Aprendizado não supervisionado de hierarquias de tópicos a partir de coleções textuais dinâmicas

Ricardo Marcondes Marcacini

Aprendizado não supervisionado de hierarquias de tópicos a partir de coleções textuais dinâmicas

Ricardo Marcondes Marcacini

Orientadora: *Profª Drª Solange Oliveira Rezende*

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*.

USP - São Carlos
Julho de 2011

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

Marcacini, Ricardo Marcondes
M313a Aprendizado não supervisionado de hierarquias de tópicos a partir de coleções textuais dinâmicas / Ricardo Marcondes Marcacini; orientadora Solange Oliveira Rezende -- São Carlos, 2011.
117 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2011.

1. Mineração de Textos. 2. Hierarquias de Tópicos. 3. Clustering. 4. Aprendizado de Máquina. I. Rezende, Solange Oliveira, orient. II. Título.

Agradecimentos

a Deus pelo sopro da vida e por Seu amor incondicional;

aos meus pais, Marcondes Marcacini e Davínia Gomes Marcacini, grandes responsáveis pela pessoa que me tornei, gostando eles ou não. Aos meus irmãos Renan e Renato que, mesmo com a distância dos últimos anos, sempre estiveram presentes na minha vida;

à minha orientadora e amiga, Dra. Solange Oliveira Rezende, que com sua paciência, competência e generosidade tem me auxiliado muito não apenas na vida acadêmica, mas também na minha formação como ser humano. Sua trajetória de vida e atitudes têm sido um exemplo que irei levar para sempre comigo. Sou muito agradecido por ter acreditado no meu trabalho e ser minha psicóloga nos tempos difíceis;

à todos os amigos e companheiros do Labic, pela ajuda indispensável durante a realização deste trabalho. Em especial ao grupo de mineração de textos e, também, à Dra. Maria Fernanda Moura, que plantou a semente e até hoje vem dando muitos frutos;

aos amigos de longa data Daniel, Dênis e Otávio, sempre presentes e compreensivos. Ao Rafael R. Rossi, amigo e companheiro de república, que desde a graduação tem me acompanhado nas atividades acadêmicas e nas partidas de futebol;

aos professores e funcionários do ICMC-USP, profissionais atentos e compententes nos momentos que precisei;

ao Anandsing Dwarkasing, ou simplesmente Anand, pela generosidade e valiosa ajuda no inglês, mesmo com seu pouco tempo disponível;

à Eloni M. Oliveira, uma mulher especial na minha vida, pelo apoio, confiança e compreensão. Serei sempre grato pelos momentos de alegria que têm me proporcionado;

à banca examinadora, que gentilmente aceitou o convite para essa defesa;

e à FAPESP pelo apoio financeiro.

Resumo

A necessidade de extrair conhecimento útil e inovador de grandes massas de dados textuais, tem motivado cada vez mais a investigação de métodos para Mineração de Textos. Dentre os métodos existentes, destacam-se as iniciativas para organização de conhecimento por meio de hierarquias de tópicos, nas quais o conhecimento implícito nos textos é representado em tópicos e subtópicos, e cada tópico contém documentos relacionados a um mesmo tema. As hierarquias de tópicos desempenham um papel importante na recuperação de informação, principalmente em tarefas de busca exploratória, pois permitem a análise do conhecimento de interesse em diversos níveis de granularidade e exploração interativa de grandes coleções de documentos. Para apoiar a construção de hierarquias de tópicos, métodos de agrupamento hierárquico têm sido utilizados, uma vez que organizam coleções textuais em grupos e subgrupos, de forma não supervisionada, por meio das similaridades entre os documentos. No entanto, a maioria dos métodos de agrupamento hierárquico não é adequada em cenários que envolvem coleções textuais dinâmicas, pois são exigidas frequentes atualizações dos agrupamentos. Métodos de agrupamento que respeitam os requisitos existentes em cenários dinâmicos devem processar novos documentos assim que são adicionados na coleção, realizando o agrupamento de forma incremental. Assim, neste trabalho é explorado o uso de métodos de agrupamento incremental para o aprendizado não supervisionado de hierarquias de tópicos em coleções textuais dinâmicas. O agrupamento incremental é aplicado na construção e atualização de uma representação condensada dos textos, que mantém um sumário das principais características dos dados. Os algoritmos de agrupamento hierárquico podem, então, ser aplicados sobre as representações condensadas, obtendo-se a organização da coleção textual de forma mais eficiente. Foram avaliadas experimentalmente três estratégias de agrupamento incremental da literatura, e proposta uma estratégia alternativa mais apropriada para hierarquias de tópicos. Os resultados indicaram que as hierarquias de tópicos construídas com uso de agrupamento incremental possuem qualidade próxima às hierarquias de tópicos construídas por métodos não incrementais, com significativa redução do custo computacional.

Abstract

The need to extract new and useful knowledge from large textual collections has motivated researchs on Text Mining methods. Among the existing methods, initiatives for the knowledge organization by topic hierarchies are very popular. In the topic hierarchies, the knowledge is represented by topics and subtopics, and each topic contains documents of similar content. They play an important role in information retrieval, especially in exploratory search tasks, allowing the analysis of knowledge in various levels of granularity and interactive exploration of large document collections. Hierarchical clustering methods have been used to support the construction of topic hierarchies. These methods organize textual collections in clusters and subclusters, in an unsupervised manner, using similarities among documents. However, most existing hierarchical clustering methods is not suitable for scenarios with dynamic text collections, since frequent clustering updates are necessary. Clustering methods that meet these requirements must process new documents that are inserted into textual colections, in general, through incremental clustering. Thus, we studied the incremental clustering methods for unsupervised learning of topic hierarchies for dynamic text collections. The incremental clustering is used to build and update a condensed representation of texts, which maintains a summary of the main features of the data. The hierarchical clustering algorithms are applied in these condensed representations, obtaining the textual organization more efficiently. We experimentally evaluate three incremental clustering algorithms available in the literature. Also, we propose an alternative strategy more appropriate for construction of topic hieararchies. The results indicated that the topic hierarchies construction using incremental clustering have quality similar to non-incremental methods. Furthermore, the computational cost is considerably reduced using incremental clustering methods.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Capítulo

1

Introdução

O avanço das tecnologias para aquisição e armazenamento de dados tem permitido que o volume de informação gerado em formato digital aumente de forma significativa nas organizações. Estimativas indicam que, no período de 2003 a 2010, a quantidade de informação no universo digital aumentou de 5 hexabytes (aproximadamente 5 bilhões de gigabytes) para 988 hexabytes (Gantz et al., 2008). Até o ano de 2008, contabilizou-se que a humanidade produziu cerca de 487 hexabytes de informação digital (Gantz e Reinsel, 2009, 2010).

Cerca de 80% desses dados estão em formato não estruturado, na qual uma parte significativa são textos (Kuechler, 2007). Esses textos constituem um importante repositório organizacional, que envolve o registro de histórico de atividades, memorandos, documentos internos, e-mails, projetos, estratégias e o próprio conhecimento adquirido (Han e Kamber, 2006). A organização inteligente dessas coleções textuais é de grande interesse para a maioria das instituições, pois agiliza processos de busca e recuperação da informação.

A organização de coleções textuais por meio de hierarquias de tópicos é uma das abordagens mais populares, em que os documentos são organizados em tópicos e subtópicos, e cada tópico contém documentos relacionados a um mesmo tema (Feldman e Sanger, 2006; Manning et al., 2008; Fung et al., 2009). Dessa forma, o usuário pode visualizar a informação de interesse em diversos níveis de granularidade e explorar interativamente grandes coleções de documentos.

As hierarquias de tópicos desempenham um papel importante na recuperação de informação, principalmente em tarefas de busca exploratória. Nesse tipo de tarefa, o usuário geralmente tem pouco domínio sobre o tema de interesse, o que dificulta expressar o objetivo diretamente por meio de palavras-chave (Marchionini, 2006). Assim, torna-se necessário disponibilizar previamente algumas opções para guiar o processo de busca da

informação. Para tal, cada grupo possui um conjunto de descritores que contextualizam e indicam o significado dos documentos ali agrupados. Essa organização está relacionada com a hipótese de que se um usuário está interessado em um documento específico pertencente a um determinado tópico, deve também estar interessado em outros documentos desse tópico e de seus subtópicos (Manning et al., 2008).

A construção de hierarquias de tópicos de maneira supervisionada, a exemplo do *Dmoz - Open Directory Project*[1] e *Yahoo! Directory*[2], exige um grande esforço humano. Ainda, essa construção é limitada pela grande quantidade de documentos disponíveis e pela alta frequência de atualização. Desse modo, é de grande importância a investigação de métodos para automatizar a construção de hierarquias de tópicos. Uma das maneiras de alcançar esse objetivo é por meio da Mineração de Textos, que é um conjunto de técnicas e processos com o objetivo de descobrir conhecimento inovador em coleções textuais (Rezende et al., 2003).

O processo de Mineração de Textos pode ser dividido em cinco fases principais: Identificação do Problema, Pré-Processamento, Extração de Padrões, Pós-Processamento e Uso do Conhecimento (Rezende et al., 2003). Esse projeto contribui com a fase de Extração de Padrões, no qual métodos de agrupamento de documentos podem ser utilizados para a organização de coleções textuais de maneira não supervisionada (Feldman e Sanger, 2006). Em tarefas de agrupamento, o objetivo é organizar um conjunto de objetos em grupos, em que objetos de um mesmo grupo são altamente similares entre si, mas dissimilares em relação aos objetos de outros grupos. Em outras palavras, o agrupamento é baseado no princípio de maximizar a similaridade intragrupo e minimizar a similaridade intergrupos (Everitt et al., 2001). Os métodos de agrupamento também são conhecidos como algoritmos de aprendizado por observação ou análise exploratória dos dados, pois a organização obtida é realizada por observação de regularidades nos dados, sem uso de conhecimento externo (Xu e Wunsch, 2008).

Os algoritmos de agrupamento podem ser classificados em hierárquicos ou particionais, de acordo com a estratégia de geração dos grupos (Jain et al., 1999). Nos métodos hierárquicos, um conjunto de objetos é organizado em uma hierarquia de grupos e subgrupos enquanto nos métodos particionais os objetos são divididos em uma partição com k grupos, em que o valor de k deve ser definido pelo usuário. Os métodos de agrupamento hierárquico têm sido utilizados para apoiar o aprendizado não supervisionado de hierarquias de tópicos, pois organizam coleções de documentos em grupos e subgrupos, permitindo busca exploratória em diversos níveis de granularidade (Zhao et al., 2005).

Entretanto, a maioria dos algoritmos de agrupamento considera que as coleções textu-

[1]Dmoz - Open Directory Project: http://www.dmoz.org/
[2]Yahoo Directory!: http://dir.yahoo.com/

ais são estáticas, ou seja, é necessário que todos os documentos estejam disponíveis para iniciar o processo de agrupamento. Por outro lado, existem cenários em que as coleções textuais são dinâmicas, ou seja, estão em constante crescimento, por exemplo, notícias de internet, correio eletrônico, listas de discussões e publicações científicas. Repetir o processo de agrupamento sempre que ocorrerem atualizações significativas nas coleções textuais pode não ser uma solução eficaz devido ao grande custo computacional envolvido. Assim, em coleções textuais dinâmicas, é necessário que o agrupamento seja realizado de maneira incremental, visando minimizar o custo computacional e o tempo de atualização (Sahoo et al., 2006; Xu e Wunsch, 2008; Garcia e Porrata, 2010). Esse é um cenário relevante, uma vez que a maioria das aplicações lidam com bases de textos dinâmicas, que com o passar do tempo adicionam, descartam ou alteram seus documentos, o que gera novos desafios para os métodos de organização e extração de conhecimento.

Em vista disso, neste projeto de mestrado são explorados métodos de agrupamento incremental de documentos, visando o aprendizado não supervisionado de hierarquias de tópicos em coleções textuais dinâmicas. Embora existam trabalhos que investigam o agrupamento incremental, existem poucas pesquisas voltadas especificamente para o aprendizado não supervisionado de hierarquias de tópicos. Os métodos de agrupamento incremental, explorados neste trabalho, foram aplicados na tarefa de obtenção e atualização dinâmica de uma representação condensada dos textos. A representação condensada dos textos mantém um sumário com as características principais de uma coleção textual, e permite processar uma grande quantidade de informação de forma eficiente e dinâmica. Os algoritmos de agrupamento hierárquico podem, então, ser aplicados sobre as representações condensadas sempre que necessário, obtendo-se a organização da coleção textual em tempo hábil, com redução significativa do custo computacional.

1.1 Objetivos e Hipóteses

Motivado pelos desafios comentados anteriormente e pela necessidade de métodos de aprendizado de hierarquias de tópicos que sejam úteis nos cenários atuais, este projeto de mestrado tem como objetivo principal a investigação de métodos de agrupamento incremental para permitir o aprendizado não supervisionado de hierarquias de tópicos, com foco em coleções de textos dinâmicas. A hipótese levantada neste trabalho é que é possível construir hierarquias de tópicos, utilizando métodos de agrupamento incremental, com qualidade próxima à qualidade das hierarquias de tópicos obtidas com métodos de agrupamento não incremental. O conceito de "qualidade", neste trabalho, é medido por critérios objetivos baseados na eficácia de recuperação da hierarquia de tópicos.

O objetivo geral deste projeto de mestrado é dividido em dois objetivos específicos:

- Avaliar os métodos de agrupamento incremental, investigados neste trabalho, na tarefa de obtenção de representação condensada dos textos; e

- Analisar a qualidade do agrupamento hierárquico obtido a partir de representações condensadas dos textos, em comparação com a qualidade agrupamento hierárquico obtido de forma totalmente não incremental.

É importante ressaltar que este projeto de mestrado também tem o objetivo de apoiar um projeto maior, denominado *TopTax - Topic Taxonomy* (Moura, 2009), que está em desenvolvimento no Laboratório de Inteligência Computacional (LABIC), do Instituto de Ciências Matemáticas e de Computação (ICMC-USP). No TopTax, um processo de mineração de textos é instanciado para aprendizado de hierarquias de tópicos a partir de coleções de textos. No entanto, atualmente o projeto não suporta a construção incremental de hierarquias de tópicos. A incorporação de métodos de agrupamento incremental capazes de manter a representação do conhecimento sempre válida e atualizada é de grande importância, pois irá permitir a descoberta de conhecimento nos cenários atuais, caracterizados pela alta frequência de publicação e atualização das informações.

1.2 Resultados

De forma a cumprir os objetivos propostos neste projeto de mestrado, inicialmente foi realizado um estudo das técnicas e métodos existentes para agrupamento de documentos, no contexto de organização não supervisionada de coleções textuais. Esse estudo permitiu compor uma base teórica que suporta os métodos desenvolvidos e avaliações realizadas. Também foram obtidos resultados relacionados às avaliações experimentais, além da proposta de um novo método. A seguir, um resumo dos resultados alcançados ao longo deste trabalho é apresentado.

- Com o objetivo de permitir o aprendizado de hierarquias de tópicos a partir de coleções textuais dinâmicas, foram avaliadas quatro diferentes estratégias para agrupamento incremental de documentos. Na literatura encontram-se avaliações deste tipo apenas para métodos de agrupamento não incremental, e pouca atenção tem sido dada na avaliação de métodos de agrupamento incremental. As hierarquias de tópicos obtidas com base no agrupamento incremental foram comparadas usando dois critérios de qualidade: eficácia de recuperação do agrupamento e eficácia de recuperação dos descritores. Um resultado interessante obtido dessa comparação é que algumas estratégias de agrupamento incremental permitem obter hierarquias de tópicos de qualidade similar aos métodos tradicionais da literatura (não incre-

mentais) e, ainda, com significativa redução de custo computacional. Esse resultado permitiu reforçar a hipótese levantada neste projeto de mestrado.

- Foi proposto um método de agrupamento incremental que viabiliza a seleção de descritores juntamente com a formação de grupos de documentos (Marcacini e Rezende, 2010a). Durante o processo de agrupamento, o método mantém uma rede de coocorrência de termos, que é uma estrutura útil para auxiliar atividades exploratórias e identificação de tópicos na coleção. O método, denominado *IHTC - Incremental Hierarchical Term Clustering*, se mostrou competitivo comparado aos existentes na literatura, além de permitir o aprendizado de hierarquias de tópicos com maior eficácia de recuperação dos descritores.

- Por fim, foi desenvolvido um ambiente para aprendizado de hierarquias de tópicos a partir de coleções textuais dinâmicas. O ambiente é implementado pela ferramenta *Torch - Topic Hierarchies* (Marcacini e Rezende, 2010b), que disponibiliza um conjunto de algoritmos para pré-processamento de coleções textuais dinâmicas, um módulo para configuração e execução de algoritmos de agrupamento, técnicas de seleção de descritores e identificação de tópicos, além de um módulo para avaliação e exploração visual de hierarquias de tópicos. O ambiente possui uma interface para configuração de um processo de aprendizado de hierarquias de tópicos e está disponível para a comunidade científica[3].

Durante o projeto de mestrado foram publicados três trabalhos em conferências: um artigo em conferência internacional para descrição do método IHTC (Marcacini e Rezende, 2010a), um artigo em conferência nacional para divulgação da ferramenta *Torch - Topic Hierarchies* (Marcacini e Rezende, 2010b) e um artigo em conferência nacional que discute o uso de agrupamento incremental de termos na tarefa de construção automática de diretórios web (Marcacini e Rezende, 2011). Durante o trabalho, foi possível também contribuir com um projeto de iniciação científica relacionado à avaliação de agrupamento hierárquico de documentos (Nishida et al., 2010).

1.3 Organização do Texto

O texto desta dissertação está organizado em sete capítulos.

No Capítulo 1, foram apresentadas a contextualização e motivação do tema deste projeto de mestrado, além de fornecer uma visão geral dos objetivos, hipóteses e resultados alcançados.

[3]Torch - Topic Hierarchies: http://sites.labic.icmc.usp.br/marcacini/ihtc

No Capítulo 2 é apresentada uma revisão do processo de mineração de textos para aprendizado de hierarquias de tópicos. Cada etapa desse processo é descrita em mais detalhes, apontando os trabalhos relacionados da literatura. Em especial, são apresentados os métodos de agrupamento hierárquico de documentos para a etapa de extração de padrões, além de técnicas de seleção de descritores para o agrupamento.

As estratégias de agrupamento apropriadas para cenários dinâmicos são revisadas e organizadas no Capítulo 3. Os principais algoritmos de agrupamento incremental são discutidos, apresentando os respectivos algoritmos e análise de complexidade.

No Capítulo 4 é apresentado o método IHTC, proposto durante o desenvolvimento do mestrado. São descritas as motivações para a proposta do IHTC e os algoritmos que implementam o método. Ainda, é ilustrada uma breve prova de conceito que visa analisar o modelo conceitual a partir de uma aplicação prática.

No Capítulo 5, os métodos de agrupamento discutidos, além do método proposto, são avaliados experimentalmente na tarefa de aprendizado não supervisionado de hierarquias de tópicos. São apresentadas as coleções de textos utilizadas nos experimentos, o ajuste de parâmetros dos algoritmos, os critérios de avaliação, os experimentos realizados e a análise dos resultados obtidos.

Para complementar as avaliações experimentais, no Capítulo 6 é apresentado um estudo exploratório sobre construção automática de diretórios web, utilizando dados de um projeto real denominado Dmoz. O objetivo do estudo exploratório é analisar o uso de hierarquias de tópicos em um cenário real e dinâmico, que é característico dos diretórios web.

Finalmente, no Capítulo 7, são resumidas as conclusões do trabalho, discutidas as limitações, assim como as possíveis direções para trabalhos futuros.

Capítulo

2

Mineração de Textos para Aprendizado de Hierarquias de Tópicos

A Mineração de Textos (MT) pode ser definida como um conjunto de técnicas e processos para descoberta de conhecimento inovador a partir de dados textuais (Ebecken et al., 2003). Em um contexto no qual grande parte da informação corporativa, como e-mails, memorandos internos e blogs industriais, é registrada em linguagem natural, a MT surge como uma poderosa ferramenta para gestão do conhecimento.

Pode-se afirmar que a MT é uma especialização do processo de mineração de dados. A principal diferença entre os dois processos é que, enquanto a mineração de dados convencional trabalha exclusivamente com dados estruturados, a MT lida com dados inerentemente não-estruturados (Weiss et al., 2005). Logo, na MT o primeiro desafio é obter alguma estrutura a partir dos textos e então, a partir dessa, tentar obter conhecimento.

O processo de Mineração de Textos pode ser dividido em cinco grandes etapas, formando um ciclo no qual, ao final, obtém-se o conhecimento acerca dos dados analisados. As etapas são: identificação do problema, pré-processamento, extração de padrões, pós-processamento e utilização do conhecimento (Figura 2.1). Cada etapa pode ser instanciada de acordo com a necessidade dos usuários e de cada aplicação.

Para o aprendizado não supervisionado de hierarquias de tópicos, a grande diferença está na etapa de extração de padrões, na qual são utilizados métodos de agrupamento hierárquico que organizam coleções de documentos em grupos e subgrupos. Em seguida, são aplicadas algumas técnicas de seleção de descritores para os agrupamentos formados, ou seja, palavras e expressões que auxiliam a interpretação dos grupos. Após validação dos resultados, o agrupamento hierárquico e seus descritores podem ser utilizados como uma hierarquia de tópicos para tarefas de análise exploratória dos textos (Cutting et al., 1992;

Figura 2.1: Etapas do processo de mineração de textos (Rezende et al., 2003)

Sanderson e Croft, 1999; Moura e Rezende, 2010), além de apoiar sistemas de recuperação de informação (Zeng et al., 2004; Carpineto et al., 2009).

Nas próximas seções deste capítulo, são descritos mais detalhes das etapas constituintes do processo de MT instanciadas para o aprendizado de hierarquias de tópicos, com a qual este trabalho colabora.

2.1 Identificação do Problema

Nessa etapa, delimita-se o escopo do problema, definindo o objetivo da aplicação do processo de MT. Basicamente, é necessário selecionar a base de textos com a qual irá trabalhar, o que se espera obter com a análise dos textos, as restrições existentes, a tarefa de extração de padrões apropriados para o problema e como o resultado da análise pode ser utilizado.

No contexto deste trabalho, o objetivo do processo de mineração de textos é extrair uma hierarquia de tópicos a partir de textos de um determinado domínio de conhecimento. A hierarquia de tópicos obtida pode ser aplicada para auxiliar usuários nas tarefas de organização dos documentos e análise dos tópicos da coleção, além de apoiar sistemas de recuperação de informação.

Um aspecto importante identificado nessa etapa é a restrição ao uso de informação externa, ou seja, o problema analisado não considera o uso de conhecimento de mundo ou especialistas de domínio. Diante disso, uma solução é o uso de aprendizado não supervisionado para extração do conhecimento implícito nos textos. Em especial, os métodos de agrupamento de documentos podem ser utilizados, uma vez que permitem obter uma organização dos textos de forma não supervisionada. Essa organização em grupos de doc-

umentos similares possibilita segmentar a coleção em tópicos, obtendo-se resultados de acordo com os objetivos iniciais.

Os resultados obtidos por algoritmos de agrupamento de documentos auxiliam diversas tarefas de organização da informação textual, partindo-se da hipótese que se um usuário está interessado em um documento específico pertencente a um grupo, deve também estar interessado em outros documentos desse grupo (Chakrabarti, 2002; Manning et al., 2008). Essa hipótese é utilizada em atividades de busca exploratória que ocorrem quando um usuário pode não ter certeza do que ele está procurando até que as opções disponíveis sejam apresentadas e/ou o objetivo não pode ser expresso por palavras-chave, como nos sistemas de busca tradicionais (Marchionini, 2006).

Outro aspecto de grande importância é a definição da coleção de documentos a ser utilizada, devendo-se selecionar textos que sejam mais relevantes ao domínio e à aplicação do conhecimento a ser extraído. Essa é uma atividade crítica, uma vez que os textos podem não estar disponíveis no formato adequado, como documentos não-digitalizados. Ainda, mesmo após a digitalização dos textos, é necessário convertê-los em um padrão de texto puro para que possam ser processados de maneira mais fácil. Assim, é possível iniciar a etapa conhecida como pré-processamento, conforme discutida a seguir.

2.2 Pré-processamento dos Textos

Na etapa de pré-processamento se encontra a principal diferença entre os processos de MT e processos de mineração de dados: a estruturação dos textos em um formato adequado para a extração de conhecimento. Muitos autores consideram essa etapa a que mais tempo consome durante todo o ciclo do processo de MT. O objetivo do pré-processamento é extrair de textos escritos em língua natural, inerentemente não estruturados, uma representação estruturada, concisa e manipulável por algoritmos de agrupamento de documentos. Para tal, são executadas atividades de tratamento e padronização da coleção de textos, seleção dos termos (palavras) mais significativos e, por fim, representação da coleção textual em um formato estruturado que preserve as características necessárias aos objetivos definidos na etapa de identificação do problema (Feldman e Sanger, 2006).

Os documentos da coleção podem estar em diferentes formatos, uma vez que existem diversos aplicativos para apoiar a geração e publicação de textos eletrônicos. Dependendo de como os documentos foram armazenados ou gerados, há a necessidade de padronizar as formas em que se encontram. Na **padronização dos textos**, geralmente, os documentos são convertidos para o forma de texto plano sem formatação.

Um dos maiores desafios do processo de MT é a alta dimensionalidade dos dados. Uma pequena coleção de textos pode facilmente conter milhares de termos, muitos deles

redundantes e desnecessários, que tornam lento o processo de extração de conhecimento e prejudicam a qualidade dos resultados.

A **seleção de termos** tenta solucionar esse desafio e tem o objetivo de obter um subconjunto conciso e representativo de termos da coleção textual. O primeiro passo é a eliminação de *stopwords*, que são os termos que nada acrescentam à representatividade da coleção ou que sozinhos nada significam, como artigos, pronomes e advérbios. O conjunto de *stopwords* é a *stoplist*. Essa eliminação reduz significativamente a quantidade de termos diminuindo o custo computacional das próximas etapas (Manning et al., 2008). Posteriormente, busca-se identificar as variações morfológicas e termos sinônimos. Para tal, pode-se, por exemplo, reduzir uma palavra à sua raiz por meio de processos de *stemming* ou mesmo usar dicionários ou *thesaurus*. Além disso, é possível buscar na coleção a formação de termos compostos, ou *n-gramas*, que são termos formados por mais de um elemento, porém com um único significado semântico (Manning et al., 2008; Conrado et al., 2009).

Outra forma de realizar a seleção de termos é avaliá-los por medidas estatísticas simples, como a frequência de termo, conhecida como TF (do inglês *term frequency*), e frequência de documentos, conhecida como DF (do inglês *document frequency*). A frequência de termo contabiliza a frequência absoluta de um determinado termo ao longo da coleção textual. A frequência de documentos, por sua vez, contabiliza o número de documentos em que um determinado termo aparece.

O método de Luhn (Luhn, 1958) é uma técnica tradicional para seleção de termos utilizando a medida TF. Nesse método, o autor baseou-se na Lei de Zipf (Zipf, 1932), também conhecida como Princípio do Menor Esforço. Em textos, ao contabilizar a frequência dos termos e ordenar o histograma resultante em ordem decrescente, forma-se a chamada Curva de Zipf, na qual o k-ésimo termo mais comum ocorre com frequência inversamente proporcional a k. Os termos de alta frequência são julgados não relevantes por geralmente aparecerem na grande maioria dos textos, não trazendo, em geral, informações úteis para discriminar este texto. Já os termos de baixa frequência são considerados muito raros e não possuem caráter discriminatório. Assim, são traçados pontos de corte superior e inferior da Curva de Zipf, de maneira que termos com alta e baixa frequência são descartados, considerando os termos mais significativos os de frequência intermediária (Figura 2.2).

Dado o baixo processamento demandado por esse método, ele é facilmente escalável para coleções textuais muito grandes (Soares et al., 2008). Entretanto, os pontos de corte superior e inferior sugeridos pelo autor não são exatos, sendo a subjetividade da escolha desses pontos a principal desvantagem do método.

Uma vez selecionados os termos mais representativos da coleção textual, deve-se buscar

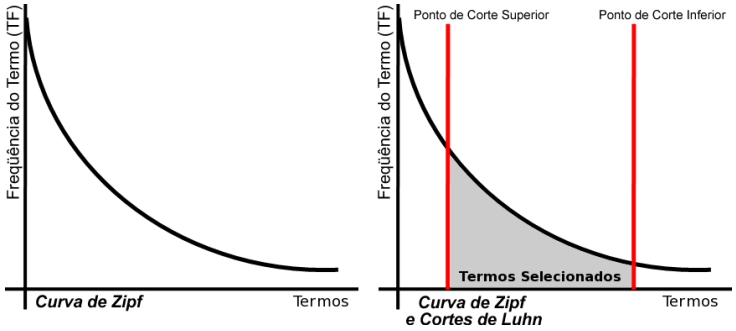

Figura 2.2: Método de Luhn para seleção de termos (adaptado de Soares et al. (2008))

a **estruturação dos documentos**, de maneira a torná-los processáveis pelos algoritmos de agrupamento que são utilizados para apoiar o aprendizado de hierarquias de tópicos. O modelo mais utilizado para representação de dados textuais é o modelo espaço-vetorial, no qual cada documento é um vetor em um espaço multi-dimensional, e cada dimensão é um termo da coleção (Feldman e Sanger, 2006). Para tal, pode-se estruturar os textos em uma *bag-of-words*, na qual os termos são considerados independentes, formando um conjunto desordenado em que a ordem de ocorrência das palavras não importa. A *bag-of-words* é uma tabela documento-termo, como ilustrado na Tabela 2.1 na qual d_i corresponde ao i-ésimo documento, t_j representa o j-ésimo termo e a_{ij} é um valor que relaciona o i-ésimo documento com o j-ésimo termo. Observe que nessa representação não há informação de classe, uma vez que a tarefa de aprendizado com métodos de agrupamento é não supervisionada.

	t_1	t_2	$\ldots$	t_M
d_1	a_{11}	a_{12}	$\ldots$	a_{1M}
d_2	a_{21}	a_{22}	$\ldots$	a_{2M}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
d_N	a_{N1}	a_{N2}	$\ldots$	a_{NM}

Tabela 2.1: Tabela documento-termo: representação da matriz atributo×valor

Por meio da tabela documento-termo, cada documento pode ser representado como um vetor $\vec{d_i} = (a_{i1}, a_{i2}, \ldots, a_{iM})$. Geralmente, o valor da medida a_{ij} é obtido de duas formas:

- um valor que indica se um determinado termo está presente ou não em um dado documento; e
- um valor que indica a importância ou distribuição do termo ao longo da coleção de documentos, por exemplo, o valor de TF. Outras formas, baseadas em critérios de ponderação e normalização, podem ser encontradas em Salton e Buckley (1988) e Liu et al. (2005). Entre elas, destaca-se o critério TF-IDF (*Term Frequency Inverse*

Document Frequency), que leva em consideração tanto o valor de TF quanto o valor de DF (Salton et al., 1996).

A representação por meio da tabela documento-termo permite o emprego de um grande leque de algoritmos de agrupamento de documentos, além de outras técnicas de extração de conhecimento. Deve-se ressaltar que essa etapa de Pré-Processamento pode ser redefinida e então repetida após as próximas etapas, uma vez que a descoberta de alguns padrões pode levar a estabelecer melhorias a serem empregadas sobre a tabela documento-termo, como, ponderar a importância de cada termo ou até mesmo refinar a seleção dos termos (Rezende et al., 2003).

2.3 Extração de Padrões usando Agrupamento de Documentos

Após a identificação/delimitação do problema e representação dos textos, o processo avança para a etapa de extração de padrões usando agrupamento de documentos.

Em tarefas de agrupamento, o objetivo é organizar um conjunto de objetos em grupos, baseado em uma medida de proximidade, na qual objetos de um mesmo grupo são altamente similares entre si, mas dissimilares em relação aos objetos de outros grupos (Everitt et al., 2001). Em outras palavras, o agrupamento é baseado no princípio de maximizar a similaridade interna dos grupos (intragrupo) e minimizar a similaridade entre os grupos (intergrupos) (Everitt et al., 2001). A análise de agrupamento também é conhecida como aprendizado por observação ou análise exploratória dos dados, pois a organização dos objetos em grupos é realizada apenas pela observação de regularidades nos dados, sem uso de conhecimento externo (Xu e Wunsch, 2008). Assim, ao contrário de métodos supervisionados, como algoritmos de classificação, em processos de agrupamento não há classes ou rótulos predefinidos para treinamento de um modelo, ou seja, o aprendizado é realizado de forma não supervisionada (Jain et al., 1999; Han e Kamber, 2006).

O processo de agrupamento depende de dois fatores principais: (1) uma medida de proximidade e (2) uma estratégia de agrupamento. As **medidas de proximidade** determinam como a similaridade entre dois objetos é calculada. Sua escolha influencia a forma como os grupos são obtidos e depende dos tipos de variáveis ou atributos que representam os objetos. Existe uma variedade de medidas de proximidade e as principais adotadas em dados textuais são discutidas na Seção 2.3.1. As **estratégias de agrupamento** são os métodos e algoritmos para definição dos grupos (Seção 2.3.2). Em geral, pode-se classificar os algoritmos de agrupamento em métodos particionais e métodos hierárquicos. Por fim, ainda na etapa de extração de padrões, é importante encontrar descritores que indicam o significado do agrupamento obtido para os usuários. Os conjuntos de descritores

de cada grupo formam os possíveis tópicos na coleção. Os principais métodos de seleção de descritores são descritos na Seção 2.3.3.

2.3.1 Medidas de Proximidade

A escolha da medida de proximidade para calcular o quão similar são dois objetos é fundamental para a análise de agrupamentos. Essa escolha depende das características do conjunto de dados, principalmente dos tipos e escala dos dados. Assim, existem medidas de proximidade para dados contínuos, discretos e mistura entre dados contínuos e discretos. As medidas de proximidade podem calcular tanto a similaridade quanto dissimilaridade (ou distância) entre objetos. No entanto, as medidas de similaridades podem ser, geralmente, convertidas para medidas de dissimilaridade, e vice-versa.

A seguir, serão descritas duas medidas de similaridade comumente utilizadas em dados textuais: Cosseno e Jaccard. Para tal, considere dois documentos $x_i = (x_{i1}, x_{i2}, ..., x_{im})$ e $x_j = (x_{j1}, x_{j2}, ..., x_{jm})$, representados no espaço vetorial m-dimensional, no qual cada termo da coleção representa uma dessas dimensões.

A medida de similaridade **Cosseno** é definida de acordo com ângulo cosseno formado entre os vetores de dois documentos, conforme a Equação 2.1 (Tan et al., 2005; Feldman e Sanger, 2006).

$$cosseno(x_i, x_j) = \frac{x_i \bullet x_j}{\mid x_i \mid\mid x_j \mid} = \frac{\sum_{l=1}^{m} x_{il} x_{jl}}{\sqrt{\sum_{l=1}^{m} {x_{il}}^2} \sqrt{\sum_{l=1}^{m} {x_{jl}}^2}} \tag{2.1}$$

O valor da medida está no intervalo $[0, 1]$ quando aplicada em dados textuais[1]. Assim, se o valor da medida de similaridade Cosseno é 0, o ângulo entre x_i e x_j é 90°, ou seja, os documentos não compartilham nenhum termo. Por outro lado, se o valor da similaridade for próximo de 1, o ângulo entre x_i e x_j é próximo de 0°, indicando que os documentos compartilham termos e são similares. É importante observar que essa medida não considera a magnitude dos dados para computar a proximidade entre documentos.

Em algumas situações os vetores são representados por valores binários, ou seja, indicam a presença ou ausência de algum termo. O cálculo da proximidade entre dois documentos representados por vetores binários pode ser realizado pela medida **Jaccard**. Seja x_i e x_j dois documentos, a medida Jaccard pode ser derivada a partir das seguintes contagens:

- f_{11} = número de termos presentes em ambos documentos;

[1] A medida de cosseno pode variar no intervalo $[-1, 1]$ quando são utilizados valores negativos na representação dos atributos. Para dados textuais, geralmente utiliza-se valores baseados em frequência que são maiores ou igual a zero.

- f_{01} = número de termos ausentes em x_i e presentes em x_j; e
- f_{10} = número de termos presentes em x_i e ausentes x_j.

A partir das contagens, a medida Jaccard é definida na Equação 2.2 (Tan et al., 2005; Feldman e Sanger, 2006).

$$jaccard(x_i, x_j) = \frac{f_{11}}{f_{01} + f_{10} + f_{11}} \tag{2.2}$$

O valor da medida Jaccard fica no intervalo $[0, 1]$. Quanto mais próximo de 1 maior a similaridade entre os dois documentos.

Pode-se observar que as medidas Cosseno e Jaccard são medidas de similaridade. Conforme comentado anteriormente, as medidas de similaridade podem ser transformadas em medidas de dissimilaridade (ou distância). A Equação 2.3 e a Equação 2.4 definem medidas de dissimilaridade baseadas no Cosseno e Jaccard, respectivamente.

$$d_{cos}(x_i, x_j) = 1 - cosseno(x_i, x_j) \tag{2.3}$$

$$d_{jac}(x_i, x_j) = 1 - jaccard(x_i, x_j) \tag{2.4}$$

A literatura apresenta uma variedade de medidas de proximidades. Nessa seção, foram apresentadas duas medidas relacionadas ao contexto deste projeto. Uma revisão mais extensa está disponível nos trabalhos de Everitt et al. (2001) e Tan et al. (2005).

Uma dúvida pertinente que surge com relação às várias medidas de proximidade é qual escolher no processo de agrupamento. Não existe uma regra geral para essa escolha. Geralmente, essa decisão é baseada de acordo com a natureza dos dados a serem analisados, acompanhada de um processo de validação da qualidade do agrupamento obtido.

2.3.2 Métodos de Agrupamento

Após a escolha de uma medida de proximidade para os documentos, é selecionado um método para o agrupamento. Os métodos de agrupamento podem ser classificados considerando diferentes aspectos. Jain et al. (1999) organizam os métodos de agrupamento de acordo com a estratégia adotada para definir os grupos. Uma análise de diferentes métodos de agrupamento considerando o cenário de Mineração de Dados é apresentada em Berkhin (2006).

Em geral, as estratégias de agrupamento podem ser organizadas em dois tipos: agrupamento particional e agrupamento hierárquico. No **agrupamento particional** a coleção de documentos é dividida em uma partição simples de k grupos, enquanto no **agrupamento hierárquico** é produzido uma sequência de partições aninhadas, ou seja, a

coleção textual é organizada em grupos e subgrupos de documentos (Feldman e Sanger, 2006). Além disso, o agrupamento obtido pode conter sobreposição, isto é, quando um documento pertence a mais de um grupo ou, até mesmo, quando cada documento possui um grau de pertinência associado aos grupos. No contexto deste trabalho, são exploradas as estratégias que produzem agrupamento sem sobreposição, também conhecidas como estratégias rígidas ou *crisp* (Everitt et al., 2001). Assim, se o conjunto $X = \{x_1, x_2, ..., x_n\}$ representa uma coleção de n documentos, uma partição rígida $P = \{G_1, G_2, ..., G_k\}$ com k grupos não sobrepostos é tal que:

- $G_1 \cup G_2 \cup ... \cup G_k = X$;
- $G_i \neq \emptyset$ para todo $i \in \{1, 2, ..., k\}$; e
- $G_i \cap G_j = \emptyset$ para todo $i \neq j$.

As diversas estratégias de agrupamento são, na prática, algoritmos que buscam uma solução aproximada para o problema de agrupamento. Para exemplificar, um algoritmo de força bruta que busca a melhor partição de um conjunto de n documentos em k grupos, precisa avaliar $k^n/k!$ possíveis partições (Liu, 1968). Enumerar e avaliar todas as possíveis partições é inviável computacionalmente. A seguir, são descritos alguns dos principais algoritmos que são utilizados para agrupamento de documentos.

Agrupamento Particional

O agrupamento particional também é conhecido como agrupamento por otimização. O objetivo é dividir iterativamente o conjunto de objetos em k grupos, na qual k geralmente é um valor informado previamente pelo usuário. Os grupos de documentos são formados visando otimizar a compactação e/ou separação do agrupamento.

O algoritmo *k-means* (MacQueen, 1967) é o representante mais conhecido para agrupamento particional e muito utilizado em coleções textuais (Steinbach et al., 2000). No *k-means* utiliza-se um representante de grupo denominado centroide, que é simplesmente um vetor médio computado a partir dos demais vetores do grupo. A Equação 2.5 define o cálculo do centroide C para um determinado grupo G, em que x representa um documento pertencente a G e o número total de documentos no grupo é $|G|$.

$$C = \frac{1}{|G|} \sum_{x \in G} x \tag{2.5}$$

Dessa forma, o centroide mantém um conjunto de características centrais do grupo, permitindo representar todos os documentos que pertencem a este grupo. Ainda, é importante observar que o *k-means* só é aplicável em situações na qual a média possa ser calculada.

O pseudocódigo para o *k-means*, contextualizado para agrupamento de documentos, está descrito no Algoritmo 1.

Algoritmo 1: O algoritmo k-means

Entrada:

$X = \{x_1, x_2, ..., x_n\}$: conjunto de documentos

k: número de grupos

Saída:

$P = \{G_1, G_2, ..., G_k\}$: partição com k grupos

```
1 selecionar aleatoriamente k documentos como centroides iniciais;
2 repita
3     para cada documento x ∈ X faça
4         computar a (dis)similaridade de x para cada centroide C ;
5         atribuir x ao centroide mais próximo ;
6     fim
7     recomputar o centroide de cada grupo;
8 até atingir um critério de parada;
```

O critério de parada do *k-means* é dado quando não ocorre mais alterações no agrupamento, ou seja, a solução converge para uma determinada partição. Outro critério de parada pode ser um número máximo de iterações.

Durante as iterações do *k-means*, o objetivo é minimizar uma função de erro E, definida na Equação 2.6, em que x é um documento da coleção; e C_i é o centroide do grupo G_i. Observe que é utilizado uma medida de dissimilaridade $dis(x, C_i)$ para calcular o valor da função de erro E.

$$E = \sum_{i=1}^{k} \sum_{x \in G_i} \mid dis(x, C_i) \mid^2 \tag{2.6}$$

Ao minimizar este critério, o *k-means* tenta separar o conjunto de documentos diminuindo a variabilidade interna de cada grupo e, consequentemente, aumentando a separação entre os grupos.

A complexidade do *k-means* é linear em relação ao número de objetos, o que possibilita sua aplicação eficiente em diversos cenários. No entanto, a necessidade de informar com antecedência o número de grupos pode ser vista como uma desvantagem, pois esse valor geralmente é desconhecido pelos usuários. Além disso, o método apresenta variabilidade nos resultados, pois a seleção dos centroides iniciais afeta o resultado do agrupamento.

Para minimizar esse efeito, o algoritmo é executado diversas vezes, com várias inicializações diferentes, e a solução que apresenta menor valor de erro E é selecionada.

Agrupamento Hierárquico

Os algoritmos de agrupamento hierárquico podem ser aglomerativos ou divisivos. No **agrupamento hierárquico aglomerativo**, inicialmente cada documento pertence a um grupo e, em cada iteração, os pares de grupos mais próximos são unidos até se formar um único grupo (Feldman e Sanger, 2006). Já no **agrupamento hierárquico divisivo**, inicia-se com um grupo contendo todos os documentos que é, então, dividido em grupos menores até restarem grupos unitários (grupo com apenas um documento) (Steinbach et al., 2000; Zhao et al., 2005).

Tanto os métodos aglomerativos quanto os divisivos organizam os resultados do agrupamento em uma árvore binária conhecida como dendrograma (Figura 2.3). Essa representação é uma forma intuitiva de visualizar e descrever a sequência do agrupamento. Cada nó do dendrograma representa um grupo de documentos. A altura dos arcos que unem dois subgrupos indica o grau de compactação do grupo formado por eles. Quanto menor a altura, mais compactos são os grupos. No entanto, também espera-se que os grupos formados sejam distantes entre si, ou seja, que a proximidade de objetos em grupos distintos seja a menor possível. Essa característica é representada quando existe uma grande diferença entre a altura de um arco e os arcos formados abaixo dele (Metz, 2006).

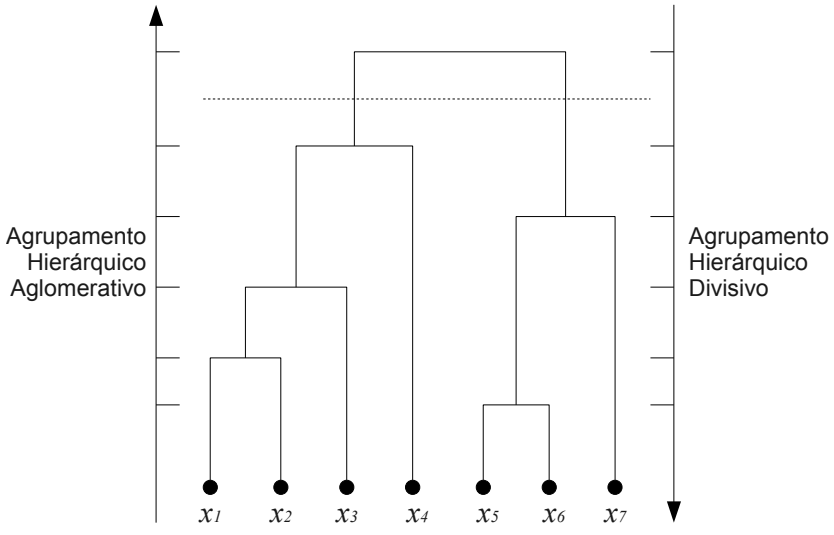

Figura 2.3: Exemplo de um dendrograma (adaptado de Xu e Wunsch (2008))

A partir do dendrograma também é possível obter uma partição com um determinado número de grupos, como nos métodos particionais. Por exemplo, a linha tracejada

na Figura 2.3 indica uma partição com dois grupos de documentos: $\{x_1, x_2, x_3, x_4\}$ e $\{x_5, x_6, x_7\}$.

O pseudocódigo para um algoritmo típico de agrupamento hierárquico aglomerativo está descrito no Algoritmo 2.

Algoritmo 2: Agrupamento hierárquico aglomerativo

Entrada:

$X = \{x_1, x_2, ..., x_n\}$: conjunto de documentos

Saída:

$S = \{P_1, ..., P_k\}$: lista de agrupamentos formados

```
1 fazer com que cada documento x ∈ X seja um grupo;
2 computar a dissimilaridade entre todos os pares distintos de grupos;
3 repita
4 |   selecionar o par de grupos mais próximo;
5 |   unir os dois grupos para formar um novo grupo G;
6 |   computar a dissimilaridade entre G e os outros grupos;
7 até obter um único grupo com todos os documentos;
```

A diferença principal entre os algoritmos de agrupamento hierárquico aglomerativo está no critério de seleção do par de grupos mais próximo (Linha 4 do Algoritmo 2). Os três critérios mais conhecidos são:

- *Single-Link* (Everitt et al., 2001; Sneath, 1957): utiliza o critério de vizinho mais próximo, no qual a distância entre dois grupos é determinada pela distância do par de documentos mais próximos, sendo cada documento pertencente a um desses grupos. Esse método de união de grupos apresenta um problema conhecido como "efeito da corrente", em que ocorre a união indevida de grupos influenciada pela presença de ruídos na base de dados (Larose, 2004);

- *Complete-Link* (Everitt et al., 2001; Sorensen, 1948): utiliza o critério de vizinho mais distante, ao contrário do algoritmo *Single-Link*, e a distância entre dois grupos é maior distância entre um par de documentos, sendo cada documento pertencente a um grupo distinto. Esse método dificulta a formação do efeito da corrente, como ocorre no *Single-Link*, e tende a formar grupos mais compactos e em formatos esféricos (Larose, 2004); e

- *Average-Link* (Everitt et al., 2001; Sokal e Michener, 1958): a distância entre dois grupos é definida como a média das distâncias entre todos os pares de documentos

em cada grupo, cada par composto por um documento de cada grupo. Esse método elimina muitos problemas relacionados à dependência do tamanho dos grupos, mantendo próxima a variabilidade interna entre eles (Larose, 2004).

A escolha do critério de união de grupos dos algoritmos aglomerativos depende geralmente do conjunto de dados e dos objetivos da aplicação. Por exemplo, em dados textuais, avaliações experimentais têm mostrado o *Average-Link* como a melhor opção entre os algoritmos que adotam estratégias aglomerativas (Zhao e Karypis, 2002).

A maioria dos trabalhos relacionados com agrupamento hierárquico na literatura referenciam as estratégias aglomerativas, mostrando pouco interesse nas estratégias divisivas. A possível causa é a complexidade das estratégias divisivas, que cresce exponencialmente em relação ao tamanho do conjunto de dados, proibindo sua aplicação em conjuntos de dados grandes (Kaufman e Rousseeuw, 1990). Para lidar com esse problema, Steinbach et al. (2000) propuseram o algoritmo *Bisecting k-means*, que basicamente utiliza agrupamento particional baseado no *k-means* sucessivamente, possibilitando sua aplicação em conjuntos de dados maiores, inclusive em coleções textuais. O pseudocódigo do *Bisecting k-means* está ilustrado no Algoritmo 3.

Algoritmo 3: Bisecting k-means

Entrada:

$X = \{x_1, x_2, ..., x_n\}$: conjunto de documentos

Saída:

$S = \{P_1, ..., P_k\}$: lista de agrupamentos formados

1 formar um agrupamento contendo todos os documentos de X;
2 **repita**
3 | selecionar o próximo grupo (nó folha) a ser dividido;
4 | dividir este grupo em dois novos subgrupos *(k-means com k=2)*;
5 **até** *obter apenas grupos unitários*;

Existem diferentes maneiras para seleção do próximo grupo a ser dividido (Linha 3 do Algoritmo 3). Um critério simples e eficaz é selecionar o maior grupo (de acordo com o número de documentos) ainda não dividido em uma iteração anterior (Zhao e Karypis, 2002). Uma propriedade interessante do *Bisecting k-means* é o fato ser menos sensível à escolha inicial dos centroides quando comparado com o *k-means* (Tan et al., 2005).

Os algoritmos de agrupamento hierárquico aglomerativos e divisivos apresentam complexidade quadrática de tempo e espaço, em relação ao número de documentos. Avaliações

experimentais indicam que o *Bisecting k-means* obtém melhores resultados em coleções de documentos, seguido do agrupamento hierárquico aglomerativo com o critério *Average-Link* (Zhao e Karypis, 2002; Zhao et al., 2005).

Outros Métodos de Agrupamento

Até o momento, foram apresentados os algoritmos de agrupamento mais conhecidos e de interesse ao trabalho proposto. No entanto, é importante observar que a literatura na área de análise de agrupamento de dados apresenta uma vasta contribuição (Xu e Wunsch, 2008). A seguir, é realizado uma breve descrição de outros métodos de agrupamento existentes.

- **Agrupamento baseado em densidade:** esses métodos assumem que os grupos são regiões de alta densidade separados por regiões de baixa densidade no espaço dimensional formado pelos atributos dos objetos. A ideia básica é que cada objeto possui um número mínimo de vizinhos dentro de uma esfera com raio pré-definido pelo usuário. Se a esfera contém um número mínimo de objetos, então é considerada uma região com densidade e utilizada para formação do agrupamento. O algoritmo *DBSCAN* (Ester et al., 1996) é um exemplo de algoritmo de agrupamento baseado em densidade.

- **Agrupamento baseado em grade:** o diferencial desses métodos é o uso de uma grade para construir um novo espaço aos objetos de forma que todas as operações de agrupamento sejam realizadas em termos do espaço da grade. É uma abordagem eficiente para grandes conjuntos de dados, alta dimensionalidade e para detecção de ruídos. Um algoritmo que realiza o agrupamento baseado em grade é o *CLIQUE* (Agrawal et al., 1998).

- **Agrupamento com sobreposição:** os algoritmos mencionados no decorrer deste capítulo obtém grupos exclusivos, ou seja, cada objeto pertence exclusivamente a um único grupo. Os métodos que permitem sobreposição podem associar os objetos a um ou mais grupos. Essa sobreposição pode ser simples, na qual um objeto está em um ou mais grupos ou, ainda, pertencer a todos os grupos com um grau de pertinência/probabilidade. Os algoritmos de agrupamento *fuzzy* e probabilísticos associam níveis de pertinência ou probabilidade dos objetos aos grupos encontrados (Witten e Frank, 2005).

- **Agrupamento baseado em redes auto-organizáveis:** também conhecidos como redes *SOM - Self Organizing Map* (Kohonen, 1982), esses métodos utilizam o conceito de redes neurais para realizar o agrupamento dos dados. A ideia básica é

organizar um conjunto de neurônios em um reticulado bidimensional, na qual cada neurônio fica conectado em todas as entradas da rede. Conforme os objetos são apresentados à rede, os neurônios atualizam seus pesos de ligação da rede, ativando uma região diferente do reticulado. No final do processo, cada região do reticulado representa um grupo de objetos. O algoritmo *SomPak* (Kohonen et al., 1996) é um exemplo de agrupamento baseado em redes SOM.

2.3.3 Seleção de Descritores para Agrupamento

Uma vez obtido o agrupamento (particional ou hierárquico) de documentos, deve-se selecionar descritores que auxiliam a interpretação dos resultados. Essa é uma tarefa importante, pois o agrupamento geralmente é utilizado em atividades exploratórias para descoberta de conhecimento e, assim, é necessário indicar o significado de cada grupo para que usuários e/ou aplicações possam interagir com o agrupamento de forma mais intuitiva (Manning et al., 2008).

Conforme comentado anteriormente, o centroide mantém o conjunto de características centrais do grupo, permitindo representar todos os documentos pertencentes a este grupo. Por este motivo, algumas técnicas utilizam o centroide como ponto de partida para seleção dos descritores de um grupo. Uma estratégia simplista é selecionar os termos mais frequentes de um grupo, porém, a literatura indica que os resultados obtidos por essa estratégia não são satisfatórios (Chuang e Chien, 2004). Outra estratégia é selecionar os termos dos j documentos mais próximos ao centroide como descritores (Cutting et al., 1992). Manning et al. (2008) discute que as técnicas existentes de seleção de atributos em tarefas de aprendizado de máquina podem ser aplicadas na seleção de descritores de agrupamento. Assim, é possível obter um *ranking* dos termos que melhor discriminam um determinado grupo. Abaixo, é descrito uma abordagem genérica para obtenção deste *ranking*.

Seja um grupo G e seu respectivo centroide C. O conjunto de termos que compõem o centroide C é identificado como T, ou seja, os termos que representam as dimensões no espaço vetorial. A ideia básica é obter uma lista ordenada (*ranking*) dos termos contidos em T e selecionar os melhores j termos como descritores do grupo G. Diversos critérios podem ser utilizados para construção do *ranking*, e esses critérios podem ser derivados a partir de uma tabela de contingência.

Assim, para cada termo $t \in T$, realiza-se uma expressão de busca $Q(t)$ sobre toda a coleção de documentos $X = \{x_1, x_2, ..., x_n\}$, recuperando-se um subconjunto de documentos que contêm o termo t. Com o conjunto de documentos recuperados $Q(t)$ e o conjunto de documentos do grupo G, é construído uma tabela de contingência do termo t, conforme ilustrado na Figura 2.4 (adaptado de (Chu, 2003))

Q(t) \ G	Relevante	Não Relevante
Relevante	*acertos*	*ruído*
Não Relevante	*perda*	*rejeitos*

Figura 2.4: Tabela de contingência com os possíveis resultados de recuperação por meio da expressão de busca Q(t).

Os itens da tabela de contingência são calculados da seguinte forma (Chu, 2003):

- *acertos*: número de documentos recuperados por $Q(t)$ que pertencem a G;
- *perda*: número de documentos em G que não foram recuperados por $Q(t)$;
- *ruído*: número de documentos recuperados por $Q(t)$ que não pertencem a G; e
- *rejeitos*: número de documentos que não pertencem a G e que também não foram recuperados por $Q(t)$.

A partir desses itens, pode-se derivar diversos critérios para avaliar o poder discriminativo do termo t para o grupo G. Um desses critérios é o F-Measure (F_1), descrito na Equação 2.9, obtido por uma média harmônica entre precisão (Equação 2.7) e revocação (Equação 2.8).

$$Precisao(t) = \frac{acertos}{acertos + ruido} \tag{2.7}$$

$$Revocacao(t) = \frac{acertos}{acertos + perda} \tag{2.8}$$

$$F - Measure(t) = \frac{2 * Precisao(t) * Revocacao(t)}{Precisao(t) + Revocacao(t)} \tag{2.9}$$

O valor de F-Measure varia no intervalo $[0, 1]$, e quanto mais próximo de 1, melhor o poder discriminativo de t. O processo é aplicado para todos os termos $t \in T$, e um *ranking* é obtido em ordem decrescente da F-Measure, por exemplo. Dessa forma, os descritores do grupo G são formados pelos melhores j termos. Com o conjunto de descritores e seu respectivo grupo de documentos forma-se um tópico na coleção. Uma lista de critérios que podem ser derivadas a partir dos itens da tabela de contingência pode ser encontrada no trabalho de Forman (2003).

A seleção de descritores ilustrada aqui é aplicável em agrupamentos particionais e hierárquicos. No entanto, o agrupamento hierárquico tem certas particularidades, pois

documentos de um grupo (filho) também estão presentes em seu grupo superior (grupo pai). Nesse sentido, pode-se refinar a seleção dos termos considerando a estrutura hierárquica (Moura et al., 2008; Moura e Rezende, 2010).

2.4 Pós-processamento

A etapa de pós-processamento é responsável pela validação do conhecimento extraído. A avaliação pode ser realizada de forma **subjetiva**, utilizando um conhecimento de um especialista de domínio, ou de forma **objetiva** por meio de índices estatísticos que indicam a qualidade dos resultados. Nesta seção, serão abordados alguns desses índices para validação objetiva.

No contexto deste trabalho, a qualidade da hierarquia de tópicos está diretamente relacionada com a qualidade do agrupamento na extração de padrões. Assim, a validação do conhecimento extraído é realizada por meio de índices utilizados na análise de agrupamentos.

A validação do resultado de um agrupamento, em geral, é realizada por meio de índices estatísticos que expressam o "mérito" das estruturas encontradas, ou seja, quantifica alguma informação sobre a qualidade de um agrupamento (Faceli et al., 2005; Xu e Wunsch, 2008). O uso de técnicas de validação em resultados de agrupamento é uma atividade importante, uma vez que algoritmos de agrupamento sempre encontram grupos nos dados, independentemente de serem reais ou não (Halkidi et al., 2001).

Em geral, existem três tipos de critérios para realizar a validação de um agrupamento: critérios internos, relativos e externos (Everitt et al., 2001).

Os **critérios internos** obtêm a qualidade de um agrupamento a partir de informações do próprio conjunto de dados. Geralmente, um critério interno analisa se as posições dos objetos em um agrupamento obtido corresponde a matriz de proximidades. Já os **critérios relativos** comparam diversos agrupamentos para decidir qual deles é o mais adequado aos dados. Finalmente, os **critérios externos** avaliam um agrupamento de acordo com uma informação externa, geralmente uma intuição do pesquisador sobre a estrutura presente nos dados ou um agrupamento construído por um especialista de domínio. Por exemplo, um critério externo pode medir se o agrupamento obtido corresponde com uma partição dos dados já agrupados manualmente.

Alguns trabalhos na literatura descrevem e comparam técnicas e índices de validação. No trabalho de Milligan e Cooper (1985), trinta índices de validação são comparados na tarefa de estimar o número de grupos em conjuntos de dados. Uma avaliação similar é realizada por Vendramin et al. (2010), com uma comparação de diversos índices de validade relativa de agrupamento. Uma revisão geral de diversas abordagens para validação

de agrupamento é encontrada em (Jain e Dubes, 1988; Halkidi et al., 2001; Xu e Wunsch, 2008).

A seguir, serão discutidos três diferentes índices de validação, selecionados para o desenvolvimento deste trabalho de mestrado, que avaliam a qualidade de agrupamento sob diferentes perspectivas. Foram selecionados os índices Silhueta, Entropia e FScore, aqui discutidos conforme descrito no livro *Introduction to Data Mining* (Tan et al., 2005), Capítulo 8 - *Cluster Analysis: Basic Concepts and Algorithms*, Seção 8.5 - *Cluster Evaluation*.

2.4.1 Silhueta

A Silhueta (Kaufman e Rousseeuw, 1990; Tan et al., 2005) é um índice de critério relativo utilizado para avaliar partições. Experimentos recentes comparando vários índices de validade relativa indicaram que a Silhueta é um dos índices de validade mais eficazes (Vendramin et al., 2010).

A medida de Silhueta verifica o quão bem os documentos estão situados dentro de seus grupos. Dada uma coleção de n documentos $X = \{x_1, x_2, ..., x_n\}$, o valor de Silhueta $s(x_i)$ do documento x_i é obtido pela Equação 2.10.

$$s(x_i) = \frac{b(x_i) - a(x_i)}{max\{a(x_i), b(x_i)\}} \tag{2.10}$$

no qual, $a(x_i)$ é a dissimilaridade média entre x_i e todos os documentos de seu grupo; e $b(x_i)$ a dissimilaridade média entre x_i e todos os documentos do seu grupo vizinho.

O valor do índice de Silhueta fica no intervalo $[-1, 1]$, em que valores positivos indicam que o documento está bem alocado no seu grupo e valores negativos indicam que o documento possivelmente está erroneamente agrupado. Com os valores de Silhueta de cada documento, calcula-se o valor global de Silhueta de uma partição (S_P), por meio da média das Silhuetas, conforme ilustrado na Equação 2.11.

$$S_P = \frac{\sum_{i=1}^{n} s(x_i)}{n} \tag{2.11}$$

O índice de Silhueta permite comparar partições obtidas por diferentes algoritmos de agrupamento e diferentes números de grupos. Assim, fixado um conjunto de dados, a Silhueta é uma medida útil para determinar o melhor número de grupos, ou seja, auxilia a estimação de parâmetros de algoritmos de agrupamento.

A complexidade de tempo para computar o índice é $O(n^2)$, uma vez que é necessário computar dissimilaridades entre todos os documentos. Uma versão simplificada da Silhueta, baseada em centroides, proporciona resultados competitivos com custo computacional reduzido para $O(n)$ (Vendramin et al., 2010).

2.4.2 Entropia

O índice de Entropia (Steinbach et al., 2000; Tan et al., 2005) é empregado como um critério externo de validação, ou seja, utiliza um conhecimento prévio (informação externa) a respeito das categorias ou tópicos dos documentos. A ideia é medir a desordem no interior de cada grupo. Assim, quanto menor o valor de Entropia melhor a qualidade do agrupamento. Um grupo com valor 0 (zero) de Entropia, que é a solução ideal, contém todos os documentos de um mesmo tipo de categoria ou tópico.

Para computar este índice considere que

- P é uma partição obtida por um determinado algoritmo de agrupamento;
- L_r é uma determinada categoria (informação externa) representando um conjunto de documentos de um mesmo tópico ou classe; e
- G_i é um determinado grupo, e seu respectivo conjunto de documentos, pertencente à partição P.

Assim, a Entropia E de um determinado grupo G_i em relação a uma coleção textual com c categorias é calculada conforme a Equação 2.12.

$$E(G_i) = -\sum_{r=1}^{c}(\frac{|L_r \cap G_i|}{|G_i|})log(\frac{|L_r \cap G_i|}{|G_i|}) \tag{2.12}$$

Na Equação 2.12, $|L_r \cap G_i|$ representa o número de documentos do grupo G_i pertencentes à categoria L_r.

O valor de Entropia global do agrupamento (partição P) é calculado como a soma das entropias de cada grupo ponderada pelo tamanho de cada grupo (Equação 2.13).

$$Entropia(P) = \sum_{i=1}^{k} \frac{|G_i| * E(G_i)}{n} \tag{2.13}$$

Na Equação 2.13, k é o número de grupos na partição P e n é o número de documentos da coleção.

2.4.3 FScore

O índice FScore é uma medida que utiliza as ideias de *precisão* e *revocação*, da recuperação de informação, para avaliar a eficácia de recuperação em agrupamentos hierárquico de documentos (Larsen e Aone, 1999; Zhao e Karypis, 2002; Tan et al., 2005). É empregado como critério externo de validação, pois utiliza o conhecimento prévio (informação

externa) sobre categorias ou tópicos existentes no conjunto de dados. A ideia básica é verificar o quanto o agrupamento hierárquico conseguiu reconstruir a informação de categoria associada a cada documento.

Para o cálculo do índice FScore, considere que

- H é um agrupamento hierárquico obtido (dendrograma) por um determinado algoritmo;
- L_r é uma determinada categoria (informação externa) representando um conjunto de documentos de um mesmo tópico ou classe; e
- G_i é um determinado grupo, e seu respectivo conjunto de documentos, pertencente ao agrupamento hierárquico H.

Assim, dada uma categoria L_r e um grupo G_i, calcula-se as medidas de precisão P e revocação R conforme a Equação 2.14 e Equação 2.15, respectivamente. Em seguida, é obtida a média harmônica F (Equação 2.16), que representa um balanceamento entre a precisão e revocação.

$$P(L_r, G_i) = \frac{|L_r \cap G_i|}{|G_i|} \tag{2.14}$$

$$R(L_r, G_i) = \frac{|L_r \cap G_i|}{|L_i|} \tag{2.15}$$

$$F(L_r, G_i) = \frac{2 * P(L_r, G_i) * R(L_r, G_i)}{P(L_r, G_i) + R(L_r, G_i)} \tag{2.16}$$

A medida F selecionada para uma determinada categoria L_r é o maior valor obtido por algum grupo da hierarquia H, considerando todas os grupos e subgrupos existentes, conforme a Equação 2.17.

$$F(L_r) = \max_{G_i \in H} F(L_r, G_i) \tag{2.17}$$

Finalmente, o valor FScore global de um agrupamento hierárquico com n documentos e c categorias, é calculado como o somatório da medida F de cada categoria ponderada pelo número de documentos da categoria (Equação 2.18).

$$FScore = \sum_{r=1}^{c} \frac{|L_r|}{n} F(L_r) \tag{2.18}$$

Conforme o agrupamento hierárquico consegue reconstruir a informação das categorias predeterminadas de uma coleção, o valor de FScore se aproxima de 1. Caso contrário, a FScore tem valor 0. Observe que essa medida trata cada grupo da hierarquia como se

fosse o resultado de uma consulta e cada categoria predefinida da coleção como o conjunto de documentos relevantes para essa consulta.

2.5 Uso do Conhecimento

Na etapa de uso do conhecimento, os resultados estão validados e aptos a serem utilizados para apoiar algum processo de tomada de decisão, de acordo com os objetivos estabelecidos na etapa de Identificação do Problema.

Entre as aplicações que se beneficiam das hierarquias de tópicos, destacam-se os trabalhos envolvendo bibliotecas digitais (Krowne e Halbert, 2005; Marcacini et al., 2007; Zhang e Wu, 2008), sistemas de gestão de conhecimento (Bedford, 2008; Zhong e Liu, 2010), *web mining* (Song, 2009) e sistemas de recuperação de informação (Zeng et al., 2004; Carpineto et al., 2009).

2.6 Considerações Finais

A extração de conhecimento a partir de bases de dados textuais tem ganhado importância para diversas aplicações, recuperando conhecimento útil dos textos para auxiliar algum processo de tomada de decisão. Um processo de mineração de textos executado com sucesso transforma os dados textuais em conhecimento, extraindo dos documentos informações novas, interpretáveis e potencialmente úteis.

Neste capítulo procurou-se cobrir o processo de mineração de textos visando o aprendizado de hierarquias de tópicos. Foram discutidos os principais aspectos envolvidos no **Pré-processamento** dos textos, como a limpeza, seleção de termos e estruturação dos textos em um formato adequado para o aprendizado. Conforme mencionado, a etapa de **Extração de Padrões** define como o conhecimento será extraído dos textos. No caso das hierarquias de tópicos, os algoritmos de agrupamento hierárquico são utilizados nessa etapa, além de um processo de seleção de descritores para os grupos visando a formação dos tópicos. Por fim, foram apresentadas três critérios para validação agrupamento de documentos a ser executado na etapa de **Pós-processamento**.

O maior foco deste trabalho está na etapa de Extração de Padrões. Nessa etapa, foi visto que o algoritmo de agrupamento hierárquico *Bisecting k-means*, seguido do *Average-Link*, são os que obtêm melhores resultados em coleções textuais, demonstrado por diversos experimentos (Zhao e Karypis, 2002; Zhao et al., 2005). No entanto, uma limitação típica desses algoritmos é que não permitem a inclusão de novos documentos no agrupamento sem que seja necessário repetir o processo de agrupamento para toda a coleção

textual. Ainda, a complexidade quadrática de tempo e espaço dos algoritmos de agrupamento hierárquico inviabiliza seu uso em grandes coleções de textos.

Em cenários dinâmicos, a inclusão de novos documentos à coleção textual é frequente. Para incorporar este novo conhecimento sempre que disponível e sem reprocessamento redundante, uma alternativa é realizar o agrupamento de maneira incremental. Assim, no próximo capítulo é discutido as estratégias de agrupamento para cenários dinâmicos.

Capítulo

3

Agrupamento em Cenários Dinâmicos

Nos cenários atuais, a natureza dos repositórios textuais é inerentemente dinâmica. Com o passar do tempo, novos documentos de textos são publicados para registro de novo conhecimento, bem como documentos antigos podem ser descartados por não mais representar adequadamente o conhecimento atual.

Extrair conhecimento de grandes bases de textos com essa característica dinâmica é um dos desafios atuais (Kriegel et al., 2007; Jain, 2010). As abordagens tradicionais de agrupamento, não incrementais, não são eficazes nesse contexto. No agrupamento não incremental, a base de dados é considerada estática e alterações realizadas na base requerem a repetição de todo o processo de agrupamento. Esse reprocessamento é computacionalmente custoso, principalmente quando são exigidas frequentes atualizações do agrupamento (Can, 1993; Zhang et al., 1996; Xu e Wunsch, 2008). Ainda, pode ser considerado redundante pois a estrutura de agrupamento existente, em geral, não sofre modificações significativas com a adição de novos documentos (Sahoo et al., 2006). Nesse contexto, métodos de agrupamento incremental são muito úteis, pois permitem a revisão e atualização dos grupos já existentes ao invés de gerá-los sempre que um novo documento é observado (Charikar et al., 1997; Xu e Wunsch, 2008).

Em geral, há duas estratégias principais para organização hierárquica de dados em cenários dinâmicos (Nassar et al., 2004; Liu et al., 2006). A primeira estratégia envolve a especialização de um algoritmo de agrupamento para tratar diretamente a inserção de novos dados nos grupos da hierarquia. Na segunda estratégia, aplica-se uma técnica de sumarização para construir e manter, de forma incremental, uma representação condensada dos dados e, então, um algoritmo tradicional de agrupamento hierárquico é aplicado nessa representação condensada (Zhang et al., 1996; Xu e Wunsch, 2008). Uma forma eficaz para obtenção da representação condensada é por meio de algoritmos de agrupa-

mento incremental do tipo "single-pass", capazes de sumarizar os dados em uma estrutura de agrupamento com poucas iterações, geralmente em tempo linear (Bradley et al., 1998; Farnstrom et al., 2000; Xu e Wunsch, 2008; Kishida, 2010).

A segunda estratégia é mais flexível, pois permite o uso de uma diversidade de algoritmos de agrupamento hierárquico disponíveis na literatura, enquanto na primeira estratégia normalmente os algoritmos são meras adaptações para aplicações particulares (Nassar et al., 2004; Liu et al., 2006). Além disso, as representações condensadas podem ser utilizadas para outras tarefas de descoberta de conhecimento, pois mantêm um resumo com as principais características dos dados.

Dada as evidências da literatura, apoiadas por experimentações realizadas ao longo deste trabalho, durante o desenvolvimento desse projeto de mestrado buscou-se focar a organização hierárquica de coleções textuais dinâmicas por meio de métodos pertencentes à segunda estratégia. Na próxima seção é apresentada uma revisão geral dos trabalhos da literatura sobre agrupamento incremental. Em seguida, os algoritmos de agrupamento incremental utilizados nesse projeto de mestrado, para realizar as avaliações comparativas, são descritos com mais detalhes.

3.1 Trabalhos Relacionados a Agrupamento Incremental

O agrupamento incremental é baseado no pressuposto de que é possível observar um objeto de cada vez e alocá-lo a um agrupamento existente (Maimon e Rokach, 2005). De uma maneira mais formal, dada uma sequência de objetos $x_1, x_2, ..., x_n$, é possível construir um agrupamento P_{h+1}, com base apenas no agrupamento anterior P_h e no objeto atual x_i (Giraud, 2000). Ainda, os algoritmos de agrupamento incremental não devem armazenar todos os objetos em memória, mas apenas os representantes de cada grupo existente (Xu e Wunsch, 2008). Em geral, os requisitos de tempo e espaço dos algoritmos de agrupamento incremental são pequenos (Maimon e Rokach, 2005), o que os tornam úteis para extração de conhecimento em grandes bases de dados e em cenários dinâmicos. A seguir, é apresentada uma breve revisão dos principais métodos de agrupamento incremental encontrados na literatura.

O algoritmo *Leader-Follower* (Hartigan, 1975; Crouch, 1975; Jain et al., 1999), aqui chamado de Leader, é um do métodos de agrupamento incremental mais simples e comumente utilizado (Xu e Wunsch, 2008). No Leader, para cada novo objeto apresentado compara-se sua proximidade com os grupos existentes. Se o valor da proximidade respeitar um limiar definido pelo usuário, então o objeto é alocado ao grupo existente mais próximo. Caso contrário, um novo grupo é criado para o objeto. Em geral, usa-se o centroide dos grupos para o cálculo da proximidade com objetos. Ainda, quando um objeto

é alocado a um grupo existente, o respectivo centroide é ajustado. É importante observar que o ajuste dos centroides é realizado de forma eficiente com equações específicas para cálculo incremental de médias (Finch, 2009).

Em Cutting et al. (1992), é apresentada uma estratégia de agrupamento incremental para organização de textos. Entre os algoritmos propostos no trabalho, destaca-se o algoritmo Buckshot. Sua principal característica é o uso de uma amostragem aleatória para obtenção de um subconjunto de documentos. O tamanho desse subconjunto deve ser escolhido conforme a memória disponível para a aplicação. O subconjunto de documentos é utilizado para formação de um agrupamento inicial. Os documentos restantes e novos documentos são apresentados ao agrupamento existente e alocados aos grupos mais próximos. Após cada inserção, ajusta-se os centroides dos grupos para manter o agrupamento atualizado. Por fim, um algoritmo de agrupamento convencional pode ser aplicado nos centroides dos grupos para obter uma hierarquia e, assim, é obtido uma organização dos documentos (Grossman e Frieder, 2004).

No trabalho de Zhang et al. (1996), é proposto o Birch (*Balanced Iterative Reducing and Clustering using Hierarchies*). Os autores apresentam um algoritmo que atualiza incrementalmente um sumário com as principais informações da base de dados, sem a necessidade de armazenar todos os objetos em memória. A estrutura responsável por manter esse sumário é denominada *Cluster Feature* (CF), que são organizadas em uma árvore chamada CF-Tree. Quando um novo objeto é apresentado, realiza-se uma busca pelo nó folha com a CF mais similar ao objeto, partindo-se da raiz. Ao encontrar um nó folha, o objeto é inserido e as informações da CF são atualizadas, assim como as CF's dos nós ancestrais. Zhang et al. (1996) afirmam que os nós folhas da CF-Tree podem ser, então, agrupados por um algoritmo de agrupamento tradicional para obtenção de uma organização dos dados. Originalmente, o Birch foi proposto para trabalhar com bases de dados nas quais medidas métricas podem ser aplicadas. No trabalho de Wong e Fu (2000), é proposta a DC-Tree, uma estrutura muito similar à CF-Tree, mas apropriada para dados textuais.

Zamir e Etzioni (1998) apresentam o algoritmo STC (*Suffix Tree Clustering*) para o agrupamento incremental de documentos. Para tal, é utilizada uma estrutura conhecida como *Suffix Tree*, que armazena uma cadeia de sufixos, com as palavras de todas as frases contidas em uma coleção textual. Os autores descrevem uma versão incremental da *Suffix Tree* para realizar o processamento dos textos em cenários dinâmicos. A formação do agrupamento é feita com a união dos documentos que possuem frases em comum. As avaliações experimentais do STC mostram que o método é eficaz apenas em coleções de textos de domínio restrito, com um dicionário controlado de termos (Chim e Deng, 2007). Assim, geralmente o método é aplicado na organização de resultados de sistemas de busca

(Zeng et al., 2004).

Ester et al. (1998) apresentam um algoritmo de agrupamento incremental baseado no conceito de densidade. O algoritmo é chamado de *Incremental DBScan*, inspirado em outro algoritmo proposto anteriormente. No *Incremental DBScan*, um novo objeto observado é inserido em uma região de densidade que mais se aproxima do seu conjunto de atributos. Uma vez alocado em uma região, os objetos vizinhos pertencentes a um raio de valor predeterminado são atualizados. Uma desvantagem deste algoritmo é determinar os valores de densidade apropriados em cada base de dados.

Allan (2002) adaptam o algoritmo Leader para a tarefa de agrupamento incremental de notícias, geralmente tratada como *TDT - Topic Detection and Tracking*. Assim, uma nova notícia observada é apresentada aos grupos de notícias existentes. Se a similaridade entre a notícia e os grupos existentes estiver acima de uma limiar, então a notícia é incluída ao grupo mais similar. Caso contrário, um novo grupo é criado para a notícia. Nesse caso, os autores afirmam que um novo evento foi identificado, ou seja, um tópico novo ou assunto está sendo publicado. Algumas variações desse algoritmo incluem o uso de valores temporais durante o cálculo da similaridade (Allan, 2002).

No trabalho de Widyantoro et al. (2002), é proposto o algoritmo IHC (*Incremental Hierarchical Clustering*). A inserção de um novo objeto ao agrupamento é realizado em duas fases: (1) inicialmente o nó da hierarquia com o grupo mais similar ao objeto é selecionado e (2) após a inserção do novo objeto, o algoritmo realiza uma reestruturação da hierarquia, mas limitada aos nós pais e filhos do nó selecionado. A reestruturação é feita com o objetivo de manter a coesão e homogeneidade dos grupos afetados.

Hammouda e Kamel (2003) apresentam um algoritmo de agrupamento incremental de documentos baseado no conceito de "histograma de similaridade do grupo". Para tal, em cada grupo é calculado um histograma com os valores similaridades entre os documentos ali presentes. Esse histograma, na prática, funciona como uma medida de coesão interna do grupo. Quando um novo documento é apresentado ao algoritmo, é selecionado o grupo que apresenta a menor degradação de coesão interna. Se a degradação for maior que um valor α, então um novo grupo é criado para o documento. Na prática, os autores reapresentaram o algoritmo Leader, mas com uso de medidas estatísticas de qualidade dos grupos para guiar o agrupamento incremental.

No trabalho de Sahoo et al. (2006), é apresentada uma adaptação dos algoritmos Classit (Gennari et al., 1989) e Cobweb (Fisher, 1987). O Classit e o Cobweb realizam agrupamento hierárquico conceitual de objetos. O Classit é apropriado para dados com atributos reais enquanto o Cobweb para atributos categóricos. Nesses algoritmos, o agrupamento é realizado de acordo com uma função de utilidade, medida conforme uma distribuição de probabilidades pré-estabelecida. Dado um novo objeto, o mesmo é alocado ao grupo que

maximiza a função de utilidade. Sahoo et al. (2006) afirmam que os algoritmos originais não são apropriados para dados textuais, e propuseram uma alteração na distribuição de probabilidades, apresentando a distribuição de Katz. Nessa distribuição, são exploradas as coocorrências entre termos da coleção para o cálculo da função de utilidade. No entanto, a distribuição de Katz precisa ser ajustada com parâmetros que dependem das características dos textos.

Nguyen-Hoang et al. (2009) adaptaram o algoritmo *Incremental DBScan* (Ester et al., 1998) para agrupamento incremental de documentos baseado em grafo. Nesse trabalho, os documentos são mapeados em um grafo, no qual os vértices são os documentos e as arestas representam a conexão entre dois documentos similares. Dois documentos são conectados no grafo se a similaridade for maior que um limiar definido pelo usuário. As regiões de densidade são definidas de acordo com o grau dos vértices. O agrupamento é obtido por meio de uma estratégia para identificar subgrafos com alto grau de conexão.

Já em Garcia e Porrata (2010), é proposto um método agrupamento hierárquico dinâmico também baseado em grafos. Nesse trabalho, vários grafos são construídos conforme um limiar de similaridade definido pelo usuário. Um valor baixo de similaridade, produz um grafo com muitas conexões (arestas) entre os documentos. Por outro lado, um valor alto de similaridade produz um grafo em que apenas os documentos mais similares estão conectados. O grafo é atualizado de maneira incremental, conforme novos documentos são apresentados. Uma hierarquia de grupos é obtida com a variação dos valores de similaridade, em que o topo da hierarquia contém os grupos provenientes do grafo com valor baixo de similaridade, enquanto os nós de níveis mais profundo contêm grupos formados pelos grafos com alto valor de similaridade.

Todos os métodos aqui apresentados podem, a princípio, ser aplicados para auxiliar na organização de coleções textuais dinâmicas. No entanto, neste projeto de mestrado foram selecionados três algoritmos: Leader, Buckshot e DCTree. Essa decisão foi baseada em aspectos que, em geral, buscam unir a simplicidade de implementação com a diversidade de estratégias. O Leader é um algoritmo de simples implementação, além de ser computacionalmente eficiente. Já o Buckshot foi um dos primeiros trabalhos especificamente voltado para organização de coleções textuais. Por fim, o DCTree proporciona uma estrutura similar à CF-Tree, que é uma estratégia interessante para sumarização dos dados.

Na próxima seção são apresentados os pseudocódigos (contextualizados para dados textuais) para esses algoritmos, a análise de complexidade e é discutido o uso do agrupamento incremental para representação condensada dos textos.

3.2 Algoritmo Leader

Para descrição dos algoritmos de agrupamento incremental utilizados nesse trabalho, considere o uso da similaridade cosseno, adaptada para medida de dissimilaridade. Ainda, os documentos estão representados no modelo espaço-vetorial, conforme apresentado no Capítulo 2.

O pseudocódigo para o Leader é apresentado no Algoritmo 4. O Leader recebe, como parâmetros de entrada, uma fonte incremental de documentos X_{inc} e um valor de Dissimilaridade Mínima α.

Algoritmo 4: O algoritmo Leader

Entrada:
$X_{inc} = \{x_1, x_2, ...\}$: fonte incremental de documentos
α: Dissimilaridade Mínima
Saída:
$P = \{G_1, G_2, ..., G_k\}$: partição com k grupos

```
1  para cada novo documento x ∈ X_inc faça
       // busca pelo grupo mais próximo
2      MinDist ← +∞;
3      G_sel ← ∅;
4      para cada grupo G_i ∈ P faça
5          Dist ← d(x, G_i);
6          se Dist < MinDist então
7              MinDist ← Dist;
8              G_sel ← G_i^t;
9          fim
10     fim
       // agrupando o documento x
11     se MinDist < α então
12         alocar x no grupo mais próximo G_sel;
13     senão
14         criar novo grupo G_novo em P;
15         alocar x no grupo G_novo;
16     fim
17 fim
```

Para cada novo documento x apresentado, é feita uma busca pelo grupo existente mais próximo de x. Para tal, usa-se uma função $d(x, G)$ (Linha 5) para calcular a dissimilaridade cosseno entre x e o centroide de um grupo G_i. Nas linhas 1 à 10 é descrito o processo de busca pelo grupo mais próximo. A variável $MinDist$ armazena o valor de dissimilaridade do grupo mais próximo de x. Nas linhas 11 à 16, o documento x é alocado a um dos grupos existentes, ou um novo grupo é criado para x.

Configurar o parâmetro de Dissimilaridade Mínima do Leader, em geral, é uma tarefa difícil. Um valor baixo para Dissimilaridade Mínima corresponde a um grande número de k grupos, pequenos e compactos. Por outro lado, valores altos geram um pequeno número

de grupos, com baixa coesão. Em geral, esse valor é dependente das características da coleção textual e deve ser configurado com auxílio de alguma medida de validação de agrupamentos.

Pode-se observar que o Leader, originalmente, não apresenta um critério para controlar o número máximo de grupos que podem ser gerados. Em alguns trabalhos, são explorados critérios para união e remoção de grupos conforme a evolução do agrupamento, auxiliando a controlar o número de grupos gerados (Allan, 2002). No entanto, neste trabalho de mestrado, adotou-se uma solução mais simples, baseada no conceito de *Winner-takes-all* (Zhong, 2005). Nesse caso, quando o número de grupos k atinge um valor máximo $MaxSubGrupos$, o documento x deve ser alocado ao grupo mais próximo, independentemente do valor de $MinDist$. A definição do valor de $MaxSubGrupos$ é razoavelmente simples, pois depende apenas da memória disponível para a aplicação.

A complexidade de tempo do algoritmo Leader é $O(k \cdot n)$. Como $n >> k$, o algoritmo Leader obtém o agrupamento em tempo linear em relação ao número de documentos.

3.3 Algoritmo Buckshot

A execução do Buckshot pode ser dividida em duas fases: (1) construção do agrupamento hierárquico inicial e (2) agrupamento incremental de documentos. O pseudocódigo para o Buckshot é apresentado no Algoritmo 5.

Algoritmo 5: O algoritmo Buckshot

Entrada:
$X_{inc} = \{x_1, x_2, ...\}$: fonte incremental de documentos
$Y_{sample} = \{y_1, y_2, ..., y_n\}$: amostra de documentos para agrupamento inicial
Saída:
$P = \{G_1, G_2, ..., G_k\}$: partição com k grupos

```
// construindo agrupamento inicial
H ← agrupamento hierárquico inicial;
se Y_sample ≠ ∅ então
    H ← AgrupamentoHierarquico(Y_sample);
senão
    // agrupando novos documentos
    para cada novo documento x ∈ X_inc faça
        buscar o nó folha G mais próximo a partir da raiz de H;
        alocar x no grupo G;
    fim
fim
```

Na primeira fase, é necessário apresentar uma amostra inicial de documentos Y_{sample}. O tamanho da amostra, em geral, é definido de acordo a memória disponível para a aplicação. Um algoritmo de agrupamento hierárquico é aplicado a partir da amostra,

construindo o agrupamento inicial dos textos (Linhas 2 à 4). Na segunda fase, novos documentos são inseridos ao agrupamento existente por meio de uma fonte incremental de documentos X_{inc}. Durante a inserção, o documento percorre a hierarquia de grupos, a partir da raiz, até encontrar o grupo folha para ser alocado (Linhas 5 à 8). O centroide do grupo folha e de todos seus ancestrais são ajustados para atualização do agrupamento.

No algoritmo original (Cutting et al., 1992), os autores não definem o método de agrupamento para construção do agrupamento inicial (Linha 3). No Algoritmo 5 apresentado, adotou-se a implementação descrita em Grossman e Frieder (2004), que utiliza agrupamento hierárquico para a construção do agrupamento inicial. Nesse caso, a saída do algoritmo é a partição formada por todos os grupos folha da hierarquia.

É importante observar que o Buckshot mantém um número fixo de grupos durante o agrupamento incremental. Assim, o número final de grupos é dependente do tamanho da amostra inicial.

A complexidade de tempo do algoritmo Buckshot é $O(k \cdot n)$, em que k é o número de grupos obtido do agrupamento inicial. Como $n >> k$, o algoritmo Buckshot obtém o agrupamento em tempo linear em relação ao número de documentos.

3.4 Algoritmo DCTree

O algoritmo DCTree, descrito nesta seção, é apresentado com mais detalhes no trabalho de Wong e Fu (2000). No DCTree, utiliza-se uma estrutura de dados denominada DC (*Document Cluster*), que armazena o sumário de um conjunto de documentos por meio de três informações: (1) o número de documentos alocados em DC, (2) o conjunto de identificadores desses documentos e (3) um vetor de características, calculado como a soma linear dos vetores dos documentos alocados em DC. Cada estrutura DC é organizada em uma árvore com estrutura similar a uma $B^+ - Tree$, conforme apresentado na Figura 3.1.

A árvore de agrupamentos utilizada no DCTree possui um parâmetro de Fator de Ramificação, que define o número máximo de filhos para os nós. Ainda, os nós da hierarquia possuem B *slots*, no qual cada *slot* armazena uma estrutura do tipo DC. Para simplificar, neste trabalho, adotou-se que o número máximo de *slots* de um nó é igual o Fator de Ramificação da árvore. O DCTree também possui um parâmetro de Dissimilaridade Mínima para a formação dos grupos.

O agrupamento incremental no DCTree é realizado da seguinte maneira (Algoritmo 6). Dado um novo documento x proveniente de X_{inc}, busca-se o nó folha com o DC mais próximo de x, iniciando da raiz (Linha 3). Para tal, é utilizado a medida cosseno entre x e o vetor de características de DC. Se a proximidade entre x e DC respeitar o limiar

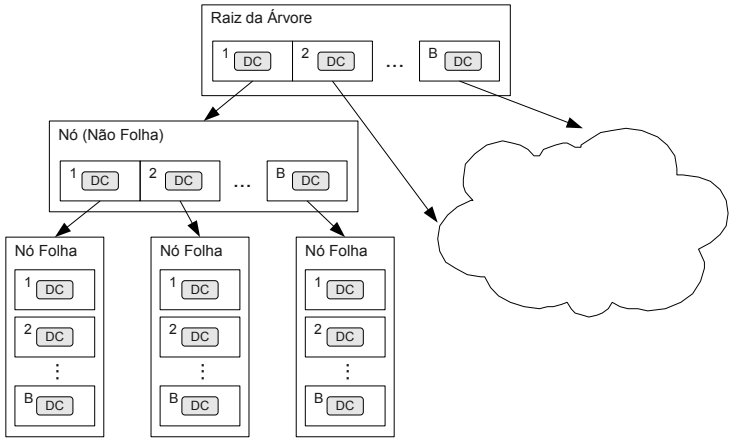

Figura 3.1: Exemplo de estrutura hierárquica utilizada no DCTree

de Dissimilaridade Mínima, então x é alocado em DC (Linhas 5 à 6). Caso contrário, o algoritmo verifica se há algum *slot* livre no mesmo nó que se encontra o DC em questão. Se há espaço, um novo DC é criado e x é alocado (Linha 9), senão um nó filho é criado para agrupar x (Linhas 11 à 12).

Algoritmo 6: O algoritmo DCTree ⇒ inserção de documento

Entrada:
$X_{inc} = \{x_1, x_2, ...\}$: fonte incremental de documentos
α: Dissimilaridade Mínima
β: Fator de Ramificação

Saída:
$P = \{G_1, G_2, ..., G_k\}$: partição com k grupos

```
1  T ← inicializar estrutura hierárquica do DCTree;
2  para cada novo documento x ∈ X_inc faça
3      buscar o nó folha N com o DC mais próximo de x a partir da raiz de T;
4      Dist ← d(x, DC);
5      se Dist < α então
6          alocar x em DC;
7      senão
           // verifica se existe espaço para um novo DC em N
8          se número de slots de N < β então
9              criar um novo DC_new em N e alocar x;
10         senão
11             criar um nó filho N_child para N;
12             criar um novo DC_new em N_child e alocar x;
13         fim
14     fim
15 fim
```

Uma característica do DCTree é que a árvore de grupos é dinâmica, que evolui conforme a necessidade de agrupar novos documentos. Neste trabalho de mestrado, adotou-se apenas a operação de inserção, mas o algoritmo original possui operações para união (*merging*) e remoção de nós da hierarquia. Para controlar a geração do número máximo

de grupos foi utilizada uma estratégia similar ao discutido no Leader, ou seja, após um determinado número de grupos ($MaxSubGrupos$) os novos documentos são alocados aos grupos folhas mais similares, sem considerar o limiar de Dissimilaridade Mínima.

O DCTree é um algoritmo similar ao CF-Tree (Zhang et al., 1996), adaptado para dados textuais. Neste caso, assim como no CF-Tree, a saída do algoritmo é a partição formada por todos os grupos folha da hierarquia. A complexidade geral do algoritmo, conforme apresentado neste trabalho, é linear em relação ao número de documentos (Xu e Wunsch, 2008).

3.5 Agrupamento Hierárquico Baseado em Representação Condensada dos Textos

Os algoritmos de agrupamento incremental Leader, Buckshot e DCTree, são eficientes para a obtenção de um sumário dos dados. Dessa forma, uma aplicação interessante desses algoritmos é na construção da representação condensada de um conjunto de dados. Essa representação condensada é utilizada em diversas tarefas de descoberta de conhecimento, com redução significativa do custo computacional (Xu e Wunsch, 2008; Kishida, 2010). Para exemplificar, considere a tarefa de agrupamento hierárquico sobre a base de dados ilustrada na Figura 3.2.

Em um primeiro momento, o agrupamento hierárquico é obtido a partir do conjunto total dos dados (Figura 3.2a). Alternativamente, na Figura 3.2b, a base de dados é sumarizada em uma representação condensada em 14 grupos. O agrupamento hierárquico aplicado sobre esta representação obtém uma estrutura hierárquica com propriedades similares à anterior. No entanto, ao utilizar apenas 14 centroides como representantes dos grupos, há uma significativa redução dos requerimentos computacionais.

A técnica utilizada para obter a representação condensada dos dados, principalmente em cenários dinâmicos, deve construir e atualizar esta representação de maneira incremental. Assim, alguns trabalhos na literatura têm explorado o uso de algoritmos de agrupamento incremental para obter essa representação condensada de forma eficiente (Zhang et al., 1996; Nassar et al., 2004; Liu et al., 2006; Kishida, 2010).

Supondo que uma coleção textual de n documentos esteja representada por centroides de k grupos, a complexidade do agrupamento hierárquico com representação condensada é reduzida de $O(n^2)$ para $O(k^2)$. Como geralmente $n >> k$, essa estratégia é eficaz para manter a organização de coleções textuais dinâmicas, que exigem frequentes atualizações do agrupamento hierárquico.

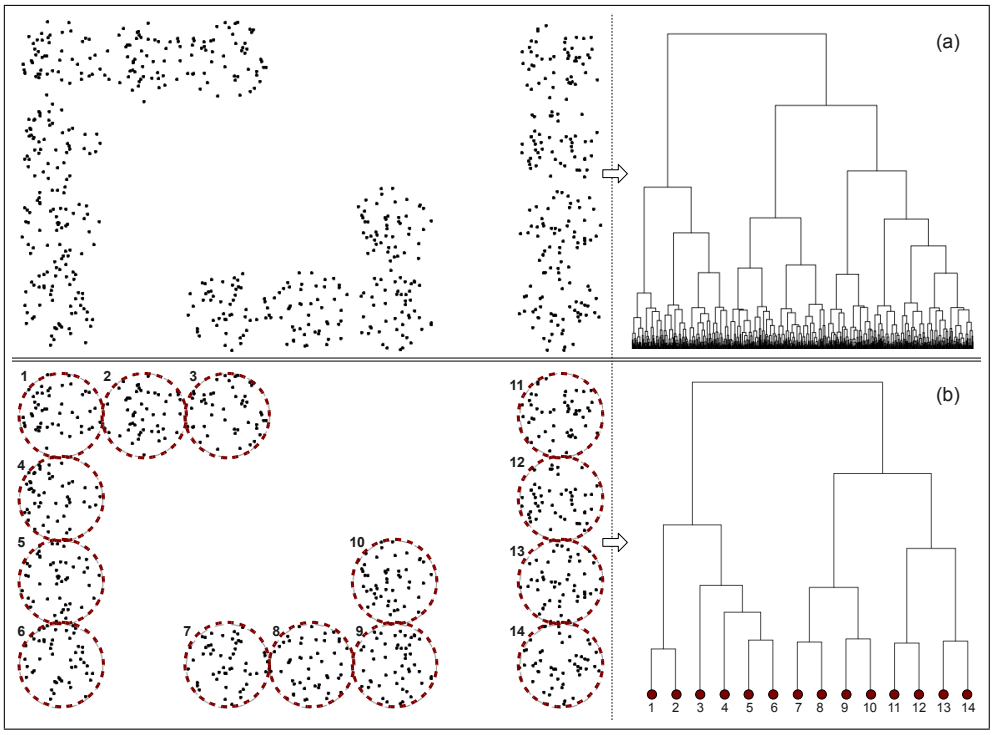

Figura 3.2: (a) Agrupamento hierárquico. (b) Agrupamento hierárquico resultante da representação condensada dos dados.

3.6 Considerações Finais

O uso de agrupamento incremental tem se mostrado bastante eficiente em cenários que lidam com dados dinâmicos. Em contextos como o deste trabalho, no qual deseja-se aprender hierarquias de tópicos a partir de coleções textuais dinâmicas, os métodos de agrupamento incremental são extremamente importantes, na medida em que evitam um reprocessamento redundante de dados.

Neste capítulo, foram discutidos os principais trabalhos e algoritmos na literatura que, de alguma forma, lidam com o agrupamento incremental de dados textuais. Dentre esses, foram selecionados três algoritmos para serem investigados: Leader, Buckshot e DCTree; visando a simplicidade de implementação e diversidade das estratégias. Para cada um, foi apresentado e comentado o respectivo pseudocódigo, os parâmetros envolvidos e a análise de complexidade.

O uso dos algoritmos de agrupamento incremental, neste trabalho, segue a mesma estratégia proposta no algoritmo Birch (Zhang et al., 1996), e nos trabalhos de Nassar et al. (2004); Liu et al. (2006); Kishida (2010). O agrupamento incremental é aplicado para obtenção e atualização incremental de uma representação condensada dos textos, que é utilizada no agrupamento hierárquico, reduzindo significativamente o custo computacional.

É importante ressaltar que, para o aprendizado de hierarquias de tópicos com uso de agrupamento hierárquico de documentos, a interpretação do agrupamento obtido é um grande obstáculo para os usuários. Em geral, é necessário aplicar uma técnica, após o agrupamento, para seleção dos termos mais discriminativos e utilizá-los como descritores do agrupamento. Em vista disso, no próximo capítulo é proposto e analisado um método alternativo de agrupamento incremental para obter uma representação condensada dos textos, que visa a seleção de descritores juntamente com a formação de grupos de documentos. Assim, espera-se facilitar o aprendizado das hierarquias de tópicos, principalmente em relação à interpretação dos agrupamentos.

Capítulo

4

O método *IHTC - Incremental Hierarchical Term Clustering*

No contexto de organização de coleções textuais, os algoritmos de agrupamento podem ser aplicados de duas formas: (1) agrupamento de documentos e (2) agrupamento de termos (Kowalski e Maybury, 2002; Ebecken et al., 2003).

No agrupamento de documentos, os documentos são representados como vetores de termos e o agrupamento é obtido com base em uma medida de (dis)similaridade entre estes vetores. Essa é a forma mais utilizada para organização de coleções textuais, no entanto, o agrupamento obtido não possui descritores para interpretação dos resultados, sendo necessário uma etapa adicional para seleção desses descritores (Fung et al., 2003).

Já no agrupamento de termos, os termos são agrupados baseado na coocorrência dos termos nos documentos. Essa forma geralmente é utilizada para a auxiliar a construção de *thesaurus*, extração de conceitos e como redução de dimensionalidade (Ebecken et al., 2003). Recentemente, o agrupamento de termos também tem sido utilizado para organização da coleção de documentos (Sanderson e Croft, 1999; Slonim e Tishby, 2000; Fung et al., 2003; Li et al., 2008; Zhang et al., 2010). Esses métodos possuem a vantagem de que os grupos de documentos obtidos (a partir dos grupos de termos) são acompanhados de descritores que indicam um significado para cada grupo. De forma geral, existe um mapeamento entre cada grupo de termos para seu respectivo grupo de documentos. A qualidade da organização obtida por esses métodos tem se mostrado comparável aos métodos clássicos e são utilizados com sucesso em algumas aplicações, como na organização de resultados em sistemas de busca e segmentação temática de coleções textuais (Fung et al., 2003; Li et al., 2008). No entanto, ainda existem poucos trabalhos na literatura que exploram a organização de coleções textuais dinâmicas usando algoritmos de agrupamento

de termos (Fung et al., 2009).

Dada as vantagens de estratégias de agrupamento de termos, neste trabalho foi proposto um método de aprendizado de hierarquias de tópicos baseada em agrupamento incremental de termos denominada *IHTC - Incremental Hierarchical Term Clustering*. O IHTC permite organizar coleções textuais dinâmicas em uma representação condensada, na qual cada grupo possui candidatos a descritores para formação dos tópicos. Assim, neste capítulo são apresentadas as especificações do método proposto e o respectivo algoritmo.

O método proposto neste trabalho de mestrado atua na etapa de extração de padrões de um processo de mineração de textos (Seção 2.3 do Capítulo 2). O método é dividido em quatro fases, conforme ilustrado na Figura 4.1: (1) construção da rede de coocorrência de termos; (2) agrupamento incremental de termos; (3) agrupamento incremental de documentos e (4) extração da hierarquia de tópicos. O objetivo é manter uma representação condensada da coleção textual, de maneira incremental, e utilizar os grupos de termos para apoiar o aprendizado de hierarquias de tópicos.

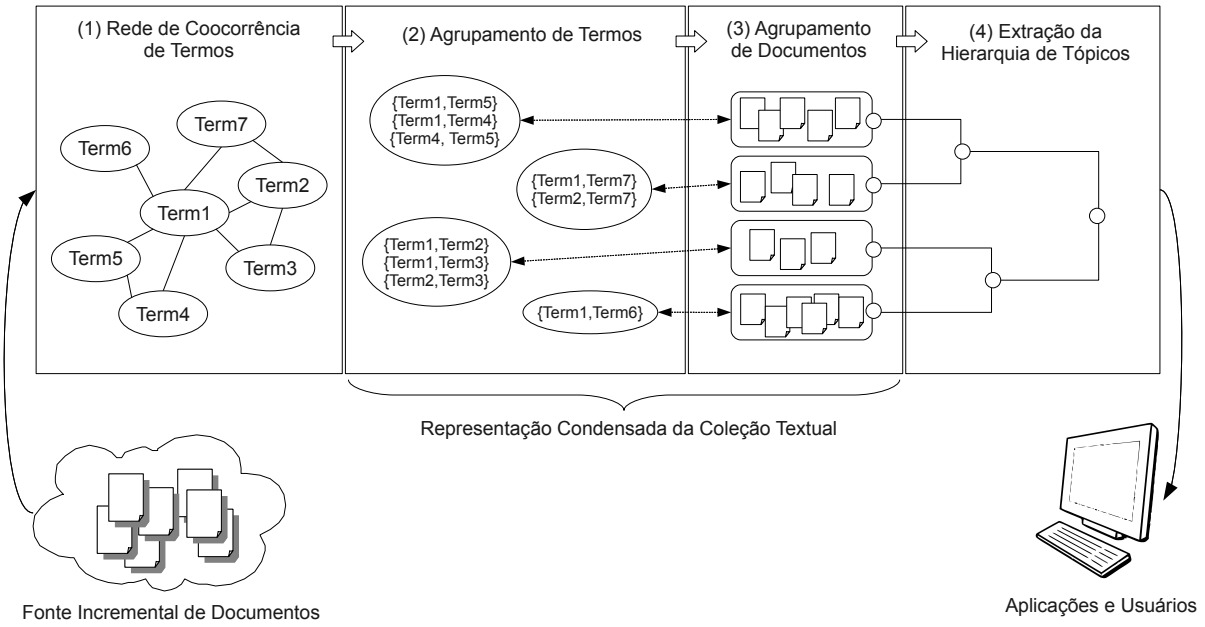

Figura 4.1: Visão geral do método IHTC

Nas próximas seções serão discutidos mais detalhes das etapas do método proposto.

4.1 Rede de Coocorrência de Termos

Uma rede de coocorrência de termos é definida como um grafo $GRAFO(V, E, W)$, no qual V é um conjunto de vértices, E é um conjunto arestas (não dirigidas) que relacionam

dois vértices e, por fim, W é um conjunto de pesos associados às arestas, identificando a intensidade de cada relação.

Os **vértices** são os termos existentes na coleção textual, mais especificamente, termos selecionados para representação de cada documento no modelo espaço-vetorial. No entanto, nem todos os termos são utilizados como vértices. Para um termo ser utilizado, é necessário que o mesmo participe de uma relação de coocorrência com outro termo da coleção, ou seja, ocorrerem juntos em múltiplos documentos da coleção textual.

A coocorrência entre dois termos também identifica as **arestas** do grafo. Assim, dois termos são conectados por uma aresta se existe coocorrência significativa entre eles. Neste trabalho, a coocorrência entre dois termos é considerada significativa se a frequência dessa coocorrência é maior que um determinado limiar de frequência mínima. Esse limiar é dependente das características de cada coleção textual. No entanto, será visto mais adiante que não é necessário que o usuário defina diretamente um valor de frequência mínima, uma vez que o algoritmo desenvolvido para o IHTC identifica automaticamente esse limiar.

Em geral, os **pesos** das arestas são valores numéricos (números inteiros, racionais ou reais) usados para identificar a intensidade da relação entre dois termos. Neste trabalho, é proposto o uso de um centroide para identificar essa relação, pois permite representar de forma sucinta um conjunto de documentos no modelo espaço-vetorial. Para tal, seja $e = \{t_i, t_j\}$ uma aresta que une os termos t_i e t_j, então $w(e)$ é uma função que associa um centroide à uma aresta e de acordo com Equação 4.1:

$$w(e) = C(t_i \cap t_j) \tag{4.1}$$

em que $C(t_i \cap t_j)$ é o centroide que representa o subconjunto de documentos com ambos os termos t_i e t_j.

Dessa forma, a rede de coocorrência de termos, conforme aplicada neste trabalho, pode ser vista como uma estrutura com duas características principais:

1. Capaz de identificar relações significativas entre os termos da coleção textual, baseada na frequência de coocorrência; e

2. Capaz de extrair subconjuntos de documentos (representados por centroides), em que os pares de termos (arestas) podem ser utilizados como candidatos a descritores.

Para exemplificar esse processo, considere os 10 títulos de livros apresentados na Tabela 4.1 como uma pequena base de textos. Os livros são provenientes da área de mineração de dados e textos, aprendizado de máquina e recuperação de informação. Uma possível rede de coocorrência de termos extraída a partir dos títulos é apresentada na Figura 4.2.

Tabela 4.1: Base textual composta por 10 títulos de livros

ID	Título do Livro
D1	Advances in knowledge discovery and data mining
D2	Information visualization in data mining and knowledge discovery
D3	Visual information retrieval
D4	Visual data mining: techniques and tools for data visualization and mining
D5	The text mining handbook: advanced approaches in analyzing unstructured data
D6	Text mining: theoretical aspects and applications
D7	Data mining: practical machine learning tools and techniques
D8	Learning from data: concepts, theory, and methods
D9	Machine learning: applications in expert systems and information retrieval
D10	Text information retrieval systems

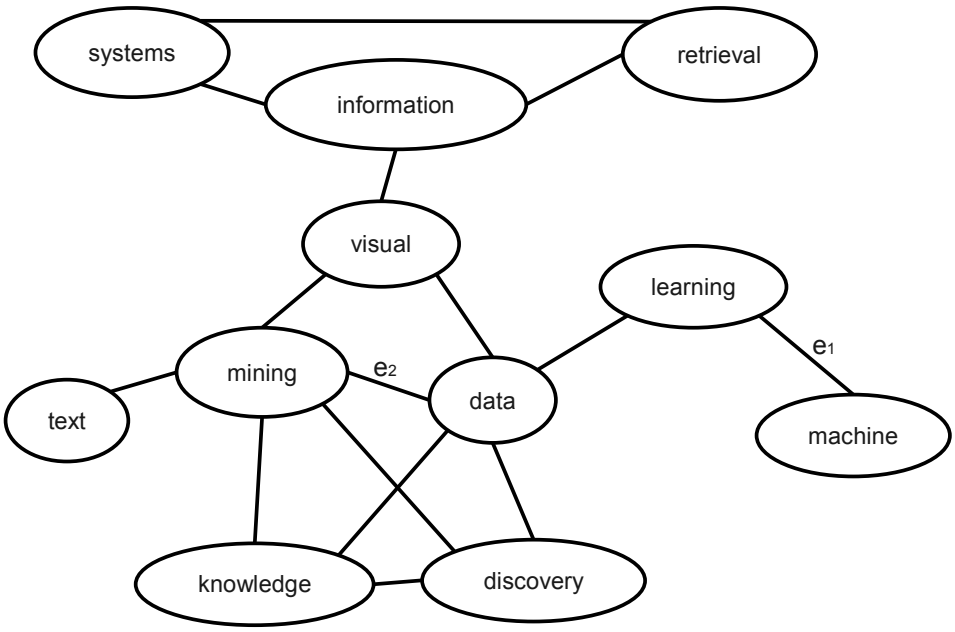

Figura 4.2: Rede de coocorrência de termos da base da Tabela 4.1

Neste exemplo, o par de termos (aresta) $e_1 = \{machine, learning\}$ está associado a um centroide que representa o subconjunto de documentos $G_1 = \{D_7, D_9\}$. De forma análoga, o par de termos $e_2 = \{data, mining\}$ está associado a um centroide que representa o subconjunto de documentos $G_2 = \{D_1, D_2, D_4, D_7\}$. Pode-se notar que o conteúdo desses subconjuntos de documentos são bem identificados por seus descritores. Uma outra observação importante é que a rede de coocorrência permite a identificação de múltiplos tópicos na coleção por meio da sobreposição de grupos. Por exemplo, o documento D_7="*Data mining: practical machine learning tools and techniques*" está presente tanto em $e_1 = \{machine, learning\}$ quanto em $e_2 = \{data, mining\}$.

A rede de coocorrência de termos é uma estrutura útil para identificação de tópicos, utilizada em trabalhos relacionados à organização de informação textual. Suas aplicações,

principalmente quando integradas à ferramentas de visualização, permitem exploração dos tópicos de forma interativa pelo usuário. Uma revisão com outras variações de rede de coocorrência, para diferentes cenários, pode ser encontrada em Feldman e Sanger (2006).

4.2 Agrupamento de Termos

A rede de coocorrência de termos, em geral, contém todas as relações que possuem coocorrência significativa. Muitas dessas relações de coocorrência podem ser agrupadas, obtendo-se um agrupamento de termos.

Dessa forma, o objetivo do agrupamento de termos é sumarizar as relações existentes na rede de coocorrências em grupos de termos compactos e bem separados. Para tal, cada par de termos (aresta), representado pelo seu centroide, é visto como um objeto pelo algoritmo de agrupamento. As medidas de (dis)similaridades estudadas até o momento podem ser utilizadas normalmente, uma vez que é mantida a representação no espaço-vetorial devido ao uso dos centroides para representação desses objetos.

Considerando o exemplo anterior, ou seja, a rede de coocorrência de termos ilustrada na Figura 4.2, um exemplo de um resultado para a fase de agrupamento de termos é uma organização em 5 grupos, conforme apresentado na Tabela 4.2. O mesmo esquema de agrupamento é ilustrado diretamente sobre a rede de coocorrência de termos (Figura 4.3), na qual cada aresta possui um rótulo que identifica seu grupo.

Tabela 4.2: Exemplo do agrupamento de termos a partir da rede de coocorrência de termos da Figura 4.2

Grupo 1	**Grupo 2**	**Grupo 3**	**Grupo 4**	**Grupo 5**
{data,discovery} {knowledge,mining} {discovery,mining} {knowledge,discovery} {data,knowledge}	{visual,information} {data,mining} {visual,data} {visual,mining}	{information,retrieval} {retrieval,systems} {information,systems}	{text,mining}	{data,learning} {machine,learning}

Os grupos de termos obtidos nessa fase identificam candidatos aos tópicos existentes na coleção textual. Um diferencial é que a relação de coocorrência entre os termos é mantida após o agrupamento, por exemplo, "{ *machine, learning* }". A literatura indica conjuntos de termos são mais apropriados para representação de conceitos, tanto em relação à semântica e interpretabilidade para os usuários quanto em tarefas de aprendizado preditivas e descritivas (Zhang et al., 2010).

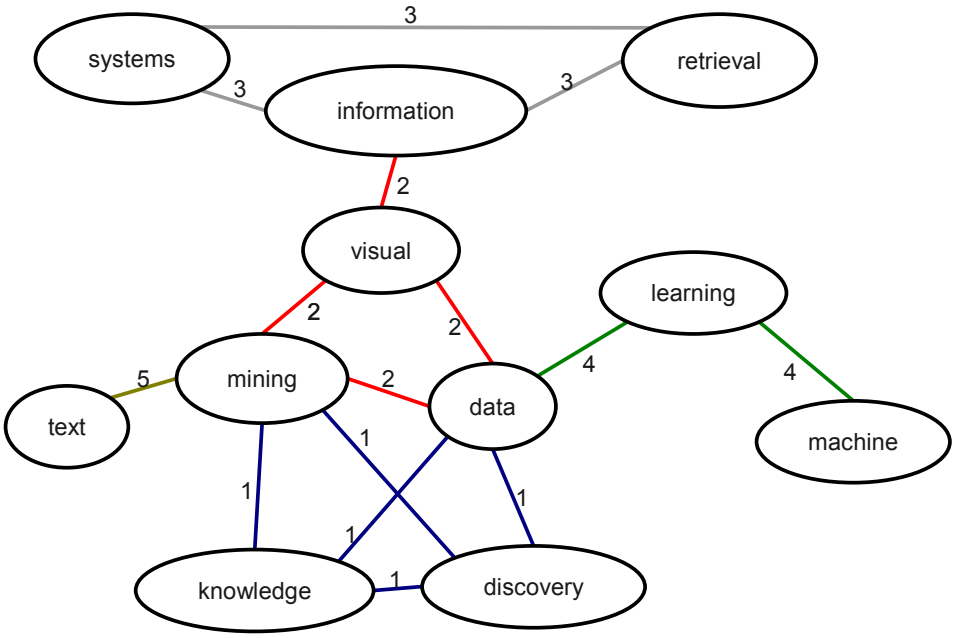

Figura 4.3: Exemplo do agrupamento de termos

4.3 Agrupamento de Documentos

No agrupamento de documentos, os documentos da coleção são mapeados aos grupos de termos mais próximos, conforme uma medida de (dis)similaridade. Para tal, considere que G_i^t representa um grupo de termos e G_i^d um grupo de documentos. Dado um documento x e uma partição $P_t = \{G_1^t, G_2^t, ..., G_k^t\}$ com k grupos de termos, calcula-se a similaridade entre x e o centroide de cada grupo G_i^t e, em seguida, o documento x é mapeado ao grupo com maior valor de similaridade (ou menor valor de dissimilaridade). Dessa forma, o agrupamento final é formado por grupos do tipo $G_i = (G_i^d, G_i^t)$, ou seja, composto por um grupo de documentos G_i^d e um grupo de termos G_i^t. Ao contrário das estratégias clássicas, que precisam de um processo adicional de seleção de descritores para os grupos de documentos (ver Seção 2.3.3), neste método os descritores são obtidos automaticamente durante o processo.

Por outro lado, durante o mapeamento dos documentos utilizando os grupos de termos, pode ocorrer que um documento esteja próximo à dois ou mais grupos, já que algumas coleções textuais possuem tópicos com sobreposição. Assim, ocasionalmente há empates na seleção do grupo de termos mais similar ao documento. Uma solução é mapear o documento a todos os grupos empatados e permitir um agrupamento com sobreposição. Outra solução é selecionar aleatoriamente apenas um dos grupos empatados para mapear o documento (agrupamento rígido). A segunda solução foi adotada neste trabalho, a fim de comparar o método proposto com as demais estratégias da literatura.

Ao final desse processo, a coleção textual está organizada em grupos, em que cada grupo de documentos possui um grupo de termos para auxiliar na descrição do seu conteúdo. Neste trabalho, o processo de agrupamento aqui descrito é proposto como uma alternativa para a etapa de condensação da coleção textual. Assim, é necessário que todo esse processo seja realizado de maneira incremental. Na próxima seção, é apresentado o algoritmo que realiza o processo incremental do método proposto.

4.4 Algoritmo

Foi desenvolvido um algoritmo para o método IHTC, responsável por instanciar a etapa de extração de padrões de um processo de mineração de textos, com o objetivo de organizar coleções textuais dinâmicas em uma hierarquia de tópicos. A seguir são descritas as principais características do algoritmo.

1. *Construção e manutenção incremental da rede de coocorrência de termos.*
   A construção da rede de coocorrência de termos é uma etapa importante do método proposto. O IHTC permite construir e manter, de forma incremental, uma rede de coocorrência de termos conforme novos documentos são apresentados ao algoritmo. Para tal, foi utilizado um método denominado "*Top-K Frequent Elements*" (Metwally et al., 2005, 2006), proposto originalmente para estimar os elementos mais frequentes de um "*data stream*". No IHTC, esse método é adaptado para identificar pares de termos que ocorrem em múltiplos documentos na coleção textual. Conforme comentado anteriormente, os pares de termos identificados são utilizados como arestas da rede de coocorrência.

2. *Agrupamento incremental de termos.*
   Para cada atualização ocorrida na rede de coocorrência de termos, a aresta inserida ou modificada é utilizada como um novo objeto para o agrupamento incremental de termos. O agrupamento incremental segue uma estratégia similar ao do algoritmo Leader, com a diferença de que não há um parâmetro com o valor de similaridade mínima (*threshold*). Nesse caso, os primeiros objetos formam grupos unitários até o limite de $MaxSubGrupos$ e, após isto, os novos objetos são alocados ao grupo mais próximo.

3. *Agrupamento incremental de documentos.*
   Os documentos apresentados são armazenados em um *buffer* temporário, de tamanho limitado a $MaxSubGrupos$. Assim que uma nova iteração de agrupamento incremental de termos é processada, os documentos armazenados no *buffer* são utilizados para o agrupamento incremental de documentos. Dessa forma, tanto o agrupamento

de termos quanto o agrupamento de documentos acompanha as atualizações realizadas na rede de coocorrência de termos.

4. *Extração da Hierarquia de Tópicos.*
   A organização em grupos de documentos e grupos de termos obtida até o momento é utilizada como uma representação condensada da coleção textual. Ainda, esta representação é atualizada de forma incremental, conforme novos documentos são apresentados. A extração da hierarquia de tópicos é realizada a partir desta representação condensada, por meio da aplicação de um algoritmo de agrupamento hierárquico. No IHTC, a seleção dos descritores de cada grupo é baseada nos respectivos grupos de termos, ou seja, o algoritmo mantém um conjunto de candidatos a descritores.

No Algoritmo 7 é apresentado o pseudocódigo do *IHTC*. O agrupamento incremental de termos é realizado por uma função auxiliar ilustrada no Algoritmo 8. Já o processo de agrupamento incremental de documentos é ilustrado no Algoritmo 9. A seguir, são apresentados alguns comentários relevantes a respeitos dos pseudocódigos.

- O algoritmo IHTC recebe como entrada uma fonte incremental de documentos X_{inc} e um parâmetro r, que indica o número máximo de arestas para a rede de coocorrência de termos. Na prática, o parâmetro r é relacionado com a memória disponível para monitorar pares de termos frequentes que ocorrem na coleção textual.

- Nas linhas 1 à 5 do Algoritmo 7, as variáveis utilizadas pelo algoritmo são inicializadas. A variável M é uma lista de contadores de tamanho r que armazena as arestas da rede, e todos contadores são inicializados com valor zero. Cada contador é identificado unicamente por um par de termos e contém um valor inteiro para estimar a frequencia de ocorrência do par de termos. A variável B é um *buffer* de documentos, de tamanho $MaxSubGrupos$. A rede de coocorrência de termos é identificada pela variável G. É importante observar que a estrutura da rede em G está implícita nos contadores, pelas conexões de cada par de termos. Assim, não são utilizadas outras estruturas para o grafo, como matriz ou lista de adjacências. As variáveis P_t e P_d são as variáveis relacionados ao agrupamento de termos e agrupamento de documentos, respectivamente.

- Na linha 8 do Algoritmo 7, os diferentes pares de termos do documento x são obtidos e armazenados em E_x. A obtenção desses pares de termos é realizada combinando-se os termos que representam x no modelo espaço-vetorial, dois a dois, e cada par é ordenado lexicograficamente. Dessa forma, se $|x|$ é o número de termos presente no documento x, então são obtidos $\frac{|x|*(|x|-1)}{2}$ pares de termos distintos.

Algoritmo 7: O algoritmo IHTC

Entrada:

$X_{inc} = \{x_1, x_2, ...\}$: fonte incremental de documentos

r: número máximo de arestas para a rede de coocorrência de termos

Saída:

H_T: hierarquia de tópicos

```
1  M ← inicializar lista global com r contadores;
2  B ← inicializar buffer de documentos de tamanho MaxSubGrupos;
3  G ← inicializar rede de coocorrência de termos;
4  P_t ← inicializar agrupamento de termos;
5  P_d ← inicializar agrupamento de documentos;
6  para cada novo documento x ∈ X_inc faça
7      Armazenar documento x no buffer B;
8      E_x ← conjunto com os diferentes pares de termos obtidos do documento x;
9      para cada par de termos e = {t_i, t_j} ∈ E_x faça
10         se e = {t_i, t_j} está na lista de contadores M então
11             m_e ← contador de e na lista M;
12             m_e ← m_e + 1;
13             atualizar centroide da aresta e ∈ G com o documento x ;
14             AgrupamentoTermos(P_t, e);
15         senão// substituir pelo elemento menos frequente
16             m_old ← menor contador da lista M;
17             m_new ← novo contador para e;
18             m_new ← m_old + 1;
19             substituir m_old por m_new na lista M;
20         fim
21     fim
22     se a rede de coocorrência de termos G foi atualizada então
23         para cada documento x_i ∈ B faça AgrupamentoDocumentos(P_d, P_t, x_i);
24         limpar o buffer B;
25     fim
26     se GerarHierarquiaTopicos() == "sim" então
27         H_T ← AgrupamentoHierarquico(P_d);
28         SelecionarDescritores(H_T, P_t);
29     fim
30 fim
```

- Nas linhas 9 à 21 do Algoritmo 7, é executado o processo de construção e manutenção incremental da rede de coocorrência de termos. Para cada par de termos $e = \{t_i, t_j\}$ apresentado (Linha 9), verifica-se se o mesmo já está sendo monitorado por algum contador de M. Se e já estiver sendo monitorado (Linha 10), significa que foi adicionado por um documento anterior, então o valor de frequência de coocorrência de e é incrementado (Linha 12) e a rede é atualizada (Linha 13). Caso contrário,

busca-se o contador de M com menor valor de frequência m_{old} (Linha 16) e, em seguida, m_{old} é substituído pelo contador de e, identificado como m_{new} (Linha 17). Um detalhe desse processo é que o valor de frequência de coocorrência para m_{new} é atualizado $m_{old}+1$ (Linha 18), ou seja, o valor é superestimado a fim de garantir que cada elemento frequente seja monitorado. Dessa forma, ao longo do tempo a lista de contadores M mantém os pares de termos mais frequentes, enquanto os menos frequentes são gradativamente removidos (Linha 19).

- Cada atualização realizada na rede de coocorrência (Linha 13) é seguida de um processo de agrupamento incremental de termos (Linha 14). No Algoritmo 8, o processo de agrupamento incremental de termos é descrito em mais detalhes. Dado um par de termos $e = \{t_i, t_j\}$, recupera-se um centroide que representa o subconjunto dos documentos que contêm os termos t_i e t_j (Linha 1 do Algoritmo 8). O centroide é utilizado, então, como um objeto para a fase de agrupamento. Assim, se o número de grupos existentes for menor que $MaxSubGrupos$, então é criado um novo grupo de termos (Linhas 2 à 4 do Algoritmo 8). Caso contrário, o objeto é inserido no grupo de termo mais próximo, de acordo com uma medida de dissimilaridade (Linhas 6 à 15 do Algoritmo 8).

Algoritmo 8: IHTC ⇒ Agrupamento de termos

Entrada:
$P_t = \{G_1^t, G_2^t, ..., G_n^t\}$: agrupamento de termos
$e = \{t_i, t_j\}$: par de termos

Saída:
$P_t = \{G_1^t, G_2^t, ..., G_n^t\}$: agrupamento de termos atualizado

1 $C_e \leftarrow w(e)$; `// obter centroide associado à` e (ver Equação 4.1)
2 **se** $|P_t| < MaxSubGrupos$ **então**
3 $\quad G_{new}^t \leftarrow$ criar novo grupo de termos em P_t;
4 $\quad$ alocar C_e em G_{new}^t;
5 **senão**
$\quad$ `// procurando pelo grupo de termos mais próximo`
6 $\quad MinDist \leftarrow +\infty$;
7 $\quad G_{sel}^t \leftarrow \emptyset$;
8 $\quad$ **para cada** *grupo* $G_i^t \in P_t$ **faça**
9 $\quad\quad Dist \leftarrow d(C_e, G_i^t)$;
10 $\quad\quad$ **se** $Dist < MinDist$ **então**
11 $\quad\quad\quad MinDist \leftarrow Dist$;
12 $\quad\quad\quad G_{sel}^t \leftarrow G_i^t$;
13 $\quad\quad$ **fim**
14 $\quad$ **fim**
15 $\quad$ alocar C_e no grupo G_{sel}^t;
16 **fim**

- Nas linhas 22 à 25 do Algoritmo 7, é realizado o processo de agrupamento incremental de documentos. Os documentos apresentados ao IHTC são mantidos em um *buffer* até que haja uma atualização da rede. No Algoritmo 9, a execução do agrupamento incremental de documentos é apresentada com mais detalhes. Para cada novo documento x, calcula-se a dissimilaridade entre x e os grupos de termos existentes (Linhas 1 à 9 do Algorimo 9). O grupo de termo mais próximo é selecionado e o documento é mapeado ao seu grupo correspondente (Linhas 10 e 11 do Algoritmo 9).

- A extração da hierarquia de tópicos é realizada conforme o valor da função "GerarHierarquiaTopicos" (Linha 26 do Algoritmo 7). Esta função pode ser configurada para atender uma requisição do usuário ou ser executada em períodos predeterminados. Assim, um algoritmo de agrupamento hierárquico é aplicado na representação condensada da coleção textual (Linha 26). Em seguida, um processo de seleção de descritores é executado nos grupos da hierarquia. Conforme pode-se observar na Linha 27, a seleção de descritores utiliza o agrupamento de termos, ou seja, os candidatos a descritores para a formação dos tópicos.

Algoritmo 9: IHTC ⇒ Agrupamento de documentos

Entrada:

$P_d = \{G_1^d, G_2^d, ..., G_n^d\}$: agrupamento de documentos
$P_t = \{G_1^t, G_2^t, ..., G_n^t\}$: agrupamento de termos
x: documento no espaço-vetorial

Saída:

$P_d = \{G_1^d, G_2^d, ..., G_n^d\}$: agrupamento de documentos atualizado

```
// procurando pelo grupo de termos mais próximo ao documento
```
1 $MinDist \leftarrow +\infty$;
2 $G_{sel}^t \leftarrow \emptyset$;
3 **para cada** *grupo* $G_i^t \in P_t$ **faça**
4 $\quad Dist \leftarrow d(x, G_i^t)$;
5 $\quad$ **se** $Dist < MinDist$ **então**
6 $\quad\quad MinDist \leftarrow Dist$;
7 $\quad\quad G_{sel}^t \leftarrow G_i^t$;
8 $\quad$ **fim**
9 **fim**

```
// mapeando x conforme grupo de termos selecionado
```
10 $G_{sel}^d \leftarrow$ grupo de documentos relacionado ao grupo de termos G_{sel}^t;
11 alocar x no grupo G_{sel}^d;

4.5 Análise de Complexidade

A complexidade global do algoritmo *IHTC* pode ser calculada como $O(G + S + H)$, em que G é a complexidade para a manutenção da rede de coocorrência e o agrupamento incremental de termos (Linhas 8 à 21 do Algoritmo 7), S é a complexidade para o agrupamento incremental de documentos (Linhas 22 à 25 do Algoritmo 7) e H é a complexidade da extração de hierarquia de tópicos (Linhas 26 à 29 do Algoritmo 7). Para a análise da complexidade, considere os seguinte itens:

- **n:** número de documentos da coleção textual;
- **k:** número máximo de subgrupos para a representação condensada da coleção textual (o mesmo valor de $MaxSubGrupos$);
- **d:** número de termos (atributos) para representação dos documentos no espaço-vetorial, ou seja, o número de palavras-chave selecionadas para cada documento; e
- **r:** número máximo de arestas para a rede de coocorrência de termos.

A manutenção da rede de coocorrência e o agrupamento incremental de termos tem complexidade de tempo $G = O(n \cdot d^2 \cdot r \cdot k)$. Nesse cálculo, d^2 é o número de operações para obter um conjunto de diferentes pares de termos de um documento (Linha 8 do Algoritmo 7). Cada par de termos, no pior caso, precisa percorrer todos os r contadores da lista para estimar os valores de coocorrência. Os pares de termos são apresentados ao algoritmo de agrupamento incremental (Algoritmo 8), percorrendo k subgrupos para selecionar o mais próximo. Esse processo é realizado com os n documentos da coleção textual.

No agrupamento incremental de documentos (Algoritmo 9), a complexidade é de $S = O(n \cdot k)$, no qual k é o numero de comparações para localizar o grupo mais próximo para cada documento.

A extração da hierarquia de tópicos é realizada por meio de um agrupamento hierárquico sobre k subgrupos (representação condensada). Como o algoritmo de agrupamento hierárquico tem complexidade quadrática, a complexidade desta parte é de $H = O(n \cdot k^2)$, pois no pior caso a hierarquia de tópicos é extraída para cada novo documento apresentado ao algoritmo.

Considerando um cenário com coleções textuais dinâmicas, no qual o número de documentos pode crescer indefinidamente ao longo do tempo, a complexidade do algoritmo varia em termo de n, enquanto k, d e r são constantes. Assim, quando $n \gg k$, a complexidade de tempo final do IHTC será linear em relação ao número de documentos.

4.6 Prova de Conceito

Com o objetivo de analisar os resultados obtidos com o IHTC, foi desenvolvido um protótipo de prova de conceito. Neste protótipo, o objetivo é aplicar o IHTC para construir hierarquias de tópicos a partir de resultados recuperados em sistemas de busca. Esse tipo de aplicação é interessante para verificar se o modelo teórico produz resultados úteis na prática, auxiliando na busca exploratória dos resultados.

Neste protótipo para organização de resultados de sistemas de buscas, o IHTC foi integrado a um sistema de consultas de documentos do tipo PDF, fornecido pela empresa Google[1]. Inicialmente, o usuário informa as palavras-chave que descrevem o conteúdo a ser recuperado. As palavras-chave são submetidas a uma ferramenta de busca integrada ao Google e no máximo 1000 títulos de documentos PDF podem ser retornados, por meio de 10 consultas sucessivas que retornam 100 títulos de documentos cada. As listas com os resultados são submetidas ao IHTC, em que são considerados apenas os termos dos títulos dos documentos. Na Figura 4.4 é ilustrado o resultado final de uma consulta com o termo "linux", o agrupamento hierárquico obtido e respectivos descritores, selecionados a partir da rede de coocorrência de termos. Nesta figura, os tópicos de primeiro nível da hierarquia são exibidos à esquerda, enquanto os documentos (páginas web) de cada tópico selecionado são exibidos à direita.

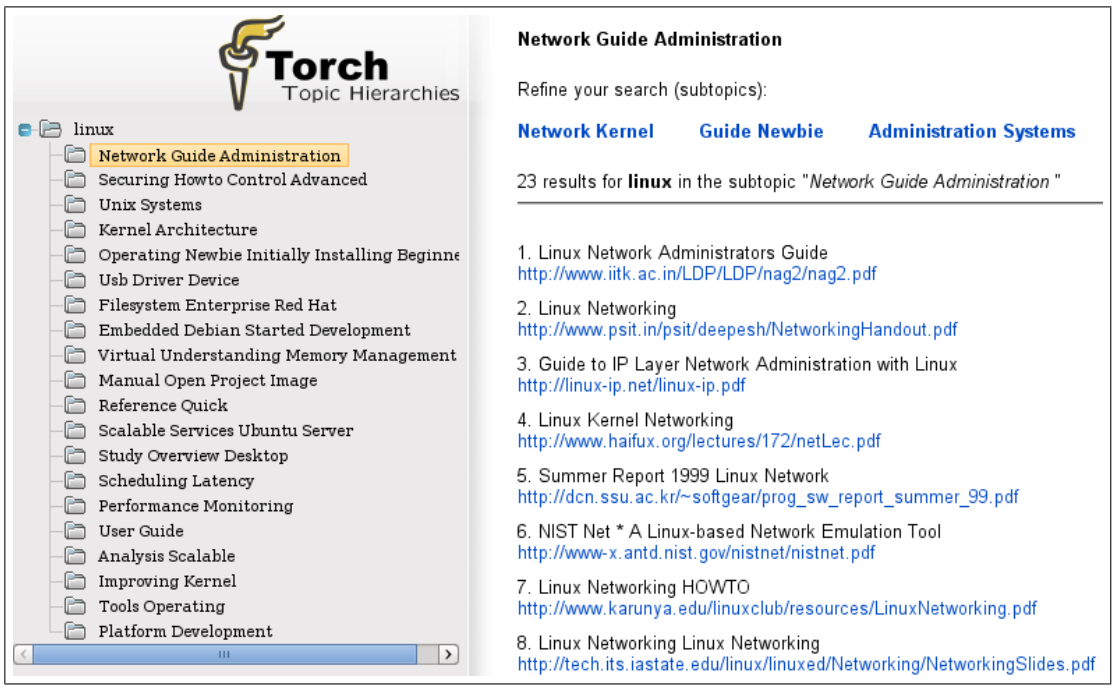

Figura 4.4: Exemplo de hierarquias de tópicos com uso do IHTC na organização de resultados de busca

[1]Google: http://www.google.com

A cada tópico selecionado, são apresentados os respectivos subtópicos para que o usuário refine a consulta. É importante observar que, neste protótipo, aplica-se a técnica de *stemming* sobre os descritores. No entanto, no momento da visualização dos tópicos, os "*stems*" são substituídos pelas palavras originais.

Foram realizados seguidos testes com várias palavras-chave relacionadas à computação em geral. As hierarquias de tópicos finais obtidas a partir do método IHTC, em geral, foram satisfatórias na organização dos temas. Os descritores selecionados por meio da rede de coocorrência de termos, auxiliam na interpretação dos resultados e análise exploratória. Entretanto, essa é uma análise subjetiva e, no próximo capítulo, é apresentada uma avaliação experimental baseada em métricas objetivas comparadas por meio de testes de significância estatística.

Enfim, o objetivo deste protótipo é permitir a integração dos conceitos com aplicações práticas, ou seja, realizar provas de conceitos como forma de complementar as avaliações experimentais. O protótipo está disponível online no endereço `http://sites.labic.icmc.usp.br/marcacini/pdf_search`.

4.7 Considerações Finais

Neste capítulo foi apresentado o método IHTC, suas fases e o algoritmo que permite realizar todo o processo de maneira incremental. O IHTC explora o agrupamento incremental de termos para organização de coleções textuais, motivado pela facilidade de interpretação dos resultados.

A representação condensada dos textos obtido pelo IHTC possui um diferencial, em que os grupos de documentos possuem grupos de termos relacionados. Dessa forma, os grupos de termos são utilizados como candidatos a descritores para a hierarquia de tópicos. Os descritores são provenientes da rede de coocorrência de termos, que tem característica de identificar as relações mais significativas. Uma análise preliminar, baseada na prova de conceito apresentada, indica que os descritores fornecidos automaticamente pelo IHTC auxiliam os usuários na interpretação do agrupamento. Outros experimentos realizados indicam que o método proposto é eficaz para apoiar o agrupamento hierárquico de documentos, sendo competitivo em relação aos algoritmos não incrementais (Marcacini e Rezende, 2010a).

No próximo capítulo é apresentada uma avaliação experimental do processo de aprendizado de hierarquias de tópicos a partir de coleções textuais dinâmicas. Na avaliação são utilizados os algoritmos de agrupamento incremental apresentados no Capítulo 3 e o IHTC, proposto neste projeto de mestrado. Ainda, as hierarquias de tópicos construídas com base em agrupamento incremental também são comparadas com hierarquias de

tópicos construídas por métodos de agrupamento não incremental.

Capítulo

5

Avaliação Experimental

Neste capítulo, o processo de aprendizado de hierarquias de tópicos a partir de coleções textuais dinâmicas, com uso de algoritmos de agrupamento incremental, é avaliado experimentalmente. A avaliação experimental, neste trabalho, é focada em três pontos:

- Comparar os algoritmos de agrupamento incremental Leader, DCTree, Buckshot e IHTC, na tarefa de obtenção da representação condensada dos textos;
- Analisar a eficácia de recuperação dos agrupamentos hierárquicos construídos a partir das representações condensadas dos textos. Ainda, compará-los com uma estratégia de agrupamento hierárquico não incremental; e
- Analisar o desempenho dos descritores para recuperar os documentos dos grupos. Neste caso, o objetivo é comparar a seleção de descritores obtidos pelo IHTC, proposto neste trabalho, com a seleção de descritores baseada em centroides.

Nas próximas seções, são apresentadas as coleções textuais a serem utilizadas na avaliação experimental. Em seguida, é descrita a configuração dos experimentos, com o ajuste de parâmetros dos algoritmos e os critérios de avaliação. Por fim, são apresentados os experimentos realizados e uma análise dos resultados.

5.1 Coleções de Textos

Para a avaliação experimental, foi utilizado um total de oito coleções textuais de diferentes tamanhos e características. Uma visão geral das características das coleções textuais é apresentada na Tabela 5.1.

Tabela 5.1: Características das coleções textuais utilizadas na avaliação experimental.

Coleção de Textos	Origem	#Documentos	#Termos	#Categorias	Referência
20ng	Mensagens de e-mail da Usenet	18828	32816	20	Rennie (2008)
acm	Biblioteca Digital da ACM	3498	6361	40	Rossi (2010)
Hitech	Artigos do San Jose Mercury (TREC)	2301	7402	6	Karypis (2003)
LATimes	Artigos do Los Angeles Times (TREC)	6279	10241	6	Karypis (2003)
NSF	National Science Foundation	10524	7102	16	Pazzani e Meyers (2003)
RE8	Subcoleção (RE8) da Reuters-21578	7674	11148	8	Pang (2010)
Reviews	Artigos do San Jose Mercury (TREC)	4069	10635	5	Karypis (2003)
WebACE	Sites do Yahoo! Directory	3900	8642	21	Boley et al. (1999)

A menor coleção textual possui 2301 documentos enquanto a maior possui 18828. As coleções são provenientes de diversas fontes e já foram utilizadas em outros trabalhos sobre agrupamento hierárquico de documentos (Steinbach et al., 2000; Zhao e Karypis, 2002; Zhao et al., 2005).

A coleção **20ng** é composta por mensagens de e-mail organizadas em 20 listas de discussão (Rennie, 2008). Essa coleção é geralmente utilizada em tarefas de categorização e agrupamento de textos.

A coleção **acm** (Rossi, 2010) é composta por artigos (*proceedings*) científicos de computação coletados da Biblioteca Digital da ACM[1]. A coleção é organizada em 40 categorias e cada categoria representa conferências de um tema da computação.

As coleções **Hitech** e **Reviews** são notícias do jornal *San Jose Mercury*[2]. Ambas foram coletadas e distribuídas como parte da *Text REtrieval Conference (TREC)*[3] e disponibilizadas para tarefas de agrupamento por Karypis (2003). As notícias são organizadas por temas que representam as categorias da coleção. A coleção **Hitech** é organizada em seis categorias: *computers*, *electronics*, *health*, *medical*, *research* e *technology*. Já a coleção **Reviews** contém 5 categorias: *food*, *movies*, *music*, *radio* e *restaurants*.

A coleção **LATimes** é composta por artigos do jornal *Los Angeles Times*, também coletada e distribuída como parte da *TREC*. Essa coleção é organizada em seis categorias: *entertainment*, *financial*, *foreign*, *metro*, *national* e *sports*.

A coleção **NSF** possui documentos públicos que descrevem projetos científicos submetidos para a *National Science Foundation (USA)*[4] (Pazzani e Meyers, 2003). Cada documento é composto por um título e um pequeno resumo de até 150 palavras. Os documentos são organizados em 16 categorias representando áreas de pesquisa, como, Estatística, Biologia e Computação.

A coleção **RE8** é composta por notícias de jornal, baseada em outra coleção de textos bem conhecida: Reuters-21578 (Lewis, 1997). A RE8 é disponibilizada pelo *CSMining*

[1]Biblioteca Digital da ACM: http://portal.acm.org/
[2]San Jose Mercury: http://www.mercurynews.com/
[3]Text REtrieval Conference (TREC): http://trec.nist.gov/
[4]National Science Foundation (USA): http://www.nsf.gov/

Group (Pang, 2010) e está organizada em oito categorias: *acq*, *crude*, *earn*, *grain*, *interest*, *money-fx*, *ship* e *trade*.

Por fim, a coleção **WebACE** é composta por páginas de internet do *Yahoo! Directory*[5] (Boley et al., 1999). As páginas são organizadas em 21 categorias, ou seja, diretórios que reúnem páginas de um mesmo tema.

É possível observar que as coleções textuais descritas possuem uma organização predefinida em categorias. O uso deste conhecimento *apriori* permite avaliar os resultados dos experimentos por meio de medidas supervisionadas de agrupamento, como a FScore e Entropia, conforme descrito na Seção 2.4 do Capítulo 2. As oito coleções textuais estão disponíveis para *download* no endereço http://sites.labic.icmc.usp.br/marcacini/ihtc.

5.2 Configuração dos Experimentos

A avaliação de métodos não supervisionados para aprendizado de hierarquias de tópicos, neste trabalho, é baseada em um processo de Mineração de Textos. Mais especificamente, o objetivo é avaliar os resultados obtidos na etapa de extração de padrões, ou seja, os algoritmos de agrupamento hierárquico e a seleção de descritores para os grupos.

Para realizar essas avaliações, antes é necessário um procedimento de configuração dos experimentos para definição dos parâmetros envolvidos no processo. No contexto deste trabalho, é preciso definir a técnica de pré-processamento dos textos, os parâmetros dos algoritmos de agrupamento, a técnica de seleção de descritores para o agrupamento e, por fim, os critérios a serem utilizados na avaliação dos resultados.

Para a configuração dos experimentos, neste trabalho, o pré-processamento dos textos deve ser adequado para o cenário incremental. Na extração de padrões são utilizados os algoritmos de agrupamento incremental Leader, Buckshot, DCTree e IHTC para obtenção das representações condensadas dos textos. O agrupamento hierárquico a partir das representações condensadas é realizado por meio do Bisecting k-means, seguido de um processo de seleção de descritores. Por fim, no pós-processamento, a avaliação dos resultados é baseada em três critérios, visando comparar a qualidade das representações condensadas dos textos, o agrupamento hierárquico e os respectivos descritores selecionados para os grupos. Uma visão geral da configuração dos experimentos realizada é ilustrada na Figura 5.1.

Nas próximas seções, cada ponto da configuração dos experimentos será descrito com mais detalhes.

[5]Yahoo! Directory: http://dir.yahoo.com/

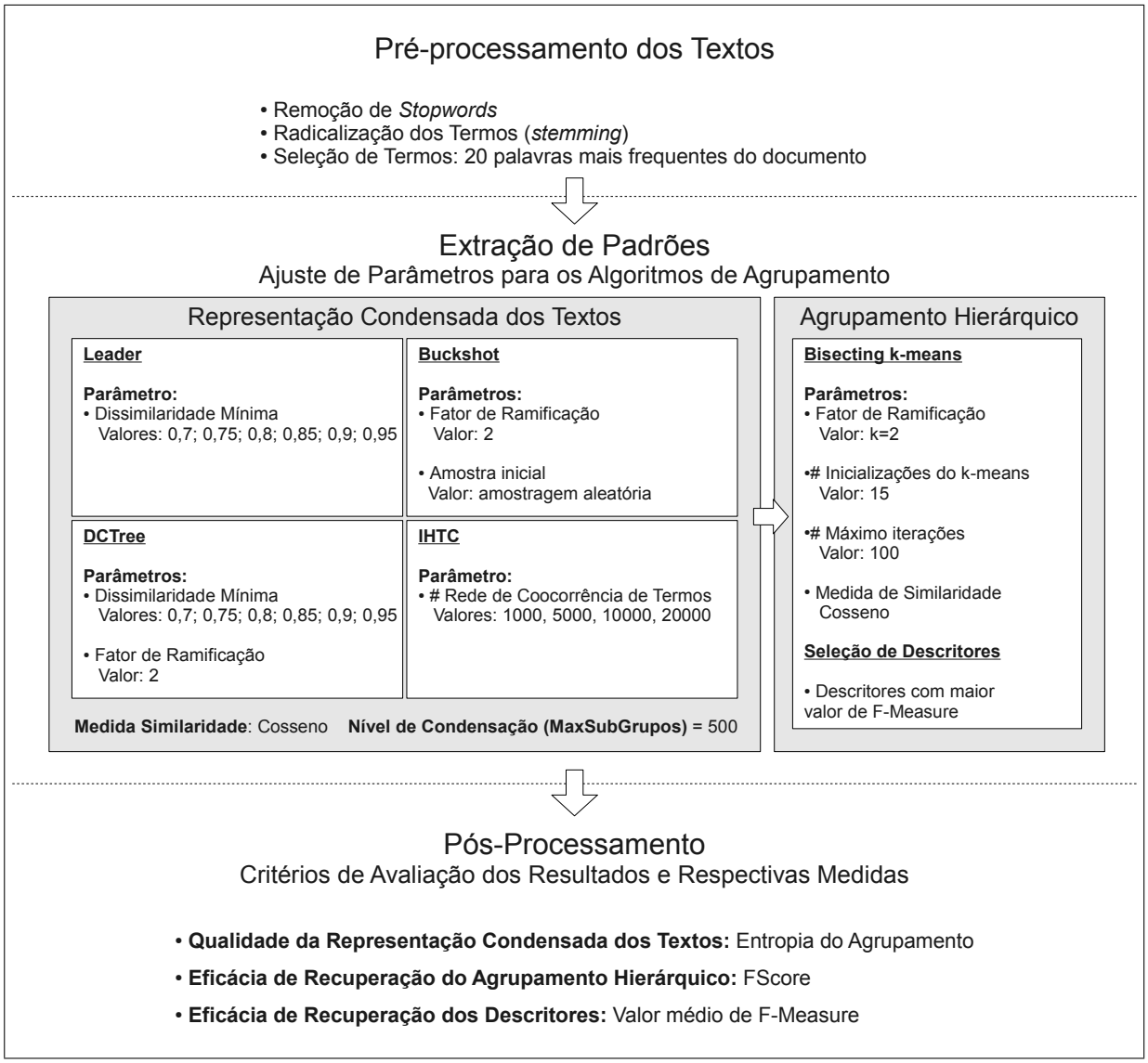

Figura 5.1: Visão geral da configuração dos experimentos

5.2.1 Pré-processamento de Textos

No **pré-processamento dos textos** em cenários dinâmicos, os documentos são pré-processados individualmente, conforme são apresentados ao algoritmo de agrupamento incremental. Para cada novo documento, aplica-se a remoção de *stopwords* e a técnica de *stemming* para radicalização dos termos. Em seguida, os 20 termos mais frequentes do documento são selecionados para representá-lo no modelo espaço-vetorial. O número de termos selecionados é baseado em resultados de outros trabalhos da literatura (Koller e Sahami, 1997; Chang e Hsu, 2005), que indica que o uso de 10 à 25 palavras-chave geralmente são suficientes para representar o conteúdo de um documento, tanto em tarefas de agrupamento quanto de classificação.

Como o pré-processamento dos textos para agrupamento incremental é realizado con-

forme novos documentos são inseridos na coleção, na Tabela 5.2 é apresentado o resultado do pré-processamento no final do processo de aprendizado.

Tabela 5.2: Resultado do pré-processamento de textos

Coleção de Textos	# Termos Total	# Termos Selecionados	# Termos com DF $\geq$ 2
20ng	93905	33235	46479
acm	224014	6373	59696
Hitech	21935	7402	12829
LATimes	15930	10241	9933
NSF	7102	7102	3893
RE8	17320	11148	8887
Reviews	35926	10635	22664
WebACE	11503	8642	8575

Na coluna "# Termos Total" é apresentado o número de termos da coleção textual, após a remoção de *stopwords* e aplicação de *stemming*. Já a coluna "# Termos Selecionados" indica o número de termos que foram selecionados com a técnica de pré-processamento, contabilizados no final do processo de aprendizado da hierarquia de tópicos. É importante observar que o processo de aprendizado é incremental e, assim, o número de termos pode alterar conforme novos documentos são apresentados. No entanto, como este é um experimento controlado, a tarefa de aprendizado é concluída quando todos os documentos da coleção forem agrupados. Na coluna "# Termos com DF $\geq$ 2" é indicado o número de termos que ocorrem em dois ou mais documentos (após remoção de *stopwords* e aplicação de *stemming*), ou seja, os termos que podem ser úteis para a tarefa de agrupamento. O número de termos com DF $\geq$ 2 foi incluído apenas para efeito de comparação, pois é um critério usualmente utilizado em seleção de termos para tarefas de agrupamento.

5.2.2 Ajuste de Parâmetros dos Algoritmos de Agrupamento

Um dos objetivos da avaliação experimental, neste trabalho, é comparar o desempenho de diferentes algoritmos de agrupamento. Alguns algoritmos de agrupamento incremental, como o Leader, DCTree e o IHTC, possuem parâmetros que devem ser ajustados para cada coleção de textos. Assim, para permitir uma comparação justa, foi executado um processo de seleção de parâmetros para estes algoritmos. Em todas as execuções, foi utilizada a **medida de similaridade Cosseno** adaptada para ser utilizada como medida de dissimilaridade (Seção 2.3.1 do Capítulo 2).

O ajuste de parâmetro de cada algoritmo foi apoiado pelo índice de Silhueta, apresentado na Seção 2.4.1 do Capítulo 2. O valor de Silhueta pode ser utilizado para selecionar, dentre vários agrupamentos, o agrupamento que melhor se ajusta aos dados conforme

a coesão interna dos grupos (documentos de um mesmo grupo devem ser similares) e separação entre os grupos (documentos em grupos distintos devem ser dissimilares). Por ser uma medida de validação não supervisionada, é uma maneira prática para ajuste de parâmetros em cenários reais, pois não exige informação externa a respeito dos dados. Uma forma usual de aplicação da medida Silhueta é selecionar, por exemplo, o melhor valor k (parâmetro) do algoritmo *k-means*, ou seja, determinar o número de grupos.

Para exemplificar o procedimento adotado para ajuste de parâmetros, considere o caso do algoritmo Leader. O Leader possui o parâmetro de Dissimilaridade Mínima para formação dos grupos e foram utilizadas 6 variações para este parâmetro: 0,70; 0,75; 0,80; 0,85; 0,90; e 0,95. Assim, para cada base de dados, são gerados 6 agrupamentos usando cada parâmetro e a medida de Silhueta é calculada em cada um desses agrupamentos. O agrupamento com maior valor de Silhueta (que varia de -1 à 1) indica o melhor parâmetro (dentre os analisados) para ser utilizado em determinada coleção textual. Na Figura 5.2 é ilustrada a análise do parâmetro Dissimilaridade Mínima do algoritmo Leader na coleção textual 20ng. Neste caso, o valor selecionado para a coleção **20ng** é "Dissimilaridade Mínima = 0,80", pois apresentou maior valor de Silhueta.

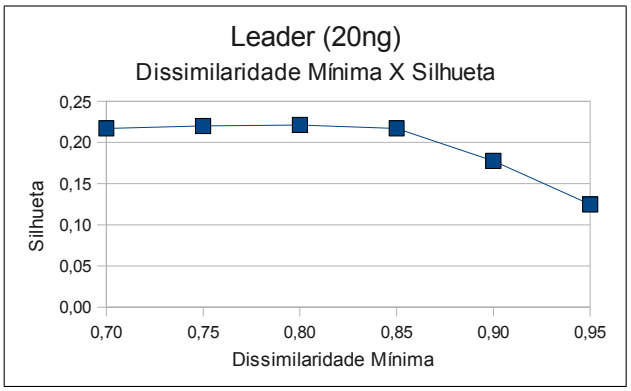

Figura 5.2: Gráfico para análise de parâmetro do algoritmo Leader na coleção textual **20ng**

Na Tabela 5.3 são apresentados os valores de parâmetros analisados em cada algoritmo de agrupamento incremental.

Tabela 5.3: Parâmetros analisados em cada algoritmo de agrupamento incremental

Algoritmo	**Parâmetros**
Leader	Dissimilaridade Mínima = {0,7; 0,75; 0,8; 0,85; 0,9; 0,95}
Buckshot	Fator de Ramificação = {2}
DCTree	Fator de Ramificação = {2} Dissimilaridade Mínima = {0,7; 0,75; 0,8; 0,85; 0,9; 0,95}
IHTC	# Rede de Coocorrência de Termos = {1000, 5000, 10000, 20000}

Um ponto importante é a definição do número máximo de grupos para a representação condensada dos dados (*MaxSubGrupos*). Para todas as coleções de textos, foi utilizado o valor 500 para o número máximo de grupos. Dessa forma, é possível obter um nível de condensação diferente para cada coleção textual, uma vez que as coleções tem diferentes número de documentos. Em um caso real, este valor é definido conforme a memória disponível para a aplicação, uma vez que o tamanho das coleções textuais em cenários dinâmicos é indefinido.

Assim como foi exemplificado a análise de parâmetro para o algoritmo Leader (Figura 5.2), nas Figuras 5.3 e 5.4 é ilustrada a análise de parâmetros para algoritmo DCTree e IHTC, respectivamente, na coleção textual **20ng**. A análise do parâmetro é similar, ou seja, busca-se o valor do parâmetro que maximize o valor de Silhueta.

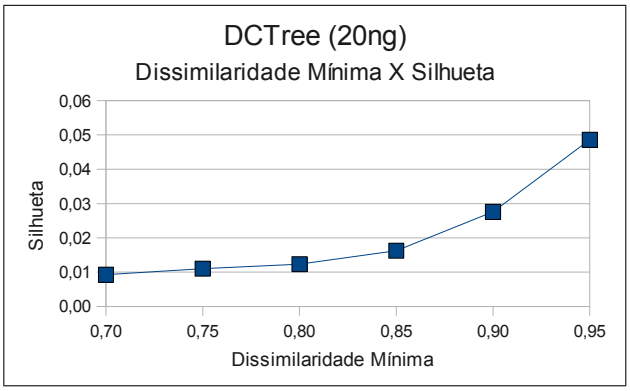

Figura 5.3: Gráfico para análise de parâmetro do algoritmo DCTree na coleção textual 20ng

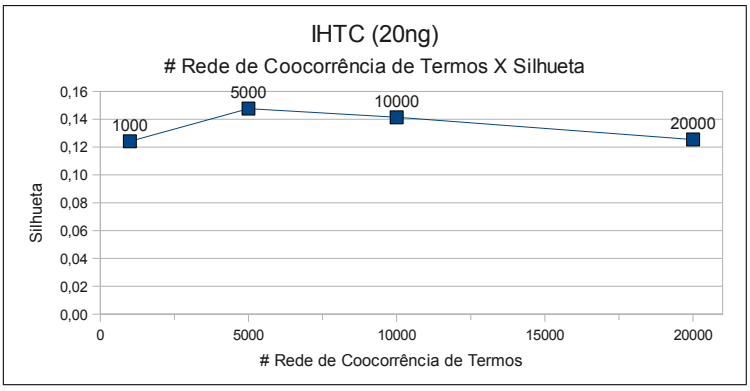

Figura 5.4: Gráfico para análise de parâmetro do algoritmo IHTC na coleção textual 20ng

Os algoritmos Leader, DCTree e IHTC foram executados em todas as coleções textuais, variando-se os valores de parâmetros conforme a Tabela 5.3. Para diminuir a influência da ordem de apresentação dos documentos durante o agrupamento incremental, selecionou-se o melhor resultado de 100 execuções com diferentes ordens de apresentação dos documen-

tos no agrupamento. Na Tabela 5.4 são apresentados os valores de parâmetros selecionados para cada algoritmo nas coleções textuais.

Tabela 5.4: Parâmetros selecionados para os algoritmos de agrupamento incremental em cada coleção textual.

	Leader	**Buckshot**	**DCTree**		**IHTC**
	Dissimilaridade Mínima	*Fator de Ramificação*	*Dissimilaridade Mínima*	*Fator de Ramificação*	*# Rede de Coocorrência de Termos*
20ng	0,80	2	0,95	2	5000
acm	0,70	2	0,75	2	20000
Hitech	0,70	2	0,85	2	20000
LATimes	0,70	2	0,85	2	20000
NSF	0,70	2	0,90	2	20000
RE8	0,70	2	0,80	2	20000
Reviews	0,70	2	0,85	2	20000
WebACE	0,70	2	0,80	2	20000

Todos os gráficos referentes à análise de parâmetros estão disponíveis no Apêndice A (Leader), Apêndice B (DCTree) e Apêndice C (IHTC).

Os algoritmos de agrupamento incremental Leader, Buckshot, DCTree e IHTC são utilizados para obter a representação condensada dos textos. O algoritmo de agrupamento hierárquico aplicado a partir da representação condensada dos textos é o Bisecting k-means, conhecido por ser um dos mais eficazes em dados textuais (Steinbach et al., 2000). Neste algoritmo, foram executadas 15 inicializações do k-means ($k = 2$) e, em cada execução, o número máximo de iterações é 100, conforme recomendado na literatura (Zhao et al., 2005).

5.2.3 Seleção de Descritores para o Agrupamento

Durante o aprendizado de hierarquias de tópicos, é necessário selecionar os descritores para o agrupamento hierárquico obtido. Nesta avaliação experimental, utilizou-se o índice F-Measure para avaliar os termos mais importantes de um grupo, conforme descrito na Seção 2.3.3 do Capítulo 2.

Nos agrupamentos hierárquicos obtidos a partir das representações condensadas do algoritmo Leader, Buckshot e DCTree, definiu-se que os candidatos a descritores dos grupos são os termos presentes no centroide de cada grupo. No caso do IHTC, os candidatos a descritores são identificados automaticamente durante o processo de agrupamento incremental.

5.2.4 Critérios de Avaliação

A avaliação do processo de aprendizado não supervisionado de hierarquias de tópicos, neste trabalho, é baseada em três critérios: (1) qualidade da representação condensada

dos dados, (2) eficácia de recuperação do agrupamento hierárquico de documentos; e (3) eficácia de recuperação dos descritores selecionados para o agrupamento. Cada critério analisa uma etapa da construção da hierarquia de tópicos e permite comparar os diferentes algoritmos utilizados.

A **qualidade da representação condensada dos textos** é, na prática, a qualidade do agrupamento obtido pelos algoritmos de agrupamento incremental. Neste trabalho, utilizou-se o índice de Entropia dos grupos (Seção 2.4.2 do Capítulo 2), que é uma medida supervisionada que analisa a desordem interior de cada grupo de acordo com as categorias predeterminadas de cada documento. A ideia é que se o valor global de Entropia do agrupamento for baixo, então cada grupo é composto predominantemente por documentos de uma mesma categoria e, dessa forma, a qualidade da condensação é considerada alta.

Em relação à **eficácia de recuperação do agrupamento hierárquico de documentos**, espera-se medir o quão a organização hierárquica dos documentos reflete a organização conhecida *apriori* de acordo com as categorias. Para tal, utilizou-se o índice FScore (Seção 2.4.3 do Capítulo 2) de maneira similar à avaliação de algoritmos de agrupamento hierárquico realizada por Zhao e Karypis (2002).

As medidas de Entropia (para agrupamento particional) e FScore (para agrupamento hierárquico) são descritas como medidas supervisionadas para avaliação de agrupamentos no livro *Introduction to Data Mining* (Tan et al., 2005), Capítulo 8 - *Cluster Analysis: Basic Concepts and Algorithms*, Seção 8.5.7 - *Supervised Measures of Cluster Validity*. Ainda, outros trabalhos da literatura tem empregado essas medidas para a avaliação de agrupamento, especialmente em tarefas de organização de coleções de documentos (Steinbach et al., 2000; Zhao e Karypis, 2002; Zhao et al., 2005).

Finalmente, a **eficácia de recuperação dos descritores selecionados para o agrupamento** tem como objetivo verificar o quanto (de acordo com a F-Measure) os descritores dos grupos auxiliam na interpretação do agrupamento. Dessa forma, dado um agrupamento hierárquico, o valor global deste critério é calculado como uma média dos valores de F-Measure dos $j = 3$ melhores termos (Seção 2.3.3 do Capítulo 2).

5.3 Experimentos Realizados e Análise dos Resultados

Uma vez realizada a configuração dos experimentos, foi executado o processo de aprendizado de hierarquias de tópicos nas oito coleções textuais apresentadas na Tabela 5.1. Para tal, foi utilizada a ferramenta *Torch - Topic Hierarchies* (Marcacini e Rezende, 2010b), desenvolvida durante este projeto de mestrado, que disponibiliza vários algoritmos de agrupamento incremental e agrupamento hierárquico, além de técnicas para pré-processamento de textos, avaliação e exploração visual de hierarquias de tópicos. Para

realização de testes de significância estatística e construção dos rankings de desempenho dos algoritmos, foi utilizado o software *KEEL Data Mining* (Alcalá et al., 2011).

Os parâmetros selecionados para cada algoritmo de agrupamento nas coleções de textos, apresentados na Tabela 5.4, foram utilizados neste processo de aprendizado de hierarquia de tópicos. Novamente, para diminuir o efeito da ordem de apresentação dos documentos durante o agrupamento incremental, o processo de aprendizado foi repetido algumas dezenas de vezes e os melhores resultados foram selecionados.

Nas próximas seções são apresentados e analisados os resultados obtidos com os experimentos. Os algoritmos discutidos durante este trabalho são avaliados e comparados de acordo com os três critérios apresentados anteriormente: (1) qualidade da representação condensada dos dados, (2) eficácia de recuperação do agrupamento hierárquico de documentos; e (3) eficácia de recuperação dos descritores selecionados para o agrupamento.

5.3.1 Qualidade da Representação Condensada dos Textos

Os valores de Entropia global obtidos por cada algoritmo de agrupamento incremental são apresentados na Tabela 5.5. O algoritmo IHTC obteve os menores valores de Entropia em cinco de oito coleções textuais, ficando em primeiro no *ranking*. Na Figura 5.5, estas informações são apresentadas em forma de gráfico de barras para facilitar a análise.

Tabela 5.5: Valores de Entropia obtidos pelos algoritmos de agrupamento incremental em oito coleções de textos.

	Leader	**Buckshot**	**DCTree**	**IHTC**
20ng	**2,091***	2,902	2,639	2,266
acm	2,100	2,339	2,019	**1,834***
Hitech	**0,811***	0,939	0,851	0,920
LATimes	1,245	1,324	1,216	**1,123***
NSF	1,674	2,113	2,000	**1,548***
RE8	0,430	0,469	0,427	**0,347***
Reviews	0,572	0,526	**0,505***	0,551
WebACE	0,924	1,097	0,936	**0,815***
Ranking	***2,375***	***3,750***	***2,250***	***1,625***
Posição	***3***	***4***	***2***	***1***

Para afastar a hipótese de que as diferenças entre os algoritmos sejam meramente aleatórias, foi aplicado o teste de significância estatística de Friedman com o pós-teste de Nemenyi, conforme apresentado em Demšar (2006). Neste teste é possível comparar o desempenho de múltiplos algoritmos em múltiplas coleções de textos. Os resultados do teste são apresentados na Tabela 5.6. Na matriz triangular superior estão dispostos os valores de $p - value$ da comparação entre dois algoritmos (linha X coluna). Na matriz

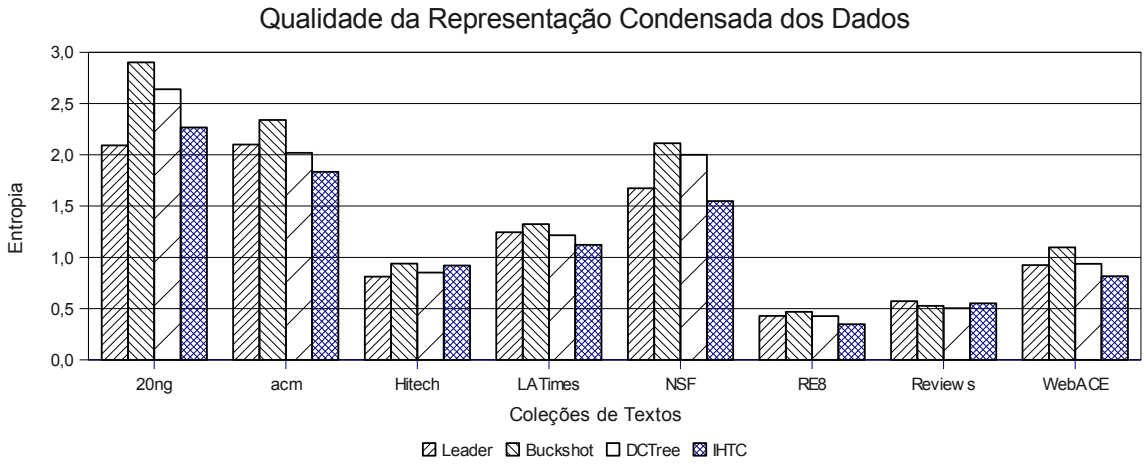

Figura 5.5: Representação dos valores de Entropia obtidos pelos algoritmos de agrupamento incremental em oito coleções de textos.

triangular inferior, é ilustrado quando o algoritmo da linha é superior ($\oplus$), inferior ($\ominus$) ou não apresentar diferença estatisticamente significativa ($\odot$) em relação ao algoritmo da coluna, com nível de significância de $\alpha = 0,05$. Nesses dados, a diferença entre dois algoritmos é considerada estatisticamente significativa quando o $p - value \leq 0,0083$.

Tabela 5.6: Resultados do teste de significância estatística de acordo com os valores de Entropia obtidos pelos algoritmos de agrupamento incremental.

	Leader	**Buckshot**	**DCTree**	**IHTC**
Leader		0,0332	0,8464	0,2453
Buckshot	$\odot$		0,0201	0,0010
DCTree	$\odot$	$\odot$		0,3329
IHTC	$\odot$	$\oplus$	$\odot$	

O resultado da aplicação do teste de Nemenyi indica que somente em um caso há diferença estatisticamente significativa, no qual o algoritmo IHTC, proposto neste trabalho, apresenta melhor desempenho do que o algoritmo Buckshot. Esses resultados são apresentados na Figura 5.6, que apresenta o diagrama de diferença crítica em relação ao *ranking* dos algoritmos. Nesses diagramas, os grupos de algoritmos conectados por uma linha não apresentam diferenças estatisticamente significativas de desempenho entre si.

Dessa forma, apesar de não existir diferenças relevantes na escolha do algoritmo de agrupamento incremental, que irá obter a representação condensada dos dados, o IHTC possui a vantagem de identificar automaticamente conjuntos de descritores para o agrupamento. Esta vantagem é especialmente útil para o aprendizado de hierarquias de tópicos, uma vez que a interpretação dos resultados é uma tarefa importante para os usuários.

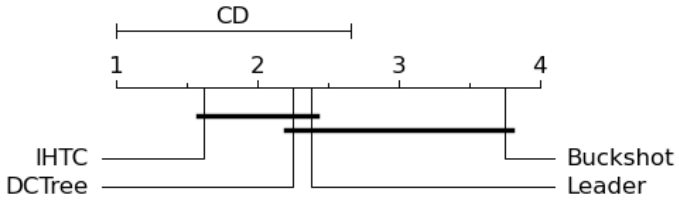

Figura 5.6: Diferença crítica (CD) sobre o *ranking* dos algoritmos de agrupamento incremental de acordo com os valores de Entropia.

Por outro lado, o IHTC tem um custo para manter a rede de coocorrência de termos, que pode aumentar o tempo computacional para agrupamento incremental, em comparação com os outros algoritmos.

O Leader e o DCTree apresentaram resultados similares. O Leader é uma boa escolha quando se está interessado em uma solução simples e com baixo custo computacional. Já o DCTree é uma alternativa razoável quando o usuário deseja explorar a representação condensada em diversos níveis de granularidade.

O Buckshot ficou em último lugar no ranking, mas não se pode afirmar que essa diferença é significativa em relação ao DCTree e ao Leader. O Buckshot é influenciado pela amostra dos dados usada na construção do agrupamento hierárquico inicial. Se a amostra não for representativa para a coleção textual, a representação condensada obtida com o agrupamento incremental fica prejudicada. Neste experimento, foram selecionados aleatoriamente subconjuntos da coleção textual para serem utilizados como amostra inicial do Buckshot. Por outro lado, o Buckshot é uma solução interessante quando a seleção da amostra pode ser feita de forma mais eficaz, por exemplo, com o auxílio de especialistas de domínio.

5.3.2 Eficácia de Recuperação do Agrupamento Hierárquico de Documentos

As representações condensadas dos dados, obtidas na etapa anterior, foram utilizadas para o agrupamento hierárquico de documentos com o algoritmo Bisecting k-means. O objetivo é obter uma organização hierárquica das coleções de forma computacionalmente mais eficiente. Por exemplo, os 18828 documentos da coleção textual 20ng são representados por 500 centroides (que são atualizados de maneira incremental), diminuindo-se os requerimentos computacionais para a execução do agrupamento hierárquico.

Nessa etapa, também foi aplicado um processo de agrupamento hierárquico sem uso da representação condensada, ou seja, foi obtida uma organização hierárquica das coleções de forma não incremental com o algoritmo Bisecting k-means, com uma execução tradicional. Assim, é possível comparar os agrupamentos hierárquicos baseados em representação condensada, explorada em cenários dinâmicos, com uma estratégia comumente aplicada em

cenários estáticos.

A medida FScore foi utilizada para avaliação do agrupamento hierárquico. Na Tabela 5.7 são apresentados os valores de FScore para os agrupamentos hierárquicos originados a partir de cada estratégia.

Tabela 5.7: Valores de FScore calculados para os agrupamentos hierárquicos.

	Bisecting k-means				
	Leader	**Buckshot**	**DCTree**	**IHTC**	***Não Incremental***
20ng	0,336	0,311	0,337	0,312	**0,339***
acm	0,345	0,323	0,330	**0,359***	0,354
Hitech	0,520	0,547	0,543	0,545	**0,576***
LATimes	**0,529***	0,454	0,496	0,461	0,491
NSF	0,447	0,381	0,406	**0,451***	0,442
RE8	0,690	0,805	0,707	0,735	**0,806***
Reviews	0,746	0,765	0,756	**0,771***	0,764
WebACE	**0,593***	0,575	0,590	0,577	0,591
Ranking	***3,125***	***3,875***	***3,375***	***2,625***	***2***
Posição	***3***	***5***	***4***	***2***	***1***

O agrupamento hierárquico obtido pelo Bisecting k-means sem uso de representação condensada ficou em primeiro no *ranking*, seguido do agrupamento hierárquico baseado na representação condensada do IHTC. Para facilitar a análise desses resultados, na Figura 5.7 é apresentado um gráfico com os valores de FScore dos agrupamentos hierárquicos em cada coleção textual. O agrupamento hierárquico obtido pelo Bisecting k-means sem uso de representação condensada é indicado como "Não Incremental" e, no gráfico, é representado por uma linha horizontal que facilita a comparação com os outros agrupamentos hierárquicos.

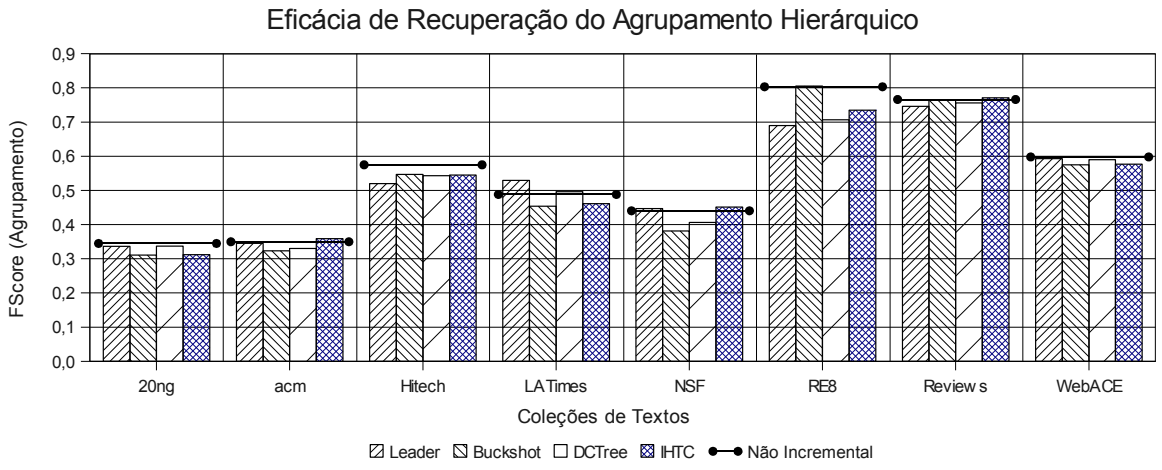

Figura 5.7: Valores de FScore calculados para os agrupamentos hierárquicos.

A eficácia de recuperação dos agrupamentos hierárquicos baseados nas representações condensadas do Leader, Buckshot, DCTree e IHTC, é análoga à qualidade das representações condensadas. Assim, os algoritmos que obtiveram um menor valor de Entropia durante o agrupamento incremental, resultaram em agrupamentos hierárquicos com maior eficácia de recuperação pela medida FScore.

Para afastar a hipótese de que as diferenças entre os valores de FScore sejam meramente aleatórias, também foi aplicado o teste de significância estatística de Friedman com o pós-teste de Nemenyi, com nível de significância de $\alpha = 0,05$. Na Tabela 5.8 é apresentado o resultado do teste de significância estatística. A leitura da tabela segue o mesmo esquema apresentado na seção anterior. Nesses dados, a diferença entre dois algoritmos é considerada estatisticamente significativa quando o $p - value \leq 0,005$.

Tabela 5.8: Resultados do teste de significância estatística de acordo com os valores de FScore calculados para os agrupamentos hierárquicos.

	Leader	**Buckshot**	**DCTree**	**IHTC**	**Não Incremental**
Leader		0,3428	0,3429	0,5271	0,1547
Buckshot	⊙		0,5271	0,1138	0,0177
DCTree	⊙	⊙		0,3428	0,0820
IHTC	⊙	⊙	⊙		0,4292
Não Incremental	⊙	⊙	⊙	⊙	

De acordo com o resultado do teste de Nemenyi, não foi possível indicar diferenças estatisticamente significativas na eficácia de recuperação dos agrupamentos hierárquicos. Isto pode ser observado no diagrama da Figura 5.8, na qual todas as estratégias estão conectadas por uma reta, ou seja, as diferenças das posições no *ranking* não podem ser consideradas estatisticamente significativas.

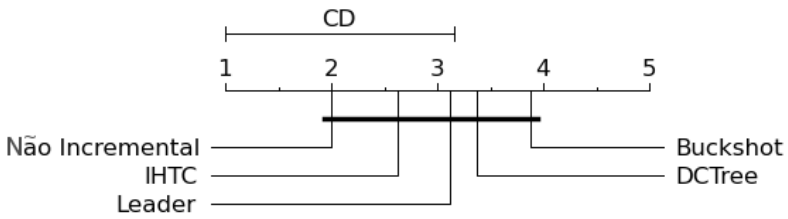

Figura 5.8: Diferença crítica (CD) sobre o *ranking* dos agrupamentos hierárquicos de acordo com os valores de FScore.

Este resultado é interessante, uma vez que a estratégia do Bisecting k-means sem uso de representação condensada dos textos (não incremental) não obteve resultado superior às estratégias incrementais, nos experimentos realizados, baseadas em representação condensada dos textos. Dessa forma, há uma evidência na direção da hipótese deste trabalho,

que diz que é possível aprender hierarquias de tópicos utilizando estratégias de agrupamento incremental com qualidade similar às hierarquias de tópicos usando estratégias não incrementais.

Os experimentos realizados indicam que o processo de aprendizado de hierarquias de tópicos a partir de coleções textuais dinâmicas, explorando as estratégias de agrupamento incremental, produz resultados satisfatórios. Ainda, há uma redução do custo computacional devido ao uso das representações condensadas dos textos para o agrupamento hierárquico de documentos.

5.3.3 Eficácia de Recuperação dos Descritores

Para avaliação da eficácia de recuperação dos descritores, foram selecionados os agrupamentos hierárquicos gerados com base nas representações condensadas do método IHTC em cada uma das coleções de textos. Em cada agrupamento hierárquico, executou-se dois processos de seleção de descritores: (1) baseada em centroides e (2) baseada na rede de coocorrência de termos. O primeiro é um método usualmente aplicado na literatura e o segundo representa os conjuntos de descritores obtidos pelo IHTC durante o agrupamento incremental.

O objetivo é analisar se, para um mesmo agrupamento hierárquico, os descritores obtidos a partir da rede de coocorrência de termos auxiliam na interpretação dos grupos. Para isto, foi utilizada a medida F-Measure e na Tabela 5.9 são apresentados os valores médios de F-Measure dos agrupamentos hierárquicos. A seleção de descritores baseada na rede de coocorrência de termos obteve maiores valores médios de F-Measure em cinco das oitos coleções textuais. Na Figura 5.9 são ilustrados os resultados por meio de um gráfico de barras para facilitar a análise.

Tabela 5.9: Valores médio de F-Measure dos descritores selecionados para os agrupamentos hierárquicos

	Baseada no Centroide	**Baseada na Rede de Coocorrência**
20ng	***0,2321**	0,1956
acm	0,2943	***0,3419**
Hitech	***0,3300**	0,2948
LATimes	0,3020	***0,3167**
NSF	0,2301	***0,2503**
RE8	0,2859	***0,3031**
Reviews	***0,3021**	0,2806
WebACE	0,3774	***0,3905**

Para afastar a hipótese de que as diferenças entre os valores médios de F-Measure sejam meramente aleatórias, foi aplicado o teste de significância estatística de Wilcoxon. Neste teste é possível comparar o desempenho entre dois algoritmos em múltiplas coleções de textos. Utilizando um nível de significância de $\alpha = 0,05$, não foi possível identificar diferenças estatisticamente significativas.

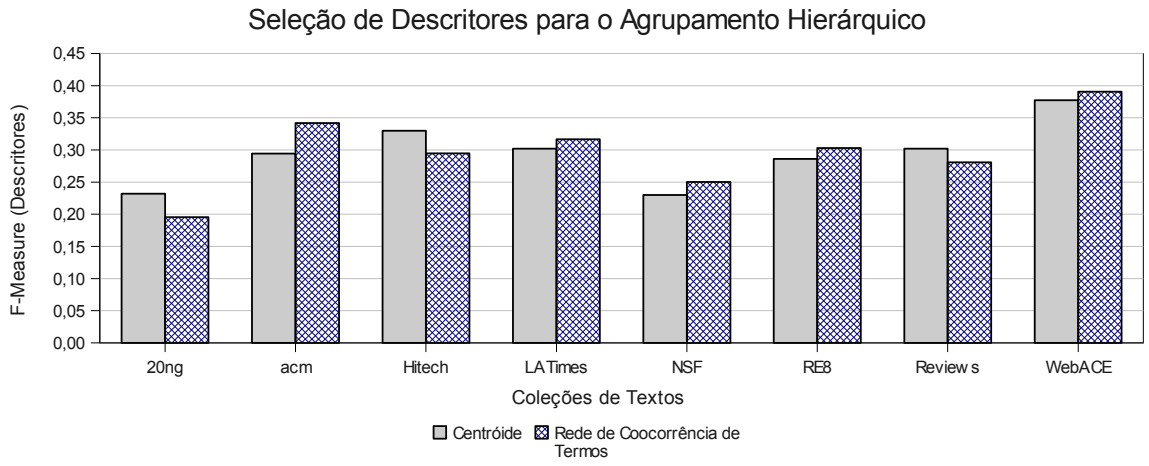

Figura 5.9: Valores médio de F-Measure dos descritores selecionados para os agrupamentos hierárquicos

Este resultado sugere que, de acordo com a medida F-Measure, o uso da rede de coocorrência de termos no IHTC tem desempenho similar na interpretação do agrupamento, quando comparado com os descritores selecionados a partir dos centroides. No entanto, uma das principais motivações do IHTC é a habilidade de obter descritores automaticamente durante o agrupamento incremental, a partir das relações na rede de coocorrência de termos. Como esta análise de "interpretação de agrupamento" foi feita apenas com medidas objetivas, na Figura 5.10 é ilustrada parte da hierarquia de tópicos extraída da coleção **acm**, que trata de artigos sobre "Rede de Computadores". A imagem contém duas estruturas hierárquicas iguais provenientes do mesmo agrupamento hierárquico com base no IHTC. O quadro da esquerda é o agrupamento hierárquico com descritores selecionados a partir da rede de coocorrência de termos, conforme proposto no IHTC, enquanto o quadro da direita é a seleção de descritores baseada nos centroides dos grupos.

É possível notar que os tópicos formados a partir da rede de coocorrência de termos são mais detalhados e fornecem uma ideia mais precisa do conteúdo dos documentos ali agrupados. No entanto, essa análise deve ser realizada com a presença de usuários e especialistas de domínio (Chung et al., 2008), que não está no escopo deste projeto de mestrado.

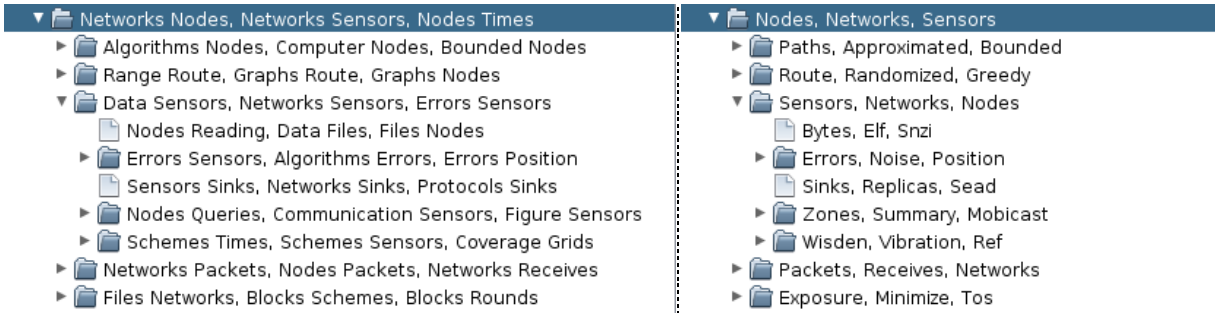

Figura 5.10: Exemplo comparativo entre a seleção de descritores baseada em rede de coocorrência de termos e a seleção de descritores baseada em centroides.

5.4 Considerações Finais

Neste capítulo foi descrita a avaliação experimental conduzida a fim de analisar o processo de aprendizado não supervisionado de hierarquias de tópicos em coleções textuais dinâmicas.

Inicialmente, foram detalhadas as características das oito coleções textuais utilizadas no experimento. Em seguida, realizou-se ajustes de parâmetros dos algoritmos de agrupamento, bem como foram definidos os critérios de avaliação dos resultados.

Foram avaliadas quatro estratégias de agrupamento incremental para obtenção da representação condensada dos dados: Leader, Buckshot, DCTree e IHTC. O IHTC, proposto neste trabalho, se mostrou competitivo em relação à qualidade da representação condensada dos dados, com a vantagem de se obter automaticamente descritores para o agrupamento formado. No geral, os resultados dos experimentos indicaram que as estratégias de agrupamento incremental obtém resultados similares e, dessa forma, analisou-se em que situações cada estratégia pode ser empregada.

Foi visto que os agrupamentos hierárquicos construídos a partir das representações condensadas, com uma estratégia incremental, possui eficácia de recuperação próxima aos agrupamentos hierárquicos construídos de forma não incremental. Na prática, este resultado sugere que o aprendizado de hierarquias de tópicos a partir de coleções textuais dinâmicas, conforme descrito neste trabalho, é satisfatório e exige menos recursos computacionais.

Por fim, analisou-se a eficácia de recuperação dos descritores dos agrupamentos. Uma das motivações do IHTC é a identificação de descritores durante o agrupamento incremental, com auxílio de uma rede de coocorrência de termos. Apesar de que com o uso do método proposto obteve-se maiores valores de eficácia de recuperação dos descritores, não foi possível identificar diferenças estatisticamente significativas. Para uma análise mais profunda em relação a este critério, é necessário um estudo com foco em experimentos

qualitativos, com presença de usuários e especialistas de domínio.

Capítulo

6

Construção Automática de Diretórios Web: Estudo Exploratório com o Projeto Dmoz

A avaliação experimental realizada no capítulo anterior possibilitou comparar as estratégias existentes na literatura e, também, avaliar o método proposto neste trabalho para a tarefa de aprendizado de hierarquias de tópicos em coleções textuais dinâmicas. Foram utilizadas coleções textuais de *benchmark* de diversas características na simulação de cenários incrementais. Neste capítulo, é apresentado um estudo exploratório sobre o uso do IHTC em um problema real: a construção automática de diretórios web.

Os diretórios web desempenham um papel importante na recuperação de informação da Internet, principalmente em tarefas de busca exploratória. Neste tipo de tarefa, o usuário geralmente tem pouco domínio sobre o tema de interesse, dificultando expressar o objetivo diretamente por meio de palavras-chave (Marchionini, 2006). Assim, é necessário disponibilizar previamente algumas opções para guiar o processo de busca da informação. Para tal, cada diretório possui um conjunto de descritores que contextualizam e indicam algum significado dos documentos ali agrupados. Assim como nas hierarquias de tópicos, esta organização está relacionada com a hipótese de que se um usuário está interessado em um documento específico pertencente a um determinado diretório deve também estar interessado em outros documentos desse diretório e de seus subdiretórios (Manning et al., 2008).

A maioria dos diretórios web é construída de maneira supervisionada, a exemplo do *Dmoz - Open Directory Project*[1] (Figura 6.1) e *Yahoo! Directory*[2] (Figura 6.2), editadas manualmente por especialistas humanos. Apesar da boa qualidade dos resultados obtidos,

[1]Dmoz - Open Directory Project: http://www.dmoz.org/
[2]Yahoo Directory!: http://dir.yahoo.com/

a construção manual desses diretórios é limitada pela grande quantidade de documentos disponíveis e pela alta frequência de atualização (Yang e Lee, 2004; Kim, 2006). Este cenário é um exemplo de aplicação em que as estratégias exploradas ao longo deste projeto de mestrado podem contribuir.

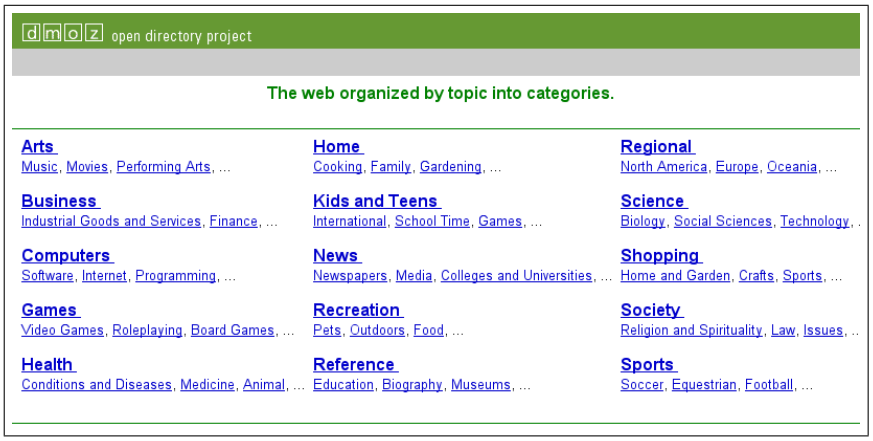

Figura 6.1: Diretórios Web do projeto DMOZ

Figura 6.2: Diretórios Web do *Yahoo! Directory*

Um dos diretórios web mais conhecidos e utilizados é o Dmoz, também chamado de "*Open Directory Project*". O Dmoz é um diretório público gerenciado por uma comunidade global de editores voluntários. Para exemplificar, em 2006 o Dmoz já contava com cerca de setenta mil editores humanos, e mais de cinco milhões de sites organizados em uma hierarquia com aproximadamente 590 mil tópicos e subtópicos. Atualmente, várias ferramentas de busca na internet, como o Google e AOL Search, incorporam o Dmoz em seus sistemas (Chung et al., 2008). A qualidade dos resultados do projeto depende diretamente dos voluntários, que são selecionados de acordo com sua especialidade para gerenciar cada tópico. Um dos problemas conhecidos do Dmoz é a grande demora para atualização do seu conteúdo[3], pois cada inclusão é avaliada subjetivamente. Em alguns

[3]Processo de submissão no Dmoz (http://www.dmoz.org/add.html)

casos, há falta de editores voluntários com conhecimento especializado para gerenciar determinados tópicos.

Em vista da relevância e da disponibilidade pública da sua base de dados[4], o Dmoz foi utilizado neste trabalho para um estudo exploratório sobre a construção automática de diretórios web. O objetivo é avaliar a abordagem proposta com o método IHTC e compará-la com uma estratégia similar baseada no algoritmo Leader, analisando-se a qualidade do agrupamento hierárquico e dos descritores dos grupos. As avaliações experimentais realizadas indicam que a construção de diretórios web é possível, no entanto, com diferenças na estrutura hierárquica. O diretório web construído com base no IHTC possui qualidade de agrupamento similar ao Leader. No entanto, o IHTC tem a capacidade de obter melhores descritores para os diretórios. Isto auxilia usuários em tarefas de busca exploratória, o que o torna especialmente útil na construção de diretórios web.

A seguir, é apresentado o desenvolvimento deste estudo exploratório, com a descrição da base de dados utilizada, realização dos experimentos e análise mais detalhada dos resultados.

6.1 Dados Utilizados

O projeto Dmoz disponibiliza sua base dados em um formato RDF/XML ("*Resource Description Framework*"), um padrão utilizado para troca de informação na web[5]. Para a avaliação experimental, foi selecionado um subconjunto dos dados com 5 diretórios, provenientes do projeto Dmoz: "*Business*", "*Computers*", "*Health*", "*Science*" e "*Sports*". Cada diretório possui subdiretórios e documentos relacionados. Ao final, obteve-se uma base de dados com 566.234 documentos organizados em 43.817 diretórios e subdiretórios. Na Tabela 6.1 é apresentada uma visão geral de algumas características da base de dados.

Tabela 6.1: Visão geral da base de dados selecionada a partir da base do Dmoz

Características da base de dados	
Documentos da Coleção	566234
Número de Termos	246280
Número de Termos com DF ≥ 2	84039
Número de (Sub)Diretórios	43817
Altura da Hierarquia	11
Média de Termos por Documento	12,41

O número total de termos (após remoção de *stopwords* e aplicação de *stemming*) é de 246.280. Deste total, há 84.039 termos que ocorrem em dois ou mais documentos (DF

[4]Licença de Uso do Dmoz: http://www.dmoz.org/license.html
[5]RDF ("*Resource Description Framework*"): http://www.w3.org/RDF/

≥ 2) e que podem ser úteis para a tarefa de agrupamento. Os diretórios e documentos estão distribuídos em uma hierarquia com 11 níveis conforme apresentado na Tabela 6.2.

Tabela 6.2: Frequência de diretórios e documentos por nível da hierarquia

Nível da Hierarquia	**Número de Diretórios**	**Número de Documentos**
0	1	0
1	5	3
2	231	2667
3	2463	54804
4	8304	148284
5	11510	153882
6	8778	102387
7	6597	58399
8	4612	36327
9	1167	8132
10	141	1285
11	8	64
Total	**43817**	**566234**

Os documentos associados aos diretórios da hierarquia representam uma página na web. Cada documento é composto por quatro campos: (1) o título da página, (2) a URL com o endereço de internet para a página, (3) uma breve descrição com 25 à 30 palavras sobre o conteúdo da página; e (4) o diretório da hierarquia na qual a página está alocada (removido para a realização dos experimentos). Na Tabela 6.3 é ilustrado um exemplo de um dos documentos existentes no conjunto de dados.

Tabela 6.3: Exemplo de um documento na base de dados do Dmoz

TITLE: *Robots.net*
URL: *http://robots.net*
DESC: *A news and discussion site for those interested in robots and robotics. Home of the Robot Competition FAQ and a variety of resource pages.*
TOPIC: *Top/Computers/Robotics*

Para a tarefa de construção automática de diretórios web foi removido o atributo "*TOPIC*" dos documentos, uma vez que um dos objetivos é aprender a organização de forma não supervisionada.

Para permitir a avaliação dos resultados por meio de critérios objetivos, foram selecionados 5 subconjuntos de dados contendo diretórios de referência. A seleção dos subconjuntos foi realizada de forma arbitrária, conforme a evolução dos experimentos, focando diretórios com temas bem definidos e, também, para facilitar uma análise exploratória

dos resultados. No total, foram selecionados 35 diretórios de referência divididos em 5 subconjuntos, conforme detalhado na Tabela 6.4.

Tabela 6.4: Dmoz - Subconjuntos selecionados para validação

Subconjunto de Validação 1 (dmz1)	
Diretório	**# Documentos**
Accounting	1447
Air Compressors	168
Air Dispersion Modeling	190
Alpacas	383
Arabian	712
Auctions	262
Batteries	298
Total:	**3460**

Subconjunto de Validação 2 (dmz2)	
Diretórios	**# Documentos**
Bowling	586
Cancer	2059
Coffee	205
Cosmetic and Plastic	729
Domain Names	853
Fire Fighting	1276
Franchising	643
Total:	**6351**

Subconjunto de Validação 3 (dmz3)	
Diretórios	**# Documentos**
Hospitals	1558
Hosting	1926
Industrial Yarns and Sewing	1504
Karate	786
Land Trusts	303
Leather and Fur	902
Molding	577
Total:	**7556**

Subconjunto de Validação 4 (dmz4)	
Diretórios	**# Documentos**
Money Managers	614
NCAA Division_II	179
Nursing	1051
Personal Chefs	230
Robotics	838
Search and Rescue	565
Sewing Machines	269
Total:	**3746**

Subconjunto de Validação 5 (dmz5)	
Diretórios	**# Documentos**
Signage	1820
Skiing	1903
Translation	2564
Venture Capital	841
Veterinarians	2587
Voice Talent	653
Wedding and Events	2717
Total:	**13085**

6.2 Configuração do Experimento

Para a realização dos experimentos foi instanciado um processo de mineração de textos para o aprendizado de hierarquias de tópicos. Assim, é possível dividir a configuração do experimento em três etapas: pré-processamento dos textos, extração de padrões e pós-processamento. Os experimentos foram realizados por meio da ferramenta *Torch - Topic Hierarchies*.

A seguir, são descritas as configurações realizadas em cada etapa. Os valores de parâmetros para os algoritmos e técnicas envolvidas no processo foram configurados de acordo com o conhecimento adquirido na avaliação experimental realizada no capítulo anterior.

6.2.1 Pré-processamento dos Textos

Os documentos são pré-processados individualmente, conforme são adicionados a uma fonte incremental de textos (diretório monitorado pela ferramenta). Para cada novo documento identificado, aplica-se a remoção de *stopwords* e a técnica de *stemming* para radicalização os termos.

Em seguida, os 20 termos mais frequentes são selecionados para representação do documento no modelo espaço-vetorial. Este valor é baseado em trabalhos anteriores da literatura (Koller e Sahami, 1997; Chang e Hsu, 2005), que indica que o uso de 10 à 25 palavras-chave geralmente são suficientes para representar o conteúdo de um documento, tanto em tarefas de agrupamento quanto de classificação.

Na Figura 6.3 é ilustrada a configuração da etapa de pré-processamento na ferramenta *Torch - Topic Hierarchies*.

Figura 6.3: Ferramenta Torch - configuração do pré-processamento dos textos

6.2.2 Extração de Padrões

Conforme comentando anteriormente, a base de dados coletada para o experimento contém 566.234 documentos, dificultando o emprego de algoritmos de agrupamento hierárquico tradicionais. Por exemplo, ao aplicar um algoritmo de agrupamento hierárquico

tradicional, baseado em matriz de distâncias, em uma base de dados com 500 mil objetos, seria necessário cerca de 230 Gigabytes[6] de memória RAM para a matriz de distâncias, o que é impraticável nos dias atuais. Este cenário é um exemplo da utilidade de se obter uma representação condensada dos dados, de maneira incremental, e então aplicar um algoritmo de agrupamento hierárquico sobre a representação condensada, diminuindo-se os requerimentos computacionais.

Assim, na etapa de extração de padrões, são explorados dois algoritmos de agrupamento incremental para obtenção da representação condensada: IHTC e Leader. Baseado na avaliação experimental realizada no capítulo anterior, o parâmetro r (número máximo de arestas na rede de coocorrência de termos) é configurado em 20.000 no IHTC. No algoritmo Leader, o parâmetro de distância mínima foi configurado em 0,8. Por fim, o nível de condensação foi configurado em 3000 grupos, de acordo com os recursos computacionais disponíveis para execução dos experimentos[7]. A hierarquia de tópicos final é obtida por meio do algoritmo de agrupamento hierárquico Bisecting-kmeans.

Na Figura 6.4 são ilustradas as configurações da etapa de extração de padrões na ferramenta *Torch - Topic Hierarchies*.

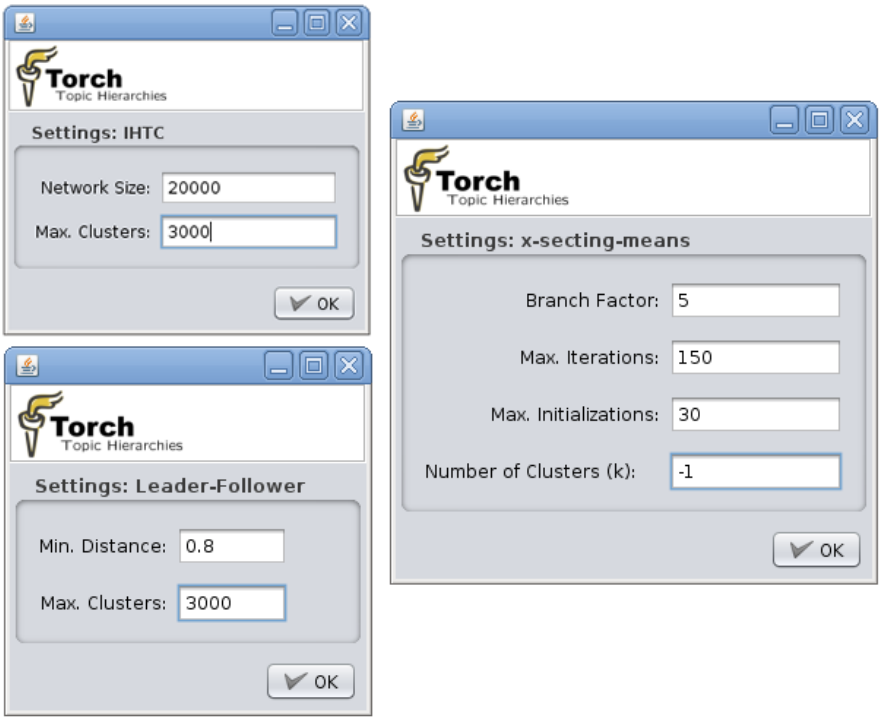

Figura 6.4: Ferramenta Torch - configuração dos algoritmos utilizados na etapa de extração de padrões

[6] Considerando apenas a matriz triangular com $(n * (n-1))/2$ posições de 16 bits para alocar valores das distâncias

[7] Computador IBM/PC, processador Intel 2.4 Ghz, 2GB RAM e Sistema Operacional GNU/Linux (Debian)

6.2.3 Pós-processamento

Para avaliação dos resultados, na etapa de pós-processamento, é aplicada a medida FScore (Seção 2.4.3 do Capítulo 2). Neste experimento, a FScore é utilizada para análise de dois critérios: eficácia da recuperação do agrupamento hierárquico e eficácia da recuperação dos descritores selecionados para o agrupamento. Os valores de FScore são calculados a partir dos subconjuntos de validação apresentados na Seção 6.1.

Para calcular a eficácia de recuperação dos descritores com a FScore, o conjunto de documentos associado a um determinado grupo é substituído pelo conjunto de documentos recuperados utilizando-se os descritores do grupo em uma expressão de busca.

6.3 Experimentos e Análise dos Resultados

Uma vez definida a configuração dos parâmetros para o experimento visando a construção automática de diretórios web, foi executado o processo de aprendizado de hierarquias de tópicos a partir do conjunto de dados proveniente do Dmoz. A seguir, é realizada uma análise dos resultados de acordo com dois critérios calculados com o índice FScore: (1) a eficácia de recuperação do agrupamento hierárquico; e (2) a eficácia de recuperação dos descritores selecionados para o agrupamento.

Na Tabela 6.5 é apresentado o resultado geral da avaliação experimental nos subconjuntos de validação. Para determinar o grau de confiança das comparações realizadas, foi aplicado o teste estatístico não paramétrico de Wilcoxon conforme descrito em Demšar (2006).

Tabela 6.5: Resultado geral da avaliação experimental

	Agrupamento		**Descritores**	
	IHTC	**LEADER**	**IHTC**	**LEADER**
dmz1	0,557	**0,579**	**0,567**	0,452
dmz2	0,487	**0,559**	**0,543**	0,466
dmz3	**0,496**	0,468	**0,491**	0,339
dmz4	**0,443**	0,407	**0,438**	0,317
dmz5	0,654	**0,660**	**0,665**	0,567

Em relação ao critério de **eficácia da recuperação do agrupamento hierárquico**, o IHTC e o Leader obtiveram resultados similares. Na Figura 6.5 é ilustrada uma comparação entre a medida FScore nos 5 subconjuntos de validação. Neste cenário, o Leader apresentou resultados um pouco superiores quando comparado com o IHTC, entretanto, não foi possível indicar diferença estatisticamente significativa na comparação deste critério.

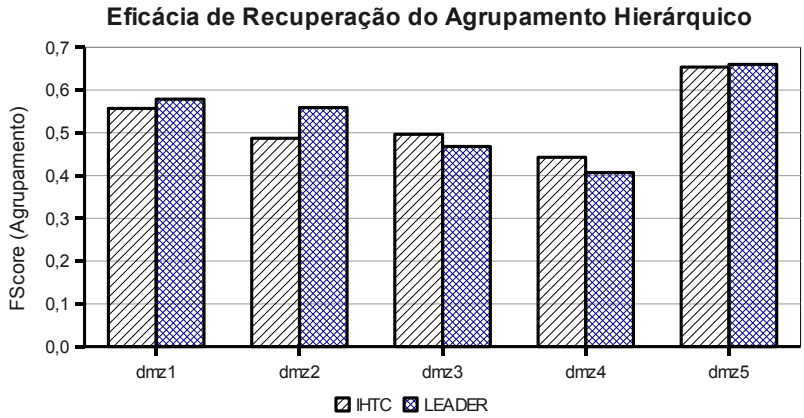

Figura 6.5: Dmoz - Eficácia de recuperação do agrupamento hierárquico baseados no IHTC e Leader

Por outro lado, o IHTC obteve resultados superiores em relação ao critério de **eficácia da recuperação dos descritores**, com diferença estatisticamente significativa. Conforme descrito anteriormente, o IHTC permite fornecer grupos de termos como descritores para o agrupamento. Ao utilizar esses grupos de termos, os descritores selecionados para a hierarquia obtiveram maior eficácia de recuperação do que a estratégia baseada em centroide adotada no algoritmo Leader. Na Figura 6.6 é ilustrada a comparação da eficácia da recuperação dos descritores.

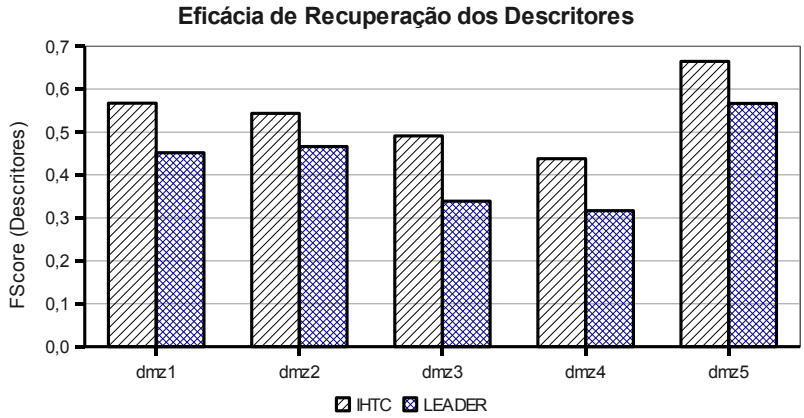

Figura 6.6: Dmoz - Eficácia de recuperação dos descritores selecionados para o agrupamento hierárquico

Este resultado está em conformidade com a proposta do IHTC, que visa auxiliar a tarefa de interpretação do agrupamento. Para isto, o IHTC utiliza as relações existentes na rede de coocorrência de termos para identificação de possíveis tópicos na coleção textual. Na Tabela 6.6 são apresentados alguns dos diretórios de referência extraídos do Dmoz (construída por humanos). Para cada tópico de referência, também são apresentados os

respectivos descritores obtidos pelos experimentos realizados com o IHTC e o Leader. A lista completa com os descritores selecionados, em comparação com os diretórios de referência do Dmoz, está disponível no Apêndice D.

Tabela 6.6: Exemplo de diretórios obtidos automaticamente pelo IHTC e Leader em comparação com o diretório original da Dmoz

Origem	Diretório (Tópico)	Eficácia da Recuperação	
		Agrupamento	Descritores
DMOZ	Cancer	-	-
IHTC	{Cancers,Treatments}, {Cancers,Researchers}, {Cancers,Information}	**0,335**	0,499
Leader	Cancers, Breasts, Nci	0,332	**0,582**
DMOZ	Veterinarians	-	-
IHTC	{Animals,Services}, {Animals,Hospital}, {Hospital,Services}	0,706	**0,691**
Leader	Veterinary, Hours, Hospital	**0,782**	0,430
DMOZ	Translation	-	-
IHTC	{English,Translations}, {Germans,Translations}, {Frenchs,Translations}	**0,706**	**0,684**
Leader	Translations, Interpreting, English	0,691	0,481
DMOZ	Robotics	-	-
IHTC	{Researchers,Robotics}, {Mobile,Robotics}, {Controls,Robotics}	0,418	0,379
Leader	Robotics, Estimators, Autonomous	**0,524**	**0,486**
DMOZ	Money_Managers	-	-
IHTC	{Financial,Investment}, {Investment,Planning}, {Advisory,Investment}	**0,406**	**0,228**
Leader	Investment, Advisors, Advisory	0,260	0,041
DMOZ	Search_and_Rescue	-	-
IHTC	{Rescue,Searches}, {Searches,Volunteer}, {Searches,Teams}	0,428	**0,737**
Leader	Rescue, Trucks, Volunteer	**0,450**	0,184
DMOZ	Accounting	-	-
IHTC	{Services,Taxes}, {Accounting,Services}, {Accounting,Taxes}	0,593	0,611
Leader	Taxes, Accounting, Cpas	**0,655**	**0,653**

É importante observar que o IHTC possui a vantagem de utilizar pares de termos (arestas da rede de coocorrências) como descritores dos grupos. Em geral, uma expressão ou conceito identificado por conjuntos de termos relacionados tem maior poder discriminativo do que termos simples (Zhang et al., 2010). Mesmo em situações em que termos simples apresentam melhor eficácia de recuperação, o uso de conjunto de termos pode auxiliar mais os usuários na interpretação dos grupos, por exemplo, no caso de "*Cancers, Treatments*" e "*Cancers, Researchers*" em vez do termo simples "*Cancers*".

Em comparação com a hierarquia de diretórios original do Dmoz, nota-se que a construção automática de diretórios web leva a uma estrutura hierárquica diferente. Os primeiros níveis dos diretórios web (mais próximos da raiz), tanto no IHTC quanto no Leader, muitas vezes reúnem documentos de tópicos distantes, o que dificulta a interpretação. Uma razão para este problema é que não foi utilizada uma estratégia para determinar quais partições exibir para os usuários durante a navegação na hierarquia. Assim, os diretórios de níveis mais altos geralmente não tem uma divisão natural dos tópicos. No entanto, esta análise subjetiva deve ser realizada com a presença de usuários e especialistas de domínio (Chung et al., 2008), que não estava no escopo deste projeto de mestrado.

Na Figura 6.7 é ilustrada uma comparação de diretórios relacionados com o tópico

"Negócios e Serviços", baseado no IHTC e Leader. No caso do IHTC, os descritores são extraídos a partir da rede de coocorrência de termos e, assim, são mais detalhados e fornecem uma ideia mais precisa do conteúdo dos documentos ali agrupados.

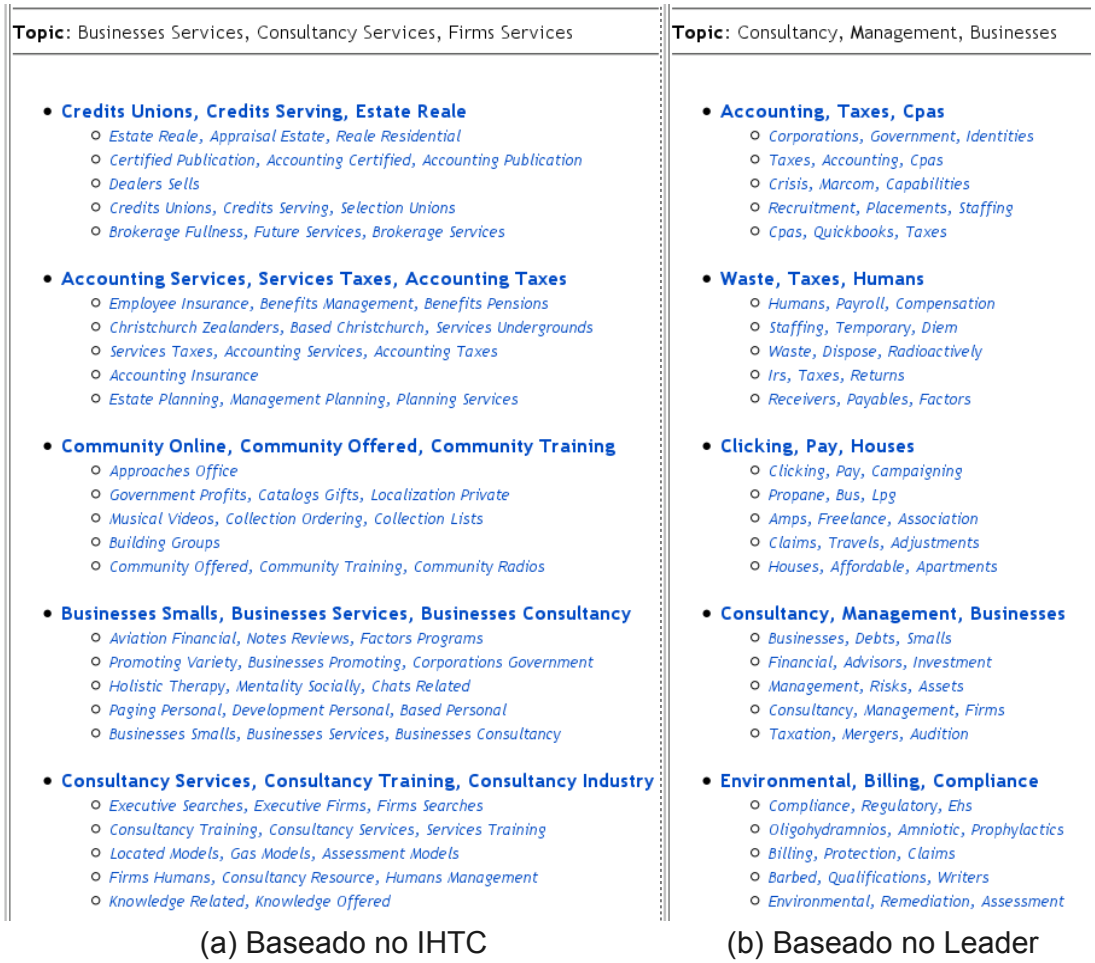

Figura 6.7: Dmoz - Comparação de diretórios com o tópico "Negócios e Serviços" baseado (a) IHTC e (b) Leader

Os diretórios web completos obtidos com a realização deste trabalho estão disponíveis *online* no endereço http://sites.labic.icmc.usp.br/marcacini/ihtc, incluindo a base de dados utilizada, a ferramenta *Torch - Topic Hierarchies*, e um módulo computacional para apoiar a construção e publicação de diretórios web.

6.4 Considerações Finais

Neste capítulo, foi descrito um estudo exploratório para construção automática de diretórios web, com base nos dados proveniente do projeto Dmoz. O estudo exploratório permitiu analisar dois desafios atuais da área de agrupamento, em um conjunto de dados

relativamente grande: o agrupamento incremental e a interpretação dos resultados do agrupamento.

Um processo de mineração de textos para aprendizado de hierarquias de tópicos foi instanciado com base nos conceitos estudados ao longo deste projeto de mestrado. Foram aplicados dois algoritmos de agrupamento incremental para obtenção da representação condensada dos textos: o IHTC e o Leader.

As avaliações experimentais realizadas com a base do Dmoz indicam que o diretório web construído a partir do IHTC possui qualidade de agrupamento similar ao Leader, no conjunto de dados avaliado. No entanto, o IHTC tem a capacidade de obter melhores descritores para os diretórios. Isto auxilia usuários em tarefas de busca exploratória, o que o torna especialmente útil na construção de diretórios web.

O estudo exploratório aqui apresentado contém os primeiros resultados obtidos a partir de uma aplicação prática. Para aprofundar neste tema, este tipo de análise deve considerar também a participação de usuários, além de ser desejável a utilização de um maior número de conjuntos de validação e emprego de outros critérios para análise do agrupamento. No entanto, os resultados alcançados até o momento podem ser considerados promissores, baseado na ideia de que o processo para construção de diretórios web aqui descrito é não supervisionado, em contraste com o método de gerenciamento atual do Dmoz que é totalmente dependente de especialistas humanos.

Os resultados apresentados e discutidos neste capítulo foram submetidos no Encontro Nacional de Inteligência Artificial (Marcacini e Rezende, 2011).

Capítulo

7

Conclusões

Em um contexto no qual grande parte das informações armazenadas pelas organizações está na forma textual, o desenvolvimento de técnicas computacionais para a organização destas bases e a exploração do conhecimento nelas contido é uma tarefa importante. Dentre as abordagens existentes, destacam-se iniciativas que organizam o conhecimento em uma hierarquia de tópicos, dado que esta é a forma mais intuitiva de se estruturar o conhecimento para os usuários, uma vez que os grupos e subgrupos obtidos fornecem uma descrição sucinta e representativa do conhecimento em diferentes níveis de granularidade. Ainda, a maioria das aplicações lidam com bases de textos dinâmicas, que com o passar do tempo adicionam, descartam ou alteram seus documentos, surgindo então um novo desafio para os métodos de organização e extração de conhecimento.

Neste projeto de mestrado, contribuiu-se com o aprendizado não supervisionado de hierarquias de tópicos a partir de coleções textuais dinâmicas. No Capítulo 2 foi apresentado um processo de mineração de textos para apoiar o aprendizado de hierarquias de tópicos, no qual se aplicam métodos de agrupamento hierárquico de documentos na etapa de extração de padrões, seguido da seleção de descritores para os grupos formados. Foi visto que os métodos de agrupamento hierárquico tradicionais não são apropriados para cenários dinâmicos, que exigem frequente atualização do agrupamento. Assim, no Capítulo 3 discutiu-se estratégias de agrupamento incremental que possibilitam adicionar novos documentos aos grupos, sem reprocessamento redundante. O agrupamento incremental foi aplicado para obter e manter atualizada uma representação condensada dos textos, que contém as principais características dos dados, e esta representação condensada é utilizada para a construção do agrupamento hierárquico com significativa redução do custo computacional. Neste sentido, durante o desenvolvimento do trabalho foi proposto um método alternativo para aprendizado de hierarquias de tópicos, denominado

IHTC, que explora o agrupamento incremental de termos, conforme descrito no Capítulo 4. O diferencial do IHTC é a construção, de maneira incremental, de grupos de termos associados aos grupos de documentos. Os grupos de termos são utilizados como descritores o que torna o IHTC especialmente útil para a interpretação e exploração das hierarquias de tópicos. No Capítulo 5 foi apresentada uma avaliação experimental para comparar, sob diferentes critérios, as estratégias de agrupamento incremental e a qualidade das hierarquias de tópicos obtidas. Por fim, visando também analisar a contribuição prática deste projeto de mestrado, foi realizado um estudo exploratório sobre a construção automática de diretórios web baseada em dados reais do projeto Dmoz utilizando o IHTC, apresentado no Capítulo 6.

Os resultados obtidos com as avaliações experimentais permitiram reforçar a hipótese deste projeto de mestrado, de que é possível realizar o aprendizado não supervisionado de hierarquias de tópicos a partir de coleções textuais dinâmicas, por meio de estratégias de agrupamento incremental, com qualidade próxima ao obtido por estratégias de agrupamento não incremental. Embora a eficácia de recuperação do agrupamento hierárquico obtido por uma estratégia não incremental tenha, nas coleções textuais analisadas, superado a eficácia dos agrupamentos hierárquicos obtidos com base em estratégias incrementais, a diferença não é estatisticamente significante.

7.1 Contribuições

Uma das contribuições deste projeto de mestrado se refere às comparações objetivas das estratégias de agrupamento incremental. Alguns trabalhos na literatura apresentam comparações similares entre os algoritmos de agrupamento não incrementais, mas pouca atenção tem sido dada para agrupamento incremental. Vários algoritmos de agrupamento incremental de documentos têm sido propostos, muitas vezes com pequenas variações entre eles, e assim foi feito um trabalho de revisão e organização desses algoritmos de acordo com suas estratégias. Com as comparações realizadas, foi possível identificar algumas situações em que cada estratégia é mais adequada.

Outra contribuição decorrente deste projeto é a proposta e avaliação do método IHTC, potencialmente útil para o aprendizado não supervisionado de hierarquias de tópicos. O IHTC utiliza uma estratégia de agrupamento incremental de termos, na qual o foco principal é encontrar relações significativas entre os termos da coleção textual e, assim, utilizar estas relações para apoiar a formação dos grupos de documentos. O método é apropriado quando o usuário está interessado em análise exploratória dos textos, pois apresenta automaticamente uma descrição detalhada dos agrupamentos obtidos.

Durante o desenvolvimento do mestrado, trabalhou-se com a ideia de que ferramentas

computacionais associadas aos métodos propostos aumentam consideravelmente a visibilidade e importância da pesquisa realizada, principalmente na área de computação. Dessa forma, um resultado direto deste projeto é ferramenta computacional *Torch - Topic Hierarchies*, uma aplicação que instancia as principais etapas de um processo de mineração de textos para aprendizado de hierarquias de tópicos. A ferramenta foi desenvolvida na linguagem Java, com auxílio de alguns componentes de software livre, disponibilizando em um só ambiente os seguintes recursos:

- Etapa de pré-processamento dos textos: estruturação dos textos, seleção de termos com base em cortes Luhn, remoção de stopwords, radicalização de termos com *stemming* disponível em língua portuguesa e inglesa.
- Etapa de extração de padrões: algoritmos de agrupamento incremental (Leader, Buckshot, DCTree, IHTC e Cobweb), agrupamento não incremental (k-means, bisecting k-means) e módulos de seleção de descritores para agrupamento.
- Etapa de pós-processamento: medidas de validação de agrupamento (Silhueta, Entropia e FScore), módulo para exploração visual de hierarquias de tópicos (grafos e árvores) e módulo para publicação de hierarquias de tópicos na web.

A ferramenta Torch está disponível online[1] publicamente para a comunidade e usuários interessados. Nas Figuras 7.1 e 7.2 são apresentadas algumas telas da ferramenta Torch.

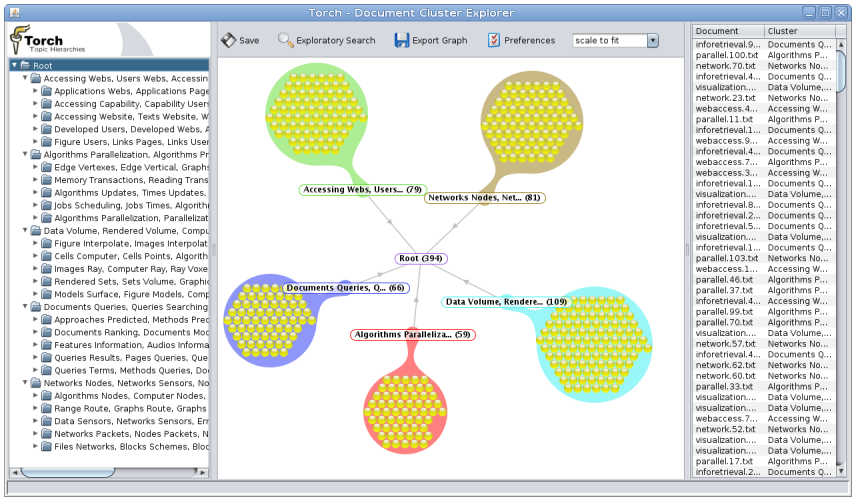

Figura 7.1: Tela da ferramenta Torch ilustrando a análise visual da hierarquia de tópicos e respectivo agrupamento.

Na Figura 7.1, percebe-se à esquerda a hierarquia de tópicos (e subtópicos), com seus respectivos descritores, e uma visualização dos grupos de documentos ao centro. Do lado

[1]Torch - Topic Hierarchies: http://sites.labic.icmc.usp.br/marcacini/ihtc

direito da imagem, são apresentados os documentos relacionados ao tópico selecionado com seu respectivo grupo. Na Figura 7.2, é visualizado a rede de coocorrências de uma hierarquia de tópicos construída pelo método IHTC. A rede de coocorrência apresentada é formada a partir dos termos selecionados pelo usuário.

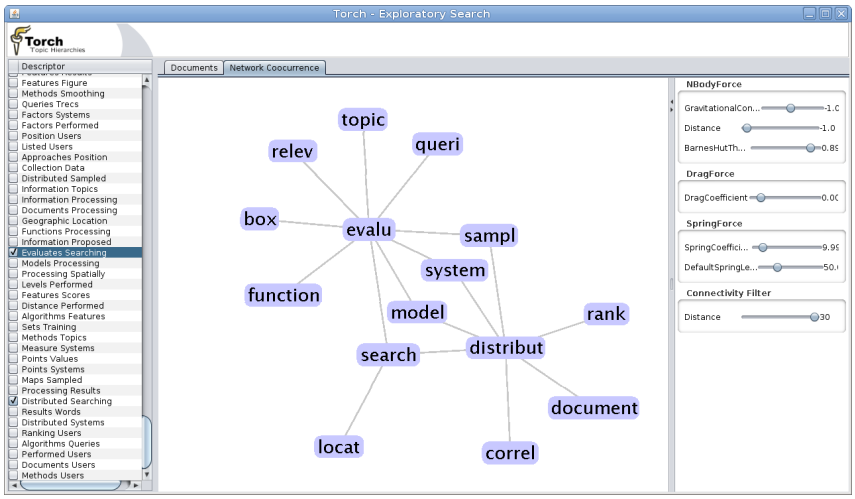

Figura 7.2: Tela da ferramenta Torch ilustrando a análise visual de uma rede de coocorrência de termos extraída com o método IHTC.

Até o presente momento, algumas dessas contribuições foram divulgadas por meio de três artigos em conferências relacionadas com Mineração de Textos, disponíveis em Marcacini e Rezende (2010a), Marcacini e Rezende (2010b) e Marcacini e Rezende (2011). Em Marcacini e Rezende (2010a) foi apresentado o método para agrupamento incremental que viabiliza a seleção de descritores juntamente com a formação de grupos de documentos. O método, denominado *IHTC - Incremental Hierarchical Term Clustering*, se mostrou competitivo comparado aos existentes na literatura, além de permitir o aprendizado de hierarquias de tópicos com maior qualidade na eficácia de recuperação dos descritores. Já em Marcacini e Rezende (2010b) foi apresentada a ferramenta Torch para aprendizado de hierarquias de tópicos a partir de coleções textuais dinâmicas. A ferramenta Torch foi agraciada com prêmio "Melhores Ferramentas" durante o IX Workshop de Ferramentas e Aplicações de 2010, realizado em conjunto com o XVI Simpósio Brasileiro de Sistemas Multimídia e Web. Por fim, em Marcacini e Rezende (2011) foi discutido o uso de agrupamento incremental de termos para a construção automática de diretório web, visando explorar os ganhos práticos dos métodos investigados durante o mestrado.

7.2 Limitações

Embora o desenvolvimento deste projeto de mestrado tenha cumprido os objetivos propostos, algumas limitações devem ser observadas.

As coleções textuais dinâmicas são caracterizadas por frequentes operações de inserção de novos documentos, assim como remoções e alterações dos documentos existentes. No entanto, neste trabalho considerou-se apenas a operação de inserção para facilitar a realização e análise da avaliação experimental. Dessa forma, os algoritmos de agrupamento incremental foram implementados visando apenas adicionar novo conhecimento no agrupamento existente. Seria interessante permitir que o agrupamento fosse ajustado também após a remoção ou alteração de documentos que não são mais desejados na coleção textual.

Outra limitação se deve ao processo de avaliação. As hierarquias de tópicos são úteis para análise ou busca exploratória de uma coleção textual. Esta análise exploratória é geralmente uma tarefa realizada por um usuário humano que deseja extrair algum conhecimento novo e útil dos textos. No entanto, a avaliação realizada focou em comparações por medidas objetivas e testes de significância estatística. Neste tipo de tarefa, seria desejável uma avaliação qualitativa, com a presença de usuários ou especialistas de domínio.

7.3 Trabalhos Futuros

As primeiras propostas para trabalhos futuros estão relacionadas com as limitações encontradas no desenvolvimento deste projeto. Para os algoritmos de agrupamento incremental, pretende-se implementar as operações de remoção e atualização de documentos previamente agrupados e explorar métodos eficazes para ajustar o agrupamento após essas operações.

Embora o processo de aprendizado aqui estudado seja não supervisionado, é importante que usuários possam interferir ou validar alguma das etapas do processo até a obtenção final das hierarquias de tópicos. Assim, pretende-se também investigar métodos que permitam aos usuários adicionarem algum conhecimento de domínio, principalmente durante o agrupamento de documentos e seleção de descritores para os grupos.

Por fim, uma direção para desenvolvimentos futuros é explorar melhor os recursos existentes no método proposto IHTC. A rede de coocorrência de termos mostrou-se uma estrutura interessante para análise visual de conceitos existentes em coleção de textos. No entanto, atualmente utiliza-se apenas a frequência de coocorrência entre dois termos para construir as relações da rede. É possível aplicar outras medidas objetivas de validação, como as utilizadas em regras de associação, para analisar o grau de coocorrência entre os termos e, assim, identificar as relações mais significativas.

Referências Bibliográficas

Agrawal, R., Gehrke, J., Gunopulos, D., e Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. *ACM SIGMOD Record*, 27(2):94–105. Citado na página 20.

Alcalá, J., Fernández, A., Luengo, J., Derrac, J., GarcÍa, S., Sánchez, L., e Herrera, F. (2011). KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *Journal of Multivalued Logic and Soft Computing*, 17(2):255–287. Citado na página 66.

Allan, J. (2002). Introduction to topic detection and tracking. páginas 1–16. Citado nas páginas 32 e 35.

Bedford, D. A. D. (2008). *Knowledge management in practice: connections and context*. American Society for Information Science and Technology. Citado na página 27.

Berkhin, P. (2006). A survey of clustering data mining techniques. In Kogan, J., Nicholas, C., e Teboulle, M., editors, *Grouping Multidimensional Data*, chapter 2, páginas 25–71. Springer-Verlag, Berlin, Heidelberg. Citado na página 14.

Boley, D., Gini, M., Gross, R., Han, E. H., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., e Moore, J. (1999). Document categorization and query generation on the world wide web using webace. *Artificial Intelligence Review*, 13(5):365–391. Citado nas páginas 58 e 59.

Bradley, P. S., Fayyad, U. M., e Reina, C. (1998). Scaling clustering algorithms to large databases. In *Knowledge Discovery and Data Mining*, páginas 9–15. Citado na página 30.

Can, F. (1993). Incremental clustering for dynamic information processing. *ACM Transactions on Information Systems*, 11:143–164. Citado na página 29.

Carpineto, C., Osiński, S., Romano, G., e Weiss, D. (2009). A survey of web clustering engines. *ACM Computing Surveys*, 41:1–17. Citado nas páginas 8 e 27.

Chakrabarti, S. (2002). *Mining the web: discovering knowledge from hypertext data*. Science & Technology Books. Citado na página 9.

Chang, H. C. e Hsu, C. C. (2005). Using topic keyword clusters for automatic document clustering. In *ICITA'2005: Third International Conference on Information Technology and Applications*, páginas 419–424. Citado nas páginas 60 e 80.

Charikar, M., Chekuri, C., Feder, T., e Motwani, R. (1997). Incremental clustering and dynamic information retrieval. In *STOC'97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, páginas 626–635, New York, NY, USA. ACM. Citado na página 29.

Chim, H. e Deng, X. (2007). A new suffix tree similarity measure for document clustering. In *WWW'07: Proceedings of the 16th international conference on World Wide Web*, páginas 121–130, New York, NY, USA. ACM. Citado na página 31.

Chu, H. (2003). *Information Representation and Retrieval in the Digital Age*. Information Today. Citado nas páginas 21 e 22.

Chuang, S.-L. e Chien, L.-F. (2004). A practical web-based approach to generating topic hierarchy for text segments. In *CIKM '04: Proceedings of the thirteenth ACM International Conference on Information and Knowledge Management*, páginas 127–136, New York, NY, USA. ACM. Citado na página 21.

Chung, W., Lai, G., Bonillas, A., Xi, W., e Chen, H. (2008). Organizing domain-specific information on the web: An experiment on the spanish business web directory. *International Journal of Human-Computer Studies*, 66(2):51–66. Citado nas páginas 72, 76, e 84.

Conrado, M. S., Marcacini, R. M., Moura, M. F., e Rezende, S. O. (2009). O efeito do uso de diferentes formas de geração de termos na compreensibilidade e representatividade dos termos em coleções textuais na língua portuguesa. In *WTI'09: II International Workshop on Web and Text Intelligence*, páginas 1–10. Citado na página 10.

Crouch, D. B. (1975). A file organization and maintenance procedure for dynamic document collections. *Information Processing & Management*, 11(1):11–21. Citado na página 30.

Cutting, D. R., Pedersen, J. O., Karger, D., e Tukey, J. W. (1992). Scatter/gather: A cluster-based approach to browsing large document collections. In *SIGIR'92: Proceedings of the 15th Annual International Conference on Research and Development in Information Retrieval*, páginas 318–329. Citado nas páginas 7, 21, 31, e 36.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30. Citado nas páginas 66 e 82.

Ebecken, N. F. F., Lopes, M. C. S., e de Aragão Costa, M. C. (2003). Mineração de textos. In Rezende, S. O., editor, *Sistemas Inteligentes: Fundamentos e Aplicações*, chapter 13, páginas 337–370. Manole, 1ª edição. Citado nas páginas 7 e 41.

Ester, M., Kriegel, H.-P., Sander, J., Wimmer, M., e Xu, X. (1998). Incremental clustering for mining in a data warehousing environment. In *VLDB'98: Proceedings of the 24rd International Conference on Very Large Data Bases*, páginas 323–333, San Francisco, CA, USA. Morgan Kaufmann Publishers. Citado nas páginas 32 e 33.

Ester, M., Kriegel, H.-P., Sander, J., e Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD'96: Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, páginas 226 – 231. Citado na página 20.

Everitt, B. S., Landau, S., e Leese, M. (2001). *Cluster Analysis*. Arnold Publishers. Citado nas páginas 2, 12, 14, 15, 18, e 23.

Faceli, K., Carvalho, A. C. P. L. F., e Souto, M. C. P. (2005). Validação de algoritmos de agrupamento. Relatório Técnico 254, Instituto de Ciências Matemáticas e de Computação - ICMC - USP. Citado na página 23.

Farnstrom, F., Lewis, J., e Elkan, C. (2000). Scalability for clustering algorithms revisited. *ACM SIGKDD Explorations Newsletter*, 2:51–57. Citado na página 30.

Feldman, R. e Sanger, J. (2006). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press. Citado nas páginas 1, 2, 9, 11, 13, 14, 15, 17, e 45.

Finch, T. (2009). Incremental calculation of weighted mean and variance. University of Cambridge Computing Service. http://www-uxsup.csx.cam.ac.uk/~fanf2/hermes/doc/antiforgery/stats.pdf. Citado na página 31.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172. Citado na página 32.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305. Citado na página 22.

Fung, B. C. M., Wang, K., e Ester, M. (2003). Hierarchical document clustering using frequent itemsets. In *SDM'03: Proceedings of SIAM International Conference on Data Mining*, volume 30. Citado na página 41.

Fung, B. C. M., Wang, K., e Ester, M. (2009). *The Encyclopedia of Data Warehousing and Mining*, chapter Hierarchical Document Clustering, páginas 970–975. Idea Group, Hershey, PA. Citado nas páginas 1 e 42.

Gantz, J. F., Chute, C., Manfrediz, A., Minton, S., Reinsel, D., Schlichting, W., e Toncheva, A. (2008). IDC - The Diverse and Exploding Digital Universe. *External Publication of IDC (Analyse the Future) Information and Data*, páginas 1–10. Citado na página 1.

Gantz, J. F. e Reinsel, D. (2009). As the economy contracts, the digital universe expands. *External Publication of IDC (Analyse the Future) Information and Data*, páginas 1–10. Citado na página 1.

Gantz, J. F. e Reinsel, D. (2010). The digital universe decade - are you ready? *External Publication of IDC (Analyse the Future) Information and Data*, páginas 1–16. Citado na página 1.

Garcia, R. G. e Porrata, A. P. (2010). Dynamic hierarchical algorithms for document clustering. *Pattern Recognition Letters*, 31(6):469 – 477. Citado nas páginas 3 e 33.

Gennari, J. H., Langley, P., e Fisher, D. (1989). Models of incremental concept formation. *Artificial Intelligence*, 40:11–61. Citado na página 32.

Giraud, C. (2000). A note on the utility of incremental learning. *AI Communications*, 13:215–223. Citado na página 30.

Grossman, D. A. e Frieder, O. (2004). *Information Retrieval: Algorithms and Heuristics*. The Kluwer International Series of Information Retrieval. Springer. Citado nas páginas 31 e 36.

Halkidi, M., Batistakis, Y., e Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145. Citado nas páginas 23 e 24.

Hammouda, K. M. e Kamel, M. S. (2003). Incremental document clustering using cluster similarity histograms. In *WI '03: Proceedings of the 2003 IEEE International Conference on Web Intelligence*, página 597, Washington, DC, USA. IEEE Computer Society. Citado na página 32.

Han, J. e Kamber, M. (2006). *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2nd^{a} edição. Citado nas páginas 1 e 12.

Hartigan, J. (1975). *Clustering algorithms*. Wiley New York. Citado na página 30.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666. Citado na página 29.

Jain, A. K. e Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Upper Saddle River, NJ, USA. Citado na página 24.

Jain, A. K., Murty, M. N., e Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323. Citado nas páginas 2, 12, 14, e 30.

Karypis, G. (2003). The clustering toolkit (cluto) data sets. Página da internet acessada em 19/02/2010. http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download. Citado na página 58.

Kaufman, L. e Rousseeuw, P. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Interscience, New York. Citado nas páginas 19 e 24.

Kim, H. J. (2006). On text mining algorithms for automated maintenance of hierarchical knowledge directory. In *Knowledge Science, Engineering and Management*, Lecture Notes in Computer Science, páginas 202–214. Citado na página 76.

Kishida, K. (2010). High-speed rough clustering for very large document collections. *Journal of the American Society for Information Science and Technology*, 61:1092–1104. Citado nas páginas 30, 38, e 39.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69. Citado na página 20.

Kohonen, T., Hynninen, J., Kangas, J., e Laaksonen, J. (1996). SOM PAK: The self-organizing map program package. *Relatório Técnico A31, Helsinki University of Technology, Laboratory of Computer and Information Science*. Citado na página 21.

Koller, D. e Sahami, M. (1997). Hierarchically classifying documents using very few words. In *ICML'97: Proceedings of the Fourteenth International Conference on Machine Learning*, páginas 170–178, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. Citado nas páginas 60 e 80.

Kowalski, G. e Maybury, M. T. (2002). *Information Storage and Retrieval Systems*, volume 8, chapter Document and Term Clustering, páginas 139–163. Springer Netherlands. Citado na página 41.

Kriegel, H. P., Borgwardt, K., Kroger, P., Pryakhin, A., Schubert, M., e Zimek, A. (2007). Future trends in data mining. *Data Mining and Knowledge Discovery*, 15:87–97. Citado na página 29.

Krowne, A. e Halbert, M. (2005). An initial evaluation of automated organization for digital library browsing. In *JCDL'05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, páginas 246–255, New York, NY, USA. ACM. Citado na página 27.

Kuechler, W. L. (2007). Business applications of unstructured text. *Communications of ACM*, 50(10):86–93. Citado na página 1.

Larose, D. T. (2004). *Discovering Knowledge in Data: An Introduction to Data Mining*. Wiley-Interscience, 1ª edição. Citado nas páginas 18 e 19.

Larsen, B. e Aone, C. (1999). Fast and effective text mining using linear-time document clustering. In *SIGKDD'99: Proceedings of the 5th ACM International Conference on Knowledge Discovery and Data Mining*, páginas 16–22. Citado na página 25.

Lewis, D. D. (1997). Reuters-21578 text categorization test collection. Página da internet acessada em 26/09/2010. http://www.daviddlewis.com/resources/testcollections/. Citado na página 58.

Li, Y., Chung, S., e Holt, J. D. (2008). Text document clustering based on frequent word meaning sequences. *Data and Knowledge Engineering*, 64(1):381–404. Citado na página 41.

Liu, B., Shi, Y., Wang, Z., Wang, W., e Shi, B. (2006). Dynamic incremental data summarization for hierarchical clustering. In Yu, J., Kitsuregawa, M., e Leong, H., editors, *Advances in Web-Age Information Management*, volume 4016 of *Lecture Notes in Computer Science*, páginas 410–421. Springer Berlin / Heidelberg. Citado nas páginas 29, 30, 38, e 39.

Liu, C. L. (1968). *Introduction to combinatorial mathematics*. McGraw-Hill, New York, USA. Citado na página 15.

Liu, L., Kang, J., Yu, J., e Wang, Z. (2005). A comparative study on unsupervised feature selection methods for text clustering. In *NLP-KE '05. Proceedings of 2005 International Conference on Natural Language Processing and Knowledge Engineering*, páginas 597–601. Citado na página 11.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal os Research and Development*, 2(2):159–165. Citado na página 10.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. e Neyman, J., editors, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, páginas 281–297. University of California Press. Citado na página 15.

Maimon, O. e Rokach, L. (2005). *Data Mining and Knowledge Discovery Handbook*. Springer-Verlag, Secaucus, NJ, USA. Citado na página 30.

Manning, C. D., Raghavan, P., e Schütze, H. (2008). *An Introduction to Information Retrieval*. Cambridge University Press. Citado nas páginas 1, 2, 9, 10, 21, e 75.

Marcacini, R. M., Moura, M. F., e Rezende, S. O. (2007). Biblioteca Digital do IFM: uma Aplicação para a Organização da Informação por meio de Agrupamentos Hierárquicos. In *WDL'07: III Workshop on Digital Libraries do Simpósio Brasileiro de Sistemas Multimídia e Web (Webmedia)*, páginas 1–10, Gramado - RS. SBC. Citado na página 27.

Marcacini, R. M. e Rezende, S. O. (2010a). Incremental Construction of Topic Hierarchies using Hierarchical Term Clustering. In *SEKE'2010: Proceedings of the 22nd International Conference on Software Engineering and Knowledge Engineering*, páginas 553–558, Redwood City, San Francisco, USA. KSI - Knowledge Systems Institute. Citado nas páginas 5, 54, e 90.

Marcacini, R. M. e Rezende, S. O. (2010b). Torch: a tool for building topic hierarchies from growing text collection. In *WFA'2010: IX Workshop de Ferramentas e Aplicações. Em conjunto com o XVI Simpósio Brasileiro de Sistemas Multimídia e Web (Webmedia)*, páginas 1–3. Citado nas páginas 5, 65, e 90.

Marcacini, R. M. e Rezende, S. O. (2011). Construção automática de diretórios web usando agrupamento incremental de termos. In *ENIA'2011: Encontro Nacional de Inteligência Artificial*, páginas 1–12, Natal, RN, Brasil. Sociedade Brasileira da Computação - SBC. Citado nas páginas 5, 86, e 90.

Marchionini, G. (2006). Exploratory search: from finding to understanding. *Communications of ACM*, 49(4):41–46. Citado nas páginas 1, 9, e 75.

Metwally, A., Agrawal, D., e Abbadi, A. E. (2005). Efficient computation of frequent and top-k elements in data streams. In *ICDT'05: Proceedings of 10th International Conference on Database Theory*, páginas 398–412. Citado na página 47.

Metwally, A., Agrawal, D., e Abbadi, A. E. (2006). An integrated efficient solution for computing frequent and top-k elements in data streams. *ACM Transactions on Database Systems (TODS)*, 31:1095–1133. Citado na página 47.

Metz, J. (2006). Interpretação de clusters gerados por algoritmos de clustering hierárquico. Dissertação de mestrado, Instituto de Ciências Matemáticas e de Computação - ICMC - Universidade de São Paulo - USP. Citado na página 17.

Milligan, G. e Cooper, M. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179. Citado na página 23.

Moura, M. F. (2009). *Contribuições para a construção de taxonomias de tópicos em domínios restritos utilizando aprendizado estatístico*. Tese de doutorado, Instituto de Ciências Matemáticas e de Computação - ICMC - USP, São Carlos, SP. Citado na página 4.

Moura, M. F., Marcacini, R. M., e Rezende, S. O. (2008). Easily labelling hierarchical document clusters. In *WAAMD'08: IV Workshop em Algoritmos e Aplicações de Mineração de Dados, XXIII Simpósio Brasileiro de Banco de Dados - SBBD*, páginas 37–45. Porto Alegre : SBC. Citado na página 23.

Moura, M. F. e Rezende, S. O. (2010). A simple method for labeling hierarchical document clusters. In *IAI'10: Proceedings of the 10th IASTED International Conference on Artificial Intelligence and Applications*, páginas 363–371, Anaheim, Calgary, Zurich : Acta Press, 2010. Citado nas páginas 8 e 23.

Nassar, S., Sander, J., e Cheng, C. (2004). Incremental and effective data summarization for dynamic hierarchical clustering. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, SIGMOD'04, páginas 467–478, New York, NY, USA. ACM. Citado nas páginas 29, 30, 38, e 39.

Nguyen-Hoang, T.-A., Hoang, K., Bui-Thi, D., e Nguyen, A.-T. (2009). Incremental document clustering based on graph model. In *ADMA'09: Proceedings of the 5th International Conference on Advanced Data Mining and Applications*, páginas 569–576, Berlin, Heidelberg. Springer-Verlag. Citado na página 33.

Nishida, C. E. H., Marcacini, R. M., e Rezende, S. O. (2010). Um estudo sobre validação estatística para agrupamento hierárquico de documentos. In *XVIII Simpósio Internacional de Iniciação Científica da USP (SIICUSP)*. Universidade de São Paulo - USP. Citado na página 5.

Pang, S. (2010). Csmining group - the r8 of reuters 21578 data set. Página da internet acessada em 12/06/2010. http://csmining.org/index.php/r52-and-r8-of-reuters-21578.html. Citado nas páginas 58 e 59.

Pazzani, M. J. e Meyers, A. (2003). Nsf research award abstracts 1990-2003 data set. Página da internet acessada em 18/10/2010. http://archive.ics.uci.edu/ml/databases/nsfabs/. Citado na página 58.

Rennie, J. (2008). The 20 newsgroups data set. Página da internet acessada em 01/10/2010. http://people.csail.mit.edu/jrennie/20Newsgroups/. Citado na página 58.

Rezende, S. O., Pugliesi, J. B., Melanda, E. A., e Paula, M. F. (2003). Mineração de dados. In Rezende, S. O., editor, *Sistemas Inteligentes: Fundamentos e Aplicações*, chapter 12, páginas 307–335. Manole, 1ª edição. Citado nas páginas ix, 2, 8, e 12.

Rossi, R. R. (2010). The ACM digital library data set. Página da internet acessada em 27/10/2010. http://sites.labic.icmc.usp.br/marcacini/ihtc/datasets. Citado na página 58.

Sahoo, N., Callan, J., Krishnan, R., Duncan, G., e Padman, R. (2006). Incremental hierarchical clustering of text documents. In *ICIKM'06: Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, páginas 357–366, New York, NY, USA. ACM. Citado nas páginas 3, 29, 32, e 33.

Salton, G., Allan, J., e Singhal, A. (1996). Automatic text decomposition and structuring. *Information Processing & Management*, 32(2):127–138. Citado na página 12.

Salton, G. e Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *An International Journal of Information Processing and Management*, 24(5):513–523. Citado na página 11.

Sanderson, M. e Croft, B. (1999). Deriving concept hierarchies from text. In *SIGIR '99: Proceedings of the 22nd International Conference on Research and Development in Information Retrieval*, páginas 206–213, New York, NY, USA. ACM. Citado nas páginas 8 e 41.

Slonim, N. e Tishby, N. (2000). Document clustering using word clusters via the information bottleneck method. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, páginas 208–215, New York, NY, USA. ACM. Citado na página 41.

Sneath, P. (1957). The application of computers to taxonomy. *Journal of General Microbiology*, 17(1):201–226. Citado na página 18.

Soares, M. V. B., Prati, R. C., e Monard, M. C. (2008). Pretext ii: Descrição da reestruturação da ferramenta de pré-processamento de textos. Relatório Técnico 333, Instituto de Ciências Matemáticas e de Computação, USP, São Carlos. Citado nas páginas ix, 10, e 11.

Sokal, R. R. e Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 28:1409–1438. Citado na página 18.

Song, M. (2009). *Handbook of Research on Text and Web Mining Technologies*. Information Science Reference. Citado na página 27.

Sorensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter*, 5(5):35–43. Citado na página 18.

Steinbach, M., Karypis, G., e Kumar, V. (2000). A comparison of document clustering techniques. In *KDD'2000: Workshop on Text Mining*, páginas 1–20. Citado nas páginas 15, 17, 19, 25, 58, 64, e 65.

Tan, P.-N., Steinbach, M., e Kumar, V. (2005). *Introduction to Data Mining*. Addison-Wesley Longman Publishing, Boston, MA, USA. Citado nas páginas 13, 14, 19, 24, 25, e 65.

Vendramin, L., Campello, R. J. G. B., e Hruschka, E. R. (2010). Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining*, 3(4):209–235. Citado nas páginas 23 e 24.

Weiss, S. M., Indurkhya, N., Zhang, T., e Damerau, F. J. (2005). *Text Mining: Predictive Methods for Analizing Unstructured Information*. Springer Science Media. Citado na página 7.

Widyantoro, D., Ioerger, T., e Yen, J. (2002). An incremental approach to building a cluster hierarchy. In *ICDM'02: Proceedings of the 2002 IEEE International Conference on Data Mining*, página 705. Citado na página 32.

Witten, I. H. e Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, CA, 2ª edição. Citado na página 20.

Wong, W. e Fu, A. (2000). Incremental document clustering for web page classification. In *IEEE International Conference on Information Society in the 21st Century: Emerging Technologies and New Challenges*. Citado nas páginas 31 e 36.

Xu, R. e Wunsch, D. (2008). *Clustering*. Wiley-IEEE Press, IEEE Press Series on Computational Intelligence. Citado nas páginas ix, 2, 3, 12, 17, 20, 23, 24, 29, 30, e 38.

Yang, H.-C. e Lee, C.-H. (2004). A text mining approach on automatic generation of web directories and hierarchies. *Expert Systems with Applications*, 27(4):645–663. Citado na página 76.

Zamir, O. e Etzioni, O. (1998). Web document clustering: a feasibility demonstration. In *SIGIR'98: Proceedings of the 21st annual international ACM Conference on Research and Development in Information Retrieval*, páginas 46–54, New York, NY, USA. ACM. Citado na página 31.

Zeng, H.-J., He, Q.-C., Chen, Z., Ma, W.-Y., e Ma, J. (2004). Learning to cluster web search results. In *SIGIR'04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, páginas 210–217, New York, NY, USA. ACM. Citado nas páginas 8, 27, e 32.

Zhang, C. e Wu, D. (2008). Concept extraction and clustering for topic digital library construction. In *WI-IAT'08: Proceedings of International Conference on Web Intelligence and Intelligent Agent Technology*, páginas 299–302, Los Alamitos, CA, USA. IEEE Computer Society. Citado na página 27.

Zhang, T., Ramakrishnan, R., e Livny, M. (1996). Birch: an efficient data clustering method for very large databases. *ACM SIGMOD Record*, 25(2):103–114. Citado nas páginas 29, 31, 38, e 39.

Zhang, W., Yoshida, T., Tang, X., e Wang, Q. (2010). Text clustering using frequent itemsets. *Knowledge-Based Systems*, 23(5):379–388. Citado nas páginas 41, 45, e 84.

Zhao, Y. e Karypis, G. (2002). Evaluation of hierarchical clustering algorithms for document datasets. In *CIKM '02: Proceedings of the 11th international conference on Information and Knowledge Management*, páginas 515–524, New York, NY, USA. ACM. Citado nas páginas 19, 20, 25, 27, 58, e 65.

Zhao, Y., Karypis, G., e Fayyad, U. (2005). Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10(2):141–168. Citado nas páginas 2, 17, 20, 27, 58, 64, e 65.

Zhong, J. e Liu, J. (2010). Automatic construction of knowledge tree based on text clustering. *Application Research of Computers*, 27:475–478. Citado na página 27.

Zhong, S. (2005). Efficient online spherical k-means clustering. In *IJCNN'05: Proceedings. 2005 IEEE International Joint Conference on Neural Networks*, volume 5, páginas 3180 – 3185. Citado na página 35.

Zipf, G. K. (1932). *Selective Studies and the Principle of Relative Frequency in Language*. Harvard University Press. Citado na página 10.

Apêndice

A

Gráficos para Análise de Parâmetro para o Algoritmo Leader

Neste apêndice podem ser observadas os gráficos com os valores de Silhueta e Número de grupos, em comparação com o parâmetro de Dissimilaridade Mínima utilizado no Leader.

Todos os gráficos aqui apresentados foram analisados para configuração do algoritmo Leader. A melhor configuração para cada coleção textual foi selecionada na avaliação experimental apresentada no Capítulo 5.

Os parâmetros que maximizam o valor de Silhueta, em cada coleção textual, foram selecionados para a avaliação experimental, conforme apresentado na Tabela 5.4 do Capítulo 5.

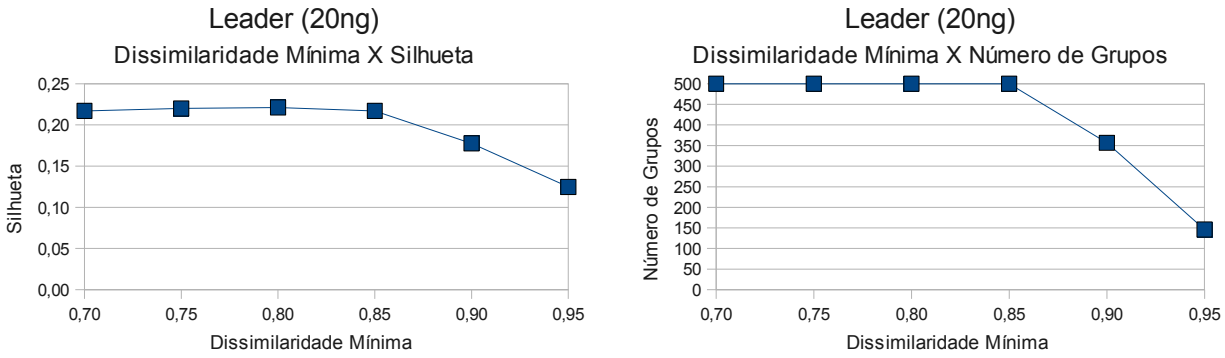

Figura A.1: Gráfico para análise de parâmetro do algoritmo Leader na coleção textual 20ng

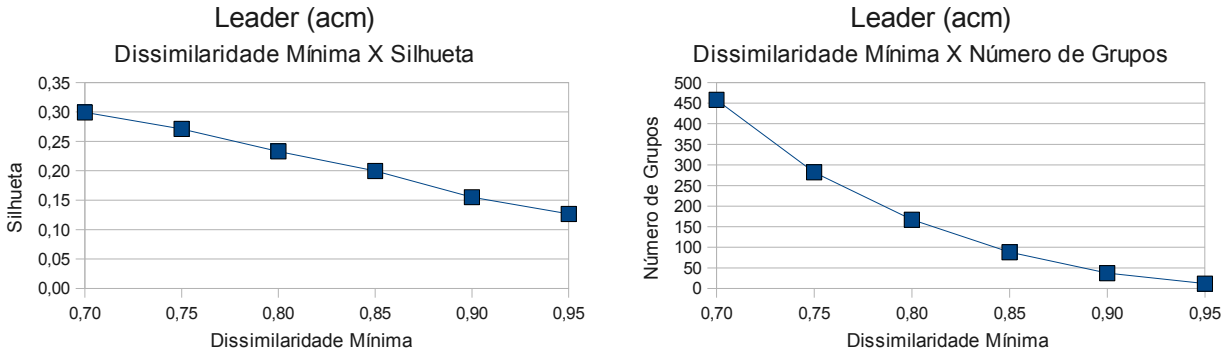

Figura A.2: Gráfico para análise de parâmetro do algoritmo Leader na coleção textual acm

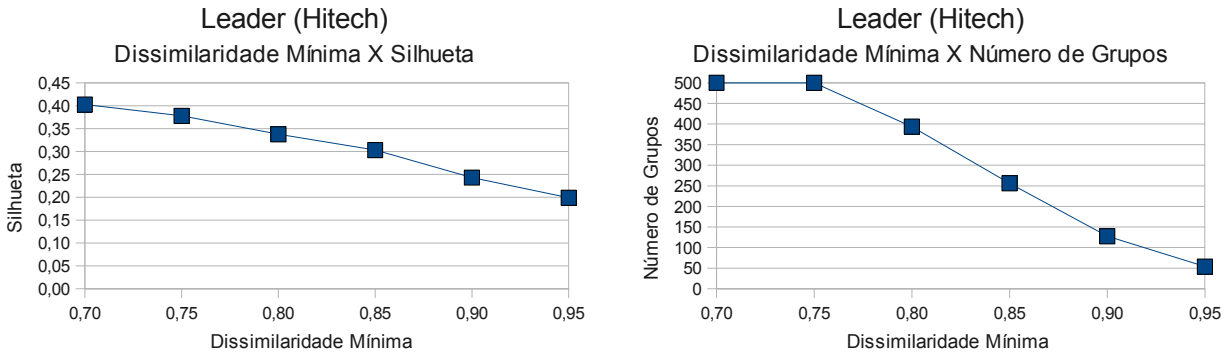

Figura A.3: Gráfico para análise de parâmetro do algoritmo Leader na coleção textual Hitech

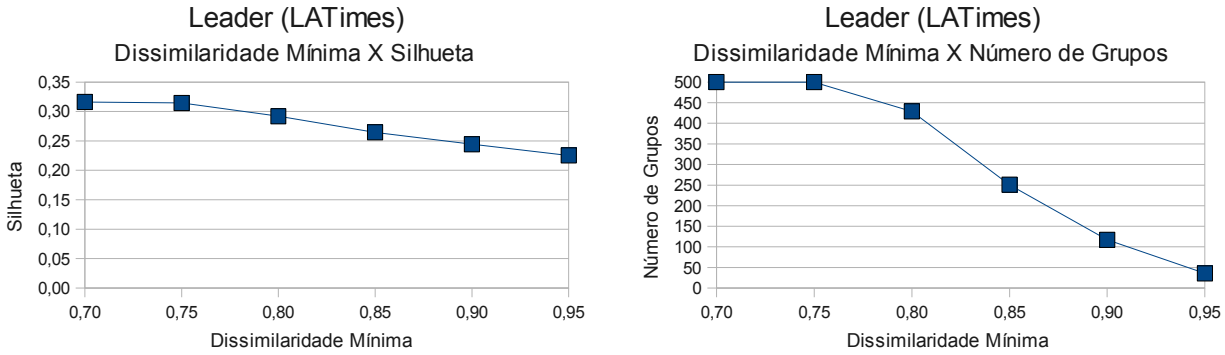

Figura A.4: Gráfico para análise de parâmetro do algoritmo Leader na coleção textual LATimes

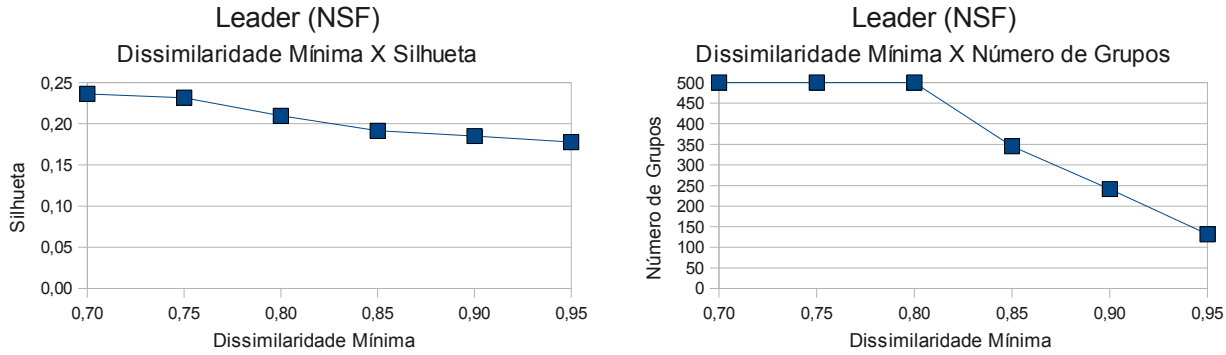

Figura A.5: Gráfico para análise de parâmetro do algoritmo Leader na coleção textual NSF

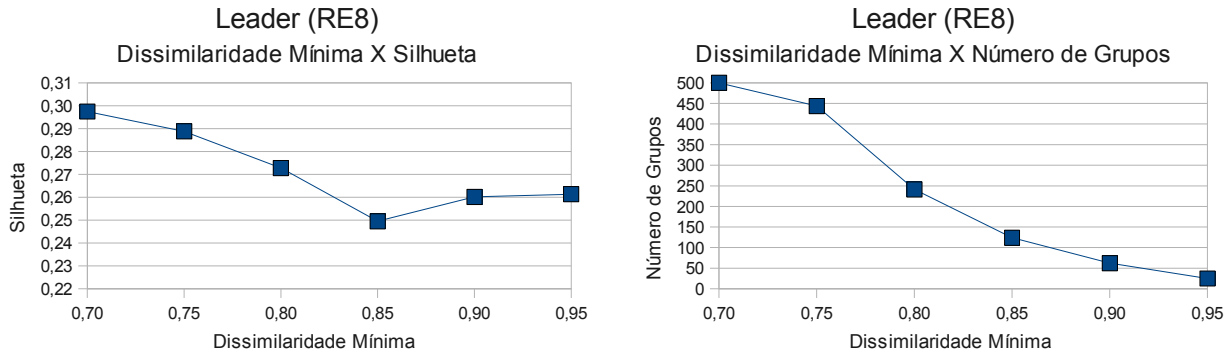

Figura A.6: Gráfico para análise de parâmetro do algoritmo Leader na coleção textual RE8

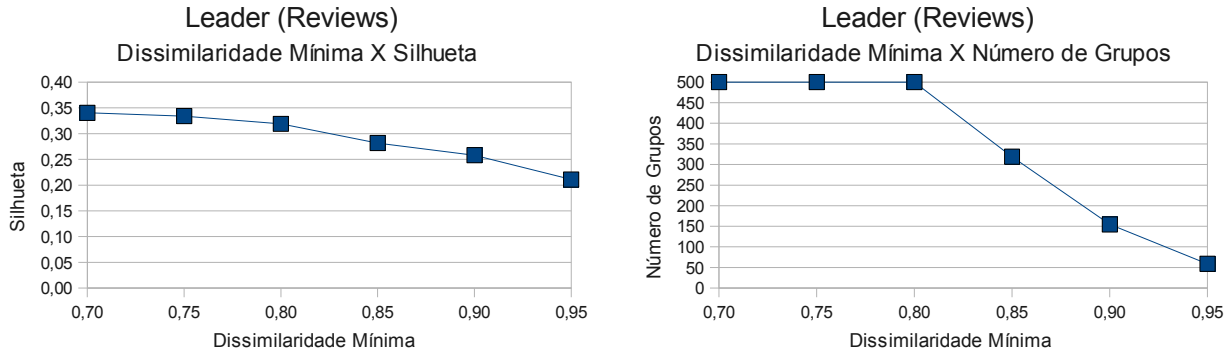

Figura A.7: Gráfico para análise de parâmetro do algoritmo Leader na coleção textual Reviews

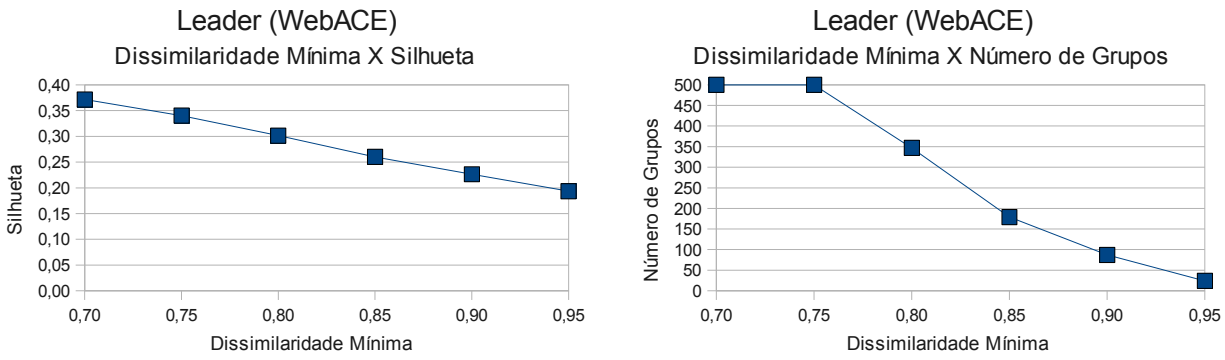

Figura A.8: Gráfico para análise de parâmetro do algoritmo Leader na coleção textual WebACE

Apêndice

B

Gráficos para Análise de Parâmetro para o Algoritmo DCTree

Neste apêndice podem ser observadas os gráficos com os valores de Silhueta e Número de grupos, em comparação com o parâmetro de Dissimilaridade Mínima utilizado no DCTree.

Todos os gráficos aqui apresentados foram analisados para configuração do algoritmo DCTree. A melhor configuração para cada coleção textual foi selecionada na avaliação experimental apresentada no Capítulo 5.

Os parâmetros que maximizam o valor de Silhueta, em cada coleção textual, foram selecionados para a avaliação experimental, conforme apresentado na Tabela 5.4 do Capítulo 5.

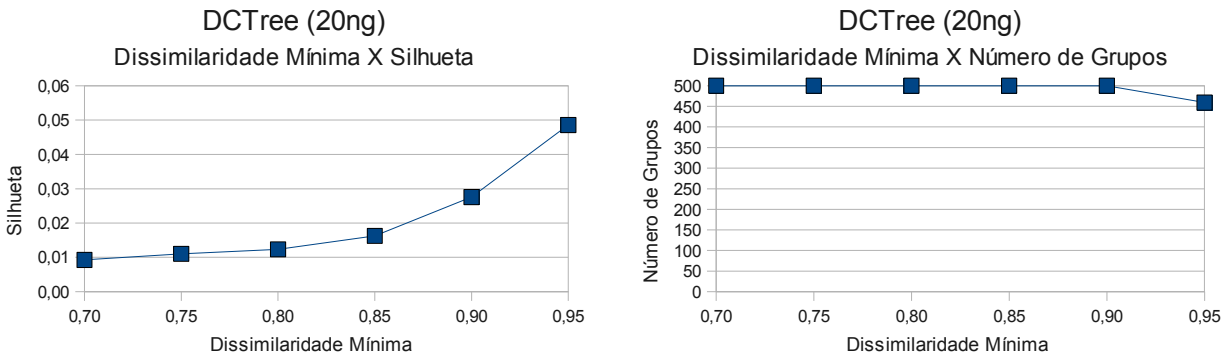

Figura B.1: Gráfico para análise de parâmetro do algoritmo DCTree na coleção textual 20ng

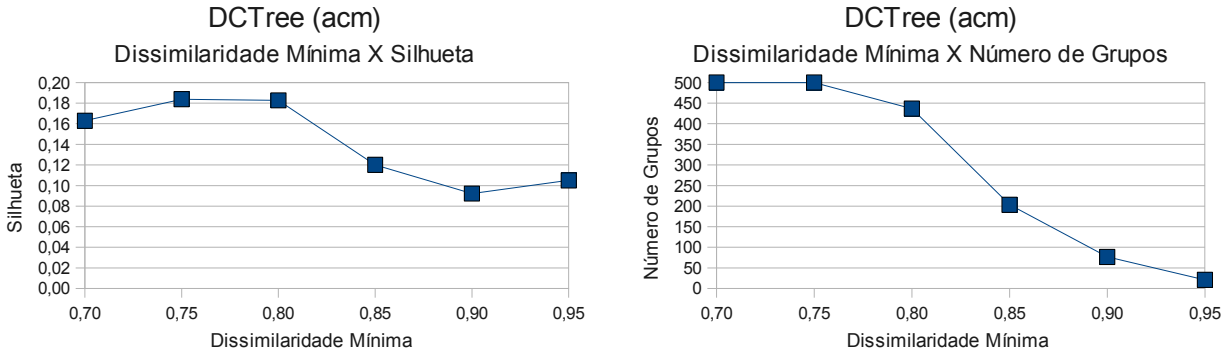

Figura B.2: Gráfico para análise de parâmetro do algoritmo DCTree na coleção textual acm

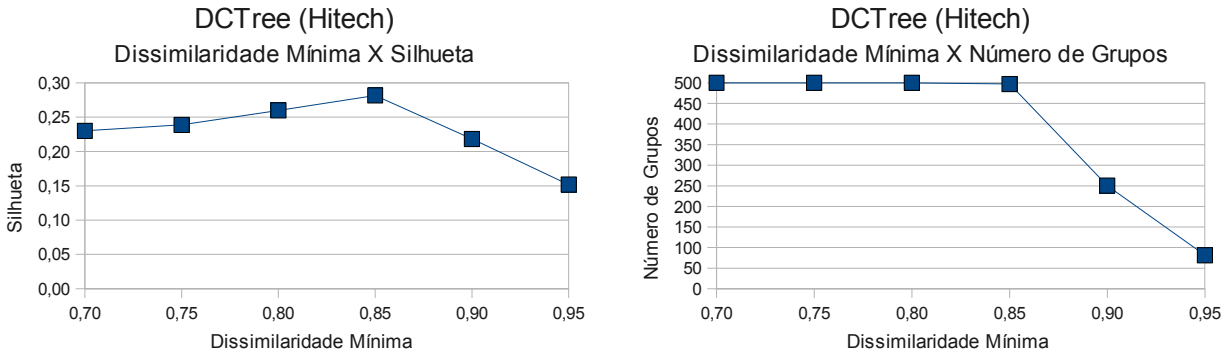

Figura B.3: Gráfico para análise de parâmetro do algoritmo DCTree na coleção textual Hitech

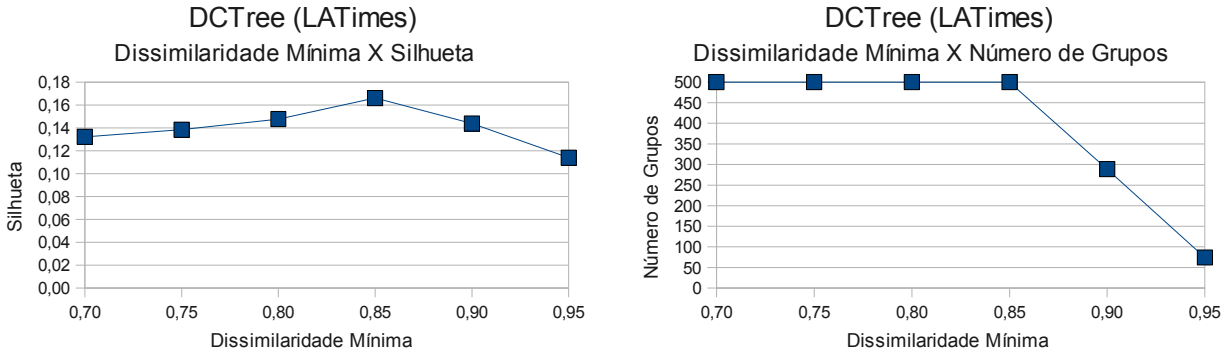

Figura B.4: Gráfico para análise de parâmetro do algoritmo DCTree na coleção textual LATimes

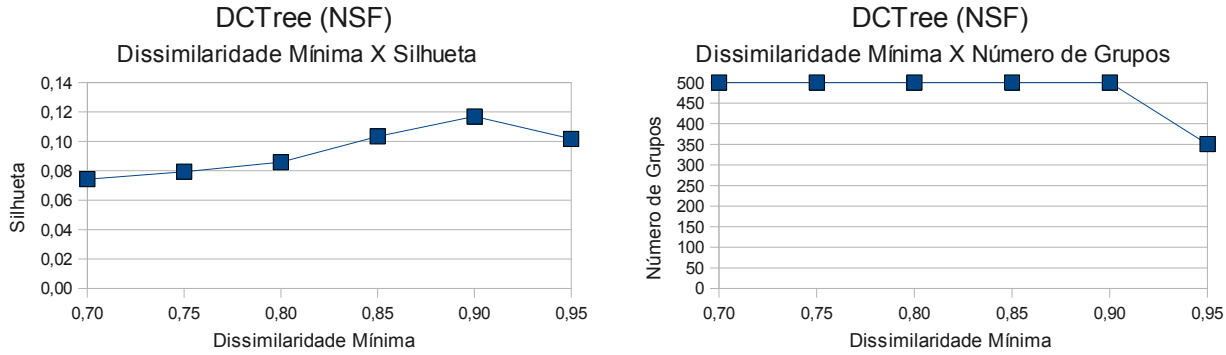

Figura B.5: Gráfico para análise de parâmetro do algoritmo DCTree na coleção textual NSF

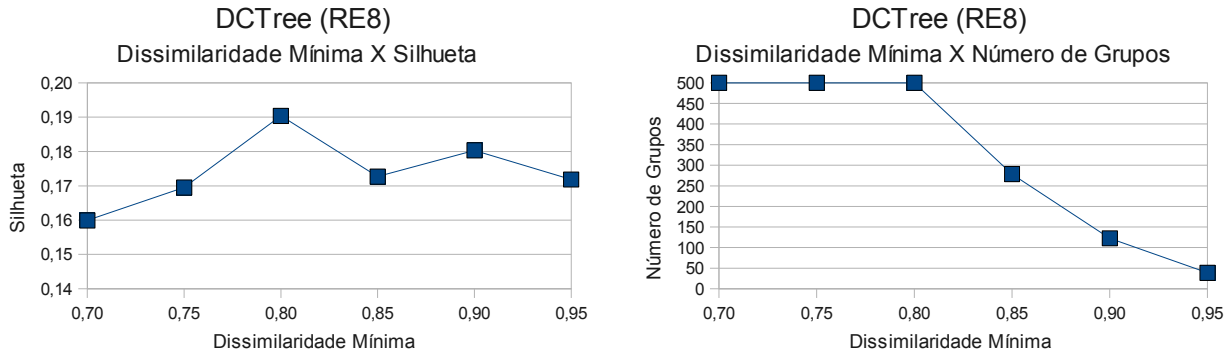

Figura B.6: Gráfico para análise de parâmetro do algoritmo DCTree na coleção textual RE8

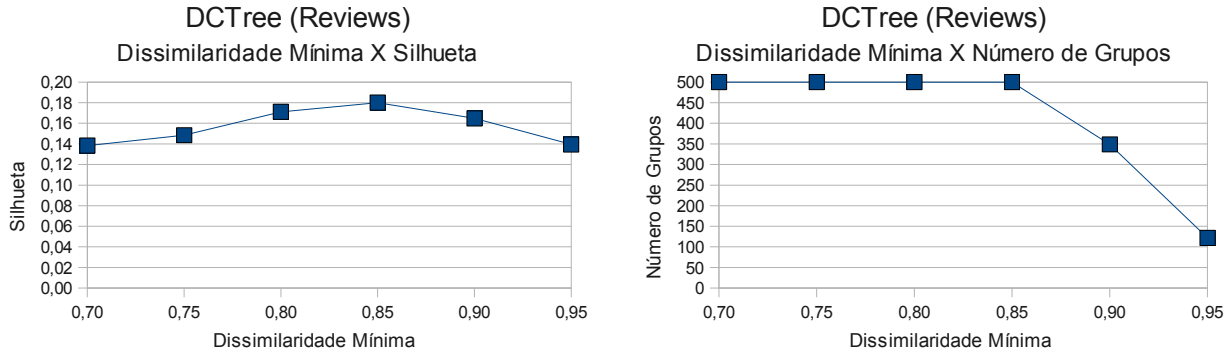

Figura B.7: Gráfico para análise de parâmetro do algoritmo DCTree na coleção textual Reviews

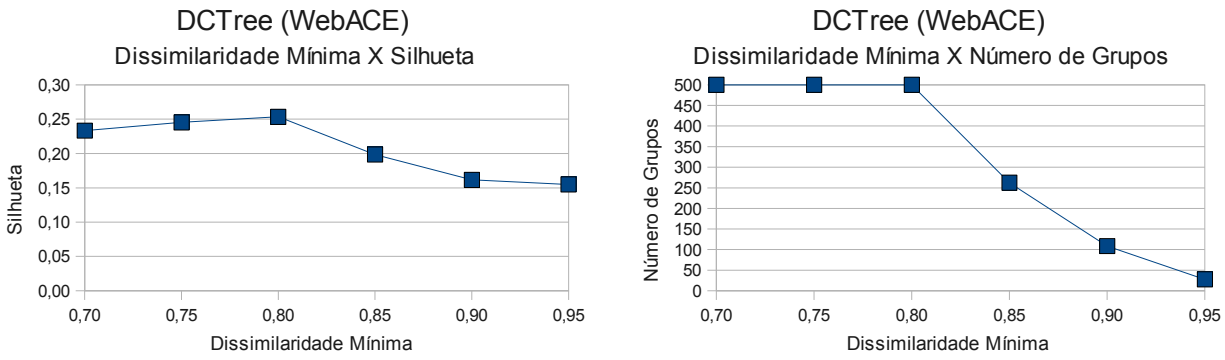

Figura B.8: Gráfico para análise de parâmetro do algoritmo DCTree na coleção textual WebACE

Apêndice

C

Gráficos para Análise de Parâmetro para o Algoritmo IHTC

Neste apêndice podem ser observadas os gráficos com os valores de Silhueta em comparação com o parâmetro de Número Máximos de Arestas da Rede de Coocorrência de Termos utilizado no IHTC.

Todos os gráficos aqui apresentados foram analisados para configuração do algoritmo IHTC. A melhor configuração para cada coleção textual foi selecionada na avaliação experimental apresentada no Capítulo 5.

Os parâmetros que maximizam o valor de Silhueta, em cada coleção textual, foram selecionados para a avaliação experimental, conforme apresentado na Tabela 5.4 do Capítulo 5.

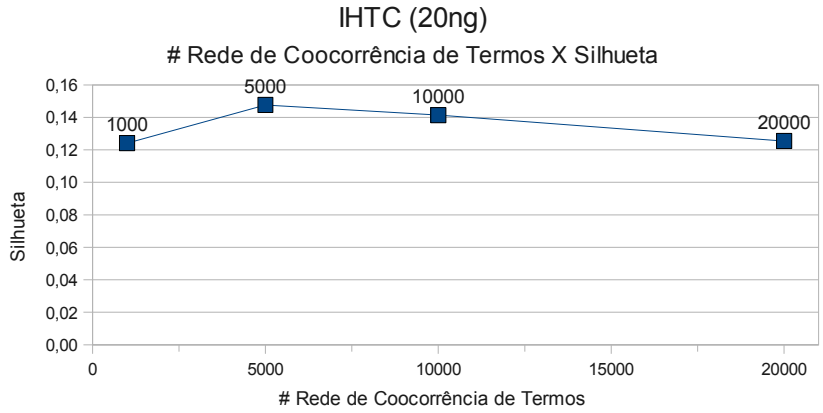

Figura C.1: Gráfico para análise de parâmetro do algoritmo IHTC na coleção textual 20ng.

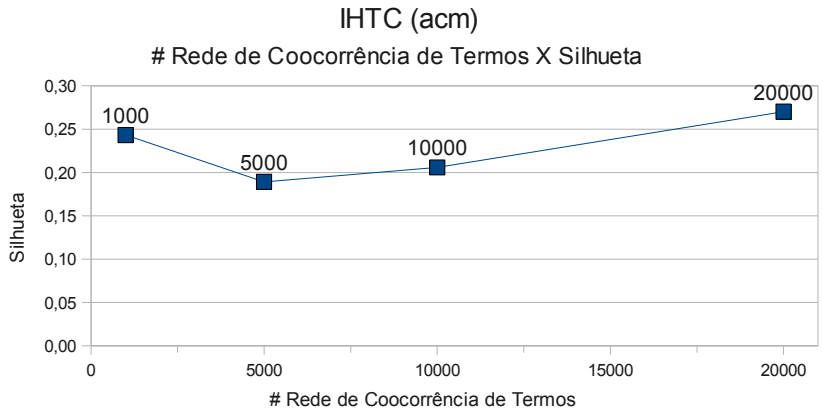

Figura C.2: Gráfico para análise de parâmetro do algoritmo IHTC na coleção textual acm.

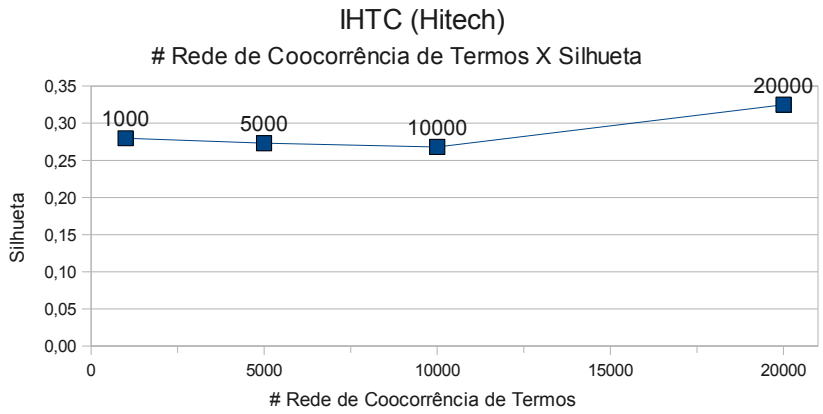

Figura C.3: Gráfico para análise de parâmetro do algoritmo IHTC na coleção textual Hitech.

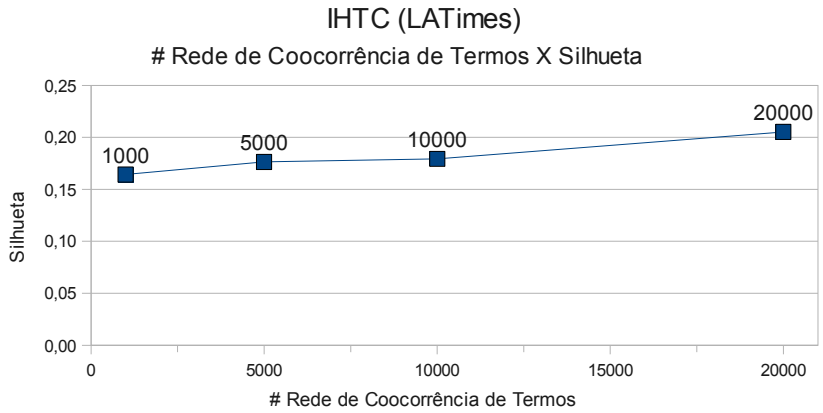

Figura C.4: Gráfico para análise de parâmetro do algoritmo IHTC na coleção textual LATimes.

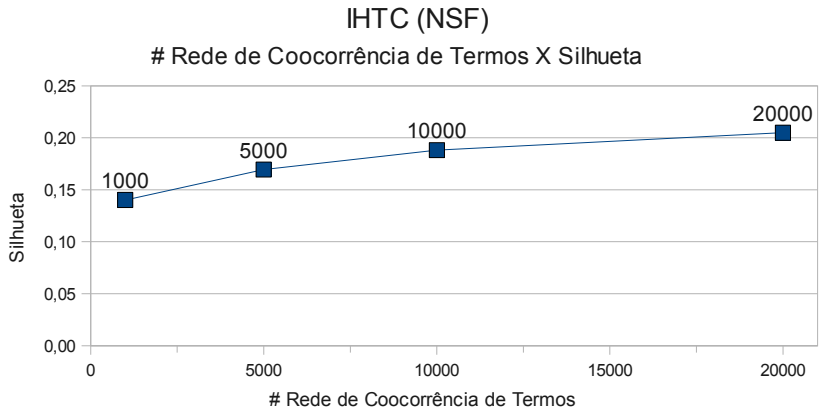

Figura C.5: Gráfico para análise de parâmetro do algoritmo IHTC na coleção textual NSF.

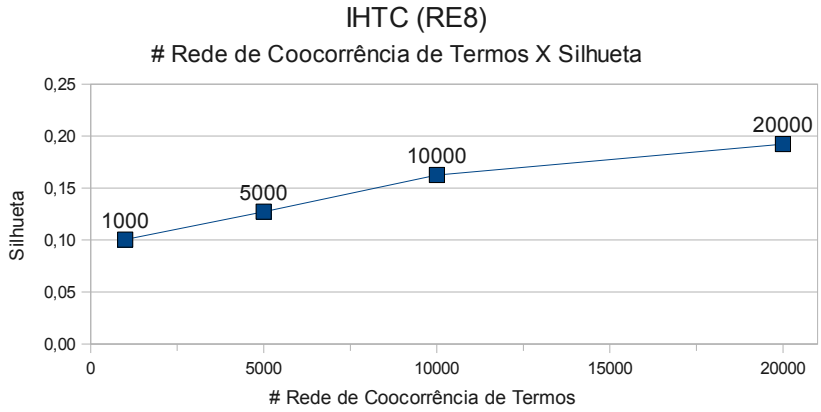

Figura C.6: Gráfico para análise de parâmetro do algoritmo IHTC na coleção textual RE8.

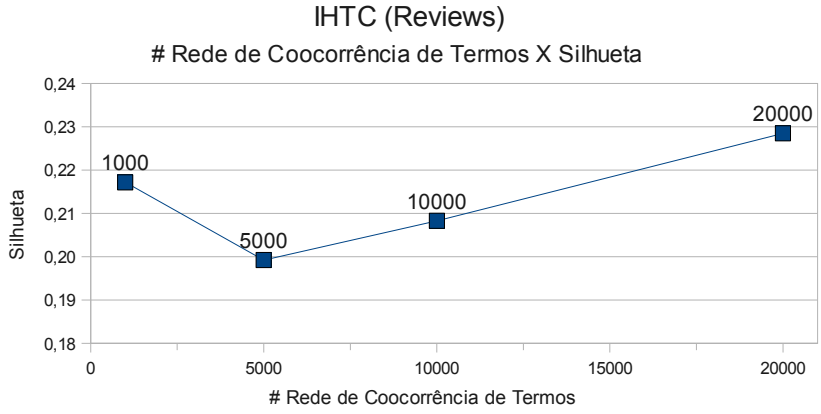

Figura C.7: Gráfico para análise de parâmetro do algoritmo IHTC na coleção textual Reviews.

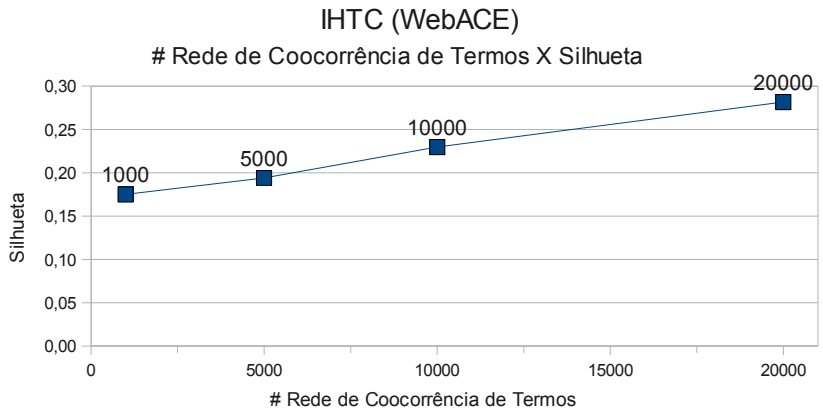

Figura C.8: Gráfico para análise de parâmetro do algoritmo IHTC na coleção textual WebACE.

Apêndice

D

Tabelas com os Resultados do Estudo Exploratório do Projeto Dmoz

Neste apêndice podem ser observadas as tabelas com os valores de eficácia de recuperação do agrupamento hierárquico, eficácia de recuperação dos descritores e os próprios descritores obtidos nos experimentos para construção automática de diretórios web. Na Tabela D.1 são apresentados valores obtidos com base no IHTC. Já na Tabela D.2 são apresentados valores obtidos com base no Leader.

Todos os valores aqui apresentados foram utilizados para análise estatísticas e para construção dos gráficos apresentados na Seção 6.3 do Capítulo 6.

Tabela D.1: Descritores selecionados com base no método IHTC em comparação com o Dmoz

Diretórios do Dmoz Manual	IHTC - Descritores Automático	Fscore	
		Agrupamento	Descritores
Accounting	{Services, Taxes}, {Accounting, Services}, {Accounting, Taxes}	0,593	0,611
Air Compressors	{Aire, Compression}, {Compression, Services}, {Compression, Powerful}	0,451	0,220
Air Dispersion Modeling	{Located, Models}, {Gas, Models}, {Assessment, Models}	0,597	0,190
Alpacas	{Alpacas, Breeders}, {Alpacas, Farming}, {Alpacas, Located}	0,478	0,807
Arabian	{Arabians, Horses}, {Arabians, Performing}, {Arabians, Straights}	0,550	0,697
Auctions	{Auctions, Collection}, {Auctions, Sales}, {Company, Liquidations}	0,675	0,285
Batteries	{Batteries, Manufacture}, {Chargers, Manufacture}, {Leading, Manufacture}	0,436	0,421
Bowling	{Bowling, Centers}, {Bowling, Leagues}, {Bowling, News}	0,530	0,548
Cancer	{Cancers, Treatments}, {Cancers, Researchers}, {Cancers, Information}	0,335	0,499
Coffee	{Coffee, Products}, {Distributors, Espressos}, {Offered, Wholesale}	0,510	0,246
Cosmetic and Plastic	{Plastics, Surgery}, {Cosmetic, Surgery}, {Information, Surgery}	0,563	0,557
Domain Names	{Domains, Nics}, {Cctlds, Domains}, {Cctlds, Nics}	0,561	0,329
Fire Fighting	{Department, Fires}, {Fires, Volunteer}, {Fires, Information}	0,541	0,793
Franchising	{Franchise, Offered}, {Forms, Opportunities}, {Faqs, Forms}	0,634	0,550
Hospitals	{Hospital, Information}, {Community, Hospital}, {Hospital, Opportunities}	0,335	0,427
Hosting	{Hosting, Offered}, {Domains, Hosting}, {Hosting, Webbing}	0,500	0,363
Industrial Yarns and Sewing	{Knitted, Yarns}, {Knitted, Weaving}, {Weaving, Yarns}	0,582	0,645
Karate	{Karate, Shotokan}, {Events, Karate}, {History, Karate}	0,415	0,585
Land Trusts	{Lands, Trusts}, {Lands, Protection}, {Protection, Trusts}	0,467	0,776
Leather and Fur	{Footwear, Leathers}, {Leathers, Manufacture}, {Goods, Leathers}	0,631	0,484
Molding	{Customs, Molding}, {Manufacture, Molding}, {Customs, Injection}	0,614	0,426
Money Managers	{Financial, Investment}, {Investment, Planning}, {Advisory, Investment}	0,406	0,228
NCAA Division II	{Basketball, Coaching}, {Mens, Results}, {Mens, Statistics}	0,622	0,219
Nursing	{Nurse, Registered}, {Nurse, Offered}, {Nurse, University}	0,375	0,444
Personal Chefs	{Chefs, Menus}, {Chefs, Samples}, {Chefs, Services}	0,751	0,756
Robotics	{Researchers, Robotics}, {Mobile, Robotics}, {Controls, Robotics}	0,418	0,379
Search and Rescue	{Rescue, Searches}, {Searches, Volunteer}, {Searches, Teams}	0,428	0,737
Sewing Machines	{Equipment, Sewing}, {Industry, Needle}, {Machine, Representatives}	0,519	0,318
Signage	{Customs, Signs}, {Manufacture, Signs}, {Banners, Signs}	0,544	0,482
Skiing	{Clubs, Skiing}, {Resorts, Skiing}, {Skiing, Snowboarding}	0,631	0,640
Translation	{English, Translations}, {Germans, Translations}, {Frenchs, Translations}	0,706	0,684
Venture Capital	{Focus, Investment}, {Focus, Stages}, {Investment, Stages}	0,583	0,727
Veterinarians	{Animals, Services}, {Animals, Hospital}, {Hospital, Services}	0,706	0,691
Voice Talent	{Commercial, Voice}, {Radios, Voice}, {Demos, Voice}	0,517	0,580
Wedding and Events	{Photography, Wedding}, {Photography, Portraits}, {Portraits, Wedding}	0,701	0,761

Tabela D.2: Descritores selecionados a partir dos centroides do agrupamento com base no Leader em comparação com o Dmoz

Diretórios do Dmoz	**Leader - Descritores**	**Fscore**	
Manual	**Automático**	**Agrupamento**	**Descritores**
Accounting	Taxes, Accounting, Cpas	0,655	0,655
Air Compressors	Aire, Compressors, Pollution	0,201	0,067
Air Dispersion Modeling	Models, Logics, Dispersion	0,066	0,104
Alpacas	Alpacas, Huacayas, Suris	0,755	0,882
Arabian	Horses, Arabians, Quarters	0,337	0,182
Auctions	Auctions, Antiques, Consignment	0,667	0,296
Batteries	Batteries, Chargers, Devices	0,445	0,134
Bowling	Bowling, Lanes, Tenpin	0,613	0,622
Cancer	Cancers, Breasts, Nci	0,332	0,582
Coffee	Coffee, Espressos, Roasters	0,466	0,502
Cosmetic and Plastic	Surgery, Cosmetic, Facials	0,569	0,302
Domain Names	Cctlds, Nics, Domains	0,667	0,371
Fire Fighting	Fires, Apparatus, Volunteer	0,775	0,422
Franchising	Franchise, Opportunities, Requests	0,686	0,341
Hospitals	Hospital, Appraisal, Healthcare	0,444	0,124
Hosting	Hosting, Domains, Webbing	0,311	0,234
Industrial Yarns and Sewing	Yarns, Weaving, Spun	0,549	0,499
Karate	Karate, Shotokan, Ryus	0,467	0,701
Land Trusts	Lands, Trusts, Surveyors	0,582	0,152
Leather and Fur	Leathers, Footwear, Tanned	0,622	0,439
Molding	Molding, Injection, Modularization	0,550	0,302
Money Managers	Investment, Advisors, Advisory	0,260	0,041
NCAA Division II	Games, Items, Scores	0,099	0,209
Nursing	Nurse, Geological, Geology	0,420	0,470
Personal Chefs	Chefs, Hypnosis, Meals	0,721	0,442
Robotics	Robotics, Estimators, Autonomous	0,524	0,486
Search and Rescue	Rescue, Trucks, Volunteer	0,450	0,184
Sewing Machines	Machine, Cncs, Sewing	0,176	0,063
Signage	Signs, Signage, Letters	0,553	0,526
Skiing	Skiing, Snowboarding, Resorts	0,655	0,700
Translation	Translations, Interpreting, English	0,691	0,481
Venture Capital	Venture, Stages, Early	0,645	0,457
Veterinarians	Veterinary, Hours, Hospital	0,782	0,430
Voice Talent	Narration, Voiceover, Promos	0,514	0,573
Wedding and Events	Photography, Wedding, Portraits	0,629	0,745