Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Dissertação de Mestrado

Criação de Vetores Temáticos de Domínios para a Desambiguação Polissêmica de Termos

Magna Celi Tavares Bispo

Campina Grande, Paraíba, Brasil.
Novembro - 2012

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Dissertação de Mestrado

Criação de Vetores Temáticos de Domínios para a Desambiguação Polissêmica de Termos

Magna Celi Tavares Bispo

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande – Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Recuperação de Informação

ULRICH SCHIEL
(Orientador)

CARLOS EDUARDO SANTOS PIRES
(Coorientador)

Campina Grande, Paraíba, Brasil.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

B622c Bispo, Magna Celi Tavares.
Criação de vetores temáticos de domínio para desambiguação polissêmica de termos/Magna Celi Tavares Bispo. – Campina Grande, 2012.
97f.: il.col.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.
Orientador: Prof. Dr. Ulrich Schiel
Referências.

1. Indexação de Termos. 2. Wikipédia. 3. Linguagem Natural. I. Título

CDU 004:025.4(043)

"CRIAÇÃO DE VETORES TEMÁTICOS DE DOMÍNIOS PARA A DESAMBIGUAÇÃO POLISSÊMICA DE TERMOS"

MAGNA CELI TAVARES BISPO

DISSERTAÇÃO APROVADA EM 30/11/2012

ULRICH SCHIEL, Dr., UFCG
Orientador(a)

CARLOS EDUARDO SANTOS PIRES, Dr., UFCG
Orientador(a)

LEANDRO BALBY MARINHO, Dr., UFCG
Examinador(a)

EDBERTO FERNEDA, Dr., UNESP
Examinador(a)

CAMPINA GRANDE - PB

Resumo

A ambiguidade de termos é um dos fatores que dificulta o processo de indexação de documentos e recuperação de informação desejada por um usuário. O presente trabalho se baseia na hipótese de que parte deste problema pode ser minimizado sabendo-se de antemão o domínio do documento que contém termos ambíguos. Para determinar este domínio foram construídos vocabulários temáticos por meio da extração de termos de documentos de domínios de conhecimento pré-determinados, com o uso de regras sintáticas. A Wikipédia foi usada como base de consulta, por ser uma enciclopédia digital contendo as categorias definidas semelhantes à Classificação Decimal Universal (CDU), e cada categoria com uma vasta quantidade de documentos específicos, sendo essa característica fundamental para formação de um vocabulário específico do domínio de um conhecimento. A escolha das categorias foi baseada na CDU, composta de 10 domínios e seus respectivos subdomínios. Os vocabulários obtidos, denominados de Vetores Temáticos de Domínio (VTD), serviram de base para a classificação de novos documentos. Para validação dos VTD's, foram realizados três tipos de experimentos diferentes, o primeiro foi classificar novos documentos utilizando o método vetorial, tendo o VTD como base de consulta. O segundo experimento foi uma classificação utilizando outro classificador, o *Intellexer Categorizer*, e o terceiro experimento, criou-se um vetor de termos através do *Weka,* o qual foi submetido a servir de base de consulta para classificar novos documentos, utilizando o modelo vetorial. Os resultados foram satisfatórios, pois mostrou que o VTD obteve uma melhor classificação em relação aos outros métodos, dos 14 novos documentos, classificou 10 corretamente e 4 errados, apresentando uma acurácia de 80%, contra a acurácia de 57% do *Intellexer Categorizer* e de 50% da classificação utilizando o vetor de termos criado pelo *Weka.*

Palavras-chave: extração da informação, indexação de documentos, processamento da linguagem natural, *Postagger*, vocabulários temáticos

Abstract

Terms ambiguity is one of the factors that hinders the document indexation and information retrieval processes desired by a user. This work is based on the hypothesis that part of this problem can be minimized by knowing beforehand the field of the document that contains ambiguous terms. To determine this domain, typical vocabularies were created through the extraction of terms from documents of predetermined knowledge domains, with the use of syntactical rules. Wikipedia was used as a consultation base because it is a digital encyclopedia that contains the categories defined similar to the Universal Decimal Classification (UDC), each category containing a vast amount of specific documents, being this feature essential for the formation of a domain-specific vocabulary. The choice of the categories was based on the UDC, composed of 10 domains and their respective subdomains. The vocabularies obtained, denominated as Thematic Domain Vectors (TDV), served as the basis for the classification of new documents. For the validation of the TDV's, three different types of experiments were performed: the first was to classify new documents using the vectorial method, with the TDV as a basis of consultation. The second experiment was a classification using another classifier, the Intellexer Categorizer. For the third experiment was created a vector of terms through Weka, which was submitted to serve as a a consultation base to classify new documents using the vectorial model. The results were satisfactory, because they showed that the TDV obtained a better classification relative to other methods. Of the 14 new documents, properly it rated 10 and 4 incorrectly, with an accuracy of 80%, against 57% accuracy of the Intellexer Categorizer program and 50% of the classification using the Weka created vector of terms.

Keywords: information extraction, document indexing, natural language processing, Postagger, thematic vocabularies.

Agradecimentos

Agradeço antes de tudo a Deus, pois sem acreditar nos seus desígnios eu não estaria aqui. Dedico esse trabalho em agradecimento aos meus pais, que não estão mais nesse plano, por terem priorizado sempre a educação e me ter dado a base para conseguir tudo que consegui.

Agradeço aos meus tios que são meus segundos pais, e sempre tão dedicados e pacientes me dando apoio em todos os aspectos.

Agradeço aos meus filhos por sempre estarem do meu lado em todas as empreitadas da vida e Hélio, por sempre ter me dado apoio e acreditado em mim, encorajando-me no retorno ao estudo.

Agradeço aos meus orientadores por terem tido paciência comigo e dispensado atenção durante todo o período do mestrado.

Agradeço ao professor Jacques Sauvé que no momento crítico do meu retorno soube agir como um verdadeiro mestre.

Agradeço aos professores com quem tive a oportunidade de estudar, pela contribuição na minha formação.

Agradeço aos meus colegas de mestrado, pois junto aos jovens eu rejuvenesço, captando a energia positiva deles.

Agradeço a minha amiga Pryscilla Dóra, pois ela foi uma verdadeira amiga desde o início do mestrado, na hora da dificuldade ela estava lá pronta para ser minha parceira independente de rótulos.

Agradeço aos meus amigos do LSI, todos são excelentes, e a minha amiga Isabel Nunes, que conheci depois, mas uma excelente amiga e companheira.

Agradeço a Dona Inês pelo apoio fornecendo uma boa comida e aos meninos da limpeza que sempre que eu solicitava para limpar o laboratório eles estavam ali respondendo a um chamado e a Socorro que toma conta da minha casa para mim.

Agradeço a alguns amigos e a um em especial 'Anjinho' que fazem parte da minha vida e me dão apoio para a concretização desse trabalho.

Enfim, agradeço a todos que diretamente e indiretamente contribuíram para minha permanência no mestrado.

Conteúdo

Lista de Símbolos

AF – Anenia Falciforme

CDD – Classificação Decimal de *Dewey*

CDU – Classificação Decimal Universal

CEEI – Centro de Engenharia Elétrica e Informática

DSC – Departamento de Sistemas e Computação

EI – Extração de Informação

PLN – Processamento de Linguagem Natural

POS – Tagger–*Part-of-speech Tagger*

ODT– *Open Document Text*

RI – Recuperação de Informação

RISO-T – Recuperação da Informação Semântica de Objetos Textuais

SINBAD – Grupo de pesquisa em Sistemas de Informação e Bancos de Dados

SRI – Sistema de Recuperação de Informação

SVSim – Space Vetor Similarity

UFCG – Universidade Federal de Campina Grande

VTD – Vetor Temático de Domínio

Vetor*Weka* – Vetor de termos criado pelo *Weka*

VSM – Vetor Space Model

RISOExtrator – Extrator de Termos do RISO

RISO-VTD - Sistema de Criação dos Vetores Temáticos de Domínio do RISO

Lista de Figuras

Lista de Equações

Lista de Tabelas

Capítulo 1

Introdução

Nos últimos anos a quantidade de informação digital gerada é imensa, ocasionada pela publicação de uma grande quantidade de documentos. A utilização adequada destes documentos depende diretamente da forma como eles são organizados. Por exemplo, imagine uma grande biblioteca tradicional onde estão chegando caixotes lotados de livros. Os bibliotecários devem catalogar esses livros, classificá-los e organizá-los em prateleiras, para facilitar a obtenção do livro no momento em que for solicitado pelo usuário. Para uma correta classificação dos documentos, são necessárias tarefas de leitura, interpretação do conteúdo, contextualização e catalogação. O princípio para organizar os livros na biblioteca se baseia em uma divisão sistemática do conhecimento humano em grandes áreas, refinadas em subáreas[1]. O usuário, para encontrar os livros que procura, deve conhecer esta organização e pesquisar os títulos existentes na subárea de seu interesse.

Diferentemente, documentos digitais não precisam ser organizados em prateleiras. Neste caso, a tarefa de 'encontrar um livro' é auxiliada por um Sistema de Recuperação da Informação (SRI). Esse sistema trata de encontrar documentos (ou partes deles), a partir de consultas fornecidas como entrada (Loh, Wives e Frawer, 1997).

A tarefa de recuperação de informação é avaliada de acordo com a capacidade que o sistema tem de recuperar o maior número de informações relevantes e excluir ao máximo os itens irrelevantes. As ferramentas de Recuperação de Informação geralmente trabalham associadas a técnicas de indexação capazes de mapear e acessar rapidamente documentos em uma base de texto (Baeza-Yates, 1996).

Enquanto em uma biblioteca convencional o usuário realiza a busca por um livro procurando pela subárea de conhecimento de seu interesse, em uma biblioteca digital o Sistema de Recuperação de Informação é baseado em palavras-chave e

[1]As classificações mais usadas são a CDU - Classificação Decimal Universal e a CDD - Classificação Decimal de Dewey.

retorna todos os documentos contendo as palavras-chave indicadas, sem considerar as áreas de interesse do usuário. Como os documentos não são classificados previamente para atender uma procura mais específica, mesmo que o usuário indique sua área de interesse, o sistema retorna informações que não atendem a busca.

Em muitos casos a indexação de documentos não é suficiente para atender a necessidade expressa pelo usuário. Um dos motivos desta situação é a ambiguidade do sentido das palavras que ocorrem tanto na consulta formulada pelo usuário como nos documentos.

A ambiguidade pode ser *polissêmica* ou *estrutural* (Monnerat, 2003). No primeiro caso, a ambiguidade deve-se à possibilidade de os vocábulos apresentarem mais de um significado; no segundo, ela se prende a problemas de construção, ou seja, à forma como uma determinada frase foi construída. A ambiguidade derivada da polissemia do vocábulo pode ser evitada pelo esclarecimento maior do contexto, ou pela substituição do vocábulo polissêmico por outro de sentido único; já no caso da ambiguidade estrutural, as causas são muitas e as possibilidades de eliminá-la variam conforme o problema que a origina (Monnerat, 2003).

Um exemplo de ambiguidade estrutural é *Tomei a faca de José*. Essa frase gera uma dúvida se tomei a faca que estava com José ou se tomei a faca que pertencia a José. Como exemplo de uma ambiguidade léxica ou polissêmica, temos *O rapaz está perto do banco*. O termo "banco" pode se referir a um banco de praça ou uma agência bancária. O correto seria *O rapaz está perto do banco da praça* ou *O rapaz está perto do Banco do Brasil*, ou ainda *O rapaz está perto de uma agência bancária*. Todas estas frases especificam o tipo de banco. A ambiguidade que iremos focar neste trabalho é a ambiguidade estrutural.

Uma possibilidade de reduzir esta ambiguidade é classificar os documentos por domínios ou pelo assunto tratado nos mesmos. Ao contrário de uma biblioteca convencional, em que esta classificação é realizada pelo bibliotecário, em uma biblioteca digital esta classificação tem que ser automática, devido ao grande volume de documentos digitais a serem processados.

Como hipótese deste trabalho, supõe-se que um domínio de conhecimento pode ser caracterizado pelos termos ou pela terminologia utilizada nos documentos específicos do domínio.

Um termo pode ter vários significados, sendo que, muitas vezes, cada significado está inserido em um domínio diferente. Logo, caso se saiba o domínio do documento, a ambiguidade é minimizada.

Alguns documentos tratam de um assunto de um domínio pré-definido que muitas vezes não está colocado de uma maneira clara. Contudo, pode-se supor que a terminologia usada em diversos documentos de uma mesma área é semelhante. Desta forma é possível obter esta terminologia típica analisando antecipadamente um número significativo de documentos de um domínio. Para tal, pode ser aplicado um processo de aprendizado no sentido de determinar esta terminologia a partir de uma amostra representativa de documentos do domínio.

Um processo de treinamento ou aprendizado pode ser realizado de duas maneiras: utilizando algoritmos de aprendizado supervisionado ou não supervisionado (Resende, 2005). Um algoritmo de aprendizagem supervisionada possui uma saída pré-determinada e é baseado na correção de diferenças entre as saídas geradas e as pré-determinadas. Com um algoritmo de aprendizado não supervisionado a saída não é conhecida, e a classificação é feita automaticamente, baseando-se na análise das correlações entre as entradas de forma a agrupá-las, codificá-las ou categorizá-las (Resende, 2005).

A classificação é um método supervisionado, que consiste em classificar um dado conjunto de objetos que tenham um alto grau de similaridade entre si em coleções de acordo com critérios estabelecidos. Objetos que pertençam a grupos distintos devem possuir uma menor similaridade entre si.

Em nosso caso deverá ser usado um aprendizado supervisionado, pois a terminologia será criada a partir de documentos de domínios previamente conhecidos. Essa terminologia será armazenada em vetores, os quais serão denominados Vetores Temáticos de Domínio (VTD) os quais serão construídos através de um sistema denominado RISO-VTD.

Com um conjunto de vetores temáticos, sendo um para cada domínio de conhecimento, é possível determinar o contexto de um novo documento. Para isso, deve-se comparar o vetor do temático do novo documento com o vetor temático de cada domínio e, em seguida, escolher o domínio que está mais próximo. Esta

proximidade será obtida pelo cálculo do cosseno entre os vetores, baseado no modelo espaço vetorial de Salton (1988).

Na biblioteca convencional a tarefa de classificação exerce o importante papel na organização dos documentos, já na biblioteca digital, ajuda na desambiguação de termos e, consequentemente melhorar a seleção de documentos retomados em uma pesquisa.

De acordo com o problema de ambiguidade de termos apresentado, nossa pesquisa visa melhorar a recuperação de documentos em uma biblioteca digital. Utilizando uma técnica de aprendizado de máquina supervisionado a partir de um conjunto de documentos de cada domínio, é detectado o vocabulário típico daquele domínio. Esta linguagem pode ser usada posteriormente como padrão de referência para classificação de novos documentos.

1.1 Objetivo

Diante da explanação apresentada, este trabalho tem como objetivo propor um método de criação de vocabulários de domínio, a partir de documentos com o conhecimento prévio do domínio, para posteriormente poder compará-los com o vocabulário de um documento qualquer e, assim, classificar o documento automaticamente.

Para alcançar este objetivo geral, devem ser atingidos os seguintes objetivos específicos:

- Criação de um método para extração dos termos de documentos através de regras sintáticas criadas de acordo com a posição dos termos no texto;
- Criação de vetores dos termos encontrados nos documentos, levando em consideração sua frequência e importância, para a formação de terminologias de diversos domínios do conhecimento;
- Implementação do modelo vetorial para verificação da qualidade dos vetores criados por meio de testes de classificação de novos documentos.

1.1 Relevância

As pessoas têm acesso livre a uma enorme quantidade de informações, tornando as bibliotecas digitais um importante meio de interação. Contudo, ainda existem muitos problemas na qualidade da recuperação de informação como, por exemplo, a interpretação do conteúdo das informações encontradas nos documentos, e a indexação desses documentos de acordo com sua relevância e a ausência de classificação de documentos digitais.

A qualidade da recuperação é fortemente afetada pela ambiguidade dos termos contidos em uma consulta, o que reduz a precisão do resultado da recuperação. Espera-se que, com a classificação dos documentos baseada em vetores temáticos, consiga-se reduzir a ambiguidade de termos e, com isso, melhorar a qualidade no processo de recuperação.

1.2 Estrutura da Dissertação

O restante deste trabalho está organizado da seguinte maneira:

No capítulo 2 é apresentada uma fundamentação teórica sobre os principais assuntos tratados no trabalho.

No capítulo 3 são apresentados os trabalhos relacionados.

No capítulo 4 é apresentado o sistema de criação dos vetores temáticos utilizando a técnica de Processamento de Linguagem Natural.

No capítulo 5 são mostrados os experimentos e discussão dos resultados.

No capítulo 6 é apresentada a validação do trabalho utilizando medidas de desempenho para validar o experimento.

No capítulo 7 são apresentadas as considerações finais deste trabalho e as sugestões de trabalhos futuros.

Capítulo 2

Fundamentação Teórica

A fim de atingir os objetivos proposto neste trabalho de pesquisa, realizou-se um estudo da área na qual o trabalho está inserido. Desta forma, serão apresentados uma introdução à Recuperação da Informação, à Extração da Informação e alguns aspectos de Processamento de Linguagem Natural.

2.1 Recuperação de Informação

A Recuperação de informação é uma subárea da Ciência da Computação que estuda o armazenamento e a recuperação automática de documentos, na maioria textuais (Cardoso, 2004).

O objetivo do processo de recuperação de informação é identificar, num conjunto de documentos de um sistema, quais os documentos que atendem à necessidade de informação do usuário. Os documentos são o foco principal desse processo, cujo conteúdo é não estruturado, ou seja, é composto por estruturas de informações irregulares e difíceis de serem abstraídas em modelos de dados. Um conjunto de documentos que apresentam essas características denomina-se *corpus* ou coleção.

Um *corpus* é uma coletânea de documentos de textos selecionados segundo critérios linguísticos, codificados de modo padronizado e homogêneo (Bidermann, 2001).

A análise do *corpus* deve ser realizada pelo Sistema de Recuperação de Informação (SRI) para o fornecimento das informações de maior relevância. A relevância da informação está diretamente relacionada com o usuário, com a sua necessidade de informação e com o momento em que isso ocorre (Wives, 2002).

Os documentos além de serem analisados com a finalidade de agregar assuntos que não estão explícitos claramente, podem ser facilmente ordenados por um indexador humano. A indexação é efetuada tendo em vista a sua recuperação, com a

preocupação de tornar seu conteúdo visível para os usuários de um sistema de informação (Ferneda, 2012, p.16).

Os métodos automáticos de indexação no geral utilizam filtros para eliminação de palavras de pouca importância, como as *stopwords* e reduz as palavras aos radicais (*stemming*). Esse modo de indexação seleciona conteúdo significante (termos ou frases) dos documentos, sem levar em consideração os vários significados que os mesmos podem assumir de acordo com o domínio. Embora esta maneira de indexação seja muito utilizada, as falhas e limitações são evidentes por não se preocupar com a semântica da linguagem (Ferneda, 2012, p.16).

A eficiência de um SRI depende diretamente do modelo de recuperação de informação que ele utiliza, influenciando diretamente em seu modo de operação.

Os modelos de SRI foram criados por volta dos anos 60 e 70, aperfeiçoados na década de 80. Na maioria dos sistemas atuais e nos mecanismos de busca da web, outros modelos como Algoritmos Genéticos, Redes Neurais, Lógica Fuzzy entre outros, os quais representam novas abordagens baseadas na Inteligência Artificial, possuem um futuro promissor a ser explorado (Ferneda, 2012. p.20).

Os modelos clássicos, usados no processo de recuperação de informação, tais como vetorial, boleano e probabilístico, representam estratégias de busca de documentos relevantes para uma consulta (Cardoso, 2002).

Os três modelos consideram que cada documento é representado por um conjunto de palavras-chave, denominadas termos de indexação. A cada termo de indexação t_i, em um documento d_j, é associado a um peso $w_{ij} > 0$, que quantifica a correlação entre os termos e o documento (Cardoso, 2002).

No presente estudo é utilizada uma abordagem para classificação de documentos, baseada no modelo vetorial.

2.1.1 Modelo Vetorial

Esse modelo, também chamado de modelo espaço-vetorial, representa documentos e consultas como vetores de termos, onde os termos são ocorrências únicas no documento (Cardoso, 2002). Como resultado de uma consulta, os documentos são classificados de acordo com a similaridade entre o vetor da consulta e

o vetor do documento. O resultado é um conjunto de documentos ordenados pelo grau de similaridade (Ferneda, 2012, p.31).

No modelo vetorial, um documento é representado por um vetor no qual para cada elemento é representado o peso do respectivo termo de indexação em relação ao documento. Cada vetor descreve a posição do documento em um espaço multidimensional, em que cada termo de indexação representa uma dimensão ou eixo. Cada elemento do vetor, é normalizado de forma a assumir valores entre 0 e 1. Os pesos mais próximos de 1 indicam termos com maior importância para o documento (Ferneda, 2012, p.31).

Salton (1983) afirma que a importância de um termo é diretamente proporcional à sua frequência no documento e inversamente à sua frequência na coleção de documentos. Ele sugere uma fórmula para obter o peso de um termo k em um documento i, baseados na frequência do termo no documento e na frequência do termo na coleção completa.

O cálculo da frequência de um termo (*term frequency-tf*) é a quantidade de vezes que o termo **t** ocorre no texto de um documento d divido pela frequência do termo mais frequente no documento:

$$tf(t,d) = \frac{freq_{t,d}}{Freq_{max,d}} \qquad \textbf{Equação 3-1}$$

Para considerar a importância de um termo na coleção é usado o cálculo do inverso da frequência do termo (*idf*) pela Equação 2:

$$idf(t) = log_{10}\left(\frac{N}{n_t}\right) \qquad \textbf{Equação 3-2}$$

onde:

N é o número total de documentos na coleção;
n_t é o número total de documentos contendo o termo;

Esses valores tf e idf são usados para calcular o peso de um termo em um documento, utilizando a seguinte equação:

$$w_{t,d} = tf(t,d) \times idf(t) \qquad \text{Equação 3-3}$$

Em um espaço vetorial contendo N dimensões, a similaridade entre o vetor de um documento $\mathbf{d_j}$ e o vetor de uma expressão de consulta $\mathbf{q}$ pode ser calculada utilizando o cosseno entre os dois vetores:

$$sim(d,q) = \frac{\sum_{i=1}^{N}(w_{i,d} \times w_{i,q})}{\sqrt{\sum_{i=1}^{N} w_{i,d}^2} \times \sqrt{\sum_{i=1}^{N} w_{i,q}^2}} \qquad \text{Equação 3-4}$$

onde $w_{i,d}$ é o peso do i-ésimo termo do documento $\mathbf{d_j}$ e $w_{i,q}$ é o peso do i-ésimo termo da expressão da consulta $\mathbf{q}$, determinando a proximidade da ocorrência, como mostra a Equação 3-4.

O desenvolvimento de sistemas de recuperação de informação que tem a possibilidade de processar os documentos necessita de técnicas computacionais complexas. Essas técnicas computacionais compõem o Processamento de Linguagem Natural, que será tratado a seguir.

2.2 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é um ramo da Inteligência Artificial que estuda os problemas surgidos da manipulação da língua natural. É uma área multidisciplinar que envolve diversas áreas do conhecimento, tais como: Ciência da Computação, Linguística e Ciências Cognitivas. Para tratar esse problema há ferramentas linguísticas próprias ou ferramentas de PLN que recebem um texto ou trecho de texto em línguagem natural e agregam ou extraem algum tipo de informação ou traço linguístico.

Ferramentas distintas de PLN podem ser encadeadas de forma sequencial, com o objetivo de obter um processamento mais complexo. Devido a essa característica, se acontecerem erros no início do processo, esses erros são propagados para as próximas ferramentas, podendo resultar em erros muito grandes ao fim do processo.

2.2.1 Tokenizador

Um tokenizador tem como função básica indentificar onde acaba uma sentença e inicia outra. Essa tarefa não é trivial, sendo aplicada em um pré-processamento.

O tokenizador deve determinar o que é um *token*. *Token* é o menor bloco estruturado de um texto, por exemplo, uma palavra. Um token delimitador é representado por caracteres tais como, ponto (.), ponto e vírgula (;), vírgula (,) entre outros. Um bom tokenizador deve reconhecer outros delimitadores e lidar com abreviações, números que dependem da língua do texto de entrada. Se uma sentença for quebrada erroneamente, o erro será propagado para as próximas etapas.

2.2.2 *Tagger* ou Etiquetador

Chamado também de *part-of-speech tagger* ou *tagger* simplesmente, tem como função identificar informação morfológica e sintática (*Part of Speech ou POS*) a cada *token* de um texto. A informação adicionada é denominada de *tag*, etiqueta ou marcação. Temos como exemplos de tags: substantivo, adjetivo, verbo, conjunção, entre outros.

Na grande maioria, o aprendizado dos etiquetadores é baseado em *corpus* e os *tokens* são marcados automaticamente com uma etiqueta. Os corpora tornaram-se um recurso importante para pesquisas em linguística computacional. Para a língua inglesa a criação do Penn Treebank[2] (Marcus et al.,1993), motivou o desenvolvimento de uma grande variedade de corpora com rica e variada anotação de fenômenos, por exemplo: a anotação de papéis semânticos como os do PropBank[3], do NomBank[4], anotação discursiva como a do Penn Discourse Treebank[5], entre outros.

2.2.3 *Chunker*

Um *chunker* segmenta uma sentença em unidades estruturais chamadas de *chunks*. Um *chunker* é um conjunto de *tokens* consecutivos, agrupados por função sintática, como sintagmas:

- nominais (frases substantivas ou *NX*),
- verbais (frases verbais ou VX),
- adjetivais (frases adjetivas ou AX).

[2] http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html
[3] http://verbs.colorado.edu/~mpalmer/projects/ace.html
[4] http://nlp.cs.nyu.edu/meyers/NomBank.html
[5] http://www.seas.upenn.edu/~pdtb/

Os *chunkers* não se sobrepõem, nunca dois *chunkers* agrupam o mesmo *token*, e não são hierárquicos. A entrada de um *chunker* é um texto tokenizado e etiquetado morfossintaticamente.

2.2.4 Extração de Informação

Segundo Zambenedetti (2002), há uma grande quantidade de dados no formato de textos eletrônicos que é impossível as pessoas lerem tanta informação diariamente. Então, devido a essa enorme quantidade de informação muitas vezes desperdiçada, a motivação cresceu para a realização de pesquisas e estratégias abordadas para gerenciar esse problema.

A Extração de Informação (EI) é a tarefa de encontrar informações a partir de grandes volumes de documentos ou textos, estruturados ou livres. Essa tarefa inicia com a coleção de tais textos, modificando a informação para fácil compreensão e posterior análise, de uma maneira simples, onde os pedaços de texto são isolados, a informação relevante é extraída e então os pedaços são unidos corretamente, segundo Zambenedetti (2002).

Ellen Riloff (1999) afirma que o objetivo de um Sistema de Extração da Informação é extrair informações de domínios específicos em textos de linguagem natural. O Sistema de Extração de Informação conta com dois recursos de domínios específicos, um deles é o dicionário de padrões de extrações e o outro é o dicionário léxico semântico. Esses padrões podem ser construídos manualmente ou gerados automaticamente através de técnicas como textos anotados com *tags* em relação ao domínio específico.

A grande quantidade de informação digital disponível acarretou o crescimento do interesse de desenvolver sistemas de Extração de Informação, tendo como uma abordagens de pesquisa o Processamento de Linguagem Natural, devido a uma quantidade dessa informação estar no formato de linguagem natural.

Segundo Zambenedetti (2002), "uma tecnologia de extração de informação bem desenvolvida permitiria criar rapidamente sistemas de extração para novas tarefas cujo desempenho ficaria no mesmo nível que o desempenho humano, mas ainda não se está neste nível. Porém, sistemas com um desempenho mais modesto, que perdem alguns termos e incluem alguns erros, podem ser proveitosos".

Não se deve confundir Extração de Informação (EI) com a área de Recuperação de Informação (RI). RI seleciona de uma coleção de documentos, documentos relevantes de acordo com a consulta do usuário, enquanto EI extrai informações relevantes de documentos (Gaizauskas e Wilks, 1998).

Para o processo de extração de informação, seguem-se dois passos. No primeiro passo, os fatos são extraídos do documento após uma análise do texto. O segundo passo, o processo de IE integra os fatos, produzindo fatos maiores ou novos fatos, por inferência. Finalizando, os fatos pertinentes são escolhidos para o formato requerido (Zambenedetti, 2002).

Os documentos podem apresentar algum nível de estruturação dos dados, como podem ser totalmente livres. Os tipos de texto poder ser definidos da seguinte forma:

Estruturado: texto apresenta regularidade no formato de apresentação das informações. Essa regularidade permite que cada elemento seja identificado com base em regras uniformes, que consideram marcadores textuais tais como delimitadores e ordem de apresentação dos elementos. Por exemplo, um formulário.

Semi-estruturados: textos que apresentam alguma regularidade na distribuição dos dados. Alguns dados apresentam uma formatação, enquanto outras informações aparecem de forma irregular, por exemplo: blogs, e-mail, mensagem de texto, não seguem um formato rígido, permitindo variações na ordem e da forma como os dados são apresentados.

Não-estruturados: são textos livres, sem estrutura, não existindo regularidade na apresentação dos dados. Por exemplo, documentos em formato pdf.

2.2.5 Abordagem para Extração de Informação

Segundo (Matos, 2010), há três tipos de abordagem para extração de informação: abordagem baseada em regras, abordagem baseada em dicionários e abordagem baseada em aprendizagem de máquina.

Abordagem baseada em dicionário: utiliza uma lista de termos para identificar ocorrências no texto. Apresenta a vantagem de armazenar informações de um determinado domínio e identificação de termos.

Abordagem baseada em regras: em geral faz uso de algum tipo de conhecimento, assumindo a forma geral do conhecimento como ele é estruturado. Utiliza padrões para extrair informações, utiliza análises linguísticas e semânticas para reconhecer uma ampla gama de possíveis maneiras de fazer afirmações sobre uma categoria de coisas. Porém, esta abordagem, apresenta algumas desvantagens como: redução da capacidade de adaptação de regras em outro sistema e exclusão de termos que não correspondem aos padrões predefinidos. Apresenta problemas de adaptação para novos domínios, contudo, tem bom desempenho, melhor que as outras abordagens (Matos, 2010).

Abordagem baseada em aprendizagem de máquina: utilizada para automatizar a obtenção das regras a serem usadas em um novo domínio. Apresentando alguns problemas como: necessidade de grande massa de dados e necessidade de treinamento com a entrada de novos dados. Contudo, apresenta como vantagens independência de domínio e a alta qualidade na predição (Matos, 2010).

Capítulo 3

Trabalhos Relacionados

Neste capítulo, são apresentados alguns trabalhos relacionados com a extração de termos para formação de vocabulários temáticos de um domínio pré-determinado utilizando Processamento de Linguagem Natural. São identificadas vantagens e desvantagens de cada trabalho e uma análise comparativa entre eles e o trabalho proposto nessa dissertação.

3.1 Construção Automática de Estruturas de Conceitos de Domínio Específico

Chen (2006) propõe a criação de uma estrutura de conceitos para cada domínio de conhecimento de interesse do usuário. A estrutura inclui os conceitos mais importantes de um domínio de conhecimento específico e o relacionamento entre os conceitos. A estrutura serve para padronizar vocabulários de vários domínios de conhecimento, auxiliando na ligação entre os vocabulários do usuário, criadores de informação e motores de busca. Apresenta como principais contribuições extração de conceitos e determinação de relacionamento.

O autor teve como motivação para criar a estrutura de conceitos o resultado insatisfatório na pesquisa realizada na Web, causado pela falta de padronização do vocabulário para descrever um mesmo conceito, gerando o problema de ambiguidade de palavras. Foi usada uma abordagem estatística para construção automática da estrutura de conceitos selecionando primeiro uma coleção de texto adequada para representar os domínios de interesse, encontrando evidências estatísticas sobre os termos e por fim realizando a análise estatística para a construção das estruturas conceituais. Apresenta como ponto positivo, análise de abordagem baseada em ocorrência e abordagem baseada em conteúdo, combina diferentes tipos de medidas de similaridade e diferentes tipos de probabilidade para determinação de melhor relacionamento. Apresentou desvantagem de relacionamento quando o contexto de coocorrência tem conteúdo semelhante.

3.2 Extração de Automática de Sintagmas Nominais

O trabalho de Lopes et al. (2010) propõe a extração automática de termos de um conjunto de textos do domínio de Pediatria. Para isso utilizou um modelo linguístico, para a extração de todos os sintagmas nominais, cujo núcleo é um nome, existentes no conjunto de textos. A análise foi apenas sobre bigramas e trigramas para posterior comparação com o conjunto verdade construído manualmente.

A extração automática de termos foi realizada em duas etapas: na primeira etapa usou-se um *parser* chamado PALAVRAS para a marcação de palavras e o extrator *ExATOLP*[6]. Foi utilizada nesse trabalho uma abordagem híbrida para extrair e organizar, de forma automática, o conhecimento de um domínio específico em Pediatria na forma de uma hierarquia de conceitos para construção de uma ontologia. Os termos são ordenados de acordo com sua relevância utilizando a frequência relativa do termo.

O trabalho foi validado comparando os termos extraídos automaticamente com os termos extraídos por especialistas. Obtendo como resultado da extração dos termos uma precisão entre 40% e 70%, dado como satisfatório.

3.3 Extração de Informação sobre Efeitos de Doenças em Artigos Científicos

Matos (2010) observou que diante da grande quantidade de informação eletrônica sobre doenças, disponível em formato textual, torna-se difícil a leitura e assimilação de tanta informação pelas pessoas.

Como solução, o autor propõe usar mineração de textos para identificar e extrair informações de artigos científicos que tratam de efeitos de doenças relacionadas ao domínio biomédico, com o objetivo de estruturar e armazenar essas informações em um banco de dados. Essas informações extraídas identificam os efeitos causados pela doença genética e degenerativa *Anemia Falciforme* (AF): efeito negativo da doença, efeito negativo do tratamento e efeito positivo do tratamento. O autor propôs, para essa metodologia de pré-processamento textual, uma combinação de três tipos de abordagem: aprendizagem de máquina, regras e dicionário.

[6] http://www.inf.pucrs.br/~ontolp/exato.php

A abordagem baseada em aprendizagem de máquina classifica as sentenças dos documentos textuais e classifica como efeitos negativos, positivos e outros. A abordagem baseada em dicionário identifica os termos relevantes nas sentenças classificadas. Finalmente, a abordagem baseada em regras identifica os padrões de extração de efeitos com expressões regulares.

3.4 Extração de Referências Bibliográficas

Gonçalves (2010) observou a falta de informação estruturada (metadados) nos documentos textuais digitais dificultando assim o processamento desses documentos, como o reconhecimento do título, autor, assunto, dados de publicação, quem referenciou, entre outros.

Para solucionar o problema, o autor usou técnicas de extração de informação de forma a reconhecer automaticamente nomes de entidades no texto não estruturado e as referências bibliográficas estruturadas, tais como: atributos de nomes de pessoas e referências temporais.

Este sistema foi desenvolvido utilizando uma abordagem manual, onde foi criada uma gramática de regras que dá suporte ao processo de extração de informação, mas apresentou como ponto fraco: limitações quanto ao domínio, devido às diferenças de estilo bibliográfico encontradas nos domínios. Como ponto forte, destacamos: o reconhecimento do maior número de referências bibliográficas.

Motivado para aumentar a abrangência da solução proposta sobre um maior número de referências bibliográficas o autor criou um sistema híbrido, utilizando uma abordagem manual e outra baseada em aprendizagem de máquina. Com essa solução o trabalho apresentou melhorias. No entanto, continuou apresentando um ponto fraco quando confrontado com referências ao longo do texto, contudo o ponto forte foi de aumentar a abrangência para o maior número de referências bibliográficas possíveis.

3.5 Considerações Finais

O trabalho proposto nesta dissertação, denominado RISO-VTD, usa uma abordagem híbrida, regras linguísticas e estatística, havendo diferenças em relação aos trabalhos citados, como mostra na Tabela 1. São considerados termos contendo a

combinação de mais de três palavras (multigramas), enquanto os outros trabalhos extraem até trigramas.

O trabalho não é restrito a um domínio específico, tornando o trabalho amplo. São identificadas siglas (acrônimos), formando um dicionário de siglas por domínios, para futura desambiguação.

Tabela 1 - Tabela de comparação entre os trabalhos relacionados e o trabalho proposto

Tabela de Comparação					
Criação de Vetores Temáticos de Domínios para a Desambiguação Polissêmica de Termos	Construção Automática de Estruturas de Conceitos	Extração de Automática de Sintagmas Nominais	Extração de Informação sobre Efeitos de Doenças em Artigos Científicos	Extração de Referências Bibliográficas	RISO-VTD
Abordagem Estatística	X	X	X	X	X
Abordagem Linguística		X			X
Criação de Regras sintáticas			X	X	X
Extração de Termos Compostos (N >3)					X
Independência de domínios					X
Dicionário de Acrônimos					X

Capítulo 4

RISO-VTD - Sistema de Criação de Vocabulários Temáticos de Domínio para Classificação de Documentos Digitais

Segundo Monnerat (2003) boa parte da ambiguidade polissêmica dos termos em um documento pode ser evitada com um maior esclarecimento do contexto. Uma forma de esclarecê-lo é ter o conhecimento prévio do domínio ao qual o documento trata. Assim, a classificação prévia de um documento digital permite identificar melhor o real significado dos termos ambíguos contidos no documento.

Como cada domínio de conhecimento costuma ter um vocabulário específico, pretende-se determinar vocabulários de domínios para então poder compará-los com o vocabulário de um documento qualquer e, assim, conseguir classificar o documento automaticamente.

Para determinar a terminologia típica de um domínio de conhecimento é preciso analisar previamente um número significativo de documentos associados ao domínio em questão.

Portanto, o presente capítulo mostrará o processo de criação desta terminologia típica de um domínio de conhecimento, detalhando cada etapa realizada. Para formação deste vocabulário criou-se regras sintáticas para a formação e extração de termos compostos de documentos, utilizando o *MontyLingua*[7], uma ferramenta para processamento de linguagem natural que realiza análise morfológica e sintática da língua inglesa. Serão criados vocabulários específicos para os principais domínios de conhecimento, denominados de **Vetores Temáticos de Domínio (VTD)** por meio de um sistema denominado de RISO-VTD.

[7] http://web.media.mit.edu/~hugo/montylingua/

4.1 Projeto RISO-T (Recuperação da Informação Semântica de Objetos Textuais)

Essa pesquisa está situada no contexto do projeto RISO-T (Recuperação da Informação Semântica de Objetos Textuais) cujo objetivo é criar um ambiente de indexação e recuperação semântica de documentos possibilitando uma recuperação mais acurada, melhorando o fator de precisão dos resultados mediante a diminuição da ambiguidade do sentido dos termos.

Neste contexto, o presente trabalho representa uma parte do ambiente RISO-T, um projeto de pesquisa desenvolvido pelo grupo de Sistemas de Informação e Bancos de Dados (SINBAD), do DSC/CEEI/UFCG, composto por vários professores, pesquisadores e bolsistas de iniciação científica. A Figura 1 mostra a estrutura geral do ambiente RISO-T onde o presente trabalho se enquadra na parte superior do projeto destacado por uma moldura.

Faz parte do projeto, criar Vetores Temáticos para os principais domínios de conhecimento, por intermédio de um processo de extração de termos baseado em Processamento da Linguagem Natural (PLN). Os Vetores Temáticos devem ser utilizados para classificar novos documentos para compor uma biblioteca digital. Com essa classificação, torna-se possível diminuir a ambiguidade dos termos indexados enriquecidos semanticamente, proporcionando uma recuperação de documentos mais precisa.

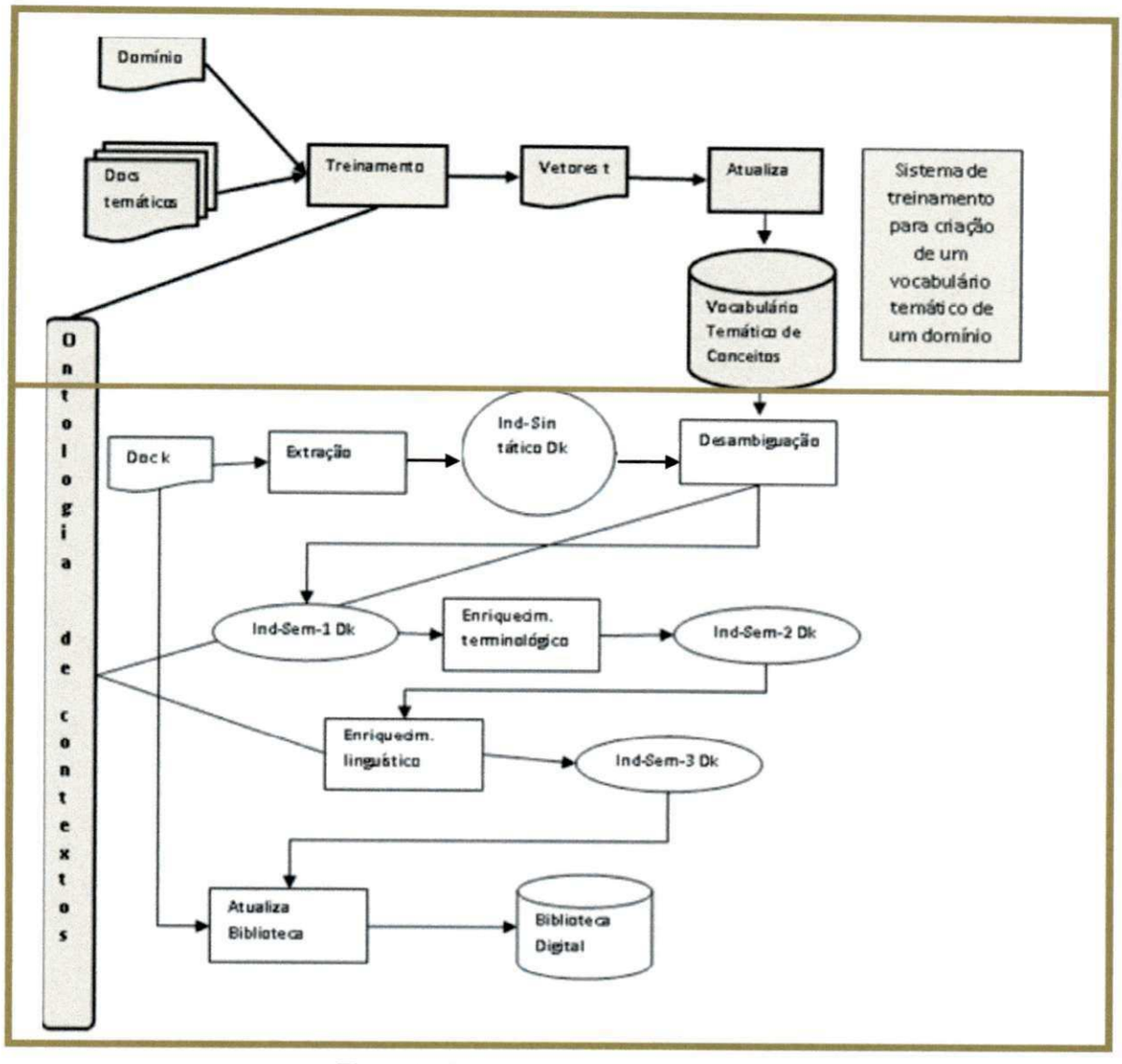

Figura 1 - Estrutura geral do ambiente RISO-T

4.2 Determinação da Coleção de Documentos para criação dos Vetores Temáticos

A Wikipédia[8], enciclopédia virtual e colaborativa, foi utilizada como fonte de informação para a escolha dos domínios de conhecimento e os seus respectivos documentos. Estes documentos formam a coleção que servirá como base para criação dos vetores temáticos.

A Wikipédia foi escolhida pelos seguintes motivos:

[8] http://en.wikipedia.org/wiki/Main_Page

- O compartilhamento de propriedades das categorias da Wikipédia com outras redes semânticas e lexicais, como exemplo, a *WordNet* (Miller, 1995);
- Seu acervo conta com milhões de entradas, centenas de milhares de colaboradores e milhões de artigos revisados (Spinellis, 2008);
- Abrange a maioria dos domínios de conhecimento;
- É dinâmica, sofrendo constantes atualizações e evoluções, acompanhando a evolução natural do conhecimento;
- O conteúdo possui qualidade comparável a enciclopédias tradicionais, tendo a vantagem das imprecisões serem rapidamente consertadas (Kittur e Kraut, 2008).

4.3 RISO-VTD

O processo de criação dos vetores temáticos é constituído por quatro módulos.

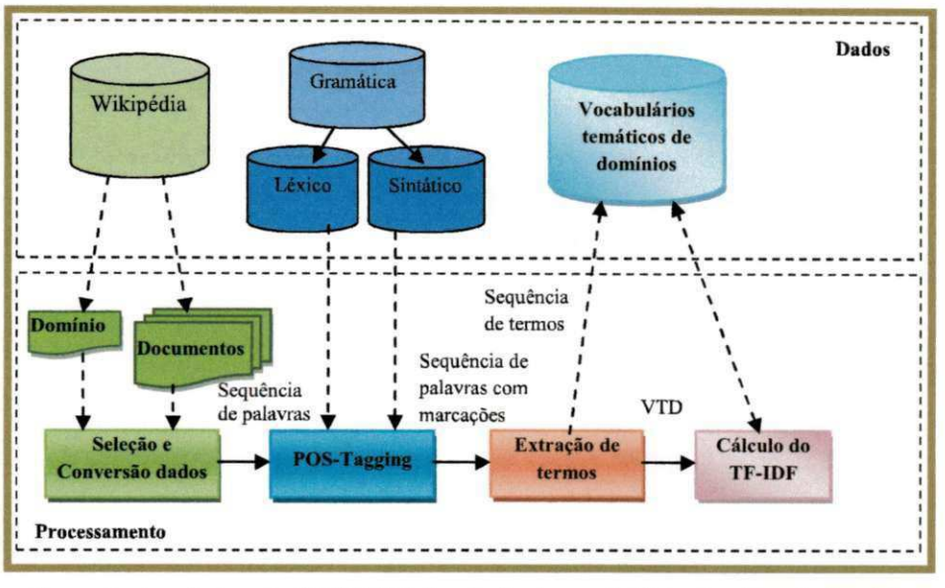

Figura 2 - Arquitetura do RISO-VTD de criação de vocabulários temático de domínio.

Estes módulos compõem o Sistema de Criação de Vetores Temáticos denominado de **RISO-VTD** que recebe como entrada um domínio de conhecimento *D* e um conjunto de documentos textuais (*corpus*), pertencentes ao domínio escolhido. O objetivo é determinar o vocabulário específico referente à *D*. Este processo repete-se para todos os domínios de interesse.

O funcionamento de cada módulo será detalhado a seguir, para melhor entendimento do processo.

4.3.1 Seleção e Conversão dos Dados de Entrada

A primeira etapa da solução proposta consiste em determinar, para os principais domínios de conhecimento, documentos representativos. Os documentos da Wikipédia estão organizados em uma hierarquia de categorias e esta, com suas subcategorias, podem ser entendidas como uma estrutura de termos (Xavier, 2009), com o objetivo de indexar os artigos (Völkel et. at. 2006).

Segundo a Wikipédia, artigo[9] é um texto sobre algum assunto. O conteúdo de um artigo pode envolver informações como tabelas, imagens, referências a outros artigos, hiperlinks para locais externos, entre outras informações. O formato do artigo depende do seu tamanho. Um artigo longo é dividido em seções e subseções e possui índice (Schönhofen et. al. 2007).

Cada artigo pode estar associado a diversas categorias da Wikipedia o que permite uma categorização múltipla e simultânea de tópicos, ou seja, algumas categorias podem ter mais de uma supercategoria (Syed et. al. 2008).

As categorias são uma organização de artigos que, segundo a própria Wikipédia[10], permite que os mesmos sejam agrupados e que estes grupos sejam categorizados. Desta forma, cada artigo contém um *link* para a página que descreve a categoria a qual ele pertence.

Cada página de categoria contém uma lista de *links* para artigos e subcategorias que pertencem à categoria, por exemplo:

Categoria: *Areas of Computer Science*, (contém 17 subcategorias)

Subcategorias: *Algorithms and data structures* (contém 5subcategorias e 4 artigos)

Artificial Intelligence (contém 31 subcategorias e 242 artigos),

onde, a categoria *Áreas da Ciência da Computação* é uma subcategoria de *Ciência da Computação*, como mostra a Figura 3.

[9] http://pt.wikipedia.org/wiki/Artigo

[10] http://en.wikipédia.org./wiki/categorization

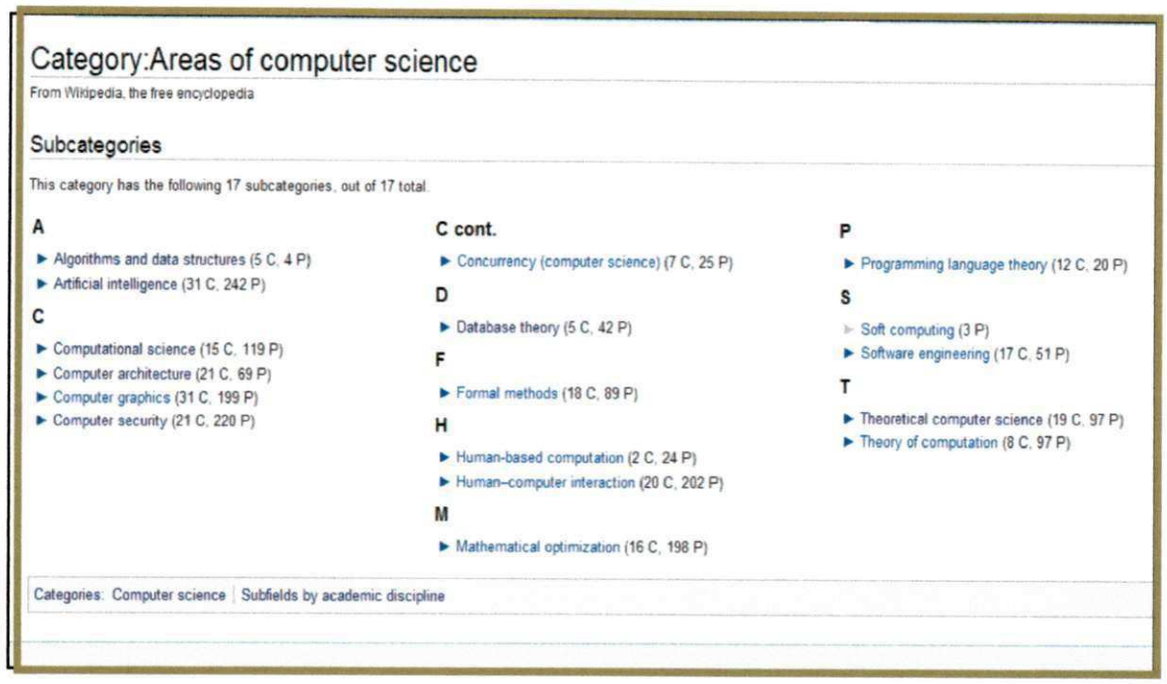

Figura 3- Página de categoria: Categoria de Áreas de Ciência da Computação

Para este trabalho as categorias da Wikipédia serão citadas como domínios de conhecimento, e os artigos como documentos, considerando os artigos pertencentes a cada categoria.

A escolha dos artigos é realizada aleatoriamente e os artigos são submetidos a um mecanismo de conversão cujo nome é "Criar um livro", disponibilizado pela Wikipédia. Após a escolha dos artigos, é gerado um livro, que tem como título o domínio (categoria) e, opcionalmente, um subtítulo (subcategoria). Em seguida, o livro é exportado gerando um arquivo no formato texto cuja extensão é *.odt*. ODT (*Open Document Text*) é um tipo de extensão usada por arquivos do *OpenOfficer Writer* para editar documentos de texto.

Contudo, para ser utilizado no Sistema RISO-T, o livro precisa ser convertido para o formato (.txt), visto que o modulo de análise do documento recebe como entrada arquivo deste tipo.

Um ponto importante da pesquisa foi a escolha dos domínios de conhecimento adequados e os documentos correspondentes.

A determinação dos domínios é inspirada na classificação utilizada em bibliotecas convencionais, denominada Classificação Decimal Universal (CDU)[11], que abrange todos

[11] http://pt.wikipedia.org/wiki/Classificação_decimal_universal

os domínios do conhecimento humano e segue uma estrutura hierárquica de domínios e subdomínios.

A CDU é um sistema de classificação de documentos que usa um código numérico para auxiliar a indicação de termos específicos de um assunto ou relações entre assuntos, sendo definida como "*uma linguagem de indexação e de recuperação de todo o conhecimento registrado, na qual cada assunto é simbolizado por um código baseado nos números arábicos*" (Souza, 2004, p.27).

Como a CDU representa uma classificação do conhecimento humano muito bem aceita e, por outro lado, os documentos a serem usados na criação dos vetores temáticos são escolhidos da Wikipédia, é necessário estabelecer uma relação entre as classes da CDU com as categorias de classificação da Wikipédia. Neste sentido foi feita uma comparação das principais classes e subclasses da CDU e as categorias e subcategorias da Wikipédia, determinando, na medida do possível, as devidas correspondências. Assim, as próprias entradas da Wikipédia podem ser utilizadas como documentos adequados para a criação dos **Vetores Temáticos**.

Nesse trabalho, usaremos os termos domínio e subdomínio para referenciar às classes e subclasses da CDU mostradas no Apêndice A e às categorias e subcategorias da Wikipédia mostradas no Apêndice B.

Durante a escolha dos domínios e dos seus subdomínios, a flexibilidade da estrutura de categorias e subcategorias da Wikipédia, gerou questionamentos importantes:

(i) Até que nível de profundidade deve ser considerado para escolher as subcategorias?

(ii) Quais subcategorias devem ser escolhidas para representar o domínio?

(iii) Quais documentos devem ser selecionados?

Levando-se em consideração esses questionamentos, decidimos considerar três níveis de profundidade para artigos referentes ao domínio de Ciência da Computação. Para os demais domínios descemos apenas o primeiro nível de profundidade. Descer mais do que três níveis torna-se inviável, dado que a grande quantidade de documentos demandaria um tempo de processamento elevado, o qual foi comprovado com os primeiros testes realizados utilizando o extrator. Durante os testes, alguns domínios demoraram cerca de oito horas de execução, como por exemplo, o domínio de literatura com cerca de 200 documentos.

Para obter uma melhor representação dos domínios da Wikipédia, escolhemos as categorias que tinham uma classe correspondente na CDU. Após a escolha das categorias e

das subcategorias de nosso interesse, selecionamos os documentos do primeiro nível de cada categoria para compor o livro criado na Wikipédia com a coletânea de artigos, rotulado pela categoria escolhida, que chamaremos de domínio.

A Tabela 2 ilustra uma classificação parcial da CDU, mostrando a relação das classes e subclasses. A Tabela 3 ilustra a distribuição de categorias da Wikipédia.

Tabela 2 - Relação parcial das classes e subclasses da CDU

0 Generalities
000 Computer science, knowledge &systems
001 Science and knowledge
002 Documentation. Books. Writings. Authorship
004 Computer science and technology. Computing
004.2 Computer architecture
1Philosophy. Psychology
100 Psychology
2 Religion. Theology
3 Social Sciences
300 Social sciences, sociology &anthropology
...

Tabela 3 - Distribuição parcial das categorias da Wikipédia

Concepts
Category
General reference
Culture na the arts
Geography and places
Health and fitness
History and events
Mathematics and logic
Technology and applied sciences
Computing
Computer Science
...

Segundo Urdician (2004), a CDU organiza o conhecimento em dez classes utilizando uma classificação decimal. A CDU contém uma tabela principal e tabelas auxiliares. A tabela principal contém todos os assuntos da classificação, enumerados hierarquicamente nas referidas 10 classes (Melro 2006). Para cada uma das dez classes da CDU, escolhemos categorias correspondentes na Wikipédia.

Os documentos escolhidos compõem um *corpus* de um determinado domínio de conhecimento o qual serve de base para a extração de termos e para a criação de uma terminologia característica para o domínio considerado.

A próxima etapa recebe o *corpus*, no formato de um arquivo de texto livre, contendo todos os documentos de um domínio de conhecimento escolhido, e faz a análise morfossintática das frases.

Para a realização da análise morfossintática das frases e extração dos termos escolheu-se *MontyLíngua*[12], é um conjunto de ferramentas individuais de Processamento de Linguagem Natural (PLN), para suprir todos os aspectos de processamento de texto em inglês. A sua escolha foi definida por ser diferente de várias outras ferramentas de PLN, enriquecido de senso comum e um software destinado à pesquisa. O *MontyLíngua* extrai o sujeito, verbo e o objeto das sentenças, além das frases adjetivas, nominais e verbais.

Para isso, o *MontyLíngua* possui classes para efetuar as tarefas sobre um texto:

- *MontyTokenizer:* separa o texto em *tokens*. Cada *token* representa uma palavra. Essa tokenização é feita separado as palavras por espaços em branco e outros caracteres como ponto, vírgula, ponto e vírgula, etc.
- *MontyTagger:* após a tokenização, são colocadas as *tags* para indicar as classes gramaticais, como, sujeito, verbo, artigo, adjetivo, etc. Para efetuar essa marcação é utilizado um conjunto de *tags Penn Treebank*.
- *MontyChunker*: após separar o textos em *tokens* e marcar com *tags*, os *tokens* são reagrupados em frases ou *chunks*. Um *chunker* pode ser:
    - Frase nominal (NX): essa frase tem um substantivo como núcleo.
    - Frase verbal (VX): essa frase tem um verbo como núcleo
    - Frase adjetiva (AX): essa frase tem o adjetivo como núcleo;

4.3.2 Análise do Documento utilizando o *MontyLíngua* (*PoS-Tagging*)

Como mostra Figura 4, a análise de cada documento é realizada em três etapas: Processamento Morfológico, Processamento Sintático e Extração de Termos, as quais serão detalhadas a seguir.

[12] http://web.media.mit.edu/~hugo/montylingua/

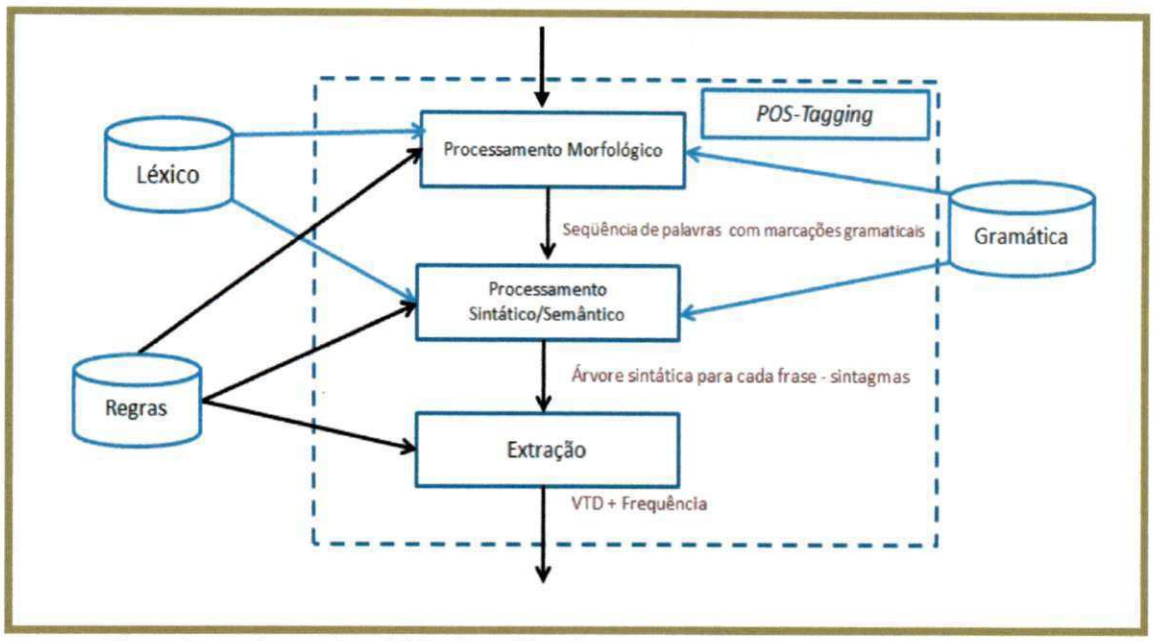

Figura 4- Arquitetura do módulo de Análise de Documento utilizando um POS-Tagging

a) Etapa de Separação de Texto – *MontyTokenizer*

Esse processo recebe o documento como uma única sentença que chamaremos de texto. O texto é separado em *tokens*. A separação do texto é feita com base em *tokens* delimitadores das frases, como espaços, ponto, vírgula, caracteres especiais, entre outros. Os *tokens* delimitadores podem variar de acordo com o tipo do texto, tipo de fonte e o idioma.

Para exemplificar a *tokenização,* é mostrado o exemplo:

'*The textual information retrieval has been the target of many research projects.*'

O resultado da *tokenização*, assumindo o espaço como *token* delimitador, é:

'The', 'textual', 'information', 'retrieval', 'has', 'been', 'the', 'target', 'of', 'many'', 'res earch', 'projects', '.'

A tokenização do texto é feita com o objetivo de criar estruturas de dados do tipo lista para separar o texto em termos para que, desta maneira, o processo de marcação (etapa seguinte) possa ser realizada termo a termo sem perder a referência do texto.

b) Etapa de Marcação do Texto – *MontyTagger*

A marcação de texto é realizada utilizando a técnica *Part-of-Speech Tagging* (*POS-Tagging*) (Brill, 1995; Brill, 1994), que consiste em etiquetar os termos identificados na etapa anterior utilizando uma *tag* de acordo com uma classe gramatical para indicar se

termo é verbo, sujeito, substantivo adjetivo, entre outras. As classes gramaticais usadas na marcação do texto são baseadas no conjunto de *tags Penn Treebank Tagset* (Marcus et al.,1993), em Sistemas de Processamento de Linguagem Natural (PLN), conforme mostrado na Tabela 4. Por exemplo, em "*My/PRPS*", "*cat/NN*" e "*slept/VBD*", "*PRPS*" significa pronome possessivo, "*NN*" um substantivo no singular e "*VBD*" um verbo no passado (ver Tabela 4).

Tabela 4 - Conjunto de tags que representam as categorias gramaticais baseadas no Penn Treebank Tagset

POS Tag	Descrição	Exemplos
CC	coordinating conjunction	and
CD	cardinal number	1, third
DT	Determiner	the
EX	existential there	there is
FW	foreign word	d'hoevre
IN	preposition/subordinating conjunction	in, of, like
JJ	Adjective	green
JJR	adjective, comparative	greener
JJS	adjective, superlative	grenest
LS	list marker	1)
MD	Modal	could,will
NN	noum, singular or mass	table
NNS	noun plural	tables
NNP	proper noun, singular	John
NNPS	proper noum, plural	Vikings
PDT	Predeterminer	*both* the boys
POS	possessive ending	friend's
PRP	personal pronoun	I, he, it
PRPS	possessive pronoun	my, his
RB	Adverb	however,usually,naturally, here, good
RBR	adverb, comparative	better
RBS	adverb, superlative	best
RP	Particle	give *up*
TO	To	*to go*, *to* him
UH	Interjection	uhhuhhuhh
VB	verb, base form	take
VBD	verb, past tense	took
VBG	verb, gerund present participle	taking

VBN	verb, past participle	taken
VBP	verb, sing.present, non-3d	take
VBZ	verb, 3rdperson sing. present	takes
WDT	wh-determiner	which
WP	wh-pronoun	who, what
WPS	possessive wh-pronoun	whose
WRB	wh-abverb	where, when

O objetivo do *tagging* é identificar a classe gramatical de cada termo em uma frase. Entretanto, devido à presença de ambiguidade léxica nos termos, determinar a classe gramatical correta do termo em questão torna-se difícil. Logo, para classificar, pode haver a necessidade de utilizar a classificação gramatical do termo vizinho.

Um termo pode receber classificações diferentes de acordo com as regras que determinam o comportamento dos vocábulos da linguagem. Por exemplo, a formação de plural na língua inglesa, a formação de contrações como ["'s "," 's "], ["ain't", "ai n't"] e["'ll ", " will "] e a flexão dos verbos (*past, present, progressive, past_progressive, future, infinitive*).

Dessa forma, os termos são tratados quanto ao formato, estrutura, flexão e classificação. Muitas vezes os termos são classificados de forma errada. Por exemplo, segundo a gramática da língua inglesa[13] "se uma palavra modifica um substantivo ou pronome, essa palavra é um adjetivo". Portanto, o verbo no particípio presente e no particípio passado seguido de um substantivo torna-se um adjetivo, esses adjetivos são denominados de adjetivo de particípio.

A gramática para verificação e marcação dos termos utilizada pelo *MontyLíngua* (MontyLíngua, 2005) é o *WordNet*[14]. De acordo com esta gramática, o termo "*grounded*" é considerado um verbo, então o termo "*grounded*" é marcado como Verbo no Tempo Passado ou *Verb Past Tense* (VBD) como mostra a Tabela 4. Já na Wikipédia, o termo "*grounded*" consta como adjetivo, levando em consideração a formação *"grounded theory*".

A Figura 5 mostra dois exemplos mostrando a marcação do texto realizado pelo *tagging*, com *tags* que representam as categorias gramaticais baseadas no *Penn Treebank Tagset*, mostradas na Tabela 4.

[13] http://www.rhlschool.com/eng4n7.htm
[14] http://wordnet.princeton.edu/

Exemplos:

1. Grounded theory $\Rightarrow$ *Grounded/VBD theory/NN*
2. My cat slept with me. $\Rightarrow$*My/PRPS cat/NN slept/VBD with/IN me/PRP ./.*

Figura 5 - Frases para exemplificar a marcação com tags

Considerando a frase "*grounded theory*" (Figura 5), a marcação correta seria "*grounded/JJ theory/NN*". No entanto, o termo "*grounded*" é marcado como verbo, "*grounded/VBD theory/NN*", onde "*JJ*" significa adjetivo, "*NN*" um substantivo no singular, "*VBD*" um verbo no tempo passado, como mostrado na Tabela 4.

Outro exemplo é a frase "*My cat slept with me*" (Figura 5). Após a marcação, a frase ficará no seguinte formato: "*My/PRPS cat/NN slept/VBD with/IN me/PRP./.*", onde "*PRPS*" significa pronome possessivo, "*NN*" um substantivo no singular, "*VBD*" um verbo no passado, "*IN*" significa preposição, "*PRP*" significa pronome pessoal.

Quando uma palavra é desconhecida pelo analisador, este inicialmente marca-a como desconhecida utilizando a *tag/UNK*. Por esse motivo, uma segunda análise é feita, se a classe gramatical correta não for encontrada, a palavra é marcada como substantivo utilizando a *tag/NN* (*Noun* ou Substantivo).

Após a marcação das palavras com suas respectivas *tags*, as frases são reagrupadas em frases nominais, verbais e adjetivas.

Para exemplificar temos como entrada a frase:

"*My car is very beautiful*".

Após a marcação e agrupamento dos tokens conforme as classes gramaticais, obtêm-se três frases: uma frase substantiva, uma verbal e outra adjetiva, como mostra a saída a seguir:

"<NX>*My car*<NX>" , " <VX>*is*<VX>" , "<AX>*very beautiful*<AX>"

Finalizando essa etapa, as frases são fornecidas ao extrator para efetuar a extração dos termos.

c) Etapa de extração dos termos - *RISOExtrator*

A extração de termos tem como entrada o texto marcado com *tags* e agrupado em frases nominais, verbais e adjetivas. A etapa de extração contempla as principais alterações feitas com inclusões de regras heurísticas criadas para formação e realização da extração dos termos, com o objetivo de obter uma maior combinação de palavras, obtendo termos mais relevantes.

A frase é analisada e, dependendo do seu conteúdo, as frases são criadas com mais de um *token*, sempre iniciando com um substantivo ou adjetivo. O novo termo pode ser composto por *n tokens*. Exemplificando, temos como base o primeiro exemplo mostrado na fase de tokenização.

1. Frase lida:

   '*The textual information retrieval has been the target of many research projects.*'

2. Frase tokenizada:

   '*The*', '*textual*', '*information*', '*retrieval*', '*has*', '*been*', '*the*', '*target*', '*of*', '*many*', '*research*', '*projects*', '.'

3. Frase após a marcação com as classes gramaticais:

   The/DT textual/JJ information/NN retrieval/NN has/VBZ been/VBN the/DT target/NN of/IN many/JJ research/NN projects/NNS.

4. Frases agrupadas em frases nominais, verbais e adjetivas:

   <NX The/DT textual/JJ information/NN retrieval/NN NX>

   <VX has/VBZ been/VBN VX>

   <NX the/DT target/NN NX> of/IN

   <NX many/JJ research/NN NX> projects/NNS.

onde <NX> significa frase substantiva, <VX> frase verbal.

Observa-se que as preposições ficam de fora das frases agrupadas, portanto, para a formação dos novos termos, há uma desconstrução dessas frases, analisando apenas as marcações gramaticais, com o intuito de não perder as preposições, conjunções e algumas palavras que não entram nas frases agrupadas quanto à formação gramatical.

d) Determinação dos Termos para Criação dos Vetores Temáticos

O objetivo dessa fase é extrair os termos relevantes ao domínio considerado. Esses termos são obtidos de acordo com heurísticas criadas e adicionadas ao *RISOExtrator*, mediante observação da formação dos termos no texto. Os termos são identificados, extraídos e armazenados em uma tabela rotulada com o nome do documento em evidência, no formato <dominio>. Esta tabela será adicionada ao Vetor Temático de Domínio (VTD), que servirá como base de consulta para posterior classificação de documentos nos domínios. A Figura 6 mostra o esquema da tabela de termos de um domínio.

Esquema do Vetor Temático:

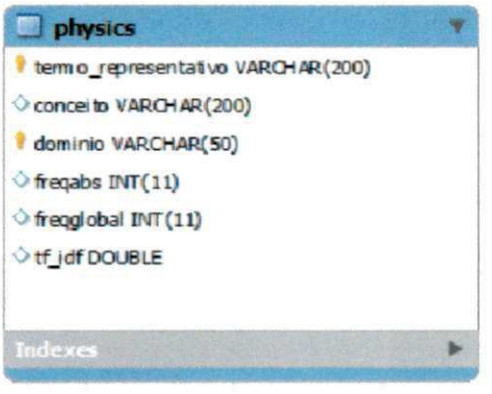

Figura 6 - Esquema da tabela que contém os termos extraídos para cada domínio.

A tabela é composta por: termos extraídos (termo_representativo), o conceito (tipo de frase) domínio ao qual o termo pertence (domínio), frequência do termo (freqabs) indicando o número de vezes em que o termo aparece no documento, frequência global que indica quantos domínios contém o termo (freqglobal). Esta informação é usada para posterior determinação da importância do termo em relação aos outros domínios.

Para determinação da importância do termo no domínio é realizado o cálculo do peso tf-idf.

e) Cálculo da Frequência Local e Global – *TF-IDF*

O cálculo da frequência do termo determina a importância de cada termo em relação ao domínio em que ele está inserido, denominado *tf* (*term frequency*). A frequência global quantifica a importância do termo em relação a todos os domínios, denominado *idf* (*inverse document frequency*).

4.4 Aplicação de Heurísticas para composição e extração de novos termos utilizando o *RISOExtrator*

As heurísticas são regras para composição do termo, combinação de adjetivos, substantivos próprios e comuns e verbos, preposições e conjunções. Os termos resultantes podem ser simples ou composto de n-gramas, ou seja, n palavras. Extraem do texto, substantivos, locuções substantivas, nomes próprios. Elas foram criadas a partir da observações do comportamento da linguagem dentro do texto.

As frases normalizadas com o processo de marcação são analisadas, selecionadas e, de acordo com as heurísticas, são formados os termos que melhor representam o domínio estudado.

Os termos duplicados são descartados, mas o número de vezes em que eles aparecem é contabilizado no campo 'frequência'. As regras que determinam quais frases devem ser mantidas para análise são:

- As frases com uma preposição são mantidas;
- As frases com conjunções geram novos termos;
- As frases com verbos são relevantes quando esses têm função de adjetivo;

A notação de termos e outros elementos linguísticos é definida por regras BNF enquanto a estrutura geral do formato da descrição de heurísticas é:
SE <condição> ENTÃO <composição> {E <composição-1>} (É TERMO ¦ SÃO TERMOS)

A relação completa das heurísticas é apresentada no Apêndice C. Mostramos aqui alguns exemplos representativos destas regras.

Os metassímbolos "<" e ">", são usados para delimitar variáveis representando tags e termos.

Como exemplo, a heurística que analisa uma frase que contém o conectivo '*and*' é:

SE <subst-1> 'and' <subst-2> ENTÃO <subst-1> E <subst-2> SÃO TERMOS

SE <adj-1> 'and' <adj-2 subst> ENTÃO <adj-1> <subst> e <adj-2> <subst> SÃO TERMOS

}

Como exemplo de extração de termos utilizando o conectivo "*and*" temos:

1. Frase lida:

   "*Employee and customer account records*"

2. Frase tokenizada:

   "Employee", "and", "customer", "account", "records"

3. Frase marcada com *taggs*:

   "Employee/NN and/CC customer/NN account/NN records/NNS"

4. Frase agrupada em frase nominal;

   <NX Employee/NN and/CC customer/NN account/NN records/NNS NX>

Nesse exemplo não houve necessidade de desfazer o agrupamento, porque todos os termos estão presentes na frase marcada. Caso contrário, não seria possível construir termos compostos com maior significado semântico. Usando as regras acima, a saída fica no seguinte formato.

5. Resultado da extração dos termos:

   a. caso 1 - extrai um substantivo

      Employee

      Customer account records

   b. caso 2 - após combinação das palavras.

      Employee account records

      Customer account records

Mostrando outro exemplo, formando dois termos compostos, com a seguinte frase:

1. Frase lida:

   "*Spatial and Temporal Database*"

2. Frase tokenizada:

   "Spatial " "and" " Temporal " "Database"

3. Frase marcada com *taggs*:

   " Spatial /NN and/CC Temporal /NN Database /NN "

4. Frase agrupada em frase nominal;

   <NX Spatial /NN and/CC Temporal /NN Database /NNS NX>

5. Resultado da extração dos termos:

   a. caso 2 - após combinação das palavras.

      Spatial Database

      Temporal Database

Os ajustes e modificações feitas nos métodos de tokenização, de marcação e de extração são relatados a seguir:

a) Ajustes na fase de tokenização

Foram feitos ajustes para aperfeiçoar a separação dos *tokens* e formar frases adicionais, facilitando a análise e melhorando o resultado final na formação de termos.

Foram adicionados aos *tokens* delimitadores alguns símbolos, como: ponto-e-vírgula, dois pontos, vírgula, interrogação, exclamação, barra, reticências, entre outros, com o objetivo de generalizar os *tokens* delimitadores ao máximo devido às diversas formatações dos textos, dado que podem ser analisados textos de diversos domínios e consequentemente formatações variadas.

Foi criado um processo de detecção de abreviaturas comuns e acrônimos para cada domínio, para facilitar a desambiguação de documentos no momento da classificação.

Os acrônimos encontrados no texto são verificados juntos aos termos adjacentes quanto à veracidade do termo em relação à sigla. Cada acrônimo encontrado é guardado juntamente com o texto equivalente ao acrônimo, em um vetor rotulado com o nome do domínio de conhecimento do qual ele foi extraído.

Foram adicionados regras de exceção, ainda no tokenizador, para reconhecimento de alguns tipos de termos que não são comuns, devido à mudança do tipo de escrita que diferem de acordo com cada domínio, essas regras são mostradas a seguir:

- Eliminação de números, preservando datas e medidas;
- Reconhecimento de endereço eletrônico, palavras entre parênteses, entre aspas duplas entre outros;
- Reconhecimento de referência bibliográfica;
- Acréscimo de uma lista de palavras comuns, como '*Edited*', '*Chapter*'.

Com esses ajustes, o reconhecimento de frases após a marcação morfossintática tornou-se mais limpo e mais detalhado ao mesmo tempo, facilitando a interpretação das mesmas.

b) Ajustes na fase de marcação

Nesta fase, onde ocorre a marcação morfossintática são feita a seguinte alteração:

- No caso do pronome pessoal "I", a classificação correta dá-se quando a escrita é maiúscula, sendo que, algumas vezes, sua apresentação escrita no texto estava no formato minúsculo o que é classificado errado.

c) Ajustes na fase de extração

As frases são recebidas da fase de marcação já classificadas como frase nominal, frase verbal ou frase adjetiva. No entanto, na extração, todas as frases sem distinção da classificação são desagrupadas, voltando ao formato de marcação por *tokens*. A frase é analisada por completo, sem descartar palavras classificadas como *stop-words*. Dessa maneira, a criação dos termos compostos torna-se mais significante. Foram criadas aproximadamente 40 heurísticas para extração de termos simples e termos compostos (Apêndice C).

4.5 Armazenamento dos Vetores Temáticos

O armazenamento dos vetores temáticos é realizado em duas etapas: na primeira etapa é realizada a geração do vetor de termos por domínio, sendo cada vetor rotulado com o nome do documento; a segunda etapa é o armazenamento de todos os vetores de termos criados por domínios armazenados numa única base denominada Vetores Temáticos de Domínios (VTD). O VTD tem o mesmo esquema dos vetores criados para cada domínio, cada registro conterá o termo extraído, o domínio ao qual ele pertence, a frequência absoluta (quantidade com que o termo aparece no domínio), a frequência global (número de domínios nos quais aparece o termo), e o peso relativo do termo (*tf-idf*). O esquema do VTD é mostrado na Figura 7.

Essa base servirá de consulta para as futuras classificações de documentos podendo ser atualizada com novos termos, mediante a chegada de novos documentos como mostra na figura 8.

Figura 7 - Esquema da tabela do Vetor Temático de Domínios

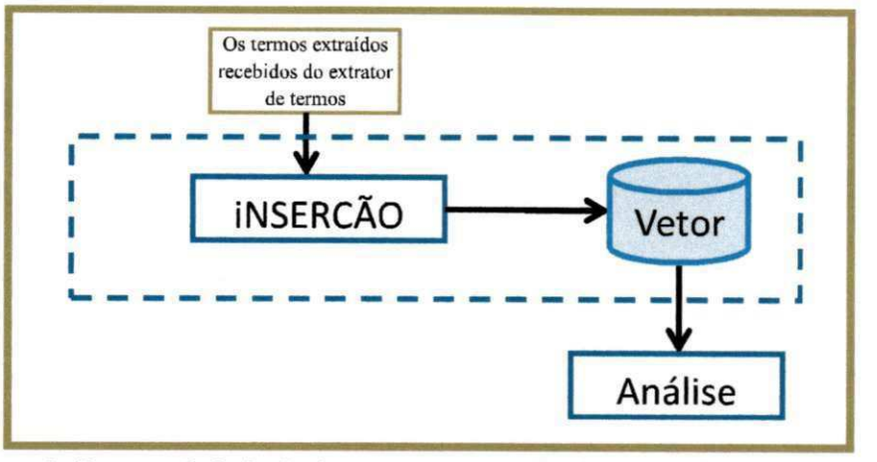

Figura 8 - Processo da inclusão dos termos no vetor temático e análise dos resultados

4.6 Considerações Finais

Para alcançar o objetivo de construir vocabulários representativos dos diversos domínios de conhecimento, foram utilizados artigos coletados da Wikipédia. Para tornar a extração dos termos mais efetiva foram efetuadas diversas análises e transformações dos textos originais. Primeiramente, foram eliminados 'ruídos', *stopwords* e outras palavras não representativas. Em seguida, foram feitas análises sintáticas das frases para determinar os termos compostos significativos oriundos da composição das palavras auxiliares, como por exemplo, as conjunções, as preposições, os adjetivos e os verbos. Para tal, foi criada uma série de heurísticas para determinar estes termos.

A identificação de sinônimos e outras relações semânticas entre termos não faz parte deste trabalho, ficando para uma etapa posterior no ambiente RISO-T, que está sendo desenvolvida por outra dissertação e mestrado.

O próximo capítulo será relatado o experimento que foi realizado para validar os VTD's.

Capítulo 5

Experimentos e Validação dos Resultados

Neste capítulo é apresentado o estudo de caso realizado para validação do processo de criação de vocabulários temáticos proposto nesse trabalho para domínios de conhecimento, como Agricultura, Artes, Ciência da Computação, entre outros. Primeiro foram explicados os tipos de experimentos e, em seguida, foi detalhada sua realização e analisados os resultados. Além disso, serão apresentados instrumentos que auxiliam na avaliação da efetividade da construção de vocabulários específicos de domínios de conhecimentos.

5.1 Classificação de documentos utilizando três bases de classificação diferentes.

Para realização dos experimentos foram escolhidos seis domínios do conhecimento com seus respectivos documentos, os quais compuseram a base de criação dos Vetores Temáticos de Domínios.

Considerando o objetivo do trabalho de criar vocabulários representativos dos domínios de conhecimento para classificar corretamente documento no domínio esperado, a qualidade do VTD foi analisada através de três experimentos diferentes, que serão detalhados no decorrer do capítulo:

1. Corretude da classificação de 43 novos documentos com auxílio dos vetores temáticos criados. Para obter esta classificação foi implementado (na linguagem de programação *Phyton*) a fórmula clássica de cálculo de similaridade do modelo espaço-vetorial usada na recuperação da informação (Salton, 1988) ou *Space Vetor Model (SVM)*. Neste modelo os documentos são representados como vetores em um espaço de termos e a cada termo está associado um peso, O termo especifica a direção e o peso o tamanho da especificação do vetor. A distância (cosseno) entre os vetores dos documentos indica a similaridade entre eles. O vetor de um novo documento de um domínio conhecido foi comparado com os vetores temáticos de todos os domínios e o vetor temático que obter um ângulo mais próximo com o vetor do novo documento corresponde ao domínio adequado.

2. Comparação com o classificador *Intellexer Categorizer*[15]. Este sistema foi treinado com a coleção de documentos usados na criação dos domínios para classificação de novos documentos. O resultado da classificação será comparado ao resultado obtido com a classificação utilizada no presente trabalho.

3. Geração de vetores de termos através do software Weka. A ferramenta Weka contém uma coleção de algoritmos de aprendizagem de máquina comumente utilizados em tarefas de mineração de dados. Foi escolhido um algoritmo para gerar os vetores de termos a partir da coleção de documentos utilizada para criação dos VTD´s e, posteriormente, testá-los. Este vetor de termos serviu de base de consulta para a classificação de novos documentos, utilizando o *SVSim*, o resultado da classificação foi comparado aos dos vetores gerados pelo VTD.

Para possibilitar a comparação dos resultados, o ambiente de cada classificador foi configurado com parâmetros similares e foi usada a mesma coleção de documentos coletados da Wikipédia.

Para compor a base de documentos a ser usada nessa validação, foram escolhidos os seguintes domínios: Agricultura, Artes, Ciência da Computação, História, Literatura e Física. No domínio de Ciência da Computação, foram escolhidos seis subdomínios: Algoritmos e Estrutura de Dados, Inteligência Artificial, Ciência da Computação, Arquitetura de Computadores, Segurança de Computador e Banco de Dados. Para cada domínio foram escolhidos aleatoriamente documentos a fim de compor uma coleção específica para o treinamento e a criação dos VTD's. Desta maneira, pode-se obter uma terminologia própria para cada domínio. A Tabela 5 mostra a lista dos domínios, subdomínios e a quantidade de documentos contidos em cada domínio.

Tabela 5 - Domínios, subdomínios e quantidade de documentos em cada domínio

Domínios	Subdomínios	Total de Documentos
Agriculture	-	158
Arts	-	55
Computer science	-	
	Algorithms and Data Structure	3
	Artificial Intelligence	200

[15] http://categorizer.intellexer.com/

	Computacional Science	119
	Computer Architecture	57
	Computer Security	200
	Database	42
History	-	34
Literature	-	77
Physics	-	187

Para cada termo extraído foi armazenada a frequência relativa do termo (*tf*) no domínio, a relação entre a quantidade de domínios que contém o termo e o domínio ao qual o termo pertence (*idf*), além da composição dos dois, calculado pela fórmula (TF-IDF) que define a importância do termo no domínio em questão. Como ilustração a Tabela 6 mostra o domínio *Agriculture*, e os 23 primeiros termos extraídos, com os respectivos valores calculados, freq(i,d), a frequência do termo no documento, n(t), a quantidade de domínios em que o termo aparece e o peso do termo no domínio em questão, obtido pela fórmula tf(t,d) * idf(t).

Lembramos as fórmulas apresentadas no capítulo 3:

$$tf(t,d) = \frac{freq(t,d)}{freq\_max(d)}$$

$$idf(t) = log10\left(\frac{total\_dominios}{n(t)}\right)$$

$$\text{peso(t)} = \text{tf(t,d)} * \text{idf(t)}$$

Ilustramos o cálculo do peso através de dois exemplos com termos do domínio *Agriculture*. Quantidade de domínios que compõe o VTD: 11.

Exemplo 1: Termo novo para ser classificado: *Agriculture*

Como este é o termo mais frequente no domínio sua frequência será igual à frequência máxima. Logo freq(t,d) = freq_max(d)=646

Número de domínios em que o termo apareçe: n(d) = 1

$$peso_{termo} = \left(\frac{646}{646}\right) \times log10\left(\frac{11}{1}\right) = 1.0413926851582251$$

Exemplo 2: Termo para obtenção do peso: *agricultural water use*

Frequência do termo no documento: freq(t,d) = 4

Número de domínios que o termo apareçe: n(d) = 1

$$peso_{termo} = float\left(\frac{4}{646}\right) \times log10\left(\frac{11}{1}\right) = 0.006448251436865472$$

Tabela 6 - Visão Parcial da relação de termos extraídos dos documentos do domínio Agricultura.

Termo Extraído	Domínio	freq(t,d)	n(d)	freqmax	Peso do termo
Agricultural water management	agriculture	4	*1*	646	0.006448251436865472
agricultural water use	agriculture	4	*1*	646	0.006448251436865472
agricultural zones	agriculture	2	*1*	646	0.003224125718432736
agricultural zoology	agriculture	4	*1*	646	0.006448251436865472
agriculturalists	agriculture	4	*1*	646	0.006448251436865472
Agriculture	agriculture	646	*1*	646	1.0413926851582251
Agriculture accounts	agriculture	4	*1*	646	0.006448251436865472
agriculture articles	agriculture	2	*1*	646	0.003224125718432736
agriculture BIA	agriculture	2	*1*	646	0.003224125718432736
Agriculture categories	agriculture	2	*1*	646	0.003224125718432736
Agriculture CGRFA	agriculture	4	*1*	646	0.006448251436865472
Agriculture Community of Practice Founding Group	agriculture	2	*1*	646	0.003224125718432736
agriculture cultural control	agriculture	2	*1*	646	0.003224125718432736
Agriculture Data Warehouse	agriculture	2	*1*	646	0.003224125718432736
agriculture dates back thousands	agriculture	4	*1*	646	0.006448251436865472
Agriculture Department	agriculture	2	*1*	646	0.003224125718432736
Agriculture Development	agriculture	2	*1*	646	0.003224125718432736

A Tabela 7 mostra a variação do peso para termos que pertencem a mais de um domínio.

Tabela 7 - Termos que constam em mais de um domínio.

Termo Extraído	Domínio	freq(t,d)	n(d)	freqmax	Peso do termo
abandoned	*Agriculture*	9	4	365	0.01083286083543129
abilities	*Artificial_Intelligence*	6	3	390	0.008681098711581757
ability	*Agriculture*	19	9	365	0.004536584440000768
ability	*Art*	15	9	634	0.002061912586268853
ability	*Artificial_Intelligence*	62	9	390	0.013854643221946314

ability	*Computational_science*	11	9	254	0.003774220151353881
ability	*Computer_architecture*	18	9	233	0.006732631517450386
ability	*Computer_security*	38	9	378	0.008761128721924846
ability	*Database*	7	9	192	0.003177350124489332
ability	*Literature*	7	9	186	0.0032798452673050713
ability	*Physics*	11	9	285	0.003363690969735269
ABox	*Artificial_Intelligence*	11	1	390	0.02937261398315144
absolute value	*Art*	6	2	634	0.007006586672545845
absorbed	*Physics*	6	5	285	0.007208898345380685
absorption	*Computational_science*	24	3	254	0.053316984994846786
abstract	*Art*	10	9	634	0.0013746083908459021
Abstract	*Artificial_Intelligence*	11	9	390	0.0024580818591469207
abstraction	*Artificial_Intelligence*	14	7	390	0.0070464742606518305

Para validar o RISO-VTD foram realizados três experimentos, sendo que todos usaram a mesma coleção de documentos para extrair os termos, e criar os seus vetores de termos. Cada experimento tem uma abordagem diferente de criação dos vetores de termos. Os vetores de termos foram utilizados para fazer a classificação de outros documentos, usando dois tipos de classificadores, *SVSim* e *Intellexer Categorizer.*

No primeiro experimento, os documentos novos foram classificados utilizando o classificador *SVSim* e o VTD como base de consulta.

5.2 Primeiro Experimento - Corretude da classificação utilizando os vetores temáticos gerados pelo *RISOExtrator*

Nesse primeiro experimento, os VTD's criados nesse trabalho serviram de base para a classificação de novos documentos. Para essa classificação foi usado o modelo vetorial (Salton, 1983), implementado em *Phyton*, para calcular a similaridade entre o vetor de um documento novo com os Vetores Temáticos denominado de *SVSim*.

O experimento se baseou em uma coleção de documentos, denominado *ground-truth* ou conjunto verdade, cujo domínio já é conhecido. Para criação deste conjunto verdade, foram coletados aleatoriamente 41 documentos na Wikipédia. Os documentos pertencentes ao conjunto verdade, foram submetidos ao *MontyLíngua* passando por todo o processo de criação semelhante aos vetores do VTD. Cada vetor foi rotulado pelo nome do próprio documento e o cálculo do peso foi semelhante ao cálculo do peso dos vetores que formaram o VTD.

Como os vetores são criados com tamanhos diferentes devido à variação da quantidade de termos extraídos, é importante a normalização do TF para minimizar o problema entre os tamanhos dos vetores de termos do VTD e do novo documento.

Para esta normalização utilizou-se a uma equação simples. Para cada termo o TF o resultado da frequência do termo no documento é dividido pela frequência máxima no documento (vide Equação 3.1.).

A Tabela 8 mostra o conjunto verdade escolhido, os documentos retirados da Wikipédia, com o conhecimento prévio da categoria, possibilitando a comparação do resultado final da classificação.

Tabela 8- Documentos a serem classificados e seus respectivos domínios

Documentos para serem classificados	**Domínios**
3D_reconstruction_from_multiple_images	Artificial Intelligence
Accelerated processing unit	Computer Architecture
Affective Computing	Artificial Intelligence
Agricultural_biodiversity	Agriculture
Agroecology	Agriculture
Algorithm Characterizations	Algorithms
Anachronism	History
Antiquarian	History
Bidirectionlization	Database
Branch predication	Computer Architecture
Business continuity	Computer Science
Classic Book	Literature
Collision detection	Physics
Collostructional analysis	Artificial Intelligence
Complex instruction set computing	Computer Architecture
Concurrent data structure	Computer Science
Database storage structures	Database
Distributed language	Literature
Ecology of contexts	Agriculture
Experimental language	Literature
Grammatology	Literature
Graph Database	Database
Hash table	Algorithms
Historic Districts Council	History
History of Computer Science	Computer Science
Hystory of Mathematics	Mathematics

Indeterminacy	Literature
Information assurance	Computer Science
Integrational linguistics	Literature
Morris method	Computer Science
Multiphysics	Physics
Object language	Literature
Outline of Computer Science	Computer Science
Potencial theory	Physics
Preservationist	History
Pseudopotential	Physics
Search data structure	Algorithms
Security information management	Computer Science
Serialization	Algorithms
Sustainable agriculture	Agriculture
The arts and politics	Arts

Para efeito de observação, na etapa do cálculo do peso, realizaram-se alguns testes com a fórmula (tf-idf), alternando no uso do IDF para o cálculo do peso:

a. Peso sem considerar o IDF: observou-se o impacto causado computando só a frequência local ao documento. Foram obtidos valores mais altos no cálculo do cosseno, mas a precisão quanto à escolha do domínio real foi menor.
b. Peso considerando o IDF: observou-se que os valores computados no cálculo são muito pequenos, contudo, a resposta quanto ao cálculo da similaridade foi mais precisa, obtendo um melhor resultado na predição dos domínios.

A Tabela 9 mostra o resultado da similaridade encontrada entre os VTD's e o documento "*History of Mathematics*", usando a fórmula do cálculo de peso variando o uso do IDF. Vê-se que ambas obtiveram resultados satisfatórios.

Tabela 9 - Similaridade entre o vetor do documento de *History of Mathematics* e dos vetores VTD com IDF e sem IDF

	History of Mathematics (math)	History of Mathematics (math)
Domínios	Ambos sem IDF	Ambos com IDF
Agriculture	16,72	1,15

Algorithms and Data Structure	3,88	0,33
Art	14,95	1,68
Artificial Intelligence	21,86	1,99
Computacional Science	24,16	2,17
Computacional Architeture	11,94	0,82
Computacional Security	14,42	1,13
Database	12,31	0,79
History	38,48	4,05
Literature	22,31	1,63
Physics	20,87	2,14

Ambas as fórmulas apresentaram resultados de classificação nos domínios corretos, entretanto a fórmula utilizando o IDF obteve maior precisão nos resultados, inclusive nos resultados secundários. Portanto a partir dos próximos experimentos só foi utilizada a fórmula com IDF, abandonando-se a fórmula sem IDF.

Tabela 10 - Tabela com o resultado de classificação de oito documentos usando o SVSim/VTD

Documentos	Affective Computing [IA]	Potencial theory [Fis]	Outline of Computer Science [SC]	Classic Book [Liter]	Bidirectionalization [BD]	Algorithm Characterizations [Algoritm]	Indeterminacy [Lit]	Antiquarian [Hist]
Domínios								
Agriculture	0,94	0,16	0,46	0,67	0,04	0,56	0,16	0,60
Algorithms and Data Structure	0,36	0,08	2,80	0,03	0,05	3,67	0,03	0,02
Art	2,95	0,25	0,60	1,92	0,07	0,98	0,49	1,71
Artificial Intelligence	**7,70**	0,59	**6,53**	1,48	0,27	**3,97**	0,34	1,09
Computacional Science	2,50	0,64	3,54	0,92	0,13	2,32	0,23	0,76
Computer Architeture	1,49	0,20	2,43	0,74	0,05	2,50	0,1	0,32
Computer Security	2,46	0,28	3,01	1,34	0,15	1,46	0,22	0,74
Database	1,52	0,26	2,62	0,87	**0,59**	1,08	0,12	0,41
History	0,74	0,21	0,47	1,36	0,03	0,42	0,23	**10,30**
Literature	1,74	0,22	0,46	**6,75**	0,04	0,94	**2,41**	1,50
Physics	1,93	**3,41**	0,91	0,83	0,05	1,40	0,24	0,54

O documento *"Outline of Computer Science"*, como pode ser visto na Tabela 10, apresenta uma característica interessante. Esse documento trata de resumos sobre todas as áreas da Ciência da Computação. Contudo o classificador *SVSim* predisse todos os domínios que o documento faz referência, mostrando valores muito próximos no cálculo da similaridade. O domínio que ele prediz como dominante, é Inteligência Artificial, porque é o assunto a que ele mais se refere no texto. De acordo com o conjunto verdade esse documento pertence ao domínio de Ciência Computacional.

A realização desse experimento utilizando os VTD's e o classificador *SVSim*, apresentou resultados satisfatórios. Segundo os resultados mostrados no Apêndice D, na tabela 18, dos 41 documentos analisados 38 documentos foram classificados corretamente, mostrados em cinza de acordo com o conjunto verdade, e 13 documentos foram classificados errado mostrados em marrom diferente do conjunto verdade.

Mediante a necessidade de validar o VTD quanto a sua qualidade na extração dos termos provenientes das regras sintáticas, outros experimentos foram realizados, como será visto nas seções 5.3 e 5.4.

5.3 Segundo Experimento - Classificação com o *Intellexer Categorizer*

Neste experimento, foi utilizado o classificador *Intellexer Categorizer*[16], por ser um sistema já testado e ter um ambiente de fácil configuração.

Inicialmente foram criadas as categorias, e adicionada a cada categoria a mesma coleção de documentos utilizada para criar os VTD's. Para realizar a classificação foram utilizados 14 documentos que pertecem ao mesmo conjunto verdade do experimento anterior.

A maneira como o categorizador cria as regras para a extração de termos dos documentos, como ele indexa os documentos, qual o método para calcular os pesos dos termos, enfim, qual método ele utiliza para calcular a similaridade entre termos e documentos não é claro, por se tratar de um software proprietário.

O nosso propósito foi de confrontar os resultados desta classificação com os resultados da classificação utilizando os VTD's, com o objetivo de verificar a qualidade do mesmo.

---

[16] http://categorizer.intellexer.com/

Os resultados obtidos pelo classificador *Intellexer Categorizer* são mostrados parcialmente para alguns domínios como, *Agricultura*, *Inteligência Artificial*, *Artes*, *História*, *Literatura* apresentando bons resultados (Figura 9). Classificou de 14 documentos, 8 corretos e 6 errados.

Domínios					
Documentos	**Agriculture**	**Inteligência Artificial**	**Arts**	**History**	**Literature**
Accelerated processing unit (CA)	40,29	41,69	31,11	25,2	35,78
Affective Computing (IA)	52,10	**67,35**	41,04	36,38	47,61
Algorithm Characterizations (Algorithm)	39,87	58,6	35,69	30,09	44,56
Anachronism (History)	47,08	51,99	50,28	53,75	**59,39**
Antiquarian (History)	35,66	36,85	39,38	**60,81**	49,19
Bidirectionlization (DB)	19,64	**25,13**	14,68	18,96	19,66
Classic Book (Literature)	34,40	38,92	39,98	41,38	**63,55**
Graph Database (DB)	24,92	33,31	16,56	15,21	21,84
History of Computer Science (CS))	37,86	56,28	36,95	**59,97**	41,63
Hystory of Mathematics (Mathematics)	46,24	52,9	43,79	**62,37**	54,98
Indeterminacy (Literature)	24,26	31,49	26,94	24,85	**49,57**
Outline of Computer Science (CS)	40,77	**66,2**	33,02	31,38	36,51
Potencial theory (Physics)	28,05	40,51	25,48	26,6	30,29
The arts and politics (Arts)	30,50	33,25	**68,47**	0,96	22,29

Figura 9 - Resultados da categorização realizada pelo Intellexer Categorizer

5.4 Terceiro Experimento - Geração dos vetores de termos de documentos com o Weka

Neste terceiro experimento, usamos a ferramenta *Weka*[17]. que oferece uma coleção de algoritmos de aprendizagem de máquina para tarefas de mineração de dados. A ferramenta contém algoritmos para pré-processamento de dados, classificação, clusterização, entre outras tarefas. O *Weka* foi escolhido unicamente para criar um vetor de termos a partir da mesma coleção de documentos que gerou os VTD's utilizando algoritmos de pré-processamento e extração de termos. Foi levada em consideração a

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

facilidade de configuração de alguns parâmetros, com o objetivo de deixar o ambiente o mais próximo possível do *RISOExtrator*. O vetor foi denominado *VetorWeka*, e o objetivo foi utilizá-lo como base de consulta para a classificação de novos documentos usando o classificador *SVSim*.

Para isto, a coleção de documentos passou pelo processo de extração de termos, o algoritmo escolhido foi o *StringToWordVector*, o qual faz a tokenização, contagem e extração dos termos, gerando o vetor de termos. As regras de extração desse algoritmo não são explícitas.

Os ajustes de alguns parâmetros tornaram possível a semelhança na criação dos vetores de termos, como: quantidade de *tokens* para formação dos termos variando entre 1 a 4 *tokens*, formação de palavras compostas, cálculo da frequência de palavras dentro do documento e eliminação dos radicais das palavras. Contudo, os vetores criados com essa extração retornam muitas palavras sem relevância como, por exemplo: letras (A, I, T), números (9, 3, 22), símbolos (♦,♥,•) entre outros.

Estes vetores foram criados com o objetivo de confrontar os VTD's, sendo utilizados como base para classificar o mesmo conjunto verdade de documentos. Assim, foi possível validar a qualidade do VTD, comparando os resultados da classificação. Para o cálculo da similaridade foi utilizado o *SVSim*.

O propósito da observação dos resultados não é para medir o melhor classificador e sim, a melhor base de classificação, comprovando a qualidade dos Vetores Temáticos de Domínio (VTD).

A Tabela 11 mostra a saída da classificação através do *SVSim*, utilizando os vetores criados no *Weka*, servindo como base de consulta para comparação da similaridade com o 14 documentos pertencentes ao mesmo conjunto verdade, como está sendo mostrado na Tabela 12. O resultado apresentado mostra a classificação no domínio predominante (em cinza escuro) e no domínio secundário (em cinza claro). Classificou dos 14 documentos, 9 corretos e 5 errados.

Tabela 11 - Resultados da classificação dos novos documentos a partir do vetor gerado pelo Weka

Documentos	Antiquarian	Graph Database	Accelerated processing unit	The arts and politics	Hystory of Mathematics	Affective Computing	Bidirectionli zation
Domínios							
Agriculture	0,62	0,65	0,28	0,42	0,70	0,81	0,41
Algorithms and Data Structure	0,00	0,00	0,00	0,00	0	0	-
Art	5,25	2,67	0,82	28,44	3,85	5,23	0,09
Artificial Intelligence	2,65	3,46	0,82	1,91	4,54	10,60	1,18
Computacional Science	2,12	5,21	1,46	1,19	6,38	7,15	0,68
Computer Architeture	0,86	5,18	3,68	0,57	2,39	3,74	0,26
Computer Security	1,56	6,61	1,48	2,14	2,18	4,84	0,27
Database	1,53	28,90	1,89	2,17	2,87	4,75	0,44
History	26,63	1,03	0,51	4,02	20,56	2,27	0,10
Literature	8,17	1,32	0,75	5,17	6,28	3,85	0,60
Physics	1,37	1,28	0,42	0,92	5,47	4,45	0,06

A métrica utilizada para o cálculo de similaridade mostrado na Tabela 11, é o cosseno entre o *VetorWeka* e o VTD.

A comparação dos três resultados foi realizada através de uma análise estatística que comprovou a qualidade dos vetores verificando qual o vetor que proporcionou ao classificador obter uma melhor classificação. Portanto, foram realizados estudos experimentais com o objetivo de avaliar o trabalho realizado através de uma classificação de documentos usando a base criada para validar sua qualidade.

Para o presente trabalho a metodologia de avaliação utilizada foi um Experimento.

5.5 Roteiro do Estudo Experimental

Este trabalho segue um roteiro do estudo experimental de uma classificação de documentos digitais que foi realizada usando a mesma base de consulta, de três maneiras diferentes de criação, com o objetivo de validar o nosso método de criação do VTD e a sua qualidade.

Definição:

1) **Tema do experimento:** classificação de documentos usando bases de classificação diferentes.

2) **Área de estudo:** recuperação de Informação

3) **O problema:** falta de vocabulários específicos para domínios

4) **A importância do problema:** automatizar a classificação de documentos propiciando uma melhoria na acurácia da indexação devido à utilização dos vetores temáticos.

5) **O objetivo:** analisar os resultados fornecidos na classificação para verificação da melhor qualidade do VTD.

6) **Hipótese Nula:** todas as classificações são semelhantes.

7) **Hipótese Alternativa:** as classificações não são semelhantes quanto ao resultado da classificação, ou seja, um vetor é melhor do que o outro.

5.5.1 Planejamento e Design

Nesta seção, são detalhados os experimentos que utilizam a mesma base de consulta que foram criadas de maneiras diferentes.

- ✓ Definir três classificadores de documentos:
  - *SVSim* usando como base de consulta os *VTD's* - Sistema Vetorial de Similaridade (SVSim) foi implementado baseado no Space Vetorial Model (SVM) de Salton para ser usado para validar as bases de consulta.
  - *Intellexer Categorizer* - Classificador proprietário *usando a coleção de do*cumentos utiliada para criação dos VTD's
  - *SVSim* - usando como base de consulta o *VetorWeka: C*lassificador *SVSim* utilizando a base de consulta gerada pelo *Weka.*
- ✓ Estabelecer as fases do processo de criação dos Vetores Temáticos de Domínio.
  - Fase 1- Coleta dos documentos na Wikipédia por categoria, para formação dos vetores temáticos
  - Fase 2 - Pré-processamento, análise morfossintática e extração dos termos
  - Fase 3 - Junção dos Vetores
  - Fase 4 - Cálculo do peso dos termos
- ✓ Coletar os documentos novos para classificação

Coleta aleatória de documentos com a categoria conhecida, para formação de um conjunto verdade, a fim de serem submetidos à classificação, como é mostrado na Tabela 12.

Tabela 12- Tabela dos documentos para serem classificados com seus respectivos domínios

Documentos para serem classificados	Domínio
Affective Computing	Artificial Intelligence
Potencial theory	Physics
Outline of Computer Science	Computational Science
History of Computer Science	Computational Science
Indeterminacy	Literature
Classic Book	Literature
Anachronism	History
Antiquarion	History
Bidirectionlization	Database
Graph Database	Database
Algorithm Characterizations	Algorithms
Accelerated processing unit	Computer Architecture
The Arts and Politics	Arts
History of Mathematics	Mathematics

- ✓ Analisar os resultados obtidos;
  - o Validade de conclusão
- ✓ Validação Comparativa
  - o Validação comparativa com o conjunto verdade escolhido.

5.5.2 Seleção de Variáveis

- ✓ Seleção das variáveis independentes: os documentos
- ✓ Seleção das variáveis dependentes: o cosseno de similaridade encontrado entre os VTD's e o vetor do novo documento, indicado a similaridade entre eles.
- ✓ Seleção de Unidades Experimentais: *Weka, SVSim, Intellexer Categorizer.*

5.5.3 Cálculo de Medidas de Desempenho para Comparação dos Resultados da Classificação.

A observação dos resultados obtidos na classificação consiste na comparação das classes reais x classe preditas, que são respectivamente, as classes verdadeiras comparadas com as respostas propostas pelo classificador. Esta comparação deu-se entre os resultados da classificação dos documentos e do conjunto verdade, documentos escolhidos aleatoriamente na Wikipédia para compor a coleção dos documentos com a finalidade de serem classificados, sabendo-se de antemão o domínio. A partir do resultado fornecido pelo classificador, mostrando a similaridade entre o documento e todos os domínios que fazem parte da base de conhecimento, criou-se uma matriz quadrada, em que as colunas representam os domínios propostos pelo algoritmo de classificação (classes preditas), e as linhas representam os domínios reais (classes reais). Esta matriz é chamada Matriz de Confusão, onde a diagonal principal da matriz de confusão representa a quantidade de documentos classificados corretamente e a diagonal secundária representa os erros de classificação.

Para fazer a análise desses resultados, fazemos uso da Matriz de Confusão que fornece a medida da efetividade do modelo de classificação, mostrando o número de classificações reais x classificações preditas em cada classe.

Para criação dessa matriz, foi analisado cada resultado de similaridade que o classificador forneceu, indicando a classe predita. Como são mostradas na Tabela 13, as classes reais e as classes proposta pelo classificador (classe predita).

Tabela 13 - Classes reais e as classes preditas

Classe real (a classe rotulada)		**Classe predita (a classe que o documento será classificado)**		**Razão do erro na classe**
		C+	**C-**	
Verdadeiro	C_+	TP	FN	FN/(TP+FN)
Falso	C_-	FP	TN	TP/(FP+TN)
				Erro Total = (FP+FN)/n

Onde temos:

TP - Documento predito correto e classificado correto

TN - Documento predito errado e classificado errado

FP - Documento não predito e classificado correto

FN - Documento não predito e classificado errado.

Para conhecer a qualidade de cada vetor servindo como base de classificação, utilizamos algumas medidas de desempenho obtidas através da Matriz de Confusão.

Taxa de acerto (Acurácia total)	Ac = (TP+ TN)/(TP+TN+FP+FN)
Sensibilidade ou TPR (*recall*)	Sen = TP/(TP+FN)
FPR (Taxa de Falso Positivo)	FPR = FP/(TN+FP)
Especificidade (Precisão)	Esp = TN/ (FP + TN)
F-measure	F= 2*(*precison***recall*) / (*precision*+*recall*)

Tabela 14- Resultados da classificação dos documentos a esquerda da tabela, utilizando os três vetores propostos

Classificação de Documentos			
	Resultado da classificação utilizando os três métodos		
Experimentos	***SVSim***	***Intellexer Categorizer***	***Weka***
Affective Computing	Classificou correto	Classificou correto	Classificou correto
Potencial theory	Classificou correto	Classificou correto	Classificou correto
Outline of Computer Science	**Classificou errado**	**Classificou errado**	**Classificou errado**
History of Computer Science	**Classificou errado**	**Classificou errado**	**Classificou errado**
Indeterminacy	Classificou correto	Classificou correto	Classificou correto
Classic Book	Classificou correto	Classificou correto	Classificou correto
Anachronism	Classificou correto	**Classificou errado**	Classificou correto
Antiquarion	Classificou correto	Classificou correto	Classificou correto
Bidirectionalization	Classificou correto	**Classificou errado**	**Classificou errado**
Graph Database	Classificou correto	Classificou correto	Classificou correto
Algorithm Characterizations	**Classificou errado**	**Classificou errado**	**Classificou errado**
Accelerated processing unit	Classificou correto	Classificou correto	Classificou correto
The Arts and Politics	Classificou correto	Classificou correto	Classificou correto
Hystory of Mathematics	**Classificou errado**	**Classificou errado**	**Classificou errado**

5.5.4 Preparação

Para realizar um experimento alguns preparativos devem ser cautelosamente analisados antes e durante a execução do experimento:

- ✓ Treinamento;
  - o Extrair termos a partir de regras gerando um VTD para cada domínio
- ✓ Cálculo do peso;
  - o Calcular os pesos dos termos baseado na frequência relativa do termo ao domínio ao qual ele pertence e a frequência global em relação aos domínios que o contém.
- ✓ Configuração do ambiente;
  - o No *Weka* foi usado o algoritmo *StringToWordVetor*, para fazer a extração de termos.O algoritmo foi configurado para contar a frequência das palavras e número de *tokens* para formação do novo termo (n>=5).
  - o No *Intellexer Categorizer* não *houve configuraçõe*s

5.5.5 Metodologia de Execução

Para a execução de um processo experimental válido, alguns princípios são fundamentais: organização, controle, acompanhamento, medições, análise e interpretação dos dados. Visando facilitar a elaboração dos experimentos, várias metodologias foram desenvolvidas, cada uma com suas peculiaridades em relação às fases, aos objetivos, às ferramentas de empacotamento, às métricas, entre outras diferenças (Travassos, 2002).

Portanto, o experimento foi desenvolvido de acordo com as metodologias exemplificadas abaixo:

Utilizando o classificador *SVSim* - Base de consulta: VTD's

1. Cálculo do peso tf-idf dos VTD's
2. Processamento de extração de termos do novo documento usando o *MontyLingua*.
3. Cálculo do peso tf-idf do novo documento.
4. Cálculo da similaridade entre o vetor do documento novo e os VTD's

5. Resultado da classificação do novo documento, através do cosseno de similaridade, indicando o domínio mais similar o que obtiver o maior peso
6. Comparação dos resultados da classificação com o conjunto verdade.

Utilizando o classificador *Intellexer Categorizer*

1. Criar os domínios no *Intellexer Categorizer.*
2. Alimentação de cada domínio com a coleção de documentos usados para criação dos VTD's, respectivamente.
3. Alimentação do novo documento.
4. Categorização do novo documento.
5. Resultado da classificação do novo documento, mostrando a qual domínio ele pertence.
7. Comparação dos resultados da classificação com o conjunto verdade.

Utilizando o classificador *SVSim* - Base de consulta: *VetorWeka*

1. Ajuste dos parâmetros do processo de indexação do *Weka.*
2. Criação de vetores de termos para cada domínio através do software *Weka(VetorWeka)*
3. Geração do vetor de termos do novo documento pelo *MontyLingua.*
4. Cálculo do peso tf-idf do novo documento
5. Cálculo da similaridade entre o vetor do documento novo e o *VetorWeka.*
6. Resposta da classificação do novo documento, mostrando a qual domínio ele pertence através do cosseno de similaridade.
8. Comparação dos resultados da classificação com o conjunto verdade.

5.5.6 Cálculos Das Medidas De Desempenho

As medidas de desempenho são utilizadas para medir um bom classificador, são elas: cálculo da acurácia, sensitividade (*recall*), especificidade e *f-measure*, utilizando a Matriz de confusão gerada com a saída da classificação. Assim temos de acordo com as Tabelas 17 a 24 dos Apêndices D, E, e F, os resultados numéricos da classificação.

1. Resultado da classificação com o resultado da classificação com o *SVSim* utilizando como base de consulta os VTD's.

TP= 8	FN=2
FP= 2	TN=2

Acurácia	TP+TN/(TP+TN+FP+FN)	0,80
Especificidade (Precisão)	TN /(TN +FP)	0,50
Sensitividade (*Recall*)	TP/(TP+FN)	0,80
F-Measure	2*(precisão**recall*/(precisão+*recall*))	0,80
FPR	FP/(FP+TN)	0,50

2. Resultado da classificação com o *Intellexer Categorizer.*

Observação: Os documentos que obtiveram um resultado abaixo de 50% de similaridade não foram categorizados, ou seja, não houve uma classe predita, contudo podem ser classificados corretamente condizentes com a classe real. A partir daí, temos:

TP = 6	FN=2
FP = 4	TN=2

Acurácia	TP+TN/(TP+TN+FP+FN)	0,57
Especificidade (Precisão)	TN /(TN +FP)	0,33
Sensitividade (*Recall*)	TP/(TP+FN)	0,75
F-Measure	2*(precisão**recall*/(precisão+*recall*))	0,67
FPR	FP/(FP+TN)	0,67

3. Resultado da classificação com o *SVSim* utilizando como base de consulta os vetores gerados pelo *Weka* (*VetorWeka*)

TP = 6	FN = 4
FP = 3	TN =1

Acurácia	TP+TN/(TP+TN+FP+FN)	0,50
Especificidade (Precisão)	TN /(TN +FP)	0,25
Sensitividade(Recall)	TP/(TP+FN)	0,60
F-Measure	2*(precisão**recall*/(precisão+*recall*))	0,63
FPR	FP/(FP+TN)	0,75

Fazendo um resumo dos resultados calculados para cada classificador, mostrado na tabela 15.

Tabela 15 - Resultados finais dos três experimentos.

Medidas de desempenho	*SVSim* (VTD)	*Intellexer*	*SVSim (VetorWeka)*
Acurácia	0,80	0,57	0,50
Especificidade (Precisão)	0,50	0,33	0,25
Sensitividade(*Recall*)	0,80	0,75	0,60
F-Measure	0,80	0,67	0,63
FPR	0,50	0,67	0,75

Analisando os resultados obtidos a partir dos três experimentos, através das medidas de desempenho, temos como saber qual o classificador que obteve a melhor resposta de acordo com a base de consulta que está utilizando, Os três classificadores não estão sendo analisados e sim, a base que cada um está utilizando. Portanto, o VTD mostrou ser uma base com melhores termos, permitindo que o classificador obtivesse um resultado satisfatório na predição corretas das classes.

Quando sensitividade é alta, mais documentos são classificados corretamente, combinando a classe real e a classe predita pelo classificador. Quando a especificidade é alta, diminui os documentos classificados erroneamente, pois pertencem a uma determinada classe e foram classificados em outra.

A Taxa de Falsos Positivos (FPR) mede a porcentagem de amostras **erroneamente classificadas como positivas** dentre todas as **negativas reais,**

Portanto, baseado nessas medidas de desempenho, o primeiro experimento *SVSim*/VTD apresentou uma acurácia de 80%, significando que os documentos foram classificados na classe correta, mostrou também uma sensitividade alta, da mesma forma, mostra que mais documentos foram classificados corretamente, mostrados na Tabela 14 (p.66), e com casos positivos. Uma sensitividade alta apresenta o erro tipo II baixo, significando que a hipótese nula, que diz que os classificadores são iguais, é falsa. Uma vez a hipótese nula refutada, a hipótese alternativa diz que os classificadores não são semelhantes quanto a qualidade da classificação, isto é, um classificador é melhor que outro. Então, baseados nas medidas de desempenho mostrando que *f-measure*=0,80 e com uma cobertura=80% aceita a hipótese alternativa que os classificadores não são semelhantes, mas um melhor que o outro. Baseados nos resultados encontrados na classificação efetuada pelo *SVSim* utilizando como base de consulta os VTD's com f-

measure=0,8 e recall=0,80, mostra que os VTD's cobriram uma maior quantidade de termos específicos promovendo uma melhor classificação dos documentos contra os resultados do *Intellexer Categorizer* com *f-measure*= 0,67 e do *SVSim* utilizando como base o Vetor*Weka* com *f-measure*=0,63.

Capítulo 6
Conclusões e Trabalhos Futuros

A ambiguidade dos termos existentes em um texto é um dos principais fatores que dificultam uma melhor qualidade na recuperação da informação desejada por um usuário. O presente trabalho se baseia na hipótese de que parte desta ambiguidade poderia ser eliminada sabendo-se de antemão o domínio de um documento a ser indexado contendo termos ambíguos. Para determinar este domínio propõe-se construir vocabulários típicos dos diversos domínios do conhecimento e comparar o conjunto de termos contidos em um documento com estes vocabulários. O que estiver mais próximo define o domínio do documento.

Para alcançar o objetivo de construir estes vocabulários representativos, chamados de vetores temáticos, foram utilizados como entradas, documentos retirados da Wikipédia correspondentes aos domínios pré-escolhidos. Para tornar a extração dos termos mais efetiva foram efetuadas análises e transformações dos textos originais. De um lado foram eliminados sinais sem significado, *stopwords* e outras palavras não representativas. Por outro lado, foram feitas cuidadosas análises sintáticas das frases para determinar até que ponto palavras auxiliares, como conjunções, preposições, adjetivos e verbos, determinam termos compostos significativos. Foram criadas diversas regras para determinar os termos compostos para criação de um vocabulário mais rico.

O trabalho foi validado utilizando documentos com domínio previamente conhecido, conjunto verdade, e analisando a classificação realizada utilizando os vetores temáticos. Para efetuar a validação do trabalho, foram realizados três tipos de experimentos. Primeiro, foi usado o modelo vetorial para determinação de similaridades entre textos, para determinar a eficácia dos VTD's na classificação de 41 documentos. Para os experimentos citados a seguir, foram escolhidos 14 dos 41 documentos classificados no primeiro experimento, permitindo fazer uma comparação dos três experimentos. Como segundo experimento foi utilizado o sistema de classificação *Intellexer Categorizer* e, no terceiro, foi criado vetores de termos pelo software Weka.

Observando os resultados dos três experimentos, na Tabela 15, podemos observar a diferença entre os valores calculados pelo *SVSim* e o classificador *Intellexer Categorizer*, isso ocorre devido à normalização que é realizada na frequência do termo (*tf*), para

minimizar o problema da diferença de tamanho dos vetores. Contudo, o resultado final foi muito aproximado. Outra observação é a diferença existente entre o valor de similaridade entre os domínios, onde os domínios que *SVSim* prediz como positivo verdadeiro e entre os que não são verdadeiros, mostra uma significativa diferença em percentual, apresentando claramente o domínio predominante. Por exemplo, na Tabela 10 (p. 61), para o documento *Antiquarian*, o primeiro domínio classificado obteve 26,63% de similaridade, o segundo domínio obteve 8,17%. Obtendo um percentual de 30,67% de diferença entre os dois domínios o que ocorreu na maioria dos documentos.

Tabela 16- Tabela com todos os valores de similaridade dos três experimentos.

Documentos novos	Vetores/ Classificador		
	VSM /VTD	Intellexer	*VSM/Vetor Weka*
Accelerated processing unit	1,79	45,79	3,68
Affective Computing	7,70	67,30	10,60
Algorithm Characterizations	3,97	63,74	10,44
Anachronism (history)	8,44	59,39	12,77
Antiquarian	10,30	60,81	26,63
Bidirectionlization	0,59	25,13	1,18
Classic Book	6,75	63,55	18,38
Graph Database	10,93	62,06	28,90
History of Computer Science	3,01	59,97	16,13
Hystory of Mathematics	4,05	62,37	20,56
Indeterminacy	2,41	49,57	9,13
Outline of Computer Science	6,53	66,20	11,44
Potencial theory	3,41	51,55	5,48
The arts and politics	7,76	68,47	28,44
Total de documentos = 14			
Classificados corretamente	10	8	9
Classificados errados	4	6	5

Diante dos resultados aqui mostrados na Tabela 16, houve uma proximidade nas três classificações, contudo o *SVSim/VTD*, utilizando os VTD's como base de consulta, apresentou uma melhor desempenho, no total de 14 documentos ele classifica 10 documentos corretamente e 4 documentos negativos, quer dizer o domínio predito pelo classificador não é o domínio real. Em segundo lugar, veio o *SVSim/VetorWeka,* utilizando a base de consulta *VetorWeka*, classificando 9 documentos no domínio correto e 5 documentos negativos, em último lugar, o *Intellexer Categorizer*, que classifica 8 documentos corretamente e 6 documentos no domínio errado, de acordo com o conjunto verdade obtido.

Todos os três apresentaram resultados satisfatórios, mas o que obteve maior desempenho quanto às classificações, foi o processo proposto nesta dissertação, utilizando

como base de consulta os VTD's. Apresentou uma acurácia de 80% contra, 57% e 50% dos outros dois experimentos.

Como trabalhos futuros a primeira tarefa será a geração um número bem mais amplo de vetores temáticos dos domínios do conhecimento. A princípio todas as principais classes da CDU (Apêndice A) deverão ser consideradas. Só assim o sistema poderá ser utilizado na prática.

A criação dos vetores temáticos poderá ser melhorada em vários sentidos. Uma análise lingüística mais detalhada dos textos poderá incrementar a qualidade dos termos extraídos. Por exemplo, a detecção de pronomes e sua substituição pelos termos representados poderá apurar mais a contagem da freqüência de termos. Como em um idioma a linguagem utilizada sofre uma evolução contínua, o processo de geração dos vetores deverá ser atualizado considerando documentos recentes.

No âmbito do projeto RISO a desambiguação por domínios será complementada por uma desambiguação contextual dos termos baseada na proximidade de outros termos em uma frase e na identificação de referências espaço-temporais. Com isso espera-se converter os termos contidos em um documento em **conceitos**.

Após a detecção dos conceitos a identificação de sinônimos e outras relações semânticas entre conceitos será a próxima etapa do processo de indexação semântica de textos com o objetivo de aumentar a precisão e cobertura em um ambiente de recuperação da informação.

Referências Bibliográficas

Baeza-Yates, R. (1996). *An extended model for full text databases. Journal of the Brazilian Computer Society*, V.2.n.3, Abril de 1996.

Baeza-Yates, R., Ribeiro-Neto, B.(1999). *Modern Information Retrieval*. Addison Wesley.

Bidermann, M.T.C. (2001). Teoria Linguística. 2. Ed. São Paulo: Martins Fontes.

Brill, E. (1994). Some Advances in Transformation-Based Part of Speech Tagging. In Proceedings of the twelfth national conference on artificial intelligence. Seattle, Wa. American Association for Artificial Intelligence.

Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. Computational Linguistics 21(4), 543–565.

Cardoso, O. N. P. (2002). Recuperação de Informação. Departamento de Ciência da Computação- Universidade Federal de Lavras, Lavras, p. 6.

Cross, V. (1994). Fuzzy information retrieval. Journal of Intelligent Information Systems, **v.5.**

Chen, L. (2006). Automatic construction of domain-specific concept structures. Technischen Universitat Darmstadt.

Denning, P. J. (2005). Is Computer Science Science? In: ACM Communication April.

Earl, L. L. (1970). Experiments in automatic abstracting and indexing. Information Storage and Retrieval, 6 (4), pp. 313-334.

Ferneda, E. (2012) Introdução aos Modelos Computacionais de Recuperação de Informação. Rio de Janeiro: Editora Ciência Moderna Ltda.

Gaizauskas, R., Wilks Y. (1998). "Information extraction: beyond document retrieval", Journal of Documentation, Vol. 54 Iss: 1, pp.70- 105

Galho, T. S., Moraes, S. M. W. (2004). Categorização Automática de Documentos de Texto Utilizando Lógica Difusa, Universidade Luterana do Brasil (ULBRA) – Gravataí, RS, Brasil.

Gonçalves, R.J.A. (2010). Extracção de Referências Bibliográficas. 2010. 86 p. Dissertação (Mestrado em Engenharia Informática e de Computadores) – Programa de Pós-Graduação Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa.

Gonzalez, M., Lima, V. L. S. (2003). Recuperação de Informação e Processamento da Linguagem Natural. XXIII Congresso da Sociedade Brasileira de Computação, Campinas, 2003. Anais do III Jornada de Mini-Cursos de Inteligência Artificial, Volume III, p.347-395.

Green, G. M. (1995).Ambiguity resolution and discourse interpretation.In: Semantic Ambiguity and Underspecification. Kees van Deemter and Stanley Peters, editors.CSLI Publications, Stanford.

Grishman, R. (1997).Information Extraction: techniques and challenges. In: International Summer School on Information Extraction, SCIE, 1997, Frascati, It. Information Extraction: a multidisciplinary approach to an emerging information technology. Berlin: Springer Verlag. (Lecture Notes in Artificial Inteligence, v. 1299).

Ingwersen, P. (1996). Congnitive perspectives of information retrieval interaction: elements of a cognitive IR theory. Journal of Documentation, v.52, n.1.

Kerlinger, C., Taylor, R.(1979) Marketing research: an applied approach. Tóquio: McGraw-Hill, Kogakusha.

Kittur, A., Kraut, R. E. (2008)."Harnessing the wisdom of crowds in wikipedia: quality through coordination". In: ACM Conference on Computer-Supported Cooperative Work (CSCW 2008).

Lewis, D. D., Sparck-Jones, K. (1996*).Natural Language processing for information retrieval.* Communications of the ACM, v.39, n.1, Janeiro de 1996.

Loh, S., Wives, L.K., Franeir, A.S.(1997). "Uma abordagem para a Busca Contextual de Documentos na Internet", v4, p.79-92.

Lopes, L., Vieira, R., Martins, D. (2010). Hierarquias de conceitos extraídas automaticamente de corpus de domínio específico - Um experimento sobre um corpus de Pediatria. In: XII Congresso Brasileiro de Informática em Saúde CBIS, 2010, Recife. Anais do XII CBIS.

Lopes, L., Fernandes, P., Vieira, R., Fedrizzi, G. & Martins D. (2010). ExATOLP – a tool for domain relevant terms extraction. In: 9th International Conference on Computational Processing of Portuguese Language, PROPOR 2010, Porto Alegre, RS.

Lopes, L.; Oliveira, L.; Vieira, R. (2010)."Portuguese Term Extraction Methods: Comparing Linguistic and Statistical Approaches". In: PROPOR 2010 - International Conference on Computational Processing of Portuguese Language

Maarek, Y. S. (1992).Automatically constructing simple help systems from natural language documentation, in Jacobs, P. S. (1992). Text-based intelligent systems: current research and practice in information extraction and retrieval. New Jersey: Lawrence Erlbaum, 1992.

Marcus, M., Santorini, B., Marcinkiewicz,M. A. (1993). Building a large annotated corpus of English: The Penn Treebank.In Computational Linguistics, volume 19, number 2, pp313-330.

Matos, P.F. (2010). Metodologia de Pré-processamento Textual para Extração de Informação sobre efeitos de Doenças em Artigos Científicos do Domínio Biomédico. 2010. 161 p. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal São Carlos, São Carlos.

Melro, M. C. (2006). "A Classificação Decimal Universal (CDU): uma prática na Biblioteca da UFP", Revista da Faculdade de Ciências Humanas e Sociais, 3: 101 - 109.

Miller, G. A.(1995). "*WordNet: A lexical database for English*". Communications of the ACM, vol 38-11, novembro 1995, pp.39-41.

Monnerat, R. S. M. (2003), A ambiguidade e o emprego de pronomes. p. 4. 2003. Disponível em: <http://www.filologia.org.br/viiicnlf/anais/caderno13-01.html>

Magennis, M., van Rijsbergen, C. J. (1997). The potential and actual effectiveness of interactive query expansion. In: BELKIN, Nicholas J. et alli (eds). Proceedings XX

International ACM SIGIR Conference on Research and Development in Information Retrieval. Proceedings... Philadelphia: ACM Press.

MontyLingua – A FREE, Commonsense-Enriched Natural Language Understander for English.Disponível em: <http://web.media.mit.edu/~hugo/montylingua/>

Peres, J.A., Móia, T. (1995), Língua, comunidade linguística, variação e mudança. Áreas Críticas da Língua Portuguesa, Ed. Caminho, Lisboa, (pp.34-41).

Pfleeger, L.(1999) Albert Einstein and Empirical Software Engineering. In: IEEE Computer, V. 32.

Resende , S. O., (2005) Sistemas inteligentes: fundamentos e aplicações. Barueri, SP. Malone

Riloff, E., Lehnert, W. (1994). Information extraction as a basis for high-precision text classification. ACM Transactions on Information Systems, v.12, n.3.

Riloff, E., Hollar, L. (1996). "Text Database and Information Retrieval", ACM computing surveys, Vol. 28, No. 1, March 1996 , pp. 133-135.

Salton, G. (1968).Automated Language Processing, Annual Review of Information Science and Technology, Vol. 3, Pag.169-199.

Salton, G. (1983). Introduction to Modern Information Retrieval. New York: McGraw-Hill.

Salton, G. (1984). The use of extended Boolean logic in information retrieval, Technical Report TR 84-588, Cornell University, Computer Science Dept. Ithaca, N.Y.

Salton, G., Buckley, C. (1988). Term-weighting Approaches in Automatic Text Retrieval, Information and Processing Management, v 24, n. 5, p.513-523.

Salton, G., Buckley, C. (1997). Term-weighting Approaches in Automatic Text Retrieval. In Sparck-Jones, K., Willet, P. (eds). Readings in Information Retrieval.San Francisco: Morgan Kaufmann.

Scarinci, R. G. (2007). "SES – Sistema de extração Semântica de Informações", Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

Schönhofen, P., Benczúr, A., Bíró, I., Csalogány, K. (2007).Performing Cross-Language Retrieval with Wikipedia, in CLEF.

Sparck-Jones, K., Willet, P. (1997). (eds). *Readings in Information Retrieval.* San Francisco: Morgan Kaufmann.

Saracevic, T. (1997). Relevance: a review of and a framework for the thinking on the notion in information science. In: Sparck-Jones, K., Willet, P. (1997). (eds). Readings in Information Retrieval. San Francisco: Morgan Kaufmann

Sparck-Jones, Karen et al. (1997b). Experiments in spoken document retrieval. In: Sparck-Jones, K., Willet, P. (1997). (eds). Readings in Information Retrieval. San Francisco: Morgan Kaufmann

Spinellis, D., Louridas, P. (2008)."The collaborative organization of knowledge",Communications of the ACM, vol. 51-8, agosto 2008, pp.68-73.

Sarawagi, S. (2008).Information Extraction, Foundations and Trends in Databases 2(1), Texto apresentado no seminário sobre Theodor Adorno, realizado no Instituto Goethe.

Syed, Z.; Finin, T., Joshi, A. (2008) "*Wikipedia as an Ontology for Describing Documents*" in Proceedings of the Second International Conference on Weblogs and Social Media. AAAI Press.

Souza, S. (2004). CDU: como entender e utilizar a edição-padrão internacional em língua portuguesa, Brasília: Thesaurus.

Travassos, G., Gurov, D., Amaral, E.(2002) Introdução à Engenharia de Software Experimental. Relatório Técnico ES 590/02. Rio de Janeiro, PESC/COPPE/UFRJ.

Van Rijsbergen, C. J. (1979). *Information retrieval*,2.ed. London: Butterworths, p. 1-35.

Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H., Studer, R. (2006).Semantic Wikipedia,In: 15th International Conference on World Wide Web (WWW2006), pp.585-594.

Wives, L.K.(1997). "Um Estudo sobre Técnicas de recuperação de informações com ênfase em Informações Textuais", Trabalho Individual (Mestrado em Ciência da Computação)– Instituto de Informática, Universidade Federal do Rio Grande do Sul, Posto Alegre.

WIVES, L. (2002). Tecnologias de Descoberta de Conhecimento em Textos aplicadas à Inteligência Competitiva. Porto Alegre, 2002. 100 f. Pós-Graduação em Computação. Universidade Federal do Rio Grande do Sul, Porto Alegre.

Wohlin, C. et. al.(2000) Experimentation in Software Engineering – An Introduction. Massachusets.Kluwer Academic Publishers.

Xavier, C. C., Lima, V. L. S. (2009).Construção de uma Estrutura Ontológica de Domínio a partir da Wikipédia, In: 7th Brazilian Symposium in Information and Human Language Technology (STIL 2009).

Zambenedetti, C.(2002).Extração de Informação sobre Base de Dados Textuais,Dissertação de Mestrado, Programa de Pós-Graduação em Computação, UFRGS, 144p.

Zelkowitz, V., Wallace, R., Binkley, W.(2003). Experimental Validation of New Software Technology, Lecture Notes on Empirical Software Engineering, World Scientific Publishing,

Apêndice A– Tabela das Classes da CDU

0 Generalities			
	000 Computer science, knowledge &systems		
		001 Science and knowledge in general. Organization of intellectual work	
		002 Documentation. Books. Writings. Authorship	
		003 Writing systems and scripts. Including: signs and symbols	
		004 Computer science and technology. Computing	
			004.2 Computer architecture
			004.3 Computer hardware
			004.4 Software
			004.5 Human-computer interaction
			004.6 Data
			004.7 Computer communication
			004.8 Artificial intelligence
			004.9 Application-oriented computer-based techniques
		005 Management (Revision from 2001)	
			005.1 Management Theory
			005.2 Management agents. Mechanisms. Measures
			005.3 Management activities
			005.32 Organizational behaviour. Management psychology
			005.5 Management operations. Direction
			005.6 Quality management. Total quality management
			005.7 Organizational management (OM)
			005.9 Fields of management
			005.92 Records management
			005.93 Plant management. Physical resources management
			005.94 Knowledge management
			005.95/.96 Personnel management. Human Resources
		006 Standardization of products, operations, weights, measures and time	
		007 Activity and organizing. Information. Communication and control	
		008 Civilization. Culture. Progress	
		009 Humanities. Arts subjects in general	
1 Philosophy. Psychology			
	100 Psychology		
2 Religion. Theology			
3 Social Sciences			
	300 Social sciences, sociology & anthropology		
4 Null			
5 Mathematics and natural sciences			
	500 Mathematics		
		520 Astronomy	
		530 Physics	
		540 Chemistry	
6 Applied sciences. Medicine. Technology			

	610 Medicine
	620 Engineering. Technology in general
	630 Agriculture
7 The arts	
	700 Arts
	720 Architecture
	780 Music
	790 Sports, games & entertainment
8 language, linguistics, literature	
	810 Linguistics and languages
	820 Literature
9 geography, biography, history	
	900 History
	910 Geography & travel

Apêndice B- Tabela das categorias da Wikipédia

A tabela abaixo contém as categorias e a quantidade de documentos que compõem o *corpus* de cada categoria.

Concepts			
	Computer science		
		Areas of computer science	
			Algorithms and Data Structures – 3 documentos
			Artificial Intelligence- 200 documentos
			Computational Science – 119 documentos
			Computer Architecture - 57 documentos
			Computer Graphics – 119 documentos
			Computer Security - 200 documentos
			Concurrency - 29 documentos
			Database Theory - 42 documentos
			Formal Methods - 77 documentos
			Human-based computation - 24 documentos
			Human–computer Interaction - 126 documentos
			Mathematical Optimization - 195 documentos
			Programming Language Theory – 20 documentos
			Soft Computing - 3 documentos
			Software Engineering - 52 documentos
			Theoretical Computer Science - 97 documentos
			Theory of Computation - 96 documentos
	Philosophy, Psychology - 29 documentos		
		Psychology- 93 documentos	
	Religion. Theology -55 documentos		
	Social Sciences – 178 documentos		
	Mathematics – 8 documentos		
		Astronomy -6 documentos	
		Physics- 187 documentos	
		Chemistry – 70 documentos	
	Applied Sciences, Medicine, Technology		
		Medicine – 129 documentos	
		Engineering, Technology in general – 106 documentos	
		Agriculture – 158 documentos	
	The Arts		
		Arts - 55 documentos	
		Architecture – 123 documentos	
		Music – 9 documentos	
		Sports, Games & Entertainment – 1 documento	
	Language, Linguistics, Literature		
		Linguistics and Languages - 200 documentos	
		Literature – 77 documentos	
	Geography, Biography, History		
		History – 34 documentos	
		Geography & travel - 110 documentos	

Apêndice C - Heurísticas

Otimização das regras heurísticas, definidas abaixo, baseada na programação lógica utilizando técnica de recursividade.

Para definição das regras foi utilizado o padrão BNF ((Backus Normal Form)[18] é muito usada como notação para representar partes de gramáticas de linguagens naturais.

As regras BNF foram escritas com base na terminologia do conjunto de tags que representam as categorias gramaticais baseadas no Penn Treebank Tagset, como mostra na Tabela 4 no Capítulo 4.

Padrão:

<termo> ::= <termo-1> | <termo-2>| <termo-3>

<tipo-termo> ::= <tipo-subst>| <tipo-adj> | <tipo-verbo>

<oper> ::= <preposição> | <separador>

< preposição> ::= "of" | "in"

<separador> ::= "and" | "or"

<tipo-subst> ::= "NN" | "NNS" | "NNP" | "NNPS"

<tipo-adj> :: = "JJ" | "JJR" | "JJS"

<tipo-verbo> ::= "VB" |"VBG" |"VBN"

Regras de formação de termos:

De acordo com as expressões criadas, não iremos repetir todos os casos, mas sim os casos mais significativos.

Regras para termos gramaticais marcados por tags:

<subst> ::= <palavara> "/" <tipo-subst>

<adj> ::= <palavara> "/" <tipo-adj>

<verbo> ::= <palavara> "/" <tipo-verbo>

Regras dos tipos de termos:

<termo-1> ::= <subst> | <subst><termo-1>

[18] http://pt.wikipedia.org/wiki/Formalismo_de_Backus-Naur

<termo-2>::= <adj><termo-1>

<termo-3>::= <verbo> | <verbo><termo-3>

Regras gerais para formação de termos:

<termo> ::= <termo-1><preposição><termo-2>

<termo> ::= <termo-1><termo-2><preposição><termo-3>

Como exemplo, temos a frase de entrada:

A review of CIO vacancy positions and requirements between April 2010 and May 2010 from the Chronicle of Higher Education.

Após a marcação com *tags*, a frase fica dessa forma:

A/DT review/NN of/IN CIO/NNP vacancy/NN positions/NNS and/CC requirements/NNS between/IN April/NNP and/CC May/NNP from/IN the/DT Chronicle/NNP of/IN Higher/JJR Education/NNP ./.

Neste exemplo os seguintes termos compostos são extraídos da frase de entrada:

- ✓ *review of CIO vacancy positions*
- ✓ *Chronicle of Higher Education*
- ✓ *review*
- ✓ *CIO vacancy positions*
- ✓ *Chronicle*
- ✓ *Higher Education*

Mostraremos agora regras para os diversos casos de extração de termos compostos a partir de frases. Para essas regras será usada a seguinte notação:

SE <condição> ENTÃO <composição> {E <composição-1>} (É TERMO | SÃO TERMOS)

1º Caso: A busca, em uma frase, pela preposição (of) abre a possibilidade de extração de diversos termos compostos.

caso 1:
SE <subst-1> 'of' <subst-2> ENTÃO <susbt-1> 'of' <subst-2> É TERMO

caso 2:
SE <subst-1>< subst-2> 'of' <subst-3> ENTÃO <subst-1> <subst-2> 'of' <subst-3> É TERMO

caso 3:

SE <adj>< subst-1> 'of' <subst-2> ENTÃO <adj> <subst-1> 'of' <subst-2> É TERMO

caso 4:

SE <adj> 'of'< subst> ENTÃO <adj> 'of' <subst> É TERMO

caso 5:

SE <adj-1><subst-1> 'of'<adj-2><subst-2>ENTÃO<adj-1>< subst-1> 'of' <adj-2> <subst-2> É TERMO

caso 6:

SE <subst-próp-1>'of'< subst-próp-2> ENTÃO <subst-próp-1> 'of' <subst-próp-2> É TERMO

caso 7:

SE <subst-próp-1> <subst-próp-2> 'of'< subst> ENTÃO < subst-próp-1> <subst- próp-2 'of' subst> É TERMO

caso 8:

SE <subst-próp-1> <subst-próp-2> 'of' <adj> subst> ENTÃO <subst-próp-1> <subst-próp-2> 'of' <adj> <subst> É TERMO

caso 9:

SE <verbo> <verb-gerun> 'of' <subst> ENTÃO <verbo> <verb-gerun> of <subst> É TERMO

caso 10:

SE <verbo> <subst-1> 'of'<subst-2> ENTÃO <verb> <subst.>. 'of' <subst-2> É TERMO

caso 11:

SE <subst-1><verb-nom> 'of' <subst-2> ENTÃO <subst-1> <verb-nom> 'of' <subst-2> É TERMO

caso 12:

SE <subst-próp-1> 'of' <adj-superl> <subst-próp-2> ENTÃO <subst-próp-1> 'of' adj-superl> <subst-próp-2> É TERMO

caso 13:

SE <subst-próp-1> 'of' <adj-superl><subst-próp-2> ENTÃO <subst-próp-1> 'of' <adj-superl> <subst-próp-2> É TERMO

caso 14:

SE <subst-1> 'of' <verbo-nom> <subst-2> ENTÃO <subst-1> 'of' <verbo-nom> subst-2> É TERMO

caso 15:

SE <subst-1> 'of' <verbo-gerun> <subst-2> ENTÃO <subst-1. 'of' <verbo-gerun> <subst -2> É TERMO

No exemplo são retirados da frase de entrada os termos:

- ✓ review of CIO vacancy positions
- ✓ Chronicle of Higher Education

Com essas heurísticas são formados novos termos com n-gramas de acordo com a classificação dada.

2º Caso: A busca dentro da frase pela conjunção '*and*' abre a possibilidade de extração de vários termos compostos.

Como exemplo, temos como a frase de entrada:

Educause and HigherEdJobs.com.

Após a marcação com *tags*, a frase retorna dessa forma:

Educause/NNP and/CC HigherEdJobs/NNP ./. com/NN ./.

Outro exemplo:

Tropical fish and birds.

Após marcação fica da seguinte forma:

Tropical/JJ fish/NN and/CC birds/NN

caso 1:

SE<adj> <subst-1> 'and' <subst-2> ENTÃO <adj-1> <subst-1> E <adj> <subst-2> SÃO TERMOS

caso 2:

SE<verb> <subst-1> 'and' <subst-2> ENTÃO <verb> <subst-1> E <verb> <subst-2> SÃO TERMOS

caso 3:

SE <subst-1> <verb-nom-1> 'and' <subst-2> ENTÃO <subst-1> <verb-nom-1> E <subst-2> SÃO TERMOS

caso 4:

SE <subst-próp-1>'and' <adjsuperl> <subst-próp-2> ENTÃO <subst-próp-1> E <adj-superl> <subst-próp-2> SÃO TERMOS

caso 5:

SE<subst-próp-1> 'and' <adj-superl-1> <subst-próp-2> ENTÃO <subst-próp-1> E <adjet-superl-1 subst-próp-2> SÃO TERMOS

caso 6:

SE <subst-1> 'and' <verb-nom> <subst-2> ENTÃO <subst-1> E <verb-nom> <subst -2> SÃO TERMOS

caso 7:

SE <subst-1>'and'<verb-gerun> <subst-2> ENTÃO <subst-1> E <verb-gerun> <subst-2> SÃO TERMOS

caso 8:

SE <adj-1> <subst-1> 'and' <verbo-gerun> <subst-2> <subst-3> <subst-4> ENTÃO <subst-1> E <verbo gerun> <subst-2> <subst-3> <subst-4> SÃO TERMOS

Termos são retirados da frase de entrada:

- ✓ *Educause*
- ✓ *HigherEdJobs.com*
- ✓ *Tropical fish*
- ✓ *Tropical birds*

3º Caso: A busca dentro da frase pela conjunção 'or' abre a possibilidade de extração de vários termos compostos.

Como exemplo, temos como a frase de entrada:

*AIS can take days **or** even weeks.*

Após a marcação com *tags*, a frase retorna dessa forma:

*AIS/NNP can/MD take/VB days/NNS **or/CC** even/JJ weeks/NNS ./.*

caso 1:

SE <adj> <subst-1> 'or' <subst-2> ENTÃO <adj> <subst-1> E <adj> <subst-2> SÃO TERMOS

caso 2:

SE <subst-próp-1> <subst-próp-2> 'or' <adjt-1><subst-1> ENTÃO <subst-próp-1 subst-próp-2> E <adjt-1 subst-1> SÃO TERMOS

caso 3:

SE <verb> <verb-gerun>'or' <subst> ENTÃO <verb> <verb-gerun> <subst> É TERMO

caso 4:

SE <verb> <subst-1> 'or' <subst-2> ENTÃO <verb> <subst-1> E <verb> <subst-2> SÃO TERMOS

Termos retirados da frase de entrada, dois termos simples:

- ✓ *days/NNS*
- ✓ *even/JJ weeks/NNS*

4º Caso: Casos diversos

caso 1 :

SE <adj-1> <subst-1> 'ENTÃO <adj-1 subst-1> É TERMO

caso 2:

SE <subst-1> <verb-nom-1> ENTÃO<subst-1> <verb-nom-1> E SÃO TERMOS

caso 3:

SE <adjet-superl> <subst-próp> ENTÃO <adjet-superl> <subst-próp> SÃO TERMOS

caso 4:

SE <verb-nom-1> <subst-1> ENTÃO <verb-nom-1 subst-1> SÃO TERMOS

caso 5:

SE <verbo> <gerun> <subst> ENTÃO <verbo> <gerun> <subst> SÃO TERMOS

5º **Caso:** A busca na frase por preposição ou advérbio determina a extração de mais de um termo simples

caso 1:

SE <subst-1> 'preposição' <subst-2> ENTÃO <subst-1> preposição <subst-2> É TERMO.

caso 2:

SE <subst-1> advérbio <subst-2> ENTÃO <subst-1> advérbio <subst-2> É TERMO

6º Caso: A busca na frase por verbo nominal determina a extração de um termo composto.

caso 1:

SE <verbo-nom> <subst> ENTÃO <verbo nom> subst> É TERMO.

7º Caso: A busca na frase por adjunto superlativo determina a extração de mais de um termo.

8º Caso: A busca na frase por siglas ou acrônimos identificação do seu significado formando um dicionário de conceitos, gerando o termo e o conceito.

caso 1:

SE <termo composto> "("<sigla>")" ENTÃO <sigla> É UM TERMO

Apêndice D - Resultados da classificação de documentos utilizando como base de consulta o VTD (Vetor Temático de Domínios)

Tabela 17 - Referência dos domínios

Domínios	**Referência na tabela de classificação**
Agriculture	**D1**
Algorithms and Data Structure	**D2**
Art	**D3**
Artificial Intelligence	**D4**
Computacional Science	**D5**
Computer Architeture	**D6**
Computer Security	**D7**
Database	**D8**
History	**D9**
Literature	**D10**
Physics	**D11**

ela 18 - Classificação dos 41 documentos utilizando os VTD's através do *SVSim*, dos quais 13 classificados errado ·ron) e 38 classificados corretamente (cinza claro). Os códigos dos domínios constam na Tabela 17 acima.

Documentos	Domínios										
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
_reconstruction_from_multiple_ges	11,03	3,51	7,44	23,00	18,17	11,2	16,74	12,96	4,38	11,79	15,54
elerated processing unit	0,16	0,01	0,26	0,55	0,99	1,73	0,67	0,55	0,11	0,29	0,27
ective Computing	0,94	0,36	2,95	7,70	2,50	1,49	2,46	1,52	0,74	1,74	1,93
icultural_biodiversity	36,18	1,30	5,53	12,05	11,94	6,33	11,86	7,91	3,12	7,15	8,93
oecology	32,13	1,39	7,11	15,27	13,04	6,85	11,47	7,79	5,49	9,69	10,47
orithm Characterizations	0,56	3,67	0,98	3,97	2,32	2,50	1,46	1,08	0,42	0,94	1,4
achronism	0,77	0,06	3,87	1,47	0,99	0,76	1,16	0,64	8,44	2,74	0,75
tiquarian	0,60	0,02	1,71	1,09	0,76	0,32	0,74	0,41	10,30	1,50	0,54
irectionlization	0,04	0,05	0,07	0,27	0,13	0,05	0,15	0,59	0,03	0,04	0,05
anch predication	7,38	5,82	5,01	13,71	13,36	33,40	15,11	9,95	2,73	5,91	8,90
siness continuity	12,34	3,91	6,18	16,53	14,71	12,12	30,25	14,23	4,49	8,87	9,72
assic Book	0,67	0,03	1,92	1,48	0,92	0,74	1,34	0,87	1,36	6,75	0,83
llision detection	16,46	17,83	11,43	27,11	27,96	17,20	20,02	16,6	7,71	16,55	21,66
llostructional analysis	3,34	1,11	2,46	7,58	6,24	2,90	4,20	2,62	1,48	6,23	3,65
mplex instruction set computing	12,57	6,40	8,31	19,78	19,61	39,44	17,87	14,21	5,17	12,09	14,72
ncurrent data structure	8,34	8,16	9,70	15,15	16,06	27,74	19,1	20,46	2,91	8,26	10,14
tabase storage structures	7,53	3,19	3,57	7,98	10,00	10,56	10,41	17,52	3,20	5,40	7,86
stributed language	6,42	2,30	5,48	12,16	8,48	3,65	7,03	5,26	3,35	12,50	4,85
cology of contexts	8,10	1,28	3,83	8,22	6,42	4,28	7,82	4,68	2,16	4,32	6,00
perimental language	11,74	3,61	10,17	18,67	13,70	8,90	12,75	9,63	6,14	23,54	11,66
rammatology	8,09	2,01	8,82	14,00	9,90	4,95	8,83	5,05	16,32	16,70	10,03
raph Database	0,49	0,02	0,75	1,35	1,41	1,75	2,12	10,93	0,27	0,38	0,59
ash table	15,36	12,89	10,00	22,71	24,51	22,69	22,88	25,65	6,54	15,03	19,27
istoric Districts Council	3,14	0,84	4,98	5,13	4,34	2,50	6,77	3,03	1,66	2,91	2,52
istory of Computer Science	0,68	0,70	0,90	3,01	1,95	1,48	1,34	0,72	2,81	0,81	0,97
ystory of Mathematics	1,15	0,33	1,68	1,99	2,17	0,82	1,13	0,79	4,05	1,63	2,14
ndeterminacy	0,16	0,03	0,49	0,34	0,23	0,10	0,22	0,12	0,23	2,41	0,24
nformation assurance	9,74	2,57	4,54	13,63	11,97	11,10	27,94	14,04	2,62	5,99	7,58
ntegrational linguistics	8,72	2,76	7,75	17,84	12,43	8,39	12,13	9,12	4,93	19,66	9,40
Morris method	10,86	3,13	6,47	15,31	18,87	11,82	13,13	10,93	3,86	9,75	12,35
Multiphysics	5,65	1,00	3,63	8,90	9,82	4,62	8,66	7,83	1,99	5,40	10,19
Object language	10,10	5,57	8,96	20,81	17,66	16,46	15,68	11,41	5,37	16,71	15,62
Outline of Computer Science	0,46	2,80	0,60	6,53	3,54	2,43	3,01	2,62	0,47	0,46	0,91
Potencial theory	0,16	0,08	0,25	0,59	0,64	0,20	0,28	0,26	0,21	0,22	3,41
Preservationist	9,52	2,90	4,91	10,64	8,24	3,87	9,56	6,14	6,21	8,08	5,76
Pseudopotential	5,41	2,09	3,95	9,66	10,36	5,26	7,08	6,09	2,21	6,82	15,38
earch data structure	4,61	10,76	2,80	7,41	7,75	7,66	7,55	20,00	2,76	5,38	6,90
ecurity information management	6,60	1,44	2,96	10,59	9,77	12,86	20,43	15,97	2,06	4,23	5,15
erialization	12,38	4,48	9,66	22,57	23,76	23,44	25,22	30,37	5,60	13,51	17,00
ustainable agriculture	39,82	2,48	8,76	17,22	17,60	9,39	15,87	11,68	6,19	11,22	14,52

Tabela 19 - Classificação de três documentos utilizando os VTD's através do *SVSim*, utilizando no cálculo os dois tipos de fórmulas, usando IDF e sem usar IDF.

Documentos	Indeterminacy (literature)		History of Computer Science		Anachronism (history)		Accelerated processing unit (CA)	
Domínios	Sem IDF	Com IDF	Sem IDF	Com IDF	Sem IDF	Com IDF	Sem IDF	Com IDF
Agriculture	0,04	0,16	13,25	0,68	20,59	0,77	0,0486	0,16
Algorithms and Data Structure	0,02	0,03	6,98	0,70	8,76	0,06	0,0123	0,01
Art	0,05	0,49	11,41	0,90	22,68	3,87	0,0368	0,26
Artificial Intelligence	0,07	0,34	25,56	**3,01**	**30,24**	1,47	0,0758	0,55
Computacional Science	0,05	0,23	24,84	1,95	27,65	0,99	0,0867	0,99
Computer Architeture	0,03	0,1	15,09	1,48	20,3	0,76	**0,091**	**1,73**
Computer Security	0,05	0,22	16,00	1,34	23,09	1,16	0,0841	0,67
Database	0,04	0,12	11,41	0,72	20,49	0,64	0,0732	0,55
History	0,03	0,23	**32,88**	2,81	23,23	**8,44**	0,0174	0,11
Literature	**0,18**	**2,41**	16,34	0,81	29,62	2,74	0,0562	0,29
Physics	0,06	0,24	13,25	0,97	21,00	0,75	0,0528	0,27

Apêndice E - Resultado da classificação de documentos utilizando como base de consulta o Vetor gerado pelo *Intellexer Categorizer*

Tabela 20 - Classificação dos documentos nos domínios de Agricultura , Inteligência Artificial, Artes, História, Literature através do *Intellexer Categorizer*

	Domínios				
Documentos	Agriculture (%)	Inteligência Artificial(%)	Artes	História	Literature
Accelerated processing unit	40,29	41,69	31,11	25,2	35,78
Affective Computing	52,10	**67,35**	41,04	36,38	47,61
Algorithm Characterizations	39,87	58,6	35,69	30,09	44,56
Anachronism (history)	47,08	51,99	50,28	53,75	**59,39**
Antiquarian	35,66	36,85	39,38	**60,81**	49,19
Bidirectionlization	19,64	**25,13**	14,68	18,96	19,66
Classic Book	34,40	38,92	39,98	41,38	**63,55**
Graph Database	24,92	33,31	16,56	15,21	21,84
History of Computer Science	37,86	56,28	36,95	**59,97**	41,63
Hystory of Mathematics	46,24	52,9	43,79	**62,37**	54,98
Indeterminacy	24,26	31,49	26,94	24,85	**49,57**
Outline of Computer Science	40,77	**66,20**	33,02	31,38	36,51
Potencial theory	28,05	40,51	25,48	26,6	30,29
The arts and politics	30,50	33,25	**68,47**	0,96	22,29

Tabela 21 - Classificação dos documentos nos domínios de Física, Algoritmos e Estrutura de dados , Ciência Computacional, Arquitetura de Computadores, Segurança de Computadores, Banco de Dados através do *Intellexer Categorizer*

Documentos	Domínios					
	Física	Algoritmos e Estrutura de dados	Ciência Computacional	Arquitetura de Computadores	Segurança de computadores	Banco de Dados
Accelerated processing unit	36,2	15,19	42,89	**45,79**	39,49	34,42
Affective Computing	53,42	34,61	65,46	56,4	57,48	54,96
Algorithm Characterizations	47,14	**63,74**	53,08	47,84	41,81	39,9
Anachronism (history)	46,61	25,23	48,56	42,31	41,57	39,61
Antiquarian	34,85	13,02	33,98	26,14	28,38	26,72
Bidirectionlization	18,50	11,75	25,07	19,95	20,78	15,6
Classic Book	35,22	17,56	32,07	29,33	29,86	27,92
Graph Database	22,21	41,96	37,00	30,25	32,35	**62,06**
History of Computer Science	39,88	32,18	54,60	48,57	44,59	32,63
Hystory of Mathematics	51,51	30,86	52,37	39,43	40,33	38,06
Indeterminacy	29,45	13,76	22,80	20,37	21,88	22,71
Outline of Computer Science	47,04	49,44	65,46	56,40	57,48	54,96
Potencial theory	**51,55**	33,75	39,80	26,95	26,50	25,29
The arts and politics	29,45	13,45	26,83	22,43	29,96	22,29

Apêndice F - Resultado da classificação de documentos utilizando como base da classificação os vetores gerados pelo *Weka*

Tabela 22 - Classificação dos documentos 1 a 4 utilizando os vetor de termos do *Weka* através do *SVSim*

Documentos	Antiquarian	Graph Database	The arts and politics	Hystory of Mathematics
Domínios				
Agriculture	0,62	0,65	0,42	0,70
Algorithms and Data Structure	0,00	0,00	0,00	0
Art	5,25	2,67	28,44	3,85
Artificial Intelligence	2,65	3,46	1,91	4,54
Computacional Science	2,12	5,21	1,19	6,38
Computer Architeture	0,86	5,18	0,57	2,39
Computer Security	1,56	6,61	2,14	2,18
Database	1,53	28,90	2,17	2,87
History	26,63	1,03	4,02	20,56
Literature	8,17	1,32	5,17	6,28
Physics	1,37	1,28	0,92	5,47

Tabela 23 - Classificação dos documentos 5 a 9 utilizando os vetor de termos do *Weka* através do *SVSim*

Documentos	Affective Computing	Bidirectionliz ation	Classic Book	Algorithm Characterizati ons	Indetermina cy
Domínios					
Agriculture	0,81	0,41	0,41	0,73	0,38
Algorithms and Data Structure	0	-	-	-	
Art	5,23	0,09	4,36	1,99	1,37
Artificial Intelligence	10,60	1,18	2,82	10,44	2,08
Computacional Science	7,15	0,68	1,63	8,08	0,95
Computer Architeture	3,74	0,26	2,62	5,74	0,45
Computer Security	4,84	0,27	1,99	2,84	0,79
Database	4,75	0,44	2,24	4,01	0,79
History	2,27	0,10	5,15	1,74	1,95
Literature	3,85	0,60	18,38	3,07	9,13
Physics	4,45	0,06	2,02	4,33	1,24

Tabela 24 - Classificação dos documentos 10 a 14 utilizando os vetor de termos do *Weka* através do *SVSim*

Documentos	History of Computer Science	Anachronism (history)	Accelerated processing unit	Potencial theory	Outline of Computer Science
Domínios					
Agriculture	0,7	0,75	0,28	0,21	0,49
Algorithms and Data Structu	0,00	0	0,00	0	0,00
Art	2,43	8,85	0,82	0,91	2,20
Artificial Intelligence	6,74	3,19	0,82	2,62	11,44
Computacional Science	5,43	2,7	1,46	2,19	8,97
Computer Architeture	4,54	2,23	3,68	0,86	5,85
Computer Security	2,63	2,38	1,48	1,02	10,34
Database	2,70	2,17	1,89	1,45	10,69
History	16,13	12,77	0,51	1,49	1,71
Literature	2,69	8,59	0,75	1,33	1,32
Physics	1,94	1,58	0,42	5,48	2,63