as informações disponíveis a respeito do assunto.

A Figura 1 apresenta o projeto desta pesquisa, destacando as fases e as etapas executadas em cada fase.

- Fase 1: nesta fase fez-se o levantamento bibliográfico sobre a área de recuperação de informação e dos recursos do processamento da língua natural disponíveis que nos ajudasse o obter bons resultados na implementação de um método de recuperação, foco desta pesquisa. Buscou-se complementar o estudo inicial referente a essa áreas do conhecimento. Após, verificou-se o funcionamento do módulo de recuperação de duas ferramentas *CASE* utilizadas para gerência de requisitos de software, o que permitiu identificar uma oportunidade de pesquisa.

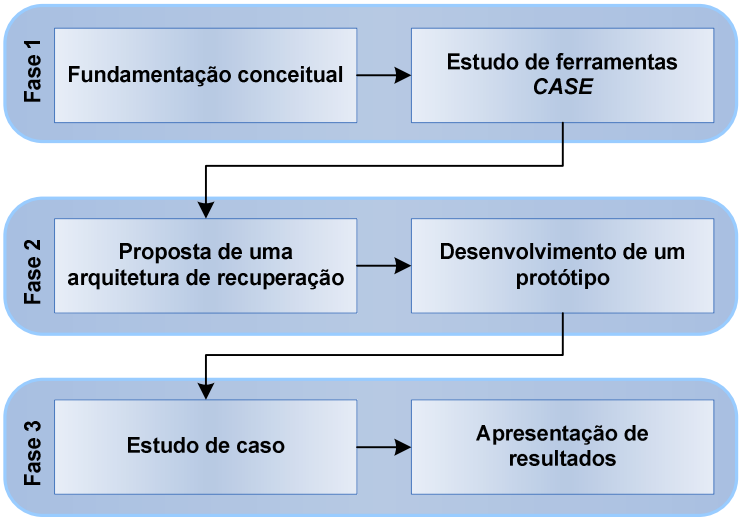


Figura 1 – Desenho da pesquisa

- Fase 2: nesta fase especificou-se uma arquitetura para recuperação de documentos de casos de uso. A arquitetura é composta de um método que define como o corpus é preparado, como a recuperação é realizada e por uma ferramenta, criada como um *plugin*, que executa o método proposto. Essa fase é detalhada na seção 5.
- Fase 3: nesta fase definiu-se o corpus para o estudo de caso onde os testes foram executados e o protocolo utilizado na validação da ferramenta. Os resultados são discutidos na seção 6.

1.4. Organização do trabalho

Esta dissertação está organizada em 6 capítulos. O capítulo 2 apresenta a fundamentação conceitual que apóia o desenvolvimento desta pesquisa. O capítulo 3 mostra os módulos existentes na solução, explicando a interação entre eles. O capítulo 4 descreve o método proposto neste trabalho, para recuperação semântica de documentos e o desenvolvimento de um protótipo que apóia a aplicação do método. O capítulo 5 apresenta os experimentos realizados e fornece uma análise dos resultados obtidos. O capítulo 6 faz as considerações finais discutindo a aplicabilidade do método e apresenta sugestões de trabalhos futuros. Por último, encontram-se as referências bibliográficas e anexos.

2. FUNDAMENTAÇÃO CONCEITUAL

2.1. Modelo vetorial de recuperação

O modelo vetorial de recuperação propõe um ambiente no qual é possível obter documentos que respondam parcialmente a uma expressão de busca. Isso é feito associando-se pesos aos termos de índice e aos termos da busca, que posteriormente são utilizados para calcular o grau de similaridade entre a expressão de busca e cada um dos documentos do corpus. Como resultado tem-se um conjunto de documentos ordenados pelo grau de similaridade em relação à expressão de busca.

No modelo vetorial, um documento é representado por um vetor de termos onde cada elemento representa a relevância do respectivo termo para o documento. Cada elemento do vetor é normalizado de forma a assumir valores entre zero e um, sendo os termos, cujo peso mais se aproximar de um, os de maior importância na descrição do documento. Analogamente, o corpus é descrito através de uma matriz onde cada linha representa um documento e cada coluna representa a presença de um determinado termo nos diversos documentos. Assim, um corpus contendo ***n*** documentos e ***m*** termos de índice pode ser representado conforme a Tabela 1.

Tabela 1 – Representação de um corpus com n documentos

	T_1	T_2	T_3	...	T_m
Doc_1	$w_{1,1}$	$w_{2,1}$	$w_{3,1}$	...	$w_{m,1}$
Doc_2	$w_{1,2}$	$w_{2,2}$	$w_{3,2}$	...	$w_{m,2}$
.	.	.	.	...	.
.	.	.	.	...	.
.	.	.	.	...	.
Doc_n	$w_{1,n}$	$w_{2,n}$	$w_{3,n}$	...	$w_{m,n}$

Onde:

$w_{m,n}$ representa a relevância do *m*-ésimo termo no *n*-ésimo documento e é dado pela Equação 1 (cosseno).

O modelo vetorial representa da mesma forma os documentos e as expressões de busca. Essa característica faz com que se possa calcular o percentual de similaridade entre a

expressão de busca e os documentos do corpus. A função para cálculo de similaridade é definida como:

Equação 1

$$similaridade(x,y) = \frac{\sum_{m=1}^{t}(w_{m,x} \star w_{m,y})}{\sqrt{\sum_{m=1}^{t}(w_{m,x})^2} \star \sqrt{\sum_{m=1}^{t}(w_{m,y})^2}}$$

onde:

- *x*: vetor que representa um documento do corpus
- *y*: vetor que representa a expressão de busca
- $w_{m,x}$: peso do *m*-ésimo elemento do vetor *x*
- $w_{m,y}$: peso do *m*-ésimo elemento do vetor *y*

Os valores de similaridade entre a expressão de busca e cada um dos documentos do corpus são utilizados no ordenamento dos documentos resultantes. Assim, no modelo vetorial o resultado de uma busca é um conjunto de documentos ordenados pelo percentual de similaridade entre cada documento e a expressão de busca. Esse ordenamento permite restringir o resultado a um número máximo de documentos desejados ou ainda definindo um limite mínimo para o valor da similaridade. Desta forma o usuário pode definir que a máquina de busca recupere somente os documentos com um valor mínimo de relevância em relação à expressão de consulta.

Baeza-Yates e Ribeiro-Neto [Yat99] definem como vantagens principais do modelo vetorial:

- possibilitar a independência entre os termos de indexação, o que faz com que a máquina de busca possa recuperar documentos que se “parecem” com alguns ou todos os termos da busca;
- impedir ao usuário formular expressões utilizando lógica booleana, o que pode restringir a flexibilidade das consultas, mas em contrapartida, faz com que usuários comuns possam ter resultados tão bons quantos usuários treinados em formular consultas booleanas; e
- ter bom desempenho de recuperação em corpora com diversidade de assuntos.

2.2. Modelo probabilístico de recuperação

O modelo probabilístico de recuperação propõe um *framework* em que os problemas de recuperação são tratados com a utilização de princípios probabilísticos. Desta forma, dada uma expressão de busca, o corpus é dividido em quatro subconjuntos distintos (Figura 2): o conjunto de documentos relevantes (*dRel*), o conjunto de documentos recuperados (*dRec*), o conjunto dos documentos relevantes que foram recuperados (*dRR*) e o conjunto dos documentos não relevantes e não recuperados. O conjunto de documentos relevantes e recuperados é o resultado da interseção dos conjuntos *dRel* e *dRec* [Dom01].

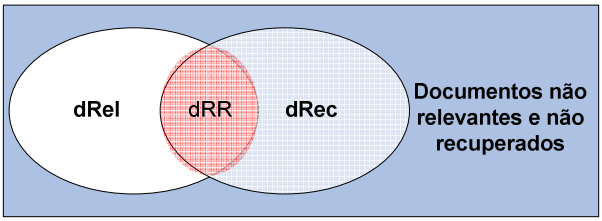


Figura 2 – Subdivisão do corpus após a execução de uma busca.

O resultado de uma busca ideal é o conjunto que contenha apenas os documentos relevantes para o usuário, o conjunto *dRR*. Entretanto os documentos que formam este conjunto não são previamente conhecidos. Assim, através da formulação de uma expressão de busca, tenta-se encontrar uma descrição probabilística inicial que descreva este conjunto. Os resultados obtidos após a execução da primeira buscam podem gradativamente ser melhorados [Yat99].

Seja *dRel* o conjunto de documentos relevantes e ¬*dRel* o conjunto dos documentos não relevantes. A probabilidade de um documento *d* ser relevante em relação a uma expressão de busca é dada por *p(dRel | d)* e a probabilidade de um documento ser considerado não relevante é representada por *p(¬dRel | d)*. Já a similaridade de um documento (*x*) em relação à expressão de busca (*y*) é definida como:

Equação 2

$$similaridade(x,y) = \frac{p(dRel|d)}{p(\neg dRel|d)}$$

Usando a função de Bayes obtém-se a seguinte expressão:

Equação 3

$$similaridade(x,y) = \frac{p(d|dRel) \star p(dRel)}{p(d|\neg dRel) \star p(\neg dRel)}$$

Onde:

- *p(d | dRel)* representa a probabilidade de se selecionar um documento *d* do conjunto de documentos relevantes *dRel*;
- *p(d | ¬dRel)* representa a probabilidade de ser selecionar um documento *d* do conjunto de documentos não relevantes;
- *p(dRel)* representa a probabilidade de um documento selecionado aleatoriamente ser relevante; e
- *p(¬dRel)* representa a probabilidade de um documento não ser relevante.

Baeza-Yates e Ribeiro-Neto [Yat99] observam que, caso se considere *p(dRel)* e *p(¬dRel)* iguais para todos os documentos do corpus, a fórmula de similaridade pode ser simplificada e escrita como:

Equação 4

$$similaridade(x,y) \approx \frac{p(d|dRel)}{p(d|\neg dRel)}$$

O modelo probabilístico representa os documentos do corpus por um vetor binário, onde os valores um e zero são utilizados para representar a presença ou ausência de um determinado termo de índice (Tabela 1), onde $w_{n,m}$ assumirá o valor zero para indicar a ausência de um termo, ou o valor um para indicar a presença do termo de indexação T_m no conjunto de termos do documento *Doc*.

A probabilidade de um termo T_m estar presente em um documento selecionado do conjunto dRel é representada por $p(T_m \mid dRel)$ e $p(\neg T_m \mid dRel)$ é a probabilidade do termo T_m não estar presente em um documento selecionado de *dRel*. Desta forma, temos a função de similaridade fundamental para ordenar os resultados do modelo probabilístico:

Equação 5

$$similaridade(x,y) \approx \sum_{m=1}^{T} \left(log \frac{p(T_m|dRel) \star p(\neg T_m|\neg dRel)}{p(T_m|\neg dRel) \star p(\neg T_m|dRel)} \right)$$

Uma vez definida a função de similaridade, é necessário que o modelo seja treinado de forma a reconhecer, baseado em uma pesquisa do usuário, os documentos que são relevantes e os que não são relevantes. Como no início do processo de busca não se sabe qual o conjunto de documentos relevantes, Baeza-Yates e Ribeiro-Neto [Yat99] sugerem que alguns valores de referência sejam assumidos: a) $p(T_m \mid dRel)$ igual a 0.5, onde este valor é constante para todos os termos T_m; b) assumir que a distribuição dos termos de indexação dos documentos é uniforme. Assim, o usuário realiza a busca e todos os documentos com *similaridade* ≥ *LimiteCorte* são apresentados em ordem descendente. Tendo esse conjunto de documentos, o usuário pode selecionar alguns documentos que considera relevantes para a sua necessidade. O sistema então pode utilizar esta informação para tentar melhorar os resultados subseqüentes. O usuário poderá repetir este processo de seleção de documentos relevantes até que o conjunto de documentos recuperados satisfaça a sua necessidade de informação, a esse processo denomina-se genericamente *feedback* de relevância.

As principais virtudes do modelo probabilístico estão em reconhecer que a atribuição de relevância é uma tarefa do usuário e o de apresentar os documentos em uma ordem que é definida de forma probabilística, onde são apresentados os de maior relevância primeiro. As suas principais desvantagens incluem:

- a necessidade de treinar o sistema, classificando os documentos como relevantes e não relevantes; e
- não considerar a freqüência com que um determinado termo ocorre dentro do documento, somente reconhecendo se ele existe ou não;

2.3. Avaliação dos sistemas de recuperação

Os sistemas de recuperação de informação realizam o processo de recuperação baseando-se em expressões de consulta que traduzem as necessidades de informações dos usuários do sistema. Na literatura [Yat99] encontra-se vários modelos de recuperação que foram desenvolvidos com o objetivo de criar um ambiente que maximize a recuperação de informações relevantes, sem com isso aumentar a complexidade na formalização da expressão de consulta. Com desenvolvimento de novos sistemas de recuperação, por vezes, na programação destes sistemas surge a dúvida: Qual modelo de recuperação deve ser adotado?

Neste contexto, aplicam-se metodologias de avaliação, que visam fornecer subsídios aos desenvolvedores sobre o quão bons são os resultados apresentados por cada modelo de recuperação. Na subseção 2.3.1 discutiremos brevemente o problema da relevância e nas subseções seguintes apresentaremos o método de julgamento de documentos recuperados comuns, técnica também conhecida como *pooling,* assim como outras variáveis utilizadas no processo de avaliação dos métodos de recuperação.

2.3.1. A noção de relevância

Gonzalez e seus co-autores [Gon07] afirmam que os problemas de um sistema de recuperação de informação não são menos complexos que aqueles inerentes à interpretação de significado. Este fato permite antever as dificuldades de se avaliar os resultados da recuperação, que ainda podem se agravar devido à noção de relevância. Assim, para ser possível avaliar se um sistema teve sucesso em recuperar um determinado documento em resposta a uma expressão de consulta, é necessário determinar o quão relevante é esse documento em relação à expressão de consulta.

A questão fundamental é que a relevância precede os sistemas de recuperação de informação, tendo como objeto principal o usuário. O seu julgamento é que define o quão relevante é o resultado de um sistema de recuperação em relação às suas necessidades de informação. Desta forma, diversos documentos podem ser relevantes. O desafio do sistema de recuperação é encontrar quais documentos são os mais relevantes, considerando a necessidade de informação do usuário e o contexto em que ele está inserido. Já o desafio de se avaliar estes sistemas é o de fazê-lo sem considerar o contexto em que o usuário está inserido ao formular uma expressão de busca – já que cada usuário tem contextos e necessidades de informações diferentes – mas, ainda assim, os sistemas melhor avaliados se comportam como se considerassem esse contexto.

Estratégias para minimizar os problemas da determinação de relevância na avaliação de sistemas de recuperação de informações incluem a construção de coleções de referência e o uso do método de julgamento de documentos recuperados comuns, que são tratados nas próximas subseções.

2.3.2. Metodologias de avaliação para sistemas de recuperação

A avaliação consiste em submeter os sistemas de recuperação de informações a um conjunto de testes, onde os procedimentos de indexação, recuperação e classificação de relevância são executados e os resultados apresentados pelo sistema são comparados, levando em conta uma coleção de referência. Uma coleção de referência é, normalmente, constituída por um conjunto de documentos, um conjunto de consultas já formuladas e a indicação dos documentos relevantes.

Para avaliar a qualidade da relevância gerada pelos sistemas de recuperação, uma possibilidade está em adicionar a uma lista, em ordem de relevância, os documentos que foram recuperados, classificando-os como relevantes ou não relevantes. Método conhecido como julgamento de documentos recuperados comuns. Os sistemas de recuperação são julgados, manualmente, por uma equipe de avaliadores cuja área de conhecimento corresponde aos temas das consultas realizadas. Após este julgamento, são utilizadas algumas medidas de avaliação, dentre as quais, se destacam as medidas de precisão e cobertura, necessárias para definir o quão bom é o sistema de recuperação de informações [Gon07]. Estas medidas são utilizadas para avaliar o desempenho dos sistemas de recuperação em relação à relevância dos documentos recuperados no processo de consulta e são discutidas com mais detalhes na próxima seção.

2.3.3. Precisão, cobertura e média harmônica

Na tarefa de recuperação de informação, a cobertura consiste na relação entre o total de documentos corretamente recuperados pelo sistema e o total de documentos corretos presentes na coleção. Precisão consiste na relação entre a quantidade de documentos corretamente recuperados pelo sistema e o número total de documentos recuperados (relevantes + não relevantes) [Gon07]. Portanto, cobertura refere-se a quantidade de informações relevantes que foram corretamente recuperadas, enquanto precisão refere-se à confiança da informação recuperada. Essas medidas, contudo, levam a objetivos conflitantes. Quando se tenta aumentar a cobertura, a precisão tende a piorar e vice-versa [Gon07]. Por isso, muitas vezes adota-se uma média harmônica [Gon07] que avalia o desempenho geral de um sistema. As medidas de precisão e cobertura são definidas, respectivamente, na Equação 6 e Equação 7:

Equação 6

$$P = \frac{n_r}{n}$$

Equação 7

$$C = \frac{n_r}{d_r}$$

Onde:

- *n*: quantidade de documentos recuperados;
- n_r: quantidade de documentos relevantes em *n*;
- d_r: quantidade de documentos relevantes a uma determinada consulta;

Combinando-se as duas medidas anteriores, encontramos a medida-F, média harmônica das anteriores, que é definida na Equação 8:

Equação 8

$$F - measure = \frac{(\beta^2 + 1) * C * P}{\beta^2 * (C + P)}$$

Onde, o parâmetro *β* é um fator que modifica a preferência da cobertura sobre a precisão. O mais freqüente é usar *β=1*, com o objetivo de se avaliar os sistemas de recuperação balanceando-se as duas medidas. Assim, temos a Equação 9 como uma simplificação da Equação 8.

Equação 9

$$F_1 = \frac{2*C+P}{C+P}$$

2.4. Processamento da língua natural na recuperação de informações

O PLN envolve um conjunto de técnicas computacionais para a análise de textos em um ou mais níveis lingüísticos, com o propósito de simular o processamento humano da língua [Jur00]. Esta abordagem surge como uma possível solução a alguns problemas relacionados à recuperação de informações, visto que tanto o corpus como as expressões de consulta formuladas pelos usuários se apresentam em língua natural.

O desenvolvimento de sistemas de recuperação de informação que possam “interpretar” os documentos exige um alto custo computacional. Por esta razão, as técnicas de PLN são utilizadas para melhorar o desempenho geral do sistema, atacando problemas

lingüísticos que possam interferir nos resultados da recuperação, como por exemplo: solucionando ambigüidades ou mesmo buscando contextualizar a expressão de busca do usuário.

Jurafsky e Martin [Jur00] propõem a abordagem do PLN em seis níveis, a saber: fonológico; morfológico; sintático; semântico; pragmático e discurso.

a) Fonológico: é o de interpretação dos sons e fala; o de maior interesse na implementação dos sistemas de reconhecimento da fala, onde o usuário pode expressar verbalmente um comando, ou receber resposta de forma audível.

b) Morfológico: neste são analisadas as variações que podem ocorrer em uma palavra. As variações são detectadas observando-se os prefixos, sufixos e radicais que compõem a palavra analisada. Um exemplo de processamento morfológico na recuperação de informação são as técnicas de extração de radicais (*stemming*) que visam substituir as variantes de uma palavra por uma forma normalizada.

c) Sintático: no qual é determinada a estrutura sintática das frases de um texto. Por causa da enorme quantidade de estruturas frasais presentes em um texto, determinar precisamente a estrutura de uma frase requer um alto custo computacional, degradando a performance do sistema. Por este motivo o processamento sintático é evitado nos modelos tradicionais de recuperação de informação.

d) Semântico: busca interpretar o significado de palavras individuais e também o significado de expressões ou frases. Um exemplo do processamento neste nível é a resolução de ambigüidades, visto que muitas vezes as ambigüidades só podem ser solucionadas quando analisadas dentro de um frase ou parágrafo.

e) Pragmático: neste, o sistema de recuperação utiliza ontologias, dicionários ou quaisquer outros conhecimentos externos aos documentos e expressões de busca executadas anteriormente. Este conhecimento pode ser específico a um determinado domínio ou pode versar sobre as necessidades dos usuários, como preferências e objetivos na formulação das expressões de busca.

f) Discurso: aqui são analisados as estruturas e os princípios organizacionais de um documento.

Entre os níveis (a) e (b), se insere o nível lexical onde é tratada individualmente a palavra. O exemplo mais comum de processamento neste nível é a construção de lista de palavras – *stopwords* – de pouco valor semântico, como artigos e preposições. Este nível está relacionado, por exemplo, com a geração e uso de vocabulários controlados – tesauro ou ontologias – na indexação de documentos e na formulação e expansão de expressões de busca.

2.4.1. Variações lingüísticas

A importância de reconhecer as variações lingüísticas dentro de um texto se dá, principalmente, pela possibilidade de controle de vocabulário, o que permite melhorar o desempenho geral do sistema, visto que a quantidade de palavras que são processadas diminuirá. A normalização lingüística pode ser tratada em três casos distintos: morfológica; sintática e léxico-semântica [Jur00].

A normalização morfológica produz a redução dos itens lexicais de forma que dois ou mais termos são representados através de uma única forma. Assim, todas as variantes de uma palavra são percebidas da mesma forma pelo sistema de recuperação. Para realizar esta normalização, as técnicas mais conhecidas são o ***stemming*** e a **lematização**, processo que reduz uma palavra à sua forma canônica. Esta pesquisa utiliza o algoritmo RSLP para fazer *stemming*. Em [Ore06] está disponível os experimentos e os resultados alcançados por este algoritmo.

A normalização sintática ocorre quando há a normalização de frases semanticamente equivalentes em uma forma única e representativa das mesmas, como “a casa foi pintada de azul e amarelo” e “a casa foi pintada de amarelo e azul”.

A normalização léxico-semântica ocorre quando são utilizados relacionamentos semânticos entre os itens lexicais de forma a criar um agrupamento de similaridades semânticas, que são identificadas por um item lexical que representa um conceito único. Esta é a forma utilizada quando o sistema emprega um tesauro para melhorar os resultados de busca de expressões formuladas pelos usuários.

2.4.2. Resolução de ambigüidade

A ambigüidade é a propriedade que faz com o que um termo, uma palavra ou todo um texto, possa ser interpretado de modos diferentes. A ambigüidade pode ser do tipo sintático ou semântico [Jur00]. A ambigüidade sintática ocorre quando um termo pertence a mais de uma classe gramatical, como “forte”, que pode ser um substantivo “o forte no alto do morro” ou um adjetivo “o café é forte”. Já a ambigüidade semântica ocorre quando um termo apresenta mais de um significado, por exemplo, o verbo passar, que pode significar “passar a ferro”, “passar no vestibular” e “passar no trabalho”.

As ambigüidades podem ser classificadas como lexicais, quando é possível a um termo assumir múltiplos significados; e estruturais, quando é possível mais de uma estrutura sintática para a sentença. Jurafsky [Jur00] aponta que a ambigüidade lexical pode ser resolvida com abordagens cognitivas ou lingüísticas. A primeira procura investigar como fatores semânticos, sintáticos e neuropsicológicos podem contribuir na resolução desse tipo de ambigüidade. A abordagem lingüística considera estratégias em nível sintático e semântico. Em nível sintático, são levadas em consideração as palavras vizinhas da palavra ambígua. Já a abordagem semântica considera metodologias para representação do conhecimento sobre os termos, sendo necessário especificar contextos ou domínios restritos. Nota-se que, em determinados casos, a ambigüidade sintática somente pode ser resolvida com a utilização da abordagem semântica.

Abordagens atuais procuram resolver a ambigüidade de forma semântica. Desta forma, os termos relacionados encontrados na base ontológica são utilizados como fatores contextuais ao termo ambíguo.

2.5. Abordagens semânticas na recuperação de informações

As abordagens semânticas têm como principal característica o enriquecimento da expressão de consulta com informações contextuais de interesse do usuário. O enriquecimento normalmente é realizado adicionando-se ao conjunto de termos da pesquisa, outros termos relacionados ao domínio em questão. Normalmente o enriquecimento é utilizado com o objetivo de aumentar a medida de cobertura, mantendo a medida de precisão em padrões aceitáveis pelo usuário. Esta seção apresenta as abordagens utilizadas nesta pesquisa.

2.5.1. Ontologias

Ontologia é o ramo da filosofia que tem por objeto o estudo das propriedades mais gerais do ser. Este termo foi adotado pela comunidade de Inteligência Artificial (IA) para se referir aos conceitos e termos que podem ser usados para descrever alguma área do conhecimento ou construir uma representação desse conhecimento [Bre05].

Ao longo do tempo, diversas áreas do conhecimento têm emprestado da filosofia este termo quando se deseja uma estrutura que descreva alguma coisa. Por esse motivo, é comum encontrar diversas definições para ontologia. Breitman [Bre05] faz um apanhado geral das definições nas mais diversas áreas do conhecimento e faz uma discussão sobre o tema. Conclui dizendo que "*independente da definição escolhida, é necessário entender que ontologias têm sido utilizadas para descrever artefatos com variados graus de estruturação e diferentes propósitos. A variação vai de simples taxonomias (...) até representações para metadados (...)*".

Na literatura [Bre05, Gom04] encontramos algumas abordagens para o desenvolvimento de ontologias. As abordagens foram analisadas sob o ponto de vista de construção e do produto resultante. Para construção, elas fornecem um conjunto de técnicas e atividades para o desenvolvimento de um modelo que represente o domínio modelado. As abordagens indicam quais atividades devem ser executadas, mas não indicam a ordem em que devem ser executadas. Como o desenvolvimento de ontologias não é um processo linear, a ordem de execução dos passos é definida pela equipe de construção, e devem ser executadas de acordo com o refinamento do modelo que se está construindo.

Do produto resultante, temos que a ontologia é formada por classes, relações e instâncias. Classes representam os conceitos, no seu sentido mais geral. Relações representam as associações entre os conceitos no domínio. E as Instâncias são utilizadas para definir um elemento dentro da ontologia.

Classes, relações e as instâncias são partes de uma estrutura ontológica, sobre qual se pode construir uma base de conhecimentos [Suc07, Bas07]. A ontologia fornece um conjunto de conceitos para descrever um determinado domínio, enquanto a base de conhecimento usa esses conceitos para descrever uma determinada realidade. Caso essa realidade seja modificada, a base de conhecimentos também o é; porém a ontologia permanecerá inalterada desde que o domínio se mantenha inalterado.

Dentre as vantagens do uso de ontologias na Ciência da Computação [Bre05], destacam-se:

- Fornecem um vocabulário para representação do conhecimento, que tem por trás uma conceituação que o sustenta, evitando, assim, interpretações ambíguas.
- Permitem o compartilhamento de conhecimento. Sendo assim, caso exista uma ontologia que modele adequadamente certo domínio de conhecimento, essa pode ser compartilhada e usada por pessoas que desenvolvam aplicações dentro desse domínio. Para exemplificar, considere que exista uma ontologia para o domínio de enciclopédias. Uma vez que essa ontologia está disponível, vários sistemas podem ser desenvolvidos no sentido de recuperar informações baseados em consultas semânticas, sem a necessidade de se fazer, para cada sistema de recuperação, uma análise do domínio de enciclopédia.
- Fornecem uma descrição exata do conhecimento. Diferente da língua natural, em que as palavras podem ter semântica totalmente diferente conforme o seu contexto, a ontologia por ser escrita em linguagem formal, não deixa espaço para as ambigüidades existentes na linguagem natural.
- Possibilitam fazer o mapeamento da linguagem da ontologia sem que com isso seja alterada a sua conceituação, ou seja, um mesmo conceito pode ser expresso em várias línguas.

Essas são as principais vantagens da utilização de ontologias. Existem outras, mas todas derivadas das citadas anteriormente. Das diversas abordagens estudadas, optamos pela abordagem proposta por Natalya Noy e Deborah McGuiness [Noy01], por sua simplicidade e fácil adaptação à aplicação que esta pesquisa propõe.

2.5.2. Abordagem de Noy e McGuiness

Noy e McGuiness propõem que uma ontologia pode ser construída através de sete passos que são executados de forma iterativa:

1. **Determinar o domínio e o escopo da ontologia**: neste passo se definem o escopo da ontologia, os tipos de respostas que ela fornecerá e como são feitas as atualizações.
2. **Considerar o reuso de outras ontologias**: busca-se por ontologias disponíveis que possam ser integradas a ontologia em desenvolvimento.

3. **Enumerar os termos importantes da ontologia**: sugere-se criar uma lista com os termos importantes do domínio, suas propriedades, seus relacionamentos e as suas respectivas descrições.
4. **Definir classes e hierarquia de classes**: da lista de termos criada no passo anterior, normalmente os substantivos são as classes. Define-se a hierarquia verificando os relacionamentos de hiponímia e hiperonímia e arranja-se os termos de forma a definir uma taxonomia onde os termos mais gerais descrevem os termos mais específicos.
5. **Definir as propriedades das classes**: os termos da lista criada no passo 3 que não se tornaram classes devem ser definidos como propriedades de classes.
6. **Definir os valores das propriedades**: definem-se características das propriedades, tais como, restrições, cardinalidade, intervalo de valores, etc.
7. **Criar instâncias**: criam-se as instâncias para as classes na hierarquia.

2.5.3. Ontologias aplicadas na expansão da consulta

O objetivo da expansão de consulta é encontrar, a partir da análise dos termos utilizados em uma consulta, novos termos relacionados que possam ser de interesse do usuário. Uma vez descobertos, esses novos termos são agregados à consulta original e a busca é então realizada.

A literatura aponta duas abordagens principais para a expansão de consulta: a abordagem probabilística [Cro00] e a ontológica [Bho07]. A abordagem probabilística se utiliza do conjunto de termos mais freqüentes, encontrados nas consultas realizadas previamente, para apresentar os termos candidatos para a expansão. Já a abordagem ontológica sugere a utilização das relações semânticas presentes na ontologia para encontrar os termos para a expansão.

O benefício da adoção da abordagem ontológica advém do fato que se pode escolher a relação semântica que é utilizada para fazer a expansão ou, ainda, um conjunto de relações que estão ligados a um termo. Essa característica permite desenvolver sistemas que utilizem a expansão para obter resultados mais relevantes para uma consulta realizada.

2.6. Considerações

Esta seção apresentou os recursos do processamento da língua natural que forneceram subsídios para a concepção desta proposta. A área de recuperação de informações se preocupa, principalmente com a organização da informação e mecanismos que facilitem recuperar a informação solicitada. Há vários modelos para recuperação de informações. Discutimos o modelo vetorial e o modelo probabilístico de recuperação, mostrando que cada modelo tem suas especificidades e que elas devem ser levadas em consideração no desenvolvimento de um sistema de recuperação. Escolhemos o modelo vetorial para o desenvolvimento dessa proposta, pois ele mantém independência entre os termos de indexação, característica que nos permitiu integrar uma ontologia para auxiliar o mecanismo de recuperação.

Uma ontologia provê mecanismos que permitem que o conhecimento sobre um domínio seja descrito. A descrição do domínio é feita na forma *sujeito-predicado-objeto*, onde o predicado representa o relacionamento entre um termo e o seu significado. Essa forma de descrição pode alavancar os resultados do sistema de recuperação quando este se utiliza desse conhecimento para recuperar informações relacionadas às necessidades do usuário.

A próxima seção apresenta o desenho da solução desenvolvida nesta pesquisa. Apresenta também trabalhos recentes de recuperação semântica e faz uma discussão sobre a aplicabilidade da nossa proposta de recuperação semântica.

3. DESENHO DA SOLUÇÃO

A solução proposta é dividida em duas fases que são executadas em momentos distintos. Cada fase é constituída por componentes específicos e outros compartilhados, conforme descreve a Figura 3:

- Fase de recuperação de documentos: caso de uso, contextualização, recuperação e *ranking* e a resposta do sistema;
- Fase de preparação da ontologia: configuração do ambiente e elicitação de palavras-chave;
- Componentes compartilhados: usuário, pré-processamento, extração de termos, ontologia e corpus.

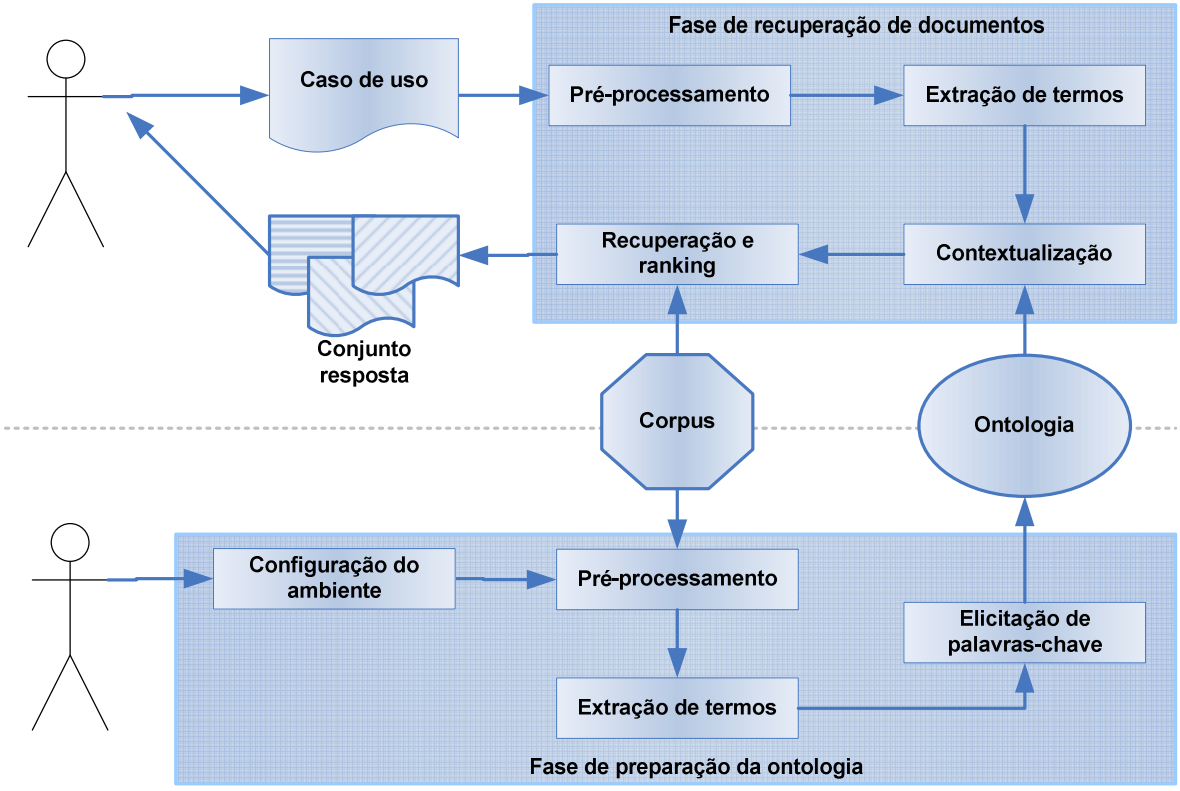


Figura 3 – Fases e etapas da solução proposta

A fase de preparação da ontologia tem o objetivo de auxiliar o usuário a criar ou expandir uma ontologia que descreva o corpus. A etapa de configuração do ambiente é onde os especialistas do projeto identificam os casos de uso com potencial de reutilização. A etapa de pré-processamento aplica ao corpus a remoção de *stopwords.* A etapa de extração de termos tem o objetivo de montar uma lista na forma "r*adical = conjunto de termos"*. Essa etapa faz isso agrupando todos os termos extraídos pelo radical dos

termos. Todos que tem o mesmo radical são agrupados juntos. A etapa de elicitação de palavras-chave apresenta em uma interface todos os termos extraídos e permite ao usuário descartar os termos que não são importantes para descrição do domínio. A ontologia é onde o usuário faz o enriquecimento semântico do corpus, ou seja, usa a lista de termos resultante para descrever os termos e relações presentes no domínio.

A fase de recuperação de documento se inicia com a necessidade do usuário em recuperar casos de uso similares a outro caso de uso informado. Nesse processo são aplicadas as técnicas de pré-processamento e extração de termos. A lista de termos é então enriquecida com as relações presentes na ontologia e então é utilizada para recuperar casos de uso similares. Após é aplicada uma função de similaridade e os casos de uso em ordem de similaridade são apresentados ao usuário.

3.1. Trabalhos relacionados

Os atuais sistemas de recuperação de informações são baseados em pesquisa por palavra-chave [Yat99], onde dada uma expressão de busca, o sistema retorna um conjunto de documentos que contenham alguns ou todos os termos presentes na expressão de busca e apresenta esses documentos ordenados por algum critério de relevância.

Bast e seus co-autores [Bas07] dizem que, se por um lado esses sistemas já foram considerados suficientes para resolver os problemas de recuperação, hoje em dia estão superados, pois a expectativa é que os novos sistemas sejam capazes de recuperar informações considerando também a semântica presente na expressão de busca. Os autores apresentam um sistema que usa uma ontologia aliada a uma interface interativa. O papel da interface é apresentar as relações semânticas presentes na ontologia que sejam relacionadas aos termos informados pelo usuário e desta forma conduzí-lo a formular uma expressão de consulta que representa o seu real interesse de informação.

Hu e seus co-autores apresentam em [Hu+08] um método que utiliza o *Wikipédia* como fonte para desenvolvimento de uma base semântica. A base semântica é constituída por sinonímia, hiperonímia e outras relações entre termos, que são extraídas de forma automática dos artigos do *Wikipédia*. A idéia que fundamenta a pesquisa é que os relacionamentos presentes na base semântica alavancariam os resultados de um sistema

de *clusterização* de documento. Os resultados da pesquisa apresentam uma melhora na faixa de 16,20 a 18,80 % na *clusterização* quando utilizadas a base semântica.

Chu-Carroll e seus co-autores desenvolvem em [Car+06] um método de recuperação de informações que usa um corpus etiquetado para melhorar os resultados da recuperação. O método deles se interessa por etiquetas sobre conceitualização, restrições e outras relações entre os termos. A etiqueta de conceitualização é utilizada como fator contextual, expandindo a consulta com termos relacionados. A etiqueta de restrição é utilizada para direcionar a consulta a um assunto específico, e as outras relações entre os termos são apresentadas ao usuário para que este faça a escolha dos fatores contextuais, controlando assim a abrangência da expansão de termos.

3.2. Considerações

Apesar de existirem diferenças entre os sistemas apresentados na subseção 3.1, tem-se como pontos principais:

- **Recursos de pré-processamento do corpus**: A utilização de listas de *stopwords* com o intuito de diminuir a quantidade de termos utilizados na indexação e uso de *stemming*, agrupando os termos sob um mesmo radical. Em [Bas07] *stemming* é utilizado como estratégia para encontrar as relações semânticas similares e em [Hu+08] é utilizado na construção dos relacionamentos da base semântica.
- **Indexação do corpus**: A indexação do corpus é realizada utilizando variações da lista invertida de termos, onde o peso do termo em um documento é definido em relação a sua freqüência no corpus.
- **Ontologias como núcleo do sistema de recuperação**: As ontologias são utilizadas como base para o mecanismo de recuperação, pois disponibilizam aos sistemas relações que são utilizadas para expandir ou para se especializar os resultados da recuperação, de acordo com a preferência do usuário.

Esta pesquisa compartilha os pontos em comum apresentados, fornecendo uma metodologia que organiza os recursos do processamento da língua em duas fases: a construção da ontologia e a recuperação dos documentos. A metodologia é apresentada na próxima seção.

4. RECUPERAÇÃO SEMÂNTICA DE DOCUMENTOS DE CASO DE USO

Essa seção descreve como os recursos do processamento da língua natural foram utilizados na construção de um sistema para recuperação semântica de documentos de caso de uso. Para a construção do sistema, especificou-se um método que organiza os trabalhos de preparação e recuperação em duas fases. A primeira descreve a forma como o corpus deve ser preparado e como os termos utilizados na preparação podem ser utilizados na definição das palavras-chave do domínio. A segunda fase explica como a recuperação de documentos é realizada, e mostra como os relacionamentos descritos na ontologia são utilizados para melhorar os resultados da recuperação.

4.1. Fase de preparação do corpus

A fase de preparação do corpus é subdividida em quatro etapas (
Figura 4): configuração do ambiente, extração dos termos, elicitação das palavras-chave do domínio e enriquecimento semântico da lista de termos. O produto final dessa fase é uma ontologia que descreve o domínio modelado no corpus. Na seqüência explicaremos em detalhes cada etapa dessa fase.

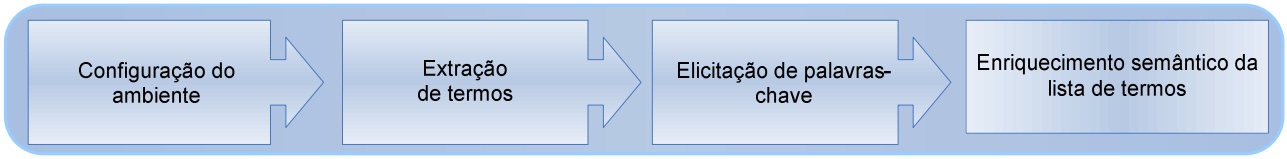


Figura 4 – Fase de preparação da ontologia

4.1.1. Configuração do ambiente

A etapa de configuração do ambiente está interessada na identificação de casos de uso com potencial de reutilização em diversos domínios e deve ser executada pelos especialistas do projeto. Entenda-se por potencial de reutilização, os casos de uso que elicitam comportamentos sistêmicos que podem ocorrer em diversos domínios com poucas variações em sua forma, e estejam implementados e testados. O exemplo mais comum de caso de uso com alto potencial de reutilização são os que descrevem os cadastros, normalmente denominados de CRUD[1].

[1] Acrônimo da expressão ***C****reate,* ***R****etrieve,* ***U****pdate e* ***D****elete*

A forma geral de executar essa etapa é verificar se, ao abstrair o domínio do cenário descrito no caso de uso, o que resta pode ser reaproveitado em outras situações. O problema dessa forma geral é a dificuldade de executá-la em projetos onde o volume de casos de uso seja muito grande, problema este que deverá ser gerenciado pela equipe para que o resultado dessa fase seja satisfatório.

Uma vez identificados os casos de uso com potencial de reutilização, os especialistas do projeto devem verificar se os componentes de software desenvolvidos para implementá-los são genéricos o bastante para que possam ser reutilizados em outros domínios. Caso não sejam, esses componentes devem ser refatorados[2], a fim de torná-los genéricos.

Uma dúvida comum nessa etapa é como identificar todos os componentes de software que implementam um determinado caso de uso. A resposta a esse questionamento é que o projeto de software deve contar com um processo de rastreabilidade forte. Para que o método descrito neste trabalho funcione, rastreabilidade é um requisito necessário. Isso pode causar um impacto inicial, mas deve-se ter em mente que qualquer empresa que deseje alcançar CMMI nível 2 deverá atender a prática específica **REQM[3] SP 1.4-2 – Manter a rastreabilidade bidirecional dos requisitos**.

Essa etapa é executada dentro do EA. Uma vez que os especialistas do projeto tenham identificado os casos de uso desejados, estes devem ser selecionados. Ao executar o protótipo – mostrado no item A da Figura 5 – o conjunto selecionado é enviado como entrada para a etapa extração de termos e a etapa de configuração de ambiente termina.

Uma vez concluída a etapa de configuração do ambiente, passamos para a etapa de extração dos termos presentes nos casos de uso identificados, descrito com detalhes na próxima subseção.

[2] Processo de modificar os componentes de software sem que suas funcionalidades sejam alteradas.

[3] Sigla usada no CMMI para Gestão de requisitos

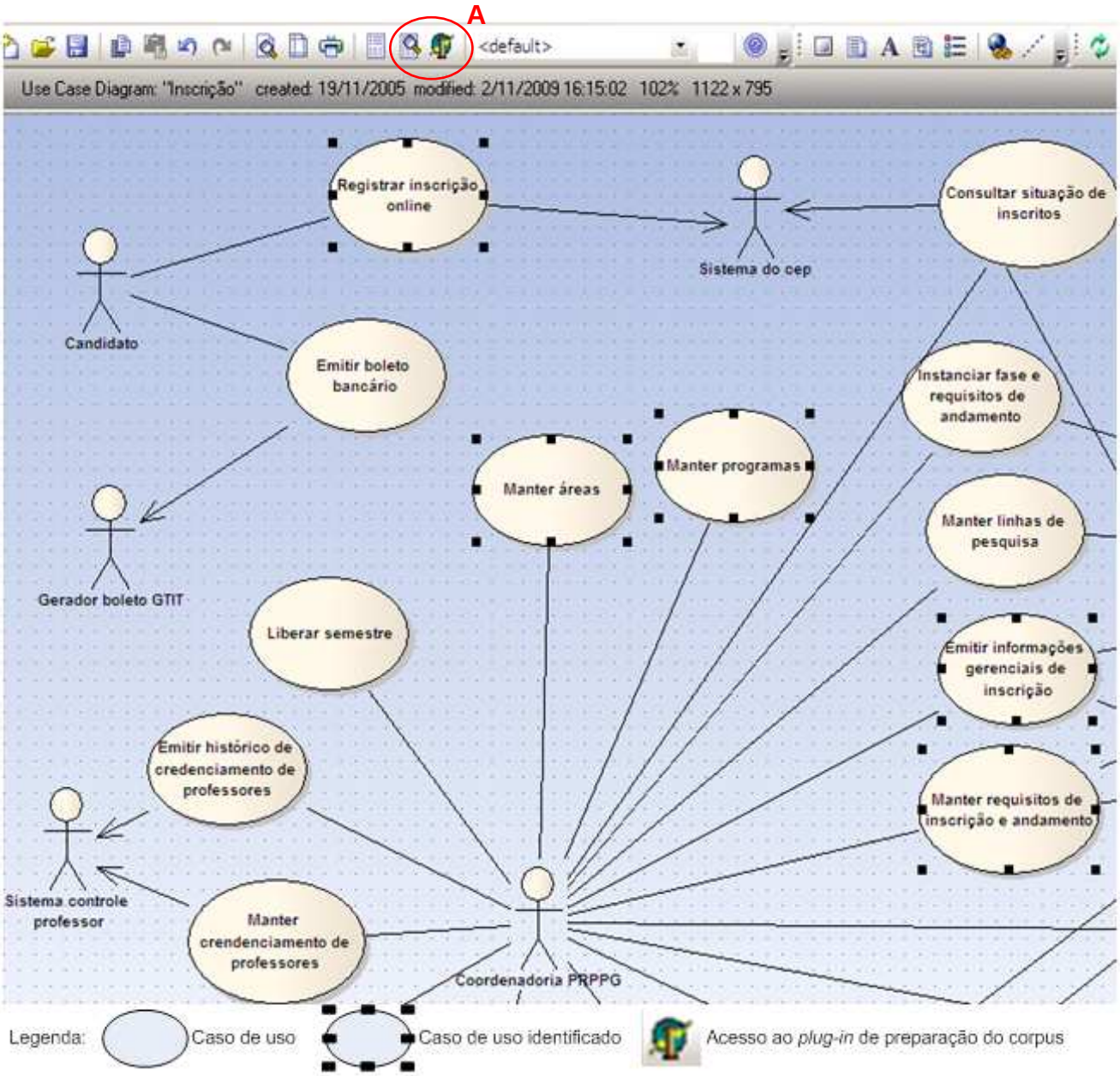
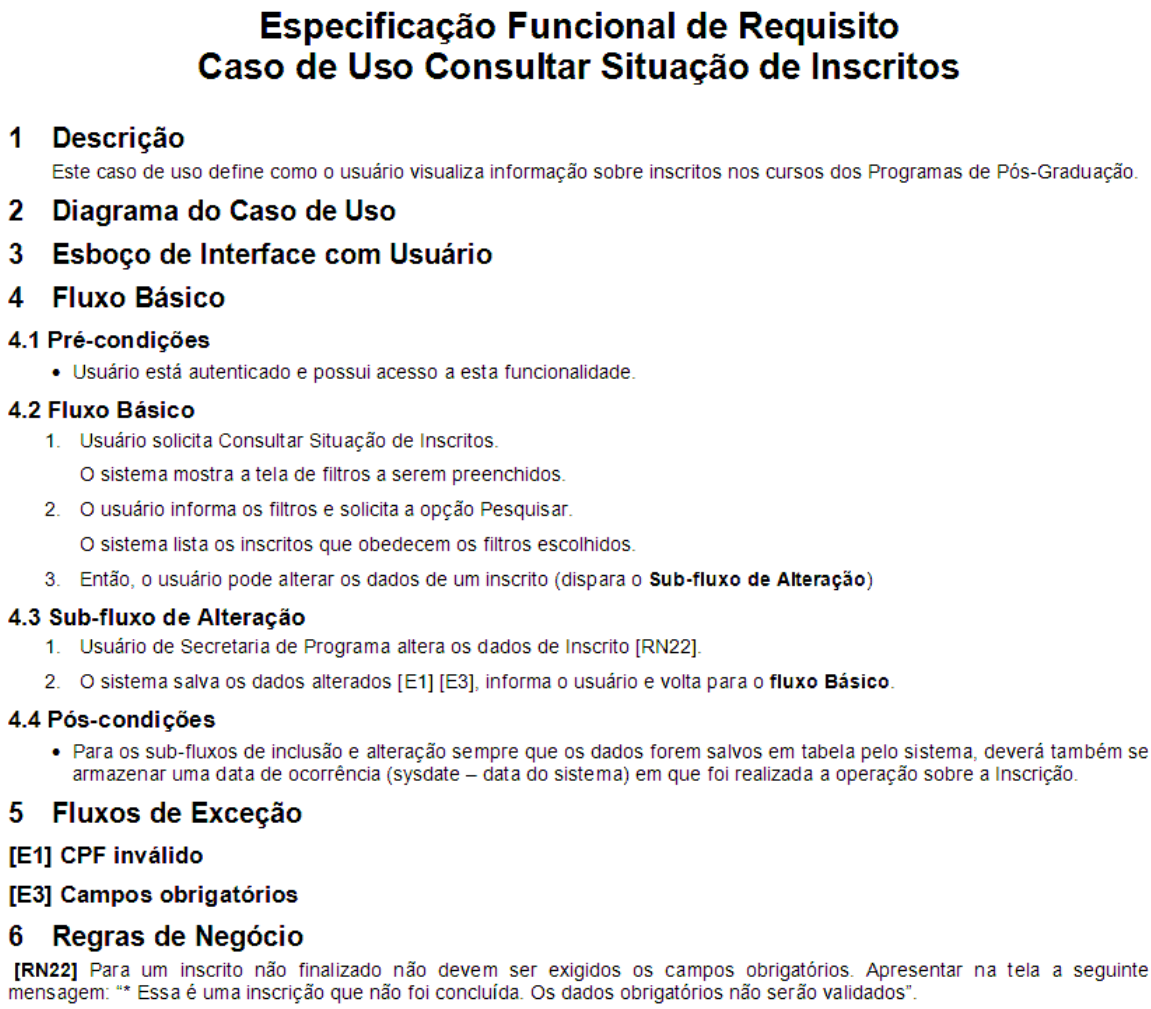


Figura 5 – Exemplo de configuração de ambiente: casos de uso selecionados para formação do corpus

4.1.2. Extração de termos dos casos de uso

A realização dessa etapa é feita de forma automatizada, onde a entrada para o software é um conjunto de casos de uso de um mesmo domínio, como mostrado na Figura 5. Por ser controlada por software, sugerimos que um analista de sistema se responsabilize por essa etapa. Descreve-se abaixo o funcionamento interno do software e as técnicas utilizadas para construí-lo.

Conceitualmente, o caso de uso é um modelo que descreve como diferentes tipos de usuários interagem com um sistema para resolver um problema. Para tal, ele descreve as metas dos usuários, as interações entre os usuários e o sistema, bem como o comportamento necessário do sistema para satisfazer estas metas. Estruturalmente, um caso de uso é um documento composto por seções, parágrafos e itens. A Figura 6 mostra um documento de caso de uso mínimo (parcialmente preenchido por questões de espaço).



Especificação Funcional de Requisito
Caso de Uso Consultar Situação de Inscritos

1 Descrição

Este caso de uso define como o usuário visualiza informação sobre inscritos nos cursos dos Programas de Pós-Graduação.

2 Diagrama do Caso de Uso

3 Esboço de Interface com Usuário

4 Fluxo Básico

4.1 Pré-condições

- Usuário está autenticado e possui acesso a esta funcionalidade.

4.2 Fluxo Básico

1. Usuário solicita Consultar Situação de Inscritos.
   O sistema mostra a tela de filtros a serem preenchidos.
2. O usuário informa os filtros e solicita a opção Pesquisar.
   O sistema lista os inscritos que obedecem os filtros escolhidos.
3. Então, o usuário pode alterar os dados de um inscrito (dispara o **Sub-fluxo de Alteração**)

4.3 Sub-fluxo de Alteração

1. Usuário de Secretaria de Programa altera os dados de Inscrito [RN22].
2. O sistema salva os dados alterados [E1] [E3], informa o usuário e volta para o **fluxo Básico**.

4.4 Pós-condições

- Para os sub-fluxos de inclusão e alteração sempre que os dados forem salvos em tabela pelo sistema, deverá também se armazenar uma data de ocorrência (sysdate – data do sistema) em que foi realizada a operação sobre a Inscrição.

5 Fluxos de Exceção

[E1] CPF inválido

[E3] Campos obrigatórios

6 Regras de Negócio

[RN22] Para um inscrito não finalizado não devem ser exigidos os campos obrigatórios. Apresentar na tela a seguinte mensagem: "* Essa é uma inscrição que não foi concluída. Os dados obrigatórios não serão validados".

Figura 6 – Exemplo de um documento de caso de uso.

No PLN a construção da lista de termos do corpus se inicia com análise léxica, passa pela eliminação de *stopwords* e conclui com a normalização dos termos. A análise léxica tem o objetivo de tratar números, hífens, símbolos, pontuação, e maiúsculas e minúsculas. A forma geral de aplicação da análise léxica é converter todos os termos em minúsculas, eliminar hífens, números e demais símbolos e remover termos que tenham seqüência de

dígitos. Devido às especificidades dos documentos de caso de uso, chamamos a atenção para o tratamento de números e símbolos.

Em documentos de caso de uso, é corriqueira a utilização do primeiro e/ou segundo caractere quando se deseja referenciar um item específico de alguma seção. Por exemplo, os termos [RN_n] e [E_n] (onde $_n$ é um seqüencial numérico) mostrados na Figura 6, se referindo as seções **R**egras de **N**egócio e Fluxos de **E**xceção, respectivamente. Essas seções descrevem requisitos não funcionais. Requisitos não funcionais são regras de domínio que devem ser satisfeita quando uma determinada operação for executada e, por esse motivo, uma mesma regra pode ser referenciada em vários casos de uso. A análise léxica preservará termos que estejam envolvidos por colchetes.

É de utilização corriqueira também, o uso de símbolos, como a barra (/), quando se deseja relacionar especificidades relacionadas a um determinado termo. Aproveitaremos esse estilo de escrita para extrair automaticamente relações do tipo “é um” dos títulos dos casos de uso. Essas relações serão sugeridas ao usuário quando a etapa de enriquecimento semântico da lista de termos for executada. A Tabela 2 mostra um exemplo de extração de relações.

Tabela 2 – Exemplo de padrão *termo/termo* no título do caso de uso

***Termo/Termo* no título do caso de uso**	
Exemplo	**Resultado**
Manter log de utilização/auditoria	Log → utilização Log → auditoria
Gerar relatório de dados do inscrito/candidato/selecionado	Relatório → inscrito Relatório → candidato Relatório → selecionado

Stopwords são termos não significativos, como artigos e preposições, mas não limitado somente a estes. Por exemplo, a seção *descrição* da Figura 6 se inicia com a frase:

“*Este caso de uso ...*”

Esta é uma palavra comum na seção de introdução dos casos de uso e que deve ser tratada como *stopword* por não agregar valor à seção de introdução. Para fins de

implementação, as *stopwords* podem ser tratadas como *uma lista*, onde os seus elementos representam termos que devem ser retirados do documento que está sendo processado, abordagem utilizada nesta pesquisa. Removidas as *stopwords*, a lista de termos é obtida através de um algoritmo guloso, que utiliza os espaços em branco presentes entre os termos como delimitador. Assim que se extrai um termo, o software verifica de qual seção aquele termo foi extraído, vincula ao termo o nome da seção e os termos da lista são normalizados (*stemming*).

4.1.3. Elicitação de palavras-chave do domínio

Aplicar a técnica de remoção de *stopwords* é necessária já que melhora o resultado da lista de palavras, mas não garante a qualidade dessa lista, pois ainda podem aparecer termos que não tem representatividade no domínio, às vezes por serem genéricos demais ou específicos demais.

Esta etapa está interessada em melhorar a qualidade dessa lista e é realizada pelo especialista no domínio. A principal tarefa desse especialista é descartar todos os termos que, no seu entendimento, não agrega informações para representação do domínio.

Neste ponto existe uma discussão pertinente: existem termos que só fazem sentido no domínio quando analisados em conjunto, são os chamados sintagmas, que podem ser nominais ou verbais. A etapa de extração de termos considera somente os termos, não extraindo sintagmas. Por esse motivo, o especialista no domínio deve ter cuidado ao analisar os termos, pois a falta de extração de sintagmas é um desafio que o especialista no domínio terá que vencer para que o resultado desta etapa e da próxima seja satisfatório.

Para auxiliar o usuário na escolha de “melhores” termos que representem o documento, os termos são normalizados e posteriormente utilizamos a medida de cálculo de freqüência inversa (TF-IDF) para calcular o peso que um determinado *stem* tem em um documento em função dos outros documentos do conjunto. A Figura 7 mostra no protótipo desenvolvido a visualização dos *stems,* dos termos agrupados, a freqüência calculada e a opção para o usuário manter ou descartar termos.

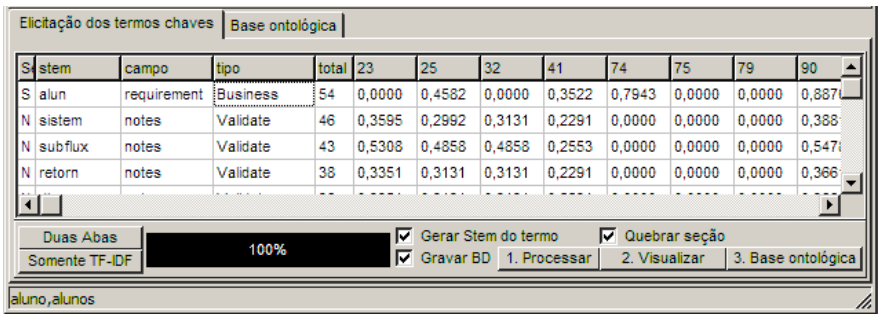


Figura 7 – Protótipo: etapa de elicitação das palavras-chave

Uma vez descartado os termos, o que resta é uma lista com termos de alta representatividade no domínio. Esses termos são usados como produto para a criação de uma ontologia, apresentada na próxima seção.

4.1.4. Criação de uma ontologia através do enriquecimento semântico da lista de termos

Esta etapa, a última da preparação do corpus, está interessada em enriquecer a lista de termos resultante da etapa anterior com relações semânticas. Ela deve ser executada pelos analistas de sistemas em conjunto com os especialistas do domínio. O resultado final é uma ontologia que descreve o domínio, sob o ponto de vista dos documentos de caso de uso e dos especialistas de domínio envolvidos no projeto.

O papel do analista de sistemas nesta etapa é apoiar os especialistas do domínio no estudo dos termos resultantes da etapa anterior. Busca-se identificar o tipo de relação existente entre esses termos, agregando assim novos termos a lista ou mesmo relacionando os termos existentes. É recomendável iniciar a identificação de novos termos e suas relações através glossário de termos do projeto. As relações semânticas – Figura 8 – previstas neste trabalho são as de sinonímia, hiperonímia e hiponímia [Gom04].

As relações de sinonímias dizem respeito aos sinônimos. Ou seja, busca-se identificar os sinônimos dos termos, incluindo aqui os jargões utilizados no domínio. Por exemplo, no domínio de pós-graduação de uma universidade, é comum que os termos “aluno_regular” seja utilizado como sinônimo do termo “aluno_cursando_disciplina”.

As relações de hiperonímia se interessam em classificar os termos quando a sua generalidade. À medida que as relações de generalidade são encontradas, inicia-se a criação de uma estrutura hierárquica, que quando mais superior se encontra um termo,

maior é a generalização em relação aos termos que estão mais abaixo na estrutura hierárquica. Por exemplo, o termo “relatório”, mostrado na
Figura 8.

Enquanto as relações de hiperonímia identificam os termos mais genéricos, as relações de hiponímia fazem exatamente o contrário. Buscam identificar os termos mais específicos presentes no domínio. Por exemplo, o termo “histórico_aluno”, mostrado na
Figura 8

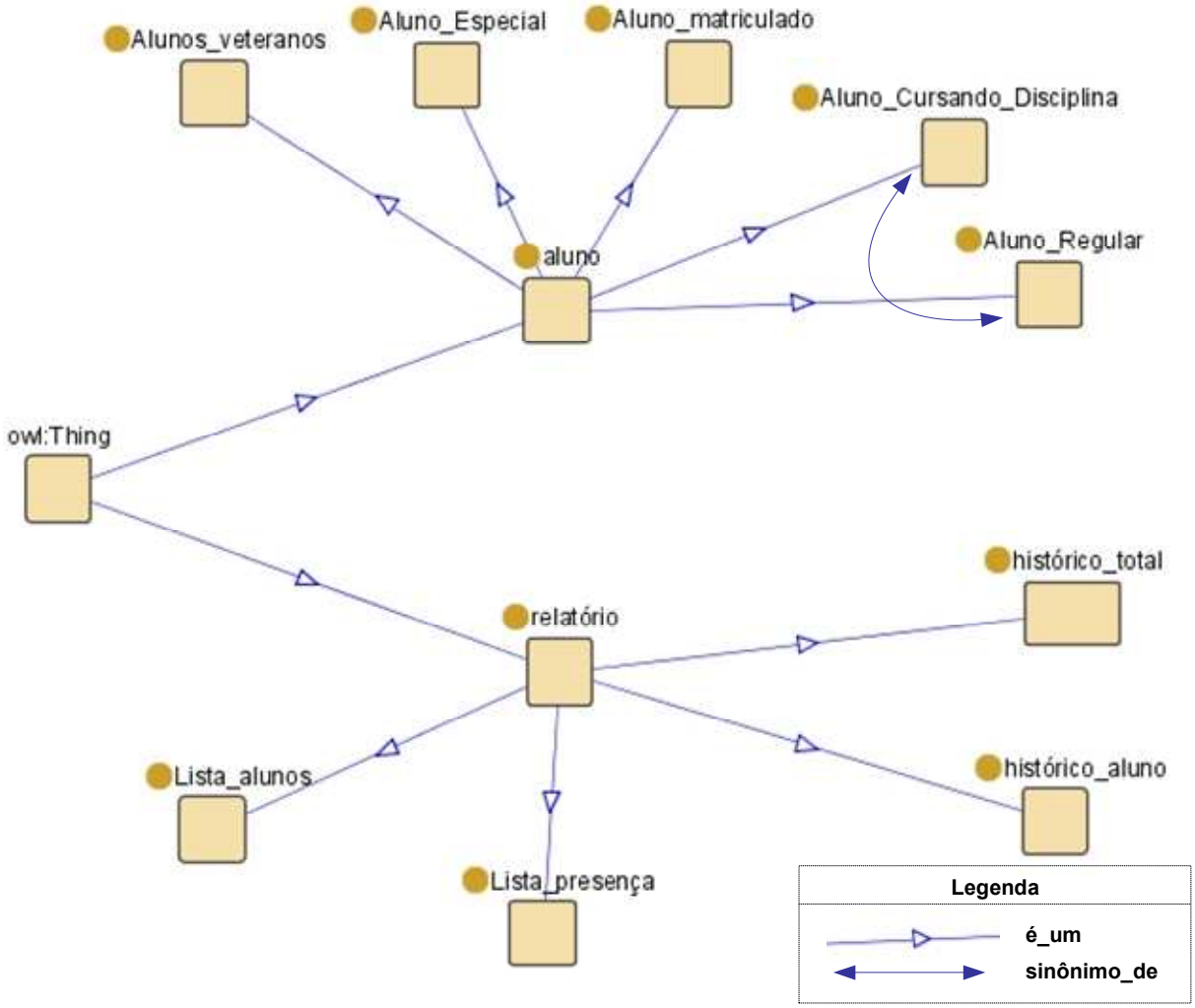


Figura 8 – Exemplo de relações em termos presentes no domínio de um departamento de pós-graduação de uma faculdade.

.

Para melhor compreensão dos conceitos hiperonímia e hiponímia, mostramos na
Figura 8 essas relações presentes em alguns termos extraídos do domínio de um departamento de pós-graduação de uma universidade.

A linguagem recomendada pelo W3C para descrição de uma ontologia é o OWL[4]. Esta especificação é baseada em XML e descreve em um arquivo a ontologia desenvolvida. Trabalhar com arquivos em uma arquitetura concorrente causaria impacto no desenvolvimento do protótipo de apoio ao método proposto. Desta forma modelamos a ontologia utilizando um banco de dados relacional. Existe na literatura discussões sobre a utilização do modelo relacional para expressar uma ontologia, mostrando assim que é possível gerar o arquivo OWL de uma ontologia a partir de um modelo relacional. Sobre esse assunto sugere-se a leitura do trabalho desenvolvido por Gomez-Perez *et al* em [Gom04].

Uma ontologia é composta por **classes, propriedades e indivíduos**, onde:

Classes descrevem **o que existe** em determinado domínio;
Propriedades descrevem **relacionamentos e outras informações** de uma classe; e
Indivíduos descrevem as **instâncias** das classes existentes.

Nesta etapa o analista do domínio deve se focar em encontrar outras classes existentes no domínio que não foram apresentadas na lista de termos e descrever os relacionamentos dessas classes. A Figura 9, exemplifica a descrição de sinonímia entre as classes listagem e relatório, que são mostradas na
Figura 8.

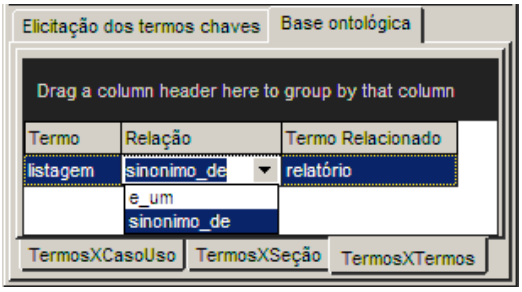


Figura 9 – Interface para descrição das relações de sinonímia, hiperonímia e hiponímia.

Uma vez que se tenha concluído essa etapa, tem-se uma ontologia que descreve o domínio modelado nos casos de uso, enriquecido com conhecimentos de um especialista.

[4] A recomendação é encontrada em
http://www.w3.org/2004/OWL/

Essa ontologia é utilizada na recuperação de documentos de casos de uso, descrita na próxima seção.

4.2. Fase de recuperação de documentos

A fase de recuperação de documentos é subdividida em quatro etapas (Figura 10): caso de uso, expansão dos termos de busca, resolução de ambigüidades e ranking de resultado. Ao final dessa fase o usuário terá como resposta todos os casos de uso potencialmente reutilizáveis que sejam semanticamente similares ao caso de uso utilizado como entrada na pesquisa, desde que o conjunto de casos de uso potencialmente reutilizáveis já tenham sido previamente indexados pela ferramenta. Na seqüência explicaremos em detalhes cada etapa dessa fase.

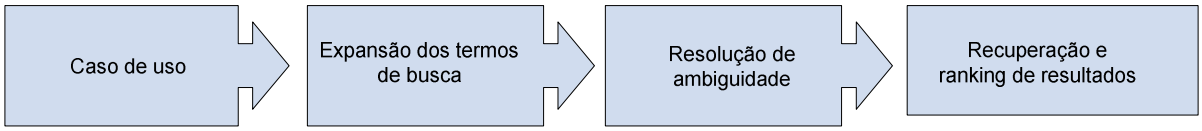


Figura 10 – Segunda fase: recuperação semântica do documento

4.2.1. Caso de uso

Nesta etapa o usuário especifica um caso de uso dentro do EA. É importante dizer que o caso de uso precisa ser especificado o mais completo possível, pois o método utiliza as informações presentes nas seções do caso de uso para gerar a lista de termos, essa lista é o princípio da recuperação. Após especificar o caso de uso, chama-se o protótipo de recuperação, passo explicado a seguir.

4.2.2. Expansão dos termos de busca

O objetivo dessa etapa é expandir a lista de termos que são utilizados para a pesquisa. A expansão é realizada utilizando os relacionamentos de sinonímia, hiperonímia e hiponímia que foram definidos na etapa de *enriquecimento semântico da lista de termos.* Para que ocorra a expansão é necessário inicialmente construir a lista de termos. Essa lista é construída utilizando o mesmo algoritmo descrito na etapa de *extração de termos*. Com a lista de termos pronta, temos que:

Expansão utilizando sinonímias: os sinônimos são relações com propriedade de simetria. Isso quer dizer que, se um dado termo “A” é sinônimo do termo “B”, “B” também é sinônimo de “A”. Assim, para cada termo da lista, busca se na ontologia os seus respectivos sinônimos e este é adicionado ao final da lista de termos, caso ainda não exista. Quando a lista de termos é expandida utilizando sinonímias, diz-se que a lista de termos está *contextualizada*.

Expansão utilizando hiperonímia: as relações de hiperonímia são realizadas com o objetivo de generalizar uma consulta. Desta forma, uma lista de termos enriquecida com essas relações tem tendência a ser mais abstrata, aumentando a cobertura da pesquisa em relação aos corpora. Quando a lista de termos é expandida utilizando as hiperonímias, diz-se que a lista de termos está *generalizada*.

Expansão utilizando hiponímia: já as relações de hiponímia, que são automaticamente especificadas no momento em que se define a relação de hiperonímia, têm o objetivo de especializar os termos da consulta a um assunto ou jargão. Quando a lista de termos é expandida utilizando hiponímia, diz-se que a lista de termos está *especializada*.

Chama-se a atenção para o comportamento antagônico das relações de hiperonímia e hiponímia quando utilizadas na expansão de termos de consulta. Desta forma, é necessário que o usuário escolha o tipo de comportamento que a expansão terá, podendo ser mais especializada ou mais genérica.

4.2.3. Resolução de ambigüidade

A ambigüidade é um desafio enfrentado por sistemas que lidam com a língua natural e diz respeito ao fenômeno lingüístico que faz com que um termo tenha significados distintos. Recomendações [Coc00] para a escrita de documentos de caso de uso chamam a atenção do designer para que este evite o uso de termos ambíguos, e quando o mesmo se fizer necessário, deve-se adotar um dicionário de terminologias onde o termo é descrito e todo o seu uso se refere àquela definição adotada. Ou seja, todo o projeto que adotar aquele dicionário tem uma definição única para os termos que no uso cotidiano são ambíguos.

Como os dicionários são orientados a projetos, projetos diferentes podem ter dicionários que definam um mesmo termo de forma distinta. Problema semelhante foi relatado em

[Bai07] no desenvolvimento de perfis que refletissem os interesses e necessidades do usuário em um sistema de recuperação de informações de uso geral. A solução adotada pelos autores foi dividir os interesses dos usuários em perfis organizados por assuntos. A conclusão obtida no estudo é que dividir os perfis por assunto melhorou o resultado do módulo de desambiguação da solução.

De forma similar [Bai07], nesta pesquisa cada projeto indexado diz respeito a um domínio (ou outro ponto de vista de um mesmo domínio) e cada domínio deve ter a sua própria base ontológica. No momento da recuperação, a ferramenta apresenta os casos de uso recuperados e para qual projeto aquele caso de uso foi especificado. Visto que os usuários que utilizarão esta solução estarão interessados em artefatos de software utilizados na realização dos casos de uso recuperados e não no caso de uso em si, essa pode ser uma solução viável para o problema da ambigüidade de termos entre projetos de software distintos.

4.2.4. Recuperação e ranking de resultados

Como resultado da etapa Expansão dos termos de busca (4.2.2), temos uma lista de termos, a sua seção e o respectivo peso do termo no documento. Essas informações são consultadas na ontologia com o objetivo de recuperar documentos similares. A consulta às instâncias dos casos de uso presentes na ontologia é realizada utilizando a linguagem SQL, com restrições no formato:

(Stem = ?stem and Secao = ?secao)
< or (Stem = ?stem1 and Secao = ?secao1) >
< or (Stem = ?stem2 and Secao = ?secao2) >
< or (Stem = ?stem3 and Secao = ?secao3) >

Onde:

Stem : propriedade da classe caso_uso
Secao: propriedade da classe caso_uso
< >: o uso dos sinais de maior e menor denota que o comando é opcional.

Uma vez recuperado casos de uso potencialmente similares ao caso de uso informado para consulta é necessário aplicar uma função que defina o quanto cada caso de uso é similar ao procurado. Na literatura consultada, boa parte dos trabalhos de recuperação de

informações adotou com sucesso a função de similaridade pelo cálculo do cosseno e por esse motivo também a utilizaremos. A função do cosseno é definida como:

Equação 10

$$similaridade(x,y) = \frac{\sum_{m=1}^{t}(w_{m,x} \star w_{m,y})}{\sqrt{\sum_{m=1}^{t}(w_{m,x})^2} \star \sqrt{\sum_{m=1}^{t}(w_{m,y})^2}}$$

onde:

- x: vetor que representa um caso de uso recuperado na ontologia.
- y: vetor que representa o caso de uso utilizado como expressão de busca
- $w_{m,x}$: peso do *m*-ésimo elemento do vetor x
- $w_{m,y}$: peso do *m*-ésimo elemento do vetor y

Os valores de similaridade entre a expressão de busca e cada um dos documentos do corpus são utilizados no ordenamento dos documentos recuperado. Assim o resultado da busca é um conjunto de documentos ordenados pelo grau de similaridade entre cada documento e a expressão de busca. Esse ordenamento permite restringir o resultado a um número máximo de documentos desejados ou ainda definindo um limite mínimo para o valor da similaridade. Desta forma o usuário pode definir para a máquina de busca recuperar somente os documentos com um valor mínimo de relevância em relação à expressão de consulta.

4.3. Desenvolvimento de protótipo para mostrar a viabilidade do método

Para que o método proposto fosse avaliado, desenvolvemos dois protótipos: um que executa a fase de preparação, guiando os especialistas pelas etapas descritas e apoiando a construção da ontologia, e outro que executa as etapas da fase de recuperação, apresentando ao usuário casos de uso previamente preparados que sejam similares a um caso de uso dado como entrada para a ferramenta. Nas seções que seguem, detalhamos questões tecnológicas e apresentamos as interfaces construídas.

4.3.1. Enterprise Architect (EA)

O EA (Figura 11) é uma ferramenta *CASE* que suporta o desenvolvimento de sistemas utilizando a UML como linguagem padrão. É altamente configurável e extensível, oferecendo uma API que permite a construção de *plugins* que expandam o funcionamento da ferramenta.

As principais funcionalidades do sistema relacionadas a esta pesquisa são:

- Caso de uso: a ferramenta disponibiliza um formato de documento de caso de uso conforme o modelo mínimo encontrado em [Lar07, Coc00] mostrado na Figura 6, além de oferecer possibilidade de customização da estrutura do documento de caso de uso, respeitando a forma <Seção> <Parágrafo descrevendo a seção> discutida na seção 4.1.2.
- Rastreabilidade de artefatos: a ferramenta suporta *links* de rastreabilidade que integram o caso de uso com todos os artefatos construídos em sua implementação. Requisito necessário para cumprir a etapa Configuração de ambiente, discutido na seção 4.1.1.
- Suporte a metodologia: pode ser configurado para trabalhar apoiando uma metodologia de desenvolvimento, como o processo unificado.
- Suporte a plugins: o modelo de componentes oferecido com a ferramenta permite acesso aos objetos gerenciados pela ferramenta de duas formas: Objetos COM+ e acesso direto ao modelo de componentes através de ODBC.

Existem outras ferramentas *CASE* com suporte a especificação de casos de uso, dentre as mais conhecidas estão o IBM RequisitePro e o Borland Caliber RM. Ambas as ferramentas provêm funcionalidades similares ao EA. A escolha pelo EA foi feita após um levantamento sobre as ferramentas CASE utilizadas nas empresas desenvolvedoras de software de Cuiabá-MT, onde constatamos uma boa aceitação dessa ferramenta.

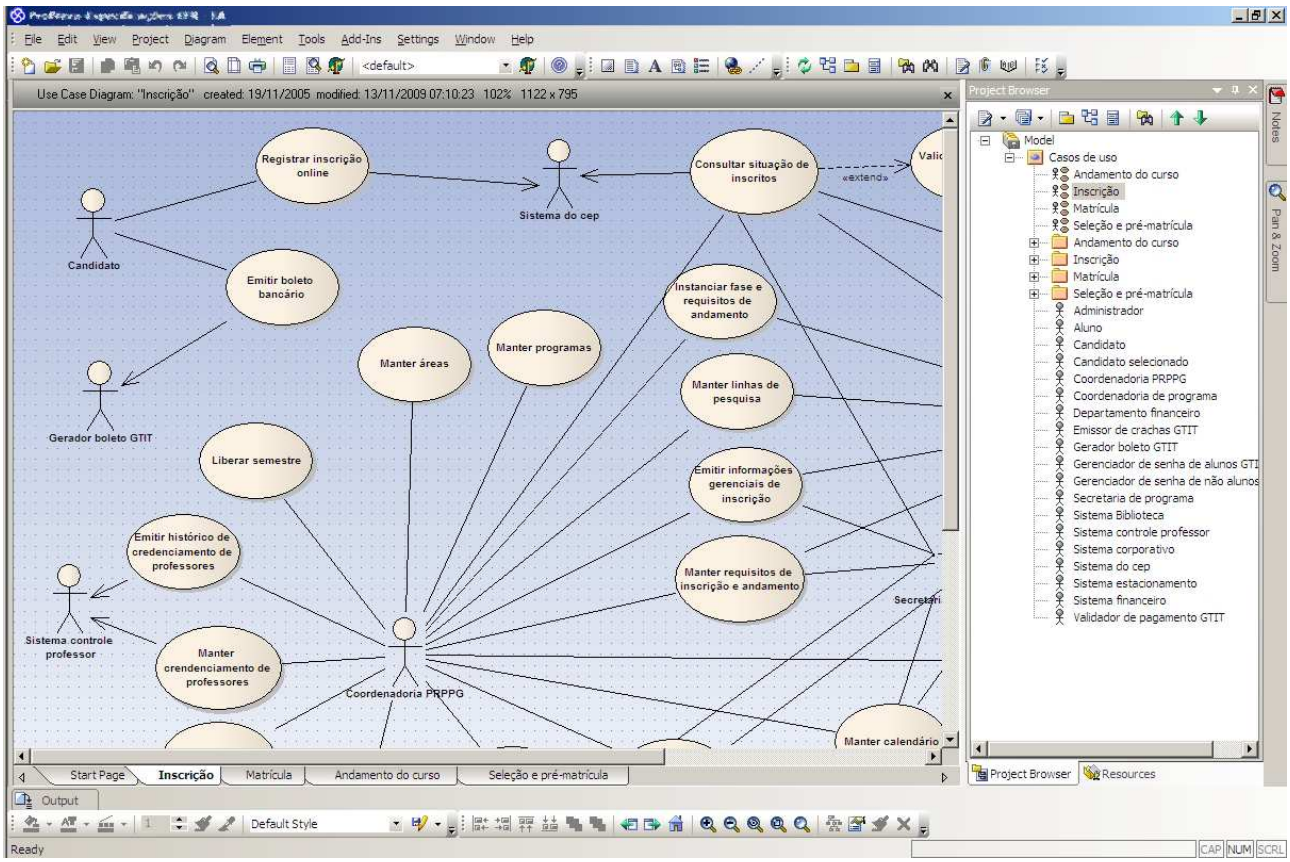


Figura 11 – Interface da ferramenta Enterprise Architect

4.3.2. *Plugins:* preparação e recuperação

O *plugin* de preparação guia o especialista nas etapas de Extração de termos (4.1.2), Elicitação de palavras-chave (4.1.3) e Enriquecimento semântico da lista de termos (4.1.4) e faz isso através de duas interfaces. A primeira, mostrada na Figura 12, faz a extração dos termos e apresenta ao usuário, permitindo que ele descarte termos que não tem representatividade no domínio. Para auxiliar o usuário na escolha de termos, o sistema usa a medida TF-IDF para apresentar ao usuário o peso que cada termo tem nos documento do corpus. A segunda interface, mostrada na Figura 13, lança automaticamente as relações encontradas na extração de termos e permite ao usuário criar as relações que em seu entendimento descrevem os relacionamentos entre os conceitos do domínio.

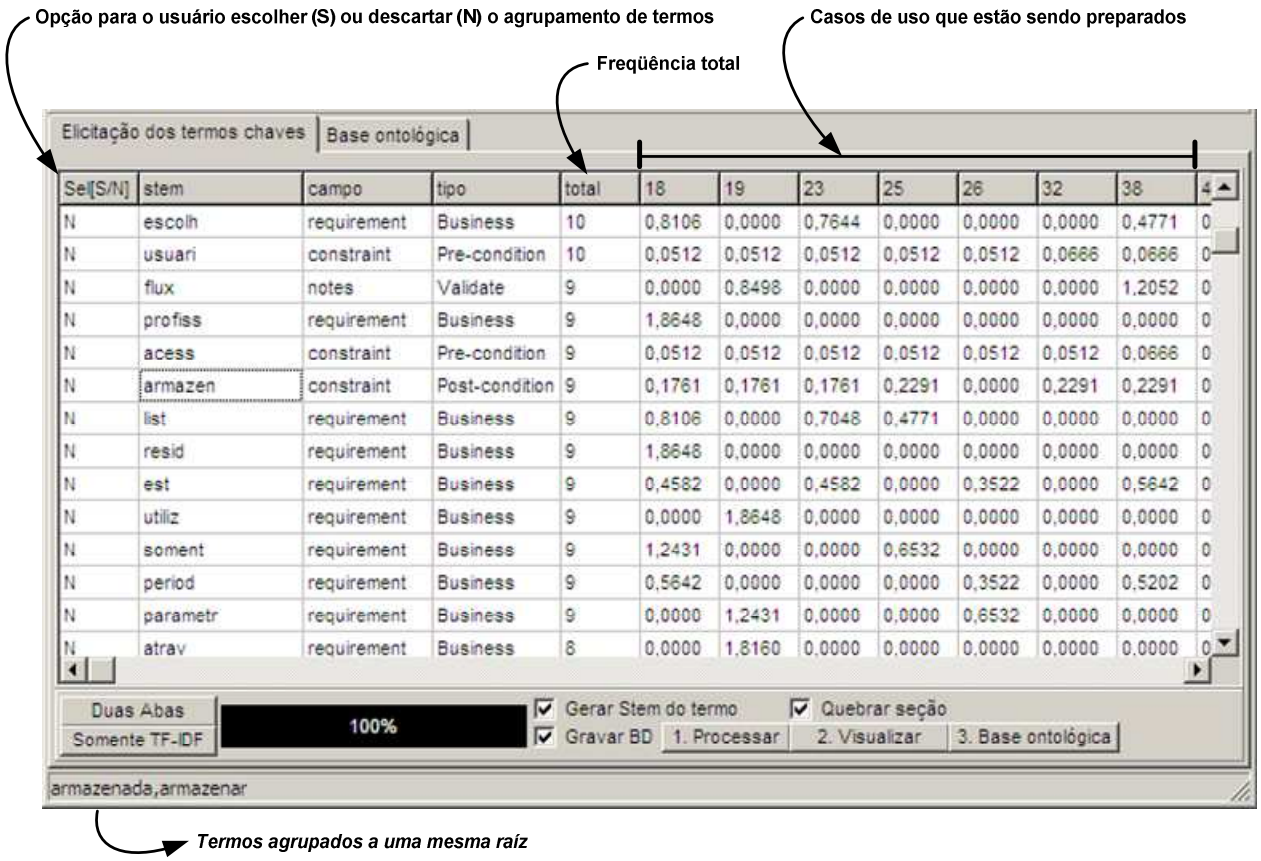


Figura 12 – Interface de extração e elicitação de termos dos casos de uso

O *plugin* de recuperação utiliza os mesmo módulos desenvolvidos para o pré-processamento e para a extração de termos utilizados no *plugin* de preparação. A lista de termos resultante da extração de termos é contextualizada, conforme explicado na seção 4.2.2 e utilizada para recuperar documentos que compartilhem os mesmos termos. Após a recuperação, os casos de uso são apresentados em ordem de similaridade. A interface desenvolvida para a recuperação é mostrada na Figura 14.

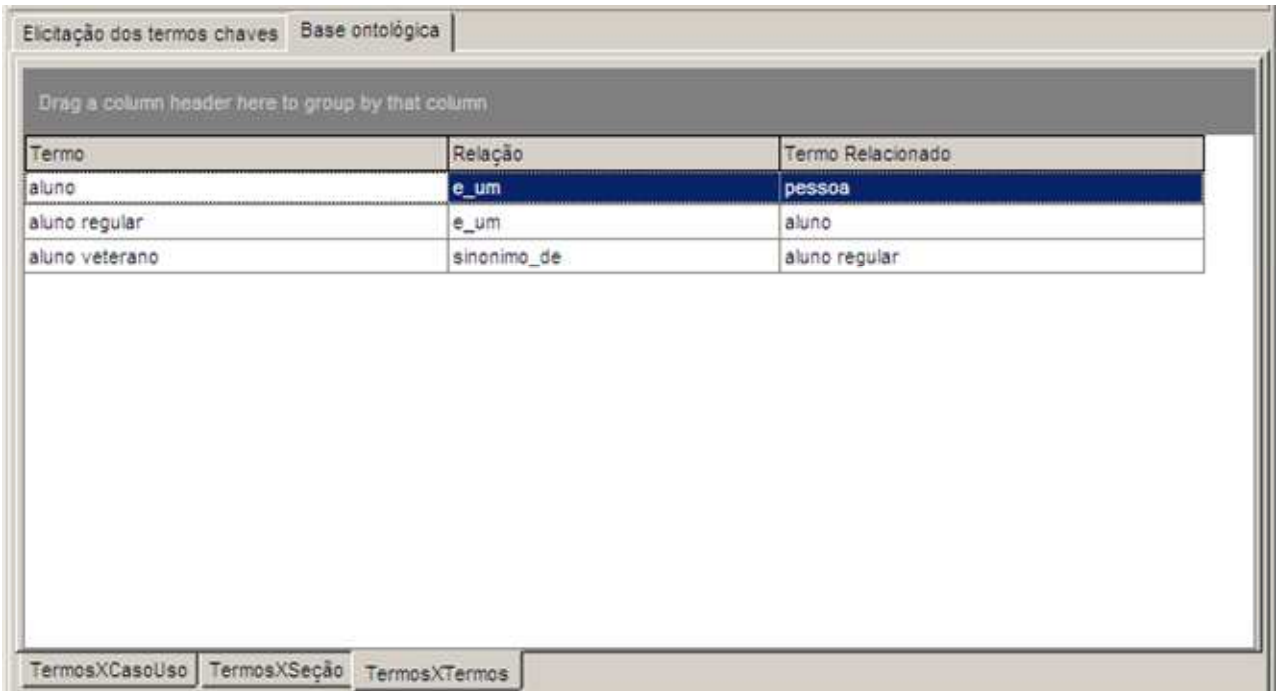


Figura 13 – interface para enriquecimento da lista de termos

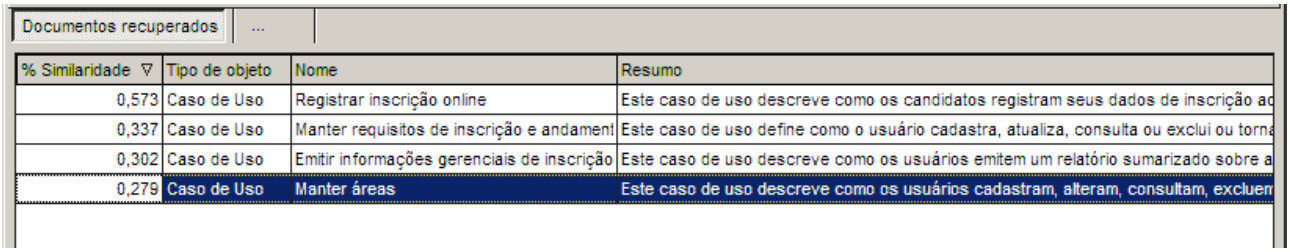


Figura 14 – interface de recuperação de documentos

5. EXPERIMENTO

Nesta seção apresentamos os experimentos de avaliação do método e da ferramenta desenvolvida.

5.1. Corpus de avaliação

Os experimentos foram realizados sobre um corpus formado por documentos de casos de uso que especificam um sistema para gestão de cursos e de professores de pós-graduação *stricto sensu* de uma universidade. O objetivo do sistema é o

> *"desenvolvimento de um novo sistema que contemple as funcionalidades necessárias para a gestão das operações envolvidas na Pós-Graduação Stricto Sensu da universidade, possibilitando o controle de calendário, processos, professores, turmas, disciplinas, alunos, bolsas. A disponibilidade e usabilidade do sistema devem ser umas das principais características, para que o sistema possa ser utilizado por diferentes Secretarias de Programa a qualquer momento, possibilitando uma visualização fácil e rápida de informações e relatórios."*

O corpus é formado por 81 casos de uso, sendo:

- 51,84 % dos documentos formados por casos de uso do tipo CRUD;
- 27,16 % dos documentos formados por casos de uso de relatórios; e
- 21 % dos documentos formados por casos de uso gerais;

5.2. Método de avaliação

Como não tivemos acesso à equipe que construiu os documentos de caso de uso utilizados neste trabalho, convidamos três analistas de sistemas com forte atuação no mercado de Cuiabá-MT, para utilizarem o método e o sistema proposto. Um dos analistas (analista de configuração) ficou responsável por criar os conjuntos de casos de uso que deveriam ser recuperados. Fez isso seguindo as recomendações da etapa de configuração do ambiente apresentadas na seção 4.1.1. Os outros dois analistas

(analistas de preparação) ficaram responsáveis pela criação da base ontológica e seguiram o método de preparação conforme descrito nas seções 4.1.2, 4.1.3 e 4.1.4.

O analista de configuração recebeu a ferramenta *CASE Enterprise Architect* 7.0 e o corpus de avaliação. Para fins de uma avaliação preliminar, o analista de configuração analisou os casos de uso do tipo CRUD, e destes separou quatro casos de uso que foram separados em dois conjuntos (Tabela 3). A estes anexamos mais alguns casos de uso escolhidos de forma aleatória antes de passá-los aos analistas de preparação.

Tabela 3 – Casos de uso separados na etapa de configuração de ambiente

Configuração do ambiente		
Conjunto A	**Conjunto B**	
Manter requisitos de inscrição e andamento	Manter áreas	C
Registrar inscrição *on line*	Manter programas	C
Manter áreas	Manter log de utilização/auditoria	A
Emitir informações gerenciais de inscrição	Registrar login não aluno	A
Manter programas	Instanciar fase e requisitos de andamento	A
	Validar requisitos inscrição	A
Legenda: C: resultado da etapa de configuração do ambiente A: casos de uso escolhidos de forma aleatória		

Cada analista de preparação recebeu a ferramenta *CASE Enterprise Architect* 7.0, o *plugin* de preparação e dois corpora, cada corpus se referindo a um dos conjuntos apresentados na Tabela 3. Os analistas tiveram uma semana para concluírem a fase de preparação. Esse tempo foi sugerido pelos próprios analistas para que eles tivessem contato com os casos de uso e entendessem o negócio. Após a conclusão da fase de preparação, os analistas nos entregaram duas bases contendo a ontologia e os índices para os documentos, essas bases foram utilizadas conforme nos foram entregues, não sendo aplicado nenhum tipo de revisão. Passamos então para a fase de recuperação.

5.3. Resultados

A fase de recuperação se inicia com a necessidade do designer em conhecer casos de uso previamente preparados e que sejam similares a um caso de uso em fase de especificação. Para que fosse possível avaliarmos a fase de recuperação, solicitamos ao

analista de configuração que nos sugerisse um caso de uso similar para cada um dos conjuntos criados na fase de preparação.

Utilizamos o corpus completo em conjunto com as ontologias criadas, para o processo de recuperação. Para efetivar a recuperação, escolhemos o caso de uso de entrada e executamos o *plugin* de recuperação. Os resultados são mostrados na Tabela 4 e Tabela 5. Nelas são demonstrados: o caso de uso utilizado como entrada para o *plugin* de recuperação; os casos de uso retornados na consulta, sendo destacados os casos de uso separados na fase de configuração de ambiente (Tabela 3); a similaridade entre o caso de uso retornado e o caso de uso de entrada; e as medidas de precisão, cobertura e média harmônica.

Tabela 4 – Resultado de recuperação do conjunto de teste A

Conjunto de teste A			
Caso de uso de entrada: Consultar situação de inscritos			
Casos de uso retornados	Similaridade (0..100)		
	Ontologia analista 1	Ontologia analista 2	Média
Registrar inscrição on line	0,57	0,47	0,52
Manter programas		0,49	0,49
Emitir informações gerenciais de inscrição		0,47	0,47
Manter requisitos de inscrição e andamento	0,34	0,48	0,41
Manter áreas	0,28		0,28
Avaliação			
Precisão	0,66	0,50	0,40
Cobertura	1	1	1
Média harmônica	0,80	0,67	0,57

Os casos de uso recuperados foram entregues para os analistas de preparação, sem uma ordem de similaridade definida. Para o analista de preparação 1, entregamos os casos de uso recuperados com a utilização da ontologia que foi preparada pelo analista de preparação 2. Foi adotado o mesmo critério para o analista de preparação 2. Solicitamos que eles analisassem o caso de uso utilizado como entrada e definissem uma ordem para

os casos de uso recuperados, sendo permitido formar conjuntos e definir uma mesma ordem ao conjunto. O resultado da ordenação é mostrado na Tabela 6 e Tabela 7.

Tabela 5 – Resultado de recuperação do conjunto de teste B

Conjunto de teste B			
Caso de uso de entrada: Manter crendenciamento[5] de professores			
Casos de uso retornados	Similaridade (0..100)		
	Ontologia analista 1	Ontologia analista 2	Média
Manter áreas	0,37	0,25	0,31
Manter programas	0,25	0,30	0,27
Instanciar fase e requisitos de andamento	0,14	0,18	0,16
Validar requisitos inscrição	0,15		0,15
Manter log de utilização/auditoria	0,11		0,11
Avaliação			
Precisão	0,40	0,67	0,40
Cobertura	1	1	1
Média harmônica	0,57	0,80	0,57

Tabela 6 – Resultado da ordenação manual do conjunto de teste A

Ordenação manual, conjunto de testes A			
Analista 1		**Analista 2**	
Ordem	**Casos de uso**	**Ordem**	**Casos de uso**
1º	Registrar inscrição on line	1º	Registrar inscrição on line
	Manter requisitos de inscrição e andamento		Manter requisitos de inscrição e andamento
	Emitir informações gerenciais de inscrição	2º	Manter áreas
2º	Manter programas		

[5] O nome do caso de uso está grafado conforme recebemos originalmente.

Tabela 7 – Resultado da ordenação manual do conjunto de teste A

Ordenação manual, conjunto de testes B			
Analista 1		**Analista 2**	
Ordem	**Casos de uso**	**Ordem**	**Casos de uso**
1º	Manter áreas	1º	Manter áreas
	Manter programas		Manter programas
2º	Instanciar fase e requisitos de andamento	2º	Instanciar fase e requisitos de andamento Validar requisitos inscrição Manter log de utilização/auditoria

5.4. Considerações

Apesar de preliminares, os resultados apresentados mostram que o método descrito neste trabalho é eficaz, visto que ele apresentou cobertura de 100% em ambos os testes. Quanto a medida de precisão, que apresentou resultado inferior a 50%, o resultado foi compensado pelo algoritmo de *ranking* que ordenou os documentos de forma similar a classificação manual feita pelos usuários. Existem ainda alguns pontos que devem ser considerados:

- Etapa de elicitação de palavras-chave: a escolha de termos é feita de forma subjetiva, e varia segundo o entendimento que o analista que a está executando tem sobre os documentos que estão sendo preparados. Nesta fase, até o papel do analista influência. Percebe-se (Tabela 4) que quando o analista tem um papel mais voltado para o negócio o comportamento do sistema tende a ser mais generalista – caso do Analista 2. O oposto ocorre quando o papel do analista é mais técnico – caso do Analista 1.
- Etapa de enriquecimento semântico da lista de termos: esta etapa captura o conhecimento que os analistas detêm sobre o domínio. Esse conhecimento tende a variar de acordo com o analista. Por esse motivo, seria mais interessante que a ontologia fosse criada a partir do conhecimento de um grupo de analistas.
- Etapa de recuperação: a recuperação de documento é realizada a partir dos índices e do conhecimento descrito na ontologia. Mesmo com o viés causado pela utilização de um único analista para a criação dos índices e da ontologia, os documentos recuperados satisfizeram ambos os analistas (Tabela 6 e Tabela 7). Isso demonstra que cada analista detém, minimamente, o conhecimento consensual do domínio.

6. CONSIDERAÇÕES FINAIS

Este trabalho objetivou o desenvolvimento de um método para recuperação semântica de documentos de casos de uso. Para tal, fundamentamos o trabalho com técnicas para construção e avaliação de sistemas de recuperação de informações e discutimos como a utilização de uma ontologia poderia melhorar os resultados do sistema. Esses recursos foram então organizados em etapas, de forma a guiar os envolvidos em um projeto de software na construção de uma ontologia para representação do domínio e na criação dos índices que seriam utilizados na fase de recuperação.

Um protótipo de software foi desenvolvimento com o objetivo principal de validar o método proposto. Com o auxílio do protótipo foi realizado experimentos de recuperação sobre o corpus no domínio de um departamento de pós-graduação de uma Universidade.

Os resultados apresentados foram considerados relevantes a essa pesquisa. No entanto percebemos que o desempenho da fase de recuperação semântica de informações é dependente da fase de preparação da ontologia, que é feita por um especialista do domínio. Como nem sempre se tem um especialista do domínio disponível na fase de implementação, ainda no contexto desta pesquisa, é desejável verificar qual o impacto na eficácia do sistema se forem utilizados outros analistas na preparação da base ontológica.

6.1. Trabalhos futuros

No desenvolvimento deste trabalho, identificamos pontos em abertos que deixamos como sugestão para trabalhos futuros:

- Pesquisar mecanismos que ajudem o analista de configuração a selecionar os documentos na fase de configuração do ambiente;
- Pesquisar mecanismo para a utilização de sintagmas na fase de extração de palavras-chave do domínio;
- Investigar se a utilização do algoritmo *C-value/NC-value* na preparação dos índices melhora o resultado do algoritmo de recuperação apresentado neste trabalho;
- Implementar algoritmos para construção semi-automática da ontologia a partir dos documentos de caso de uso do domínio.

REFERÊNCIAS BIBLIOGRÁFICAS

[Ahn+07] J. Ahn, P. Brusilovsky, J. Grady, D. He, S. Y. Syn. “Open User Profiles for Adaptive News Systems: Help or Harm?”. In: 16th International Conference on World Wide Web, 2007, pp. 11–20.

[Bai07] J. Bai, J. Y. Nie, G. Cao, H. Bouchard. “Using query contexts in information retrieval”. In: 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2007, pp. 15–22.

[Bas07] H. Bast, A. Chitea, F. Suchanek, I. Weber. “ESTER: Efficient Search on Text, Entities, and Relations”. In: 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2007, pp. 671–678.

[Bho07] J. Bhogal, A. Macfarlane, P. Smith. “A Review of Ontology Based Query Expansion”. Information Processing & Management, vol. 43-4, Jul 2007, pp. 866–886.

[Bre05] K. Breitman. “WEB SEMÂNTICA: A Internet do Futuro”, LTC, 2005, 212p.

[Car+06] J. C. Carroll, J. M. Prager, K. Czuba, D. A. Ferrucci, P. A. Duboué. “Semantic Search Via XML Fragments: A High-Precision Approach to IR”. In: 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2006, pp. 445–452.

[Chi05] P. A. Chirita, W. Nejdl, R. Paiu, C. Kohlschütter. “Using ODP Metadata to Personalize Search”. In: 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005, pp. 178–185.

[Coc00] A. Cockburn. “Writing Effective Use Cases”. Addison-Wesley Professional, 2000, 304p.

[Cro00] W. B. Croft, J. Xu. “Improving the Effectiveness of Information Retrieval with Local Context Analysis”. ACM Transactions on Information Systems (TOIS), vol. 18-1, Jan 2000, pp. 79–112.

[Dom01] S. Dominich. “Mathematical Foundations of Information Retrieval”. Springer, 2001, 284p.

[Gom04] A. Gomez-Perez, O. Corcho, M. Fernandez-Lopez. “Ontological Engineering: with Examples from the Areas of Knowledge Management, E-Commerce and the Semantic Web”. Springer, 2004, 415p.

[Gon07] M. Gonzalez, L. C. Langie, V. L. S. de Lima. “Avaliação Conjunta: Um Novo Paradigma no Processamento Computacional da Língua Portuguesa”. IST Press, 2007, 304p.

[Hu+08] J. Hu, L. Fang, Y. Cao, H. J. Zeng, H. Li, Q. Yang, Z. Chen. “Enhancing Text Clustering by Leveraging Wikipedia Semantics”. In: 31th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2008, pp. 179–186.

[Jur00] D. Jurafsky, J. H. Martin. “Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition”. Prentice Hall, 2000, 988p.

[Lar07] C. Larman. “Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientados a Objeto e ao Desenvolvimento Iterativo”, Bookman, 2007, 696p.

[Li09] Z. Li, M. Yang, K. Ramani. “A Methodology for Engineering Ontology Acquisition and Validation”. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, vol. 23-1, Fev 2009, pp. 37–51.

[Liu02] F. Liu, C. Yu, W. Meng. “Personalized Web Search by Mapping User Queries to Categories”. In: 11th International Conference on Information and Knowledge Management, 2002, pp. 558–565.

[Noy01] N. F. Noy, D. L. McGuinness. “Ontology Development 101: A Guide to Creating Your First Ontology”, Technical Reports SMI, Knowledge Systems Laboratory Stanford University, 2001, 25p.

[Ore06] V. M. Orengo, L. S. Buriol, A. R. Coelho. “Evaluation of Multilingual and Multi-modal Information Retrieval”. In: 7th Workshop of the Cross-Language Evaluation Forum, 2006, pp. 91–98.

[Par03] J. Paralic, J. I. Kostial. “Ontology-based Information Retrieval”. In: 14th International Conference on Information and Intelligent Systems, 2003, pp. 23–28.

[San07] A. R. Santos. “Metodologia Científica. A Construção do Conhecimento”. DP&A Editora, 2007, 190p.

[Sil08] C. G. da Silva Jr. “Sistemas de Recuperação de Informações Baseados em Processamento da Língua Natural: Fundamentos e Aplicações”, Trabalho Individual I, Programa de Pós-Graduação em Ciências da Computação, PUCRS, 2008, 58p.

[Som03] I. Sommerville. “Software Engineering”. Addison-Wesley, 2003, 284p.

[Suc07] F. M. Suchanek, G. Kasneci, G. Weikum. “Yago: A Core of Semantic Knowledge”. In: 16th International Conference on World Wide Web, 2007, pp. 697–706.

[Wil+06] R. Willrich, R. de Moura Speroni, C. V. Lima, A. L. de Oliveira Diaz, S. M. Penedo. “Adaptive Information Retrieval System Applied to Digital Libraries”. In: 12th Brazilian Symposium on Multimedia and the Web, 2006, pp. 165–173.

[Yat99] R. A. B. Yates, B. A. R. Neto. “Modern Information Retrieval”. Addison-Wesley, 1999, 513p.