UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

Sergio William Botero

Extração de relações semânticas via análise de correlação de termos em documentos

Campinas, SP
2008

Sergio William Botero

Extração de relações semânticas via análise de correlação de termos em documentos

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Engenharia de Computação.
Aprovação em 12/12/2008

Banca Examinadora:
Prof. Dr. Fernando José Von Zuben - UNICAMP
Prof. Dr. Ivan Luiz Marques Ricarte - UNICAMP
Profa. Dra. Sandra Maria Aluísio - USP

Campinas, SP
2008

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

B657e

Botero, Sergio William
Extração de relações semânticas via análise de correlação de termos em documentos / Sergio William Botero. --Campinas, SP: [s.n.], 2008.

Orientador: Ivan Luiz Marques Ricarte
Dissertação de Mestrado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Processamento de textos (Computação). 2. Semânticas. 3. Recuperação da informação. 4. Sistemas de recuperação da informação. 5. Ontologia. I. Ricarte, Ivan Luiz Marques. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título

Título em Inglês:	Extracting semantic relations via analysis of correlated terms in documents
Palavras-chave em Inglês:	Text Processing(Computation), Semantic, Information retrieval, Information retrieval system, Ontology
Área de concentração:	Engenharia de Computação
Titulação:	Mestre em Engenharia Elétrica
Banca Examinadora:	Fernando José Von Zuben, Sandra Maria Aluísio
Data da defesa:	12/12/2008
Programa de Pós Graduação:	Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Sergio William Botero

Data da Defesa: 12 de dezembro de 2008

Título da Tese: "Extração de Relações Semânticas Via Análise de Correlação de Termos em Documentos"

Prof. Dr. Ivan Luiz Marques Ricarte (Presidente): _____

Profa. Dra. Sandra Maria Aluísio: _____

Prof. Dr. Fernando José Von Zuben: _____

Resumo

Sistemas de recuperação de informação são ferramentas para automatizar os procedimentos de busca por informações. Surgiram com propostas simples nas quais a recuperação era baseada exclusivamente na sintaxe das palavras e evoluíram para sistemas baseados na semântica das palavras como, por exemplo, os que utilizam ontologias. Entretanto, a especificação manual de ontologias é uma tarefa extremamente custosa e sujeita a erros humanos. Métodos automáticos para a construção de ontologias mostraram-se ineficientes, identificando falsas relações semânticas. O presente trabalho apresenta uma técnica baseada em processamento de linguagem natural e um novo algoritmo de agrupamento para a extração semi-automática de relações que utiliza o conteúdo dos documentos, uma ontologia de senso comum e supervisão do usuário para identificar corretamente as relações semânticas. A proposta envolve um estágio que utiliza recursos lingüísticos para a extração de termos e outro que utiliza algoritmos de agrupamento para a identificação de conceitos e relações semânticas de instanciação entre termos e conceitos. O algoritmo proposto é baseado em técnicas de agrupamento possibilístico e de bi-agrupamento e permite a extração interativa de conceitos e relações. Os resultados são promissores, similares às metodologias mais recentes, com a vantagem de permitir a supervisão do processo de extração.

Palavras-chave: Processamento de texto, Extração de Relações Semânticas, Bi-Agrupamento, Agrupamento Possibilístico, Recuperação de Informação.

Abstract

Information Retrieval systems are tools to automate the searching for information. The first implementations were very simple, based exclusively on word syntax, and have evolved to systems that use semantic knowledge such as those using ontologies. However, the manual specification is an expensive task and subject to human mistakes. In order to deal with this problem, methodologies that automatically construct ontologies have been proposed but they did not reach good results, identifying false semantic relation between words. This work presents a natural language processing technique e a new clustering algorithm for the semi-automatic extraction of semantic relations by using the content of the document, a commom-sense ontology, and the supervision of the user to correctly identify semantic relations. The proposal encompasses a stage that uses linguistic resources to extract the terms and another stage that uses clustering algorithms to identify concepts and instance-of relations between terms and concepts. The proposed algorithm is based on possibilistic clustering and bi-clustering techniques and it allows the interative extraction of concepts. The results are promising, similar to the most recent methodologies, with the advantage of allowing the supervision of the extraction process.

Keywords: Text Processing, Semantic Relations Extraction, Bi-Clustering, Possibilistic Clustering, Information Retrieval.

Agradecimentos

Agradeço

Aos meus pais, Francisco e Ezildinha, e minha irmã, Susana, pelo apoio durante esta jornada e incentivo pelos estudos.

Ao meu orientador Ivan Luiz Marques Ricarte, pela orientação, apoio e incentivo, desde os tempos de iniciação científica.

Aos revisores desta dissertação por suas contribuições cuidadosas e ajuda na revisão desta dissertação.

Aos amigos da turma de graduação EE97, pelo companheirismo desde os tempos de faculdade.

Aos colegas do laboratório Harpia, pela amizade e suporte nos momentos difíceis.

Aos demais colegas de pós-graduação, pelas críticas e sugestões.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.

Aos meus pais, Francisco e Ezildinha Botero

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

M	-	Matriz de correlações
D_K	-	Matriz de ocorrência de termos em documentos
R_0	-	Matriz de associação fuzzy entre termos e conceitos
N	-	Número de documentos na base de documentos
L	-	Número de termos
$d_{i,k}$	-	relevância do termo i ao documento k
$ro_{i,j}$	-	associação fuzzy do termo i ao conceito j
$\overrightarrow{t_i}$	-	vetor termo i
J	-	função objetivo do agrupador
φ	-	função de avaliação da coesão do agrupamento
ρ	-	função para regulagem da cobertura do protótipo de conceito
θ	-	função de penalização para conceitos sobrepostos
η	-	nível de abstração
τ	-	fator de disjunção
$E(.)$	-	função de entropia do termo

Capítulo 1

Introdução

Informação, atualmente, é um elemento chave para o sucesso de qualquer tipo de negócio [42]. Os sistemas de computadores e as redes de telecomunicações permitiram às pessoas acessar e disponibilizar uma grande quantidade de informação sobre os mais diversos assuntos. Entretanto, restava a questão de como a informação disponível pudesse ser recuperada de forma precisa e rápida, trazendo informações que sejam realmente relevantes para uma consulta. A solução para este problema foi o desenvolvimento de ferramentas de recuperação automática de informação. Desta forma, foram desenvolvidos os primeiros sistemas de recuperação de informação (RI). Os sistemas de RI, assim como em acervos bibliográficos, realizam a indexação da informação por meio de palavras-chave que permitem ao usuário localizar uma determinada informação especificando aquelas que melhor descrevem o assunto de interesse.

Os sistemas de RI baseados em palavras-chave evoluíram de propostas bastante simples, como os sistemas booleanos [37] e os sistemas booleanos extendidos [58] [74], até os mais elaborados utilizando modelos vetoriais [63] e modelos baseados em semântica latente [17]. Em todos, o objetivo é melhorar os índices de desempenho do sistema. Os índices são expressos por métricas que permitem avaliar o desempenho dos sistemas e são medidas, principalmente, pela cobertura e precisão. Cobertura é a fração de documentos que são relevantes para uma consulta e que foram recuperados com sucesso; precisão é a fração de documentos recuperados que são relevantes para o usuário.

Os sistemas mais elaborados para a recuperação de informação tais como aqueles utilizando espaços vetoriais e semântica latente trouxeram melhoras nos índices de desempenho. No entanto, os sistemas de RI ainda falhavam quando tratavam com situações em que as palavras são polissêmicas ou em casos de palavras sinônimas. A solução para esse problema foi a introdução de conhecimento adicional sobre a semântica das palavras.

1.1 Motivação e relevância

Os modelos mais recentes de recuperação de informação incorporam conhecimento específico de domínio de modo a melhorar os índices de desempenho das máquinas de busca. Alguns desses modelos incorporam o conhecimento por meio de mecanismos formais e explícitos tais como aqueles que utilizam ontologias.

Em computação, uma ontologia é uma representação formal para os conceitos de um domínio e os relacionamentos entre conceitos. É utilizada para realizar inferências sobre as propriedades da ontologia e retirar conclusões a partir de fatos. Em recuperação de informações, ontologias são utilizadas para dar maior fundamentação semântica aos textos de modo que o processo de recuperação seja mais efetivo e constituem um dos pilares fundamentais para formação da Web Semântica [6]. Trazida da área de filosofia e, agora, aplicada na área de computação, o termo ontologia apresenta múltiplas definições porém nenhuma delas é completa e concisa o suficiente para descrevê-la. No entanto, muitos aceitam a definição dada por Gruber [33]: "ontologia é uma especificação formal e explícita de uma conceitualização compartilhada".

A caracterização de uma ontologia envolve a definição de algumas entidades que permitem estabelecer a rede semântica na qual são realizadas inferências para a descoberta de novo conhecimento. A concepção original de uma ontologia, baseada em lógica proposicional, define de maneira rígida as entidades do modelo ontológico. No entanto, as aplicações em mineração de dados necessitam trabalhar e realizar inferências utilizando informação vaga ou incerta. Esta dificuldade levou ao surgimento das ontologias fuzzy que incorporaram conceitos de lógica fuzzy na definição das entidades ontológicas [47] [45]. O modelo clássico de ontologia foi alterado de modo a dar suporte a graus de pertinência na definição de conceitos e relacionamentos. Assim, como proposto por Calegari e Ciucci [9], algumas entidades foram redefinidas: conceito passou a conter com diferentes graus de pertinência os indivíduos e os relacionamentos são caracterizados por números que representam o grau de associação entre os conceitos.

Percebe-se assim que especificar uma ontologia requer a definição manual de uma série de entidades. Contudo, a definição manual de uma ontologia é uma tarefa extremamente custosa, que demanda muito tempo e pessoal com conhecimento especializado no domínio tratado. Além disso, o procedimento é sujeito a erros humanos. Motivados por esses problemas é que foram desenvolvidas metodologias para a identificação automática ou semi-automáticas dessas entidades. Algumas dessas metodologias fazem uso de técnicas lingüísticas para identificação de entidades ontológicas. Outras fazem uso de técnicas estatísticas para determinar a correlação entre palavras. E existem, ainda, as que combinam as duas técnicas tais como aquelas exploradas pela área de processamento de linguagem natural.

1.2 Contribuições

A proposta descrita nesta dissertação faz uso de técnicas estatísticas e lingüísticas para a identificação das principais entidades ontológicas e suas relações obtidas a partir de um conjunto de documentos (corpus). O resultado dessa proposta foi o desenvolvimento de um sistema que pode ser utilizado por especialistas no domínio dos documentos para a construção e/ou manutenção de ontologias. Ainda, a proposta apresenta contribuições para a área de Processamento de Linguagem Natural e para a área de Inteligência Computacional.

Na área de Processamento de Linguagem Natural, uma técnica não-supervisionada para a identificação de conceitos e relacionamentos a partir do conteúdo de documentos foi desenvolvida. Primeiramente, é definido um modelo ontológico, que estabelece dois níveis de abstração para descrever o universo de discurso. O nível mais concreto é representado por termos e o nível mais abstrato é representado por conceitos. Entre estes dois níveis existem relações semânticas fuzzy de instanciação entre termos e conceitos.

A definição do modelo ontológico permite estabelecer os requisitos do algoritmo de construção de ontologias. Neste caso, os requisitos incluem a identificação dos termos, dos conceitos e de suas relações semânticas.

Técnicas lingüísticas são utilizadas para identificar a lista de termos no vocabulário da ontologia. Os conteúdos dos documentos são processados por um analisador léxico que extrai as palavras e categoriza-as de acordo com a sua função morfossintática. A categorização é realizada por meio de uma base lexical.

Técnicas estatísticas são utilizadas para identificar a lista de conceitos. Para este fim, uma estrutura matemática é atribuída à entidade termo e uma métrica é definida para medir a correlação, ou também proximidade, entre os termos. A identificação de conceitos parte da hipótese de que termos que apareçam de forma correlata nos documentos caracterizam algum conceito abordado por estes. Desse modo, a identificação de conceitos pode ser vista como a identificação de agrupamentos de termos correlatos.

Para a identificação de termos correlatos será utilizado uma versão modificada do agrupador possibilístico. Uma nova função-objetivo será proposta e um novo algoritmo iterativo será desenvolvido para a extração dos grupos. Ainda, técnicas de bi-agrupamento são empregadas para diminuir a complexidade computacional do agrupador.

A técnica proposta apresenta como vantagens a possibilidade de regular o nível de abstração dos conceitos extraídos, realizar atualizações na ontologia e, principalmente, supervisionar o processo de extração. Ainda são apresentados uma proposta para a atribuição automática de nomes aos conceitos e uma possível estratégia para a obtenção de relações semânticas de hiperonímia e hiponímia.

Na área de Inteligência Computacional foi proposto um novo método para a realização de agru-

pamento de objetos, no caso desta dissertação, os termos obtidos da análise dos documentos. O desenvolvimento desse novo método, motivado pela incapacidade dos métodos tradicionais de lidar com situações nas quais não havia a definição de protótipos de grupo, partiu da análise da função-objetivo do agrupador possibilístico da qual foram extraídas as principais propriedades dessa função. Essas propriedades permitiram a definição de novos critérios e funções para o agrupamento de objetos, utilizando somente a informação de distância entre eles. O processo identifica um grupo de cada vez por meio de uma estratégia iterativa, que trouxe como vantagens a definição de métricas para o número adequado de grupos e a supervisão do processo de agrupamento.

1.3 Organização

O restante da dissertação está organizado da seguinte maneira:

- No Capítulo 2, são discutidos os principais métodos de extração de relações semânticas em documentos. São apresentados os métodos baseados em processamento de linguagem natural e os métodos baseados em estatística. O modelo de documento adotado nesta dissertação também é apresentado.
- No Capítulo 3, é feita uma breve revisão dos métodos de aprendizado de máquina baseados em técnicas de agrupamento. Esta revisão apresenta as vantagens, desvantagens e características dos agrupadores mais conhecidos. Esta revisão ajudará na elaboração do método proposto para a extração semântica de conceitos.
- O Capítulo 4 apresenta a proposta para a extração das relações semânticas. São discutidos os procedimentos para a obtenção da lista de termos, a função objetivo para o agrupamento, e os algoritmos desenvolvidos para a extração e nomeação dos conceitos. Ainda, é apresentada a estratégia interativa para a identificação de conceitos.
- Os resultados são apresentados no Capítulo 5. O algoritmo é avaliado em relação à qualidade e quantidade dos conceitos extraídos havendo também comparação com outras técnicas. Em seguida, o mecanismo de atribuição de nome aos conceitos é avaliado.
- Por fim, o Capítulo 6 apresenta as conclusões da dissertação e discussões sobre suas vantangens e desvantagens. Apresenta também novas possibilidades de melhorias para o algoritmo proposto e a aplicabilidade em outros domínios.

Capítulo 2

Extração de Relações Semânticas

Recuperação de informação é a ciência que lida com a representação, armazenamento, organização e busca da informação. A idéia de se utilizar computadores para a realização de buscas surgiu com um famoso artigo de Vannevar Bush em 1945 [8], *As We May Think* e as primeiras implementações surgiram nas décadas de 50 e 60.

Os modelos de recuperação de informação surgiram com propostas bastantes simples e, ao longo do tempo; motivados pelo advento da Internet, tornaram-se mais complexos e completos. Apesar dos inúmeros avanços, a maioria dos modelos pode ser reduzida a um sistema formal descrito pela quádrupla $[D, Q, F, R]$, na qual:

- D é um conjunto representando os documentos, isto é, representa os itens armazenados;
- Q é um conjunto representando as consultas, isto é, representa as necessidades do usuário;
- F é um *framework* para modelagem dos documentos, as consultas, e seus relacionamentos;
- $R : Q \times D \rightarrow \mathbb{R}$ é uma função de similaridade que expressa quão relevante um documento é para um determinada consulta.

O processo de recuperação também é similar aos sistemas que utilizam o formalismo anterior. O usuário expressa a consulta q e submete ao sistema de recuperação de informações. A consulta é interpretada pelo framework F o qual aplica a função de relevância $R(q, d)$ para cada documento armazenado na base de documentos. Por fim, o resultado é apresentado ao usuário na ordem decrescente de relevância.

O número de documentos retornados pode ser extremamente elevado, de modo que é razoável aplicar algum critério que permita retirar um subconjunto D_r de D que potencialmente responde à consulta q. Dois problemas que podem ocorrer nesses sistemas são: nem todos os documentos em D_r são relevantes; e D_r pode não incluir documentos relevantes de D. Com base nessa constatação, duas medidas de desempenho foram propostas.

A primeira medida de desempenho, denominada cobertura e denotada por r, mede a fração de documentos que são relevantes para a consulta e que foram recuperados com sucesso, dada na forma:

$$r = \frac{ri(D_r)}{ri(D)} \tag{2.1}$$

A segunda medida de desempenho, denomindada precisão e denotada por p, mede a fração de documentos recuperados que são relevantes para a consulta do usuário, como segue:

$$p = \frac{ri(D_r)}{||D_r||} \tag{2.2}$$

onde $||D_r||$ é o número de elementos no conjunto D_r e $ri(S)$ é uma função que expressa o número de itens relevantes no conjunto S. Os índices r e p são valores no intervalo $[0, 1]$ e quanto mais próximos de 1, melhor é o sistema de recuperação.

Motivados pela desejo de se ter um sistema mais próximo possível do ideal ($r = p = 1$), diversas propostas surgiram. Uma dessas propostas trabalha com a idéia de ontologias, que permite a especificação de conceitos e relacionamentos. A utilização desses artefatos ontológicos são capazes de agregar maior valor semântico a documentos de tal forma que o computador possa "compreender" o assunto abordado por um determinado documento. A implicação desse novo paradigma é que ferramentas de busca podem ser mais efetivas na recuperação da informação.

Os métodos de recuperação de informação baseados na utilização de ontologias surgiram como uma resposta para as deficiências do modelo baseado somente na representação léxica de termos. As deficiências desse modelo afeta diretamente os requisitos de precisão e cobertura dos buscadores: documentos não relacionados à busca são recuperados e documentos relevantes que não contem ao menos um termo da busca não são recuperados.

Esses problemas ocorrem porque a busca baseada na representação léxica indexa os documentos com base em termos que representam informações vagas e, algumas vezes, representam mais ruído do que informação útil. Outro problema é que a informação requisitada pelo usuário está muito mais relacionada a conceitos ou assuntos do que na indexação de termos.

Uma solução para este problema foi a introdução de um novo elemento denominado *conceito* que associa indiretamente os termos aos documentos [69] [75]. A figura 2.1 mostra a mudança no paradigma de recuperação de informação. As arestas unindo os documentos aos conceitos e as arestas unindo os termos aos documentos são relações semânticas de instânciação. Relações semânticas entre conceitos tais como as relações de hiperonímia e hiponímia não estão representadas.

Em geral, o espaço de conceitos é de menor dimensão que a dimensão de termos. Assim, documentos podem ser recuperados mesmo quando não há um vínculo direto entre o termo requisitado na busca e o documento. Considere o exemplo da figura 2.1 e a situação em que o documento d1 con-

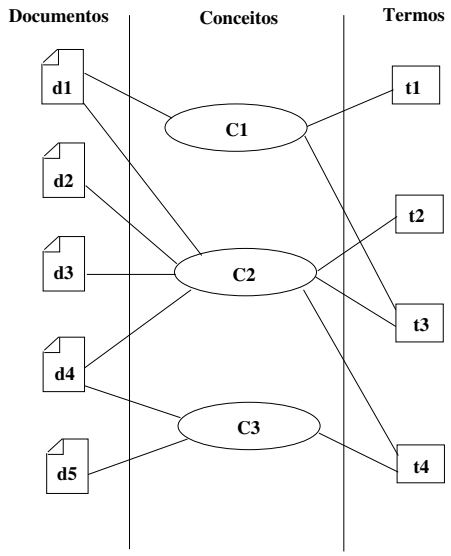


Fig. 2.1: Modelo com conceitos

tenha somente o termo t1 que é sinônimo de t3. No modelo que utiliza somente termos, o resultado de uma busca por t3 não retornará o documento d1, apesar da relação de sinonímia entre os termos t1 e t3. O novo paradigma, com a camada de conceitos, soluciona esse problema ao introduzir relações semânticas entre os termos. Observe que, nesse modelo, os termos t1 e t3 estão associados por meio do conceito C1 e, assim, uma consulta pelo termo t3 pode retornar o documento d1.

A idéia de se utilizar conceitos para melhorar a busca é interessante. Contudo, uma outra questão permanece: como obter tais conceitos? Uma estratégia é contratar um especialista para manualmente definir os conceitos e os relacionamentos com os termos e os documentos. No entanto, essa estratégia é custosa e sujeita a erro humano. Uma alternativa é utilizar métodos de aprendizado de máquina para extrair automaticamente os conceitos e seus relacionamentos.

Trabalhos relacionados à extração de relações semânticas via análise dos documentos podem ser divididos em duas abordagens diferentes. Uma faz uso de métodos lingüísticos e a outra faz uso da análise estatística de termos em documentos (métodos numéricos) [28]. Existem também as técnicas de processamento de linguagem natural que combinam técnicas lingüísticas e estatísticas. Nos métodos lingüísticos, os documentos são processados por analisadores sintáticos e semânticos. Dessas análises resulta a identificação de conceitos, termos e relações entre as entidades do modelo ontológico. Na linha de análise estatística, os métodos trabalham com a distribuição dos termos nos documentos, procurando determinar relações de semelhança, diferença ou hierárquica entre termos. Na linha de processamento de linguagem natural podemos citar aquelas que utilizam *Formal Concept Analysis* (FCA) e suas variantes [13] [1].

2.1 Ontologias

Ontologia ("conhecimento do ser") é a parte da filosofia que trata da natureza do ser, da realidade, da existência dos entes e das questões metafísicas em geral. Em computação, ontologia é um modelo de dados que permite representar diversas entidades nos mais diferentes níveis de abstração na qual é possível a criação de conceitos, relacionamentos, taxonomias e axiomas. O grande interesse nesse modelo é a aplicação de algoritmos capazes de realizar inferências com as entidades ontológicas e retirar conclusões a partir de fatos.

Ontologias têm sido utilizadas em aplicações tais como máquinas de busca [27], engenharia de software [21] e processamento de linguagem natural [20]. Em todas essas áreas as ontologias são usadas para representar o conhecimento.

Em ciência da computação não há uma definição formal e única que seja aceita por todos os pesquisadores. A definição mais conhecida é dado por Gruber [33] e diz que:*"Ontologia é uma especificação formal e explícita de uma conceitualização compartilhada"*.

Outra definição mais próxima da área de computação é dada por Guarino [34]: *"Uma ontologia é um artefato de engenharia, constituída por um vocabulário específico usado para descrever uma certa realidade, e um conjunto explícito de suposições considerando o significado pretendido das palavras desse vocabulário"*. E, também, por Chandrasekaran [11]: *"Ontologias são teorias de conteúdo sobre tipos de objetos, propriedades de objetos e relações entre objetos que são possíveis em um domínio do conhecimento específico"*.

Ontologias podem ser classificadas pela forma como são implementadas. Uschold [71] definiu três dimensões para classificar uma ontologia:

Formalidade:

- Altamente informal: Expressada sem muito rigor em linguagem natural.
- Estruturada e Informal: Expressada em um forma de linguagem natural restrita e estruturada, aumentanto a clareza ao se reduzir a ambiguidade.
- Semi-formal: expressada em um linguagem artificial definida formalmente.
- Rigorosamente formal: termos meticulosamente definidos com um semântica formal, teoremas e provas.

Propósito: refere-se a uso que se dará a ontologia. Uschold identificou três cenários possíveis:

- Comunicação. Comunicação entre pessoas; Neste caso, uma ontologia informal mas não-ambígua seria suficiente.

- Inter-Operabilidade. Neste caso, ontologias são utilizadas como um formatopara troca de informações entre os diferentes sistemas.
- Benefícios para a Engenharia de Sistemas: Re-Usabilidade, Aquisição de conhecimento, confiabilidade, especificação;

Assunto: A natureza do assunto que a ontologia irá caracterizar.

As ontologias ainda podem ser classificadas de acordo com o nível de generalidade. Existem as ontologias gerais (de senso comum), na qual contém conceitos e termos que sejam comuns a qualquer área do discurso. E existem as ontologias de domínio cujo propósito é descrever domínios específicos. Alguns exemplos de ontologia geral são apresentadas a seguir.

Cyc: Cyc [1] é uma grande ontologia que provê um vasto conhecimento de senso comum. Cyc é baseado na micro teorias, cada qual captura o conhecimento para diferentes domínios de diferentes pontos de vista tais como espaço, tempo e causalidade. Ontologias Cyc são implementadas em CycL que é uma linguagem formal cuja sintaxe deriva do cálculo de predicados de primeira ordem. O vocabulário de CycL consiste de termos, os quais podem ser combinados para formar expressões mais complexas em CycL. O conjunto dessas expressões formam a base de conhecimento.

TOVE: TOVE (Toronto Virtual Enterprise) [50] é uma abordagem composta de um modelo integrado com suporte a inferência e baseado em ontologias de núcleo. TOVE prove um método formalização para combiná-los em um sistema. A metodologia inclui os seguintes passos: define-se um conjunto de cenários de motivação (*Motivating Scenarios*), um conjunto de questões de competência informal (*Informal Competence Questions*) que a ontologia deve responder, uma terminologia para a ontologia utilizando Lógicas de Primeira Ordem, uma semântica e as restrições na terminologia.

On-To-Knowledge: On-To-Knowledge [23] é um projeto do programa europeu para as tecnologias para a sociadade da informação (European Commission Information Society Technologies). A metodologia prove orientação para a introdução de ferramentas de gerência de conhecimento em aplicações enterprises. Ela inclui a identificação de objetivos que deverão ser atingidos e são baseadas na análise de negócio e nos diferentes papéis que o conhecimento desempenha na organização. OIL [2] (Ontology Interchange Language) é a linguagem padrão desenvolvida para o projeto OnToKnowledge. Ela utiliza três paradigmas: Modelagem baseada em qua-

[1] `www.cyc.com`

[2] `http://www.ontoknowledge.org/oil/index.shtml`

dro (frame-based modeling), com semântica baseada e lógica descritiva e sintaxe baseada em padrões Web tais como XML schema e RDF schema.

WordNet: O projeto WordNet começou cerca de 20 anos atrás no Laboratório de Ciência Cognitiva de Princeton e está continuamente em atualização. WordNet é uma base de dados léxica baseada nos princípios da psico-lingüística. Originalmente foi desenvolvido para a lingua Inglesa, mas atualmente é possível encontrar versões do WordNet em muitas outras línguas. A unidade básica de informação é chamada de *synset*, o qual é um conjunto de sinônimos que podem ser permutados em um determinado contexto. WordNet contem cerca de 114.648 substantivos, 79.689 *synsets* e diversas relações semânticas entre os *synsets*. Sinonímia, Antonímia, Hiponímia, Hiperonímia, Meronímia, Holonímia e Troponímia são algumas das relações presentes no WordNet.

Nesta dissertação, a ontologia resultante do processo de extração de relações semânticas é de domínio pois a fonte de informação é o próprio conteúdo dos documentos. No entanto, será utilizada ontologia de senso comum para extrair os termos presentes nos documentos.

A seguir são listados os recursos mais comuns em ontologias tais como a definição de conceitos, relacionamentos e as declarações de fatos, também conhecidos como indivíduos.

- Indivíduos (ou Instâncias): representam as entidades mais simples de uma ontologia. Em geral, essas entidades representam elementos concretos, tais como: pessoas, animais, plantas, e objetos. Mas é possível, também, a representação de elementos abstratos, tais como: números e símbolos. Em ontologias, indivíduos pertencem necessariamente a um ou mais conceitos;
- Classes (ou conceitos): representam entidades mais complexas em uma ontologia, normalmente são entidades que remetem a idéias abstratas ou remetem à noção de coleção de indivíduos que compartilham alguma característica em comum. Exemplo de conceitos:
  - Classe Veículo. Classe de todos os meios de transporte, seja motorizado ou não, por quaisquer via: terrestre, marítima ou aérea;
  - Classe Réptil. Classe de todos os animais que são vertebrados tetrápodes e ectotérmicos, ou seja, não possuem temperatura corporal constante;
  - Classe Número. Classe de todos os números, seja ele racional, irracional, real ou complexo.
- Relações: permitem estabelecer associações entre conceitos. As relações são de dois tipos: taxonômicas ou não-taxonômicas.
  - Taxonômicas: permitem estabelecer associações entre conceitos mais gerais e conceitos mais específicos. Em lingüística estas relações também são conhecidas como hiperonímia

e hiponímia. Por exemplo: a expressão "veículo automotor" é uma hiperonímia do termo "carro" pois trata-se de uma expressão mais geral que denota todas as formas de veículos movidos a motor, não somente carros mas também ônibus e caminhões. No sentido oposto temos que o termo "carro" é uma hiponímia da expressão "veículo automotor";

- Não-Taxonômicas: são outras relações que não se enquadram nas relações taxonômicas tais como meronímia, holonímia, antonímia, troponímia, sinonímia e de instanciação (*instance of*).
  - Meronímia: permite estabelecer uma relação semântica entre conceitos determinando que uma dada entidade é parte de outra. Por exemplo, "dedo" é uma meronímia de "mão" porque dedo é parte da mão.
  - Holonímia: permite estabelecer uma relação semântica entre conceitos determinando que uma dada entidade é composta de outras entidades. Por exemplo, "roda" é uma meronímia de "automóvel".
  - Antonímia: é a relação que se estabelece entre dois conceitos ou mais que apresentam significados diferentes, contrários (antônimos). Por exemplo, economizar - gastar; bem - mal; bom - ruim.
  - Troponímia: é uma relação similar à hiponímia mas aplicada aos verbos.
  - Sinonímia: é a relação que se estabelece entre dois conceitos ou mais que apresentam o mesmo ou semelhante significado. Por exemplo, gordo - obeso; morrer - falecer; após - depois.
  - Instanciação: é a relação que se estabelece entre um termo, uma entidade mais concreta, e um conceito, uma entidade mais abstrata.

Parte de uma ontologia sobre o ramo automotivo é apresentada na figura 2.2 e ilustra os principais elementos ontológicos e a notação gráfica que será utilizada para representar cada entidade.

As marcações em números na figura denotam as principais entidades do modelo ontológico. Uma descrição mais detalhada das marcações é apresentada a seguir:

1. Toda ontologia apresenta um conceito do qual todos os elementos fazem parte. Esse é representado pelo conceito "Coisa".
2. Conceitos são representados com a figura geométrica elipse. Neste exemplo, são cinco os conceitos: "Coisa", "Veículo Automotor", "Pneu", "Ônibus", "Carro".
3. Relações taxonômicas são representadas com um arco orientado, onde o sentido do arco aponta para o conceito mais genérico. No exemplo, são quatro as relações taxonômicas.

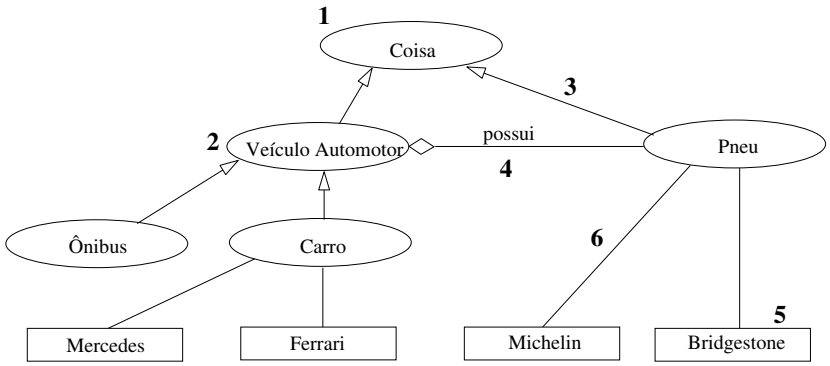


Fig. 2.2: Exemplo de parte de uma Ontologia

4. Relações não-taxonômicas são representadas com um arco entre dois conceitos. Relações não-taxonômicas de meronímia/holonímia ainda podem ser representadas com um arco com o símbolo de diamante no lado do conceito que representa a idéia do todo. No exemplo dado, as instâncias do conceito "Veículo Automotor" possuem instâncias do conceito "Pneu".
5. As entidades mais concretas, denominadas indivíduos, são representados com um retângulo. O exemplo possui quatro indivíduos: "Mercedes", "Ferrari", "Michelin" e "Bridgestone".
6. O relacionamento entre termo e conceito, denominado de relação de instanciação (*instance of*), é representado por um arco simples.

As construções ontológicas apresentadas na figura 2.2 são apenas um sub-conjunto de muitas outras possibilidades. O consórcio para a *world wide web* padronizou uma linguagem para a descrição de ontologias, denominada de OWL [3], especificando a sintaxe e semântica de cada entidade ontológica. Essa linguagem serve de referência para outras construções ontológicas listadas nesta dissertação.

2.1.1 Modelos de definição ontológica

As diferentes entidades de uma ontologia ainda podem ser categorizadas em extensional ou intencional, dependendo da forma como são definidas.

Modelo Extensional

O modelo extensional formula o significado de um conceito por meio da especificação exaustiva de todos os objetos que se enquadram na definição do conceito em questão. Por exemplo, a definição extensional para o conceito "nações do mundo" pode ser dada pela enumeração de todas as nações do

[3] `http://www.w3.org/TR/owl-ref/`

mundo. A listagem explicitando todos os objetos de um conceito somente é possível para conjuntos finitos e é somente prática para conjuntos pequenos. A definição extensional costuma ser utilizada quando os objetos pertencentes ao conceito trazem mais significado ao conceito que a sua própria definição.

Exemplos de definição extensional para conceitos

Conceito Vogais $\equiv$ {a,e,i,o,u}

Conceito Nomes $\equiv$ {João, Maria, José}

Perceba que nesta forma de definição é impossível a representação de conjuntos infinitos contáveis, como o conjunto dos números pares, ou conjuntos infinitos não contáveis, como o conjunto de pares (x, y) que satisfaçam a função $y = x^2 + 3x + 10$.

Modelo Intencional

Em lógica e matemática, uma definição intencional de um conceito permite dizer se um dado objeto pertence ou não a este conceito avaliando-se as propriedades requeridas que acompanham a definição intencional, isto é, estabelece condições necessárias e suficientes para que um objeto possa pertencer a um dado conceito. Obviamente, a definição intencional é mais aplicável quando tem-se uma perfeita definição do conjunto de propriedades. Diferentemente do modelo extensional, é possível a definição de conceitos que contenham um número infinito de objetos. Por exemplo, o conceito formado pelos números pares: é impossível uma definição extensional, pois o conjunto é composto por infinitos números; no entanto, a definição intencional é perfeitamente possível, bastando estabelecer um regra que permita associar um número a esse conceito. Nesse caso, define-se a seguinte regra: conceito formado pelos números que sejam múltiplos de dois. A definição intencional também pode ser definida por meio de um conjunto de axiomas ou regras que permitem gerar todos os membros de um determinado conceito. Por exemplo, a definição intencional de "número quadrado" pode ser "tome um inteiro e multiplique por ele próprio". Não importa o número tomado, a multiplicação deste por ele próprio sempre gera um número quadrado.

Exemplos de definição intencional para conceitos:

Conceito Vogais $\equiv$ {x: x é uma vogal}

Conceito Nomes $\equiv$ {x: x é uma pessoa da família de Maria}

Nesta forma de definição é perfeitamente possível a representação de conjuntos infinitos, sejam eles contáveis ou não contáveis.

Conceito Números Pares $\equiv \{x : x \; mod \; 2 = 0\}$

Conceito Função de Segundo Grau $\equiv \{(x, y) : y = x^2 + 3x + 10\}$

2.2 Técnicas Linguísticas

Os métodos linguísticos para a extração de relações ontológicas fazem uso de analisadores léxicos e sintáticos para tentar "compreender" os textos da base de documentos. No estágio de análise léxica, os termos são extraídos dos textos, categorizados (mapeados para classes léxicas, tais como substantivos, verbos, adjetivos, etc) e reduzidos à sua forma primitiva (também conhecido como lematização).

A análise sintática realiza o processamento parcial ou total de estruturas lingüísticas. Dessa análise resulta a descoberta de relações semânticas entre conceitos. Já a interpretação do conteúdo semântico expresso pela linguagem natural requer algum nível de conhecimento geral e, em algumas vezes, de conhecimento especializado.

As técnicas lingüísticas são de diversos níveis de complexidade, desde as mais simples, utilizando a busca de padrões sintáticos, até as mais complexas, que utilizam análise terminológica e sintática [67] [52].

A seguir, são apresentadas dois métodos para a extração de conceitos baseados nessa técnica.

2.2.1 Método de Hearst

O método de Hearst [39] para a extração de relações ontológicas baseia-se na busca de padrões sintáticos. Alguns padrões sintáticos permitem a identificação de relações de hiperonímia e hiponímia entre termos.

Alguns dos padrões identificados por Hearst são:

- NP_A such as NP_B;
  "... diseases such as hepatitis ..."
  hiponímia(disease,hepatitis), hiperonímia(hepatitis,disease)

- NP_1 such as NP_2 (and|or) NP_3;
  "... cities such as Beijing and Guangzhou ..."
  hiponímia(cities, Beijing), hiperonímia(Beijing, cities)
  hiponímia(cities, Guangzhao), hiperonímia(Guangzhao, cities)

- NP_1 such as NP_2, NP_3 (and|or) NP_4;
  "... infections such as bronchitis, sinusitis or pneumonia ..."
  hiponímia(infections, bronchitis), hiperonímia(bronchitis, infections)
  hiponímia(infections, sinusitis), hiperonímia(sinusitis, infections)
  hiponímia(infections, pneumonia), hiperonímia(pneumonia, infections)

- NP_1 {,} (or|and) other NP_2;
  "... vaccines, or other injectables ..."
  hiponímia(injectables, vaccine), hiperonímia(vaccine, injectables)

- $NP1$, $NP2$ {,} (or|and) other $NP3$;
  "... royalties, fees, and other revenues ..."
  hiponímia (revenues, royalties), hiperonímia(royalties, revenues)
  hiponímia (revenues, fees), hiperonímia(fees, revenues)

- NP_1, NP_2, NP_3 {,} (or|and) other NP_4;
  "... Italy, Canada, the US and other countries ..."
  hiponímia (countries, Italy), hiperonímia(Italy, countries)
  hiponímia (countries, Canada), hiperonímia(Canada, countries)
  hiponímia (countries, US), hiperonímia(US, countries)

- NP_1 {,} {including|especially} NP_2;
  "... cytokines, including BNF ..."
  hiponímia (cytokines, BNF)

- NP_1 {,} {including|especially} NP_2 {or|and} NP_3;
  "... technologies including ATLAS and SCAN ..."
  hiponímia (technologies, ATLAS),
  hiponímia (technologies, SCAN)

As siglas NP_A, NP_B, NP_1, NP_2, NP_3 e NP_4 são sintagmas nominais (Noun Phrase - NP). Esses padrões permitem estabelecer relações de hiperonímia e hiponímia entre uma expressão mais genérica e outra expressão mais específica. Por exemplo, no padrão "NP_A *such as* NP_B", NP_A é a expressão mais genérica e NP_B é a expressão mais específica. Hearst aplicou essas regras para extrair relações taxonômicas da Enciclopédia Acadêmica Americana Grolier [32] e comparou-as com as relações taxonômicas presentes no dicionário digital WordNet [54]. A tabela 2.1 mostra algumas instâncias das relações de hiperonímia/hiponímia encontradas; as demarcadas com * também estavam presentes no WordNet.

Os resultados presentes nesta tabela mostram que o método de Hearst têm potencial para a extração de relações semânticas. Os resultados motivaram a pesquisa por outras propostas baseadas em

Termo mais abtrato	*Termo mais específico*
cereal	rice*, wheat*
countries	Cuba, Vietnam, France*
hydrocarbon	ethylene, benzene, gasoline
substances	bromine*, hydrogenrice*, phosphorus*, nitrogen*
protozoa	paramecium
liqueurs	anisette*, absinthe*
rocks	granite*
substances	
species	steatornis, oilbirds
bivalves	scallop*
fungi	smuts*, rusts*
fabrics	acrylics*, nylon*, silk*
antibiotics	ampicillin, erythromycin*
institutions	temples, king
seabirds	penguins, albatross*
flatworms	tapeworms, planaria
amphibians	frogs*
waterfowl	ducks
legumes	lentils*, beans*, nuts
organism	horsetails, ferns, mosses
rivers	Sevier, Carson, Humboldt
fruits	olives*, grapes*
ideologies	liberalism, conservatism
industries	steel, iron, shoes
minerals	pyrite*, galena
infection	meningitis
phenomena	lightning*
dyes	quercitron

Tab. 2.1: Instâncias das Relações de Hiperonímia/Hiponímia encontradas em Grolier

buscas por padrões, tais como a proposta de Cimiano *et al.* [14] e Caraballo [10]. Cimiano utiliza fontes de dados diferentes para validar as relações extraídas com a identificação dos padrões. Em particular, a validação das relações é feita com a utilização da base de dados lexicais WordNet e via análise de ocorrência de termos em documentos na Web.

Apesar dos avanços nessas técnicas, em algumas situações a busca por padrões identificou erroneamente relações de hiperonímia/hiponímia. Por exemplo em (“king”,“institution”) que na verdade é uma relação de metonímia e, em outros casos, a relação é dependente de contexto, como acontece em (“Washington”,“nationalist”) e também em (“aircraft”,“target”).

Os padrões de Hearst já foram adaptados a outras línguas, inclusive o português. Os padrões sintáticos em língua portuguesa foram explorados por de Freitas e Quental [15].

2.2.2 Mineração de Ontologias a partir de Textos

Mineração de Ontologias a partir de Textos, *Mining Ontologies from Text*, é uma técnica desenvolvida por Maedche *et al.* [52] para a identificação de conceitos e relações taxonômicas, e apresenta um mecanismo para a descoberta de relações não-taxonômicas baseado no algoritmo de Srikant e Agrawal [68].

A técnica de Maedche é composta de diversos estágios para o processamento dos documentos. O primeiro estágio é o processamento dos textos via SMES (Sarbrucken Message Extraction System) que realiza a tokenização, a análise morfológica e léxica dos textos. O outro estágio é um processador sintático que extrai um conjunto de informações sobre a estrutura sintática dos textos. O resultado desses dois módulos é um texto anotado com rótulos no formato padronizado XML, contento as informações do processador lingüístico.

As informações retornadas pelo módulo SMES alimentam o módulo de aprendizado e descoberta (*Learning & Discovering*), que identificará os conceitos e as relações semânticas.

Os conceitos são inseridos no sistema aplicando-se mecanismos de extração de termos baseados nas medidas de relevância dos termos aos documentos, propostos por Salton [64], e os termos mais freqüentes são candidatos a conceitos, que são manualmente adicionados ao domínio da ontologia.

As relações entre conceitos são determinadas em dois estágios. O primeiro destinado a descobrir as relações taxonômicas e o segundo destinado a descobrir as relações não-taxonômicas. As relações taxonômicas são extraídas utilizando os padrões sintáticos de Hearst, que foram descritos na seção 2.2.1. As relações conceituais não-taxonômicas são extraídas utilizando o algoritmo de Srikant e Agrawal para a descoberta de regras de associação generalizada, descrita a seguir.

Na proposta de Srikant é definido, inicialmente, um conjunto de transações $T = \{t_i | i = 1 \dots n\}$, onde cada transação t_i consiste de um conjunto de itens $t_i = \{a_{i,j} | j = 1 \dots m_i, a_{i,j} \in C\}$ e cada item $a_{i,j}$ é da lista de conceitos C. O algoritmo calcula as regras de associação $X_k \Rightarrow Y_k$ $(X_k, Y_k \subset C, X_k \cap Y_k = \{\})$ para os quais as medidas para suporte e confidência excedam um limiar definido pelo usuário. O suporte da regra $X_k \Rightarrow Y_k$ é o percentual de transações que contêm $X_k \cap Y_k$ como um sub-conjunto, e confidência para $X_k \Rightarrow Y_k$ é definida como o percentual de transações em que Y_k é visto quando X_k aparece em uma transação, ou seja:

$$\text{suporte}(X_k, Y_k) = \frac{|\{t_i | X_k \cup Y_k \subseteq t_i\}|}{n} \tag{2.3}$$

$$\text{confidência}(X_k, Y_k) = \frac{|\{t_i | X_k \cup Y_k \subseteq t_i\}|}{|\{t_i | X_k \subseteq t_i\}|} \tag{2.4}$$

Os autores dessa proposta estenderam esse mecanismo básico para determinar associações quando relações taxonômicas estão presentes entre os conceitos. Para este propósito, eles primeiro estende-

ram cada transação t_i para incluir os conceitos mais abstratos de um conceito particular $a_{i,j}$, isto é, $t_i^{'} := t_i \cup \{a_{i,l} | (a_{i,j}, a_{i,l}) \in H\}$. Então, eles calcularam a confidência e o suporte para todas as possíveis regras de associação $X_k \Rightarrow Y_k$ onde Y_k não contém um superconceito de X_k. Terminada a fase de identificação das regras, realiza-se uma limpeza das regras onde regras $X_k \Rightarrow Y_k$ que estejam cobertas por uma regra mais geral $\hat{X_k} \Rightarrow \hat{Y_k}$ são removidas.

A saída do algoritmo é a lista de regras que constituem as relações não taxonômicas entre os conceitos. A avaliação dessa técnica mostrou que o processo de descoberta das relações não-taxonômicas ainda requer muita intervenção humana. Por outro lado, a proposta é muito adequada no auxílio a especialistas para a construção de ontologias, propondo possíveis relações conceituais.

2.3 Técnicas Estatísticas

As técnicas estatísticas para a extração de relações semânticas referem-se aos métodos em que são empregadas técnicas de análise da distribuição dos termos nos documentos. Portanto, são técnicas que não utilizam a estrutura sintática e/ou semântica da linguagem. O único processamento lingüístico realizado é o léxico que deverá extrair do conteúdo dos documentos a lista de termos. Extraídas as palavras dos documentos, o segundo passo é o levantamento de histogramas que permitem estabelecer a importância de cada palavra para a caracterização dos documentos. Esse procedimento é realizado por meio da contagem do número de vezes que as palavras aparecem nos documentos. Assim, ao final desse processo de contagem o resultado é um modelo que é a representação matemática dos documentos. O modelo normalmente é uma matriz que permite avaliar a importância dos termos junto aos documentos. Essa matriz será denominada de D_k e o procedimento para a sua obtenção será detalhado na próxima seção.

O terceiro e mais importante procedimento é a análise estatística das informações. Basicamente, o processo de análise tenta identificar padrões recorrentes e criar um modelo matemático simplificado que sintetiza os dados contidos na matriz.

2.3.1 Modelo de documentos

Um modelo de documento é a representação abstrata da estrutura física ou semântica de um documento. O modelo mais comum é baseado na matriz de ocorrência de termos em documentos. No entanto, existem outros modelos mais complexos, tal como o modelo baseado em frases proposto por Zamis e Eztioni [77], no qual procuram-se sufixos de frases que são compartilhadas por outros documentos. Nesta dissertação, utilizaremos o primeiro modelo, devido à metodologia adotada para a obtenção dos conceitos, que é baseada na identificação de termos correlatos.

No modelo de ocorrência de termos em documentos, também conhecido como modelo bolsão de palavras (*bag of words*) [64], a lista de termos é extraída do conjunto total de documentos. Os documentos são expressos na forma vetorial onde cada coordenada está associada a um termo que representa uma *característica* do documento. Os valores das coordenadas do vetor documento representam a relevância dos termos para o documento. A união de todos os vetores forma a matriz D_k:

$$D_k = \begin{pmatrix} dk_{1,1} & dk_{1,2} & \cdots & dk_{1,N} \\ dk_{2,1} & dk_{2,2} & \cdots & dk_{2,N} \\ \cdots & \cdots & \cdots & \cdots \\ dk_{M,1} & dk_{M,2} & \cdots & dk_{M,N} \end{pmatrix} \tag{2.5}$$

Os parâmetros N e M representam o número de documentos e o número de termos, respectivamente. Os valores da relevância dos termos aos documentos são computados utilizando-se as medidas tf.idf [64], apresentada na equação 2.6. A medida tf-idf, freqüência do termo-feqüência inversa do documento, é o peso frequentemente utilizado em recuperação de informação e mineração de textos e representa uma medida estatística para avaliar a importância de uma palavra para um documento em um conjunto de documentos. Em geral, esta medida procura estabelecer a seguinte relação: a importância de um palavra aumenta proporcionalmente ao número de vezes que a palavra aparece no documento e diminui à medida que esta palavra aparece em outros documentos.

$$d_{i,k} = (\text{tf.idf})_{i,k} = f_{i,k}.\log\left(\frac{N}{n_i}\right) \tag{2.6}$$

onde:

- i é um índice para o termo cujo valor está entre 1 e M;
- k é um índice para o documento cujo valor está entre 1 e N;
- $(\text{tf.idf})_{i,k}$ é a relevância não normalizada do termo i ao document k;
- n_i é o número de documentos contendo o termo i;

O índice tf.idf normalizado é dado por:

$$dk_{i,k} = |\,(\text{tf.idf})_{i,k}\,| = \frac{(\text{tf.idf})_{i,k}}{\sqrt{\sum_{j=1}^{N}\left((\text{tf.idf})_{i,j}\right)^2}} \tag{2.7}$$

2.3.2 Indexação por Semântica Latente

A indexação por semântica latente, *Latent Semantic Indexing* - LSI, foi originalmente proposta por Dumais *et al.* [17] como uma alternativa para resolver os problemas dos modelos de recuperação

baseados em espaços vetoriais. É uma técnica estatística que busca extrair dos documentos a estrutura latente de utilização das palavras, que foi parcialmente oculta devido à variabilidade de palavras para descrever o mesmo tópico. A versão truncada da decomposição em valores singulares (SVD) é usada para estimar essa estrutura latente. A recuperação da informação é realizada utilizando os valores singulares e os vetores obtidos com a versão truncada do SVD.

A decomposição SVD transforma a matriz termo-documento $D_{kt\times d}$ no produto de três matrizes

$$D_{kt\times d} = T_{t\times n} S_{n\times n} D'_{d\times n} \tag{2.8}$$

onde t é o número de termos, d é o número de documentos, n é o mínimo entre t e d, T,D são matrizes ortogonais, isto é, $TT' = D'D = I$ e S é uma matriz diagonal com os autovalores da matriz D_k.

Esta decomposição pode ser vista como um método para se rotacionar os eixos de um espaço n-dimensional tal que o primeiro eixo aponta para a direção de maior variação, o segundo eixo para a direção de segunda maior variação, e assim por diante.

As matrizes T e D representam termos e documentos nesse novo espaço e a matriz diagonal S contém a variação em cada um dos eixos. Ao restringir as matrizes T, S e D a suas $k < n$ linhas mais importantes obtém-se uma versão aproximada da matriz D_k, mostrada na equação 2.9.

$$\hat{D}_{kt\times d} = T_{t\times k} S_{k\times k} D'_{d\times k} \tag{2.9}$$

A matriz $\hat{D_k}$ é a melhor aproximação quadrática da matriz D_k com posto k, segundo a equação:

$$\Delta = ||D_k - \hat{D}_k||_2 = \sum_{i=1}^{t}\sum_{j=1}^{d}\left(dk_{ij} - \hat{dk}_{ij}\right)^2 \tag{2.10}$$

Para os propósitos de recuperação, a consulta do usuário representada por q faz uso das matrizes T e S para ampliar a consulta q, transformando-a em $\hat{q}$.

$$\hat{q} = q' T_{t\times k} S^{-1}_{k\times k} \tag{2.11}$$

Inicialmente, os trabalhos de Dumais foram utilizados em sistemas de recuperação de informação, mas logo outros pesquisadores começaram a aplicar essa técnica para extrair relações semânticas entre os termos.

Govind *et al.* [51] utilizaram a decomposição SVD para construir um grafo bipartido que associa termos a conceitos. O grafo é construído a partir da matriz T e da lista de termos. Para cada vetor t_i da matriz T os termos com correlação menor que um certo limiar são eliminados aplicando-se zero na coordenada do vetor referente ao termo. Já o nome do conceito é calculado tomando-se os cinco termos de maior correlação e separando-os com vírgula.

Fortuna *et al.* [25] fazem uso de LSI para construção semi-automática de ontologias. Todo o processo de construção da ontologia é manual, porém o computador continuamente fornece sugestões para novos tópicos, auxilia na atribuição de categorias aos documentos e auxilia na aplicação de nomes aos tópicos.

As principais vantagens da técnica LSI são a capacidade de tratar casos de palavras sinônimas, filtragem de ruídos, e alguns casos de palavras polissêmicas. Outras vantagens incluem a existência de algoritmos determinísticos para a decomposição SVD, a decomposição ser única e a não necessidade de se recalcular a decomposição se o valor de k for alterado.

Dentre as desvantagens, podemos citar a restrição de ortogonalidade entre os vetores, resultando em conceitos que não representam corretamente assuntos abordados pelos documentos, a dificuldade para interpretar os valores presentes nos vetores de T e D, devido à mistura de valores positivos e negativos, e a necessidade de se especificar o número k de conceitos.

2.3.3 Fatoração em Matrizes Não-Negativas

Xu *et al.* [76] e Shahnaz *et al.* [65] também fazem uso da decomposição de matrizes para extrair relações semânticas. Em seus trabalhos, a matriz D_k é decomposta utilizando a técnica de NMF, *Non-Negative Matrix Factorization*. Essa forma de decomposição foi proposta inicialmente por Lee e Seung [48] e baseia-se na decomposição de uma matriz em produto de outras duas matrizes:

$$\hat{D}_{kt\times d} = W_{t\times k} H_{k\times d} \quad W \geq 0, \;\; H \geq 0 \tag{2.12}$$

Assim como em LSI, t, d e k são, respectivamente, número de termos, número de documentos e $k < \min(t, d)$ o posto da matriz D_k. A matriz W representa bases vetoriais que auxiliaram na composição da matriz D_k e a matriz H representa os coeficientes multiplicativos dos vetores-base. Ao contrário da técnica de LSI, as matrizes são sempre maiores ou iguais a zero, garantindo a interpretabilidade dos dados. Ainda, os vetores bases W não são necessariamente ortogonais de modo que resolve a outra limitação da técnica LSI.

Esta decomposição procura minimizar a diferença entre a matriz original D_k e a sua versão aproximada:

$$\min_{ij} \Delta = ||D_k - W \cdot H||_2 = ||D_k - \hat{D}_k|| = \sum_{i=1}^{t} \sum_{j=1}^{d} \left(dk_{ij} - \hat{dk}_{ij} \right)^2 \tag{2.13}$$

Além das vantagens em relação à técnica LSI apresentada anteriormente, os autores da proposta argumentam ainda que o desempenho para o processamento de consultas e de agrupamentos são comparáveis à técnica LSI, há economia de memória, pois as matrizes resultantes W e H são esparsas

e apresenta escalabilidade boa em relação aos parâmetros t, d e k.

As desvantagens da técnica estão associadas ao método de decomposição, pois a fatoração não é única, a decomposição $\hat{D_k} = WH$ também pode ser decomposta como $\hat{D_k} = WDD^{-1}H$ onde os novos valores W^* e H^* são $W^* = WD$ e $H^* = D^{-1}H$, incapacidade de se reduzir o tamanho da base vetorial sem realizar toda a decomposição novamente, e, atualmente, não existe algoritmo que garanta o ótimo global.

O algoritmo para a decomposição NMF é baseado na atualização iterativa das matrizes W e H até que o erro quadrático seja menor que um certo limiar δ, como segue:

$$H_{n+1} \leftarrow \frac{H_n \left(W_n^T D_{kn}\right)}{W_n^T W_n H_n + \epsilon} \tag{2.14}$$

$$W_{n+1} \leftarrow \frac{W_n \left(D_{kn} H_n^T\right)}{W_n H_n H_n^T + \epsilon} \tag{2.15}$$

O parâmetro ϵ é uma constante positiva próxima de zero para evitar a ocorrência de divisão por zero.

2.3.4 Taxonomia de Termos

Holger *et al.* [4] utilizam de forma diferente a técnica LSI e propuseram um algoritmo capaz de extrair relações taxonômicas entre termos. Diferentemente das propostas que procuram estabelecer índices de pertinência de termos a conceitos, a proposta de Holger avalia curvas de similaridade entre termos para inferir qual é a hierarquia entre eles. Assim como nas técnicas LSI e NMF, define-se uma matriz D_k e uma matriz $S = D_k^{'} D_k$ para avaliar a similaridade entre os termos. Estando a matriz D_k normalizada, os elementos da diagonal principal de S são todos 1 e o elemento genérico S_{ij} é o índice de similaridade entre os termos i e j.

A determinação do tipo de relação entre os pares de termos é feita analisando-se aproximações SVD de baixo posto da matriz de similaridade. Seja $S(k) = V(k)\Sigma(k)V(k)^T$ a melhor aproximação com posto k para a matriz S. A matriz $\Sigma(k)$ é diagonal contendo os k maiores autovalores e $V(k)$ é a matriz contendo os k autovetores associados ao autovalores de Σ.

Define-se a similaridade $\text{sim}_k(i, j)$ entre o i-ésimo e o j-ésimo termo na aproximação k como sendo a entrada $S(k)_{ij}$. Com o uso de curvas de similaridade, investiga-se como a semelhança entre os termos é modificada à medida que o valor k é alterado.

Curva de Similaridade. *A curva de similaridade de dois termos t_i e t_j é definida como sendo o*

gráfico da função s definida como:

$$s(k) = sim_k(i, j) = S(k)_{ij} \tag{2.16}$$

Um termo t é corretamente representado na aproximação de ordem k da matriz de similaridade somente se ele é mais similar a si próprio que a qualquer outro termo, isto é, $\text{sim}_k(t, t) \geq \text{sim}_k(t, t')$ para qualquer outro termo $t \neq t'$. O termo t é então dito ser válido no posto k.

Validade de posto. *Um termo t é otimamente representado na aproximação de ordem k da matriz de similaridade se k-1 for o maior valor para o qual o termo não é válido, ou seja, deixa de ser mais similar a si próprio do que a outro termo. A validade de posto será denotada por $k = rank(t)$.*

A extração da taxonomia entre os conceitos é feita utilizando a validade de posto dos termos. Um conceito c^* associado ao termo c é um sub-conceito do termo a se $\text{rank}(c) < \text{rank}(a)$ e se para algum rank k tal que $\text{rank}(c) \leq k < \text{rank}(a)$, a^* é mais similar a c^* do que c^* a si próprio.

2.4 Comparações entre as Técnicas

As técnicas apresentadas neste capítulo mostram duas metodologias diferentes para a extração de relações semânticas, a lingüística e a estatística. Cada uma das estratégias possui vantagens e desvantagens, sendo que o mais indicado é que ambas estejam presentes [31].

As estratégias baseadas em métodos estatísticos encontram dificuldades em relação à falta de esquemas adequados para estabelecer os pesos para termos compostos [73] e podem apresentar falhas em comparações simples entre itens lexicais devido a problemas de ambiguidade e de composicionalidade dos termos [3].

Por outro lado, as estratégias puramente lingüísticas apresentam dificuldades principalmente devido a: (a) complexidade dos algoritmos para a análise lingüística dos textos; (b) o conhecimento lingüístico ser muitas vezes aplicável em domínios específicos, restringindo a utilização em outros domínios; (c) o sucesso na extração de relações semânticas ser dependente de propriedades dos documentos, visto que os resultados não são bons quando documentos são constituídos por poucos termos; (d) os erros, quando ocorrem, trazerem prejuízos que não são compensados pelos benefícios decorrentes da sua aplicação; e (e) as técnicas lingüísticas (como categorização gramatical, resolução de ambigüidade, análise sintática, etc.) necessitarem ter alto grau de precisão para trazer benefícios. Muitas vezes, as técnicas não lingüísticas já exploram implicitamente o conhecimento lingüístico, de maneira que a contribuição dos métodos lingüísticos seria pouca. Entretanto, os resultados obtidos pelas técnicas lingüísticas são explícitas, seus facilmente interpretáveis.

Em ambos os métodos existem alguns problemas comuns associados as variações lingüísticas [66], tais como: (a) palavras diferentes podem assumir o mesmo significado, como “sapato” e “calçado”; (b) frases com as mesmas palavras, mas ordenadas de forma diferente podem possuir significados diferentes, como “vítima juvenil de crime” e “vítima de crime juvenil”; e (c) o mesmo item lexical pode assumir diferentes significados em contextos diferentes, como “agudo” na medicina e na geometria.

As desvantagens de cada método, apresentadas anteriormente, mostram que os estatísticos ainda são a melhor escolha para o desenvolvimento de sistemas de recuperação de informações ou de sistemas de extração de ontologias.

Os métodos lingüísticos baseados em padrões sintáticos são capazes de identificar relações de hiperonímia e hiponímia entre conceitos. No entanto, as relações nem sempre trazem informações significativas sobre os principais conceitos abordados nos documentos. Na maioria dos casos, os padrões são expressões soltas que não estão diretamente relacionados com o tema do documento.

As técnicas estatísticas que utilizam processamento léxico e contagem de palavras capturam de forma mais adequada os assuntos tratados pelos documentos. Os métodos via decomposição de matrizes são promissores e renderam duas patentes, 4.839.853 e 5.301.109, à *Bell Labs* e *Telcordia*. Já a técnica NMF trouxe outras melhorias aos métodos baseados em decomposição, eliminando as restrições de ortogonalidade entre os vetores conceitos que comprometiam o resultado final.

Os melhores resultados apresentados pelos métodos estatísticos foram decisivos para a escolha dessa técnica como alvo de pesquisa. Apesar de promissores, os métodos via decomposição matricial ainda não eram capazes de identificar corretamente os conceitos, e não permitiam a supervisão do procedimento de extração dos conceitos. A técnica proposta nesta dissertação incorpora noções de aprendizado de máquina com supervisão para extrair relações semânticas que representem mais fielmente os assuntos tratados pelos documentos.

2.5 Considerações Finais

Neste capítulo, foram apresentadas as técnicas para a extração de relações semânticas, algumas técnicas fazem mais uso de abordagens lingüísticas, Hearst e Maedche, e outras fazem mais uso de abordagens estatísticas, LSI e NMF. São apresentadas também as características de cada técnica, bem como suas vantagens e desvantagens. Foram apresentadas, também, a definição de ontologia e sua utilização em sistemas de recuperação de ontologias.

A técnica adotada nesta dissertação utiliza métodos lingüísticos para a identificação e classificação dos termos e métodos estatísticos para identificação de termos que estejam semanticamente próximos. Será definida uma estrutura matemática para representar o termo e, também, uma métrica para avaliar a correlação entre os termos. A informação sobre a correlação será utilizada pelo algo-

ritmo de agrupamento para identificar grupos de termos com maiores índices de correlação. Esses grupos de termos correlatos são os responsáveis pela caracterização dos conceitos. Assim, a teoria de aprendizado de máquina e algoritmos de agrupamento serão de auxílio na elaboração de um algoritmo capaz de identificar esses grupos. A revisão dos métodos mais conhecidos de agrupamento é apresentada no próximo capítulo (Cap. 3).

Capítulo 3

Modelos de aprendizado não supervisionado

Este capítulo apresenta uma revisão dos principais métodos de aprendizado não supervisionado baseados em técnicas de agrupamento particional. A apresentação dos métodos tem o objetivo de mostrar as características, vantagens e desvantagens de cada técnica, bem como a formulação do problema de agrupamento e a sua respectiva técnica de solução. As técnicas são apresentadas em ordem de complexidade e capacidade de agrupamento, iniciando-se com as propostas mais simples e finalizando com as propostas mais complexas, que incorporam a idéia de funções kernel, aprendizado supervisionado e bi-agrupamento (*biclustering*). A apresentação das técnicas tem por objetivo final auxiliar a escolha do melhor agrupador tendo em vista que o procedimento adotado para a extração de relações semânticas é baseado na identificação de termos correlatos. Assim, as funções-objetivo e as técnicas de resolução apresentadas neste capítulo servirão de base matemática para a elaboração do algoritmo proposto.

3.1 Técnicas de Agrupamento

Em todos os métodos apresentados, a estratégia para o processo de agrupamento é sempre a mesma: deve-se especificar uma função matemática que realiza o agrupamento de acordo com as características desejadas e escolhe-se um algoritmo de otimização que minimize essa função. A escolha das características do agrupamento deve levar em conta o tipo de aplicação e a disponibilidade de recurso computacional para a determinação dos grupos. Por exemplo, para aplicações que necessitam trabalhar com incertezas ou ruído na informação, recomenda-se a utilização de agrupadores fuzzy ou possibilístico. Já para aplicações em processamento de imagens que necessitam detectar bordas e contornos, recomenda-se a utilização de funções de kernel. Existem ainda casos em que alguma informação prévia está disponível, para os quais as técnicas de agrupamento semi-supervisionadas são as mais recomendadas.

Abaixo são apresentadas as variáveis utilizadas nas funções-objetivo bem como a sua definição matemática e o seu papel nas funções.

Definição das variáveis:

- $X = \{x_i | i = 1, ..., N\}$: conjunto de objetos. São os elementos alvo do processo de particionamento;
- x_i: é o i-ésimo objeto do conjunto de objetos. Cada objeto é representado por um vetor de dimensão D, onde as coordenadas representam as característica do objeto.
- x_{ij}: j-ésima característica (atributo) do objeto i, com $j = 1, \ldots, D$;
- $\beta = \{\beta_i | i = 1, \ldots, C\}$: representa o conjunto de vetores protótipos dos grupos;
- $d(x, \beta_i)$: função distância entre um objeto particular e o protótipo do grupo i;
- C: é o número de grupos;
- N: é o número de objetos a serem agrupados;
- S_i: é um grupo associado ao protótipo β_i;
- $u_{ij} \in [0, 1]$: representa o grau de pertinência de um objeto x_j ao grupo S_i.
- $m \in [1, \infty)$: índice fuzzy. Responsável por controlar o nível fuzzy do particionamento. Valores próximos de 1 resultam em um particionamento mais próximo do particionamento rígido e valores elevados de m resultam em um particionamento mais fuzzy;
- η_i: fator que pondera a importância do tamanho do i-ésimo grupo;
- Φ: função que realiza o mapeamento em dimensões mais elevadas;
- $K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$: função kernel.

Os algoritmos de otimização para a minimização das funções-objetivo são, em sua grande maioria, baseados em métodos numéricos onde as pertinências e os protótipos de grupos são atualizados em cada iteração do algoritmo até que uma dada condição seja satisfeita. O algoritmo adaptado para o caso de agrupamento fuzzy é apresentado em 1.

3.2 Métodos Tradicionais

Os métodos tradicionais de agrupamento referem-se às primeiras e mais utilizadas técnicas de agrupamento, tais como o K-Means, Fuzzy C-Means e o Possibilistic C-Means. Essas técnicas são baseadas em algoritmos de máxima verossimilhança para misturas gaussianas, uma vez que procuram encontrar o centro natural do agrupamento dos dados e favorecem a formação de grupos esféricos ou elipsoidais. Ainda, são métodos que assumem que os atributos estão no formato vetorial e o número

Input: C: Número de grupos
Input: $t_{\max}$: Número máximo de iterações
Input: $m > 1$: Índice Fuzzy
Input: $\epsilon > 0$: Delta mínimo
Output: u_{ik}: Pertinências
begin
  Inicia os valores de pertinências u_{ik}^0 com valores aleatórios
  $C \leftarrow C_0$
  CondiçãoFinal $\leftarrow$ falso
  for $t = 1$ *até* $t_{\max}$ **do**
    Atualize protótipos v_i^t utilizando as pertinências u_{ik}^{t-1}
    Atualize pertinências u_{ik}^t utilizando os protótipos v_i^{t-1}
    Calcule o erro $E^t = \max |u_{ik}^t - u_{ik}^{t-1}|$
    if $E^t \leq \epsilon$ **then**
      Pare o processamento
    end
    else
      $t \leftarrow t + 1$
    end
  end
end

Algoritmo 1: Algoritmo de Agrupamento Fuzzy

de grupos é especificado previamente. O objetivo desses agrupadores é minimizar a variação intra-grupo ou erro quadrático. Existem, ainda, técnicas de agrupamento que realizam a minimização da variação intra-grupo e/ou a maximização da variação inter-grupo tal como proposto por Horng *et al.* [40].

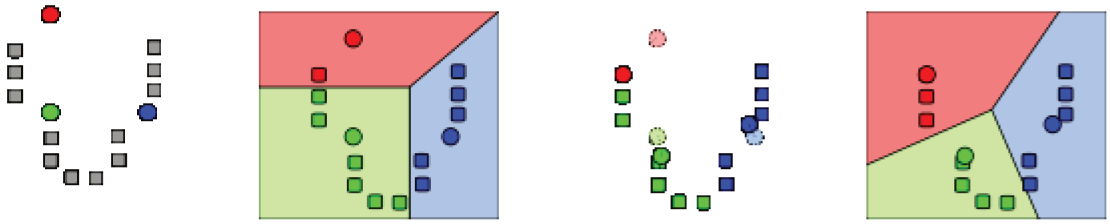
3.2.1 K-Means

K-Means foi o primeiro e o mais famoso método de agrupamento, desenvolvido por Hartigan em 1975 [38]. Trata-se de um método no qual um conjunto de objetos é particionado em um conjunto fixo de grupos de maneira a formar grupos naturais, ou seja, de mínima variação intra-grupo. A equação 3.1 apresenta a função que deve ser minimizada para a obtenção da mínima variação intra-grupo:

$$J(U, V) = \sum_{i=1}^{C} \sum_{x_j \in S_i} (x_j - \beta_i)^2. \tag{3.1}$$

A forma de solução proposta para este problema é baseada no algoritmo de Lloyd [49] e está ilustrada na figura 3.1. O conjunto de objetos de entrada é particionado em C conjuntos iniciais;

esta partição pode se valer de alguma heurística ou simplesmente serem escolhidos aleatoriamente. É calculado o centróide de cada partição (a). Uma nova partição é construída associando-se cada objeto ao centróide mais próximo (b). Os centróides são atualizados para o centro de cada grupo (c), e o algoritmo repete a aplicação alternada dos procedimentos (b) e (c) até atingir a convergência (d), a qual é obtida quando os centróides não são mais alterados.



(a) Inicialização aleatória dos centróides

(b) Objetos são associados ao centróide mais próximo

(c) Atualização dos centróides para o centro de cada grupo

(d) Passo b e c são repetidos até a convergência

Fig. 3.1: Demonstração do algoritmo K-Means

O agrupador K-Means apresenta como principais características o favorecimento de grupos de formato esférico e o particionamento rígido dos grupos, ou seja, cada objeto pertence integralmente a um e somente um grupo.

As vantagens desse agrupador são a simplicidade matemática, a rápida convergência do algoritmo de minimização e escalabilidade em relação aos dados de entrada. Apesar da simplicidade, o método apresenta muitas desvantagens, pois não possui imunidade a ruídos, é necessário especificar o número de grupos, favorece a formação de grupos esféricos e, principalmente, o algoritmo de minimização pode estar sujeito a mínimos locais.

3.2.2 Fuzzy C-Means

Fuzzy C-Means é um método de agrupamento desenvolvido por Dunn [18] e posteriormente melhorado por Bezdek [7] que, diferentemente da técnica K-Means, permite que objetos pertençam a dois ou mais grupos, sendo que o grau de pertinência varia de 0 a 1. A técnica consiste na minimização da função objetivo 3.2 e é sujeita a três restrições, mostradas em 3.3:

$$J(U,V) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m}(x_j - \beta_i)^2, \tag{3.2}$$

$$\text{s.a. } M_{\text{prb}} = \{(u_{ij}) : \sum_{i=1}^{C} u_{ij} = 1, u_{ij} \in [0,1], \forall i,j\} \tag{3.3}$$

A restrição M_{prb} é denominada de partição probabilística [56]. A otimização da função objetivo J é realizada empregando-se a técnica de atualização sucessiva de centros e pertinências, como descrito no algoritmo da seção 3.1. As funções de atualização de pertinências e protótipos de grupos são dadas por:

$$\beta_i = \frac{\sum_{j=1}^{N} u_{ij}^m x_j}{\sum_{j=1}^{N} u_{ij}^m}, \;\; u_{ij} = \left(\sum_{k=1}^{C} \left(\frac{d(x_j, \beta_i)}{d(x_j, \beta_k)} \right)^{\frac{2}{m-1}} \right)^{-1}.$$

A principal característica do agrupador Fuzzy C-Means está no particionamento fuzzy dos grupos, sendo que os objetos podem pertencer a mais de um grupo. Assim como no K-Means, existe o favorecimento de grupos de formato esférico e é necessário especificar o número de grupos.

O agrupador Fuzzy C-Means ainda apresenta um modelo matemático simples e de rápida convergência, mas sua maior vantagem é a imunidade a ruídos. As desvantagens ainda são o favorecimento de grupos esféricos, a pré-especificação do número de grupos e algoritmo de minimização sujeito a mínimos locais.

3.2.3 Possibilistic C-Means

Possibilistic C-Means (PCM) foi proposto por Krishnapuram e Keller [43] para contornar o problema de interpretabilidade dos valores de pertinência que são obtidos ao se utilizar o agrupador Fuzzy C-Means. A restrição $\sum_{i=1}^{C} u_{ij} = 1$ faz com que o agrupador FCM produza valores de pertinência u_{ij} que podem ser interpretados como grau de compartilhamento entre grupos e não graus de tipicalidade.

A figura 3.2 ilustra os problemas do agrupador fuzzy. O agrupador é obrigado a realizar uma distribuição dos valores de pertinência para manter a restrição anterior, de modo que a pertinência de um objeto O a um grupo G seja proporcional à distância desse objeto ao grupo G e inversamente proporcional à soma de todas as distâncias desse objeto aos grupos. Como conseqüência, pontos como o A e o B, demarcados na figura, apresentarão valores de pertinência iguais a 0.5 em relação aos dois grupos presentes na figura, pois estão equidistantes dos grupos. Contudo, esse não é o melhor valor de pertinência para os pontos A e B. Claramente, o ponto B está muito distante dos grupos e deveria receber um valor de pertinência próximo de 0 em relação aos grupos. Por outro lado, o ponto A deveria receber um valor de pertinência maior que 0.5 dado que está muito próximo de ambos os grupos. O mesmo problema não ocorre no agrupador possibilístico, pois o valor de pertinência é proporcional somente à distância entre o objeto e o grupo.

A solução para os problemas anteriores foi a remoção da restrição. Porém, foi necessário adicionar um novo termo à função-objetivo, $\sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} (1 - u_{ij})^m$, para evitar a ocorrência da solução trivial

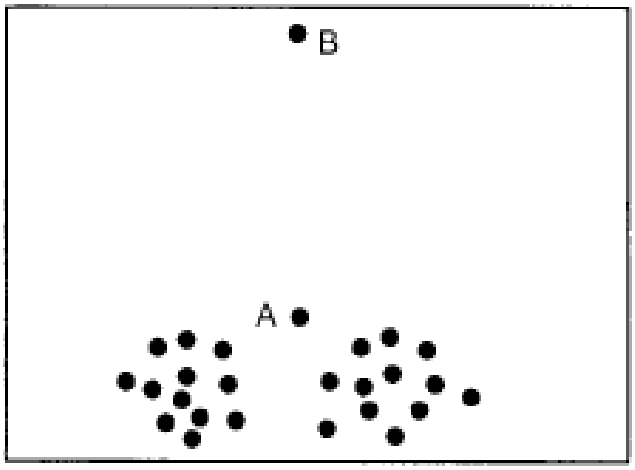


Fig. 3.2: Problemas do Agrupador Fuzzy

na qual todas as pertinências são nulas, $u_{i,j} = 0$. A função objetivo e suas restrições são mostradas abaixo:

$$J(U,V) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m}(x_j - \beta_i)^2 + \sum_{i=1}^{C}\eta_i\sum_{j=1}^{N}(1-u_{ij})^m, \tag{3.4}$$

$$\text{s.a. } M_{\text{pos}} = \{(u_{ij}) : 0 < \sum_{i=1}^{C} u_{ij} < C, u_{ij} \in [0,1], \forall i,j\}$$

A restrição M_{pos} é denominada de partição possibilística [56]. As características do agrupador Possibilistic C-Means são: promover o particionamento possibilístico dos grupos, ou seja, objetos podem pertencer a mais de um grupo, favorecimento de grupos de formato esférico e, diferentemente do agrupador FCM, as pertinências representam grau de tipicalidade e não de compartilhamento. O efeito desta mudança de paradigma é que os valores de pertinência representam de forma mais realística o grau de proximidade entre os objetos e os grupos.

As vantagens do agrupador possibilístico em relação à técnica FCM são sua maior imunidade a ruídos e habilidade em detectar grupos com grandes sobreposições. A desvantagem ainda é a identificação de grupos esféricos e algoritmo de minimização sujeito a mínimos locais.

Assim como em FCM, o algoritmo de otimização é baseado em atualização sucessiva dos protótipos de grupos e pertinências e são dadas pela equação 3.5:

$$\beta_j = \frac{\sum_{j=1}^{N} u_{ij}^m x_j}{\sum_{j=1}^{N} u_{ij}^m},\ u_{ij} = \left(1 + \left(\frac{|x_j - \beta_i|}{\eta_i}\right)^{\frac{1}{m-1}}\right)^{-1}. \tag{3.5}$$

O parâmetro η_i é um fator de escala que influencia diretamente no tamanho dos grupos obtidos. Quanto maior for o valor desse parâmetro, maior será o grupo obtido. Os casos em que $\eta_i = 0$ e $\eta_i = \infty$ resultarão em grupos que não possuem objetos e grupos que possuem todos os objetos, respectivamente. Esta constatação serve de base para a seção 4.4.7 sobre a identificação de relações taxonômicas.

No caso do agrupador possibilístico, esse parâmetro precisa ser previamente especificado. No entanto, alguns autores propuseram meios de estimar esse valor baseado na estatística da distância entre os objetos do conjunto. A estimação desse parâmetro é mostrada na equação 3.6:

$$\eta_i = \frac{\sum_{k=1}^{N} u_{ik}^m |x_k - \beta_i|}{\sum_{k=1}^{N} u_{ik}^m}. \tag{3.6}$$

Krishnapuram e Keller, em outro artigo [44], propõem uma forma diferente para a identificação de grupos. Nessa proposta, os grupos estão desacoplados de forma que a função-objetivo 3.4 torna-se C equações objetivo, uma para cada grupo, na forma:

$$J_i(U, V) = \sum_{j=1}^{N} \mu_{ij}^m (x_i - \beta_i)^2 + \eta_i \sum_{j=1}^{N} (1 - u_{ij})^m.$$

Esta mudança na forma de tratar o problema permite a introdução de métodos iterativos para a identificação de grupos e ainda métricas para avaliar o número adequado de grupos.

3.2.4 Gustafson Kessel

O algoritmo de Gustafson Kessel [35] baseia-se no agrupador FCM e introduz um mecanismo para a detecção de grupos elípticos de diferentes tamanhos e orientações. Esta característica é atingida modificando-se a métrica de distância entre os objetos. A métrica euclidiana $||x||_2$ foi alterada para a métrica de Mahalonabis $||x||_A$, onde A é a matriz que define a orientação e tamanho do elipsóide. A função objetivo é dada pela equação 3.7:

$$J(U, V, A) = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{i,j}^m \left(x_i - \beta_j\right)^T A_j \left(x_i - \beta_j\right), u_{i,j} \in [0, 1], 1 < m < \infty \tag{3.7}$$

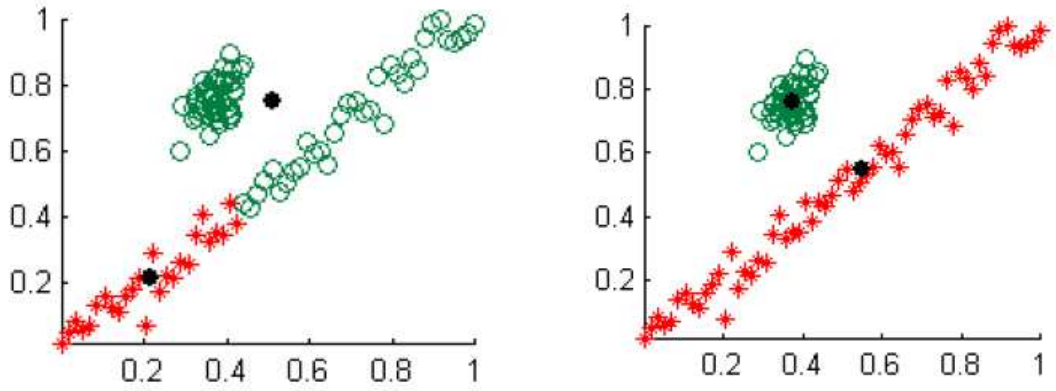
onde A_j é uma matriz simétrica, definida positiva, e é uma matriz de covariância para o grupo j. Se $A_j = I_D$, a expressão 3.7 será igual à expressão 3.2, que utiliza a distância euclidiana. A medida de distância entre os objetos é dada por:

$$d_{i,j} = \left(x_i - \beta_j\right)^T A_j \left(x_i - \beta_j\right), \quad i = 1, 2, \ldots, C, \quad k = 1, 2, \ldots, N.$$

e a matriz A_j é determinada por: $A_j = \left[\det(F_j)^{\frac{1}{n+1}} F_j^{-1}\right]$, sendo F_j calculada utilizando a equação 3.8:

$$F_j = \frac{\sum_{k=1}^{N} [u_{i,j}]^m (x_i - \beta_j)(x_i - \beta_j)^T}{\sum_{k=1}^{N} [u_{i,j}]^m}, \quad \forall i, i = 1, 2, \ldots, C. \tag{3.8}$$

A figura 3.3 mostra a diferença entre os agrupadores FCM e GK. Claramente o melhor agrupamento é dado por dois grupos, o primeiro agrupa os pontos que estão em linha reta e o segundo agrupa os pontos que estão acima desta reta. O agrupador FCM não foi capaz de identificar os grupos dessa forma, realizando uma partição que agrupou pontos mutuamente próximos. Já o agrupador GK identificou corretamente estes dois grupos, uma vez que o primeiro grupo pode ser corretamente representado por uma elipse onde um dos raios é bem pequeno comparado ao outro.



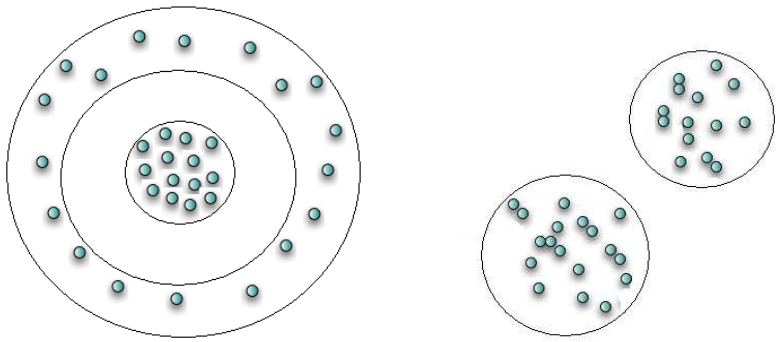
(a) Fuzzy C-Means (b) Gustafson Kessel

Fig. 3.3: Diferença no agrupamento FCM e GK

3.3 Métodos com função kernel

Os métodos de agrupamento com funções kernel foram desenvolvidos de maneira a identificar e representar padrões mais complexos e excluir padrões espúrios. Eles são baseados na abordagem de vetores suporte, *support vector clustering* - SVC.

Em SVC, os objetos são mapeados de um espaço de dados para um espaço de características de elevada dimensão utilizando uma função de mapeamento Φ. No espaço de características, figura 3.4(b), procura-se pelas menores hiper-esferas que cobrem a imagem dos dados. Essas hiper-esferas, quando mapeada de volta ao espaço de dados, figura 3.4(a), formam um conjunto de contornos os quais envolvem os objetos. Estes contornos são interpretados como grupos, e os objetos envolvidos em cada contorno são associados ao mesmo grupo.



(a) Dados originais (b) Mapeamento

Fig. 3.4: Diferença no agrupamento ao utilizar-se a função kernel

A introdução desse mapeamento resolve o problema de identificação de padrões complexos mas introduz dois inconvenientes. O primeiro é relativo à complexidade computacional e o outro é devido ao número excessivo de dimensões que o mapeamento pode acrescentar. Felizmente, estes dois problemas podem ser facilmente resolvidos com a propriedade das funções kernel. Considere a função-objetivo 3.9 do agrupador FCM com as funções de mapeamento:

$$J(U,V) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m}||\Phi(x_j) - \Phi(\beta_i)||^2. \tag{3.9}$$

O termo referente ao módulo pode ser escrito como o produto escalar dos vetores Φ. Assim:

$$J(U,V) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m}\langle\Phi(x_j) - \Phi(\beta_i), \Phi(x_j) - \Phi(\beta_i)\rangle.$$

A expansão do produto interno resulta em uma função-objetivo com três parcelas:

$$J(U,V) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m}\left(\langle\Phi(x_j), \Phi(x_j)\rangle - 2\langle\Phi(x_j), \Phi(\beta_i)\rangle + \langle\Phi(\beta_i), \Phi(\beta_i)\rangle\right).$$

O desenvolvimento acima mostra que a dimensionalidade do espaço F não é importante e nem o mapeamento Φ. Somente o produto interno entre os vetores do espaço F. Define-se, então, a função kernel.

Definição de função kernel. *Uma função kernel é uma função K tal que para todo $x, y \in X$ tem-se $K(x,y) = \langle\Phi(x), \Phi(y)\rangle$ onde Φ representa o mapeamento de X para um espaço característico F.*

Substituindo a função kernel na função-objetivo obtém-se:

$$J(U,V) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m}\left(K(x_j,x_j) - 2K(x_j,\beta_i) + K(\beta_i,\beta_i)\right). \tag{3.10}$$

A expressão 3.10 mostra que a única informação necessária é a especificação da função kernel. Esta função deve ser escolhida levando-se em conta quais são os formatos de grupos que se deseja identificar e, também, deve atender ao critério de Mercer [30].

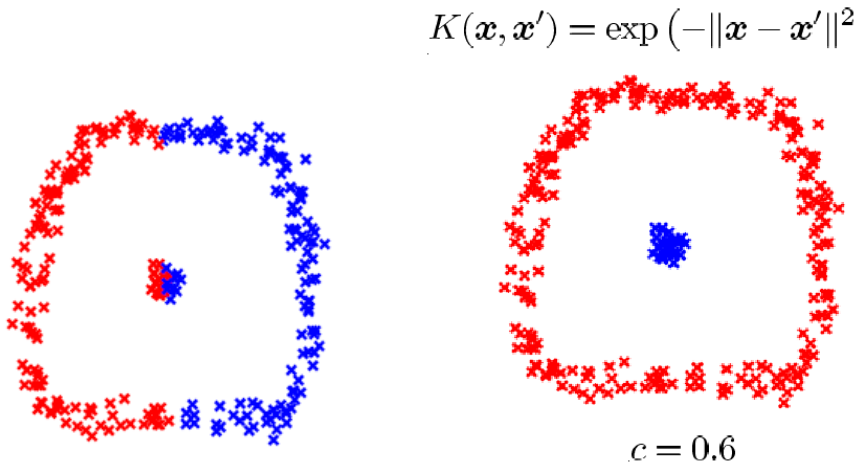
Exemplos de função kernel:

- Função kernel polinomial: $K(x,y) = \langle x,y\rangle^d$

A função polinomial é adequada para problemas em que os objetos alvos do agrupamento estejam normalizados.

- Função kernel radial: $K(x,y) = \exp(-\gamma||x-y||^2)$

A função kernel radial é adequada para a detecção de fronteiras, como ocorre no processo de agrupamento mostrado na figura 3.5. Claramente, é possível observar a existência de dois grupos de pontos contíguos, um grupo localizado no centro e outro grupo que circunda o centro. Dessa forma, espera-se que o agrupador identifique estes dois grupos. Conforme apresentado na figura 3.5(a), o agrupador Fuzzy C-Means não é capaz de realizar esse particionamento, pois não é possível separar os objetos utilizando somente hiper-esferas. Já a versão kernel desse agrupador, figura 3.5(b), é capaz de agrupar corretamente os pontos, pois realiza um mapeamento dos pontos para um outro espaço na qual é possível a utilização de hiper-esferas para separar corretamente os pontos.



(a) Agrupamento convencional (b) Agrupamento com kernel radial

Fig. 3.5: Diferença no agrupamento ao utilizar-se a função kernel

Funções de kernel tem sido aplicadas com sucesso em agrupadores fuzzy e, também, no possibilístico. A função objetivo de um agrupador possibilístico com função de kernel é mostrada em 3.11:

$$J(U,V) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m}||\Phi(x_j) - \Phi(\beta_i)||^2. \tag{3.11}$$

Observe que as restrições em relação aos valores de pertinência permanecem as mesmas do agrupador possibilístico convencional.

3.4 Métodos com supervisão parcial

Métodos de agrupamento com mecanismo de supervisão parcial foram propostos por Pedrycz em seu livro sobre agrupamento fuzzy baseado em conhecimento [60]. A supervisão parcial envolve o conhecimento prévio do subconjunto de dados rotulados e seus valores de pertinência às classes. Esse conhecimento é incluído na função-objetivo de maneira a refletir alguns dos padrões que foram rotulados. O conhecimento servirá de âncora que guiará a descoberta das estruturas no conjunto de dados. A função-objetivo de agrupamento fuzzy é expandida de forma a levar em consideração o conhecimento prévio. A formulação proposta para a função-objetivo é dada por:

$$Q = \sum_{i=1}^{C}\sum_{k=1}^{N} u_{ik}^{m} d_{ik}^2 + \alpha \sum_{i=1}^{C}\sum_{k=1}^{N} \left(u_{ik} - f_{ik} b_k\right)^2 d_{ik}^2. \tag{3.12}$$

O primeiro termo da expressão é responsável por descobrir as estruturas no conjunto de dados, assim como no FCM padrão. Já o segundo termo reflete o efeito da supervisão parcial.

Nessa equação, duas novas estruturas foram adicionadas:

- O vetor de rótulos $b = [b_1 b_2 \ldots b_k \ldots b_N]^T$. Cada coordenada b_k deste vetor assume um valor binário indicando se o padrão x_k foi rotulado ($b_k = 1$) ou não rotulado ($b_k = 0$);
- A matriz de partição $F = [f_{ik}]$, $i = 1, 2, \ldots, c$; $k = 1, 2, \ldots, N$, que contém a pertinência dos padrões selecionados.

O parâmetro α regula o balanço entre o modo de aprendizado supervisionado e o modo não-supervisionado. Para $\alpha = 0$, a função-objetivo converge para aquela do FCM padrão; para α elevado, o segundo termo da função torna-se mais importante e o agrupamento torna-se cada vez mais supervisionado.

A função de atualização de pertinências e centros de grupos são dadas pelas seguintes expressões:

$$u_{ik} = \frac{1}{1+\alpha}\left[\frac{1 + \alpha\left(1 - b_k \sum_{i=1}^{c} f_{ik}\right)}{\sum_{j=1}^{c}\left(\frac{d_{ik}}{d_{jk}}\right)^2} + \alpha f_{ik} b_k\right], \quad \beta_j = \frac{\sum_{k=1}^{N} \omega_{jk} x_k}{\sum_{k=1}^{N} \omega_{jk}}, \tag{3.13}$$

onde

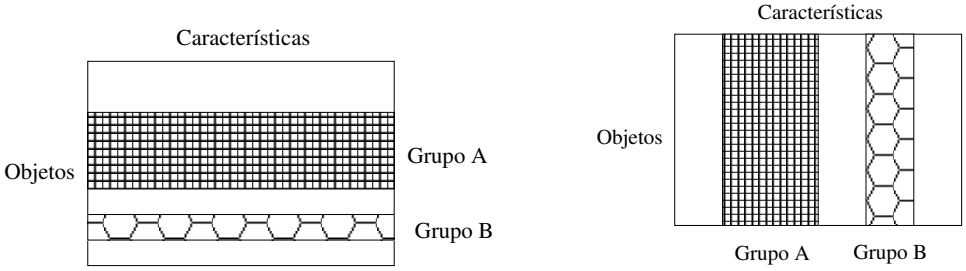
$$\omega_{ik} = u_{ik}^2 + (u_{ik} - f_{ik} b_k)^2 \,. \tag{3.14}$$

3.5 Métodos de Bi-agrupamento

Bi-agrupamento é uma técnica de mineração de dados que, diferentemente das técnicas tradicionais, busca o agrupamento simultâneo das colunas (características) e linhas (objetos) de uma matriz de maneira a se obter agrupamento de objetos que sejam similares sob algum aspecto. O termo foi introduzido por Mirkin [55] devido a sua necessidade de analisar expressões gênicas para determinar quais eram as condições em que alguns genes comportavam-se de forma semelhante.

Em bi-agrupamento o objetivo é gerar subconjuntos de objetos que exibem comportamento similar ao se analisar subconjuntos das características, ou vice-versa. Os subconjuntos de objetos e características podem ser representados por sub-matrizes. Essas sub-matrizes podem assumir diferentes formas, estarem sobrepostas e, até mesmo, serem não contíguas. O número de sub-matrizes também é variável, depende da configuração da matriz original.

A figura 3.6 apresenta dois tipos peculiares de bi-agrupamento. A figura 3.6(a) apresenta um agrupamento convencional, na qual objetos são agrupados de acordo com a sua similaridade. A figura 3.6(b), por outro lado, apresenta o agrupamento de características que sejam similares em objetos.

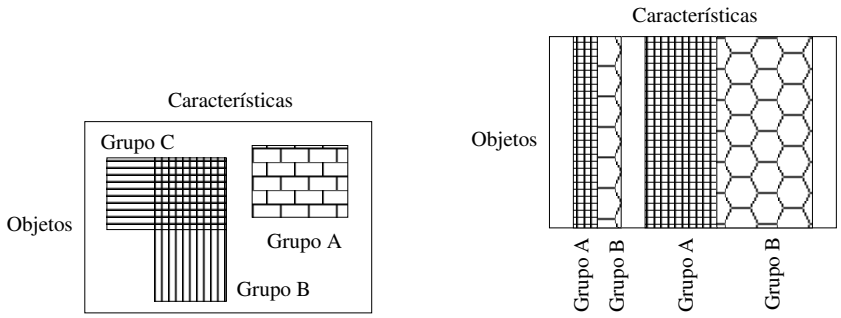


(a) Grupo de Objetos (b) Grupo de Características

Fig. 3.6: Sub-Matrizes de Objetos e Su-Matrizes de Características

É possível a existência de sub-matrizes sobrepostas como apresentado na figura 3.7(a) e sub-matrizes não contíguas como apresentado na figura 3.7(b).

Os diferentes algoritmos de bi-agrupamento podem trabalhar com diferentes definições para o que vem a ser um grupo. As principais definições de grupo são:



(a) Sub-Matrizes sobrepostas (b) Sub-Matrizes descontíguas

Fig. 3.7: Sub-Matrizes sobrepostas e Sub-Matrizes descontíguas

- Grupo com valor constante: A submatriz que representa um grupo possui valor constante em colunas e linhas:

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix} \tag{3.15}$$

- Grupo com valor constante nas linhas: A submatriz que representa um grupo possui valor constante em linhas e as colunas contêm um efeito aditivo:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix} \tag{3.16}$$

- Grupo com valor constante nas colunas: A submatriz que representa um grupo possui valor constante em colunas e as linhas contêm um efeito aditivo:

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix} \tag{3.17}$$

Atualmente, existem diversas técnicas para a solução do problema de bi-agrupamento: CTWC [29], Plaid Model [46], σ-biclusters [12], Spectral Biclustering [41] e ITWC [5]. Tanay et al. [70] apresen-

tam uma análise da maioria dos algoritmos citados anteriormente. As próximas seções detalham dois algoritmos na área de biclusterização.

3.5.1 Método de Cheng e Church

Cheng e Church [12] introduziram o bi-agrupamento ao tratar o problema de análise de expressão gênica. O algoritmo proposto por eles trata o bi-agrupamento como um problema de otimização, onde cada grupo possui uma pontuação que indica a qualidade do grupo. A função-objetivo é definida de maneira a forçar a uniformidade da matriz e dar lugar à formação de grupos grandes. Nos trabalhos de Cheng e Church, é assumido que um par (gene,condição) é um "bom" grupo se a submatriz associada apresenta valores uniformes, ou ainda valores uniformes com efeitos aditivos em colunas e/ou linhas.

A função objetivo proposta por Cheng e Church procura reduzir o resíduo quadrático médio dado por:

$$H(I,J) = \sum_{i \in I, j \in J} \frac{RS_{ij}^2}{|I||J|} \tag{3.18}$$

onde:

I: subconjunto de genes (objetos)

J: subconjunto de condições (características)

e_{ij}: elemento ij da submatriz

e_{IJ}: média da submatriz dado por $\sum_{i \in I, j \in J} e_{ij}/|I||J|$

e_{iJ}: média do subconjunto de colunas dado por $\sum_{j \in J} e_{ij}/|J|$

e_{Ij}: média do subconjunto de linhas dado por $\sum_{i \in I} e_{ij}/|I|$

$RS_{IJ}(i,j)$: resíduo da submatriz dado por $e_{ij} - e_{Ij} - e_{iJ} + e_{IJ}$

O procedimento de minimização da função H é realizado por um algoritmo guloso que iterativamente remove e adiciona linhas/colunas do grupo, procurando minimizar a função $H(I,J)$. Esse algoritmo é mostrado em 2.

Os resultados relatados por Cheng e Church mostraram que a técnica foi capaz de identificar corretamente a maioria dos grupos, tendo um desempenho similar aos métodos mais conhecidos, tais como o de Alizadeh [2]. No entanto, o algoritmo proposto é baseado em busca gulosa que não é capaz de garantir o melhor resultado em todos os casos de bi-agrupamento.

3.5.2 Método de Lazzeroni e Owen

O modelo Plaid [46] é uma abordagem inspirada na modelagem estatística desenvolvida por Lazzeroni e Owen para a análise de expressões gênicas. A idéia básica é representar a matriz de genes-

Input: U: Condições
Input: V: Genes
Input: E: Matriz de expressão gênica
Input: σ: Resíduo quadrático médio máximo
Output: L: Lista de Grupos
begin
  Inicialize o grupo (I, J) com $I = U$, $J = V$;
  $C \leftarrow C_0$
  CondiçãoFinal $\leftarrow$ falso
  while $H > \sigma$ **do**
    remova colunas/linhas que mais reduzem o valor de H;
    adicione colunas/linhas que não aumentem o valor de H;
  end
  Armazena o grupo encontrado em L;
  Mascara o grupo com valores aleatórios;
  Repete o procedimento para encontrar outros grupos;
end

Algoritmo 2: Algoritmo de Cheng-Church

condições como sendo a composição de submatrizes. Essas submatrizes são os grupos na metodologia de Lazzeroni e Owen e são caracterizadas por possuírem valores nulos nas posições relativas aos gene ou condições não pertencentes ao grupo. O modelo assume que a matriz objetos-características será aproximada pela soma de uma matriz uniforme mais um conjunto de K submatrizes, onde cada submatriz está associada a um determinado grupo. Matematicamente, a aproximação da matriz objetos-características é dada por:

$$A_{ij} = \mu_0 + \sum_{k=1}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk}$$

O parâmetro μ_0 é a matriz constante e $\theta_{ijk} = \mu_k + \alpha_{ik} + \beta_{jk}$. O parâmetro μ_k descreve a constante adicionada ao grupo k, α e β são as constantes aditivas de linha e coluna no grupo k. $\rho_{ik} \in \{0, 1\}$ é o indicador de pertinência do objeto i ao grupo k e $\kappa_{jk} \in \{0, 1\}$ é o indicador de pertinência da característica j ao grupo k. E K é o número de grupos, ou submatrizes, do particionamento

O problema de bi-agrupamento neste modelo é reduzir o erro quadrático entre a matriz A e a sua aproximação. Assim:

$$\min_{i,j} J = \min_{i,j} \sum_{ij} \left[A_{ij} - \sum_{k=0}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk} \right]^2.$$

Lazzeroni e Owen resolveram este problema de minimização com uma heurística iterativa. Na

proposta deles, $K-1$ grupos são mantidos fixos e procura-se pelo k-ésimo grupo que minimize a soma dos erros quadráticos. Seja:

$$Z_{ij}^{(K-1)} = A_{ij} - \sum_{k=0}^{K-1} \theta_{ijk}\rho_{ik}\kappa_{jk}$$

a matriz residual após a remoção dos primeiros $K-1$ grupos. Então, na iteração K deseja-se resolver o seguinte problema de programação inteira:

$$Q^{(K)} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(Z_{ij}^{(K-1)} - \theta_{ijK}\rho_{iK}\kappa_{jK}\right)^2$$

$$\text{s.a. } \sum_i \rho_{iK}^2\alpha_{iK} = 0,\ \sum_j \kappa_{jK}^2\beta_{jK} = 0$$

$$\rho_{iK} \in \{0,1\},\ \kappa_{jK} \in \{0,1\}$$

onde n é o número de linhas e p é o número de colunas das matrizes Q, Z e A.

A solução para este problema é também iterativa. Determina-se inicialmente o valor das variáveis para $K=1$, depois calcula-se o valor das variáveis para $K=2$ utilizando o resultado obtido com $K=1$, depois para $K=3$ utilizando o resultado obtido com $K=1$ e $K=2$, e assim por diante.

A determinação dos valores das variáveis em cada iteração segue a mesma metodologia de resolução para os agrupadores: as variáveis são inicializadas de maneira aleatória e procede-se com a atualização alternada de suas variáveis.

Primeiramente, calcula-se os valores das variáveis μ_K, α_{iK}, β_{jK}:

$$\mu_K = \frac{\sum_i\sum_j \rho_{iK}\kappa_{jK}Z_{ij}^{K-1}}{\left(\sum_i \rho_{iK}^2\right)\left(\sum_j \kappa_{jK}^2\right)}$$

$$\alpha_{iK} = \frac{\sum_j \left(Z_{ij}^{(K-1)} - \mu_K\rho_{iK}\kappa_{jK}\right)\kappa_{jK}}{\rho_{iK}\sum_j \kappa_{jK}^2}$$

$$\beta_{jK} = \frac{\sum_i \left(Z_{ij}^{(K-1)} - \mu_K\rho_{iK}\kappa_{jK}\right)\rho_{iK}}{\kappa_{jK}\sum_i \rho_{iK}^2}$$

Em seguida, calcula-se os valores das variáveis ρ_{iK} e κ_{jK}:

$$\rho_{iK} = \frac{\sum_j \theta_{ijK}\kappa_{jK}Z_{ij}^{(K-1)}}{\sum_j \theta_{ijK}^2\kappa_{jK}^2}$$

$$\kappa_{jK} = \frac{\sum_i \theta_{ijK}\rho_{iK}Z_{ij}^{(K-1)}}{\sum_i \theta_{ijK}^2\rho_{iK}^2}$$

Da mesma forma que os métodos de agrupamento, a técnica de bi-agrupamento *plaid model* é uma ferramenta para explorar padrões em conjunto de dados. Por essa razão, apresenta as mesmas dificuldades quando deve tratar dados com ruídos, grupos sobrepostos, sensibilidade ao escalamento dos dados e determinação do número de grupos.

3.6 Comparações entre os Agrupadores

Os agrupadores apresentados neste capítulo foram organizados de maneira a mostrar a evolução das técnicas de agrupamento. Iniciamos com a proposta pioneira K-Means, que apresenta como principal vantagem a simplicidade matemática, e finalizamos com a apresentação de técnicas de bi-agrupamento que têm sido aplicados com sucesso em diversas áreas, tais como, identificação de padrões gênicos, mineração de dados e mineração de textos.

A tabela 3.1 realiza um comparativo entre as técnicas apresentadas. Pode-se verificar que de todos os agrupadores particionais, o mais completo é o Kernel Possibilistic C-Means. No entanto, é o agrupador que exige o maior número de parâmetros e, também, o de maior complexidade. Para os bi-agrupadores, a exigência é a especificação de apenas dois parâmetros e o resultado do processo de agrupamento são sub-matrizes que agrupam objetos que sejam similares sob algum aspecto.

Agrupador	*Categoria*	*Imunidade a Ruído*	*Formato do grupo*	*Parâmetros de entrada*
K-Means	Agrupador Particional	não	esférico	C
Fuzzy C-Means	Agrupador Particional	sim	esférico	C,m,ϵ
Possibilistic C-Means	Agrupador Particional	sim	esférico	C,m,η,ϵ
Gustafson-Kessel	Agrupador Particional	sim	elíptico	C,m,ϵ
Kernel Fuzzy C-Means	Agrupador Particional	sim	depende da função de kernel	C,m,ϵ,$K(x_i, x_j)$
Kernel Possibilistic C-Means	Agrupador Particional	sim	depende da função de kernel	C,m,η,ϵ,$K(x_i, x_j)$
Church e Cheng	Bi-agrupamento	não	sub-matrizes	σ,ϵ
Plaid Models	Bi-agrupamento	não	sub-matrizes	K,ϵ

Tab. 3.1: Comparações entre agrupadores.

O parâmetro K representa o número de grupos extraídos pela técnica Plaid Models. Já o parâmetro σ é o resíduo quadrático médio máximo da proposta de bi-agrupamento de Church e Cheng. Os demais parâmetros são descritos na seção 3.1.

3.7 Considerações Finais

Neste capítulo, é feita uma revisão dos principais métodos de agrupamento de dados. Em particular, são discutidos os métodos mais conhecidos e as técnicas de bi-agrupamento. Também, são apresentadas as principais características, vantagens e desvantagens dos agrupadores.

O próximo capítulo apresenta a estratégia adotada para a extração de relações semânticas. A extração dessas relações faz uso da análise de correlação de termos para identificar agrupamento de termos que estejam conceitualmente próximos. Desta forma, a proposta apresentada faz uso de técnicas de agrupamento para a realização desta tarefa. A revisão das técnicas de agrupamento servirá para definir a função matemática que realizará o agrupamento dos termos correlatos. Verificar-se-á que o agrupador kernel possibilístico é o mais adequado para os propósitos de agrupamento de termos e servirá de base matemática para a função-objetivo do agrupador.

O capítulo a seguir ainda discutirá sobre a factibilidade da estratégia e apresentará uma simplifica-

ção ao modelo que permitirá o seu tratamento computacional. A simplificação reduziu o problema de agrupamento possibilístico a um problema de bi-agrupamento, razão pela qual essas técnicas foram revisadas.

Capítulo 4

Extração de Relações Semânticas via Análise de Correlação de Termos

Este capítulo apresenta a proposta para a extração automática de termos, conceitos e suas relações semânticas. Inicialmente é apresentado o modelo ontológico adotado, suas extensões e uma comparação com a teoria de conjuntos fuzzy que servirá de base matemática para a elaboração do algoritmo proposto. O procedimento de extração dos termos utiliza o conjunto de documentos e recursos lingüísticos de uma ontologia de senso-comum, WordNet [1], para identificar os termos que constituem o vocabulário da base de documentos. O procedimento de extração de conceitos e relações de instanciação entre termos e conceitos utiliza algoritmos de agrupamento possibilístico e de bi-agrupamento para realizar a análise de correlação de termos, ACT. É proposto, também, um algoritmo para atribuição automática de nomes aos conceitos. Por fim, são apresentadas uma discussão sobre como extrair relações semânticas de hiperonímia/hiponímia entre conceitos e uma discussão sobre a metodologia interativa para o processo de extração das relações.

4.1 Modelo Ontológico Fuzzy Relacional — FROM

O modelo ontológico adotado nesta dissertação foi inspirado no trabalho de Pereira *et al.* [62] com o modelo de recuperação de informações FROM, *Fuzzy Relational Ontological Model*.

O modelo FROM propõe dois níveis de abstração para descrever o vocabulário e a organização da informação. O primeiro nível é constituído pelas entidades mais concretas tais como termos e palavras chaves, e são representados por indivíduos na ontologia. O segundo nível representa as entidades mais abstratas e são representados por conceitos na ontologia. Os dois níveis de abstração

[1] `wordnet.princeton.edu`

estão relacionadas por meio de associações fuzzy que descrevem o grau de proximidade entre termos e conceitos.

Este modelo pode ser representado com o auxílio de um grafo bipartido onde um lado existem nós que representam os conceitos e de outro os nós que representam os termos. Unindo os dois lados existem arcos ponderados que representam as associações fuzzy. O grafo bipartido da figura 4.1 ilustra um exemplo de ontologia fuzzy relacional. No lado esquerdo, representados por elipses, estão os conceitos, os elementos mais abstratos da ontologia. Na direita estão os termos, os elementos mais concretos. Unindo essas duas entidades existem associações fuzzy, representados por arcos, indicando a pertinência dos termos aos conceitos.

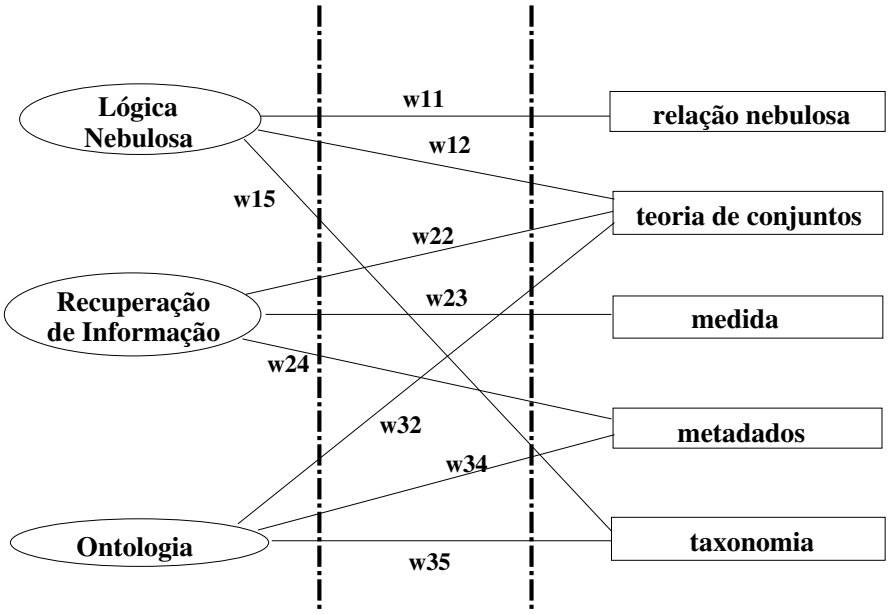


Fig. 4.1: Modelo de Ontologia

O modelo não prevê associações entre conceitos, entre termos ou qualquer outro tipo de relacionamento entre as entidades do modelo. O modelo ontológico pode ainda ser expresso na forma matricial, mostrada na figura 4.2.

A matriz R_0 contém as associações fuzzy entre os termos, representados pelas linhas da matriz, e os conceitos, representados pelas colunas da matriz.

4.2 Extensões do modelo FROM

O modelo de ontologia baseado em apenas dois níveis de abstração, apesar de simples, pode representar diversos elementos ontológicos. O modelo permite a representação de elementos simples tais

$$R_0 = \begin{array}{r} \\ \text{relação nebulosa} \\ \text{teoria de conjuntos} \\ \text{medida} \\ \text{metadados} \\ \text{taxonomia} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Lógica} & \text{Recuperação} & \\ \text{Nebulosa} & \text{de Informação} & \text{Ontologia} \end{array} \\ \begin{bmatrix} w11 & 0 & 0 \\ w12 & w22 & w32 \\ 0 & w23 & 0 \\ 0 & w24 & w34 \\ w15 & 0 & w35 \end{bmatrix} \end{array}$$

Fig. 4.2: Matriz R_0

como: termos, conceitos e suas associações fuzzy. E permite, também, a representação de elementos mais complexos como as relações taxonômicas e as definições axiomáticas tais como conceitos disjuntos, conceitos equivalentes e conceitos complementares.

Por exemplo, considere um sistema com termos K = {k_1: fuzzy relation, k_2:measure, k_3: taxonomy, k_4: set theory, k_5: metadata} e conceitos C = {c_1: Fuzzy Logic, c_2: Ontology, c_3: Information Retrieval, c_4: Artificial Intelligence}. Um possível relacionamento entre essas entidades é dado pela matriz R_0 mostrada abaixo.

$$R_0 = \begin{array}{c} \\ \textbf{k1} \\ \textbf{k2} \\ \textbf{k3} \\ \textbf{k4} \\ \textbf{k5} \end{array} \begin{array}{c} \begin{array}{cccc} \textbf{c1} & \textbf{c2} & \textbf{c3} & \textbf{c4} \end{array} \\ \begin{bmatrix} 0.9 & 0.1 & 0.0 & 0.9 \\ 0.6 & 0.1 & 0.5 & 0.7 \\ 0.1 & 0.9 & 0.3 & 0.9 \\ 0.8 & 0.4 & 0.7 & 0.9 \\ 0.1 & 0.8 & 0.7 & 0.8 \end{bmatrix} \end{array}$$

Fig. 4.3: Exemplo de Matriz R_0

A matriz R_0 representa a codificação da ontologia mostrada na figura 4.4, considerando associações fuzzy cujos valores são maiores ou iguais a 0.5.

A utilização de matrizes foi escolhida devido a facilidade de manipulação matemática e pelo fato de que o modelo que serviu de referência para este trabalho fazer uso de matrizes em sua codificação. As diferentes entidades ontológicas estão representadas na matriz de forma direta, obtidas pela estrutura da matriz, e outras são obtidas de forma indireta, utilizando mecanismos de inferência.

De maneira direta é possível a identificação de:

- lista de termos, representada pelas linhas rotuladas da matriz R_0. No exemplo da figura 4.3

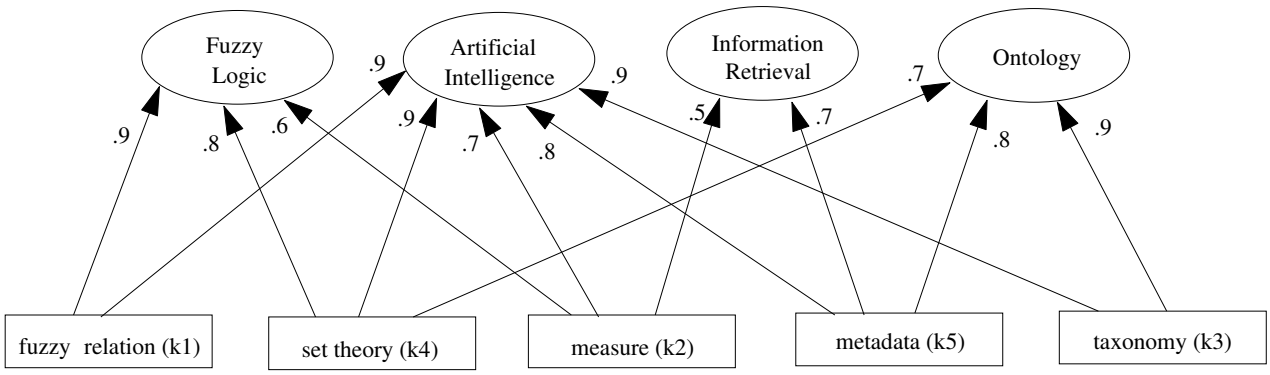


Fig. 4.4: Ontologia codificada na matriz R_0

temos os termos fuzzy relation, set theory, taxonomy, measure, metadata.

- lista de conceitos, representada pelas colunas rotuladas da matriz R_0. No exemplo da figura 4.3 temos os conceitos Fuzzy Logic, Ontology, Information Retrieval, Artificial Intelligence.
- associações fuzzy entre termos e conceitos: valor de cada célula da matriz R_0.

De maneira indireta é possível a identificação de:

- relações taxonômicas: verificando a existência de conceitos cujas associações fuzzy aos termos são sempre maiores ou iguais a de outros conceitos.
- relações axiomáticas: utilizando as definições propostas pela teoria de conjuntos fuzzy para determinar as relações entre os diferentes conceitos.

No exemplo da figura 4.3 é possível identificar, indiretamente, uma relação taxonômica entre o conceito *Artificial Intelligence* e os demais conceitos. Observe que a pertinência de todos os termos ao conceito *Artificial Intelligence* são sempre maiores ou iguais as pertinências em relação aos outros conceitos, assim pode-se inferir que o conceito em questão é mais genérico que os demais. A figura 4.5 apresenta o resultado do processo de inferência. As linhas mais grossas com setas em branco são as relações taxonômicas resultantes do processo de inferência e as linhas mais finas com setas em preto são as relações de pertinência (instanciação) de termos a conceitos.

A seção seguinte apresenta uma introdução sobre a teoria de conjuntos fuzzy e suas propriedades. Uma analogia é estabelecida entre os elementos ontológicos suportados pelo sistema e as propriedades dos conjuntos fuzzy. Essa analogia é de auxílio na identificação dos elementos ontológicos não diretamente observáveis da matriz R_0 e também auxiliará no entendimento da metodologia proposta.

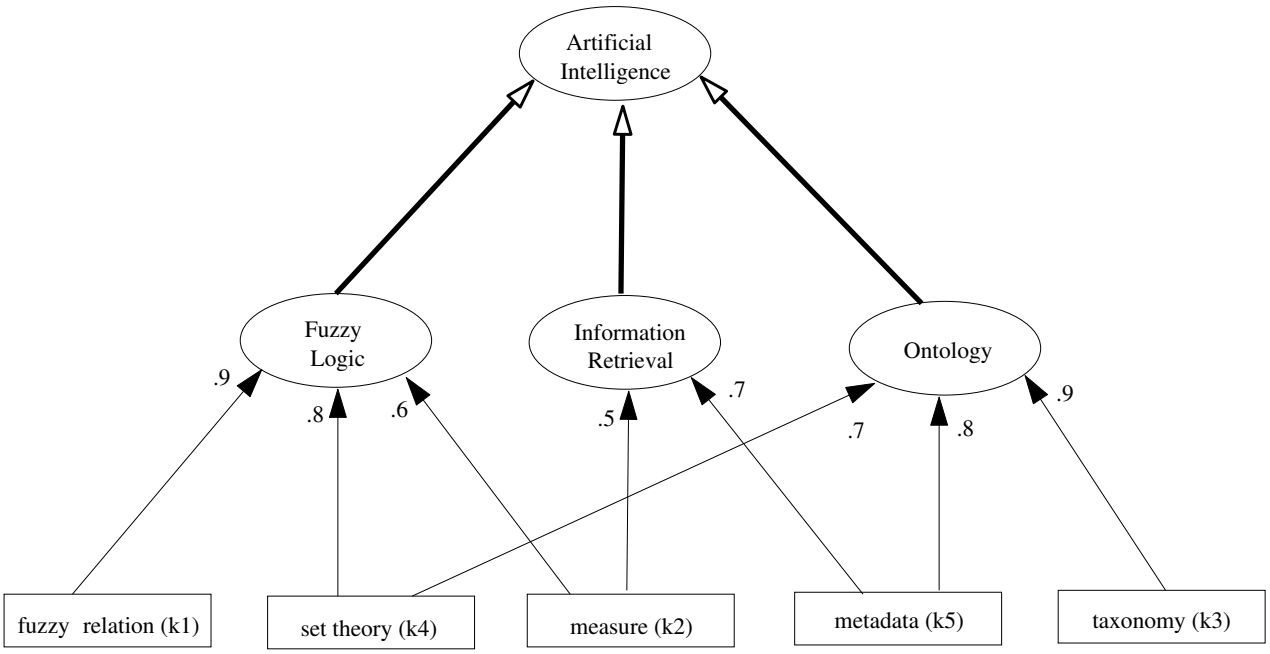


Fig. 4.5: Taxonomia inferida da matriz R_0

4.3 Conjuntos Fuzzy e os Elementos Ontológicos

Nesta seção, são apresentados a teoria de conjuntos fuzzy e a sua relação com os elementos ontológicos. A comparação entre essas duas teorias serve de suporte matemático para a proposta de extração automática de ontologias. Inicialmente, são apresentados o propósito e as principais operações com os conjuntos fuzzy. A seguir, é apresentada a equivalência entre operações em conjuntos fuzzy com os operadores e construções ontológicas.

4.3.1 Teoria de Conjuntos Fuzzy

Um conjunto fuzzy é uma classe de objetos com grau contínuo de pertinência. Tal conjunto é caracterizado por uma função de pertinência (característica) que atribui a cada objeto um grau de pertinência que varia entre 0 e 1. A noção de inclusão, união, interseção, complemento, e outros são estendidas para tais conjuntos [61]. O desenvolvimento da teoria de conjuntos fuzzy surgiu da necessidade de se tratar questões como imprecisão, ambigüidade e subjetividade. Por exemplo, a "classe de todos os números reais que são muito maiores que 1" ou ainda a "classe dos homens altos" não constituem classes ou conjuntos no sentido usual, pois não existe consenso em relação a quais são os elementos que pertencem ou não a esses conjuntos. O melhor é indicar o nível de conformidade de um certo objeto a tais conjuntos.

Definições

Seja X um conjunto de pontos (Objetos) com o elemento genérico de X denotado por x. Assim, $X = \{x\}$. Um conjunto fuzzy A em X é caracterizado pela função pertinência $f_A(x)$ que associa a cada ponto em X um número real no intervalo $[0, 1]$ com valor de $f_A(x)$ em x representando o grau de pertinência de x em A.

Axiomas e Construções

Considerando que um conjunto fuzzy pode ser representado em termos matemáticos com pares ordenados

$$A = [x, f_A(x)], x \in X$$

onde $f_A(x)$ é a função de pertinência, então pode-se definir as seguintes construções e axiomas:

Conjunto Vazio: Um conjunto fuzzy é vazio se e somente se a função de pertinência é identicamente zero em X.

$$f_A(x) = 0, x \in X$$

Conjunto Universo: Um conjunto fuzzy é universo se e somente se a função de pertinência é identicamente um em X.

$$f_A(x) = 1, x \in X$$

Identidade: Dois conjuntos fuzzy A e B são iguais se e somente se:

$$f_A(x) = f_B(x), x \in X$$

Complemento: O complemento de um conjunto fuzzy A é denotado por $\bar{A}$ e definido por:

$$f_{\bar{A}}(x) = 1 - f_A(x), x \in X$$

Continência: O conjunto fuzzy A está contido ou é subconjunto do conjunto fuzzy B se e somente se:

$$f_A(x) \leq f_B(x), x \in X$$

União: A união de dois conjuntos fuzzy A e B com funções de pertinência $f_A(x)$ e $f_B(x)$ é um conjunto fuzzy C cuja função de pertinência é dada por:

$$f_C(x) = f_A(x)\ s\ f_B(x), x \in X$$

onde s é um operador de co-norma triangular.

Interseção: A interseção de dois conjuntos fuzzy A e B com funções de pertinência $f_A(x)$ e $f_B(x)$ é um conjunto fuzzy C com função de pertinência dado por:

$$f_C(x) = f_A(x)\ t\ f_B(x), x \in X$$

onde t é um operador de norma triangular.

As normas e co-normas triangulares formam uma classe geral de operadores de união e interseção, com características de comutatividade, associatividade e monotonicidade, atendendo ainda as condições de contorno. Diferente da união e interseção, que trabalham com conjuntos definidos num mesmo universo, as operações baseadas em normas e co-normas triangulares podem operar conjuntos em universos distintos. Sejam A e B dois conjuntos nebulosos definidos nos universos X e Y, respectivamente, e a e b valores de pertinência dados por $a = f_A(x)$ e $b = f_B(x)$. Então, as normas e co-normas triangulares (norma-t e norma-s) podem ser definidas como:

- Norma-t: operador de dois argumentos $t : [0;1]^2 \rightarrow [0;1]$ que satisfaz as seguintes condições:
  - Comutatividade: $a\ t\ b = b\ t\ a$;
  - Associatividade: $a\ t\ (b\ t\ c) = (a\ t\ b)\ t\ c$;
  - Monotonicidade: se $a \leq b$ e $c \leq d$, então, $a\ t\ c \leq b\ t\ d$;
  - Condições de Contorno: $0\ t\ a = 0, \quad 1\ t\ a = a$.

  Exemplos de norma-t:

  - $a\ t\ b = ab$
  - $a\ t\ b = \min(a, b)$

- Norma-s: operador de dois argumentos $t : [0;1]^2 \rightarrow [0;1]$ que satisfaz as seguintes condições:
  - Comutatividade: $a\ s\ b = b\ s\ a$;
  - Associatividade: $a\ s\ (b\ s\ c) = (a\ s\ b)\ s\ c$;
  - Monotonicidade: se $a \leq b$ e $c \leq d$, então, $a\ s\ c \leq b\ s\ d$;
  - Condições de Contorno: $0\ s\ a = a, \quad 1\ t\ a = 1$.

  Exemplos de norma-s:

  - $a\ t\ b = a - ab + b$
  - $a\ t\ b = \max(a, b)$

Como pode ser observado, a teoria de conjuntos fuzzy é muito próxima dos modelos ontológicos no sentido que servem para descrever grupos de objetos que compartilham alguma semelhança. Observa-se ainda que as propriedades e construções existentes nos conjuntos fuzzy também estão definidos nos modelos de ontologia. Assim, uma análise comparativa entre os dois modelos servirá para dar maior fundamentação matemática aos elementos ontológicos e auxiliará na compreensão da teoria envolvida para a extração de conceitos e suas relações.

4.3.2 Relação entre Conjuntos Fuzzy e Ontologias

As equivalências entre as definições da teoria fuzzy e as construções ontológicas são apresentadas nesta seção. OWL é uma linguagem padronizada pelo W3C para especificação de ontologias que permite a criação de conceitos, indivíduos, relacionamentos e uma série de construções axiomáticas. Embora esta linguagem não permita a especificação de ontologias fuzzy, suas construções servem de apoio para o levantamento de diversos mapeamentos. A ontologia fuzzy utilizada nesta dissertação é definida como um extensão da ontologia convencional que incorpora a noção de níveis de pertinência dos indivíduos aos conceitos, assim como definida em Quan et al [1].

Na seqüência, são apresentadas as construções ontológicas, a representação dessas construções em OWL, a simbologia gráfica e a equivalência com a teoria de conjuntos fuzzy.

Conceito Coisa (owl:Thing): A definição do conceito que contém todos os indivíduos é representado pelo conjunto fuzzy universo e apresentado na figura 4.6. $f_A(x) = 1, x \in X$.

Fig. 4.6: Conceito Coisa

Conceito Nada (owl:Nothing): A definição do conceito que não possui indivíduos é representado pelo conjunto fuzzy vazio e apresentado na figura 4.7. $f_A(x) = 0, x \in X$.

Fig. 4.7: Conceito Nada

Equivalência entre conceitos (owl:Equivalent): Conceitos serão equivalentes se e somente se as representação fuzzy destes conceitos possuírem funções de pertinência idênticas. A representação gráfica para a equivalência é apresentada na figura 4.8. $f_A(x) = f_B(x), x \in X$.

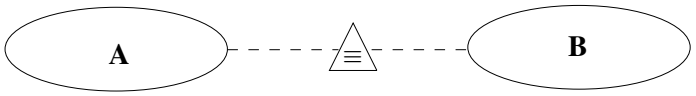


Fig. 4.8: Equivalência entre conceitos

Relação Taxonômica (owl:subClassOf): A relação taxonômica, mostrada na figura 4.9, entre os conceitos pode ser modelada com o auxílio da propriedade de continência da teoria de conjuntos fuzzy, ou seja, um conceito (conjunto) A é subclasse (subconjunto) de outro conceito B se a função de pertinência $f_A(x)$ for sempre menor ou igual à função de pertinência $f_B(x)$. $f_A(x) \leq f_B(x), x \in X$.

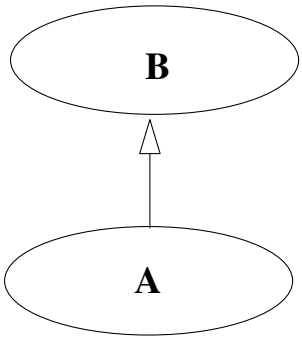


Fig. 4.9: Relações Taxonômicas

Conceito Disjuntos (owl:DisjointWith): Conceitos são disjuntos se não possuírem instâncias em comum. Em teoria fuzzy dois conjuntos são disjuntos se a interseção entre estes resultar em um outro conjunto fuzzy cuja função de pertinência é nula. A representação gráfica para conceitos disjuntos é apresentada na figura 4.10. $f_A(x) t f_B(x) = 0, x \in X$.

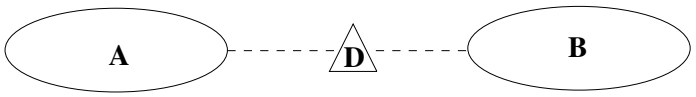


Fig. 4.10: Conceitos Disjuntos

Conceito Complementares (owl:complementOf): Dois conceitos são complementares se a sua representação fuzzy for complementar, ou seja, as funções de pertinência estão em complemento de 1.

A figura 4.11 mostra a representação gráfica para conceitos complementares. $f_{\bar{A}}(x) = 1 - f_A(x), x \in X$.

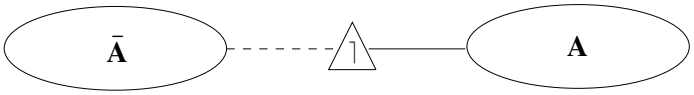


Fig. 4.11: Complemento de um conceito

União de conceitos (owl:unionOf): Um determinado conceito C será a união de outros dois conceitos A e B, se as funções de pertinência associada aos conjunto fuzzy A, B e C estabelecerem a seguinte relação $f_C(x) = f_A(x)\ s\ f_B(x)$. A representação gráfica para a união é apresentada na figura 4.12.

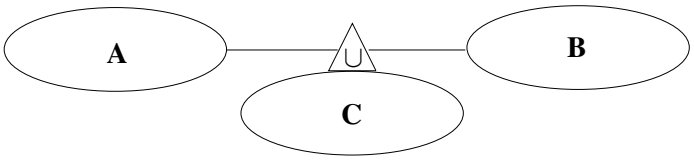


Fig. 4.12: União de Conceitos

Interseção de conceitos (owl:intersectionOf): Um determinado conceito C será a interseção de outros dois conceitos A e B, se as funções de pertinência associadas aos conjuntos fuzzy A, B e C estabelecerem a seguinte relação $f_C(x) = f_A(x)\ t\ f_B(x)$. A representação gráfica para a interseção é apresentada na figura 4.13.

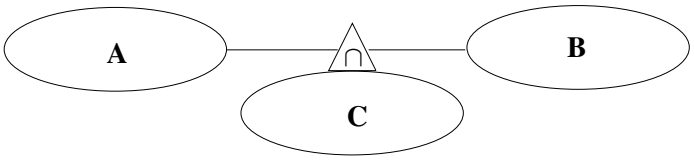


Fig. 4.13: Interseção de Conceitos

A lista de mapeamentos apresentada anteriormente ilustra as diversas possibilidades do modelo ontológico; contudo, nem todos serão utilizados. O mapeamento sobre conceitos disjuntos será importante para a identificação iterativa de conceitos. Os mapeamentos sobre o conceito "Coisa", o conceito "Nada", e relações taxonômicas serão de auxílio para entender como seria possível a identificação de taxonomia entre conceitos, veja seção 4.4.7.

4.4 Construção da árvore ontológica

A construção da árvore ontológica é baseada no processamento léxico dos documentos e de ferramentas de análise estatística para a identificação de conceitos.

O diagrama da figura 4.14 mostra os estágios necessários para a obtenção do árvore ontológica. O estágio de *Pré-Processamento* faz uso de técnicas lingüísticas para a extração da lista de termos. O conteúdo de cada documento é processado por um analisador léxico que realiza a tokenização dos textos. Os tokens são categorizados e filtrados resultando em uma lista reduzida de tokens que constituirão os termos da ontologia. O processo de categorização faz uso de uma base lexical externa. Neste caso a base lexical escolhida foi o WordNet.

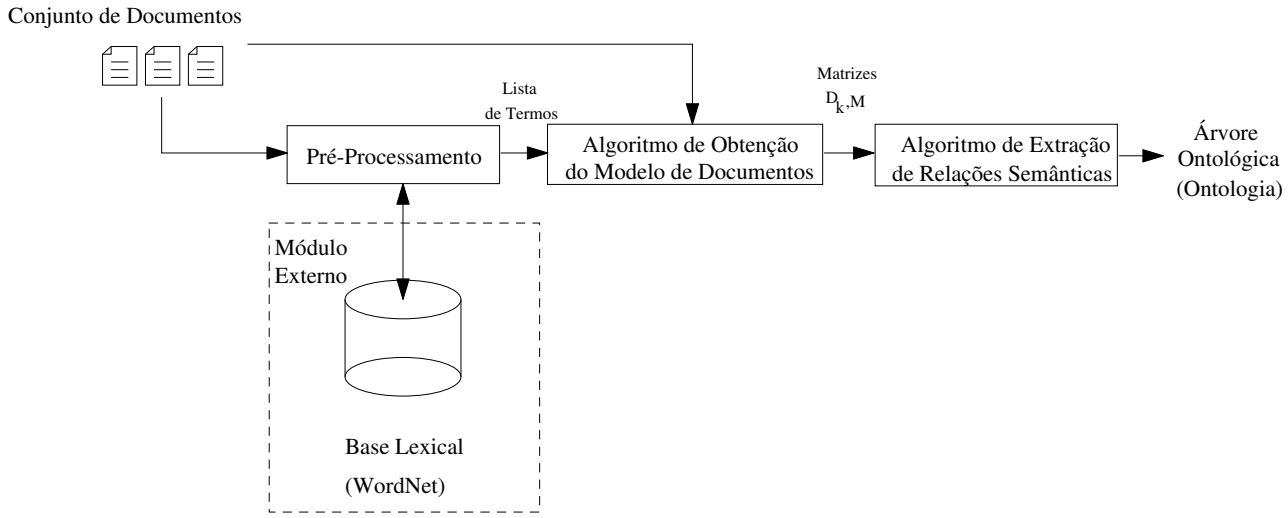


Fig. 4.14: Fluxo de processamento dos dados

O segundo estágio é constituído de um algoritmo responsável por modelar matematicamente os documentos da base. Esse algoritmo toma como entrada a lista de termos obtida no estágio anterior e o conteúdo dos documentos para determinar a matriz de ocorrência de termos em documentos. Os resultados deste estágio são duas matrizes: a matriz de ocorrência de termos em documentos D_k e a matriz que fornece a correlação entre os termos M.

O último estágio é responsável por identificar os conceitos e as relações de instanciação entre termos e conceitos. Para isto, técnicas estatísticas são utilizadas para a identificação de padrões de utilização dos termos permitindo caracterizar os conceitos discutidos pela base de documentos. Neste estágio é utilizado um algoritmo de agrupamento que recebe como entrada as matrizes do estágio anterior e produz como resultado a lista de conceitos e relações.

O modelo ontológico discutido na seção 4.1 permite identificar quais devem ser os requisitos do algoritmo de criação automática de ontologias. Definir a árvore ontológica é definir os parâmetros

que caracterizam a matriz R_0 tais como: número de linhas, número de colunas e valor de cada célula da matriz. Portanto, a definição de uma ontologia no sistema relacional fuzzy requer a resposta para as seguintes questões:

- Qual a quantidade de termos?
- Quais são estes termos?
- Qual a quantidade de conceitos da ontologia?
- Quais são os nomes destes conceitos?
- Quais são as relações taxonômicas?
- Quais são os valores das associações fuzzy entre termos e conceitos?

4.4.1 Identificação de Termos

A identificação dos termos (indivíduos) da ontologia é realizada utilizando-se o próprio conteúdo dos documentos e uma ontologia de senso comum, WordNet. Os documentos são processados por um analisador léxico que irá extrair os tokens dos documentos. Esses tokens são processados e categorizados de acordo com a categoria gramatical e posteriormente uma série de filtragens permitirá obter os termos da ontologia. É importante ressaltar que nesta dissertação foi utilizada a WordNet para a língua inglesa. Assim, a metodolologia adotada apresenta uma dependência com essa língua. Entretanto, esta limitação pode ser resolvida apenas modificando-se a versão da WordNet. Por exemplo, poderia ter sido utilizada a versão portuguesa do WordNet [2]. A escolha da língua inglesa foi motivada pelo fato de que a WordNet é mais completa nessa língua.

Dentre as possibilidades do WordNet temos:

- Determinação da classe gramatical do termo, o qual pode ser: SUBSTANTIVO, ADJETIVO, VERBO, ADVERBIO.
- Determinação do lema de um dado termo.
- Relacionamento entre termos dentre os quais podemos citar: Meronímia, Holonímia, Hiperonímia, Hiponímia, Sinônimos.

Adicionalmente, uma lista de termos não-relevantes, denominados *stop-words*, é utilizada para marcar os termos que deverão ser removidos durante o processo de filtragem. Ao final do processamento é obtido uma lista com a descrição de cada termo. O fragmento de texto abaixo servirá de exemplo para mostrar a extração dos termos:

[2] `http://www.nilc.icmc.usp.br/~arianidf/WordNet-BR.html`

"The aim of this paper is to evaluate the performance of genetic algorithm for the flowshop scheduling problem with an objective of minimizing the makespan."

O processamento deste fragmento resulta na descrição apresentada na tabela 4.1.

Termo	*Lema*	*Stop-Word*	*Substantivo*	*Adjetivo*	*Verbo*	*Adverbio*
the		•				
aim	aim		•		•	
of		•				
this		•	•			
paper	paper				•	
is	be	•			•	
to		•				
evaluate	evaluate				•	
performance	performance		•			
genetic	genetic			•		
algorithms	algorithm		•			
for		•				
flowshop						
scheduling	scheduling		•		•	
problem	problem		•			
with		•				
an	an	•				
objective	objective		•	•		
minimizing	minimize				•	
makespan						

Tab. 4.1: Descrição dos termos.

As células com um marcador indicam que o termo pertence àquela categoria e os termos cuja coluna lema não foram preenchidas representam palavras não encontradas no dicionário eletrônico WordNet. Neste trabalho, as palavras não encontrada no WordNet são ignoradas e não farão parte do vocabulário da base de documentos. Contudo, é necesário ressaltar que esse procedimento pode resultar na eliminação de palavras importantes para a descrição dos documentos. Palavras específicas de domínio possivelmente serão removidas, dado que o WordNet contém somente termos mais genéricos e de uso comum na língua escrita. Além disso, o WordNet pode conter informações imprecisas. Por exemplo, a tabela 4.1 mostra que o termo *paper* é somente um verbo; no entanto, este é, também, um substantivo.

Terminado o processo de categorização, a lista de termos será filtrada para a remoção de termos

não relevantes e outras fontes de ruído. A remoção dos termos seguirá as seguintes regras:

- Remoção dos termos que se enquadram na categoria de stop-words;
- Remoção dos termos que não se enquadram exclusivamente nas categorias: substantivo, adjetivo ou substantivo/adjetivo. Entretanto é importante ressaltar que a remoção de termos cuja classe gramatical seja o verbo pode ser ruim em certos domínios, por exemplo, para o domínio futebol que são regidos por verbos;
- Remoção dos termos que não estão presentes no dicionário WordNet.

A aplicação das regras acima para a filtragem e do procedimento de lematização resultam na seguinte lista de termos:

{ performance, genetic, algorithm, problem, objective }

Abaixo está o trecho completo de um abstract retirado de um artigo sobre algoritmos genéticos. A lista de termos selecionados está em destaque.

*"The aim of this paper is to evaluate the **performance** of **genetic algorithms** for the flowshop scheduling **problem** with an **objective** of minimizing the makespan. First we examine various **genetic operators** for the scheduling **problem**. Next we compare **genetic algorithms** with other search **algorithms** such as **local** search, taboo search and simulated annealing. By **computer simulations**, it is shown that **genetic algorithms** are a bit **inferior** to the others. Finally, we show two **hybrid genetic algorithms**: **genetic local search** and **genetic simulated** annealing. Their high **performance** is demonstrated by **computer simulations**."*

Termos selecionados após o processo de lematização:

{ performance, genetic, algorithm, problem, objective, operator, local, computer, simulation, inferior, hybrid }

4.4.2 Identificação de Relações Semânticas

A completa identificação de conceitos no sistema relacional fuzzy deve levar em consideração três fatores: determinação do número de conceitos; valor das associações fuzzy entre conceitos e termos; e sugerir nomes significativos para os conceitos obtidos.

Caracterizando o conceito

Conceitos, na definição ontológica, são utilizados para descrever um conjunto de elementos que compartilham alguma característica em comum. Por exemplo, seja o conceito Mamífero que descreve

o conjunto de animais que apresentam glândulas mamárias nas fêmeas.

conceito Mamífero ≡ { homem, baleia, morcego, gato, cachorro, macaco }

Os termos homem, baleia, morcego, gato, cachorro e macaco são utilizados para caracterizar o conceito mamífero pois todos compartilham uma ou mais características; neste caso, a característica compartilhada é que todos possuem mamas. No entanto, conceitos podem ser utilizados para descrever idéias mais abstratas ou que possuem um nível diferente de associação com os termos integrantes do conceito. Considere o conceito "Palavras chaves que descrevem o assunto de pesquisa Algoritmos Genéticos" ou, de forma simplificada, "Algoritmos Genéticos". Nesse caso, os membros deste conceito podem compartilhar nenhuma semelhança fisiológica ou de idéia quando analisadas isoladamente. Entretanto, se considerarmos o aspecto lingüístico de utilização das palavras pode-se dizer que existe uma propriedade comum entre os termos: a de que essas palavras são comuns na descrição de um determinado assunto.

conceito Algoritmos Genéticos ≡ { genetic, algorithm, operator, population, selection }

O raciocínio acima nos leva à conclusão de que é possível extrair conceitos através da identificação de termos comuns em assuntos. Devido às características da linguagem escrita, termos comuns a um determinado assunto costumam ocorrer de forma correlata nos documentos [59]. Assim, a utilização de um modelo apropriado para descrever os documentos e uma métrica para medir a correlação entre termos servirão de base para o algoritmo extrair os conceitos e suas relações.

Métrica para a correlação entre termos

Antes de escolhermos uma métrica para a medida de correlação entre termos é necessário uma definição matemática para o termo. A representação escolhida para os termos é a vetorial, na qual cada termo é um vetor obtido com as linhas que compõem a matriz D_k. Assim, o i-ésimo termo será dado por:

$$\overrightarrow{t_i} = (dk_{i,1}, dk_{i,2}, \ldots, dk_{i,k}, \ldots, dk_{i,N}) \tag{4.1}$$

A representação matemática vetorial para o termo permite a escolha de funções para o cálculo da correlação entre os termos. A função corr $\left(\overrightarrow{t_1}, \overrightarrow{t_2}\right)$, que calcula a correlação entre os termos $\overrightarrow{t_1}$ e $\overrightarrow{t_2}$, deve fornecer como resultado um valor entre 0 e 1, sendo 1 para correlação total e 0 para nenhuma correlação.

A escolha trivial para esta função é a separação angular entre os vetores. Assim

$$\text{corr}\left(\overrightarrow{t_1}, \overrightarrow{t_2}\right) = \cos\left(\overrightarrow{t_1}\overrightarrow{t_2}\right) = \frac{\overrightarrow{t_1} \cdot \overrightarrow{t_2}}{||\overrightarrow{t_1}|| \cdot ||\overrightarrow{t_2}||} \tag{4.2}$$

Outra possibilidade é o cálculo do vetor diferença entre $\overrightarrow{t_1}$ e $\overrightarrow{t_2}$:

$$\text{corr}\left(\overrightarrow{t_1}, \overrightarrow{t_2}\right) = \exp\left(-\frac{||\overrightarrow{t_1} - \overrightarrow{t_2}||}{\sigma}\right) \tag{4.3}$$

De tal sorte que se $\overrightarrow{t_1}$ e $\overrightarrow{t_2}$ forem idênticos a correlação é total e se forem muito diferentes a função será um valor próximo de zero.

Nesta dissertação será utilizada a primeira definição devido sua simplicidade matemática, uso na literatura e vetores termos já normalizados, reduzindo o cálculo da correlação à $\overrightarrow{t_1} \cdot \overrightarrow{t_2}$.

Métrica para a distância entre termos

Definição: Dado um conjunto $M \neq \phi$ e $d : M \times M \rightarrow R$, sendo R o conjunto dos números reais, indica-se por $d(x, y)$ a imagem de um par genérico (x, y) por meio da função d. Diz-se que d é uma métrica sobre M se as seguintes condições são verificadas para quaisquer x, y, $z \in M$:

- (M1) $d(x, y) = 0 \;\; para \;\; (x = y)$;
- (M2) $d(x, y) = d(y, x)$;
- (M3) $d(x, y) \leq d(x, z) + d(z, y)$.

Nessas condições, cada imagem $d(x, y)$ recebe o nome de distância de x a y e o par (M, d), onde d é uma métrica sobre M, de espaço métrico. Cada objeto de um espaço métrico será sempre referido como ponto desse espaço, seja ele um ponto em si mesmo, ou um número, ou ainda uma função ou um vetor. A propriedade (M3) é conhecida como desigualdade triangular.

Entretanto, a métrica utilizada para avaliar a distância entre dois termos não respeitará, necessariamente, a condição de desigualdade triangular, pelo seguinte motivo: em termos lingüísticos existem situações onde um determinado termo A é muito próximo dos termos B e C; contudo, isto não implica que os termos B e C estejam próximos. A situação apresentada anteriormente é freqüente quando lidamos com palavras polissêmicas, ou seja, palavras que podem ter mais de um significado. Por exemplo, considere o termo manga, o qual pode se referir à noção de fruta ou à noção de parte de uma vestimenta. O termo manga certamente estará próximo do termo fruta (quando manga refere-se a idéia de fruta) e pode estar próximo de camiseta (quanto manga refere-se a idéia de vestimenta), contudo fruta e camiseta são dois termos que não estão próximo.

Assim, a medida de distância entre os termos deve atender aos critérios M1 e M2 e ainda deve ser função da correlação entre os termos:

$$d\left(\overrightarrow{t_1}, \overrightarrow{t_2}\right) = g\left(\text{corr}\left(\overrightarrow{t_1}, \overrightarrow{t_2}\right)\right) = 1 - \text{corr}\left(\overrightarrow{t_1}, \overrightarrow{t_2}\right) \tag{4.4}$$

A função g é monotonicamente decrescente que recebe como argumento um número real $x \in [0, 1]$, indicando a correlação entre os termos, e mapeia este valor para o intervalo $[0, D]$, onde D é uma constante positiva.

Como os termos são representados por vetores de dimensões elevadas e a métrica de distância não respeita a desigualdade triangular, torna-se impossível utilizar planos cartesianos para mostrar corretamente a distância entre termos. Dessa forma, foi utilizada a representação em grafos para mostrar a distância entre eles. A figura 4.15 apresenta um exemplo de grafo de termos. Os nós deste grafo representam os termos e as arestas representam as correlações não nulas entre os termos.

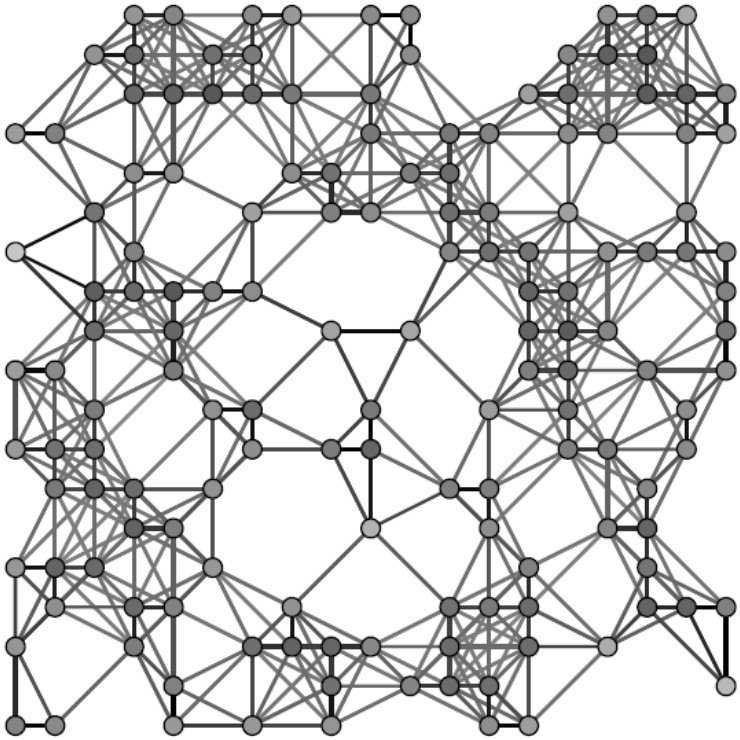


Fig. 4.15: Grafos de Termos

4.4.3 Extração de Relações Semânticas

Como foi apresentado anteriormente, a extração das relações semânticas pode ser realizada identificando-se termo correlatos, ou seja, termos que estejam próximos semanticamente entre si. A

identificação de objetos que estejam próximos é alvo de estudo das técnicas de aprendizado não-supervisionado, mais especificamente, das técnicas de agrupamento de dados.

A lista de diferentes tipos de agrupadores é bastante ampla, desde os mais simples, tal como o K-Means, até os mais complexos que envolvem funções de mapeamento (Kernel), como visto no Capítulo 3. Em nosso caso, o algoritmo de agrupamento deve possuir as seguintes características:

- Identificação automática do número de grupos;
- A pertinência dos termos aos grupos é *fuzzy*, pois o modelo ontológico de referência, FROM, utiliza associações fuzzy entre termos e conceitos. A utilização de lógica fuzzy ainda garante maior imunidade a ruídos e incertezas presentes nos dados;
- Pode haver grande sobreposição entre os grupos, ou seja, termos que pertençam fortemente a dois ou mais grupos, comum com termos que possuem mais de um significado (polissemia);
- A geometria dos grupos não é, necessariamente, esférica.

A análise dos requisitos apresentados anteriormente mostram que, conceitualmente, o agrupador mais adequado para realizar a tarefa de extração de conceitos é o agrupador Kernel Possibilístico C-Means, KPCM. A escolha desse agrupador é devido a duas características importantes: a capacidade de detectar grupos com grandes sobreposições e a atribuição de graus de pertinência dos objetos aos grupos.

A função objetivo para o agrupador Kernel Possibilistic C-Means é dada por:

$$J\left(U,V\right)=\sum_{i=1}^{C}\sum_{j=1}^{N}\left(u_{ij}\right)^{m}\left|\left|\phi\left(x_{k}\right)-\phi\left(v_{i}\right)\right|\right|+\sum_{i=1}^{C}\eta_{i}\sum_{j=1}^{N}\left(1-u_{ij}\right)^{m} \tag{4.5}$$

Sujeita às seguintes restrições

$$u_{ij}\in[0,1],\ 0<\sum_{j=1}^{N}u_{ij}<N,\ 0<\sum_{i=1}^{C}u_{ij}<C$$

Para resolver o problema de minimização dessa função objetivo, Krishnapuram e Keller [44] propuseram um tratamento diferente para o problema de agrupamento. Trataram cada grupo de maneira desacoplada e, desse modo, o problema de minimização da função 4.5 tornou-se um problema de minimização de C equações objetivos conforme mostrado na equação 4.6.

$$J_{i}\left(U,V\right)=\sum_{j=1}^{N}\left(u_{ij}\right)^{m}\left|\left|\Phi\left(x_{k}\right)-\Phi\left(v_{i}\right)\right|\right|^{2}+\eta_{i}\sum_{j=1}^{N}\left(1-u_{ij}\right)^{m} \tag{4.6}$$

Embora o agrupador Kernel Possibilistic C-Means seja o mais adequado para os propósitos do agrupador de termos, ele exige a definição de um objeto denominado protótipo de grupo. Este protótipo nada mais é do que o centro do grupo, porém o conceito de centro não existe no modelo proposto, a única informação disponível é a distância entre os termos. Dessa forma faz-se necessário definir uma nova função objetivo contemplando somente as distâncias entre os termos e as respectivas pertinências dos termos ao grupo. Ainda que o agrupamento possibilístico não possa ser utilizado, os componentes da função objetivo serão de auxílio na definição de uma nova função.

Considerando novamente a função objetivo do agrupador Kernel Possibilistic C-Means observa-se que esta é composta por três funções. A primeira função é responsável por avaliar a proximidade dos termos e será denominada de função de **coesão** φ do grupo. A segunda parcela é responsável por avaliar o tamanho do grupo e será denominada de função de **cobertura** ρ. A nova função objetivo de agrupamento também apresenta essas parcelas.

$$J = \varphi + \eta \cdot \rho$$

O fator de balanço η e a função ρ que regula o tamanho do grupo serão mantidos idêntico ao do modelo possibilístico, dado que não possuem dependência com o centro do grupo. A definição da função φ exige uma análise mais detalhada de seus elementos constituintes.

A função de coesão é composta por duas estruturas matemáticas acopladas: uma avalia a distância entre um objeto individual e o centro do grupo $(u_{ij})^m \, ||\Phi(x_k) - \Phi(v_i)||^2$, denominada de função f_i, e a outra avalia a coesão como um todo do grupo, $\sum_{j=1}^{N}(\cdot)$, denominada de função φ.

Em termos gerais pode-se dizer que f_i é a estrutura que avalia individualmente a distância do objeto k ao centro i e φ é a estrutura que avalia a coesão do grupo como um todo.

Os parâmetros de entrada para a função φ são as funções f_i's. Assim

$$\varphi = \varphi\left(\bigcup_{i \in N} f_i(\mu_i, v)\right)$$

Como não existe o conceito de centro no modelo ACT, a função que mede a distância dos objetos ao centro será modificada de modo a medir a distância entre objetos e assim a função φ torna-se

$$\varphi = \varphi\left(\bigcup_{i,j \in T} f_{d_{i,j}}(\mu_i, \mu_j)\right)$$

As funções f e φ do novo agrupador e o agrupador KPCM compartilham de algumas propriedades que serão utilizadas na especificação da função objetivo.

Propriedades da função f

- A função deve ser sempre maior ou igual a zero.
  - $f_{d_{i,j}}(\mu_i, \mu_j) \geq 0 \; \forall i, j, 1 \leq i \leq M \;\; e \;\; 1 \leq j \leq M$
- A função deve ser monotonicamente crescente em relação aos parâmetros μ_i, μ_j e distâncias.
  - $f_{d_{i,j}}(\mu_i^*, \mu_j) \geq f_{d_{i,j}}(\mu_i, \mu_j)$ para $\mu_i^* > \mu_i$
  - $f_{d_{i,j}}(\mu_i, \mu_j^*) \geq f_{d_{i,j}}(\mu_i, \mu_j)$ para $\mu_j^* > \mu_j$
  - $f_{d_{i,j}}(\mu_i, \mu_j) \geq f_{c_{i,j}}(\mu_i, \mu_j)$ para $d_{i,j} > c_{i,j}$
- A função deve ser simétrica em seus argumentos.
  - $f_{d_{i,j}}(\mu_i, \mu_j) = f_{d_{i,j}}(\mu_j, \mu_i)$

Exemplos de função f que atendem os requisitos listados acima:

- $f_{d_{i,j}}(\mu_i, \mu_j) = (\mu_i \cdot \mu_j)^m \cdot d_{i,j}^2$
- $f_{d_{i,j}}(\mu_i, \mu_j) = \left(\frac{\mu_i + \mu_j}{2}\right)^m \cdot d_{i,j}^2$
- $f_{d_{i,j}}(\mu_i, \mu_j) = (\mu_i - \mu_i \cdot \mu_j + \mu_j)^m \cdot d_{i,j}^2$

Propriedades da função φ

- A função deve ser sempre maior ou igual a zero.
  - $\varphi\left(\bigcup_{i,j \in T} f_{d_{i,j}}(\mu_i, \mu_j)\right) \geq 0$
- A função deve ser monotonicamente crescente para toda função f de entrada
  - $\varphi\left(\bigcup_{i,j \in T} f_{d_{i,j}}(\mu_i, \mu_j)\right) \geq \varphi\left(\bigcup_{i,j \in T} f^*_{d_{i,j}}(\mu_i, \mu_j)\right)$ para todo $f > f^*$
- A função deve ser simétrica em seus argumentos
  - $\varphi\left(\ldots, f_{d_{m,1}}(\mu_m, \mu_1), f_{d_{m,2}}(\mu_m, \mu_2), \ldots,\right) = \varphi\left(\ldots, f_{d_{m,2}}(\mu_m, \mu_2), f_{d_{m,1}}(\mu_m, \mu_1), \ldots\right)$

Exemplos de função φ que atendem os requisitos listados acima:

- $\varphi\left(\bigcup_{i,j \in T} f_{d_{i,j}}(\mu_i, \mu_j)\right) = \sum_{i \in T} \sum_{j \in T} f_{d_{i,j}}(\mu_i, \mu_j)$
- $\varphi\left(\bigcup_{i,j \in T} f_{d_{i,j}}(\mu_i, \mu_j)\right) = \prod_{i \in T} \prod_{j \in T} f_{d_{i,j}}(\mu_i, \mu_j)$

Uma outra forma de interpretar o papel das funções f e φ é observar que o agrupador identifica grupos de termos próximos em estrutura do tipo grafo. Os nós do grafo representam os objetos alvo do agrupamento e os arcos são rotulados com os valores das distâncias entre os objetos. A função f é responsável por avaliar individualmente cada aresta do grafo e a função φ avalia o conjunto de arestas. A figura 4.16 mostra o papel de cada função.

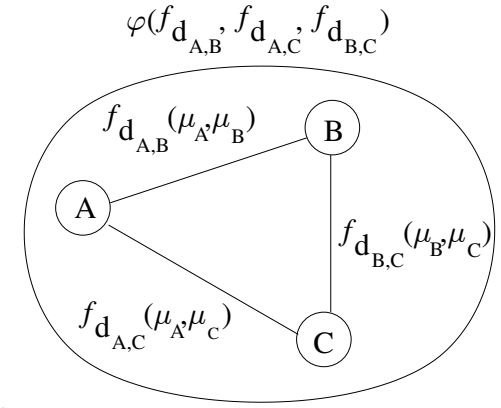


Fig. 4.16: Papel das funções f e φ

A análise conduzida acima permite estabelecer um formato geral para a função objetivo do agrupador, mostrado na equação 4.7.

$$J\left(\overrightarrow{\mu}\right) = \varphi\left(\bigcup_{i,j\in T} f_{d_{i,j}}\left(\mu_i, \mu_j\right)\right) + \eta \sum_{j=1}^{T} \left(1-\mu_j\right)^m \tag{4.7}$$

A identificação de termos correlatos é obtida minimizando-se a função objetivo J. Esta função pode apresentar um ou mais pontos de mínimo que representam as diferentes configurações para agrupamentos de termos e, conseqüentemente, caracterizam alguma relação semântica trazida pelos documentos da base. A identificação dos mínimos de J serve de referência para determinar o número de relações na ontologia.

Identificar todos os pontos de mínimo não é factível e não trará benefício para a construção da ontologia. A melhor alternativa é estabelecer uma métrica que permita avaliar a importância das relações e escolher somente os mais relevantes. Por exemplo, considere o seguinte gráfico da figura 4.17 que ilustra o comportamento de uma função J hipotética à medida que o vetor de parâmetros $\overrightarrow{\mu}$ varia do conceito nada $\overrightarrow{0}$ até o conceito coisa $\overrightarrow{1}$.

A função objetivo J apresenta seis pontos de mínimo, $\overrightarrow{\mu_1}$, $\overrightarrow{\mu_2}$, $\overrightarrow{\mu_3}$, $\overrightarrow{\mu_4}$, $\overrightarrow{\mu_5}$, $\overrightarrow{\mu_6}$, mas, evidentemente, apenas três são relevantes para descrever a ontologia: $\overrightarrow{\mu_1}$, $\overrightarrow{\mu_4}$, $\overrightarrow{\mu_6}$, pois são os pontos de mínimo mais pronunciados da função. A escolha do número de grupos poderia ser baseada nos seguintes critérios:

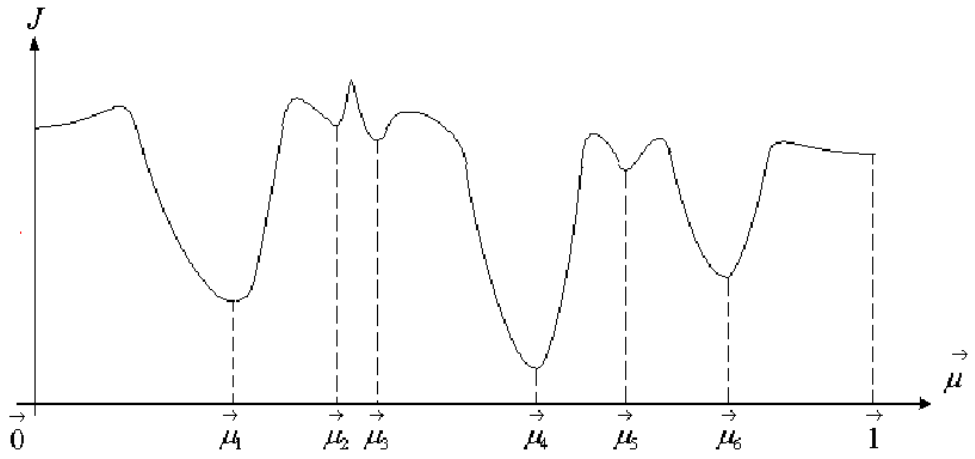


Fig. 4.17: Função-Objetivo J

valor limiar indicando o valor máximo aceito para a função objetivo ou avaliação da sobreposição dos grupos formados.

Determinação dos mínimos da função J

A determinação dos pontos de mínimo da função J envolve a resolução de um sistema de equações a derivadas parciais que de um modo geral não é simples ou possível de se resolver analiticamente. Uma possibilidade é a utilização de técnicas de otimização multi-modal. Essas técnicas são baseadas em algoritmos bio-inspirados, tais como, algoritmos genéticos [19] e colônia de formigas [22]. Uma outra alternativa, explorada e detalhada nesta dissertação, é baseada em técnicas de cálculo numérico e identificação iterativa de mínimos. Nessa linha existe a técnica do gradiente determinístico, também conhecido como Hill-Climbing.

Algoritmo do Gradiente Determinístico

O algoritmo do gradiente determinístico busca o mínimo da função partindo de uma condição inicial $\mu(0)$, e iterações sucessivas são realizadas conduzindo a variável μ à solução desejada. A adaptação dos parâmetros utiliza a informação sobre o gradiente da função no ponto $\mu(n)$ para determinar o valor de $\mu(n+1)$, conforme a equação:

$$\overrightarrow{\mu}(n+1) = \overrightarrow{\mu}(n) - \alpha \nabla J(\overrightarrow{\mu}(n))$$

O parâmetro α é chamado de passo de adaptação, pois determina qual será o tamanho do passo dado a cada iteração. A convergência ou não do algoritmo está vinculada à escolha adequada do valor de α.

Metodologia para a Extração de Conceitos

O processo de extração de conceitos será executado de forma iterativa identificando, primeiramente, os conceitos com os menores valores para a função objetivo J, pois representam os conceitos mais importantes. No exemplo de função objetivo apresentada anteriormente a seqüência de conceitos extraídos deveria ocorrer na seguinte ordem: $\overrightarrow{\mu_4}$, $\overrightarrow{\mu_1}$, $\overrightarrow{\mu_6}$, $\overrightarrow{\mu_5}$, $\overrightarrow{\mu_3}$, $\overrightarrow{\mu_2}$.

A adoção do método iterativo para a extração de conceitos foi baseada em algumas vantagens trazidas pelo método. Dentre as vantagens pode-se citar: algoritmo menos complexo se comparado com as técnicas que processam todos os conceitos simultaneamente; o procedimento de extração permite que o usuário inspecione os conceitos obtidos; permite a introdução de métricas para a determinação do número adequado de conceitos; e permite atualizar as ontologias quando novos documentos são adicionados a base de documentos sem a necessidade de se recomputar os conceitos anteriores. Obviamente que esta técnica também apresenta desvantagens como, por exemplo, a não existência de um procedimento de revisão dos conceitos obtidos.

A aplicabilidade do método iterativo no modelo ACT somente será possível se as seguintes condições forem satisfeitas: o algoritmo de otimização deve garantir a extração do conceito ótimo global e o processo de extração deve levar em consideração os conceitos já extraídos.

Identificação do ótimo global

A utilização do algoritmo do gradiente determinístico não garante encontrar o ótimo global; por este motivo, novas estratégias foram adotadas para que o valor obtido seja ótimo ou mais próximo possível do ótimo. A estratégia escolhida é baseada em técnicas de computação evolutiva, mais especificamente, em técnicas utilizando algoritmos genéticos.

O algoritmo proposto utiliza técnicas exploratórias e explotatórias. A técnica exploratória utiliza uma população cujos indivíduos estão distribuídos pelo espaço de solução e técnicas de reprodução, mutação e seleção natural são aplicados na população para a identificação dos melhores indivíduos. A técnica explotatória faz uso do algoritmo de gradiente determinístico para melhorar os indivíduos da população.

O algoritmo genético utilizado, mostrado em 3, possui as seguintes características:

- Populacional: um conjunto de candidatos a solução é utilizado em cada geração com o objetivo de explorar o espaço de solução;
- Mutação;
- Crossover não é utilizado;
- Inserção de novos indivíduos - novos indivíduos são inseridos na população em cada geração;

- Seleção elitista do melhor indivíduo - somente o melhor indivíduo é mantido e poderá se reproduzir;
- Número fixo de gerações.

Input: P: Algoritmo de Busca Local
Input: O: Função Objetivo
Input: C_0: Agrupamento Inicial
Output: C: Agrupamento otimizado
begin
  população ⟵ iniciaPopulação;
  Adiciona C_0 à população;
  for $i \leftarrow 0$ **to** *número máximo de iterações* **do**
    população ← adicionaNovosIndivíduos;
    filhos ← clona(população);
    filhos ← mutaciona(filhos);
    população ← adicionaIndivíduos(filhos);
    foreach *indivíduo na população* **do**
      otimize o indivíduo aplicando o algoritmo P;
      calcula o fitness aplicando a função objetivo O;
    end
    $C \leftarrow$ seleciona o melhor indivíduo;
    população ← C;
  end
end

Algoritmo 3: Algoritmo Genético de Busca Global

Identificação de novos conceitos

A identificação de novos conceitos poderia ser feita alterando-se o valor da condição inicial $\mu(0)$ e assim o algoritmo convergiria para diferentes mínimos locais. Contudo, essa técnica pode demandar diversas execuções do algoritmo para se encontrar os mínimos mais relevantes. Dessa forma, uma técnica diferente foi desenvolvida. Consiste basicamente de dois estágios:

- Identificação da região promissora: A identificação da região promissora é realizada executando-se o algoritmo de minimização com a função objetivo modificada de maneira a penalizar a formação de conceitos semelhantes àqueles já extraídos. A nova função objetivo torna-se:

$$J_i = \varphi + \eta \cdot \rho + \tau \cdot \theta$$

onde:

θ é a função de penalização;

τ é uma constante que representa a influência da função θ, e será denominado de fator de disjunção.

A função θ será escolhida com base na propriedade de Disjunção de Conceitos apresentada na seção 4.3 onde conceitos totalmente disjuntos resultam em $\theta = 0$ e conceitos similares resultam em valores próximos a 1. O termo aditivo θ é dado por:

$$\theta\left(T, \bigcup_{i=1}^{N_C} C_i\right) = \sum_{i=1}^{N_C} \text{prox}\,(T, C_i)$$

onde prox é uma função que mede a proximidade entre dois conceitos e é dado por:

$$\text{prox}\,(T, C_i) = \frac{1}{T} \sum_{j \in T} \max_{l} \left(\bigcup_{l \in C_i} m_{jl} \right)$$

- Refinamento: A fase de refinamento visa determinar corretamente o ponto de mínimo uma vez que a utilização da função θ pode causar desvios em sua localização. O refinamento corrigirá o desvio provocado pela função θ por meio da re-execução do algoritmo determinístico, mas neste caso a função de penalização não é aplicada e a condição inicial é dada pelo ponto de mínimo calculado anteriormente durante a fase de identificação da região promissora.

4.4.4 Validação do Modelo

Esta seção visa validar a técnica, algoritmo e função objetivo propostos anteriormente, bem como a escolha das funções matemáticas para φ e f.

As funções escolhidas são:

$$f_{d_{i,j}}\left(\mu_i, \mu_j\right) = \left(\mu_i \cdot \mu_j\right)^m \cdot d_{i,j}^2$$

$$\varphi\left(\bigcup_{i,j \in M} f_{d_{i,j}}\left(\mu_i, \mu_j\right) \right) = \sum_{i \in M} \sum_{j \in M} f_{d_{i,j}}\left(\mu_i, \mu_j\right)$$

A escolha dessas funções foi feita baseada em algumas considerações de ordem geral e outras de ordem específica.

As considerações gerais sobre a escolha das funções levam em conta os critérios abaixo:

- Funções matemáticas simples. Neste caso, simplicidade incluem simplicidade na forma, no cálculo dos valores e das derivadas parciais;
- Funções que levam a função objetivo a se assemelhar à função-objetivo dos métodos tradicionais de agrupamento.

As considerações de ordem mais específica são:

- A função f foi escolhida de tal forma que a importância de uma aresta para a composição do grupo será maior em casos onde ambos os termos sejam importantes.
- A função φ foi escolhida de modo a agrupar termos que sejam mutuamente próximos.

Função Objetivo
A função objetivo obtida é dada por:

$$J\left(\overrightarrow{\mu}\right) = \sum_{i=1}^{M}\sum_{j=1}^{M}\left(\mu_i \cdot \mu_j\right)^m \cdot d_{i,j}^2 + \eta\sum_{i=1}^{M}\left(1-\mu_i\right)^m$$

O algoritmo utilizado para encontrar os pontos de mínimo da função objetivo acima é o gradiente determinístico.

$$\overrightarrow{\mu}\left(n+1\right) = \overrightarrow{\mu\left(n\right)} - \alpha \cdot \nabla J\left(\overrightarrow{\mu}\left(n\right)\right)$$

O gradiente da função J é dado por:

$$\nabla J\left(\overrightarrow{\mu}\left(n\right)\right) = \left[\frac{\partial J}{\partial \mu_1}\frac{\partial J}{\partial \mu_2}\cdots\frac{\partial J}{\partial \mu_k}\cdots\frac{\partial J}{\partial \mu_M}\right]$$

Onde o termo geral $\partial J/\partial \mu_k$ é dado por:

$$\frac{\partial J}{\partial \mu_k} = m\left(\mu_k\right)^{m-1} \cdot \sum_{\substack{i=1\\ i\neq k}}^{M}\left(\mu_i^m d_{i,k}^2\right) - m\eta\left(1-\mu_k\right)^{m-1}$$

Para verificar se a função-objetivo é realmente capaz de agrupar termos correlatos, um sistema composto de seis termos será utilizado. Três termos são relativos ao assunto algoritmos genéticos e os outros três termos são relativos ao assunto aprendizado de máquina. O grafo apresentado na figura 4.18 mostra os seis termos e as arestas conectando os termos rotuladas com a correlação entre os termos.

Evidentemente, os termos associados a um determinado assunto de pesquisa possuem maior correlação entre si. Observe os arcos mais grossos unindo os termos {genetic, operator, algorithm} e os

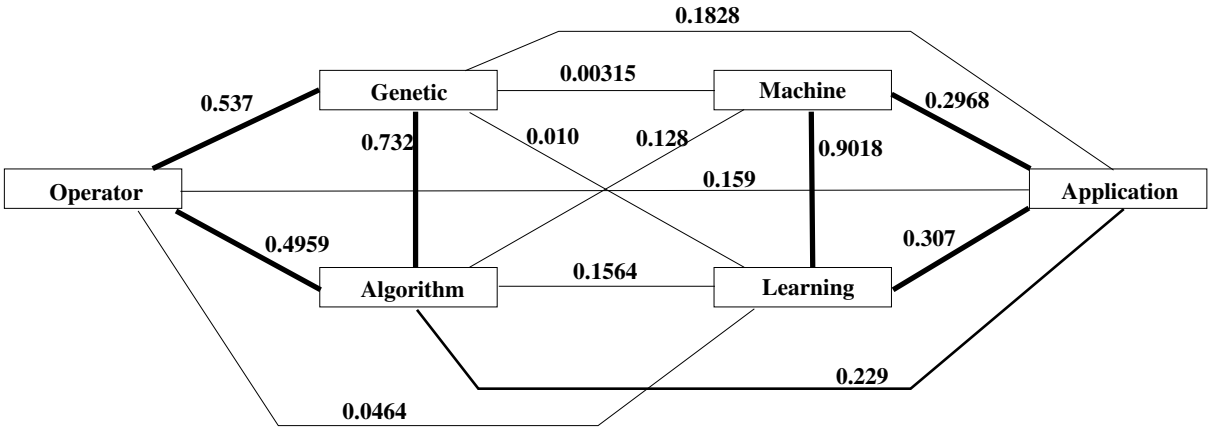


Fig. 4.18: Exemplo de Grafo

termos { machine, learning, application}. Deste modo, espera-se que o algoritmo agrupador produza dois grupos.

A seguir é mostrado os resultados da aplicação do algoritmo para diferentes valores de m e η. Em todos os casos, o resultado é um vetor com o valor de pertinência de cada termo ao grupo.

$$\overrightarrow{\mu} = [\text{genetic algorithm operator machine learning application}]$$

Os testes apresentados avaliam o comportamento do algoritmo em diversas situações para os parâmetros de entrada m e η e são comparados ao comportamento de outros agrupadores. O objetivo é verificar se a função objetivo proposta e a respectiva técnica de solução são capazes de realizar o agrupamento de objetos em espaços não-métricos.

Cenário: Obtenção do conceito nada

Este cenário valida a tese de que para η nulo o algoritmo retorna um conceito que não contém um único termo. O conceito mostrado abaixo foi obtido com $\eta = 0$.

```
conceito Nada ≡ {
        genetic(0.00),
        algorithm(0.00),
        operator(0.00),
        machine(0.00),
        learning(0.00),
        application(0.00)
 }
```

Assim, como no caso PCM, o resultado é um grupo que não contém objetos.

Cenário: Obtenção do conceito coisa

Este cenário valida a tese de que para η suficientemente grande o algoritmo retorna um conceito que inclui todos os termos. O conceito mostrado abaixo foi obtido com $\eta = 50$.

```
conceito Coisa ≡ {
       genetic(1.0),
       algorithm(1.0),
       operator(1.0),
       machine(1.0),
       learning(1.0),
       application(1.0)
 }
```

Novamente, o resultado mostra a obtenção de apenas um grupo contendo todos os termos. Esse resultado é semelhante ao comportamento do agrupador PCM.

Cenário: Obtenção de conceitos com máximo índice de fuzzificação

Este cenário mostra que, assim como no caso do agrupador PCM, valores elevados para o fator de fuzzificação promove a formação de grupos cujos índices de pertinência dos objetos aos grupos são aproximadamente iguais. O conceito mostrado abaixo apresenta o conceito obtido para $m = 5$.

```
conceito Desconhecido ≡ {
       genetic(0.56),
       algorithm(0.57),
       operator(0.56),
       machine(0.56),
       learning(0.56),
       application(0.56)
 }
```

Cenário: Obtenção de conceitos com mínima fuzzificação

Assim como nos agrupadores FCM e PCM, o valor de $m = 1$ promove identificação rígida de grupos. Os conceitos apresentados abaixo mostram que o algoritmo identificou corretamente os dois conceitos do grafo e atribuiu valor de pertinência 1 aos termos que pertencem ao grupo e 0 caso contrário

```
conceito Algoritmos Genéticos ≡ {
       genetic(1.00),
```

```
conceito Aprendizado de Máquina ≡ {
         genetic(0.00),
```

```
        algorithm(1.00),
        operator(1.0),
        machine(0.00),
        learning(0.00),
        application(0.00),
}
```

```
        algorithm(0.00),
        operator(0.00),
        machine(1.00),
        learning(1.00),
        application(1.00)
}
```

Cenário: Avaliação do parâmetro m

Neste cenário, o objetivo é mostrar que alterações no valor do parâmetro m provocam maior ou menor distribuição dos valores de pertinência.

Os conceitos apresentados abaixo foram obtidos com $\eta = 1$ e $m = 2$.

```
conceito Algoritmos Genéticos ≡ {
        genetic(0.423),
        algorithm(0.444),
        operator(0.286),
        machine(0.107),
        learning(0.111),
        application(0.141),
}
```

```
conceito Aprendizado de Máquina ≡ {
        genetic(0.197),
        algorithm(0.246),
        operator(0.201),
        machine(0.993),
        learning(0.994),
        application(0.316)
}
```

Os conceitos apresentados abaixo foram obtidos com $\eta = 1$ e $m = 1.5$

```
conceito Algoritmos Genéticos ≡ {
        genetic(0.994),
        algorithm(0.993),
        operator(0.489),
        machine(0.054),
        learning(0.059),
        application(0.099),
}
```

```
conceito Aprendizado de Máquina ≡ {
        genetic(0.065),
        algorithm(0.110),
        operator(0.069),
        machine(0.944),
        learning(0.948),
        application(0.219)
}
```

Os resultados deste cenário mostram que o agrupador proposto possuem o mesmo comportamento de outros agrupadores tradicionais, como PCM e FCM. E também promove agrupamento com maior índice de fuzzificação à medida que o valor de m torna-se maior.

Cenário: Avaliação do parâmetro η

Neste cenário, o objetivo é mostrar que alterações no valor do parâmetro η causam a identificação de grupos maiores a medida que o valor de η aumenta.

Os conceitos apresentados abaixo foram obtidos com $\eta = 1$ e $m = 2$.

```
conceito Algoritmos Genéticos ≡ {
       genetic(0.423),
       algorithm(0.444),
       operator(0.286),
       machine(0.107),
       learning(0.111),
       application(0.141),
 }
```

```
conceito Aprendizado de Máquina ≡ {
       genetic(0.197),
       algorithm(0.246),
       operator(0.201),
       machine(0.993),
       learning(0.994),
       application(0.316)
 }
```

Os conceitos apresentados abaixo foram obtidos com $\eta = 1.5$ e $m = 2$

```
conceito Algoritmos Genéticos ≡ {
       genetic(0.994),
       algorithm(0.995),
       operator(0.525),
       machine(0.266),
       learning(0.275),
       application(0.331),
 }
```

```
conceito Aprendizado de Máquina ≡ {
       genetic(0.265),
       algorithm(0.322),
       operator(0.268),
       machine(0.995),
       learning(0.995),
       application(0.399)
 }
```

Os resultados deste cenário mostram que o agrupador proposto possui o mesmo comportamento de outros agrupadores clássicos, como PCM e FCM. O tamanho dos grupos é função do parâmetro η.

A função-objetivo e o algoritmo propostos apresentaram bons resultados, identificando corretamente o agrupamento de termos correlatos. No entanto, a aplicabilidade do modelo torna-se inviável quando o número de termos é elevado. As dificuldades deste modelo estão em dois aspectos, o primeiro refere-se à complexidade computacional para a obtenção dos conceitos e o segundo é relativo à escolha adequada dos parâmetros da função objetivo e dos parâmetros do algoritmo de Hill-Climbing.

4.4.5 Simplificação do Modelo

As dificuldades apresentadas anteriormente motivaram a simplificação do modelo de modo a tornar sua aplicação factível. As simplificações ocorreram de duas formas: remoção de termos não relevantes ao processo de agrupamento e aplicação de restrições aos valores das associações fuzzy entre termos e conceitos.

Remoção de termos não relevantes

O objetivo da remoção de termos não relevantes da fase de extração de conceitos é diminuir o esforço computacional na determinação dos conceitos e aumentar a imunidade a ruídos causados por termos que não agregam muita informação ao processo de agrupamento.

A seleção dos termos que participam do processo de agrupamento é baseada no valor de entropia E do termo, cuja definição é dada abaixo:

$$E(t) = \frac{1}{N} \sum_{i=1}^{N} dk_{i,t} \log(dk_{i,t}) \tag{4.8}$$

Uma vez computado o valor de entropia de todos os termos, serão selecionados termos cujos valores de entropia estejam entre α e β, onde $0 \leq \alpha < \beta \leq 1$. Assim como proposto em Velardi *et al.* [72], esta filtragem irá remover ruídos causados por: termos muito comuns (entropias maiores que β) e termos que ocorrem em poucos documentos (entropias menores que α).

Restrição aos valores de pertinência dos termos

Os valores das associações fuzzy entre termos e conceitos serão restritos aos valores 0 e 1, tornando rígido o modelo de agrupamento. Mas, devido à importância da relação fuzzy para o modelo de ontologia, existirá um estágio de recuperação dos índices fuzzy.

A restrição imposta transforma o problema de otimização contínua em um problema de otimização discreta, eliminando a necessidade de se especificar o parâmetro de fuzzificação m e a necessidade de se computar as derivadas parciais. A função objetivo torna-se mais simples e o parâmetro de entrada para a função J não é mais um vetor de números reais e, sim, um conjunto de termos, como mostrado na equação 4.9. Agora, o resultado do processo de agrupamento não é mais um conceito, pois ainda existe um processamento adicional para a recuperação das associações fuzzy, mas um protótipo de conceito que auxiliará na obtenção das associações fuzzy.

$$J(A) = \sum_{i \in A} \sum_{j \in B} d_{i,j}^2 + \eta \cdot (|A| - M) + \theta \left(T, \bigcup_{i=1}^{N_C} C_i \right) \tag{4.9}$$

onde:
A: conjunto de termos que compõe o protótipo de conceito
$|A|$: número de termos no conjunto A
M: número de termos escolhidos pela filtragem
C_j: j-ésimo conceito extraído
$|C_j|$: número de termos que compõe o protótipo de conceito C_j

A identificação do conjunto de termos pode ser vista como a identificação de sub-matrizes de M que atendam ao critério de semelhança entre termos. No exemplo apresentado anteriormente existem duas sub-matrizes de M, uma seria dada pelo agrupamento dos termos { machine, learning, application } e a outra seria dada pelo agrupamento dos termos { operator, genetic, algorithm },

conforme ilustra a matriz abaixo. Os rótulos a, b, c, d, e, f representam respectivamente os termos genetic, algorithm, operator, machine, learning, application.

$$M = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{bmatrix} 1 & 0.732 & 0.537 & 0.003 & 0.128 & 0 \\ 0.732 & 1 & 0.496 & 0.01 & 0.156 & 0.046 \\ 0.537 & 0.496 & 1 & 0.183 & 0.229 & 0.159 \\ 0.003 & 0.01 & 0.183 & 1 & 0.902 & 0.296 \\ 0.128 & 0.156 & 0.229 & 0.902 & 1 & 0.307 \\ 0 & 0.046 & 0.159 & 0.296 & 0.307 & 1 \end{bmatrix} \end{array}$$

É importante ressaltar que as sub-matrizes de M não são necessariamente formadas por colunas e linhas contíguas. O requisito para a formação de uma sub-matriz é apenas a lista de linhas e colunas. Assim, uma sub-matriz que seja definida por um sub conjunto I de linhas e por um sub conjunto J de colunas deve possui as seguintes características: o número de linhas é dado pela cardinalidade do conjunto I, o número de colunas é dado pela cardinalidade do conjunto J, e os elementos a_{ij} da sub-matriz são dados pelos elementos da matriz original cujas linhas sejam indexadas pelos elementos do conjunto I e as colunas sejam indexadas pelos elementos do conjunto J. Por exemplo, na figura 4.19, a primeira sub-matriz foi obtida selecionando as linhas de número 1 e 3 e as colunas de número 1 e 4. Já, a segunda sub-matriz foi obtida selecionando as linhas de número 2 e 3 e as colunas de número 3 e 4.

$$\begin{bmatrix} 1 & 2 & 3 & 1 \\ 3 & 3 & 2 & 2 \\ 1 & 2 & 2 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \begin{matrix} \text{I} = \{1,3\} \\ \text{J} = \{1,4\} \end{matrix}$$
$$\begin{bmatrix} 2 & 2 \\ 2 & 1 \end{bmatrix} \quad \begin{matrix} \text{I} = \{2,3\} \\ \text{J} = \{3,4\} \end{matrix}$$

Fig. 4.19: Duas sub-matrizes extraídas da matrix original, formando dois grupos.

A identificação de sub-matrizes que compartilham alguma característica em comum já foram trabalhadas utilizando técnicas de bi-clusterização [26] e trouxeram resultados promissores. Por esse motivo, o novo algoritmo de minimização utilizará as técnicas de bi-clusterização.

O algoritmo 4 proposto para a otimização da função 4.9 é baseado na estratégia de bi-agrupamento de Cheng e Church [12]. O algoritmo inicia com uma proposta de agrupamento inicial e realiza melhoras no agrupamento por meio de três operações: inclusão de termos não presentes no agrupamento, pela remoção de termos que pertencem ao agrupamento, e se alguma das regras anteriores não

for aplicável então tente trocar os termos do agrupamento. O algoritmo termina quando a aplicação dessas estratégias não traz nenhuma melhora ao agrupamento.

Input: M: Matriz de Correlação
Input: O: Função Objetivo
Input: C_0: Agrupamento Inicial
Output: C: Agrupamento otimizado
begin
  $C \leftarrow C_0$
  CondiçãoFinal ← falso
  while *CondiçãoFinal não é verdadeira* **do**
    bomTermo ← encontreBomTermo(C,M,O)
    if *bomTermo encontrado* **then**
      Adicione bomTermo ao grupo C
    end
    mauTermo ← encontreMauTermo(C,M,O)
    if *mauTermo encontrado* **then**
      Remova o mauTermo de C
    end
    if *bomTermo ou mauTermo não foram encontrados* **then**
      (goodTerm,badTerm) ← encontreTermosParaAlterar(C,M,O)
      if *bomTermo e mauTermo foram encontrados* **then**
        Adicione bomTermo a C
        Remova mauTermo de C
      end
      else
        CondiçãoFinal ⟵ verdadeiro
      end
    end
  end
end

Algoritmo 4: Algoritmo Determinístico de Busca Local

Recuperação das associações fuzzy

A recuperação das associações fuzzy dos termos aos conceitos é baseada nos agrupamentos obtidos pela técnica de bi-clusterização e em propriedades matemáticas de aproximação por funções ortogonais. Os termos obtidos pelo agrupamento são utilizados para compor uma base vetorial ortogonal, expressa na forma de uma matriz T na qual as colunas representam os termos que formam o

protótipo e as linhas representam o conjunto de documentos. Como a matriz T não é uma base ortogonal, faz-se necessário um procedimento de ortogonalização para que seja possível aplicar a técnica de aproximação. O método de ortogonalização escolhido foi a decomposição por valores singulares, pois possui a propriedade de fornecer a importância de cada vetor da base ortogonal, propriedade que será utilizada na fase de atribuição de nomes aos conceitos. Assim, a nova base escolhida é dada pela matriz U:

$$T_{N\times L} = U_{N\times R} \cdot S_{R\times R} \cdot V'_{R\times L}$$

A associação fuzzy dos termos ao conceito é obtida calculando-se o cosseno do ângulo formado pelo vetor termo e o vetor termo aproximado. O vetor termo aproximado é calculado utilizando-se a aproximação quadrática média por funções ortogonais, conforme descrito a seguir.

Seja $\varphi_1, \ldots, \varphi_n$ funções contínuas e não identicamente zero no intervalo $[a, b]$ tal que:

$$\int_a^b \varphi_j(x) \cdot \varphi_k(x)\, dx = 0 \;\text{ se }\; j \neq k$$

Dada uma função contínua no intervalo $[a, b]$, os valores de $c_1, \ldots, c_n$ que minimizam

$$F(c) = \int_a^b |f - \sum_{i=1}^{n} c_i \varphi_i(x)|^2 dx = 0$$

são dados por

$$c_i = \frac{\langle f, \varphi_i(x)\rangle}{\langle \varphi_i(x), \varphi_i(x)\rangle}$$

onde

$$\langle \Psi, \varphi\rangle = \int_a^b \Psi(x)\, \varphi(x)\, dx$$

Desta forma, o vetor termo aproximado será dado por:

$$\widehat{t_j} = \sum_{i=1}^{n} c_i \cdot u_i = \sum_{i=1}^{n} \frac{\langle t_j, u_i\rangle}{\langle u_i, u_i\rangle} u_i = \sum_{i=1}^{n} \text{proj}_{u_i} t$$

sendo $\widehat{t_j}$ o vetor aproximado do j-ésimo termo, u_i é o vetor i da base vetorial U, n é o número de vetores que compõe a base vetorial.

A associação fuzzy entre o termo e o conceito é dado pelo cosseno do ângulo formado pelos

vetores $\overrightarrow{t_i}$ e $\widehat{t_i}$

$$ro_i = \cos\left(\overrightarrow{t_i}\widehat{t_i}\right)$$

Considerações sobre a simplificação

A simplificação adotada para o modelo teve o objetivo de tornar factível a aplicação dos algoritmos de agrupamento e facilitar a escolha dos parâmetros de configuração tais como o parâmetro de fuzzificação m da função objetivo, o parâmetro α e o critério de parada do algoritmo Hill-Climbing. A simplificação ainda acrescentou um nova vantagem ao modelo, permitindo que os conceitos possam ser definidos de maneira intencional. Agora, o índice de associação fuzzy entre termos e conceitos é baseado na definição matemática apresentada em 4.10.

$$\text{conceito Algoritmos Genéticos} \equiv \{(t_i, \mu_i) | t_i \in L, \mu_i = \cos\left(\overrightarrow{t_i}\widehat{t_i}\right)\} \tag{4.10}$$

No modelo original, um conceito é caracterizado pelo nome e a lista de termos com os respectivos graus de pertinência, como mostrado abaixo. Nesse modelo, que segue o paradigma extensional, a relação de termos é fixa, de modo que qualquer alteração na lista de termos irá invalidar os valores das pertinências ao conceito.

```
conceito Algoritmos Genéticos ≡ {
       genetic(0.423),
       algorithm(0.444),
       operator(0.286),
       machine(0.107),
       learning(0.111),
       application(0.141),
 }
```

O modelo decorrente da simplificação alterou para intencional a forma como um conceito é definido. A pertinência é dada pela definição 4.10. Assim, se um novo termo for adicionado ao vocabulário da ontologia, os valores de pertinência desse termo aos conceitos podem ser obtidos apenas executando o cálculo apresentado na seção 4.4.5.

Determinação do número de conceitos

A determinação do número de conceitos é realizada com o auxílio de uma medida de semelhança entre o protótipo de conceito obtido pelo agrupador e os protótipos de conceito já extraídos. Para

cada protótipo extraído da base de documentos, o valor de semelhança é computado entre eles. É escolhido o maior valor de semelhança e, se este for maior que algum valor limite γ, então o processo de extração é finalizado.

O valor de semelhança entre o conceito obtido e o conjunto de conceitos já extraídos é dado por:

$$\text{prox}\,(A, C_i) = \sum_{j \in A} \max_{l} \left(\bigcup_{l \in C_i} m_{jl} \right)$$

onde A é o protótipo de conceito obtido pelo agrupador, C_i é o i-ésimo protótipo de conceito extraído e m_{jl} é a correlação entre os termos k do agrupamento A e o termo l do agrupamento C_i.

4.4.6 Atribuição de Nomes às Relações

A atribuição de nomes aos conceitos utilizará a estratégia de Holger [4] de extração de ontologias. Os termos que constituem o protótipo de um conceito são submetidos ao algoritmo de Holger para que este forneça a taxonomia dos termos. Extraída a taxonomia dos termos, pode-se decidir por escolher nomes simples ou compostos para o conceito. A escolha de nomes simples é feita utilizando o termo mais geral da taxonomia. Por exemplo, um protótipo de conceito formado por termos relativos ao assunto algoritmos genéticos fornece a seguinte hierarquia de termos.

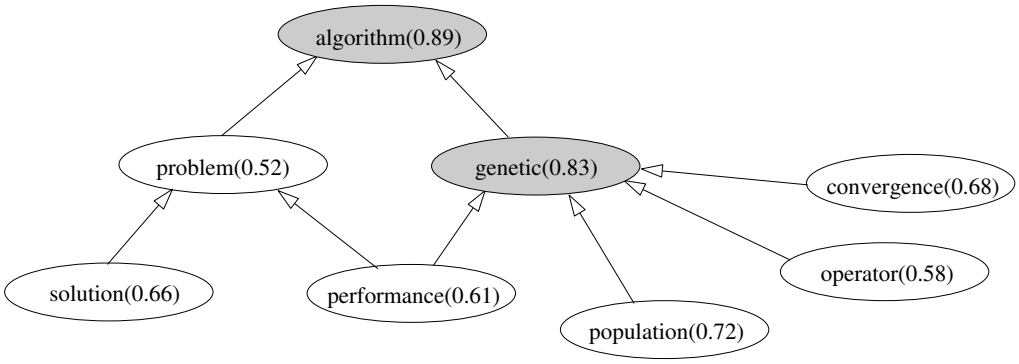


Fig. 4.20: Taxonomia para o assunto Algoritmos Genéticos

O termo algorithm é o mais geral dessa estrutura e portanto será utilizado para nomear o conceito. Para os casos onde houver mais de um termo raiz a decisão sobre o termo escolhido é dado pela cálculo da associação fuzzy do termo quando se utiliza uma base vetorial composta somente pelo vetor mais importante. Assim:

$$\widehat{t_j} = c_1 \cdot u_1 = \frac{\langle t_j, u_1 \rangle}{\langle u_1, u_1 \rangle} u_1 = \text{proj}_{u_1} t_j$$

$$ro_i = \cos\left(\overrightarrow{t_i}\widehat{t_i}\right)$$

Para o exemplo apresentado abaixo, dos dois termos (technique e machine) raízes o termo machine será escolhido, pois é o que apresenta o maior valor de associação fuzzy, 0.85, quando apenas um vetor base é utilizado.

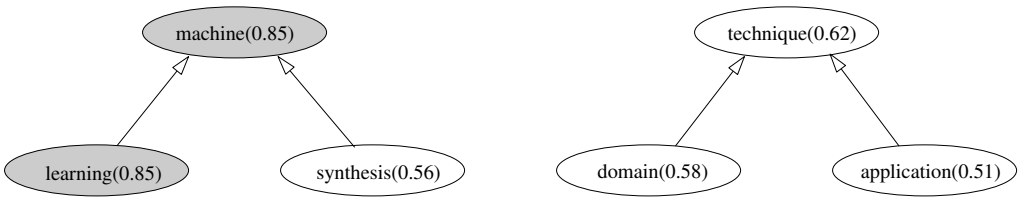


Fig. 4.21: Taxonomia para o assunto Aprendizado de Máquina

A escolha de nomes compostos segue o mesmo princípio mostrado anteriormente. Escolhe-se o termo mais geral da taxonomia, este termo na forma substantiva será o nome principal do conceito. O segundo nome é obtido escolhendo-se o termo mais geral que esteja subordinado ao primeiro nome, este termo na forma adjetivada constituirá o qualificador do primeiro nome. No exemplo apresentado, o primeiro nome na forma substantiva é "algorithm" o segundo nome da forma adjetivada é "genetic" assim o nome composto deste conceito será "genetic algorithm".

4.4.7 Identificação de Relações Taxonômicas: uma proposta

A descoberta Automática de relações taxonômicas entre conceitos não está no escopo desta dissertação. No entanto, é inevitável não falar sobre este importante recurso de ontologias que permitem descrever o universo de discurso nos mais diferentes níveis de abstração. A função objetivo proposta na seção 4.4.3 não inclui a identificação de relações taxonômicas, porém ao analisar-se o parâmetro η é possível concluir que este poderia ser utilizado para a determinação dessas relações. Antes de explicarmos a relevância deste parâmetro é necessário elucidar alguns fatos importantes:

Conceito Coisa (owl:Thing) O conceito "coisa" representa o conceito que contém todos os indivíduos. A conseqüência desta propriedade é que todos os demais conceitos são sub-conceitos do conceito "coisa". Na representação gráfica mostrada na figura 4.22, o conceito "coisa" é a raiz da árvore ontológica.

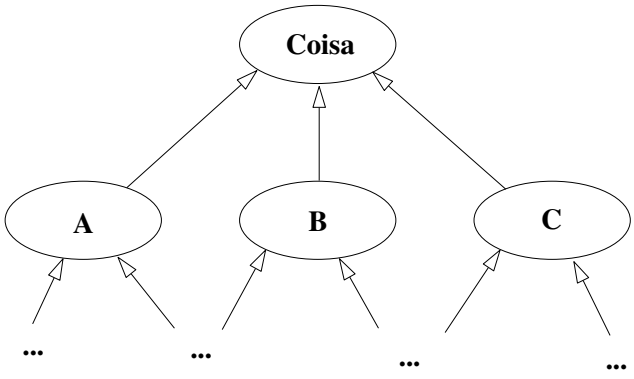


Fig. 4.22: Conceito mais abstrato

Conceito Nada (owl:Nothing) O conceito nada representa o conceito que não contém nenhum indivíduo. Devido a esta característica pode-se inferir que o conceito nada é também um sub-conceito de todos os outros conceitos. Na representação gráfica mostrada na figura 4.23, o conceito nada é o elemento ocupando o nível mais inferior da ontologia.

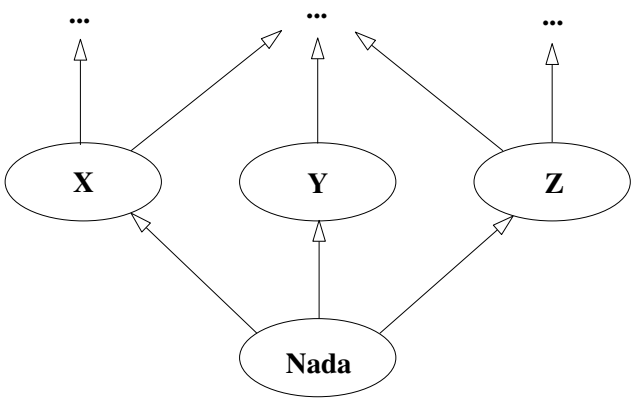


Fig. 4.23: Conceito mais específico

Conceitos Intermediários Os conceitos intermediários na estrutura taxonômica descrevem com mais ou menos detalhes uma determinada idéia de tal modo que conceitos próximos da raiz são mais genéricos e os próximos do conceito nada são os mais específicos.

Propriedades do parâmetro η O parâmetro η controla o tamanho do grupo durante o processo de agrupamento. Quanto maior o valor de η maior é o grupo formado. Em particular se η é zero o grupo formado é vazio representando o conceito nada. Se η é infinito o grupo formado inclui todos os termos representando o conceito coisa.

As constatações acima levam à conclusão de que existe um paralelo entre a escolha do parâmetro η e o nível de abstração dos conceitos obtidos no processo de agrupamento. Assim, uma possível forma de obter as relações taxonômicas envolveria a determinação de conceitos para diversos valores de η seguido de uma posterior análise dos conceitos obtidos.

Seja o exemplo da figura 4.24. O parâmetro η foi, inicialmente, escolhido com um valor η_1 e nessa configuração foram obtidos cinco conceitos V, W, X, Y, Z. O valor do parâmetro η foi alterado para η_2 é nessa nova configuração foram obtidos os conceitos M, O e P. Por fim, o parâmetro η foi alterado mais uma vez para o valor η_3 e nessa configuração somente o conceito A foi obtido.

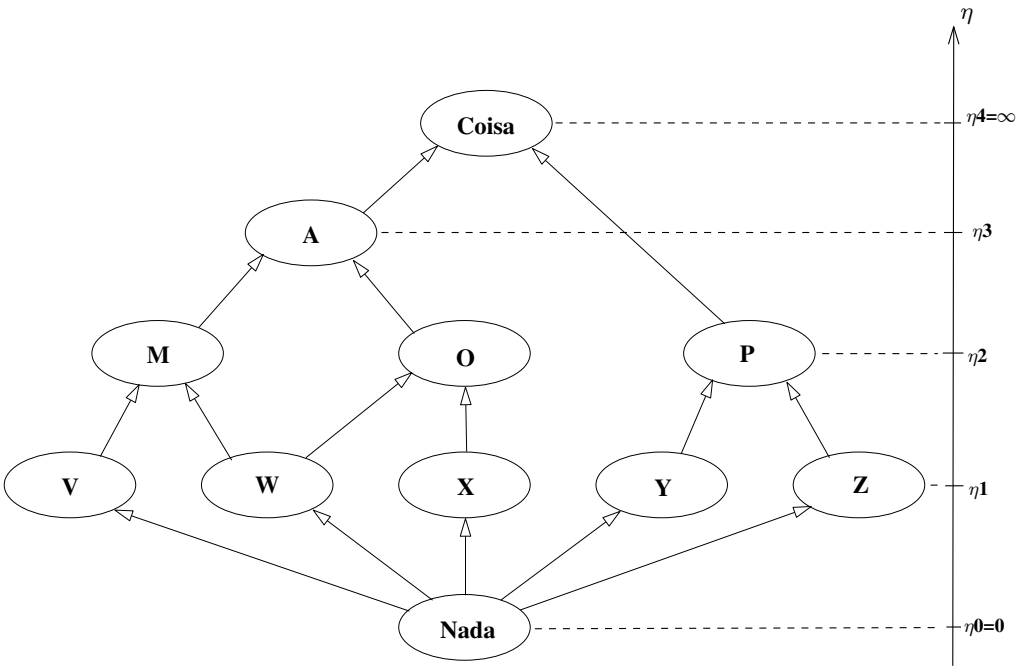


Fig. 4.24: Taxonomia de termos

Os conceitos obtidos são submetidos à análise e os mapeamentos desenvolvidos na seção 4.3 são utilizados para realizar inferências sobre as relações entre os conceitos. Para o exemplo apresentado, as relações presentes na tabela 4.2 seriam verdadeiras, permitindo deduzir as relações taxonômicas entre os conceitos.

Relação Taxonômica	*Propriedade Matemática*
$V \subset M$	$f_V(x) \leq f_M(x), x \in X$
$W \subset M$	$f_W(x) \leq f_M(x), x \in X$
$W \subset O$	$f_W(x) \leq f_O(x), x \in X$
$X \subset O$	$f_X(x) \leq f_O(x), x \in X$
$Y \subset P$	$f_Y(x) \leq f_P(x), x \in X$
$Z \subset P$	$f_Z(x) \leq f_P(x), x \in X$
$M \subset A$	$f_M(x) \leq f_A(x), x \in X$
$O \subset A$	$f_O(x) \leq f_A(x), x \in X$

Tab. 4.2: Relação entre taxonomia e as propriedades fuzzy

4.5 Extração Iterativa de Conceitos

Esta seção descreve os procedimentos executados pelo usuário para a extração de relações semânticas. O sistema desenvolvido utiliza a teoria e os algoritmos descritos anteriormente para prover um ambiente interativo e iterativo para a extração das relações.

O diagrama da figura 4.25 ilustra a interação do usuário com o sistema. Contrariando a idéia de um sistema totalmente automático, a participação do usuário no processo de extração é bastante grande. Entretanto, essa característica permite ao usuário inspecionar os resultados e atuar no sistema de forma a garantir bons resultados.

Inicialmente, o usuário ajusta os parâmetros do algoritmo genético e da função objetivo. Os parâmetros do algoritmo genético, tais como *número de gerações*, *número de indivíduos* e *número de mutações*, permitem regular o processo de busca pelo espaço de soluções, e os parâmetros da função objetivo regulam o nível de abstração η e o fator de disjunção τ do conceito extraído.

Ajustado os parâmetros, estes são passados ao algoritmo ACT descrito na seção 4.4.3 para a extração de um protótipo de conceito. O resultado é apresentado ao usuário para aprovação. Neste passo, o usuário pode ser auxiliado no processo de aprovação por meio da inspeção do protótipo extraído ou também pela informação da taxa de sobreposição entre os conceitos.

O protótipo de conceitos que foi aprovado deve receber um nome. O usuário pode simplesmente atribuir um nome que desejar, ou ainda, pode requisitar uma sugestão de nome. O algoritmo de sugestão é aquele discutido na seção 4.4.6 que utiliza o algoritmo de Holger.

Definidos o nome do protótipo e seus termos constituintes, calcula-se as associações fuzzy entre os termos e o protótipo pela aplicação do algoritmo de recuperação de associações fuzzy descrito na seção 4.4.5.

Por fim, o algoritmo retorna ao passo inicial onde o usuário pode extrair um novo protótipo ou

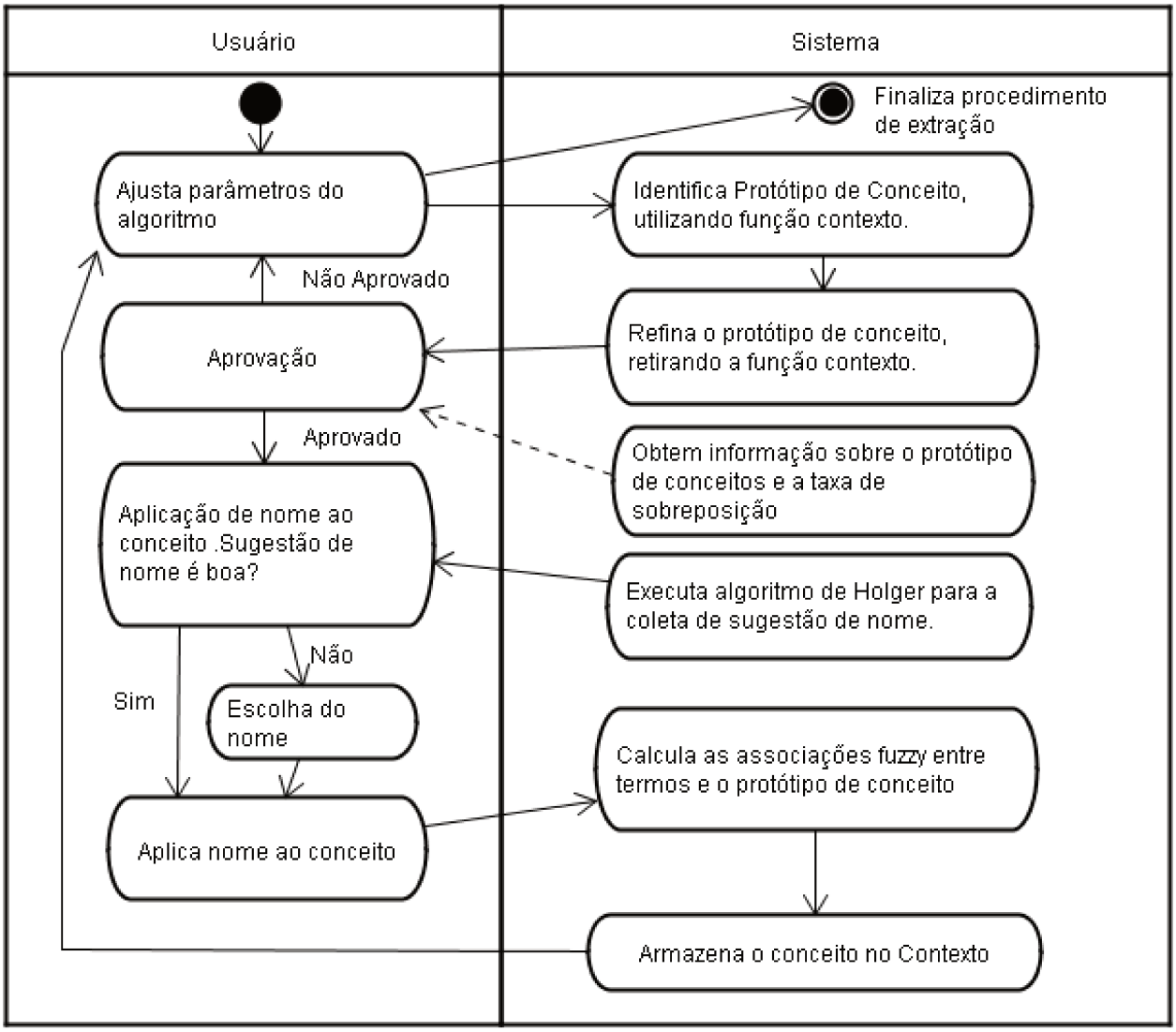


Fig. 4.25: Fluxo de Interação Usuário-Sistema

terminar a execução do algoritmo.

4.6 Considerações Finais

Neste capítulo, foi discutida a estratégia para a identificação de termos, conceitos e suas associações. A metodologia adotada parte da proposta possibilística para agrupamento de termos correlatos e introduz modificações necessárias junto à função-objetivo para que seja possível o agrupamento de dados na qual somente a informação de distância é conhecida. A seguir, averigua-se, por meio de um pequeno exemplo, se a nova função-objetivo é capaz de agrupar corretamente os termos. Observou-se

que a proposta atende aos requisitos do agrupador possibilístico; entretanto, a técnica para a solução não é factível devido à complexidade computacional e dos ajustes necessários nos parâmetros de entrada do algoritmo. Em virtude desses problemas é que foi realizada uma simplificação na proposta inicial. A simplificação transformou o problema em um problema de otimização inteira que foi resolvido utilizando técnicas de bi-agrupamento.

Por fim, foram discutidos procedimentos para a aplicação automática de nomes e identificação de relações taxonômicas. A atribuição de nomes a conceitos utiliza o algoritmo de Holger para escolher dentre os termos que compõem um protótipo de conceitos os termos mais abstratos. As relações taxonômicas foram discutidas e foi apresentada uma possível técnica para a sua identificação.

Os métodos e algoritmos desenvolvidos neste capítulo são aplicados a 4 conjunto de documentos de diferentes características e os resultados são apresentados no próximo capítulo (Capítulo 5). O próximo capítulo também apresenta o protótipo que foi desenvolvido para a obtenção dos resultados.

Capítulo 5

Resultados

Este capítulo apresenta os resultados obtidos com a aplicação da técnica ACT para a extração de relações semânticas de um conjunto de documentos. Os testes foram realizados utilizando quatro conjuntos de documentos com diferentes características em relação ao seu conteúdo. Os dois primeiros conjuntos são compostos de artigos que discutem um número fixo e conhecido de assuntos. O terceiro conjunto é composto de artigos de um único assunto. E o último conjunto é composto de artigos de várias área do conhecimento coletados de forma aleatória. Os resultados obtidos foram avaliados frente aos requisitos do algoritmo de agrupamento e comparados a resultados obtidos por outros métodos.

5.1 Requisitos de Desempenho

A avaliação da qualidade dos conceitos extraídos pelo algoritmo ACT deve levar em consideração três fatores:

1. **Número de conceitos:** O algoritmo deve identificar o "melhor" número de conceitos que descreve os assuntos tratados pelos documentos. Obviamente, este critério é subjetivo e apresenta uma grande dependência dos parâmetros de entrada do algoritmo, conforme pode ser comprovado na seção 5.4.1;
2. **Identificação do assunto:** O algoritmo deve agrupar corretamente termos que estejam semanticamente próximos, caracterizando, dessa forma, um conceito;
3. **Atribuição de nomes aos conceitos:** O algoritmo deve ser capaz de atribuir nomes signicativos aos conceitos.

É importante mencionar que os critérios de validação de algoritmos de agrupamento também são úteis para a avaliação do algoritmo ACT de extração de conceitos, uma vez que a proposta é baseada

em agrupamento de dados. Esses critérios de avaliação, segundo Han e Kamber [36], são:

Escalabilidade: Os algoritmos apresentam geralmente um melhor desempenho computacional no agrupamento de um número de dados quando comparado com o desempenho para um conjunto maior de dados. Há a necessidade de se preservar este desempenho computacional quando se trata de um número considerável de dados.

Habilidade de se adaptar aos diferentes tipos de atributos: Além de dados numéricos, para os quais muitos dos algoritmos são projetados, existem aplicações que requerem agrupamentos de outros tipos de dados, como binário, caracteres ou uma mistura destes.

Encontrar grupos com formas arbitrárias: Como observado na seção 3.3, a escolha da função kernel permite a identificação de formatos diferentes dos esféricos. Estudar e aplicar outras funções de kernel torna-se importante para desenvolver algoritmos que possam detectar grupos de formas arbitrárias.

Número mínimo de parâmetros: O parâmetro típico solicitado pelos algoritmos é o número desejado de grupos. No entanto, outros parâmetros podem ser necessários, e o resultado do agrupamento pode ser sensível à variação desses parâmetros. Em geral, valores apropriados para os parâmetros são difíceis de determinar. Portanto, quanto menor o número de parâmetros e quanto mais independente o algoritmo for do parâmetro, melhor o algoritmo.

Habilidade de trabalhar com ruído: Os ruídos são observações que se desviam da média de outras observações de forma que se suspeite de que estas observações foram geradas por um mecanismo diferente. Há a necessidade de se verificar o comportamento dos algoritmos de agrupamento de dados perante os ruídos porque os conjuntos de dados do mundo real contêm erros, pontos isolados, desconhecidos ou dados imprecisos.

Insensibilidade à ordem de apresentação dos dados: Alguns algoritmos são sensíveis à ordem em que os dados são apresentados. Alguns conjuntos de dados, quando apresentados em uma ordem diferente para um mesmo algoritmo, podem gerar grupos dramaticamente diferentes.

Alta dimensionalidade: Com a percepção humana é possível julgar a qualidade do agrupamento em até três dimensões. Ainda é um grande desafio encontrar e avaliar grupos em espaços de grande dimensão. Neste caso, para se avaliar os agrupamentos é necessário utilizar índices que verificam a qualidade destes. Existe a necessidade de encontrar funções de validação que garantam um agrupamento aceitável, sem a necessidade de utilizar a percepção humana.

Agrupamento com restrições: Aplicações do mundo real podem requerer agrupamentos sob vários tipos de restrições. A tarefa é encontrar grupos de dados que satisfaçam estas restrições.

Inteligibilidade e usabilidade: O usuário espera por resultados de agrupamento que sejam de apresentação visual legível e compreensível. Isto é, os grupos precisam estar associados a uma determinada interpretação e analisados de acordo com uma aplicação.

Neste trabalho, apenas alguns dos requisitos de desempenho foram avaliados. Por exemplo, a questão sobre a habilidade de se adaptar aos diferentes tipos de atributos, encontrar grupos com formas arbitrárias devido à natureza do problema tratado nesta dissertação. Outras não se aplicam ao algoritmo desenvolvido tal como a insensibilidade à ordem de apresentação dos dados. Para os testes conduzidos neste capítulo, foram avaliados os requisitos sobre a identificação do número e de conceitos, a atribuição de nomes, e inteligibilidade dos agrupamentos.

5.2 Preparação dos documentos

Como exposto na seção 2.3.1, os documentos possuem uma representação matemática que serve de modelo e é a forma como o software "compreende" os documentos. Por esse motivo é que se faz necessário um estágio de preparação para transformar o formato original dos documentos em um formato para processamento matemático.

Os documentos da base são constituídos de artigos no formato PDF coletados do site do IEEE ou do site `schoolar.google.com`. Para cada artigo PDF, a seção resumo foi extraída e armazenada em arquivos texto ASCII em um determinado diretório. Esses arquivos foram submetidos a um procedimento de análise léxica na qual foram identificados os termos. O analisador utilizou como delimitador de palavras espaços em branco, tabulações, pontuações e outros símbolos especiais. A seguir, os termos foram categorizados em Substantivo, Adjetivo, Advérbio ou Verbo, e foram reduzidos à sua forma primitiva. A categorização e a lematização fizeram uso do dicionário digital WordNet. Categorizados os termos, inicia-se o procedimento de filtragem que deverá remover termos que não agregam muito valor na descrição dos documentos. Os termos removidos são aqueles que pertencem à lista de palavras muito comuns, denominadas de *stop-word*, os termos não presentes no *WordNet* e os termos que não pertençam exclusivamente a classe de substantivos, adjetivos e substantivos/adjetivos.

Ainda neste estágio de preparação, os termos selecionados anteriormente serão utilizados para a construção da matriz de ocorrência de termos em documentos D_k conforme descrito na seção 2.3.1. Terminado o cálculo da matriz D_k, calcula-se o valor de entropia E de cada termo selecionado, conforme descrito na seção 4.4.5. A informação sobre a entropia de cada termo é utilizada para a realização de uma nova filtragem que objetiva a redução de ruídos causados por termos que ocorrem em poucos documentos e termos que ocorrem em muitos documentos. O resultado deste processo de

re-filtragem é uma lista de termos reduzida, o qual será o alvo do processo de agrupamento. Calcula-se novamente a matriz D_k e a matriz de correlação de termos $M = D_k^T D_k$.

Todas as informações obtidas nesta fase de preparação estão armazenadas em estruturas de dados que são armazenadas em um arquivo denominado de arquivo de sessão. Este arquivo, que nada mais é do que a serialização de objetos Java, será utilizado como fonte de dados pelo sistema de extração de conceitos. O arquivo de sessão também armazenará os conceitos extraídos pelo sistema.

5.3 Implementação do modelo

A realização dos testes foi possível por meio de um sistema que foi desenvolvido em linguagem Java e que implementa os algoritmos propostos no Capítulo 4. A escolha da linguagem foi motivada pelos recursos gráficos disponíveis e bibliotecas gratuitas para o desenvolvimento da aplicação. A figura 5.1 mostra a interface principal do programa.

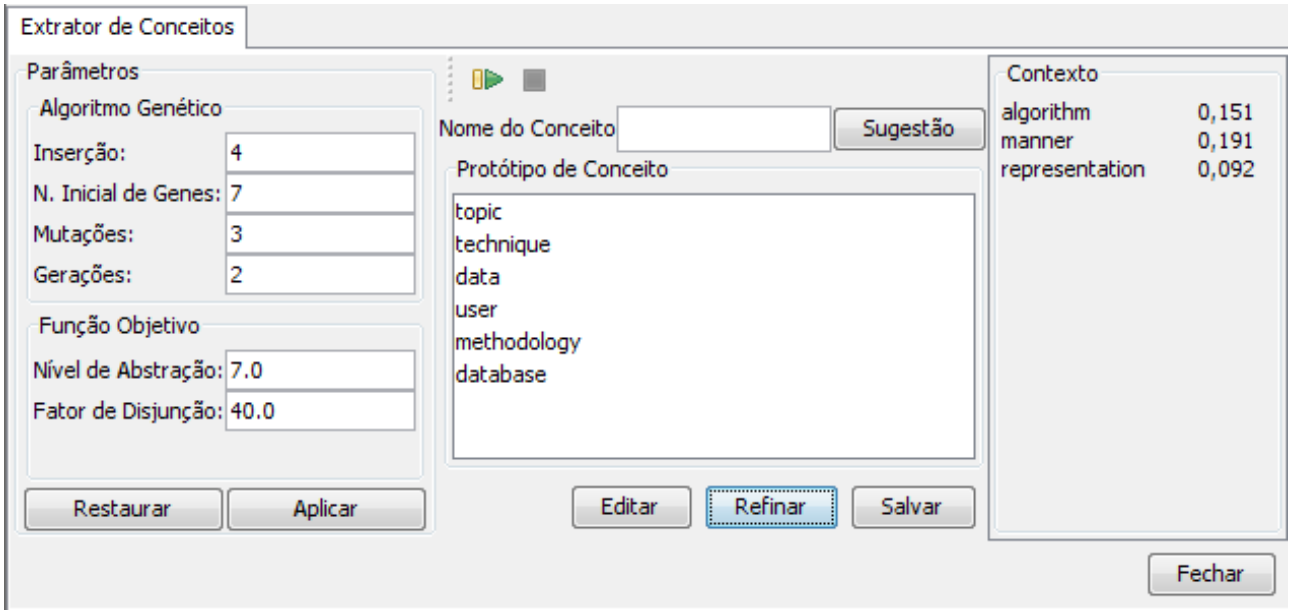


Fig. 5.1: Interface de extração de conceitos

O processo de extração inicia-se quando o usuário seleciona um arquivo de seção que contém todos os dados já preparados, ou seja, os termos já foram selecionados e as matrizes D_k e M já estão calculadas. A seguir, o usuário escolhe os parâmetros do algoritmo. Os parâmetros de configuração são descritos abaixo e mostrados na figura 5.2.

- *Inserção*: número de novos indivíduos inseridos a cada geração do algoritmo genético.
- *N. Inicial de Genes*: número de genes (termos) inicial em cada indivíduo.

- *Mutações*: número de mutações.
- *Gerações*: número de gerações do algoritmo genético.
- *Nível de Abstração*: parâmetro que regula o nível de abstração dos conceitos extraídos.
- *Fator de Disjunção*: parâmetro com influência na obtenção de novos conceitos.

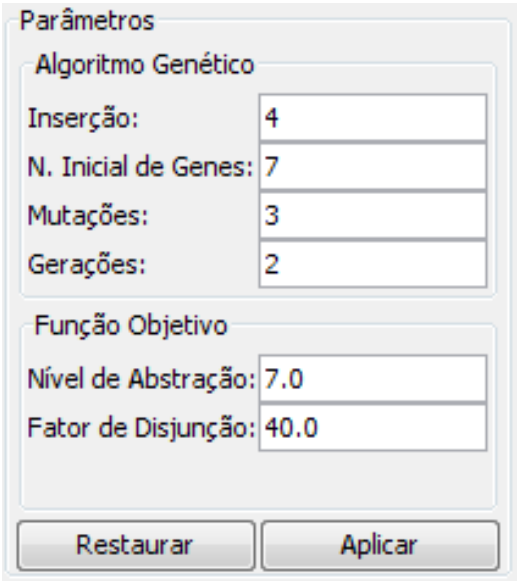


Fig. 5.2: Painel para configuração dos parâmetros do software

Definidos os parâmetros, o usuário requisita a identificação de um novo protótipo de conceito. O sistema identifica, refina e apresenta o protótipo ao usuário, conforme ilustrado na figura 5.3.

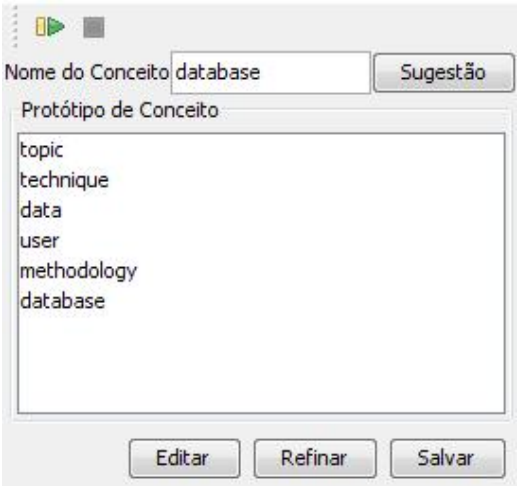


Fig. 5.3: Painel para visualização dos resultados e execução do algoritmo

Se o resultado não for “bom”, o usuário pode retornar à fase inicial, ajustar os parâmetros e re-executar o algoritmo. Na fase de aprovação, o usuário é auxiliado com a informação de sobreposição com outros conceitos, conforme mostra a figura 5.4. Esta figura mostra que o sistema já possui três conceitos (algorithm, manner, representation) e que a taxa de sobreposição entre o novo conceito e esses três são 0.151, 0.191 e 0.092, respectivamente.

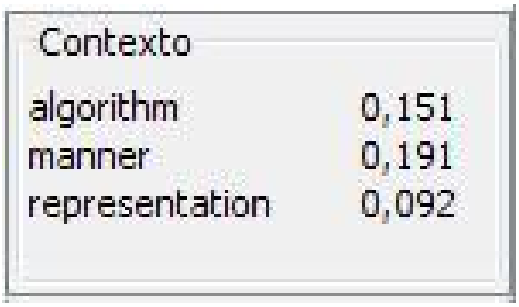


Fig. 5.4: Painel para visualização do contexto

Se o resultado for aprovado, o usuário deverá escolher um nome para este conceito. Novamente, o sistema pode auxiliar o usuário. O botão *Sugestão*, apresentado na figura 5.3, executa o algoritmo de Holger e apresenta a hierarquia de termos em uma janela separada. Automaticamente, o sistema sugere o termo mais relevante para o nome do conceito, como mostrado na figura 5.5. Entretanto, o usuário pode dar o nome que quiser utilizando a hierarquia de termos apenas como referência.

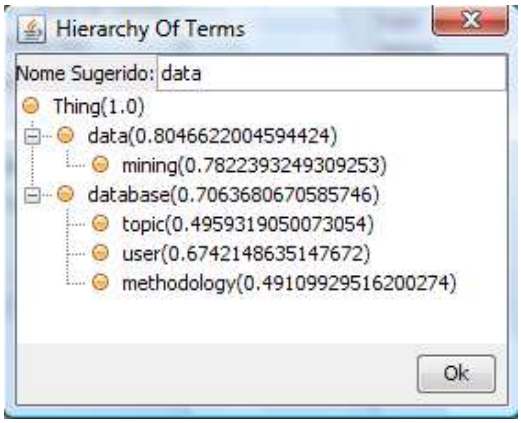


Fig. 5.5: Sugestão de nome

Por fim, o usuário armazena o novo protótipo de conceito, o sistema calcula as associações fuzzy e retorna ao início, onde o usuário pode extrair um novo conceito ou terminar o algoritmo.

5.4 Testes do modelo

Esta seção apresenta três conjuntos de resultados. O primeiro destina-se a avaliar a habilidade do algoritmo em encontrar um número adequado de conceitos e de agrupar corretamente termos que sejam de um determinado assunto, ou seja, serão avaliados os requisitos de número 1 e 2. O segundo conjunto visa comparar os resultados obtidos com outras técnicas de extração de conceitos. Por fim, o terceiro conjunto tem por objetivo avaliar o terceiro requisito, o mecanismo de atribuição de nomes aos conceitos.

5.4.1 Avaliação dos parâmetros

Esta seção avalia os requisitos 1 e 2 frente a influência dos parâmetros de configuração do algoritmo. Os parâmetros avaliados são o η, para controle do nível de abstração, e o número L de termos da base. Os parâmetros de entrada τ e γ foram mantidos fixos, conforme descrito a seguir.

O valor do parâmetro τ é escolhido de forma a se acentuar a necessidade de se obter conceitos mais disjuntos possíveis dos já extraídos. Devido à natureza iterativa do algoritmo ACT, é bastante fácil decidir qual valor escolher para este parâmetro. Deve-se escolher valores suficientemente grandes para garantir a obtenção de conceitos diferentes dos já existentes. Para os testes apresentados nesta seção, o valor 40 para τ trouxe bons resultados.

O valor de γ foi mantido em 0,5 para todos os testes, indicando que não estamos interessados em conceitos que possuem mais de 50% de seu conteúdo descrito em outro conceito.

Para cada base de documentos, foram realizados quatro diferentes testes, combinando dois valores distintos para cada parâmetro. Foi avaliada a influência desses parâmetros para a determinação do número ótimo de conceitos e os assuntos identificados pelo algoritmo.

Os resultados são apresentados de forma tabular, onde as linhas das tabelas representam os conceitos extraídos pelo algoritmo e as colunas mostram as características de cada conceito. A ordem de extração dos conceitos foi mantida nas linhas das tabelas. As características apresentadas nas colunas são: ordem de extração do conceito, o nome atribuído pelo algoritmo de Holger, a taxa máxima de sobreposição, o assunto descrito pelo conceito (quando não foi possível a identificação um * foi atribuído) e os termos significativos do conceito.

Base de Documentos 1

Base de documentos cujo conteúdo foi coletado de forma muito criteriosa. O conjunto de documentos é constituído pelos resumos de 108 artigos coletados da biblioteca digital do IEEE na área de inteligência computacional. Mais precisamente, foram coletados entre 8 e 12 artigos de dez assuntos diferentes: Cognition, Fuzzy Systems, Genetic Algorithm, Neural Networks, Symbolic Logic, Data Mining, Knowledge Management, Machine Learning, Optimization, Pattern Recognition.

Cenário 1

Para o cenário 1, foram utilizados 154 termos ($L = 154$) e o parâmetro η foi mantido em 7. A tabela 5.1 mostra que o algoritmo identificou 11 conceitos e em 7 deles foi possível identificar o assunto tratado. Somente em três situações o nome escolhido é consistente com o assunto tratado: **neuron** para o assunto Neural Networks, **management** para Knowledge Management e **algorithm** para Genetic Algoritm.

#	*Nome*	γ	*Assunto*	*Termos Relevantes*
1	aspect	0.00	Machine Learning	computer, aspect, behavior, intelligence, paradigm, years
2	benchmark	0.04	*	exploration, benchmark, architecture, variation, characteristic, interation, fact
3	neuron	0.08	Neural Networks	extension, importance, neuron, goal, stability, property, choice
4	management	0.12	Knowledge Management	storage, knowledge, capability, management, path, business
5	protocol	0.11	*	expert, protocol, difference, relations, role, individual
6	manipulation	0.12	Cognition	representation, manipulation, theory, life, language, cognition
7	elements	0.19	Pattern Recognition	conclusion, input, classification, region, elements, application
8	algorithm	0.20	Genetic Algorithm	population, fitness, optimum, member, algorithm, convergence, solution
9	variable	0.40	*	extraction, example, relations, variable, analysis, satisfaction
10	measurement	0.39	*	importance, performance, definition, statistics, algorithm, measurement
11	attention	0.19	Data Mining	attention, data, concept, generalization, addition, relationship

Tab. 5.1: Conceitos para a primeira base de documentos ($\eta = 7$, $L = 154$).

Cenário 2

Neste cenário com $L = 77$ e $\eta = 7$, o algoritmo identificou dez conceitos como mostrado na tabela 5.2. Assim como no caso anterior, sete conceitos representam assuntos discutidos pelos documentos. Somente em quatro situações o nome aplicado é significativo: **algorithm** para Genetic Algorithm, **database** para Data Mining, **knowledge** para Knowledge Management e **concept** para Symbolic Logic.

#	*Nome*	γ	*Assunto*	*Termos Relevantes*
1	algorithm	0.00	Genetic Algorithm	performance, convergence, problem, solution, operator, algorithm, population
2	representation	0.11	Cognition	principle, life, representation, theory, language, cognition
3	manner	0.16	Neural Networks	coefficient, vector, manner, example, analysis, synthesis
4	database	0.19	Data Mining	topic, technique, data, user, methodology, database
5	computer	0.12	*	computer, stability, propagation, method, years
6	simulation	0.35	*	environment, effectiveness, problem, architecture, characteristic, simulation
7	knowledge	0.19	Knowledge Management	expert, acquisition, knowledge, component, management
8	variable	0.24	Pattern Recognition	input, system, one, prediction, variable, accuracy
9	constraint	0.39	*	relationship, category, data, constraint, addition
10	concept	0.42	Symbolic Logic	concept, nature, one, theory, method, logic

Tab. 5.2: Conceitos para a primeira base de documentos ($\eta = 7$, $L = 77$).

Cenário 3

O cenário 3 mostra a influência do parâmetro η na obtenção dos conceitos. Pode-se verificar que valores elevados para η favorecem o surgimento de grupos mais abstratos abrangendo mais de um assunto. Os conceitos de número 2 e 3 mostram que são conceitos que tratam de assuntos Logic Symbolic/Cognition e Neural Networks/ Pattern Recognition, respectivamente. De uma certa forma, este resultado é consistente com a teoria desenvolvida na seção 4.4.7 sobre taxonomia de conceitos.

#	*Nome*	γ	*Assunto*	*Termos Relevantes*
1	algorithm	0.00	Genetic Algorithm Optimization	performance, mechanism, selection, operator, algorithm, operation, convergence, solution, problem, population
2	theory	0.26	Symbolic Logic Knowledge Management Cognition	one, representation, development, concept, life, knowledge, theory, language, logic, principle, idea, cognition, method
3	manner	0.44	Neural Networks Pattern Recognition	input, example, technique, variable, analysis, prediction, synthesis, vector, problem, system, coefficient, manner
4	characteristic	0.40	*	performance, characteristic, complexity, simulation, analysis, relationship, fact, architecture, information, data

Tab. 5.3: Conceitos para a primeira base de documentos ($\eta = 20$, $L = 77$).

Cenário 4

No cenário 4 com $L = 154$ e $\eta = 20$, assim como no cenário 3 observou-se a ocorrência de conceitos mais abstratos que discutiam mais de um assunto.

#	*Nome*	γ	*Assunto*	*Termos Relevantes*
1	algorithm	0.00	Genetic Algorithm/Optimization	exploration, performance, fitness, operator, member, algorithm, convergence, solution, importance, problem, population, optimum, method, crossover
2	vector	0.20	Neural Networks	extension, input, example, property, regression, analysis, neuron, procedure, realization, synthesis, vector, coefficient, manner, applicability
3	management	0.20	Data Mining/Knowledge Management	user, technique, topic, storage, knowledge, management, capability, information, methodology, data, path, business, database, application
4	aspect	0.14	Cognition/Logic Symbolic	representation, behavior, theory, life, paradigm, language, computer, principle, aspect, manipulation, intelligence, cognition, years
5	protocol	0.33	Cognition	protocol, difference, technique, relations, complexity, individual, analysis, expert, problem, role, system, cognition, method, application

Tab. 5.4: Conceitos para a primeira base de documentos ($\eta = 20$, $L = 154$).

Nesta base de documentos, o algoritmo mostrou grande insensibilidade ao número de termos L. Ao se confrontar os cenários 1 e 2, a diferença entre o número de conceitos obtidos foi de apenas 1;

o mesmo ocorreu ao se confrontar os cenários 3 e 4. Já o parâmetro η influencia de maneira determinante o número de conceitos extraídos. No entanto, este é o comportamento desejado. Como foi discutido na seção 4.4.7, um maior valor de η implica na obtenção de conceitos mais abstratos, ou seja, conceitos que podem tratar de diversos assuntos. Esta propriedade foi observada nos cenários 3 e 4, quando identificaram-se conceitos de assuntos como Algoritmos Genéticos/Otimização e também Symbolic Logic/Cognition. Observa-se também que em todos os cenários o algoritmo produziu grupos consistentes, ou seja, grupos com termos que estão conceitualmente próximos.

Base de Documentos 2

A segunda base de documentos foi constituída de 139 artigos utilizados para a elaboração desta dissertação. Os assuntos abordados por este conjunto de documentos podemos citar: Clustering tecniques, BiClustering techniques, Ontologies, Latent Semantic Indexing, Information Retrieval, Ontology Extraction, Fuzzy systems, Semantic Web.

Cenário 1

A tabela 5.5 apresenta os conceitos extraídos para o cenário com parâmetros $L = 154$ e $\eta = 7$. Neste, o algoritmo extraiu 12 conceitos dos quais é possível a identificação de 6 assuntos. Em todos, o nome sugerido não foi condizente com o assunto tratado.

#	*Nome*	γ	*Assunto*	*Termos Relevantes*
1	library	0.00	Semantic	domain, description, entity, library, semantics, version
2	detection	0.06	LSI	matrix, basis, combination, vector, detection, association, decomposition
3	principle	0.09	Clustering	finding, principle, difficulty, algorithm, tendency, membership
4	subset	0.15	Bi-Clustering	gene, idea, data, contribution, subset, analysis, variety
5	extension	0.12	Information Retrieval	extension, logic, retrieval, interpretation, evaluation, area
6	creation	0.34	*	creation, ontology, mechanism, context, domain
7	commitment	0.12	Taxonomy	commitment, hierarchy, discussion, taxonomy, resource, insight
8	period	0.16	*	period, information, minimum, property, construction
9	input	0.30	*	input, combination, degree, correlation, recognition, module
10	computation	0.33	*	computation, classification, generation, standard, algorithm, method
11	selection	0.18	*	text, baseline, strategy, selection, explanation, addition
12	methodology	0.38	*	technology, ontology, methodology, evolution, management, language

Tab. 5.5: Conceitos para a segunda base de documentos ($\eta = 7$, $L = 154$).

Cenários 2 e 3

As tabelas 5.6 e 5.7 apresentam os resultados para os cenários com parâmetros $L = 77$,$\eta = 7$ e parâmetros $L = 77$,$\eta = 20$, respectivamente. Em ambos os casos, o algoritmo trouxe um número adequado de conceitos e foi possível a identificação da maioria dos assuntos. Assim como nos cenários anteriores, a seleção automática de nomes não trouxe nomes significativos para os conceitos.

#	*Nome*	γ	*Assunto*	*Termos Relevantes*
1	application	0.00	LSI	finding, discovery, example, basis, application
2	input	0.12	Bi-Clustering	input, combination, degree, correlation, recognition
3	language	0.11	Ontology	technology, ontology, development, evolution, management, language
4	algorithm	0.19	Clustering	data, problem, algorithm, objective, kernel, method
5	environment	0.38	Ontology Extraction	extraction, ontology, domain, knowledge, environment, resource
6	kind	0.37	Information Retrieval	retrieval, user, information, kind, relationship, method
7	selection	0.33	*	text, ontology, concept, selection, addition

Tab. 5.6: Conceitos para a segunda base de documentos ($\eta = 7$, $L = 77$).

#	*Nome*	γ	*Assunto*	*Termos Relevantes*
1	algorithm	0.00	Clustering	basis, example, constraint, algorithm, objective, finding, vector, data, membership, kernel, aplication
2	knowledge	0.14	Ontology Extraction	extraction, component, development, evolution, knowledge, management, environment, resource, language, technology, text, ontology, domain,
3	decomposition	0.43	LSI	basis, combination, relations, association, decomposition, retrieval, matrix, direction, vector, collection
4	data	0.38	*	classification, concept, selection, correlation, algorithm, analysis, addition, text, data, recognition
5	information	0.43	Information Retrieval	representation, user, evaluation, technique, concept, relationship, retrieval, text, goal, ontology, information
6	data	0.44	Bi-Clustering	discovery, gene, knowledge, expression, analysis, variety, direction, attention, idea, data, method, application, ways

Tab. 5.7: Conceitos para a segunda base de documentos ($\eta = 20$, $L = 77$).

Cenário 4

O cenário 4, com $L = 154$ e $\eta = 20$, é a situação que trouxe os melhores resultados. Como pode ser comprovado pela tabela 5.8, foi possível a identificação de 7 dos 8 conceitos extraídos.

#	*Nome*	γ	*Assunto*	*Termos Relevantes*
1	subset	0.00	Bi-Clustering	discovery, gene, contribution, subset, cancer, analysis, expression, variety, direction, attention, idea, data, method, application
2	detection	0.31	LSI	item, user, basis, subspace, combination, detection, decomposition, association, retrieval, matrix, effectiveness, vector, collection, method
3	library	0.13	Semantic	development, evolution, entity, library, management, language, version, technology, ontology, methodology, domain, description, semantics
4	tendency	0.28	Clustering	example, prototype, constraint, tendency, algorithm, objective, possibility, finding, principle, data, problem, difficulty, membership
5	selection	0.18	*	extension, representation, evaluation, concept, strategy, selection, explanation, addition, logic, interpretation, text, baseline, area
6	identification	0.23	Ontology	input, mechanism, classifier, correlation, thesaurus, creation, ontology, context, integration, identification, recognition, source, module
7	resource	0.30	Taxonomy	technique, discussion, knowledge, environment, resource, commitment, hierarchy, text, ontology, domain, taxonomy, insight, application
8	kind	0.50	Information Retrieval	period, user, minimum, kind, property, relations, decomposition, retrieval, information, expansion, criterion, method, construction

Tab. 5.8: Conceitos para a segunda base de documentos ($\eta = 20$, $L = 154$).

Diferentemente do ocorrido com a base de documentos anterior, neste caso o parâmetro L influenciou o número de conceitos extraídos. No entanto, o algoritmo conseguiu agrupar termos semanticamente próximos e, na maioria dos casos, foi possível identificar o assunto tratado pelo conceito.

Base de Documentos 3

A terceira base de documentos foi constituída de 231 artigos discutindo somente algoritmos genéticos. Para esta configuração de base de documentos, o algoritmo identificou apenas um conceito na maioria dos cenários. Para o cenário onde $L = 77$ e o cenário onde $L = 154$ e $\eta = 20$, o algoritmo identificou apenas um conceito e atribuiu corretamente o nome *algorithm* ao conceito. Nos três casos, o algoritmo ACT agrupou termos referentes ao assunto Algoritmo Genético, tais como: *algorithm*, *application*, *performance*, *problem* e *solution*.

Para o cenário com parâmetros $L = 154$ e $\eta = 7$, apresentado na tabela 5.9, o algoritmo identificou quatro conceitos sobre o mesmo assunto.

Cenário 1

- Número de termos: 154;
- Constante η: 7;

#	*Nome*	γ	*Assunto*	*Termos Relevantes*
1	algorithm	0.00	Genetic Algorithm	conclusion, performance, problem, category, experimentation, algorithm, application, solution
2	elements	0.47	Genetic Algorithm	element, mathematics, category, location, complexity, algorithm, elements
3	phenotype	0.29	Genetic Algorithm	representation, computation, example, phenotype, evolution, genotype
4	manner	0.36	Genetic Algorithm	basis, description, problem, operator, relation, manner, analysis

Tab. 5.9: Conceitos para a terceira base de documentos ($\eta = 7$, $L = 154$).

O caso em que a base de documentos discute somente um assunto, observa-se que o algoritmo identificou um único conceito em 3 dos 4 cenários. A única exceção ocorreu no cenário 1 onde surgiram quatro conceitos discutindo outros aspectos da área de algoritmos genéticos.

Base de Documentos 4

A quarta base de documentos foi constituída de 200 artigos coletados aleatoriamente do site `schoolar.google.com`. O objetivo deste teste é avaliar o comportamento do algoritmo ACT quando manipula uma base de documentos onde não há muita correlação entre os termos. Os resultados mostram que o algoritmo torna-se bastante sensível ao número de termos. Para o caso onde $\eta = 7$ observa-se uma diferença de seis conceitos entre os cenários para o qual $L = 77$, tabela 5.11, e $L = 154$, tabela 5.10. Embora o nome atribuído ao conceito esteja presente nas tabelas, esta informação é irrelevante para a avaliação do algoritmo, pois não se sabe previamente os conceitos abordados pelos documentos.

#	*Nome*	γ	*Termos Relevantes*
1	money	0.00	citizen, payment, expectation, money, economy, dollar
2	failure	0.02	modification, failure, input, improvement, correlation
3	taxon	0.02	specimen, asia, genus, africa, taxon
4	commerce	0.12	agency, recommendation, efficiency, corporation, commerce, responsability
5	emphasis	0.06	manuscript, emphasis, era, education, media
6	substance	0.10	substance, examination, things, memory, depth
7	hypothesis	0.12	february, assistance, quantity, hupothesis, outcome, competition
8	mixture	0.35	modification, preparation, interval, mixture, manner
9	story	0.06	story, domain, faculty, abundance
10	transition	0.10	ratio, taylor, possibility, transition, nation
11	message	0.14	expression, argument, phenomenon, message, assumption, suggestion
12	framework	0.16	detection, circunstances, interpretation, synthesis, framework
13	implication	0.20	implication, consideration, explanation, determination, policy
14	percent	0.16	indicator, percent, manager, street, reproduction
15	hall	0.31	hall, specimen, ecosystem, survival, agent
16	magnitude	0.13	dynamics, precipitation, magnitude, agreement, resolution, formation
17	architecture	0.15	software, implementation, security, architecture, interface
18	carbon	0.31	chemistry, carbon, strengh, suggestion, user

Tab. 5.10: Conceitos para a quarta base de documentos ($\eta = 7$, $L = 154$).

#	*Nome*	γ	*Termos Relevantes*
1	dynamics	0.00	dynamics, appendix, resolution, reproduction, climate
2	policy	0.06	topic, implication, money, economy, policy
3	mixture	0.09	substance, interval, misture, manner, culture
4	implementation	0.06	input, implementation, efficiency, interface, user
5	asia	0.08	australia, asia, genus, africa, city
6	argument	0.34	argument, phenomenon, explanation, economy, things
7	construction	0.14	percent, assistence, street, hypothesis, construction
8	cambridge	0.30	consumption, recommendation , manner, efficiency, cambridge
9	city	0.49	asia, director, council, depth, city
10	consideration	0.32	implication, consideration, explanation, assumption, determination
11	limitation	0.49	implementation, stability, limitation, interface, methodology
12	examination	0.33	substance, organization, examination, decade, disease

Tab. 5.11: Conceitos para a quarta base de documentos ($\eta = 7$, $L = 77$).

Para o cenário onde $\eta = 20$, o algoritmo ACT também mostrou grande sensibilidade ao número de termos presentes para agrupamento. A tabela 5.12 e a tabela 5.13 mostram a diferença na quantidade de conceitos obtidos quando utiliza-se $L = 77$ e $L = 154$, respectivamente. Observa-se uma variação de cinco conceitos de um cenário para o outro e, assim como anteriormente, a atribuição de nomes aos conceitos não apresenta nenhum significado, pois não se sabe dos assuntos tratados.

#	*Nome*	γ	*Termos Relevantes*
1	argument	0.00	argument, law, topic, phenomenon, implication, explanation, money, policy, council, outcome
2	dynamics	0.10	chemistry, dynamics, implementation, transition, stability, appendix, resolution, limitation, interface, reproduction, climate, methodology
3	city	0.19	percent, assistence, street, australia, asia, appendix, genus, hypothesis, africa, construction, depth, city
4	mixture	0.19	substance, examination, parameter, interval, ratio, consumption, mixture, recommendation, manner, efficiency, cambridge, culture

Tab. 5.12: Conceitos para a quarta base de documentos ($\eta = 20$, $L = 77$).

#	*Nome*	γ	*Termos Relevantes*
1	payment	0.00	law, enforcement, citizen, payment, expectation, economy, agency, security, corporation, commerce, money, dollar, responsability
2	failure	0.03	modification, failure, input, preparation, interval, improvement, mixture, ecosystem, correlation, climate, culture, organism
3	hypothesis	0.14	implication, february, relations, assistence, stability, perspective, quantity, limitation, hypothesis, simulation, outcome, competition
4	suggestion	0.15	expression, argument, phenomenon, message, circumstances, explanation, assumption, suggestion, interpretation
5	africa	0.24	duration, diameter, february, specimen, australia, entry, asia, genus, africa, taxon, city
6	magnitude	0.32	knowledge, dynamics, precipitation, magnitude, stability, reproduction, simulation, climate, formation
7	nation	0.23	committee, recognition, tradition, colleague, men, director, advice, memory, nation
8	methodology	0.19	consumption, protein, domain, emphasis, quantity, eficiency, interface, extraction, methodology
9	hospital	0.31	statement, organization, hall, committee, ecosystem, hypothesis, director, cambridge, hospital

Tab. 5.13: Conceitos para a quarta base de documentos ($\eta = 20$, $L = 154$).

5.4.2 Comparação com Métodos via Decomposição de Matrizes

Nesta seção, compara-se o método ACT com outras técnicas descritas anteriormente. As técnicas escolhidas para comparação são Latent Semantic Indexing (LSI) e Non-Negative Matrix Factorization (NMF).

Duas configurações foram selecionadas para a comparação entre os métodos. A primeira configuração é dada pela base de documentos 1, cenário 1 ($\eta = 7$ e $L = 77$); a segunda configuração é dada pela base de documentos 2, cenário 4 ($\eta = 20$ e $L = 154$). A comparação completa entre os métodos está descrita no apêndice A.

Antes de iniciar a comparação entre os métodos é necessário normalizar os resultados obtidos pelas diferentes técnicas. A normalização garante que a comparação entre os conceitos não será influenciada por diferenças de norma ou desvios no valor médio das pertinências.

Seja $\mu_A = \{\mu_i\}_A$ o conjunto fuzzy associado ao conceito A e μ_i a pertinência do termo i ao conceito A. Devemos primeiramente subtrair a média das pertinências, dado por $\bar{\mu_A} = 1/L \sum_{j=1}^{L} \mu_i$, de cada valor individual de pertinência. Assim:

$$\mu_A^* = \{\mu_i\}_{A^*} = \mu_A - \bar{\mu_A} = \{\mu_i\}_A - \bar{\mu_A} = \{\mu_i - \bar{\mu}_A\}_A \tag{5.1}$$

Em seguida, devemos normalizar o valor das pertinências. Assim:

$$\mu_A^N = \frac{\mu_A^*}{\sqrt{\sum_{j=1}^{L} \mu_i^2}} = \left\{ \frac{\mu_i}{\sqrt{\sum_{j=1}^{L} \mu_i^2}} \right\}_{A^*} \tag{5.2}$$

Para cada cenário selecionado, duas tabelas são construídas para mostrar a semelhança entre os conceitos. As linhas dessas matrizes representam os conceitos extraídos com a técnica ACT e as colunas representam os conceitos obtidos com as técnicas de decomposição de matrizes. O conteúdo de cada célula da matriz é o índice de semelhança entre os conceitos cujo valor está compreendido entre 0 e 1. Valores próximos de 0 indicam pouca sobreposição entre os conceito e valores próximos de 1 indicam muita sobreposição entre os conceitos.

Para facilitar a comparação entre as técnicas, células que possuem os maiores valores de semelhança em cada linha ou em cada coluna foram demarcadas em preto. Este procedimento torna mais fácil a identificação de conceitos sobrepostos pois nesta situação a ocorrência de linhas e colunas com mais de uma marcação denunciará a sobreposição entre os conceitos. No cenário ideal de igualdade entre resultados nenhuma sobreposição de conceitos é constatada caracterizando um mapeamento um a um entre os conceitos obtidos.

As tabelas 5.14 e 5.15 mostram os resultados da comparação da técnica ACT com as técnicas NMF e LSI, respectivamente, ao utilizar-se a primeira base de documentos com o cenário 1, $L = 77$

e $\eta = 7$. A tabela comparativa mostra que os resultados obtidos com a técnica ACT traz resultados muito semelhantes aos resultados obtidos com NMF. Ao contar-se o número de linhas ou colunas com mais de uma célula destacada nota-se que a LSI possui mais casos de sobreposição entre conceitos. Outra forma de medir o índice de sobreposição é avaliar a diferença entre os dois maiores valores por linha e por coluna. Novamente, o que se observa é uma semelhança maior entre as técnicas ACT e NMF.

		NMF									
		1	2	3	4	5	6	7	8	9	10
ACT	knowledge	0,39	0,53	0,50	0,49	0,45	0,42	0,92	0,42	0,45	0,51
	algorithm	0,96	0,41	0,55	0,43	0,45	0,56	0,38	0,44	0,46	0,39
	variable	0,46	0,43	0,40	0,44	0,60	0,42	0,50	0,62	0,89	0,42
	representation	0,42	0,48	0,46	0,40	0,42	0,42	0,46	0,59	0,41	0,96
	simulation	0,58	0,41	0,58	0,47	0,44	0,92	0,41	0,43	0,45	0,39
	concept	0,46	0,48	0,45	0,43	0,46	0,41	0,41	0,92	0,51	0,61
	computer	0,50	0,80	0,44	0,44	0,48	0,44	0,50	0,62	0,46	0,41
	manner	0,45	0,50	0,43	0,48	0,96	0,44	0,48	0,45	0,55	0,43
	constraint	0,43	0,44	0,77	0,67	0,41	0,58	0,37	0,45	0,47	0,44
	database	0,46	0,47	0,49	0,96	0,52	0,42	0,48	0,46	0,46	0,40

Tab. 5.14: Comparação com NMF ($\eta = 7$, $L = 77$).

		LSI									
		1	2	3	4	5	6	7	8	9	10
ACT	simulation	0,49	0,67	0,41	0,89	0,44	0,42	0,36	0,38	0,41	0,45
	algorithm	0,48	0,75	0,50	0,50	0,50	0,44	0,33	0,36	0,45	0,54
	variable	0,57	0,62	0,56	0,38	0,39	0,65	0,43	0,62	0,40	0,38
	manner	0,48	0,60	0,49	0,41	0,49	0,46	0,36	0,93	0,42	0,77
	representation	0,48	0,47	0,61	0,39	0,47	0,40	0,94	0,40	0,40	0,52
	concept	0,75	0,61	0,40	0,36	0,47	0,53	0,70	0,43	0,43	0,40
	computer	0,57	0,44	0,41	0,39	0,70	0,61	0,43	0,44	0,41	0,44
	database	0,61	0,66	0,40	0,53	0,43	0,36	0,41	0,67	0,84	0,45
	knowledge	0,39	0,46	0,71	0,44	0,55	0,63	0,61	0,53	0,63	0,39
	constraint	0,71	0,57	0,48	0,73	0,39	0,51	0,48	0,46	0,70	0,56

Tab. 5.15: Comparação com LSI ($\eta = 7$, $L = 77$).

As tabelas 5.16 e 5.17 mostram os resultados da comparação da técnica ACT com as técnicas NMF e LSI, respectivamente, ao utilizar-se a segunda base de documentos com o cenário 4, $L = 154$ e

$\eta = 20$. Assim como no caso anterior, verifica-se que os resultados trazidos pela técnica ACT e NMF são semelhantes. Percebe-se um mapeamento um-a-um dos conceitos obtidos com as duas técnicas. A comparação com a técnica LSI, diferentemente do caso NMF, mostra que existe sobreposições entre conceitos; tome como exemplo os conceito de número 2 e 3 da técnica LSI que são mapeados a um único conceito ACT, *tendency*, ou, ainda, os conceitos 6 e 8 que são mapeados para o conceito *resource*. É possível ainda observar sobreposições de conceito analisando-se os conceitos *kind* e *detection* que foram mapeados para o conceito de número 7.

		NMF							
		1	2	3	4	5	6	7	8
ACT	selection	0,46	0,41	0,71	0,48	0,50	0,46	0,62	0,40
	tendency	0,44	0,84	0,44	0,58	0,49	0,45	0,33	0,60
	kind	0,83	0,45	0,36	0,43	0,46	0,44	0,56	0,62
	identification	0,38	0,41	0,64	0,49	0,40	0,47	0,73	0,43
	detection	0,65	0,43	0,38	0,47	0,43	0,42	0,45	0,88
	library	0,42	0,42	0,52	0,41	0,47	0,79	0,72	0,37
	resource	0,48	0,42	0,51	0,43	0,85	0,53	0,58	0,39
	subset	0,56	0,48	0,49	0,92	0,44	0,40	0,39	0,52

Tab. 5.16: Comparação com NMF ($\eta = 20$, $L = 154$).

		LSI							
		1	2	3	4	5	6	7	8
ACT	library	0,42	0,52	0,54	0,46	0,52	0,53	0,37	0,86
	identification	0,57	0,54	0,56	0,55	0,76	0,51	0,40	0,68
	tendency	0,55	0,58	0,58	0,43	0,50	0,56	0,36	0,28
	selection	0,48	0,44	0,58	0,74	0,38	0,48	0,53	0,60
	kind	0,36	0,49	0,66	0,37	0,41	0,50	0,78	0,47
	detection	0,40	0,48	0,65	0,38	0,53	0,38	0,86	0,34
	subset	0,85	0,45	0,73	0,44	0,50	0,36	0,58	0,34
	resource	0,38	0,36	0,65	0,62	0,45	0,68	0,40	0,69

Tab. 5.17: Comparação com LSI ($\eta = 20$, $L = 154$).

As tabelas apresentadas aqui e, também, as do apêndice A mostram que a técnica ACT traz resultados semelhantes à técnica NMF.

5.4.3 Análise dos nomes dos conceitos

Nesta última parte de análise dos resultados o objetivo é avaliar o nome sugerido para os conceitos extraídos. A análise será conduzida somente sobre as bases de documentos 1 e 2. Para a base de documentos 3 o algoritmo aplicou corretamente o nome *algorithm* para a maioria dos conceitos e para a base de documentos 4 não é possível avaliar o nome aplicado, pois não há conhecimento prévio sobre os assuntos abordados.

A observação dos nomes aplicados aos conceitos revela que raramente o nome sugerido condiz com o assunto tratado pelo conceito. Tomando como exemplo o primeiro cenário da base de documentos 1 observa-se que dos sete conceitos do qual pode-se identificar o assunto apenas dois possuem um nome condizente. A tabela 5.18 mostra os conceitos e seus respectivos nomes.

Nome Sugerido	*Assunto*	*Condizente*
algorithm	Genetic Algorithm	sim
representation	Cognition	não
manner	Neural Networks	não
database	Data Mining	não
computer		
simulation		
knowledge	Knowledge Management	sim
variable	Pattern Recognition	não
constraint		
concept	Simbolic Logic	não

Tab. 5.18: Sugestão de nomes aos conceitos da base de documentos 1

Entretanto, verificou-se que a escolha mais cuidadosa dos termos utilizados para o agrupamento auxilia de forma significativa a aplicação de nomes aos conceitos. A seleção de termos baseada na categoria gramatical e entropia do termo deixou de lado termos que são importantes para a representação dos documentos. E, ainda, alguns termos foram excluídos por não estarem presentes na WordNet. Dentre esses termos excluídos podemos citar: *neural*, *mining*, *pattern*, *network*, *fuzzy*, *genetic* e *symbolic*.

A inclusão de termos importantes no processo de clusterização modificou radicalmente esse cenário, produzindo conceitos mais consistentes e com nomes mais condizentes. A tabela 5.19 mostra que, dos sete conceitos obtidos, seis possuem um nome condizente com o assunto abordado e em muitos casos o nome correto para o conceito.

Nome Simples	*Nome Composto*	*Assunto*	*Condizente*
algorithm	genetic algorithm	Genetic Algorithm	sim
logic	symbolic logic	Symbolic Logic	sim
network	neural network	Neural Networks	sim
data	mining data	Data Mining	sim
fact	architecture fact	Desconhecido	não
machine	learning machine	Machine Learning	sim
knowledge	component knowledge	Knowledge Management	sim

Tab. 5.19: Nomes mais condizentes para os conceitos da base de documentos 1

Para a base de documentos 2, o cenário foi o mesmo do anterior. O algoritmo não foi capaz de atribuir um único nome condizente com o assunto tratado, veja tabela 5.20.

Nome Sugerido	*Assunto*	*Condizente*
subset	Bi-Clustering	não
detection	LSI	não
library	Semantic	não
tendency	Clustering	não
selection	Desconhecido	não
identification	Ontology	não
resource	Taxonomy	não
kind	Information Retrieval	não

Tab. 5.20: Sugestão de nomes aos conceitos da base de documentos 2

Novamente, a inserção manual de termos que foram originalmente filtrados trouxe resultados melhores para o algoritmo de atribuição de nomes, veja tabela 5.21.

Nome Simples	*Nome Composto*	*Assunto*	*Condizente*
indexing	latent indexing	Latent Semantic Indexing	sim
algorithm	cluster algorithm	Clustering	sim
paradigm	semantic paradigm	Semantic Web	não
ontology	identification ontology	Ontology	sim
microarray	subset microarray	Bi-Clustering	não
information	system information	Information Retrieval	sim

Tab. 5.21: Nomes mais condizentes para os conceitos da base de documentos 2

Observa-se assim que a boa seleção de termos é fundamental para a obtenção de nomes significativos, e conclui-se que o procedimento automático para a seleção de termos precisa ser revisto e possivelmente necessite de intervenção humana. Uma possibilidade para resolver o problema da boa seleção de termos é modificar a representação interna do documento de forma a levar em consideração a seção *palavras-chave* dos artigos. Essa seção normalmente contém os termos mais adequados para descrever os assuntos presentes nos documentos. Em relação ao problema da WordNet em não reconhecer palavras que não estejam registradas, poder-se-ía utilizar métodos de extração automática de termos, tal como proposto por Nakagawa *et al.*[57].

5.5 Considerações Finais

Neste capítulo, foram apresentados os resultados da aplicação da técnica de agrupamento de termos correlatos para a extração de conceitos. Os resultados, que foram organizados em três seções, mostram que a técnica traz resultados semelhantes à técnica NMF e ainda introduz meios de supervisionar o processo de extração, permitindo que o operador possa ajustar os parâmetros do algoritmo à medida que este é executado. Verificou-se, também, que a atribuição automática de nomes a conceitos somente é efetiva em casos em que é feita uma pré-seleção dos termos mais importantes para a descrição da base.

Capítulo 6

Conclusões e Perspectivas

As estratégias modernas para recuperação de informação fazem uso de estruturas ontológicas para dar maior fundamentação semântica aos textos. No entanto, essas estruturas precisam ser criadas manualmente por pessoas experientes nos assuntos abordados pelos textos. Conseqüentemente, esse é um trabalho que requer muito tempo e atenção daqueles envolvidos no processo de criação das ontologias. Pensando em resolver esse problema é que surgiram propostas utilizando aprendizado de máquina e processamento de linguagem natural para automatizar por completo ou parcialmente a tarefa de se construir uma ontologia.

Neste trabalho, foi proposta uma abordagem inovadora para a extração de relações semânticas contidas nos documentos. Foram apresentadas algumas propostas baseadas em técnicas lingüísticas e outras baseadas em técnicas estatísticas. A metodologia adotada para a extração das relações semânticas utiliza técnicas lingüísticas para a identificação dos termos e técnicas estatísticas para a identificação de conceitos e relações, caracterizando, dessa forma, um modelo híbrido. A técnica lingüística faz uso da base lexical WordNet para a determinação da lista de termos. A utilização da WordNet restringe a utilização do método para uma língua específica. Além disso, a WordNet é uma ontologia geral de modo que termos importantes e específicos de domínio podem não estar presentes na lista de termos. No entanto, as restrições em relação à língua utilizada e os termos específicos de domínio podem ser resolvidas utilizando-se outras bases lexicais. A técnica estatística faz uso de algoritmo de agrupamento para determinar a lista de conceitos e suas relações. O método proposto está mais focado em técnicas estatísticas e a escolha desta é devido às suas vantagens, tais como: apresentar menor complexidade computacional, não requer muita precisão para que a técnica traga benefícios e requer menor conhecimento da linguagem e suas estruturas.

A estratégia adotada para a identificação de conceitos e suas relações por meio da análise de correlação de termos mostrou-se bastante promissora, pois permitiu a correta identificação dos conceitos, trouxe um número adequado de conceitos e os resultados foram equiparáveis com as técnicas mais

conhecidas. A nova abordagem trouxe ainda muitas vantagens ao processo:

- Não é necessário calcular a derivada de funções. Conseqüentemente, é possível alterar a função-objetivo sem alterar o algoritmo de agrupamento.
- O caráter iterativo do processo permite que o usuário inspecione os conceitos durante a extração. O usuário pode escolher de maneira iterativa os valores dos parâmetros, tornando mais fácil a escolha de bons valores para os parâmetros.
- Conceitos manualmente definidos pelo usuário podem ser introduzidos na ontologia, deixando o algoritmo descobrir novos conceitos.
- A simplificação do modelo proposta na seção 4.4.5 introduziu o conceito de protótipo para os conceitos. Esse protótipo modifica a forma como um conceito é definido. Inicialmente a definição era extensional, ver seção 2.1.1, onde o conceito era caracterizado pela lista de termos com suas respectivas pertinências; agora, a definição é intencional em que é apresentada uma estrutura matemática e o cômputo das pertinências é feito utilizando-se esta estrutura, ver seção 4.4.5 sobre a recuperação dos índices fuzzy. Esta mudança de paradigma permite a introdução de novos termos e documentos sem a necessidade de re-execução do algoritmo de agrupamento.

O trabalho também apresenta contribuições para a área de Inteligência Computacional. O algoritmo desenvolvido pode ser aplicado a qualquer problema de agrupamento na qual a única informação disponível seja a distância entre os objetos. O algoritmo ainda apresenta as seguintes vantagens:

- Supervisão do processo de agrupamento.
- Métricas para a determinação do número adequado de grupos.
- Grande parametrização: inúmeras funções são candidatas à função-objetivo.

O trabalho apresentado nesta dissertação mostra uma forma alternativa àquelas utilizando decomposição de matrizes para a identificação de conceito em documentos. A proposta de agrupamento de termos correlatos mostrou-se bastante promissora com resultados similares às melhores técnicas de extração de conceitos e com as vantagens apresentadas anteriormente.

Entretanto, o modelo ainda depende da especificação dos parâmetros η, τ, γ, e do número L de termos. Todos estes influenciam de maneira decisiva o resultado final do processo.

Os pontos ainda a serem investigados em maiores detalhes são as escolhas das funções f e φ e a seleção dos termos que participam do processo de agrupamento. Atualmente, as funções f e φ favorecem a formação de grupos esféricos, pois exigem que todos os termos estejam mutuamente próximos. Uma análise da distribuição dos termos nos documentos pode servir de guia para a escolha de outras

funções que favoreçam outros formatos e que possa trazer resultados ainda melhores. A escolha dos termos é outro assunto que merece atenção, pois sua escolha é determinante para a atribuição de nomes significativos ao conceitos, como foi devidamente ilustrado na seção 5.18. Uma alternativa para a escolha adequada dos termos seria estender o atual modelo de documentos de modo a incluir informações sobre as palavras-chave dos documentos. Normalmente, as palavras-chave contêm termos que podem ser utilizados para caracterizar os conceitos.

Outra dimensão a ser explorada futuramente refere-se às entidades ontológicas suportadas pelo sistema. Neste trabalho, o sistema dá suporte somente às entidades mais simples: conceitos, termos e relações de instanciação entre termos-conceitos; no entanto outras relações também desempenham um papel importante para descrever o universo de discurso e podem ser utilizadas para melhorar os mecanismo de recuperação de informação.

A avaliação do modelo proposto ainda precisa passar pelos testes práticos de utilização das relações semânticas extraídas e, também, o comportamento dos algoritmos quando as bases de documentos possuem um volume maior de documentos. Avaliar quais são os ganhos de precisão e cobertura ao se utilizar ontologias geradas automaticamente são importantes para a conclusão de qualquer proposta de extração automática de ontologias. Alguns testes utilizando o sistema de recuperação de informação FROM e os conceitos extraídos pela proposta ACT foram realizados com a base de documentos 1. Esses testes mostraram que foi possível recuperar documentos que não estavam diretamente associados aos termos da consulta mas que estavam associados aos conceitos representados por esses termos.

Embora a proposta desta dissertação seja a apresentação de um algoritmo para extração de relações semânticas, as teorias e desenvolvimentos práticos realizados permitem a aplicação em muitas outras áreas de pesquisa.

As fundamentações teóricas das funções f e φ permitem o desenvolvimento de novos algoritmos de agrupamento em grafos, incorporando diversas funcionalidades ao método, tais como: agrupamento possibilístico, auto-determinação do número de grupos e funções kernel. Nesta dissertação, a técnica de agrupamento foi baseada na especificação de uma função-objetivo, que combina requisitos de cobertura e coesão de grupo, e em técnicas para sua otimização. Essa função poderia, ainda, ser otimizada utilizando técnicas multi-objetivo [53] [16]. Neste caso, a otimização ocorreria em um sistema de duas funções-objetivos, uma para avaliar a cobertura do grupos e outra para avaliar a coesão dos termos.

As fundamentações ainda podem ser estendidas para incorporar noções de bi-agrupamento possibilístico, como já foi apresentada por Filippone *et al.* [24].

Por fim, o algoritmo proposto pode ser utilizado para auxiliar o processo de decomposição NMF dado que nos dois casos o resultado foi similar.

Referências Bibliográficas

[1] Muhammad Abulaish and Lipika Dey. A fuzzy ontology generation framework for handling uncertainties and nonuniformity in domain knowledge description. *International Conference on Computing: Theory and Applications*, 35:287–293, 2007.

[2] A. A. Alizadeh. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[3] James Allan. HARD track overview. In *Proceedings of the Twelfth Text Retrieval Conference*, pages 18–21, 2004.

[4] Holger Bast, Georges Dupret, Debapriyo Majumdar, and Benjamin. Discovering a term taxonomy from term similarities using principal component analysis. *Lecture Notes in Computer Science, Springer, Porto, Portugal*, 4289:103–120, 2006.

[5] Sven Bergmann, Jan Ihmels, and Naama Barkai. Iterative signature algorithm for the analysis of large-scale gene expression data. *Phys Rev E Stat Nonlin Soft Mater Phys*, 67(3):1–18, 2003.

[6] T. Berners-Lee, T. Hendler, and J. Lassila. The semantic Web. *Scientific American*, 284:34–43, 2001.

[7] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[8] Vannevar Bush. As we may think. *The Atlantic Monthly*, 176(1):101–108, 1945.

[9] Silvia Calegari and Davide Ciucci. Fuzzy ontology and fuzzy-OWL in the KAON project. *IEEE Transaction on System, Man, And Cybernetics*, 34:1–6, 2007.

[10] Sharon A. Caraballo. Automatic construction of a hypernym-labeled noun hierarchy from text. In *In Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 120–126, 1999.

[11] B. Chandrasekaran, John R. Josephson, and V. Richard Benjamins. What are ontologies, and why do we need them? *IEEE Intelligent Systems*, 14(1):20–26, 1999.

[12] Y. Cheng and G. M. Church. Biclustering of expression data. *In Proc. ISMB'00, AAAI Press*, 8:93–103, 2000.

[13] Philipp Cimiano, Andreas Hotho, and Steffen Staab. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research*, 24:305–339, 2005.

[14] Philipp Cimiano, Aleksander Pivk, Lars Schmidt-Thieme, and Steffen Staab. Learning taxonomic relations from heterogeneous evidence. In P. Buitelaar, P. Cimiano, and B. Magnini, editors, *Ontology Learning from Text: Methods, Applications and Evaluation*, number 123 in Frontiers in Artificial Intelligence and Appl, pages 59–73. IOS Press, 2005.

[15] Maria Cláudia de Freitas and Violeta Quental. Subsídios para a elaboração automática de taxonomias. *TIL — V Workshop em Tecnologia da Informação e da Linguagem Humana*, pages 1584–1594, 2007.

[16] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, 2001.

[17] Susan T. Dumais, Michael W. Berry, and Gavin W. O Brien. Using linear algebra for intelligent information retrieval. *SIAM*, 37(4):573–595, 1995.

[18] J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57, 1973.

[19] S. Elo. A parallel genetic algorithm on the CM-2 for multi-modaloptimization. *Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, 2:818 – 822, 1994.

[20] Dominique Estival, Chris Nowak, and Andrew Zschorn. Towards ontology-based natural language processing. RDF/RDFS and OWL in language technology. *4th Workshop on NLP and XML (NLPXML-2004), ACL 2004*, pages 59–66, 2004.

[21] Ricardo A. Falbo, Fabiano B. Ruy, Juliana Pezzin, and Rodrigo Dal Moro. Ontologias e ambientes de desenvolvimento de software semânticos. *Jornadas Iberoamericanas de Ingenieria del Software e Ingenieria del Conocimento.*, I:277–292, 2004.

[22] Yuan-Jing Feng and Zu-Ren Feng. An immunity-based ant system for continuous space multimodal function optimization. *Conference on Machine Learning and Cybernetics, 2004. Proceedings of 2004 International*, 2:1050 – 1054, 2004.

[23] Dieter Fensel, Frank Van Harmelen, Michel Klein, and Hans Akkermans. On-to-knowledge: Ontology-based tools for knowledge management. In *In Proceedings of the eBusiness and eWork 2000 (EMMSEC 2000) Conference*, 2000.

[24] Maurizio Filippone, Francesco Masulli, Stefano Rovetta, Sushmita Mitra, and Haider Banka. Possibilistic approach to biclustering: An application to oligonucleotide microarray data analysis. In *CMSB*, pages 312–322, 2006.

[25] Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic. System for semi-automatic ontology construction. *3rd Annual European Semantic Web Conference ESWC-2006*, 2006.

[26] F. O. Franca, G. Bezerra, and F. J. Von Zuben. New perspectives for the biclustering problem. In Gary G. Yen, Simon M. Lucas, Gary Fogel, Graham Kendall, Ralf Salomon, Byoung-Tak Zhang, Carlos A. Coello Coello, and Thomas Philip Runarsson, editors, *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pages 753–760, Vancouver, BC, Canada, 16-21 July 2006. IEEE Press.

[27] Gaihua Fu, Christopher B Jones, and Alia I. Abdelmoty. Ontology-based spatial query expansion in information retrieval. *ODBASE: OTM Confederated International Conferences*, 3761:1466–1482, 2005.

[28] Yassine Gargouri, Bernard LefeBvre, and Jean-Guy Meunier. Ontology maintenance using textual analysis. *SCI2003: The 7th World Multiconference Systems, Cybernetics and Informatics*, pages 248–253, 2003.

[29] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proc. Natl. Acad. Sci. USA*, 97(22):12079–12084, 2000.

[30] M. Girolami. Mercer kernel based clustering in feature space. *IEEE Transactions on Neural Networks*, 13:780–784, 2002.

[31] M. Gonzalez and Vera L. S. Lima. Recuperação de informação e processamento de linguagem natural. *XXIII Congresso da Sociedade Brasileira de Computação*, 3:347–395, 2003.

[32] Grolier. Academic American encyclopedia. Grolier Electronic Publishing, Danbury, Connecticut., 1990.

[33] Thomas R. Gruber. A translation approach to portable ontology specifications. Technical report, Computer Science Department, Stanford University, California 94305, 1993.

[34] Nicola Guarino. Formal ontology and information systems. pages 3–15. IOS Press, 1998.

[35] D. E. Gustafson and W. Kessel. Fuzzy clustering with a fuzzy covariance matrix. *Proc IEEE CDC*, pages 761–766, 1979.

[36] Jiawei Han. *Data mining: concepts and techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[37] D. Harman. User-friendly systems instead of user-friendly front-ends. *Journal of the American Society for Information Science*, 43:164 – 174, 1992.

[38] John A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 1975.

[39] M. A. Hearst. *Automated Discovery of WordNet Relations,in WordNet: an electronic lexical database*. MIT Press, 1998.

[40] Yih-Jen Horng, Shyi-Ming Chen, Yu-Chuan Chang, and Chia-Hoang Lee. A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques. *IEEE Transaction on Fuzzy Systems*, 13:216–228, 2005.

[41] Y. Kluger, R. Barsi, J. T. Cheng, and M. Gerstein. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Res*, 13(4):703–716, 2003.

[42] Natalia Kozlova. Automatic ontology extraction for document classification. Master's thesis, Saarland University, 2005.

[43] R. Krishnapuram. Generation of membership functions via possibilistic clustering. *In Proceedings of the Third IEEE Conference on Fuzzy Systems and IEEE World Congress on Computational Intelligence*, 2:902–908, 1994.

[44] Raghu Krishnapuram and James M. Keller. The Possibilistic C-means Algorithm: Insights and recommendations. *IEEE Transaction on Fuzzy Systems*, 4:385 – 393, August 1996.

[45] N. Lammari and E. Mtais. Building and maintaining ontologies: a set of algorithms. *Data and Knowledge Engineering*, 48:155–176, 2004.

[46] L. Lazzeroni and A. Owen. Plaid models for gene expression data. *Statistica Sinica*, 12:61–86, 2002.

[47] Chang-Shing Lee, Zhi-Wei Jian, and Lin-Kai Huang. A fuzzy ontology and its application to news summarization. *IEEE Transaction on Systems, Man, and Cybernetics*, 35:859–880, 2005.

[48] Danial D. Lee and H. Sebastian Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–791, 1999.

[49] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

[50] M. Grüninger M. and Fox. Methodology for the design and evaluation of ontologies. In *IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing, April 13, 1995*, 1995.

[51] Govind Maddi, Chakravarthi Velvadapu, Sadand Srivastava, and James Gil de Lamadrid. Ontology Extraction from text documents by Singular Value Decomposition. Technical report, Bowie State University, Bowie, Maryland, USA, 2001.

[52] Alexander Maedche and Steffen Staab. *Mining Ontologies from Text*, pages 169–189. Springer Berlin/Heidelberg, 2000.

[53] Kuntinee Maneeratana, Kittipong Boonlong, and Nachol Chaiyaratana. Multi-objective optimisation by co-operative coevolution. In *Lecture Notes in Computer Science*, pages 772–781, 2004.

[54] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to WordNet: An on-line lexical database. *Journal of Lexicography*, 3:235–244, 1990.

[55] Boris Mirkin. *Mathematical Classification and Clustering*. Kluwer Academic Publishers, 1996.

[56] Kiyotaka Mizutani and Sadaaki Miyamoto. Possibilistic Approach to Kernel-Based Fuzzy c-Means Clustering with Entropy Regularization. In Vicenç Torra, Yasuo Narukawa, and Sadaaki Miyamoto, editors, *MDAI*, volume 3558 of *Lecture Notes in Computer Science*, pages 144–155. Springer, 2005.

[57] Hiroshi Nakagawa and Tatsunori Mori. A simple but powerful automatic term extraction method. In *COLING-02 on COMPUTERM 2002*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[58] C. P. Paice. Soft evaluation of boolean search queries in information retrieval systems. *Information Technology: Research and Development*, 3:33 – 42, 1984.

[59] Helen J. Peat and Peter Willett. The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the American Society for Information Science*, 42:378–383, 1991.

[60] Witold Pedrycz. *Knowledge-Based Clustering From Data to Information Granules*. John Wiley & Sons, 2005.

[61] Witold Pedrycz and Fernando Gomide. *Fuzzy Systems Engineering: Toward Human-Centric Computing*. John Wiley & Sons, Norwell, MA, USA, 2007.

[62] Raquel Pereira, Fernando Gomide, and Ivan Ricarte. *Fuzzy Logic and the Semantic Web*, chapter Fuzzy Relational Ontological Model in Information Search Systems, pages 395–412. Elsevier B.V, 2006.

[63] G. Salton, A. Wong, and C. S. Yang. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18:613 – 620, 1975.

[64] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24:513–523, 1988.

[65] Farial Shahnaz, Michael Berry, Paul Pauca, and Robert Plemmons. Document clustering using nonnegative matrix factorization. *Journal on Information Processing and Management*, 42(2):373–386, 2006.

[66] Alan F. Smeaton. Information retrieval: Still butting heads with natural language processing. In *Information Technology, M.T Pazienza ed., Springer-Verlag Lecture Notes in Computer Science 1299*, pages 115–138. Springer-Verlag, 1997.

[67] Irena Spasic, Sophia Ananiadou, John McNaught, and Anand Kumar. Text mining and ontologies in biomedicine: Making sense of raw text. *Briefings in Bioinformatics*, 6:239–251, 2005.

[68] R. Srikant and R. Agrawal. Mining generalized association rules. *Proc. of VLDB'95*, pages 407–419, 1995.

[69] T. Takagi and K. Kawase. A trial for data retrieval using conceptual fuzzy sets. *IEEE Transactions on Fuzzy Systems*, 9:497 505, 2001.

[70] Amos Tanay, Roded Sharan, and Ron Shamir. Biclustering algorithms: A survey. *Handbook of Computational Molecular Biology. Edited by Srinivas Aluru, Chapman*, 67(3 Pt 1), 2004.

[71] Mike Uschold. Building Ontologies: Towards A Unified Methodology. In *In 16th Annual Conf. of the British Computer Society Specialist Group on Expert Systems*, pages 16–18, 1996.

[72] Paola Velardi, Michele Missikoff, and Roberto Basili. Identification of relevant terms to support the construction of domain ontologies. In *Proceedings of the workshop on Human Language Technology and Knowledge Management*, pages 1–8, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[73] Ellen M. Voorhees. Natural language processing and information retrieval. In *Information Extraction: Towards Scalable, Adaptable Systems*, pages 32–48. Springer, 1999.

[74] W. G. Waller and D. H. Kraft. A mathematical model of a weighted boolean retrieval system. *Information Processing and Management*, 15:235 – 245, 1979.

[75] Dwi H. Widyantoro. A fuzzy ontology-based abstract search engine and its user studies. In *In The 10th IEEE International Conference on Fuzzy Systems*, pages 1291–1294. IEEE Press, 2001.

[76] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. *26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273, 2003.

[77] Oren Zamir and Oren Etzioni. Web document clustering: A feasibility demonstration. *Proc. of the 21st ACM SIGIR Conference*, pages 46–54, 1998.

Apêndice A

Comparativos

Este apêndice apresenta de forma detalhada a comparação entre a técnica de análise de correlação de termos proposta nesta dissertação e as técnicas baseadas na decomposição de matrizes, *Non-Negative Matrix Factorization* e *Latent Semantic Indexing*. Para cada uma das 16 configurações apresentadas na seção de resultados, duas tabelas são construídas para mostrar a semelhança entre os conceitos. As linhas destas matrizes representam os conceitos extraídos com nossa técnica e as colunas representam os conceitos obtidos com as técnicas de decomposição de matrizes. O conteúdo de cada célula da matriz é o índice de semelhança entre os conceitos cujo valor está compreendido entre 0 e 1. Valores próximos de 0 indicam pouca sobreposição entre os conceito e valores próximos de 1 indicam muita sobreposição entre os conceitos.

A.1 Base de Documentos 1

A.1.1 Cenário 1

- Número de termos: 77;
- Constante η: 7;

		NMF									
		1	2	3	4	5	6	7	8	9	10
ACT	knowledge	0,39	0,53	0,50	0,49	0,45	0,42	0,92	0,42	0,45	0,51
	algorithm	0,96	0,41	0,55	0,43	0,45	0,56	0,38	0,44	0,46	0,39
	variable	0,46	0,43	0,40	0,44	0,60	0,42	0,50	0,62	0,89	0,42
	representation	0,42	0,48	0,46	0,40	0,42	0,42	0,46	0,59	0,41	0,96
	simulation	0,58	0,41	0,58	0,47	0,44	0,92	0,41	0,43	0,45	0,39
	concept	0,46	0,48	0,45	0,43	0,46	0,41	0,41	0,92	0,51	0,61
	computer	0,50	0,80	0,44	0,44	0,48	0,44	0,50	0,62	0,46	0,41
	manner	0,45	0,50	0,43	0,48	0,96	0,44	0,48	0,45	0,55	0,43
	constraint	0,43	0,44	0,77	0,67	0,41	0,58	0,37	0,45	0,47	0,44
	database	0,46	0,47	0,49	0,96	0,52	0,42	0,48	0,46	0,46	0,40

Tab. A.1: Comparação com NMF ($\eta = 7$, $L = 77$).

		LSI									
		1	2	3	4	5	6	7	8	9	10
ACT	simulation	0,49	0,67	0,41	0,89	0,44	0,42	0,36	0,38	0,41	0,45
	algorithm	0,48	0,75	0,50	0,50	0,50	0,44	0,33	0,36	0,45	0,54
	variable	0,57	0,62	0,56	0,38	0,39	0,65	0,43	0,62	0,40	0,38
	manner	0,48	0,60	0,49	0,41	0,49	0,46	0,36	0,93	0,42	0,77
	representation	0,48	0,47	0,61	0,39	0,47	0,40	0,94	0,40	0,40	0,52
	concept	0,75	0,61	0,40	0,36	0,47	0,53	0,70	0,43	0,43	0,40
	computer	0,57	0,44	0,41	0,39	0,70	0,61	0,43	0,44	0,41	0,44
	database	0,61	0,66	0,40	0,53	0,43	0,36	0,41	0,67	0,84	0,45
	knowledge	0,39	0,46	0,71	0,44	0,55	0,63	0,61	0,53	0,63	0,39
	constraint	0,71	0,57	0,48	0,73	0,39	0,51	0,48	0,46	0,70	0,56

Tab. A.2: Comparação com LSI ($\eta = 7$, $L = 77$).

A.1.2 Cenário 2

- Número de termos: 154;
- Constante η: 7;

		NMF										
		1	2	3	4	5	6	7	8	9	10	11
ACT	aspect	0,91	0,48	0,41	0,53	0,45	0,52	0,41	0,47	0,45	0,43	0,50
	measurement	0,47	0,51	0,44	0,80	0,45	0,37	0,39	0,59	0,59	0,55	0,45
	algorithm	0,45	0,70	0,42	0,92	0,45	0,38	0,43	0,46	0,42	0,52	0,46
	protocol	0,43	0,48	0,45	0,49	0,44	0,41	0,94	0,46	0,49	0,48	0,46
	benchmark	0,43	0,47	0,51	0,65	0,41	0,43	0,45	0,47	0,42	0,90	0,43
	management	0,43	0,44	0,92	0,45	0,44	0,45	0,47	0,44	0,57	0,48	0,42
	attention	0,42	0,49	0,49	0,42	0,48	0,57	0,45	0,44	0,62	0,59	0,42
	manipulation	0,44	0,54	0,50	0,41	0,44	0,91	0,49	0,44	0,43	0,46	0,41
	variable	0,42	0,51	0,43	0,46	0,71	0,41	0,76	0,47	0,46	0,48	0,53
	neuron	0,46	0,55	0,39	0,55	0,62	0,37	0,42	0,87	0,38	0,47	0,65
	elements	0,42	0,44	0,49	0,49	0,51	0,49	0,52	0,45	0,58	0,50	0,73

Tab. A.3: Comparação com NMF ($\eta = 7$, $L = 154$).

		LSI										
		1	2	3	4	5	6	7	8	9	10	11
ACT	variable	0,59	0,42	0,40	0,82	0,50	0,40	0,45	0,45	0,62	0,61	0,46
	algorithm	0,69	0,43	0,42	0,45	0,56	0,63	0,47	0,89	0,41	0,41	0,39
	elements	0,62	0,43	0,49	0,67	0,47	0,42	0,74	0,51	0,41	0,59	0,43
	aspect	0,42	0,86	0,46	0,44	0,53	0,88	0,46	0,55	0,41	0,48	0,47
	attention	0,58	0,44	0,68	0,43	0,70	0,39	0,44	0,38	0,45	0,40	0,57
	measurement	0,65	0,51	0,60	0,37	0,45	0,52	0,48	0,78	0,58	0,49	0,37
	protocol	0,52	0,40	0,40	0,85	0,45	0,43	0,47	0,46	0,76	0,42	0,45
	neuron	0,48	0,54	0,42	0,43	0,50	0,42	0,49	0,70	0,64	0,86	0,49
	manipulation	0,50	0,46	0,55	0,43	0,43	0,57	0,55	0,39	0,49	0,39	0,90
	management	0,54	0,40	0,44	0,37	0,49	0,39	0,66	0,37	0,48	0,40	0,41
	benchmark	0,61	0,63	0,46	0,43	0,51	0,38	0,43	0,70	0,54	0,37	0,46

Tab. A.4: Comparação com LSI ($\eta = 7$, $L = 154$).

A.1.3 Cenário 3

- Número de termos: 77;
- Constante η: 20;

		NMF			
		1	2	3	4
ACT	theory	0,41	0,36	**0,95**	0,40
	algorithm	**0,90**	0,51	0,33	0,46
	characteristic	0,52	**0,85**	0,36	0,47
	manner	0,52	0,46	0,34	**0,91**

Tab. A.5: Comparação com NMF ($\eta = 20$, $L = 77$).

		LSI			
		1	2	3	4
ACT	characteristic	0,39	0,75	0,49	**0,85**
	algorithm	0,29	**0,81**	0,39	0,46
	theory	**0,90**	0,53	0,40	0,35
	manner	0,29	0,78	**0,81**	0,40

Tab. A.6: Comparação com LSI ($\eta = 20$, $L = 77$).

A.1.4 Cenário 4

- Número de termos: 154;
- Constante η: 20;

		NMF				
		1	2	3	4	5
ACT	algorithm	0,50	0,41	0,28	0,90	0,67
	aspect	0,40	0,79	0,52	0,46	0,43
	protocol	0,46	0,37	0,72	0,51	0,49
	management	0,39	0,39	0,84	0,40	0,52
	vector	0,92	0,44	0,40	0,42	0,49

Tab. A.7: Comparação com NMF ($\eta = 20$, $L = 154$).

		LSI				
		1	2	3	4	5
ACT	management	0,35	0,32	0,39	0,38	0,65
	vector	0,34	0,55	0,92	0,52	0,60
	aspect	0,83	0,46	0,40	0,74	0,45
	algorithm	0,50	0,90	0,47	0,48	0,74
	protocol	0,39	0,42	0,46	0,36	0,72

Tab. A.8: Comparação com LSI ($\eta = 20$, $L = 154$).

A.2 Base de Documentos 2

A.2.1 Cenário 1

- Número de termos: 77;
- Constante η: 7;

		NMF						
		1	2	3	4	5	6	7
ACT	language	0,51	0,39	0,42	0,39	**0,90**	0,46	0,37
	environment	0,54	0,39	0,43	0,38	**0,88**	0,44	0,39
	input	0,44	**0,93**	0,45	0,45	0,33	0,47	0,55
	kind	0,58	0,42	0,41	0,49	0,43	0,49	**0,76**
	algorithm	0,36	0,52	0,47	**0,86**	0,34	**0,60**	0,54
	application	0,34	0,45	**0,97**	0,48	0,42	0,52	0,55
	selection	**0,87**	0,43	0,41	0,38	0,60	0,42	0,41

Tab. A.9: Comparação com NMF ($\eta = 7$, $L = 77$).

		LSI						
		1	2	3	4	5	6	7
ACT	selection	0,42	0,60	0,40	0,36	**0,76**	0,62	**0,65**
	algorithm	**0,67**	**0,74**	**0,54**	0,65	0,28	0,49	0,36
	language	0,53	0,55	0,45	0,39	**0,88**	0,38	0,38
	kind	0,31	**0,71**	0,48	**0,67**	0,48	0,45	0,51
	input	0,48	0,41	0,53	0,39	0,32	**0,86**	0,49
	environment	0,49	0,57	0,42	0,36	**0,87**	0,40	0,39
	application	0,60	0,57	0,52	0,39	0,35	0,35	**0,65**

Tab. A.10: Comparação com LSI ($\eta = 7$, $L = 77$).

A.2.2 Cenário 2

- Número de termos: 154;
- Constante η: 7;

		NMF											
		1	2	3	4	5	6	7	8	9	10	11	12
ACT	input	0,43	0,91	0,52	0,47	0,49	0,50	0,47	0,40	0,64	0,51	0,38	0,44
	commitment	0,52	0,40	0,57	0,44	0,47	0,46	0,50	0,63	0,43	0,47	0,52	0,43
	subset	0,50	0,54	0,47	0,63	0,42	0,92	0,42	0,43	0,43	0,49	0,44	0,47
	library	0,43	0,52	0,53	0,41	0,91	0,44	0,50	0,47	0,44	0,48	0,59	0,45
	principle	0,91	0,48	0,46	0,61	0,45	0,48	0,62	0,42	0,47	0,44	0,39	0,43
	methodology	0,43	0,49	0,45	0,43	0,52	0,46	0,44	0,44	0,39	0,55	0,88	0,46
	computation	0,51	0,49	0,44	0,89	0,43	0,50	0,43	0,52	0,57	0,45	0,40	0,44
	selection	0,44	0,55	0,93	0,44	0,52	0,47	0,47	0,49	0,42	0,48	0,47	0,48
	extension	0,43	0,41	0,53	0,43	0,45	0,51	0,42	0,49	0,58	0,49	0,46	0,91
	period	0,49	0,42	0,41	0,47	0,46	0,41	0,68	0,56	0,48	0,43	0,56	0,56
	detection	0,46	0,48	0,42	0,50	0,45	0,46	0,63	0,58	0,92	0,43	0,36	0,44
	creation	0,43	0,45	0,47	0,39	0,60	0,48	0,41	0,47	0,42	0,76	0,73	0,50

Tab. A.11: Comparação com NMF ($\eta = 7$, $L = 154$).

		LSI											
		1	2	3	4	5	6	7	8	9	10	11	12
ACT	commitment	0,42	0,71	0,41	0,41	0,36	0,47	0,42	0,48	0,59	0,62	0,40	0,43
	creation	0,52	0,58	0,56	0,44	0,45	0,85	0,68	0,59	0,41	0,42	0,43	0,44
	extension	0,52	0,51	0,39	0,49	0,45	0,51	0,45	0,56	0,43	0,35	0,61	0,73
	computation	0,67	0,36	0,45	0,60	0,54	0,35	0,37	0,68	0,53	0,66	0,42	0,55
	input	0,45	0,29	0,71	0,58	0,67	0,44	0,44	0,50	0,67	0,36	0,45	0,53
	period	0,48	0,59	0,47	0,37	0,53	0,51	0,62	0,56	0,40	0,59	0,63	0,50
	methodology	0,54	0,61	0,59	0,47	0,55	0,81	0,53	0,58	0,43	0,52	0,61	0,39
	principle	0,55	0,63	0,46	0,52	0,69	0,33	0,52	0,55	0,44	0,40	0,45	0,37
	library	0,51	0,39	0,42	0,39	0,45	0,79	0,47	0,50	0,50	0,43	0,44	0,42
	detection	0,41	0,37	0,59	0,41	0,48	0,31	0,45	0,57	0,41	0,48	0,49	0,80
	selection	0,41	0,43	0,42	0,46	0,44	0,58	0,54	0,55	0,88	0,41	0,56	0,41
	subset	0,48	0,37	0,49	0,92	0,46	0,38	0,47	0,69	0,47	0,52	0,49	0,47

Tab. A.12: Comparação com LSI ($\eta = 7$, $L = 154$).

A.2.3 Cenário 3

- Número de termos: 77;
- Constante η: 20;

		NMF					
		1	2	3	4	5	6
ACT	decomposition	0,29	0,86	0,44	0,75	0,56	0,44
	algorithm	0,23	0,58	0,74	0,44	0,76	0,47
	data	0,41	0,44	0,73	0,57	0,61	0,42
	knowledge	0,91	0,32	0,36	0,55	0,40	0,38
	data	0,40	0,48	0,65	0,48	0,41	0,77
	information	0,72	0,66	0,43	0,45	0,35	0,41

Tab. A.13: Comparação com NMF ($\eta = 20$, $L = 77$).

		LSI					
		1	2	3	4	5	6
ACT	algorithm	0,67	0,39	0,24	0,66	0,47	0,58
	data	0,73	0,35	0,41	0,73	0,87	0,59
	information	0,28	0,58	0,66	0,73	0,40	0,58
	knowledge	0,52	0,39	0,91	0,58	0,45	0,32
	data	0,58	0,83	0,37	0,66	0,52	0,48
	decomposition	0,29	0,39	0,34	0,62	0,57	0,48

Tab. A.14: Comparação com LSI ($\eta = 20$, $L = 77$).

A.2.4 Cenário 4

- Número de termos: 154;
- Constante η: 20;

		NMF							
		1	2	3	4	5	6	7	8
ACT	selection	0,46	0,41	0,71	0,48	0,50	0,46	0,62	0,40
	tendency	0,44	0,84	0,44	0,58	0,49	0,45	0,33	0,60
	kind	0,83	0,45	0,36	0,43	0,46	0,44	0,56	0,62
	identification	0,38	0,41	0,64	0,49	0,40	0,47	0,73	0,43
	detection	0,65	0,43	0,38	0,47	0,43	0,42	0,45	0,88
	library	0,42	0,42	0,52	0,41	0,47	0,79	0,72	0,37
	resource	0,48	0,42	0,51	0,43	0,85	0,53	0,58	0,39
	subset	0,56	0,48	0,49	0,92	0,44	0,40	0,39	0,52

Tab. A.15: Comparação com NMF ($\eta = 20$, $L = 154$).

		LSI							
		1	2	3	4	5	6	7	8
ACT	library	0,42	0,52	0,54	0,46	0,52	0,53	0,37	0,86
	identification	0,57	0,54	0,56	0,55	0,76	0,51	0,40	0,68
	tendency	0,55	0,58	0,58	0,43	0,50	0,56	0,36	0,28
	selection	0,48	0,44	0,58	0,74	0,38	0,48	0,53	0,60
	kind	0,36	0,49	0,66	0,37	0,41	0,50	0,78	0,47
	detection	0,40	0,48	0,65	0,38	0,53	0,38	0,86	0,34
	subset	0,85	0,45	0,73	0,44	0,50	0,36	0,58	0,34
	resource	0,38	0,36	0,65	0,62	0,45	0,68	0,40	0,69

Tab. A.16: Comparação com LSI ($\eta = 20$, $L = 154$).

A.3 Base de Documentos 3

A.3.1 Cenário 1

- Número de termos: 77;
- Constante η: 7;

Comparação do conceito obtido pela técnica de correlação de termo e as outras baseadas em decomposição de matrizes, NMF e LSI, resultaram em um índice de semelhança de 0.92 para as duas técnicas.

A.3.2 Cenário 2

- Número de termos: 154;
- Constante η: 7;

		NMF			
		1	2	3	4
ACT	elements	0,67	0,48	0,55	0,58
	algorithm	0,69	0,43	0,68	0,62
	manner	0,81	0,35	0,52	0,55
	phenotype	0,47	0,65	0,47	0,64

Tab. A.17: Comparação com NMF ($\eta = 7$, $L = 154$).

		LSI			
		1	2	3	4
ACT	manner	0,73	0,51	0,33	0,64
	elements	0,76	0,56	0,41	0,46
	phenotype	0,71	0,60	0,55	0,46
	algorithm	0,92	0,37	0,40	0,47

Tab. A.18: Comparação com LSI ($\eta = 7$, $L = 154$).

A.3.3 Cenário 3

- Número de termos: 77;
- Constante η: 20;

Comparação do conceito obtido pela técnica de correlação de termo e as outras baseadas em decomposição de matrizes, NMF e LSI, resultaram em um índice de semelhança de 0.91 para as duas técnicas.

A.3.4 Cenário 4

- Número de termos: 154;
- Constante η: 20;

Comparação do conceito obtido pela técnica de correlação de termo e as outras baseadas em decomposição de matrizes, NMF e LSI, resultaram em um índice de semelhança de 0.93 para as duas técnicas.

A.4 Base de Documentos 4

A.4.1 Cenário 1

- Número de termos: 77;
- Constante η: 7;

		NMF											
		1	2	3	4	5	6	7	8	9	10	11	12
ACT	examination	0,45	0,46	0,62	0,55	0,42	0,44	0,63	0,44	0,50	0,48	0,46	0,44
	construction	0,47	0,47	0,44	0,85	0,42	0,46	0,42	0,47	0,47	0,43	0,45	0,81
	implementation	0,45	0,50	0,46	0,44	0,82	0,44	0,56	0,44	0,45	0,44	0,47	0,46
	limitation	0,44	0,50	0,46	0,44	0,97	0,44	0,44	0,44	0,50	0,45	0,43	0,45
	cambridge	0,41	0,46	0,56	0,44	0,47	0,54	0,83	0,46	0,45	0,45	0,47	0,52
	argument	0,42	0,94	0,49	0,46	0,41	0,51	0,40	0,56	0,43	0,50	0,51	0,47
	policy	0,46	0,61	0,43	0,45	0,42	0,78	0,47	0,58	0,43	0,48	0,53	0,47
	dynamics	0,48	0,43	0,46	0,48	0,45	0,55	0,47	0,42	0,96	0,45	0,49	0,54
	consideration	0,44	0,58	0,43	0,46	0,43	0,50	0,42	0,96	0,45	0,46	0,45	0,51
	city	0,77	0,44	0,46	0,46	0,40	0,65	0,43	0,46	0,44	0,62	0,43	0,49
	mixture	0,42	0,47	0,95	0,46	0,44	0,54	0,43	0,44	0,49	0,46	0,47	0,47
	asia	0,96	0,45	0,44	0,44	0,42	0,46	0,42	0,47	0,46	0,46	0,50	0,50

Tab. A.19: Comparação com NMF ($\eta = 7$, $L = 77$).

		LSI											
		1	2	3	4	5	6	7	8	9	10	11	12
ACT	examination	0,44	0,41	0,47	0,53	0,46	0,48	0,42	0,42	0,66	0,43	0,42	0,38
	construction	0,57	0,63	0,41	0,77	0,83	0,65	0,45	0,64	0,63	0,50	0,56	0,60
	implementation	0,43	0,65	0,60	0,42	0,58	0,47	0,41	0,41	0,41	0,40	0,41	0,64
	limitation	0,46	0,53	0,59	0,38	0,62	0,43	0,43	0,41	0,39	0,41	0,70	0,56
	cambridge	0,52	0,44	0,39	0,60	0,42	0,49	0,52	0,40	0,54	0,48	0,46	0,65
	argument	0,65	0,48	0,47	0,39	0,55	0,56	0,52	0,40	0,45	0,88	0,46	0,43
	policy	0,79	0,63	0,43	0,41	0,42	0,50	0,52	0,45	0,41	0,83	0,61	0,53
	dynamics	0,66	0,47	0,42	0,56	0,50	0,56	0,85	0,60	0,44	0,38	0,46	0,45
	consideration	0,68	0,40	0,62	0,56	0,49	0,58	0,52	0,48	0,40	0,79	0,51	0,66
	city	0,57	0,68	0,74	0,48	0,39	0,51	0,52	0,75	0,59	0,49	0,51	0,48
	mixture	0,47	0,42	0,48	0,40	0,42	0,45	0,62	0,40	0,88	0,44	0,54	0,63
	asia	0,41	0,47	0,51	0,40	0,41	0,46	0,43	0,92	0,57	0,42	0,42	0,49

Tab. A.20: Comparação com LSI ($\eta = 7$, $L = 77$).

A.4.2 Cenário 2

- Número de termos: 154;
- Constante η: 7;

		NMF																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ACT	implication	0,49	0,48	0,51	0,47	0,46	0,59	0,50	0,52	0,48	0,49	0,45	0,44	0,49	0,46	0,42	0,52	0,75	0,52
	message	0,47	0,49	0,96	0,51	0,46	0,54	0,44	0,48	0,46	0,44	0,43	0,46	0,47	0,46	0,49	0,50	0,46	0,45
	commerce	0,49	0,43	0,45	0,49	0,45	0,56	0,47	0,46	0,92	0,44	0,53	0,50	0,47	0,45	0,43	0,48	0,56	0,55
	hall	0,46	0,50	0,46	0,53	0,47	0,45	0,45	0,48	0,45	0,51	0,52	0,47	0,46	0,50	0,89	0,51	0,45	0,45
	taxon	0,49	0,46	0,46	0,47	0,47	0,42	0,44	0,45	0,46	0,79	0,46	0,46	0,49	0,51	0,81	0,51	0,44	0,46
	story	0,46	0,47	0,52	0,49	0,47	0,46	0,47	0,46	0,47	0,55	0,45	0,50	0,47	0,46	0,51	0,45	0,45	0,56
	substance	0,48	0,52	0,51	0,48	0,94	0,48	0,46	0,45	0,45	0,52	0,49	0,44	0,51	0,54	0,47	0,48	0,50	0,46
	money	0,47	0,44	0,48	0,47	0,47	0,94	0,44	0,48	0,50	0,48	0,44	0,42	0,48	0,62	0,43	0,45	0,50	0,46
	hypothesis	0,48	0,48	0,46	0,45	0,46	0,50	0,48	0,95	0,46	0,53	0,43	0,50	0,53	0,47	0,46	0,50	0,53	0,47
	architecture	0,47	0,49	0,56	0,83	0,45	0,56	0,44	0,43	0,50	0,44	0,45	0,46	0,46	0,48	0,51	0,48	0,48	0,44
	mixture	0,45	0,63	0,48	0,58	0,49	0,45	0,48	0,46	0,45	0,47	0,91	0,47	0,44	0,45	0,45	0,50	0,46	0,53
	failure	0,47	0,95	0,47	0,53	0,52	0,41	0,53	0,46	0,46	0,45	0,56	0,47	0,44	0,46	0,46	0,46	0,58	0,46
	framework	0,45	0,45	0,56	0,60	0,47	0,46	0,48	0,47	0,45	0,43	0,54	0,46	0,47	0,46	0,47	0,90	0,49	0,52
	magnitude	0,47	0,49	0,45	0,47	0,46	0,53	0,91	0,47	0,46	0,50	0,47	0,52	0,47	0,48	0,49	0,43	0,47	0,50
	percent	0,46	0,45	0,46	0,49	0,45	0,52	0,50	0,48	0,49	0,55	0,49	0,52	0,46	0,45	0,45	0,47	0,48	0,92
	carbon	0,46	0,45	0,73	0,53	0,45	0,46	0,52	0,46	0,48	0,44	0,47	0,51	0,65	0,45	0,47	0,51	0,46	0,46
	transition	0,88	0,45	0,43	0,50	0,48	0,49	0,53	0,49	0,53	0,44	0,47	0,49	0,63	0,48	0,52	0,46	0,49	0,47
	emphasis	0,46	0,45	0,45	0,46	0,45	0,45	0,50	0,48	0,46	0,49	0,47	0,81	0,51	0,52	0,48	0,46	0,46	0,48

Tab. A.21: Comparação com NMF ($\eta = 7$, $L = 154$).

		LSI																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ACT	commerce	0,49	0,48	0,64	0,55	0,41	0,43	0,48	0,49	0,51	0,57	0,68	0,51	0,67	0,49	0,71	0,38	0,46	0,49
	carbon	0,42	0,45	0,48	0,50	0,65	0,47	0,54	0,62	0,56	0,44	0,51	0,64	0,62	0,43	0,46	0,51	0,47	0,64
	hall	0,48	0,42	0,56	0,46	0,56	0,60	0,49	0,40	0,45	0,52	0,43	0,40	0,44	0,66	0,42	0,51	0,44	0,40
	implication	0,45	0,46	0,44	0,45	0,43	0,46	0,56	0,48	0,44	0,56	0,69	0,56	0,50	0,50	0,62	0,61	0,75	0,48
	message	0,42	0,53	0,54	0,45	0,85	0,52	0,44	0,42	0,60	0,50	0,60	0,64	0,64	0,57	0,55	0,64	0,56	0,72
	percent	0,60	0,44	0,43	0,52	0,40	0,60	0,57	0,63	0,52	0,64	0,56	0,61	0,67	0,54	0,50	0,50	0,47	0,53
	taxon	0,43	0,51	0,61	0,45	0,44	0,79	0,55	0,44	0,48	0,55	0,41	0,39	0,49	0,54	0,42	0,69	0,41	0,43
	failure	0,61	0,55	0,46	0,53	0,51	0,54	0,38	0,58	0,53	0,67	0,47	0,48	0,40	0,47	0,40	0,37	0,65	0,59
	hypothesis	0,60	0,54	0,74	0,55	0,40	0,42	0,44	0,48	0,63	0,43	0,64	0,53	0,41	0,47	0,48	0,76	0,65	0,44
	money	0,54	0,47	0,42	0,49	0,56	0,59	0,60	0,47	0,64	0,47	0,81	0,51	0,42	0,45	0,89	0,59	0,48	0,44
	magnitude	0,49	0,45	0,41	0,53	0,40	0,43	0,48	0,50	0,40	0,45	0,59	0,82	0,39	0,56	0,48	0,48	0,44	0,53
	substance	0,44	0,78	0,39	0,50	0,58	0,46	0,44	0,43	0,44	0,58	0,48	0,43	0,50	0,46	0,47	0,51	0,45	0,49
	story	0,45	0,50	0,49	0,41	0,47	0,49	0,45	0,45	0,41	0,49	0,36	0,45	0,49	0,43	0,45	0,52	0,43	0,49
	framework	0,45	0,50	0,46	0,62	0,61	0,53	0,67	0,51	0,44	0,47	0,52	0,52	0,72	0,60	0,45	0,48	0,68	0,53
	transition	0,42	0,46	0,46	0,50	0,46	0,45	0,53	0,55	0,64	0,45	0,52	0,48	0,44	0,67	0,53	0,43	0,44	0,42
	emphasis	0,59	0,43	0,47	0,46	0,41	0,40	0,58	0,58	0,44	0,42	0,41	0,54	0,48	0,42	0,43	0,45	0,41	0,48
	mixture	0,75	0,47	0,41	0,69	0,49	0,58	0,43	0,50	0,56	0,49	0,50	0,57	0,68	0,47	0,42	0,41	0,51	0,43
	architecture	0,49	0,47	0,61	0,50	0,78	0,72	0,48	0,61	0,47	0,41	0,55	0,46	0,42	0,51	0,55	0,41	0,50	0,56

Tab. A.22: Comparação com LSI ($\eta = 7$, $L = 154$).

A.4.3 Cenário 3

- Número de termos: 77;
- Constante η: 20;

		NMF			
		1	2	3	4
ACT	dynamics	0,81	0,41	0,34	0,59
	argument	0,39	0,47	0,86	0,36
	mixture	0,48	0,87	0,41	0,36
	city	0,44	0,41	0,36	0,82

Tab. A.23: Comparação com NMF ($\eta = 20$, $L = 77$).

		LSI			
		1	2	3	4
ACT	mixture	0,43	0,35	0,39	0,76
	argument	0,84	0,42	0,91	0,46
	dynamics	0,59	0,49	0,35	0,36
	city	0,51	0,87	0,41	0,64

Tab. A.24: Comparação com LSI ($\eta = 20$, $L = 77$).

A.4.4 Cenário 4

- Número de termos: 154;
- Constante η: 20;

		NMF								
		1	2	3	4	5	6	7	8	9
ACT	failure	0,56	0,60	0,42	0,45	0,38	0,42	0,85	0,51	0,42
	nation	0,46	0,46	0,48	0,51	0,55	0,53	0,46	0,47	0,57
	africa	0,47	0,45	0,41	0,91	0,42	0,57	0,49	0,43	0,43
	hypothesis	0,45	0,47	0,43	0,47	0,47	0,89	0,41	0,58	0,49
	magnitude	0,41	0,61	0,41	0,43	0,48	0,52	0,43	0,87	0,51
	payment	0,47	0,41	0,48	0,41	0,88	0,53	0,38	0,44	0,74
	methodology	0,59	0,73	0,41	0,41	0,40	0,46	0,50	0,50	0,54
	suggestion	0,50	0,40	0,93	0,40	0,53	0,50	0,40	0,42	0,43
	hospital	0,59	0,45	0,53	0,45	0,52	0,60	0,45	0,47	0,49

Tab. A.25: Comparação com NMF ($\eta = 20$, $L = 154$).

		LSI								
		1	2	3	4	5	6	7	8	9
ACT	suggestion	0,53	0,60	0,61	0,82	0,66	0,71	0,53	0,63	0,47
	failure	0,35	0,47	0,42	0,47	0,54	0,49	0,42	0,35	0,42
	hospital	0,50	0,46	0,60	0,63	0,52	0,43	0,40	0,53	0,58
	hypothesis	0,50	0,61	0,69	0,37	0,75	0,42	0,51	0,69	0,71
	magnitude	0,47	0,81	0,63	0,37	0,46	0,43	0,60	0,44	0,43
	payment	0,95	0,54	0,85	0,54	0,44	0,50	0,53	0,55	0,54
	nation	0,55	0,38	0,61	0,45	0,49	0,44	0,49	0,56	0,47
	methodology	0,42	0,52	0,42	0,48	0,43	0,47	0,58	0,35	0,62
	africa	0,41	0,38	0,46	0,37	0,51	0,46	0,48	0,71	0,59

Tab. A.26: Comparação com LSI ($\eta = 20$, $L = 154$).

Apêndice B

Protótipo

Os algoritmos e procedimentos propostos nesta dissertação foram implementados na linguagem de programação Java e fazem uso de algumas bibliotecas e frameworks para o processamento de textos, operações lineares com matrizes e interfaces gráficas. A escolha da linguagem foi motivada pela sua interoperabilidade entre diversos sistemas operacionais, disponibilidade de bibliotecas gratuitas, processadores XML, interface gráfica e, sobretudo, conhecimento prévio do autor nesta linguagem.

O software é um aplicativo gráfico convencional constituído de dois painéis principais, como ilustrado na figura B.1. O painel esquerdo rotulado com *Ferramentas e Dados do Sistema* permite ao usuário visualizar os documentos da base, extrair conceitos, realizar consultas, dentre outros. Já o painel direito rotulado com *Área de Trabalho* é o local onde as interfaces de visualização, extração e buscadores são abertas para que o usuário possa operar sobre os dados. As ações são disparadas ao se realizar um *double-click* nos ícones presentes no painel esquerdo.

Opcionalmente, o software ainda possui a possibilidade de salvar os dados editados pela interface em um arquivo de sessão e carregá-los para posterior edição. Estas opções estão disponíveis no menu *Arquivo*.

A Área de Trabalho permite que múltiplas instâncias das ferramentas de edição sejam abertas. A forma de gerenciamento é através de tabs que alocam um painel para cada instância. A figura B.1 em particular mostra o mecanismo de busca de documentos. O buscador implementado é aquele baseado no modelo ontológico relacional fuzzy (FROM).

B.1 Arquitetura

A arquitetura geral do software é apresentada no diagrama em blocos da figura B.2. Neste diagrama, são apresentados os principais módulos, suas interdependências e as interfaces com os dispositivos de entrada e saída. A descrição detalhada de cada módulo é apresentada abaixo:

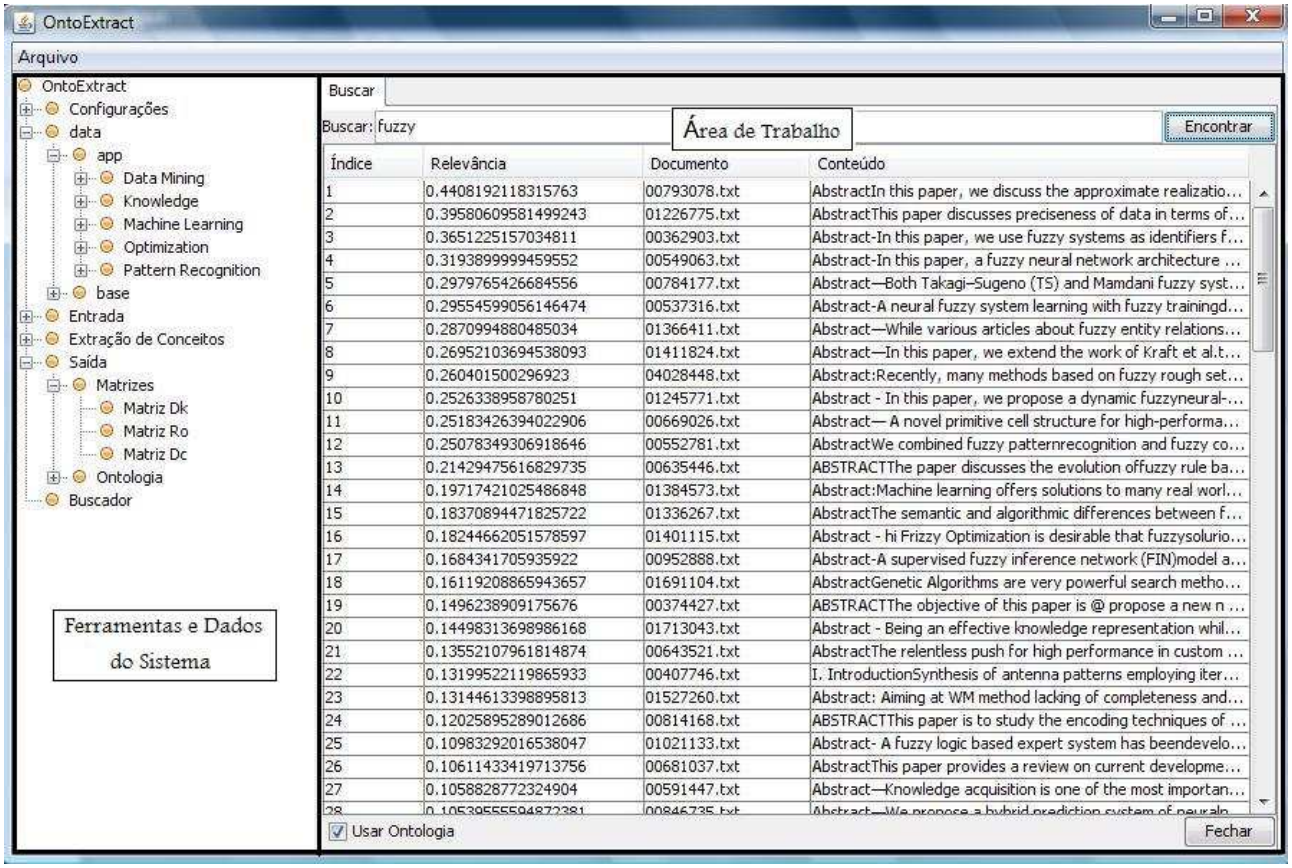


Fig. B.1: Janela Principal

Base de Documentos: Compreende a lista de documentos que é utilizada como base de teste para os procedimentos de extração de conceitos. Neste protótipo, esta base nada mais é que um diretório que contém documentos no formato texto ASCII.

Arquivos de Configuração: É um conjunto de arquivos no formato XML que configura diversos aspectos do software, do processamento dos textos e da extração dos conceitos. São nestes arquivos que estão a lista de *stopwords*, a localização da base de documentos, a localização do dicionário digital WordNet, e a definição dos filtros responsáveis por eliminar termos não relevantes.

Módulo de Entrada/Saída: É de responsabilidade desse módulo o interfaceamento com os arquivos e recursos que estão disponíveis no sistema de arquivos. Esse módulo contém métodos para leitura de arquivos no formato ASCII e métodos para leitura e processamento de arquivos no formato XML. Os arquivos XML são processados com o auxílio da biblioteca do Apache XML Beans.

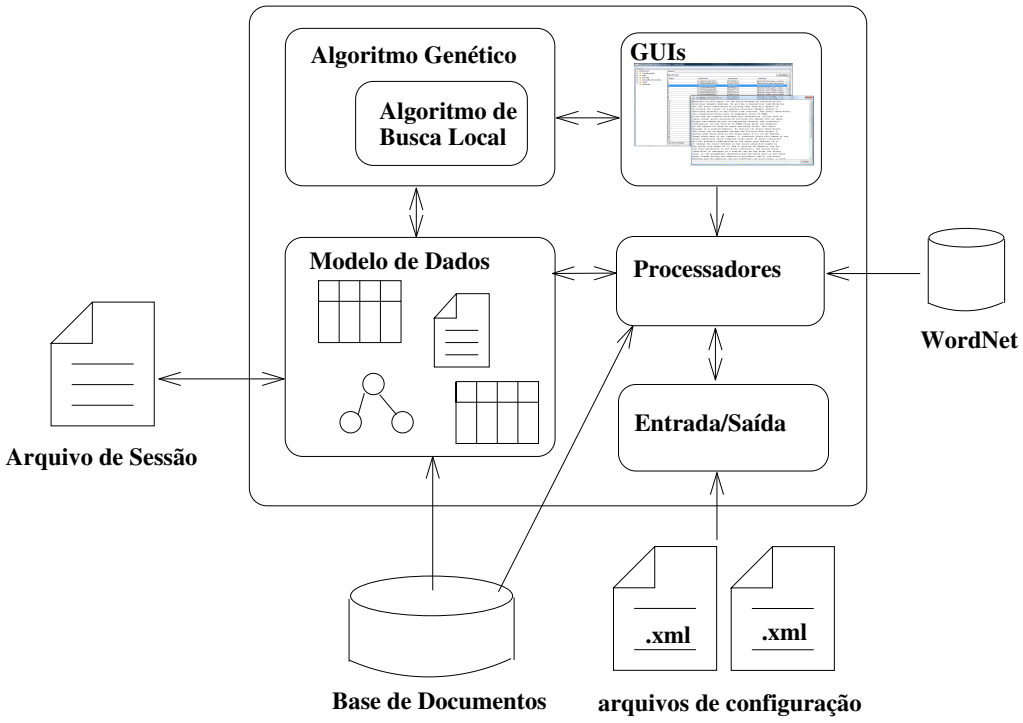


Fig. B.2: Diagrama em blocos do protótipo

Processadores: Consiste de um conjunto de procedimentos responsáveis por: realizar a análise dos documentos, obter o modelo computacional dos documentos, realizar a filtragem dos termos não-relevantes e determinação das matrizes D_k e de correlações M.

Modelo de Dados: É um repositório onde estão armazenados todos os artefatos gerados pelo módulo Processadores e pelo módulo Algoritmo Genético. É nesse repositório que são armazenados os modelos de documentos, as matrizes D_k, e R_o e as ontologias.

WordNet: Consiste de APIs de acesso e de uma base de dados que armazena um dicionário digital.

Arquivo de Sessão: É a versão persistida do repositório de dados do protótipo. O objetivo desse arquivo é armazenar todos os procedimentos e artefatos produzidos pelo operador do sistema de modo a ser possível recuperar essas informações para posterior trabalho.

Algoritmo Genético: Esse módulo é responsável por extrair conceitos da base de documentos. Trata-se de um algoritmo genético que trabalha conjuntamente com um algoritmo determinístico de busca local de maneira a garantir melhores resultados na identificação dos conceitos.

GUIs: Representa todas as interfaces gráficas disponíveis para o usuário configurar, visualizar, buscar e extrair conceitos.

B.2 Funcionalidades

As funcionalidades presentes no protótipo permitem a visualização das matrizes do modelo FROM, a visualização dos documentos da base, configurar o software, criar e editar conceitos. Todas estas funcionalidades podem ser disparadas clicando na árvore de gerência de elementos que se encontra no painel esquerdo da janela principal do aplicativo. As duas principais funcionalidades, que serão descritas em maiores detalhes, referem-se a extração e utilização de conceitos. O mecanismo de extração implementa a estratégia descrita nesta dissertação e o mecanismo de busca implementa o modelo fuzzy relacional FROM que faz uso dos conceitos obtidos pelo algoritmo de extração.

Interface para a recuperação de informações

Na interface apresentada na figura B.3, o usuário pode especificar a consulta na forma textual e o protótipo retornará a lista de documentos e a relevância de cada documento para a consulta. A consulta pode ser feita utilizando-se o conectivo lógico **e** e/ou o conectivo lógico **ou**. Para as consultas utilizando o conectivo lógico **e** basta o usuário digitar a lista de palavras separando-as com espaço. Por exemplo

fuzzy logic learning

Já a especificação de consultas utilizando o conectivo lógico **ou** o usuário deve separar as palavras com vírgula. Por exemplo

algorithm, machine, internet

A combinação de conectivos lógicos também é possível como apresentado abaixo:

genetic algorithm, machine learning, fuzzy

A interface ainda oferece a possibilidade de se escolher utilizar a ontologia para a expansão da consulta. Esta função permite avaliar a influência da ontologia para a execução da consulta.

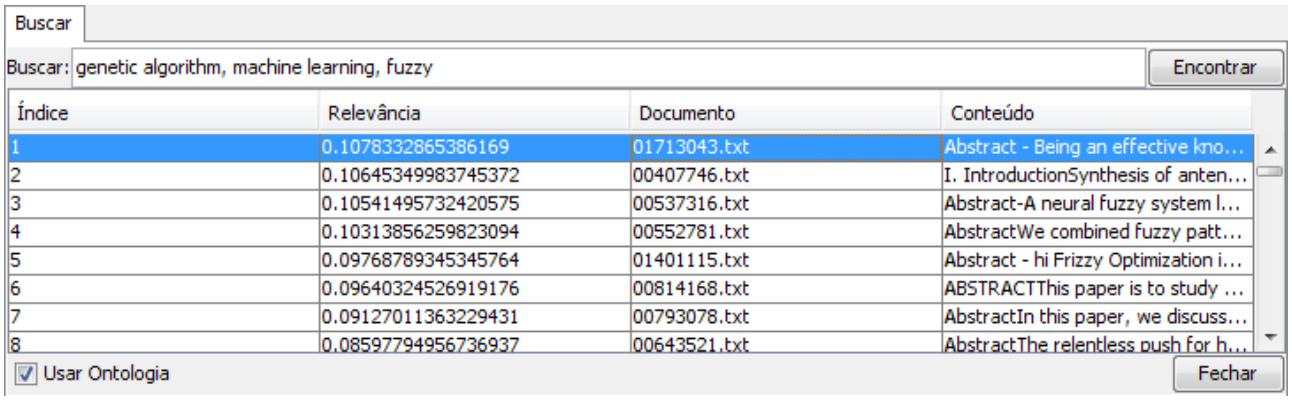


Fig. B.3: Interface FROM para recuperação de documentos

Interface para a extração de conceitos

Na interface da figura B.4, o usuário pode realizar a extração dos conceitos. A interface é composta de três áreas principais. O painel esquerdo permite que o usuário configure os parâmetros do algoritmo genético e da função-objetivo, com o painel direito o usuário pode observar os conceitos já existentes e verificar as taxas de sobreposição com o novo conceito extraído. E no painel central o usuário pode disparar a execução do algoritmo e ainda avaliar o resultado obtido. O procedimento de refinamento do conceito é realizado com o botão *Refinar* e o botão *Salvar* permite armazenar o conceito extraído. Existe ainda a opção de escolher um nome para o conceito ou obter uma sugestão para o nome. A sugestão para o nome do conceito é baseada na estratégia de Holger que foi descrita na seção 2.3.4.

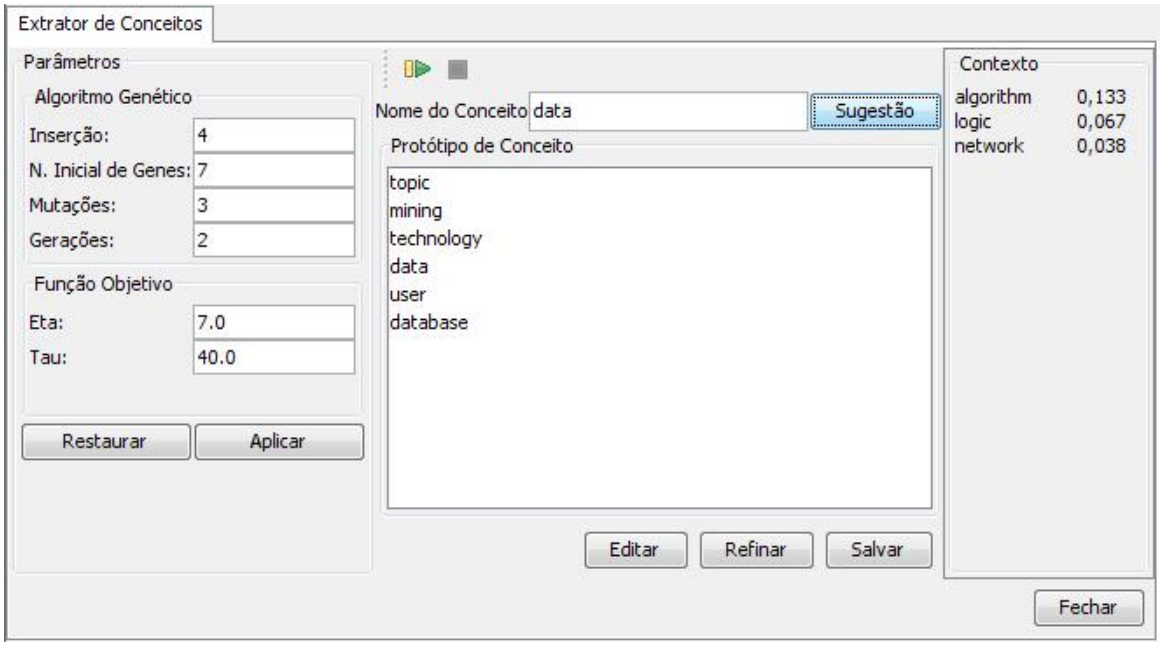


Fig. B.4: Interface para a extração de conceitos