Mapeamento de dados multidimensionais usando árvores filogenéticas: foco em mapeamento de textos

Ana Maria Cuadros Valdivia

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 10 de setembro de 2007

Assinatura: _____

Mapeamento de dados multidimensionais usando árvores filogenéticas: foco em mapeamento de textos

Ana Maria Cuadros Valdivia

Orientadora: ***Profa. Dra. Rosane Minghim***

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

USP - São Carlos
Setembro/2007

Agradecimentos

Agradeço a meus pais por estarem sempre a meu lado me apoiando em todos os momentos.

A minha orientadora Rosane Minghim e ao professor Guilherme Telles, pela oportunidade de compartilhar seus conhecimentos e idéias, assim como pelo apoio, paciência, compreensão e colaboração que tiveram durante o desenvolvimento deste trabalho.

Ao meus colegas do ICMC-USP, Fernando, Roberto, Alex e Waldo pelos seus conselhos e por compartilharem comigo seus conhecimentos. A Eduardo e Luciana pela sua amizade.

Resumo

A Visualização Computacional trata de técnicas para representar e interagir graficamente com dados complexos, em geral de alta dimensionalidade. Dados de alta dimensionalidade são caracterizados por pontos representados em espaços vetoriais de alta dimensão, cada coordenada representando um atributo do vetor. Num grande número de aplicações da visualização multidimensional uma medida de similaridade existe entre esses vetores. Técnicas de projeção multidimensional podem ser utilizadas para posicionamento desses dados num plano de forma a facilitar a interpretação das relações de similaridade. Entretanto alguns problemas dessas técnicas comprometem a interpretação dos resultados obtidos. Este trabalho identifica esses problemas e propõe, uma técnica para posicionar os pontos no plano, através da formação de árvores filogenéticas a partir de relações de similaridade. Em geral árvores filogenéticas são utilizadas para codificação de relações de ancestralidade. Um algoritmo de geração e um algoritmo de traçado dessas árvores foram implementados no contexto do sistema PEx (Projection Explorer) e a solução é comparada com a funcionalidade das projeções na interpretação de dados multidimensionais em geral e, em particular, na representação de coleções de documentos, uma aplicação bastante estratégica da visualização computacional e da mineração visual de dados.

Abstract

Computational Visualization is concerned with graphical representation and exploration of complex data, usually bearing high dimensionality. Multidimensional data are characterized by points represented in vector spaces of many dimensions, each coordinate representing an attribute of the vector. In many applications a similarity measure can be found to highlight relationships of proximity between the vectors. In this environment projection techniques offer an alternative to ease interpretation coded by the similarity measures through proximity on the display. They do so by positioning the points on a bidimensional plane. Projection techniques are very useful to display and interact with data, but present some drawbacks that in some cases compromise the interpretation of certain features in data sets. This work discusses such problems and proposes, as an alternative to represent similarity relationships and to provide point placement on a plane, the use of phylogenetic trees, a representation typically employed to represent ancestrality relationships. An algorithm for generation and an algorithm for drawing such trees were implemented in a system called Projection Explorer. The approach is compared to that of multidimensional projections for multidimensional data in general and, in particular, for document data sets, an strategic application for multidimensional visualizations, since text can be represented and interpreted as multi-dimensional entities.

Sumário

Agradecimentos **i**

Resumo **iii**

Abstract **v**

Lista de Figuras **ix**

Lista de Tabelas **xiii**

1 Introdução **1**
1.1 Contextualização 1
1.2 Motivação e Objetivo 4
1.3 Organização 5

2 Conceitos Básicos: Projeção de dados e Árvores filogenéticas **7**
2.1 Considerações Iniciais 7
2.2 Técnicas de redução de dimensionalidade (projeções multidimensionais) 8
2.2.1 Análise de Componentes Principais (PCA) 8
2.2.2 *Latent semantic index* (LSI) 9
2.2.3 *Multidimensional Scaling* (MDS) 9
2.2.4 Projeção de Sammon *Mapping* 10
2.2.5 *Pathfinder network scaling* 10
2.2.6 *Nearest Neighbor Projection* (NNP) 11
2.2.7 Fastmap 11
2.2.8 *Force Scheme* 11
2.2.9 Projeção por agrupamento (ProjClus) 12
2.2.10 Projeção de mínimos quadrados (LSP) 12
2.3 Árvores Filogenéticas 13
2.3.1 Técnica *Neighbor Joining* (NJ) 16
2.3.2 Algoritmo e exemplo de aplicação 18
2.3.3 Discussão e trabalhos correlatos 22

2.4 Considerações finais 23

3 Mapeamento visual de textos baseados em conteúdo 25
3.1 Considerações Iniciais 25
3.2 Processo de visualização de mapas de textos baseados em conteúdo 26
3.3 Técnicas de visualização de mapas de textos 30
3.3.1 *IN-SPIRE* 31
3.3.2 *Self-Organization Maps* (SOMs) 32
3.3.3 ET-Map 32
3.3.4 gCLUTO 33
3.3.5 InfoSky 33
3.3.6 *VxInsight* 35
3.3.7 *Projection Explorer (PEx)* 35
3.4 Visualização de dados usando árvores filogenéticas 37
3.5 Considerações Finais 40

4 Implementação Visual de Filogenias na Construção de Mapas de Textos 41
4.1 Considerações Iniciais 41
4.2 Processo de construção de mapas de textos usando filogenia 42
4.3 Visão geral do sistema Projection Explorer (Pex) 43
4.3.1 Funcionalidade do PEx para construção de mapas de dados 43
4.3.2 Estrutura interna do sistema PEx 45
4.4 Implementação e processo de construção da árvore filogenética 49
4.4.1 Descrição das funcionalidades de interação 51
4.5 Exemplo de aplicação para visualizar mapas de textos usando filogenia 56
4.6 Considerações finais 64

5 Discussão de Resultados e Conclusões 65
5.1 Considerações Iniciais 65
5.2 Análise dos resultados 66
5.3 Comparação da técnica NJ com técnicas de Projeção 71
5.3.1 Comparação visual entre as técnicas NJ e projeção 72
5.3.2 Comparação entre NJ e projeções relativas ao posicionamento dos pontos no mapa de visualização 76
5.3.3 Tempo de execução 79
5.4 Conclusões 80

Referências Bibliográficas 83

Lista de Figuras

1.1 Visão geral da construção dos mapas de textos baseados em distâncias (Paulovich e Minghim, 2006b). 3
1.2 Processo de construção de mapa de textos usando filogenia. 5

2.1 Representação de uma árvore filogenética. 13
2.2 Características de uma árvore filogenética. (A) Árvore filogenética com raiz e (B) árvore filogenética sem raiz. Ambas árvores possuem a mesma topologia. Na árvore com raiz A, B, C, D, E e F são nós externos ou OTU's. G, H, I, J e K são nós internos ou ancestrais hipotéticos. As setas indicam direção de evolução. A árvore sem raiz carece de nó raiz, as linhas entre os nós são os ramos e não se conhece direção de evolução (Salemi e Vandamme, 2003). 14
2.3 Exemplos de tipos de árvores baseadas em distância (a) árvores ultramétricas e (b) árvores aditivas. 15
2.4 Condição dos quatro pontos métricos. As letras em minúscula representam o comprimento dos ramos (Salemi e Vandamme, 2003). 15
2.5 Árvore sem raiz para a técnica *neighbor joining*. 16
2.6 (a) Árvore estrela e (b) Árvore onde objetos 1 e 2 (i e j) estão agrupados. 17
2.7 Matriz de distâncias inicial (a) e Topologia da árvore (b). 18
2.8 Passos do algoritmo *Neighbor Joining*: Primeira iteração. 20
2.9 Passos do algoritmo *Neighbor Joining*: Segunda iteração. 21
2.10 Passos do algoritmo *Neighbor Joining*: Terceira iteração. 21
2.11 Passos do algoritmo *Neighbor Joining*: Quarta iteração. 22

3.1 Visualização de textos incluíndo agrupamento e projeção (Spence, 2001). 27
3.2 Metáforas de visualização em *SPIRE* http://www.pnl.gov/infoviz. 31
3.3 Mapa SOM (Borner *et al.*, 2003). 32
3.4 Mapa do projeto ET-Map (Borner *et al.*, 2003). 33
3.5 Metáforas de visualização em gCLUTO (Rasmussen e Karypis, 2004). 34
3.6 Visualização DO sistema Infosky (Granitzer *et al.*, 2004). 34
3.7 VxInsight cria visualizações utilizando a metáfora de montanha (Boyack *et al.*, 2002). . 35
3.8 Funcionalidades da ferramenta PEx. 36

3.9 Principais *layouts* das árvore filogenética (Carrizo, 2004). 37
3.10 *Hypertree* (Bingham e Sudarsanam, 2000). 38
3.11 *PhyloDraw* com *layout* radial (Choi *et al.*, 2000). 38
3.12 *TreeWiz* (Rost e Bornberg-Bauer, 2002). 39
3.13 *Walrus* : *Layout* hiperbólico 3D (http://www.caida.org/tools/visualization/walrus). . . 39

4.1 Processo geral da construção de árvores filogenéticas usando conjunto de textos. . . . 42
4.2 Diagrama geral de pacotes no PEx. 45
4.3 Diagrama das principais classes dos pacotes ***corpus*** e **preprocessing** do PEx. 46
4.4 Diagrama das principais classes do pacote ***projection***. 47
4.5 Diagrama das principais classes do pacote ***graph*** e ***force***. 48
4.6 Diagrama das principais classes do pacote **view**. 48
4.7 Diagrama das principais classes do pacote **wizard**. 49
4.8 Organização do pacote NJ no contexto do PEx. 50
4.9 Representação angular do *layout* radial (Bachmaier *et al.*, 2005). 51
4.10 Visualização da árvore usando os dois *layouts* no PEx. 53
4.11 Formas de selecionar nós na árvore. 54
4.12 Criação do rótulo do ramo no conjunto de textos CBR+ILP+IR. 54
4.13 Coordenação de múltiplas visões aplicado no conjunto de textos CBR+ILP+IR. 55
4.14 Coloração do ramo como resultado de selecionar um ramo do conjunto de textos CBR+ILP+IR. 55
4.15 Ligação de 26 textos vizinhos ao vértice circundado com uma linha grossa amarela. . . 56
4.16 Processo de construção de mapas de textos empregando a técnica NJ na ferramenta PEx. 57
4.17 Representação visual da árvore do conjunto de textos *News* na janela principal do PEx. 58
4.18 Processo de identificar nós individualmente. 58
4.19 Exploração da árvore por ramos. 59
4.20 Exibição de rótulos depois da aplicação do algoritmo de agrupamento *k-means* sobre o conjunto *news*. 60
4.21 Exploração de informação por ramos usando principalmente funcionalidades de ampliação, seleção de ramos e deslocamentos de ramos. As notícias são posicionadas nos ramos conforme o cálculo da similaridade entre nós. 61
4.22 Busca de informação no conjunto *news* coloridas de acordo à freqüência do termo *president*. 62
4.23 Representação topológica da árvore para o conjunto *news* gerada por meio do sistema PEx. 63
4.24 Representação topológica da árvore para o conjunto *news* no PEx. 63
4.25 Nos vizinho do nó "Tornadoes Sweep Through Tennessee" para o conjunto de textos News. 64

5.1 Mapa de textos CBR+ILP+IR+SON. A técnica NJ é capaz de agrupar textos baseados em conteúdo em regiões bem definidas. 67
5.2 Rótulo por dois ramos do conjunto CBR+ILP+IR+SON. A função de rótulo permite mostrar coerência entre a posição dos textos e seu conteúdo. 69
5.3 Exemplo do mapa gerado para o conjunto de artigos científicos KDVIS. 69

5.4 Ampliação de uma parte do ramo do conjunto CBR+ILP+IR+SON, para mostrar a similaridade entre os nós filhos. 70
5.5 Mapa construído para o conjunto de dados *Quadruped Mammals*. 71
5.6 Mapa gerado pela técnica NJ para o conjunto *news* no PEx. 73
5.7 Mapa gerado pela técnica de projeção LSP para o conjunto *news* no PEx. . . . 73
5.8 Construção do mapa de textos InfoVis com a técnica NJ. 74
5.9 Construção do mapa de textos InfoVis com a técnica de projeção LSP. 75
5.10 Precisão na construção visual do mapa para o conjunto de textos CBR-ILP-IR-SON usando NJ e Projeções, para distâncias baseadas no cosseno. 77
5.11 Mapas de textos CBR-ILP-IR-SON. Técnica NJ (a), IDMAP (b), LSP (c), Sammon mapping (d) e PCA (e). 78
5.12 Precisão na construção visual do mapa para o conjunto de textos ALL usando NJ e Projeções, para distâncias diretas NCD. 78
5.13 Mapas de textos *ALL*, Técnica (a) NJ , (b) IDMAP, (c)LSP, (d) ProjClus (e) Sammon *mapping*. 79

Lista de Tabelas

5.1 Conjunto de dados utilizados para mostrar a validade da técnica NJ. 66
5.2 Tempo gasto para construir um mapa de textos (em segundos). 80

CAPÍTULO

1

Introdução

1.1 Contextualização

Na última década, a Internet tornou-se um importante meio de comunicação, pesquisa, lazer, comércio e serviços e conseqüentemente, a informação disponível cresceu consideravelmente. Isso motivou o crescimento das técnicas de organização e visualização de informação. Além disso, o aumento da capacidade dos meios eletrônicos de armazenamento ampliou a codificação de informação na forma de textos. As técnicas que lidam com textos precisam proporcionar ferramentas flexíveis de representação de resultados quando o usuário realiza buscas por itens de interesse e adquere conhecimento das coleções de textos.

Uma forma de representar e explorar textos é construindo um mapa de textos, que segundo Lopes *et al*. (2006) consiste em uma representação gráfica de uma coleção de textos que reflete as relações de similaridade entre eles, suas relevâncias e a possível organização de áreas e sub-áreas relacionadas a uma consulta.

Na literatura existem diversas técnicas para realizar mapeamento de textos, técnicas que segundo o estudo feito em Borner *et al*. (2003) abordam dois problemas importantes no processo de análise e visualização do conhecimento: 1) enormes quantidades de dados devem ser mostrados em um espaço limitado e 2) dados multidimensionais precisam ser exibidos em espaços de duas dimensões. O primeiro problema é minimizado aplicando-se técnicas de visualização,

interação e distorção ((Shneiderman, 1996), (Keim, 2002)). Já para o segundo problema existem as técnicas de agrupamento, as de posicionamento baseado em força e as de redução de dimensionalidade ((Berkhin, 2002), (Fruchterman e Reingold, 1991), (Borner *et al.*, 2003)); cada uma das quais apresenta vantagens e desvantagens.

No que diz respeito às técnicas de agrupamento, a principal vantagem é que elas provêem uma visão geral das relações de similaridade existentes entre os grupos de textos. No entanto, as relações internas dos grupos e as entre grupos são perdidas. As técnicas de posicionamento de pontos baseadas em força evitam realizar cálculos custosos para determinar a similaridade entre os textos, baseando seu cálculo em ajustes dos pontos por forças de atração e repulsão. Entretanto, podem se tornar ineficientes rapidamente e são altamente influenciadas por mínimos locais. As técnicas de redução de dimensionalidade, embora usadas amplamente para extrair estruturas de uma coleção de textos, tem limitações, como perda de informação ao mapear espaços de n dimensões em espaços 3D ou 2D. Esses espaços podem definir grupos de textos similares, mas localmente é impossível relacionar textos vizinhos (Huang *et al.*, 2005), (Lopes *et al.*, 2006). Tais técnicas possuem um custo computacional elevado. Dentre elas existe um grupo de técnicas, também denominadas por alguns autores de projeções multidimensionais, que são mais rápidas e tentam preservar o máximo possível as relações de similaridade entre os pontos (textos) quando mapeadas em espaços de baixas dimensões. A aplicação dessas técnicas foi testada com êxito em trabalhos prévios de Tejada *et al*. (2003) e Paulovich *et al*. (2006). Neste texto as expressões redução de dimensionalidade e projeções serão usadas indistintamente, significando técnicas que fazem mapeamentos entre domínios de dimensões diferentes (da maior para menor) procurando preservar alguma característica relevante do espaço original.

Muitas das técnicas de redução de dimensionalidade que empregam algum tipo de cálculo de similaridade entre os textos, realizam o seguinte processo para construir seus respectivos mapas de textos, como se ilustra na Figura 1.1 (Paulovich e Minghim, 2006b):

1. Pré-processamento da coleção de dados de texto;
2. Projeção multidimensional e triangulação;
3. Mapeamento de informação adicional.

No primeiro estágio, se realiza o cálculo da similaridade entre os textos em um espaço multidimensional. Uma forma de calcular essa similaridade é construir uma representação vetorial de cada texto e calcular a distância entre os vetores que representam cada texto (Salton *et al.*, 1975). Os componentes do vetor são alguma medida de freqüência de ocorrência de palavras no texto. Essa técnica é muito rápida, mas exige ajustes manuais de uma série de parâmetros com

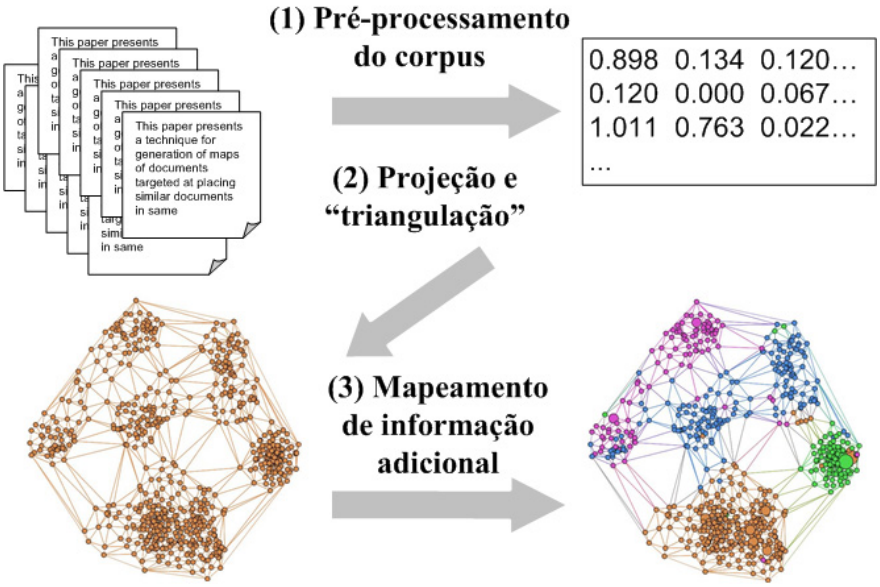


Figura 1.1: Visão geral da construção dos mapas de textos baseados em distâncias (Paulovich e Minghim, 2006b).

significado e comportamento não óbvio (Lopes *et al.*, 2006). Outra forma de calcular a similaridade entre os textos é por meio do cálculo de aproximações da complexidade de Kolmogorov, que fornece uma medida da quantidade de informação em um texto. Essa técnica dispensa a representação vetorial e seus ajustes, mas é consideravelmente mais lenta (Telles *et al.*, 2007). Depois do cálculo das similaridades, obtém-se uma matriz triangular de similaridades.

No segundo estágio, aplica-se alguma das técnicas de agrupamento, de posicionamento baseado em força ou de redução de dimensionalidade, as quais posicionam os pontos que representam os textos em um plano. Alguma técnica de melhoria do resultado pode ser usada em seguida (Tejada *et al.*, 2003). Uma vez que os pontos são posicionados no plano, eles podem ser apresentados na forma de um grafo ou triangulação. Finalmente, adiciona-se atributos visuais ao mapa, como cor de vértices e arestas, tamanho e forma dos pontos, etc. Representações visuais alternativas, como superficies e volumes, também podem ser utilizadas.

Vários mapas têm sido construídos para conjuntos variados de textos, incluindo *newsgroups*, notícias curtas ou longas de jornal, artigos científicos, patentes, *e-mails*, etc. Os resultados são bastante precisos no que diz respeito ao posicionamento dos textos e são muito bons visualmente (Paulovich e Minghim, 2006b). Técnicas dessas classes foram implementadas no ICMC-USP, numa ferramenta denominada *Projection Explorer*-PEx (Paulovich *et al.*, 2007b).

Há, no entanto, uma série de melhorias que podem ser introduzidas no processo de mapeamento. Algumas dessas melhorias estão relacionadas a vários problemas existentes na construção de mapas, como: (1) a projeção dos pontos representados pela matriz de similaridades,

embora precisa, posiciona uma fração dos pontos em vizinhanças inadequadas e (2) o posicionamento local dos pontos nas projeções dificulta a compreensão do relacionamento entre pontos de uma vizinhança, que podem ser posicionados praticamente um sobre o outro em mapas de tamanho moderado.

Para contornar esses problemas, este trabalho propõe construir mapas de textos a partir da matriz de similaridades entre eles, removendo a projeção multidimensional do processo. Para isso, usamos heurísticas para o problema de reconstrução de árvores filogenéticas.

1.2 Motivação e Objetivo

Como mencionado na seção anterior, o emprego de técnicas de projeção na construção de mapas de textos obteve bons resultados mas ainda há problemas que motivaram o desenvolvimento deste trabalho.

A mesma idéia que embasa a construção de mapas de textos é analisada na área de bioinformática, mas para outros tipos de dados e outro contexto. Referimo-nos à técnica de filogenia, que é amplamente usada por pesquisadores para tentar reconstruir as relações de evolução (proximidade e diferença) que existem entre as espécies (Swofford e Sullivan, 2003).

Uma filogenia ou árvore filogenética é representada por um grafo acíclico em que as folhas são objetos, como por exemplo, espécies ou indivíduos. A topologia da árvore indica relações de ancestralidade entre os objetos. Nós internos nessas árvores representam ancestrais hipotéticos. Um exemplo da árvore resultante é apresentado na Figura 1.2. O tamanho dos caminhos na árvore representa a distância evolucionária entre objetos. No caso de textos, as árvores serão construídas de tal forma que cada folha da árvore represente um texto e nós internos na árvore representem textos hipotéticos com conteúdo intermediário entre os textos reais.

O problema de reconstrução de filogenias é definido para dois tipos de entradas: matrizes de características e matrizes de distâncias. Casos sem ambigüidades entre as caracteristicas ou em que as distâncias são uma métrica podem ser resolvidos em tempo polinomial. Os outros casos são NP-difíceis (Gusfield, 1997).

Existem diferentes métodos de filogenia (Felsenstein, 2003), (Swofford e Sullivan, 2003), mas a principal família de métodos filogenéticos são os métodos baseados em matriz de distâncias, que foram influenciados por algoritmos de agrupamento de Sokal *et al.* (1963).

Uma das heurísticas mais usadas para a reconstrução de filogenias a partir de matrizes de distâncias é a *neighbor-joining* (Saitou e Nei, 1987). Essa heurística identifica o par de objetos mais próximos tratando de minimizar o comprimento da árvore, e que aplicada a textos tornará

mais claras as relações de similaridade entre eles, reduzindo alguns problemas associados às projeções.

A Figura 1.2 ilustra como é feito o processo de construção do mapa de textos empregando filogenia comparado com o processo que seguem as técnicas de construção de mapas de textos tradicionais.

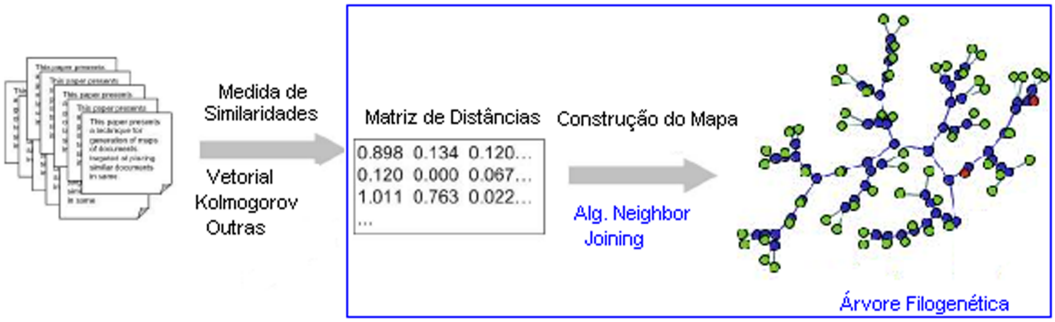


Figura 1.2: Processo de construção de mapa de textos usando filogenia.

Portanto, o objetivo deste projeto de mestrado constitui-se em demonstrar que o algoritmo de filogenia *neighbor-joining* pode ser aplicado com sucesso na construção de mapas de textos, e em avaliar seus resultados em comparação às técnicas de projeção.

1.3 Organização

Este trabalho é organizado da seguinte forma: no Capítulo 2 são apresentados os conceitos básicos pertinentes a este trabalho, relacionados principalmente a redução de multidimensionalidade e filogenia. O Capítulo 3 detalha o processo de mapeamento visual de textos e revisa as principais técnicas existentes para realizá-lo. O Capítulo 4 descreve a implementação da técnica NJ incorporada à ferramenta PEx, assim como o processo exploratorio que segue a criação de mapas de textos aplicando filogenia. No Capítulo 5 são discutidos os resultados obtidos ao comparar técnicas de projeção multidimensional com a técnica NJ em termos de visualização, posicionamento dos pontos no *layout* e tempo de processamento, ressaltando as vantagens e desvantagens da técnica NJ e finalizando com as conclusões do trabalho.

CAPÍTULO

2

Conceitos Básicos: Projeção de dados e Árvores filogenéticas

2.1 Considerações Iniciais

No contexto deste projeto, um dado, ou vetor, é um elemento do conjunto de dados a ser visualizado; um atributo, ou dimensão, corresponde a uma variável amostrada para cada dado. O conceito de multidimensionalidade se refere ao número de atributos ou dimensões (implícito ou explícito) necessário para descrever um domínio. Uma coleção de textos é considerada um conjunto de dados multidimensionais[1]. Essa coleção ou conjunto de textos típicamente é convertida numa representação vetorial para tornar possível o reconhecimento de padrões e estruturas nos dados a serem visualizados. Conseqüentemente, visualização de informação faz uso de técnicas de agrupamento, estratégias de posicionamento de pontos ou redução de dimensionalidade, para posicionar os itens de dados em espaços de duas ou três dimensões. Dentre elas, nosso estudo foi centrado em técnicas de redução de dimensionalidade, porque permitem mapear dados originais (com *n* atributos) em um espaço *m*-dimensional ($m < n$), de forma tal, que a distância entre os dados corresponda ao grau de similaridade que existe entre eles (Huang *et al.*, 2005), e que as relações existentes sejam refletidas no mapa de visualização.

[1]Em visualização de informação de textos, dados multidimensionais se refere a instâncias de dados com 4 ou mais atributos (Paulovich *et al.*, 2006).

Enquanto as razões para realizar redução de dimensionalidade são claras, os problemas inerentes a elas ainda estão em aberto (Huang *et al.*, 2005):

1. Desconhecimento da dimensionalidade: Não temos uma forma eficiente de encontrar o número mínimo de dimensões suficientes para representar adequadamente os dados.

2. Relacionamento não-linear entre os dados: Reconhecer relações fundamentais entre as variáveis em geral não é trivial.

3. Desconhecimento de informação relevante: Existe uma grande tendência a perder certa quantidade de informação quando aplicadas técnicas de redução de dimensionalidade.

Devido à natureza complexa dos problemas citados acima, na literatura foi proposta uma vasta diversidade de técnicas redução de dimensionalidade (Paulovich e Minghim, 2006a), (Berkhin, 2002). Na Seção 2.2 são descritas algumas das técnicas de redução de dimensionalidade relevantes para a análise visual de coleções de textos.

Além das técnicas de redução de dimensionalidade, foi estudada a técnica *neighbor joining* (NJ), como uma alternativa para a solução dos problemas inerentes às técnicas de redução de dimensionalidade. A NJ é usada na construção de árvores filogenéticas que empregam princípios de evolução entre as espécies para refletir as relações de similaridade entre elas. Os conceitos principais de filogenia, assim como a técnica NJ, são apresentados na Seção 2.3.

2.2 Técnicas de redução de dimensionalidade (projeções multidimensionais)

A redução de dimensionalidade tem por finalidade representar conjuntos de dados de grandes dimensões em espaços de dimensões menores, tornando mais fácil a interpretação, compreensão e extração de informação. Nesta seção serão abordadas algumas das técnicas mais freqüentemente propostas para reduzir a dimensionalidade dos textos.

2.2.1 Análise de Componentes Principais (PCA)

A análise de componentes principais (Jolliffe, 1986) é uma técnica que procura transformar linearmente um conjunto orignal de dados que possuem um alto grau de dependência entre eles (correlações) em um conjunto de menor dependência e dimensionalidade. Esses dados transformados são combinações lineares das variáveis originais, chamados de componentes principais

(CP). Existem tantos componentes principais quanto o número original de atributos, sendo que os primeiros CP identificados buscam preservar ao máximo a variância dos dados originais, esse valor vai decaindo gradualmente nos últimos CP.

As características principais da PCA são: permitir identificar as tendências dos atributos mais relevantes, conseguir capturar a maior variabilidade dos dados em poucas dimensões e permitir reduzir o ruído, redundância e ambigüidade (Ding, 2000), (Paulovich e Minghim, 2006a). A principal desvantagem é a determinação do número correto de dimensões. Se o número é pequeno tende-se a perder características importantes dos dados. Se o número é grande, se capturam características importantes, mas a representação visual é difícil. Outra desvantagem é o custo computacional, $O(n^2)$, que é inaceitável quando o número de elementos dos conjuntos de dados é muito grande, por exemplo, a partir de mil vetores.

2.2.2 *Latent semantic index* (LSI)

Latent semantic index (Deerwester *et al.*, 1990) é uma técnica que obtém conhecimento dos conjuntos de textos por extrair termos de significado similar aplicando uma técnica de álgebra não-linear chamada de decomposição de valores singulares (SDV).

Na LSI, primeiro se constrói a matriz *A* textos x termos, que é descomposta usando decomposição de valores singulares (SVD) em $A = USV^T$. As matrizes *U* e *V* são ortogonais e *S* é uma matriz diagonal com ordem de valor singular na diagonal. As colunas da matriz *U* formam bases ortogonais de um subespaço onde os vetores com mais alto valor singular possuem mais informação.

A principal característica da *LSI* é que termos sinônimos tendem a agrupar-se em uma mesma dimensão. No entanto, nem sempre a projeção reflete adequadamente o espaço original (Chen, 2004), (Lopes *et al.*, 2006). Outra desvantagem é seu elevado custo computacional, que depende do cálculo da SVD (tipicamente $O(n^2)$).

2.2.3 *Multidimensional Scaling* (MDS)

Multidimensional Scaling (Cox e Cox, 2000) é um conjunto de técnicas matemáticas que permitem descobrir estruturas ocultas nos dados a partir do cálculo de uma medida de similaridade entre eles, denominada proximidade. Essa medida pode ser obtida de formas diferentes, por exemplo, por calcular o coeficiente de correlação ou a distância Euclideana de um vetor que representa um texto. O que MDS faz é mapear um espaço multidimensional em um espaço de baixa dimensionalidade, em que cada objeto é representado por um ponto e a distância entre os pontos é proporcional ao grau de semelhança à informação original. Quanto mais alta a

dissimilaridade entre os objetos, mais distantes eles deveriam aparecer em um espaço de baixa dimensionalidade. Essas configurações geométricas de pontos refletem estruturas ocultas dos dados (Huang *et al.*, 2005).

Uma variante deste método é combiná-lo com estratégias baseadas em forças e interpolação como *Force Directed Placement* - FDP (Fruchterman e Reingold, 1991), em que cada objeto é associado a outros por uma força que é proporcional à distância calculada no espaço original. A complexidade para conseguir uma representação estável em espaços de duas ou três dimensões é $O(n^3)$. Em Chalmers (1996) se propôs uma abordagem para reduzir a complexidade do cálculo das forças em cada iteração. O cálculo das distâncias é realizado entre cada objeto e duas listas de elementos. A primeira lista S_i é formada por *Vmax* elementos escolhidos aleatoriamente em cada iteração e a segunda lista V_i armazena os *Smax* elementos mais próximos e é mantida entre iterações. A complexidade obtida é $O(n \times (Vmax + Smax))$, mas como são necessarias no mínimo *n* iterações, o processo como um todo terá complexidade $O(n^2)$. Uma melhoria nessa abordagem foi proposta por Morrison *et al*. (2004) que conseguiram reduzir a complexidade total do algoritmo para $O(n^{\frac{5}{4}})$. Outra abordagem foi apresentada em Jourdan *et al*. (2004) visando reduzir a complexidade para $O(n \log n)$.

2.2.4 Projeção de Sammon *Mapping*

A técnica Sammon *Mapping* (Sammon, 1969) procura preservar os dados de um espaço multidimensional num espaço de dimensão inferior, aplicando uma função que tenta minimizar a quantidade de informação perdida durante o processo do cálculo das similaridades entre textos. Essa função é otimizada aplicando procedimentos não-lineares de busca local em direção do gradiente da função. O custo computacional da técnica, $O(n^2)$, representa uma desvantagem.

2.2.5 *Pathfinder network scaling*

Uma técnica muito usada por algumas abordagens de visualização é *Pathfinder network scaling* (Chen, 1998), cujo objetivo é construir um grafo para preservar as relações de proximidade. No grafo, cada dado é representado por um nó e o peso ou tamanho das arestas é inversamente proporcional à similaridade entre dois textos. A técnica consiste em eliminar as arestas cuja distância seja maior do que o caminho mínimo encontrado para conectar os textos. Com isso se assegura que o grafo seja visualizado com um pequeno número de arestas.

A principal desvantagem da técnica é seu elevado custo computacional $O(n^4)$.

2.2.6 *Nearest Neighbor Projection* (NNP)

A técnica *nearest neighbor projection*, proposta por Tejada *et al.* (2003), projeta os dados com base na similaridade entre eles. Seu objetivo é preservar as relações existentes entre vizinhos locais do espaço multidimensional no espaço projetado. O processo consiste em encontrar dois pontos, q' e r', no espaço projetado que estejam o mais próximo possível a um novo ponto x' a ser projetado. A posição de x', encontra-se na interseção entre os dois círculos com centro nos pontos q' e r', e cujos raios são dados pela distância entre o ponto x e os pontos q e r no espaço multidimensional original. Quando não há interseção entre os círculos, o ponto x' é localizado em um ponto médio entre os círculos determinados. Quando há interseção, mas os círculos não são tangentes, duas soluções são possíveis e qualquer delas é escolhida.

A complexidade computacional do algoritmo é de $O(n^2)$ e somente pode se aplicar em espaços métricos onde se cumpra a propriedade de desigualdade triangular.

2.2.7 Fastmap

A técnica Fastmap (Faloutsos e Lin, 1995) projeta pontos de um espaço n-dimensional em um espaço *m*-dimensional $(n > m)$, visando preservar as relações de distância no espaço original. Em primeiro lugar, no Fastmap são selecionados dois pontos que estejam o mais distante possível, denominados pivôs, que definem uma reta no espaço *n*-dimensional e um hiperplano perpendicular à dita reta, num espaço $(n-1)-$dimensional. Posteriormente, os pontos restantes são projetados nesse hiperplano. Esse processo continua até que se atinja o número de dimensões desejadas.

O custo computacional da Fastmap é $O(n)$, sendo pouco efetivo quando se trata de revelar estruturas não-lineares.

2.2.8 *Force Scheme*

O algoritmo *Force Scheme* foi proposto por Tejada *et al.* (2003) para melhorar a precisão dos resultados obtidos pelas técnicas de projeção. Essa técnica é baseada no conceito de forças de atração e repulsão entre os objetos, realizando deslocamentos em direções determinadas.

O algoritmo consiste em, para cada ponto projetado y_i, calcular um vetor $\overline{v}_{ij} = (y_j - y_i)$ $\forall y_i \neq y_j$. A seguir, o ponto y_j é movido na direção do vetor $\overline{v}$ uma fração de Δ. No algoritmo, Δ representa o erro relativo dos pontos projetados y_i e y_j, que consiste na aproximação existente entre a distância no espaço reduzido (projetado) e a distância no espaço original. Uma vez que

todos os pontos sejam processados, se realiza a normalização das coordenadas da projeção na faixa [0,1].

A complexidade computacional do algoritmo é $O(n^2)$, mas a vantagem sobre outras técnicas baseadas em força é que no *force* em cada iteração os objetos alteram seu posicionamento $(n-1)$ vezes, ao contrário do que acontece com outras técnicas em que seu posicionamento é alterado só uma vez por iteração.

2.2.9 Projeção por agrupamento (ProjClus)

A técnica ProjClus (Paulovich e Minghim, 2006b) foi desenvolvida especialmente para projetar conjuntos de textos multidimensionais para apresentá-los em espaços bi-dimensionais. Os passos que segue são: no primeiro estágio, se divide o conjunto de textos em $\sqrt{n}$ agrupamentos, aplicando o algoritmo de agrupamento *k-means* por bisseção (Tan *et al.*, 2006) para calcular o centróide de cada grupo. Em seguida, em cada grupo formado se aplica o algoritmo Fastmap (Seção 2.2.7) e uma técnica de melhoria de projeções baseada em forças, denominada *Force Scheme* (Tejada *et al.*, 2003) (Seção 2.2.8). O resultado dessa projeção é normalizado. Por último, os agrupamentos resultantes são posicionados no espaço visual de acordo ao valor de seu centróide.

A complexida do algoritmo é $O(n^{3/2})$ e a característica principal dessa técnica é que preserva a proximidade dos pontos em uma vizinhança ao espaço *n*-dimensional original. Porém, estruturas importantes podem se perder no resultado final, devido principalmente à criação de grupos na fase inicial.

2.2.10 Projeção de mínimos quadrados (LSP)

A técnica projeção de mínimos quadrados foi desenvolvida por Paulovich *et al.* (2006), com o objetivo de criar uma superfície onde os textos estejam agrupados por relações de proximidade para que o usuário possa inferir as relações existentes no conjunto de textos. A LSP foi baseada no estudo feito por Sorkine *et al.* (2004) o qual aplica mínimos quadrados na recuperação e edição de malhas (*least-square meshes*). A reconstrução geométrica da malha é feita a partir do cálculo da distância entre um grupo de pontos representativos, denominados pontos-de-controle, onde são aplicados cálculos de distância para preservar as relações de vizinhança entre todos os pontos.

A técnica LSP realiza dois processos principais: no primeiro, são escolhidos um sub-conjunto de pontos (pontos-de-controle), resultantes de aplicar a técnica de agrupamento *k-medoids*. Em seguida, esses pontos são projetados fazendo uso de qualquer técnica de projeção convencional.

No segundo passo, se constrói um sistema linear baseado nas relações de vizinhança dos pontos em seu espaço original $\mathbb{R}^n$ e nas coordenas cartesianas dos pontos-de-controle no espaço reduzido $\mathbb{R}^m$.

A complexidade computacional é determinada pelo número de agrupamentos. Para $\sqrt{n}$ agrupamentos será $O(n\sqrt{n})$ (Paulovich *et al.*, 2007a).

Não existe ainda um método que lide com todos os problemas inerentes à redução de dimensionalidade, embora exista um grande número de algoritmos. A maioria das técnicas de redução de dimensionalidade é computacionalmente custosa e sensível ao número de dimensões originais. No entanto continua-se realizando pesquisas para melhorar o posicionamento dos dados em espaços de visualização. O uso de algoritmos típicos de árvores filogenéticas serve como uma alternativa às projeções multidimensionais para o posicionamento dos pontos baseado na similaridade entre eles. Este problema será analisado na seção seguinte.

2.3 Árvores Filogenéticas

O estudo do processo de evolução de grupos de espécies, bem como as relações existentes entre eles, é denominado filogenia (Setubal e Meidanis, 1997). Essas relações de evolução são verificadas pela construção de árvores filogenéticas cuja topologia indica relações de ancestralidade entre os organismos, onde as folhas representam as espécies existentes, os nós internos representam ancestrais hipotéticos e o tamanho dos caminhos ou ramos denotam a distância evolutiva.

As árvores filogenéticas podem ser construídas para quaisquer tipos de espécies, organismos ou unidades taxonômicas [2], as quais daqui em diante serão referidas como objetos. Na Figura 2.1, apresentamos um exemplo de árvore filogenética, que representa a relação entre espécies humana e primatas, onde podemos inferir que humano e chimpanzé estão mais próximos que as outras espécies.

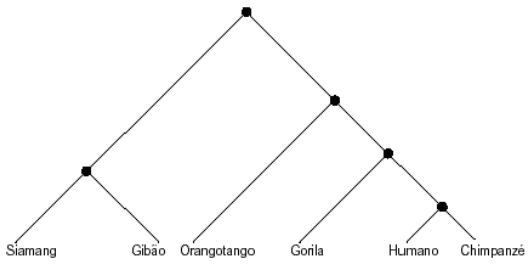


Figura 2.1: Representação de uma árvore filogenética.

[2]Táxon é um conjunto de organismos biológicos caracterizado por propriedades compartilhadas por todos os organismos.

Outra característica importante das árvores filogenéticas é se possuem raiz ou não. Numa árvore sem raiz se apresenta apenas a noção de distância sem nenhuma informação de ancestralidade. No caso das árvores com raiz, a raiz é o antepassado em comum de todos os objetos em estudo. As características gerais de uma árvore filogenética estão representadas na Figura 2.2 (Salemi e Vandamme, 2003).

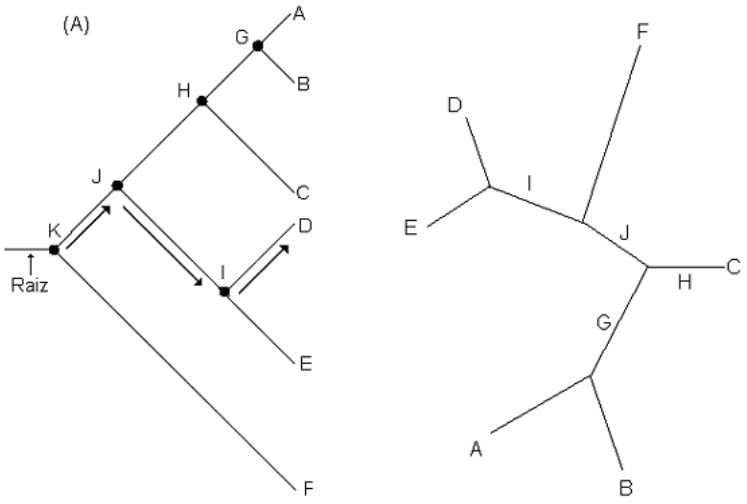


Figura 2.2: Características de uma árvore filogenética. (A) Árvore filogenética com raiz e (B) árvore filogenética sem raiz. Ambas árvores possuem a mesma topologia. Na árvore com raiz A, B, C, D, E e F são nós externos ou OTU's. G, H, I, J e K são nós internos ou ancestrais hipotéticos. As setas indicam direção de evolução. A árvore sem raiz carece de nó raiz, as linhas entre os nós são os ramos e não se conhece direção de evolução (Salemi e Vandamme, 2003).

O problema da filogenia consiste em encontrar a árvore filogenética que melhor descreva as relações de evolução entre um conjunto de objetos, e de acordo com Setubal *et al*. (1997) é definido para dois tipos de dados de entrada: distâncias ou características. Os dados de distância são obtidos da estimativa de distância evolutiva entre objetos. Essas distâncias são agrupadas em uma matriz triangular simétrica, chamada de matriz de distâncias, em que as linhas e as colunas representam objetos. Para o segundo tipo, os dados se referem à ausência ou presença de alguma característica particular nos objetos, como por exemplo, a forma do bico, número de patas, entre outras. A matriz resultante é chamada de matriz de estados em que cada coluna representa uma característica e cada linha representa um objeto.

A partir da matriz de distâncias pode-se construir dois tipos de árvores filogenéticas: árvores ultramétricas e árvores aditivas. As árvores de distâncias ultramétricas são árvores com raiz nas quais o comprimento desde a raiz até cada uma das folhas é o mesmo. A Figura 2.3(a), ilustra que as distâncias entre a raiz e as folhas A, B e C são 1+1+1, 1+1+1 e 1+2 respectivamente.

Os métodos que utilizam esse tipo de árvore são: UPGMA (*unweighted-pair group method with arithmetric means*) e WPGMA (*weighted-pair group method with arithmetric means*). As árvores de distâncias aditivas são árvores sem raiz tal que a distância entre um par de objetos é a soma dos comprimentos dos ramos que os unem. Na Figura 2.3(b), a distância entre A e B é a+b, entre A e C é a+d+c e entre B e C é b+d+c.

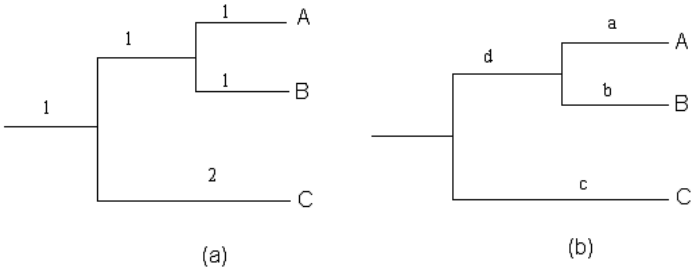


Figura 2.3: Exemplos de tipos de árvores baseadas em distância (a) árvores ultramétricas e (b) árvores aditivas.

Para que uma árvore possua distâncias aditivas, a matriz de distâncias deve obedecer às propriedades de espaço métrico[3] e satisfazer a propriedade da "condição dos quatro pontos métricos" (Salemi e Vandamme, 2003) para quaisquer quatro nós A, B, C, e D:

$$d_{AB} + d_{CD} \neq max(d_{AC} + d_{BD}, d_{AD} + d_{BC})$$

Essa propriedade nos diz que a soma dos comprimentos dos ramos que une dois nós terminais deve ser igual ou menor à distância entre todos eles. Por exemplo, na Figura 2.4, $d_{AB}+d_{CD}$ deve ser sempre igual ou menor que a mínima distância entre $d_{AC} + d_{BD}$ e $d_{AD} + d_{BC}$.

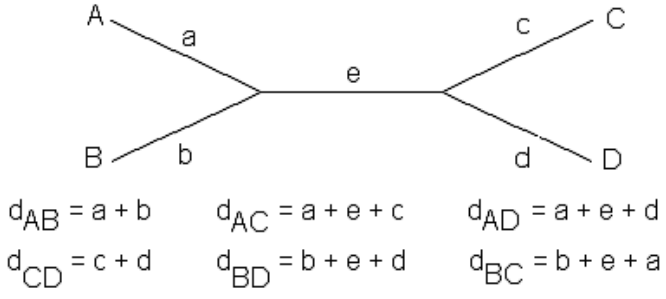


Figura 2.4: Condição dos quatro pontos métricos. As letras em minúscula representam o comprimento dos ramos (Salemi e Vandamme, 2003).

[3]Um espaço métrico é um conjunto de objetos tal que para todo par de objetos i, j associamos um número real não negativo d_{ij} com as seguintes propriedades: 1) $d_{ij} > 0$ para $i \neq j$ 2) $d_{ij} = 0$ para $i = j$ 3) $d_{ij} > d_{ji}$ para todo i e j 4) $d_{ij} = d_{jk} + d_{kj}$ para todo i, j, k (desigualdade triangular.)

As heurísticas para construir árvores aditivas geralmente empregam tempo polinomial, no pior caso a complexidade computacional é $O(n^2)$. Para o caso em que as distâncias não são aditivas, como ocorre geralmente com matrizes de distâncias usadas na prática, tentar construir uma árvore que seja a mais próxima possível a uma árvore aditiva é um problema NP-difícil.

Uma das heurísticas mais representativas para a construção de árvores filogenéticas aditivas baseadas em distância é o algoritmo *neighbor-joining*.

Uma vez expostos os conceitos necessários para entender filogenia, nas próximas seções focalizaremos nosso estudo no algoritmo *neighbor joining*, que se destaca por construir árvores filogenéticas a partir de matrizes de distâncias em tempo polinomial, além de ser um dos métodos mais usados para encontrar a verdadeira topologia da árvore ou pelo menos a mais próxima.

2.3.1 Técnica *Neighbor Joining* (NJ)

O método *neighbor joining* foi proposto por Saitou *et al.* (1987). O NJ constrói uma árvore sem raiz a partir de uma matriz de distâncias evolutivas, adaptando o critério de evolução mínima [4]. A idéia central da técnica é identificar pares de objetos mais próximos. Esses pares de objetos, conhecidos comumente como vizinhos, são conectados por um nó interno em uma árvore bifurcada. A relação de vizinhança é ilustrada na Figura 2.5, em que os nós A e B são vizinhos, mas os nós A e C não são. Se A e B forem combinados em um único nó, então nessa nova combinação A e C se tornariam vizinhos.

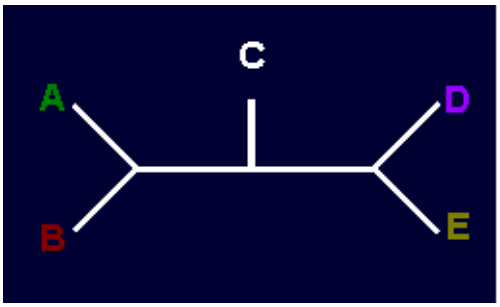


Figura 2.5: Árvore sem raiz para a técnica *neighbor joining*.

A técnica NJ começa com uma árvore estrela, como mostrado na Figura 2.6 (a), porque não há ainda nenhum par de objetos agrupados. Em seguida, escolhe-se o primeiro par de objetos para serem unidos, aplicando o critério da evolução mínima. Como podemos observar na Figura 2.6 (b), qualquer par de objetos poderia ocupar as posições 1 e 2 da árvore, então existem $\frac{n(n-1)}{2}$

[4] O critério de evolução mínima tenta minimizar a soma dos tamanhos de todos os nós da árvore.

formas de escolhê-los, onde *n* representa o número de objetos. Entre esses possíveis pares de objetos escolhemos aqueles que tenham o menor valor no cálculo da soma das distâncias entre os ramos. Para isso, aplicamos a seguinte fórmula:

$$S_{ij} = \frac{1}{2(n-2)} \sum_{k \neq i,j}^{n} (D_{ik} + D_{jk}) + \frac{1}{2} D_{ij} + \frac{1}{n-2} \sum_{(k,l \neq i,j) \wedge (k<l)}^{n} D_{kl}$$

onde D_{ij} é o valor respectivo na matriz de distâncias, *k* representa todos os objetos excepto *i* e *j* e *n* representa o número de objetos.

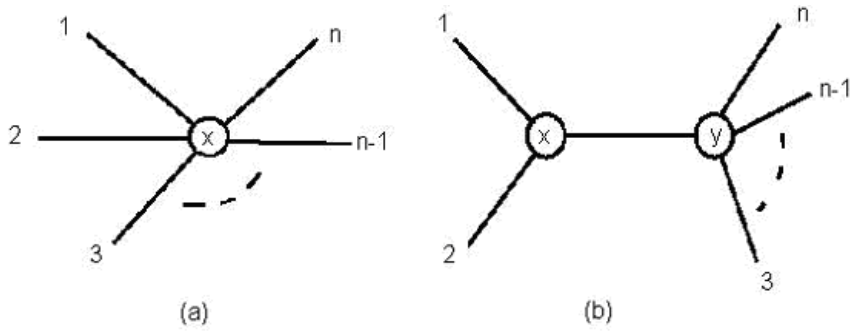


Figura 2.6: (a) Árvore estrela e (b) Árvore onde objetos 1 e 2 (i e j) estão agrupados.

Uma vez escolhido o par de objetos vizinhos a serem agrupados em um novo nó *X*, calcula-se os tamanhos dos novos ramos pelo método de *Fitch-Margoliash*, cujas equações são as seguintes:

$$L_{iX} = \frac{D_{ij} + D_{iz} - D_{jz}}{2}$$
$$L_{jX} = \frac{D_{ij} + D_{jz} - D_{iz}}{2}$$

onde *z* representa o grupo dos objetos da árvore, com exceção dos objetos *i* e *j*. As distâncias D_{iz} e D_{jz} são determinadas pelas distâncias dos objetos *i* e *j* a todos os demais objetos, na forma:

$$D_{iz} = \frac{\sum_{(k \neq j)}^{n} D_{ik}}{n-2}$$
$$D_{jz} = \frac{\sum_{(k \neq i)}^{n} D_{jk}}{n-2}$$

Como os objetos *i* e *j*, no caso da Figura 2.6 (b) 1 e 2, são o par de objetos mais próximos, ou seja, vizinhos, eles irão gerar um novo objeto e serão retirados da matriz para serem substituídos por esse novo objeto. O passo seguinte é calcular a distância entre o novo objeto e os outros objetos aplicando a seguinte fórmula:

$$D_{i-j,k} = \frac{D_{ik}+D_{jk}}{2}$$

onde $k \leq n$ excluindo *i* e *j*.

O número de objetos é reduzido de um em um e o procedimento é novamente aplicado para encontrar novos vizinhos. O ciclo se repete até que o número de objetos se torne igual a três. A complexidade computacional do NJ é $O(n^3)$. Se a árvore é aditiva, o método dá o tamanho correto para todas as arestas, conforme é mostrado por Saitou e Nei (1987).

A seguir, apresentamos o algoritmo NJ proposto por Saitou e Nei (1987) e modificado por Studier e Keppler (1988), detalhado em Salemi e Vandamme (2003).

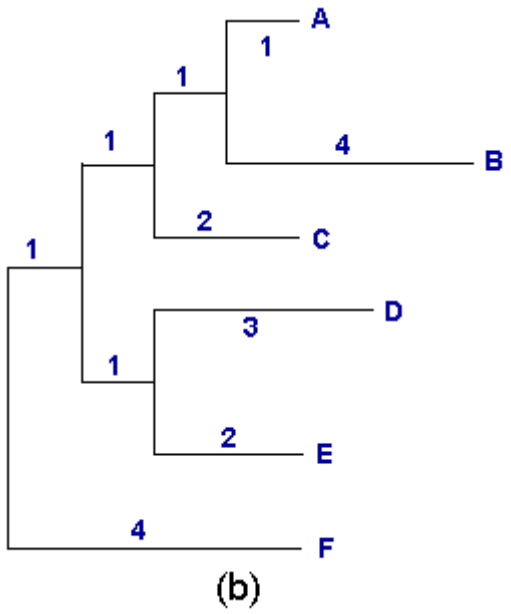
2.3.2 Algoritmo e exemplo de aplicação

O algoritmo *neighbor joining* recebe como dados de entrada uma matriz de distâncias D $(n \times n)$ de números reais não-negativos, onde a distância entre pares de objetos *i* e *j* é denotada por D_{ij}, tais distâncias provêm de comparações entre objetos. A saída é uma única árvore que representa a filogenia dos objetos. Os passos realizados pelo algoritmo são resumidos no Algoritmo 1.

Para ilustrar o algoritmo, apresentamos a seguir um exemplo da construção de árvores filogenéticas usando NJ para um número de objetos N=6, a partir da matriz de distâncias mostrada na Figura 2.7 (a). Na Figura 2.7 (b) é mostrada a correta topologia da árvore de acordo com a matriz de distâncias (Salemi e Vandamme, 2003).

	A	B	C	D	E
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8

(a)



(b)

Figura 2.7: Matriz de distâncias inicial (a) e Topologia da árvore (b).

Algoritmo 1

Inicialização

Criar una árvore com *n* objetos ligados a um nó central (árvore estrela).

Iteração

1. Para cada objeto calcular a divergência da árvore,

   $r_{(i)} = D_{i1} + D_{i2} + D_{i3} + ... + D_{ij}$

   onde:
   i é qualquer objeto e $j \neq i \ \wedge j = 1...n$
2. Calcular a nova matriz de distâncias, com a equação:

   $M_{ij} = D_{ij} - \frac{[r_i + r_j]}{n-2}$

   onde:
   M_{ij} = distância nova entre os objetos *i* e *j*.
   D_{ij} = distância atual entre *i* e *j*.
   r_i = divergência do objeto *i*.
   r_j = divergência do objeto *j*.
3. Escolher os objetos *i* e *j* para os quais M_{ij} é mínimo e criar um novo objeto *U* (nó interno na árvore) que una os objetos *i* e *j*.
4. Calcular o tamanho do ramo que una o objeto *U* aos objetos *i* e *j*.

   $S_{iU} = \frac{D_{ij}}{2} + \frac{[r_i - r_j]}{2[n-2]}$
   $S_{jU} = \mathbf{D}_{ij} - S_{iU}$

5. Calcular as distâncias entre o novo objeto *U* com os objetos restantes.

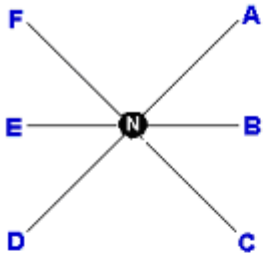
   $D_{kU} = \frac{D_{ik} + D_{jk} - D_{ij}}{2}$

   onde:
   $k \neq i \wedge k \neq j \wedge j = 1...n$
6. $n = n - 1$
7. Volte ao passo um da iteração enquanto $n > 2$.

Algorithm 1: Algoritmo *Neighbor Joining* (Salemi e Vandamme, 2003) .

Os passos detalhados que o algoritmo NJ segue para o primeiro ciclo são mostrados nas Figuras: 2.8, 2.9, 2.10 e 2.11

Inicialização



Passo 1: Cálculo de divergência da árvore

r(A) = 5 + 4 + 7 + 6 + 8 = 30
r(B) = 5 + 7 + 10 + 9 + 11 = 42
r(C) = 4 + 7 + 7 + 6 + 8 = 32
r(D) = 7 + 10 + 7 + 5 + 9 = 38
r(E) = 6 + 9 + 6 + 5 + 8 = 34
r(F) = 8 + 11 + 8 + 9 + 8 = 44

Passo 2: Cálculo da nova matriz de distâncias

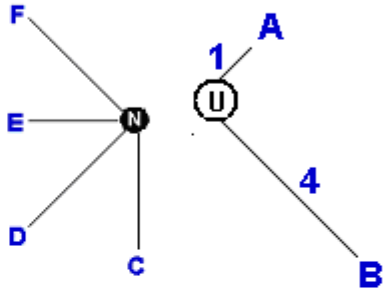
M_{AB} = 5 - [30 + 42] / (6-2) = -13
M_{AC} = 4 - [30 + 32] / 4 = -11.5
M_{AD} = 7 - [30 + 38] / 4 = -10
M_{AE} = 6 - [30 + 34] / 4 = -10
M_{AF} = 8 - [30 + 34] / 4 = -10.5
M_{BC} = 7 - [42 + 32] / 4 = -11.5
M_{BD} = 10 - [42 + 38] / 4 = -10
M_{BE} = 9 - [42 + 34] / 4 = -10
M_{BF} = 11 - [42 + 34] / 4 = -10.5
M_{CD} = 7 - [32 + 38] / 4 = -10.5
...

	A	B	C	D	E
B	-13				
C	-11.5	-11.5			
D	-10	-10	-10.5		
E	-10	-10	-10.5	-13	
F	-10.5	-10.5	-11	-11.5	-11.5

Passo 3: Escolher o par de objetos com menor valor, no caso, selecionamos os nós A e B unidos no novo objeto U

Passo 4: Calcular o tamanho dos ramos AU e BU

S(AU) = (5/2) + [30 - 42]/(2 (6-2)) = 1
S(BU) = 5 - 1 = 4



Passo 5: Calcular as distâncias entre o novo objeto e os restantes.

DCU = [4 + 7 - 5] / 2 = 3
DDU = [7 + 10 - 5] / 2 = 6
DEU = [6 + 9 - 5] / 2 = 5
DFU = [8 + 11 - 5] / 2 = 7

	U	C	D	E
C	3			
D	6	7		
E	5	6	5	
F	7	8	9	8

Passo 6: N = N - 1
6 - 1 = 5

Se inicia o processo de novo

Figura 2.8: Passos do algoritmo *Neighbor Joining*: Primeira iteração.

No segundo ciclo, o par de objetos a serem unidos pelo novo objeto R são D e E, tal como apresentado na Figura 2.9.

Passo 1: Cálculo de divergência da árvore

r(U) = 3 + 6 + 5 + 7 = 21
r(C) = 7 + 6 + 8 + 3 = 24
r(D) = 5 + 9 + 6 + 7 = 27
r(E) = 8 + 5 + 6 + 5 = 24
r(F) = 7 + 8 + 9 + 8 = 32

Passo 2: Cálculo da nova matriz de distâncias

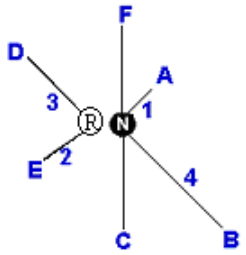
M_{UC} = 3 - [21 + 24] / 3 = -12
M_{UD} = 6 - [21 + 27] / 3 = -10
M_{UE} = 5 - [21 + 24] / 3 = -10
M_{UF} = 7 - [21 + 32] / 3 = -10.7
M_{CD} = 7 - [24 + 27] / 3 = -10
M_{CE} = 6 - [24 + 24] / 3 = -10
M_{CF} = 8 - [24 + 32] / 3 = -10.7
• • •

	U	C	D	E
C	-12			
D	-10	-10		
E	-10	-10	-12	
F	-11	-11	-11	-11

Passo 3: Escolher o par de objetos com menor valor, selecionamos os nós D e E unidos no novo objeto

Passo 4: Calcular o tamanho dos ramos DR e ER

S(DR) = (5/2) + [27 - 24]/(2(5-2)) = 3
S(ER) = 5 - 3 = 2



Passo 5: Calcular as distâncias entre o novo objeto e os restantes.

D_{UR} = [6 + 5 - 5] / 2 = 3
D_{CR} = [7 + 6 - 5] / 2 = 4
D_{FR} = [9 + 8 - 5] / 2 = 6

	U	C	R
C	3		
R	3	4	
F	7	8	6

Passo 6: N = N-1 = 5-1 = 4

Figura 2.9: Passos do algoritmo *Neighbor Joining*: Segunda iteração.

No terceiro ciclo, o par de objetos C e U são unidos em M (Figura 2.10).

Terceira Iteração

Passo 1: Cálculo de divergência da árvore

r(U) = 3 + 3 + 7 = 13
r(C) = 4 + 8 + 3 = 15
r(R) = 6 + 3 + 4 = 13
r(F) = 7 + 8 + 6 = 21

Passo 2: Cálculo da nova matriz de distâncias

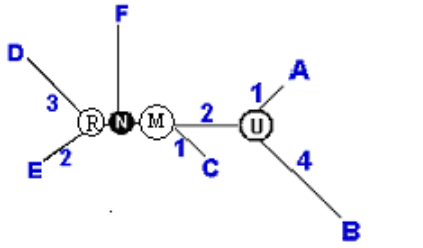
MUC = 3 - [13 + 15] /2 = -11
MUR = 3 - [13 + 13] /2 = -10
MUF = 7 - [13 + 21] /2 = -10
MCR = 4 - [15 + 13] /2 = -10
• • •

	U	C	R
C	-11		
R	-10	-10	
F	-10	-10	-11

Passo 3: Escolhermos os objetos C e U, unidos em M

Passo 4: Calcular o tamanho dos ramos UM e CM

S(UM) = (3/2) + [15 - 13] / (2(4-2))= 2
S(CM) = 3 -2 = 1



Passo 5: Calcular as distâncias entre o novo obj e os restantes

DRM = [3 + 4 - 3] / 2 = 2
DFM = [7 + 8 - 3] / 2 = 6

	M	R
R	2	
F	6	6

Passo 6: N = 4-1 = 3

Figura 2.10: Passos do algoritmo *Neighbor Joining*: Terceira iteração.

No último ciclo os objetos R e M são unidos pelo objeto T, como se apresenta na Figura 2.11. E finalmente, o último objeto F é unido ao objeto T. A topologia final da árvore do exemplo é mostrada no passo 4.

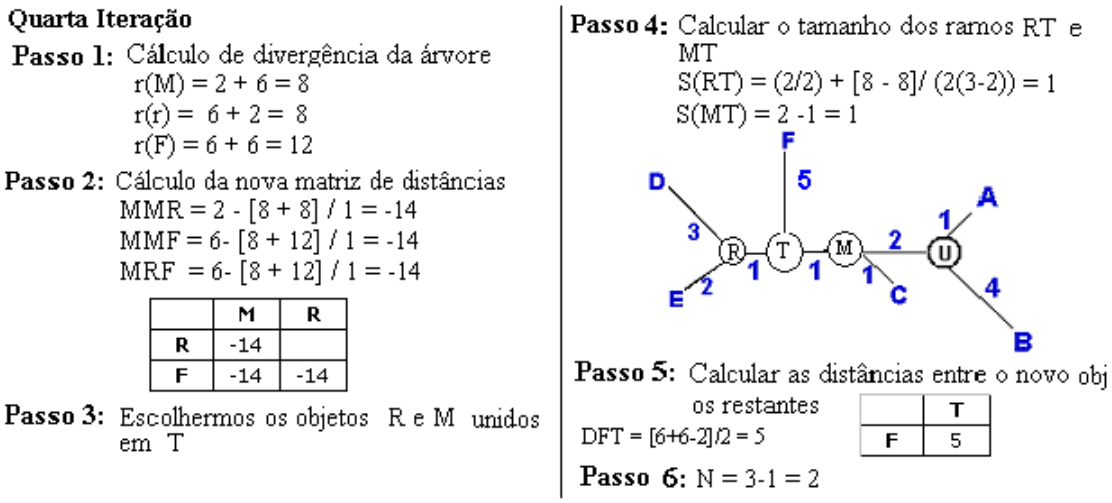


Figura 2.11: Passos do algoritmo *Neighbor Joining*: Quarta iteração.

2.3.3 Discussão e trabalhos correlatos

O método NJ é amplamente usado na construção de filogenias. Existem vários trabalhos de pesquisa que usaram o algoritmo NJ para ser comparado, segundo vários critérios, com outros métodos de reconstrução de árvores filogenéticas. Como alguns exemplos podemos citar Tateno *et al.* (1994) que pesquisaram a eficiência de alguns métodos de filogenia e o NJ considerando a topologia produzida pela árvore e a estimativa do tamanho dos ramos. Em Nakhleh *et al.* (2002) estudaram a precisão, taxa de convergência e velocidade de métodos para a reconstrução de árvores filogenéticas, dentre eles o NJ. Os resultados mostraram que o NJ recupera árvores apropriadas e sua velocidade foi melhor que todos os outros métodos avaliados.

Em Mailund *et al.* (2006), foi apresentado um algoritmo que melhora a velocidade do método NJ com o objetivo de ser aplicado a grandes conjuntos de dados. O algoritmo constrói a mesma árvore filogenética que o NJ, mas tenta diminuir o tempo de busca para encontrar o par de vizinhos que formarão um novo nó usando a estrutura *Quad-tree*. O algoritmo somente consegue melhorar o tempo de execução para o melhor caso $((O(n^2))$, ficando para o pior caso na ordem $O(n^3)$. Outro trabalho destinado a melhorar a velocidade do NJ foi realizado por Evans *et al.* (2006), aqui o tempo de processamento é $O(n^2lgn)$ conseguindo gerar árvores da mesma qualidade das geradas pelo algoritmo NJ e sem usar algum espaço adicional para realizar os cálculos.

A maior vantagem do método *neighbor joining* com respeito a outros métodos é que ele constrói árvores com maior eficiência. Além disso, o método é considerado muito rápido em relação a outros métodos usados na construção de árvores filogenéticas, tal como é mostrado por Saitou e Nei (1987).

Alguns autores estão preocupados com o fato do algoritmo NJ não examinar todas as possíveis topologias da árvore resultante, porque ao aplicar o critério de mínima evolução nem sempre é obtida a árvore de evolução mínima, pois minimizar o comprimento da árvore a cada passo do algoritmo não implica em minimizar o comprimento global da árvore.

2.4 Considerações finais

Os textos freqüentemente são apresentados por meio de um vetor de termos num modelo de espaço vetorial, no qual a dimensionalidade base está geralmente por volta de 10,000 atributos para um conjunto de 500 textos ou mais (Rohrer *et al.*, 1998). Devido a isso, dois grandes problemas surgiram provenientes da alta multidimensionalidade: por um lado, é preciso estimar uma função que cresça exponencialmente com o número de variáveis e por outro lado, o espaço multidimensional é inerentemente esparso. Por exemplo, um termo que aparece em um texto acima de 100 vezes pode não aparecer em qualquer dos outros textos restantes. Para superar esse problema, um método amplamente usado é a redução de dimensionalidade.

A principal idéia das técnicas de redução de dimensionalidade é posicionar um conjunto de dados num espaço de baixa dimensionalidade preservando o máximo possível as relações de similaridade existentes entre eles. Dessa forma dados multidimensionais representados em espaços de baixa dimensionalidade podem fazer uso da visualização e classificação de forma mais eficiente, revelando algumas características interessantes, como agrupamentos, tendências e anomalias.

Em contrapartida, o NJ identifica os objetos vizinhos que seqüencialmente minimizam o tamanho da árvore e é definido por três componentes: 1) o critério usado para selecionar pares de objetos, 2) a fórmula usada para reduzir a matriz de distâncias a cada passo e 3) a fórmula para obter o comprimento dos ramos. O núcleo do método é o critério de seleção do par de objetos que sofrera a união (Bryant, 2005).

O NJ, além de construir a topologia para uma árvore sem raiz, também fornece o valor do comprimento dos ramos da árvore resultante e é o método baseado em distâncias mais amplamente usado na construção de grandes árvores filogenéticas por ser rápido no contexto de filogenia e por inferir filogenias com alta exatidão em comparação com outros.

Como visto anteriormente, as técnicas de redução de dimensionalidade abordadas neste capítulo apresentam vantagens e desvantagens. A diferença que mais ressalta entre elas é o tempo de processamento e como conseguem definir as relações de similaridade inter- e intra-grupos, de tal forma que reflitam o melhor possível as relações existentes no espaço original. Por exemplo, as técnicas PCA e LSI possuem problemas de representatividade e custo computacional elevado. Na PCA somente projeções lineares são possíveis e muito da informação válida não é mostrada no plano de visualização. LSI falha quando tenta separar mais de três grupos com grande variância. MSD falha quando a densidade dos grupos é muito alta dificultando a identificação de mínimos locais. As técnicas ProjClus e LSP conseguem diminuir a complexidade computacional (poucos segundos para milhares de textos) alem de apresentar resultados satisfatórios em identificar similaridades entre dados multidimensionais. LSP obtém bons resultados quando aplicada em dados textuais (Paulovich *et al.*, 2006) e na precisão dos *layouts* gerados (Paulovich *et al.*, 2007a), portanto, a técnica LSP é considerada a técnica de redução de dimensionalidade ou de projeção representativa na aplicação dos resultados deste trabalho.

O resultado das técnicas de redução de dimensionalidade e do NJ podem ser exibidos no espaço de visualização através de representações visuais, tais como: pontos posicionados no plano, grafos, redes, superficies, e volumes. Existem diferentes técnicas de visualização, que baseadas nessas representações geram mapas interativos para permitir aos usuários explorar e navegar por informação presente nos textos. No próximo capítulo é tratado o processo realizado para representar visualmente mapas de textos com base no seu conteúdo. Também são apresentadas as principais técnicas de visualização desenvolvidas para representar árvores filogenéticas.

CAPÍTULO

3

Mapeamento visual de textos baseados em conteúdo

3.1 Considerações Iniciais

A visualização de informação é uma área dedicada a representar modelos gráficos a partir principalmente de dois tipos de dados: aqueles associados a fenômenos do mundo físico e que são inerentemente espaciais e aqueles que não necessariamente tem uma representação no mundo físico. Um exemplo para o primeiro tipo de dados é visualização de temperatura atmosférica sobre um mapa. Já para o segundo, talvez o exemplo mais representativo seja texto, cuja apresentação é feita em forma de artigos científicos, mensagens de texto, coleção de notícias, *snippet*[5] de buscas na Internet, patentes, entre outras, normalmente dados não estruturados. Como nesse caso os dados são de natureza abstrata, transformá-los em imagens apropriadas para serem compreendidas pelos usuários é um desafio. Para lidar com isso, muitas técnicas optaram por fazer uso de metáforas visuais com a finalidade de organizar, analisar e entender o conteúdo e relações existentes entre os dados, sem que o usuário tenha necessidade de acessar ou ler o conteúdo de todos os textos de uma coleção em particular.

[5]Um *snippet* é uma pequena descrição de uma página resultante de uma busca na Internet

Neste contexto é que está inserido o conceito de mapeamento de textos, como uma forma de recuperar, preservar e representar informação contida num conjunto de textos geralmente de elevada dimensionalidade, de modo que textos similares sejam localizados em áreas vizinhas no espaço de visualização e que sejam capazes de refletir as relações de similaridade, relevância e organização (Becks *et al.*, 2005), (Lopes *et al.*, 2006).

Para criar uma representação visual de mapas de textos tem-se que transformar esses textos para extrair alguma medida de similaridade, com a finalidade de conseguir representá-los em um espaço m-dimensional, com valores de m no intervalo de 1 a 3. Esses mapas de textos irão apresentar várias propriedades dos textos, incluindo similaridade por conteúdo, co-citação, co-ocorrência de termos e comparações de valor de atributos, tais como autores ou datas de publicação. Para uma visão detalhada de mapas com essas propriedades consultar (Spence, 2001), (Borner *et al.*, 2003), (Chen, 2004), e (Pinho e de Oliveira, 2006).

O texto apresentado aqui não representa um levantamento completo de mapas de documentos, mas sim uma amostragem dos tipos de representação visual adotados para conjuntos de textos baseados em conteúdo. O processo para construir tais mapas é descrito na Seção 3.2, e as técnicas e as principais ferramentas que surgiram com o intuito de conseguir uma apropriada representação gráfica são apresentadas na Seção 3.3.

Na Seção 3.4 é feita uma análise do estado atual de técnicas de visualização de dados usando árvores filogenéticas, que apoiam seu desenvolvimento em técnicas de visualização de informação.

3.2 Processo de visualização de mapas de textos baseados em conteúdo

Embora textos sejam entre outras coisas a mais rica e maior fonte de informação consumida, produzida e analisada por seres humanos (Rohrer *et al.*, 1998), os usuários que lidam com esse tipo de informação provavelmente não sabem apriori o conteúdo imerso nos textos para procurar por informação relevante. Para tratar esse problema, mapas de textos fornecem ao usuário uma visão global acerca do dominio de conhecimento presente em uma coleção de textos e uma visão local em vários níveis de detalhe, apoiando-lo na formulação de consultas mais precisas.

Para realizar esse processo de análise, mapas de textos baseados em conteúdo organizam textos com base alguma medida de similaridade. Essa medida deve ser capaz de agrupar ou separar textos segundo algum grau de correlação, permitindo aproximar numa vizinhança textos de alta correlação ou similaridade e separar visualmente grupos de textos altamente correlacionados.

Segundo Spence *et al.* (2001) o processo geral para analisar e visualizar textos é realizado em três etapas: análise, algoritmos e visualização. Na etapa de análise, típicamente se transformam os textos em uma representação vetorial. Os algoritmos trabalham com esses textos transformados para serem representados em espaços 2D ou 3D [6] e a visualização é responsável por apresentar informações adicionais sobre os dados, como ilustrado na Figura 3.1

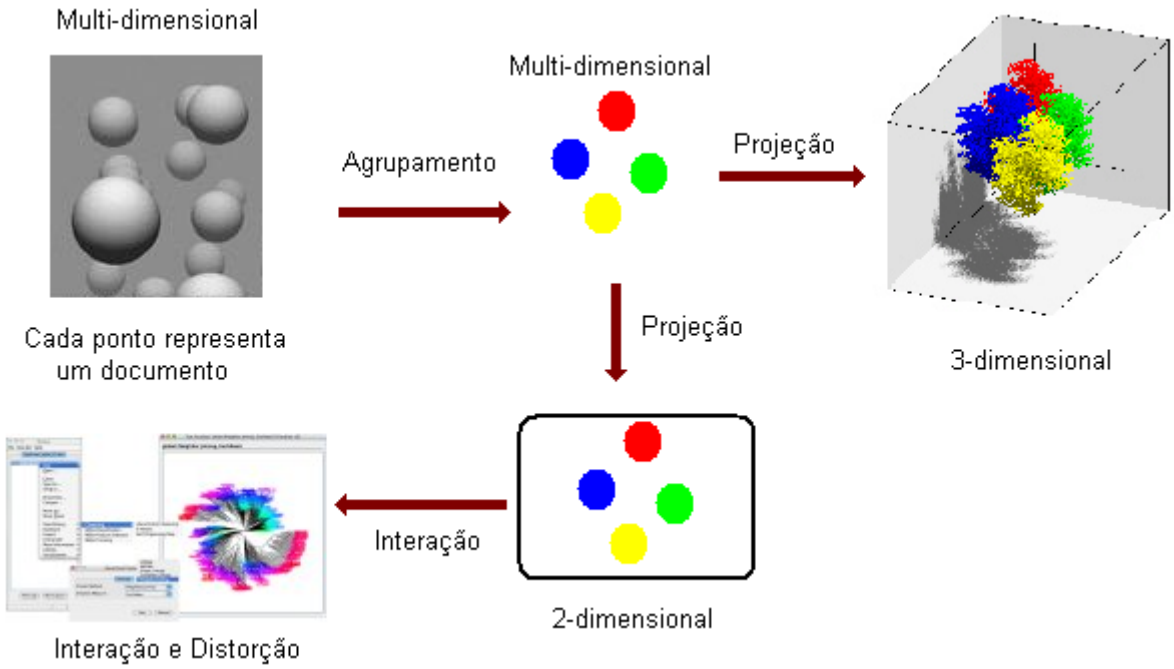


Figura 3.1: Visualização de textos incluíndo agrupamento e projeção (Spence, 2001).

Esse processo foi descrito sumariamente no Capítulo 1 e é detalhado aqui. Para explicar cada etapa, nosso estudo se baseou nos trabalhos desenvolvidos por Paulovich *et al.* (2006) e Minghim *et al.* (2007), que serviram como ponto de referência teórico. Neles, o processo para construir e explorar mapas de textos baseados em conteúdo é realizado em duas etapas: 1) criação do mapa e 2) exploração do mapa.

A etapa de ***criação do mapa*** é dividida em pré-processamento e projeção e arranjo da coleção de textos. Na etapa de pré-processamento, textos são transformados para ser comparados segundo alguma medida de similaridade. Existem duas formas para realizar esse cálculo: uma é aplicando o modelo de espaço vetorial e a outra é aplicando uma medida de texto contra texto como a complexidade de Kolmogorov.

No caso do *modelo de espaço vetorial* (Salton *et al.*, 1975), cada texto de uma coleção de textos é representado por um vetor de termos (palavras). Cada termo, que pode ser uma única

[6] Quando se trabalha com dados complexos tanto com múltiplas dimensões quanto com grande número de elementos é recomendável aplicar algoritmos de agrupamento e projeção nos dados (Spence, 2001).

palavra ou alguma expressão composta de várias palavras, possui um valor associado (peso) que indica a presença ou freqüência de cada termo encontrado no texto. Cada elemento desse vetor é considerado uma coordenada dimensional e geometricamente é modelado em um espaço Euclideano de *n* dimensões (*n* é o número de elementos).

Para que o texto seja representado em sua forma vetorial, uma seqüência de passos deve ser realizada e é dividida em duas fases. Os passos da primeira fase são (Weiss *et al.*, 2004):

- Remoção de termos não representativos, como artigos, preposições, pronomes, conjunções e outras. Esses termos são chamados de *stopwords*.
- Redução dos termos a seus radicais (*stemming*). Nesse passo são identificados o radical dos termos, removendo sufixos e prefixos. O algoritmo de Porter (Porter, 1980) geralmente é aplicado para realizar essa tarefa em textos em inglês. Assim, termos com o mesmo radical são identificados como similares.
- Contagem das palavras dentro de cada documento para determinar suas freqüências de ocorrência.
- Aplicação de cortes superiores e inferiores de Luhn (*Luhn's cut*) (Luhn, 1968) para eliminar palavras muito freqüêntes ou muito raras.
- Atribuição de um valor (peso) de acordo com a importância de cada termo no texto. Um modelo bastante usado é TF x IDF *(Term Frequency - Inverse Document Frequency)*. Nesse método, a importância dos termos é proporcional à freqüência de ocorrência dos mesmos em cada texto da coleção e inversamente proporcional ao número de textos em que os termos aparecem. Para calcular o peso w_i do termo *i* em um vetor, faz-se o cálculo do produto da freqüência (*freq*) do termo t_i nesse texto d_j pela quantidade de textos onde o termo t_i ocorre pelo menos uma vez (*dfreq*), como mostrado na seguinte equação:

$$w_i = freq(t_i, d_j) \times log(\frac{n}{dfreq(t_i)})$$

Nesse último passo é criada uma matriz de textos x termos, onde cada linha representa um texto e cada coluna representa o valor do peso do termo presente no texto.

Uma vez definida a matriz de textos x termos, a fase seguinte é realizar o cálculo da distância entre um texto e outro. Essa distância indica o grau de similaridade, ou seja, textos que possuem as mesmas palavras acabam sendo colocados em uma mesma região do mapa de visualização. O cálculo da similaridade entre dois textos pode ser feito de várias maneiras. As medidas mais populares são a distância Euclideana, Manhattan e cosseno do ângulo entre os vetores. Essa

última determinada o cosseno dos vetores $\vec{u}$ e $\vec{v}$ aplicando o produto escalar dos vetores entre seus módulos:

$$cos(\vec{u}, \vec{v}) = \frac{\sum_{i=1}^{n}(u_i \times v_i)}{\sqrt{\sum_{i=1}^{n} u_i^2} \times \sqrt{\sum_{i=1}^{n} v_i^2}}$$

A complexidade de Kolmogorov aplica uma medida de similaridade de texto contra texto (Telles *et al.*, 2007). A definição da similaridade entre dois textos é calculada de forma direta sem realizar a representação vetorial citada anteriormente, aplicando operações simples sobre o tamanho compactado dos textos. Essa técnica usa como medida de similaridade *Normalized Compression Distance* (NCD) que inicialmente foi concebida como uma medida entre seqüências de DNA. NCD calcula a similaridade entre dois textos a partir do conteúdo dos mesmos. A complexidade de Kolmogorov de uma cadeia de caracteres *x*, denotada por *K(x)*, é o tamanho do menor programa *a* que imprime *x* e pára e pode ser interpretada como a quantidade de informação presente na cadeia. A complexidade de Kolmogorov condicional *K(x/y)* de *x* com relação a outra cadeia *y* é o tamanho do menor programa que imprime *x* quando recebe *y* como entrada e pára e pode ser interpretado como a quantidade de informação em *x* que não está presente em *y*. A complexidade de Kolmogorov não é computável, mas pode ser aproximada usando compressão. No trabalho de Telles *et al.* (2007), a formulação do NCD é dada por:

$$NCD(x, y) = \frac{C(xy) - min\{C(x), C(y)\}}{max\{C(x), C(y)\}}$$

onde C(.) é o tamanho de *x* comprimida por um algoritmo de compressão, e *xy* é a concatenação de x *e y*.

A complexidade de Kolmogorov obteve bons resultados em textos ao mostrar um bom agrupamento de textos similares e alto grau de separação por conteúdo. A principal desvantagem da técnica é o elevado custo computacional, embora possua as vantagens, sobre a representação vetorial de não precisar de parâmetros ou pré-processamento.

Os dados gerados em etapas anteriores dão como resultado relações de similaridade. Na fase de ***projeção e arranjo*** os dados são posicionados em espaços de duas dimensões. Nessa fase também se pode fazer uso de técnicas de análise de agrupamento (*cluster analysis*) para aproximar mais os textos similares e definir as fronteiras entre grupos diferentes.

Na literatura existem diversas técnicas de redução de dimensionalidade ou projeção, cada uma apresenta vantagens e desvantagens, as que foram descritas detalhadamente no Capítulo 2.

A etapa de exploração do mapa é composta de **visualização e interação** na qual o conjunto de textos é visualizado em posições espaciais de 2 ou 3 dimensões, mostrando as relações entre os textos através de atributos visuais. Segundo Borner *et al.* (2003), uma boa visualização deve:

- Permitir a compreensão de grande quantidade de dados quando aplicada uma visão geral, assim como uma visão detalhada.
- Reduzir o tempo de busca visual.
- Prover um melhor entendimento de conjuntos de dados complexos.
- Revelar relações não identificáveis de outra forma.
- Permitir a visão do conjunto de dados sob diversas perspectivas simultaneamente.
- Facilitar a formulação de hipóteses.
- Servir como forma de comunicação.

Para interagir com a visualização de mapas de textos, Chen *et al*. (2004) sugere que a navegação em um espaço de informação visual deve passar pela construção de um mapa cognitivo interior, sem que a complexidade do espaço de informação, representado e disponibilizado por meio do computador, torne-se um obstáculo. Para lidar com esse desafio, surgiram varias técnicas que pretendem que a navegação no espaço de informação se torne familiar ao usuário para conseguir uma adequada manipulação e localizar informação de interesse. Segundo Keim *et al*. (2001) quatro tarefas fundamentais facilitam a manipulação de um mapa iterativo: 1) visão global (*overview*), 2) visualização de uma região de interesse (*zoom*), 3) filtro, para permitir ao usuário visualizar ou não temas de interesse e 4) informação adicional deveria estar disponível (*Detais-on-demand*).

A seguir serão apresentadas as principais ferramentas e técnicas de visualização de mapas de textos.

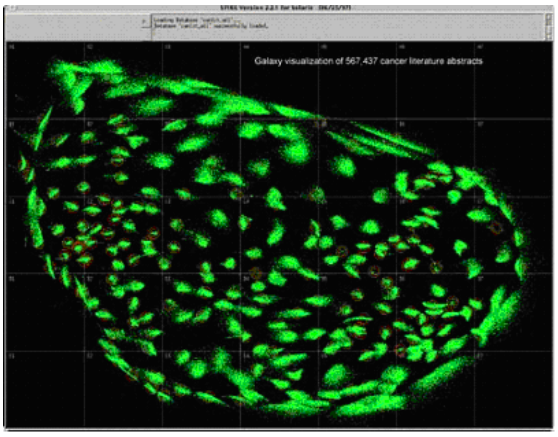
3.3 Técnicas de visualização de mapas de textos

Como mencionado na seção anterior, o processo de visualização de mapas de textos foi desenvolvido com o objetivo de mapear grandes conjuntos de textos e visualizar a representação de informação de forma global e local. Várias técnicas foram desenvolvidas (Borner *et al.*, 2003), mas nesta seção são apresentadas as principais técnicas utilizadas na visualização de textos baseadas em conteúdo.

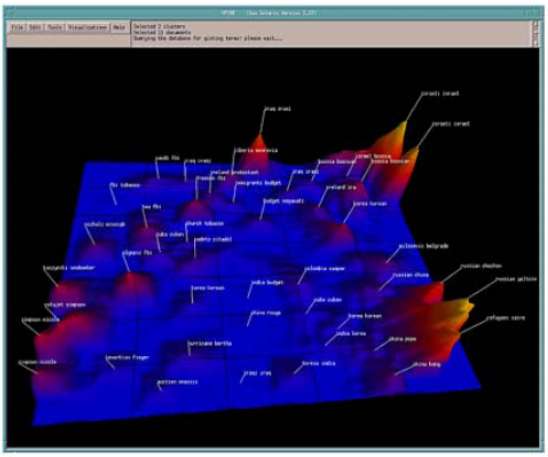
3.3.1 *IN-SPIRE*

A ferramenta *IN-SPIRE* (*Spatial Paradigm for Information Retrieval*)[7] (Wise, 1999) mostra graficamente a similaridade existente entre um conjunto de textos baseada em termos similares descobrindo automaticamente os temas chaves no texto.

Em *IN-SPIRE*, foram desenvolvidas duas metáforas de visualização para a análise de informação textual. A primeira delas, *Galaxies* (Crow *et al.*, 1994) (Figura 3.2(a)) é baseada na técnica de redução de dimensionalidade *Multidimensional Scaling* (MDS) (Seção 2.2.3) e agrupa textos similares próximos num espaço de visualização 2-dimensional em que os textos são apresentados como estrelas e os agrupamentos entre si como constelações. Essa forma de representação permite visualizar globalmente os temas e explorar a informação de interesse em diferentes níveis de granularidade.



(a) Em *Galaxies* os textos são agrupados por similitude.



(b) *Themescape* representacão com vales e montanhas baseadas na freqüência estatística de termos chaves .

Figura 3.2: Metáforas de visualização em *SPIRE* http://www.pnl.gov/infoviz.

A outra metáfora do *IN-SPIRE* é o *Themescape* descrita em (Pennock e Lantrip, 1995), onde os textos são apresentados por meio de vales e montanhas. Ela se baseia na freqüência estatística de termos chaves: quanto mais alta é a montanha um maior número de textos estão relacionados a esses termos. De outra forma, os vales representam textos com pouco conteúdo similar. A Figura 3.2(b) apresenta uma visualização gráfica dessa técnica.

[7]Ferramenta comercial: http://www.pnl.gov/infoviz.

3.3.2 *Self-Organization Maps* (SOMs)

O SOM é provavelmente o algoritmo mais popular usado no mapeamento de textos. Ele faz uso de algoritmos de redes neurais para gerar mapas auto-organizáveis de informaçao. Durante a fase de aprendizado, o algoritmo SOM modifica iterativamente os pesos de um neurônio que representa um vetor de termos de um texto, de tal forma que textos similares são alocados em posições próximas em uma malha 2-dimensional (Kohonen, 1997). Um exemplo de aplicação é apresentado em Borner *et al.* (2003) (Figura 3.3). Na superfície de visualização os textos são organizados em grupos, que são rotulados ao calcular os 3 termos com maior peso e que representam o tema principal dos textos agrupados. A altura em cada grupo representa quão organizada, focalizada e coerente é uma porção de textos pertencestes a esse grupo. As fronteiras entre grupos são desenhadas como linhas.

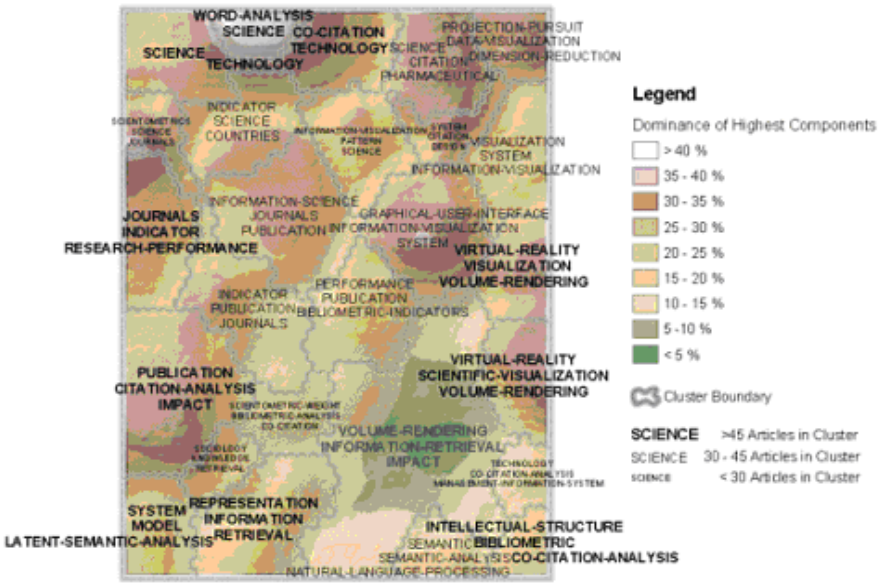


Figura 3.3: Mapa SOM (Borner *et al.*, 2003).

3.3.3 ET-Map

Roussinov e Chen *et al.* (1998) da Universidade de Arizona realizaram um projeto denominado ET-Map (*Entertainment Maps*) [8] no período de $1995 - 1998$ para analisar um conjunto de textos pertencentes à categoria de entretenimento do índice do site principal do Yahoo. ET-Map foi criado usando uma variação de mapas SOM. O espaço visual é dividido em regiões, como apresentado na Figura 3.4. Cada região possui uma cor diferente e é rotulada com um título

[8] Disponível na internet em http://ai.bpa.arizona.edu.

que resume um conjunto de páginas de conteúdo similar, assim como o número de textos que pertencem a esse grupo. O mapa possui dois níveis, isto é, se selecionamos uma região aparece outro mapa de textos pertencentes à mesma região.

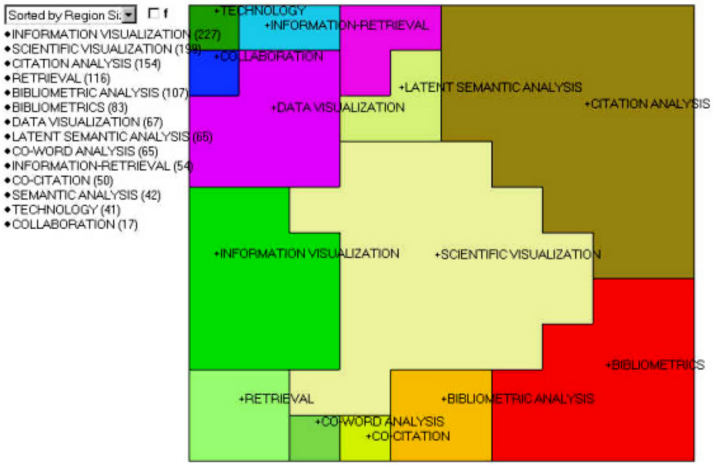


Figura 3.4: Mapa do projeto ET-Map (Borner *et al.*, 2003).

3.3.4 gCLUTO

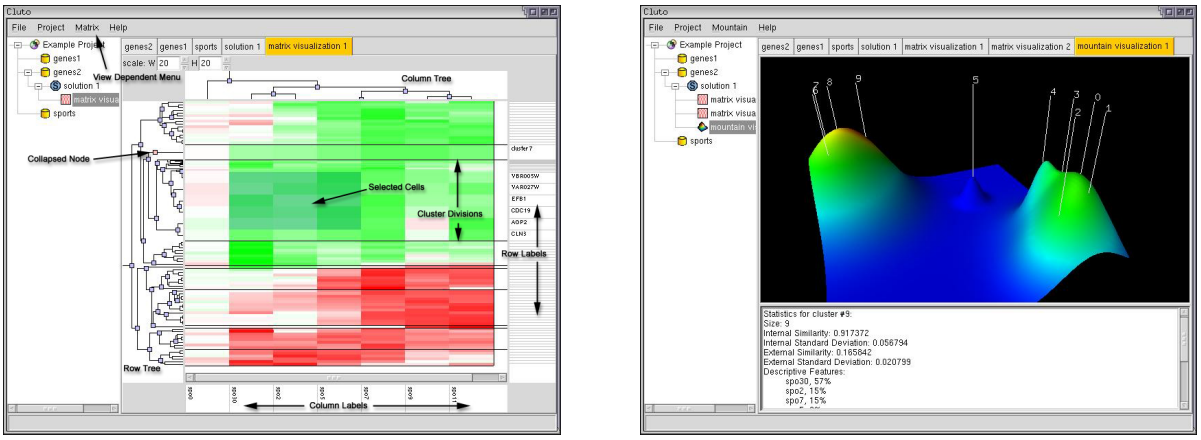
O gCLUTO (Rasmussen e Karypis, 2004)[9] é um pacote desenvolvido para agrupar conjuntos de dados de diversos domínios do conhecimento. Ele faz uso do potencial das funções de análise de dados da biblioteca CLUTO, que fornece três classes de algoritmos de agrupamento: aglomerativos, *k-means* e particionamento de grafos.

No gCLUTO existem duas formas de visualizar dados multidimensionais: visualização da matriz de dados e visualização de montanha. A visualização da matriz apresenta uma escala de cores para representar graficamente os dados originais com poucas alterações. A Figura 3.6 apresenta um exemplo de visualização de matriz. Na visualização por montanha, cada grupo representa um pico 3D. A localização do pico, o volume, a altura e as cores são usados para refletir informaçãoes sobre o grupo a que se referem, como visto na Figura 3.6.

3.3.5 InfoSky

O sistema para exploração visual Infoksy (Granitzer *et al.*, 2004) também emprega a metáfora de visão de uma galáxia, onde as estrelas representam textos e as constelações representam coleções de textos. A representação gráfica é realizada em um espaço de duas dimensões. A

[9]Disponível em www-users.cs.umn.edu/ mrasmus/gcluto.html.



(a) Visualização de matriz de dados. (b) Visualização de montanha.

Figura 3.5: Metáforas de visualização em gCLUTO (Rasmussen e Karypis, 2004).

técnica empregada para posicionar textos similares no plano é *Force-directed placement* (Seção 2.2.3) e os textos estão organizados hierarquicamente em grupos. As coleções de textos estão delimitadas por polígonos, de forma similar à representação de constelações em mapas estelares. O sistema emprega técnicas de *focus+context*[10] para a exploração interativa do mapa, isto é, regiões são detalhadas a medida em que são selecionadas pelo usuário. A Figura 3.6 apresenta o sistema Infosky para um conjunto de dados que contém aproximadamente 100,000 textos.

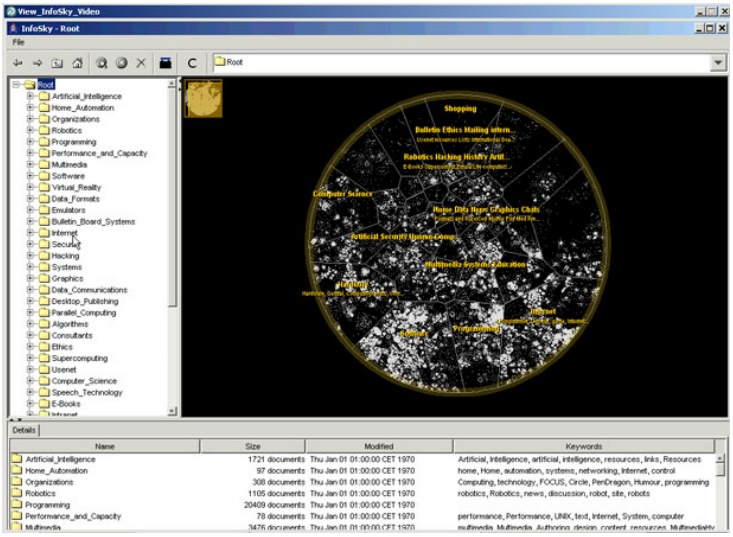


Figura 3.6: Visualização DO sistema Infosky (Granitzer *et al.*, 2004).

[10]A expressão Focus+Context se refere a técnicas que possibilitam ao usuário centralizar sua visão em uma representação visual que é exibida em detalhe (*focus*), percebendo, ainda, simultaneamente uma visão global (*context*).

3.3.6 *VxInsight*

VxInsight[11] (Boyack *et al.*, 2002) foi desenvolvido para mapear e navegar em conjunto de textos. Cada texto é representado por um vetor, usando como medida de similaridade a métrica do cosseno do ângulo. É usada a metáfora de montanha na qual a altura da montanha é proporcional ao número de objetos presentes nela, sendo cada uma rotulada para revelar o conteúdo dos textos. Na Figura 3.7 é apresentado um exemplo.

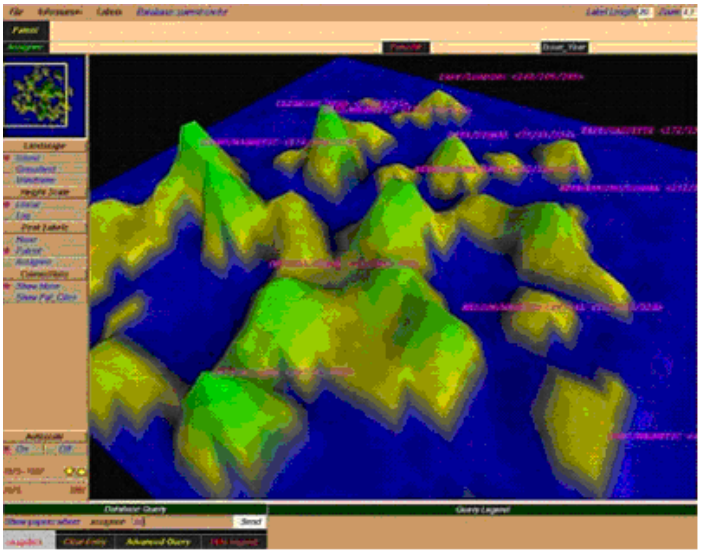


Figura 3.7: VxInsight cria visualizações utilizando a metáfora de montanha (Boyack *et al.*, 2002).

Como exemplo representativo de técnicas de visualização baseadas em projeções citaremos a seguir o *Projection Explorer*, ferramenta que serviu como base de nosso trabalho.

3.3.7 *Projection Explorer (PEx)*

O Projection Explorer [12] foi desenvolvido no contexto do projeto Infovis2 por Paulovich *et al.* (2007), pensado principalmente para criar mapas de textos. Pontos são posicionados no plano utilizando projeções multidimensionais para gerar visualizações. A maioria das técnicas apresentadas no Capítulo 2 podem ser escolhidas para posicionar os dados no plano. O resultado gráfico é um conjunto de pontos no plano que podem formar uma triangulação ou um grafo gerado pela conexão de cada ponto aos seus vizinhos mais próximos.

Sua representação visual é definida num espaço bi-dimensional, com círculos representando os objetos multidimensionais. Os objetos são ligados entre si por meio de arestas que indicam

[11] Disponível em www.cs.sandia.gov/projects/VxInsight.html.
[12] http://infoserver.lcad.icmc.usp.br/infovis2 (Proc. FAPESP 04/07866-4).

alguma relação entre os mesmos. Os círculos podem ser coloridos e ter seus tamanhos alterados com base em alguma informação extra mapeada sobre a projeção na forma de campo escalar. Por exemplo, no caso de mapas para conjuntos de textos é possível colorir os textos com base na ocorrência de um termo ou conjunto de termos ou com base na distância de um texto para todos os outros. Além disso, se os textos foram classificados de alguma forma é possível colorir o mapa de acordo com essa classificação.

Na etapa de exploração, cada ponto tem um rótulo e seu conteúdo. Além disso o PEx identifica os vizinhos mais próximos ao executar um simples clique sobre um texto. Na Figura 3.8 é apresentada a janela principal do PEx. Um estudo detalhado do PEx é apresentado na Seção 4.3.

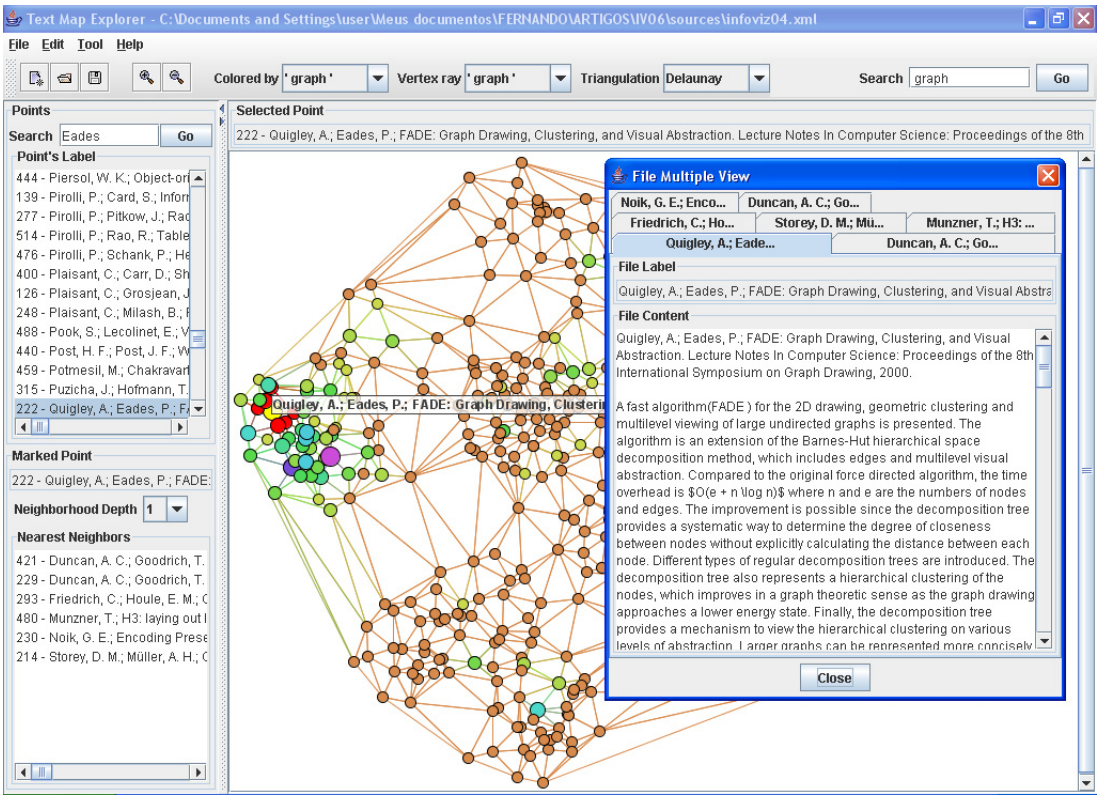


Figura 3.8: Funcionalidades da ferramenta PEx.

Existe outro grupo de técnicas desenvolvidas para visualizar o resultado de construção de árvores filogenéticas e que estão fortemente ligados a técnicas de visualização de informação para dados multidimensionais propostas aqui. Algumas delas são descritas na próxima seção.

3.4 Visualização de dados usando árvores filogenéticas

Filogenia faz uso da inspeção visual, da comparação estrutural, da manipulação e exploração de árvores filogenéticas para realizar a análise de dados. Portanto, visualizar informação faz parte das aplicações de filogenia. Enquanto vários algoritmos para inferir filogenia foram desenvolvidos com grande êxito, o mesmo não se pode dizer das técnicas de visualização para apresentar os resultados (Munzner *et al.*, 2003).

Segundo Carrizo *et al*. (2004), a visualização de filogenias enfrenta principalmente cinco grandes problemas : 1) *layout*, 2) rótulo e anotações, 3) navegação, 4) comparação de árvores 5) manipulação e edição, dos quais serão tratados os três primeiros por serem fundamentais neste trabalho.

Para o caso de *layout*, há três topologias comumente usadas: *phylogram*, radial e o *cladogram*, que são exemplificadas na Figura 3.9. Na literatura não existem critérios específicos para empregar algum deles nem se considera o fato de como esses *layouts* podem afetar o entendimento e percepção dos resultados. O rótulo e as anotações estão relacionados com a correta posição deles no *layout*. Por exemplo, visualizar os rótulos em árvores radiais é mais complexo que nos *phylograms*; o problema aumenta quando o número de nós é superior a 30 ou 40. O terceiro problema (navegar ou explorar uma árvore filogenética) refere-se ao fato de se conseguir apresentar sua estrutura global, sobretudo quando o número de nós é grande. Uma das formas mais usada para lidar com esse problema é reduzir o tamanho da árvore aplicando técnicas de agrupamento, desde que sub-árvores possam ser agrupadas com seus antepassados, e assim conseguir expandir ou comprimir a árvore.

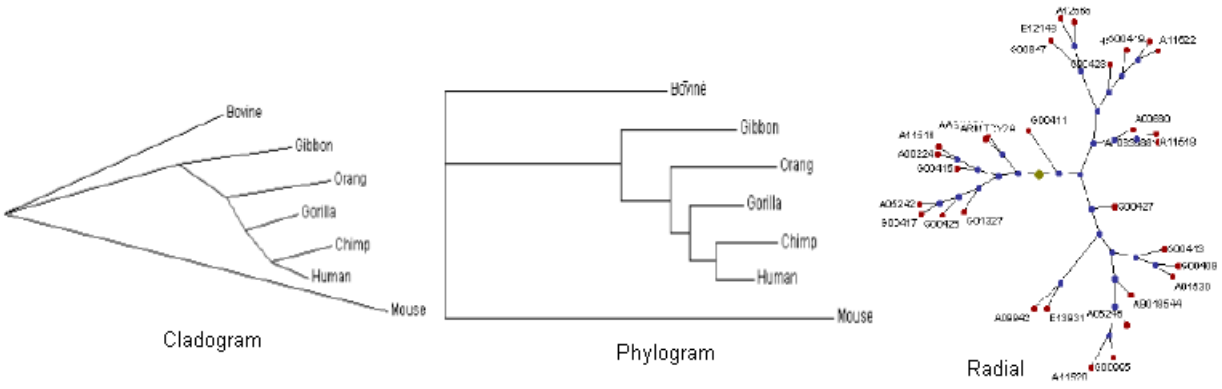


Figura 3.9: Principais *layouts* das árvore filogenética (Carrizo, 2004).

Para minimizar esses problemas surgiram vários métodos para visualizar filogenias, como ***Treeview*** (Page, 1996) e ***ATV*** (Zmasek e Eddy, 2001), ferramentas que conseguiram bom desempenho quando aplicadas á visualição de árvores pequenas em espaços bidimensionais. Porém, quando o número de nós aumenta, não é possível visualizar toda a árvore sem que aconteçam sobreposições e diminuição da legibilidade. Para contornar esses problemas surgiram ferramentas para visualizar grandes árvores. Algumas dessas ferramentas são a ***Hypertree*** (Bingham e Sudarsanam, 2000), que permite visualizar grandes árvores filogenéticas em espaços hiperbólicos bidimensionais usando *layout* radial, como mostra a Figura 3.10. É uma ferramenta implementada em linguagem Java. Além disso, a ***Hypertree*** emprega técnicas de visualização como *focus+contex* para representar sub-conjuntos de dados.

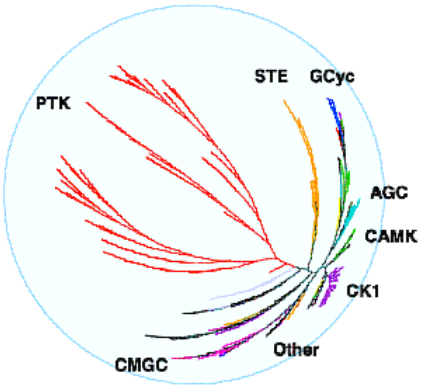


Figura 3.10: *Hypertree* (Bingham e Sudarsanam, 2000).

Outra técnica denominada ***PhyloDraw*** (Choi *et al.*, 2000) permite visualizar vários tipos de *layout* como *retangular cladograms*, *phylograms*, árvores sem raiz e árvores radiais, além de construir árvores filogenéticas a partir de uma matriz de distâncias. *PhyloDraw* é implementada em Visual C++ 6.0. Um exemplo é mostrado na Figura 3.11.

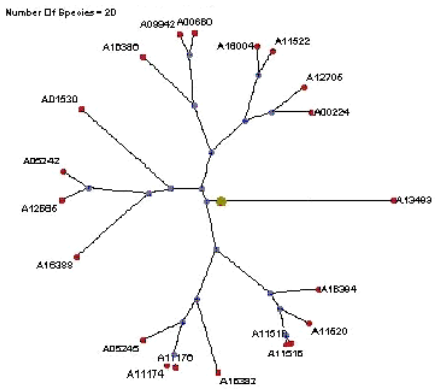


Figura 3.11: *PhyloDraw* com *layout* radial (Choi *et al.*, 2000).

TreeWiz (Rost e Bornberg-Bauer, 2002), foi projetada para visualizar e explorar o conteúdo de grandes árvores usando múltiplas visões para navegar nas sub-árvores, além de proporcionar uma visão geral e ampliação e filtragem da árvore para recuperar informação. Em *TreeWiz* é possível analisar a relação entre as folhas e explorar suas características em um segundo nível de representação aplicando agrupamento hierárquico, como ilustra a Figura 3.12. A ferramenta foi desenvolvida em Java e o código é de acesso livre.

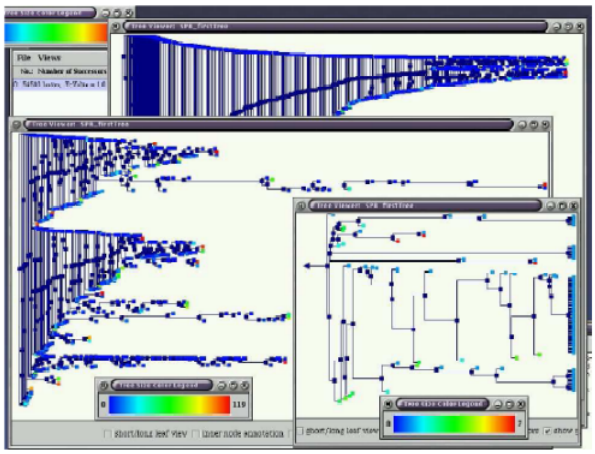


Figura 3.12: *TreeWiz* (Rost e Bornberg-Bauer, 2002).

Walrus[13] é um visualizador desenvolvido em Java, apropriado para visualizar árvores de tamanho moderado em espaços tri-dimensionais, usando representações hiperbólicas e técnicas de distorção do tipo *fisheye* (Herman *et al.*, 2000), para permitir ao usuário explorar os dados em diferentes níveis de detalhe. Um exemplo é apresentado na Figura 3.13.

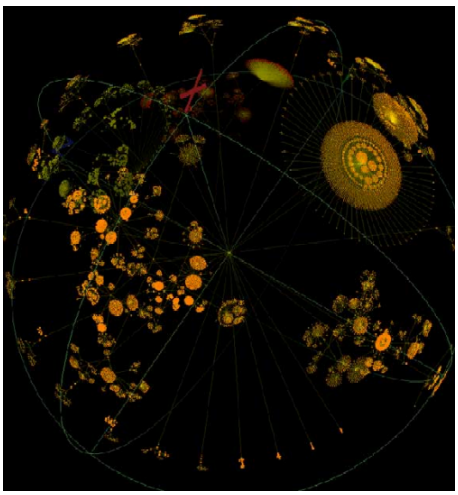


Figura 3.13: *Walrus* : *Layout* hiperbólico 3D (http://www.caida.org/tools/visualization/walrus).

[13]Disponível em http://www.caida.org/tools/visualization/walrus.

Neste trabalho, optou-se por implementar um algoritmo de *layout* de árvores (Bachmaier *et al.*, 2005) no contexto do PEx, conforme descrito no Capítulo 4.

3.5 Considerações Finais

Lidar com grandes conjuntos de textos não é uma tarefa fácil. Esse é um problema que ainda está em aberto. Várias técnicas de visualização de mapas foram desenvolvidas com o objetivo de responder fundamentalmente a duas questões: Como a visualização e a estrutura das relações de similaridade nos mapas de textos influenciam na efetividade do usuário de realizar a análise em um conjunto de textos?. E, a visualização do mapa resultante ajuda efetivamente a enfrentar a difícil tarefa de análise de textos e é facilmente aceita pelos usuários? Para responder a essas perguntas estudos apresentados em Wise *et al.* (1995) concluem que as técnicas projetadas para visualizar mapas facilitam o acesso em conjuntos de textos complexos, ajudando os usuários a entender e memorizar estruturas fundamentais de um conjunto de textos em curto tempo, a selecionar informação e localizar áreas que possam se relacionar a uma consulta específica. Portanto, mapas de textos podem ser usados para realizar diversas tarefas, incluindo a recuperação de textos, análise de domínios, determinação de desempenho e produtividade em pesquisa, gerenciamento de ciência e tecnologia e inteligência competitiva (Borner *et al.*, 2003) e investigações de diversos tipos. Suas técnicas são aplicadas em diferentes conjuntos de dados para extrair conhecimento útil relacionado, por exemplo, na inteligência e segurança de redes (Reid e Chen, 2007), além de anais de conferências e livros-texto baseados em hipertextos (Weippl, 2001).

Os sistemas para mapeamento visual de textos em geral se baseiam em posicionamento no plano baseado em redução de dimensionalidade. Reconstrução de árvores de similaridade como as usadas em filogenia podem apoiar a solução de alguns problemas inerentes às técnicas clássicas de posicionamento baseadas em projeções.

O próximo capítulo é direcionado a usar a estrutura das árvores filogenéticas para construção de mapas de textos, onde as folhas da árvore irão representar os textos e os nós internos representarão textos hipotéticos, demonstrando que essas árvores podem ser utilizadas como metáforas na construção de mapas de textos baseados em conteúdo.

CAPÍTULO

4

Implementação Visual de Filogenias na Construção de Mapas de Textos

4.1 Considerações Iniciais

Em nosso estudo foi crucial escolher uma ferramenta de visualização capaz de refletir o processo de construção das árvores filogenéticas, com o propósito de permitir ao usuário navegar e encontrar informação útil por conteúdo. Apesar dos grandes avanços algorítmicos e visuais alcançados tanto no campo da filogenia como em visualização de informação, existem poucas ferramentas que atingem esses propósitos. A implementação de nossa técnica de visualização por construção de árvores filogenéticas foi feita no sistema PEx, desenvolvido especialmente para criar, visualizar e explorar mapas de conjuntos de dados multidimensionais, especialmente coleções de documentos.

Este capítulo descreve os aspectos de implementação da nova técnica de visualização por árvores filogenéticas, denominada NJ em função do algoritmo utilizado (*neighbor-joining*). Na Seção 4.2 o processo de construção de mapas de textos usando árvores filogenéticas é detalhado, a Seção 4.3 analisa a estrutura e recursos do sistema PEx, a Seção 4.4 discute o processo de implementação da construção e visualização da técnica NJ integrada ao PEx. Finalmente, a Seção 4.5 apresenta um exemplo de aplicação para ilustrar o funcionamento da ferramenta PEx com NJ.

4.2 Processo de construção de mapas de textos usando filogenia

O processo geral para construir e explorar mapas de textos usando a técnica NJ é ilustrado na Figura 4.1. Esse processo é composto fundamentalmente por quatro etapas:

1. Cálculo da matriz de similaridades, aplicando alguma métrica para estabelecer um critério de similaridade entre os textos.
2. Construção da árvore usando o algoritmo de filogenia NJ.
3. Posicionamento do conjunto de vértices na superfície de visualização usando um *layout* de árvores não-enraizadas.
4. Exploração e interação com a árvore gerada.

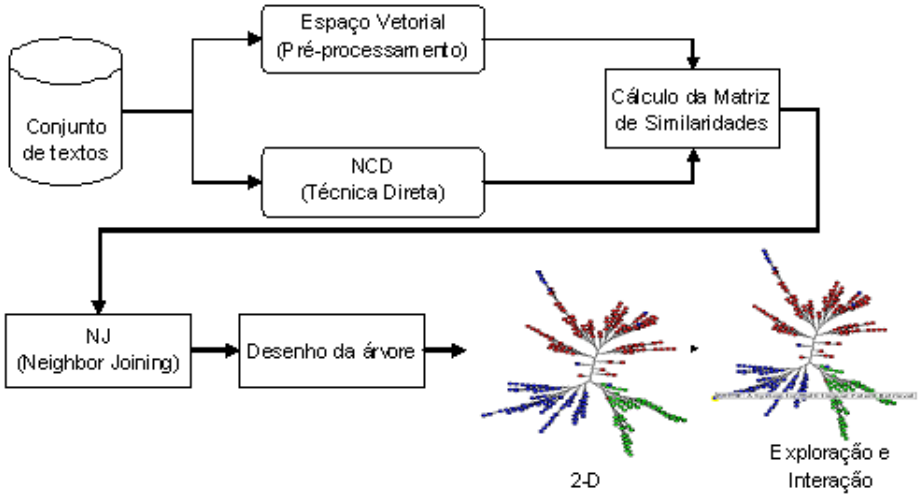


Figura 4.1: Processo geral da construção de árvores filogenéticas usando conjunto de textos.

Definir a matriz de similaridades é importante porque a partir dela se inicia o processo de construção de mapas usando filogenia. Uma matriz de similaridades *M* é uma matriz de números reais e dimensões $n \times n$ onde M_{ij} é a distância entre os textos *i* e *j*. Para o cálculo dessa matriz duas técnicas foram consideradas. A primeira cria uma representação vetorial do conjunto de textos e a segunda usa técnicas diretas para o cálculo de distâncias. Na primeira técnica, cada texto é convertido em um vetor, onde suas coordenadas refletem a freqüência de cada termo presente no texto. Logo após, esses vetores são unidos em uma matriz de textos x termos, tal como foi descrito na Seção 3.2. Em seguida, é realizado o cálculo da similaridade entre um

texto e outro aplicando uma métrica de distância por exemplo a do cosseno. A segunda técnica aplica a aproximação NCD. Para avaliar a medida de similaridade, operações sobre o tamanho compactado dos textos são realizadas (Seção 3.2). A vantagem da primeira técnica é o tempo de processamento, mas é preciso que o usuário tenha um conhecimento prévio do domínio para definir uma série de parâmetros. Com NCD não é preciso realizar passos de pré-processamento para o cálculo das distâncias, porém a desvantagem é seu alto custo computacional.

Uma vez construída a matriz de similaridades, a etapa seguinte é construir uma árvore filogenética usando a técnica *Neighbor Joining* (NJ), que gera como saída uma árvore não-enraizada (Seção 2.3). A estrutura da árvore é exibida na superfície de visualização por meio da aplicação de um algoritmo de desenho de árvores. Os nós folha representam textos, os nós internos representam textos hipotéticos e os ramos refletem as relações de similaridade entre nós folhas. Na visualização inicial do resultado é possível perceber a estrutura global da árvore e a formação dos possíveis grupos potenciais de um conjunto de textos. Na etapa final, conseguida através do espalhamento dos pontos no plano o usuário pode navegar e interagir com os detalhes da informação contida na árvore.

Em termos de implementação, o processo descrito acima foi desenvolvido como uma extensão do sistema PEx, aproveitando diversos recursos, funcionalidades e utilidades previamente existentes para manipular e explorar mapas de texto. Em razão disso foi necessário realizar um estudo detalhado do PEx, que é resumido na próxima seção.

4.3 Visão geral do sistema Projection Explorer (Pex)

Como mencionado anteriormente, o PEx é uma ferramenta útil na construção de mapas de dados multidimensionais, que visa disponibilizar recursos de visualização e exploração de dados via projeções. O sistema PEx é analisado sob dois pontos de vista. No primeiro, ele é analisado em função dos passos que realiza para gerar, visualizar, explorar e interagir com mapas de textos. No segundo, o PEx é analisado em termos de sua implementação.

4.3.1 Funcionalidade do PEx para construção de mapas de dados

O sistema PEx suporta múltiplas opções para a construção de mapas de dados. O processo para essa construção é dividido em 4 etapas: pré-processamento, projeção, visualização e interação com o mapa.

O processo é iniciado conforme os dados a serem analisados. Atualmente (Paulovich *et al.*, 2007b), existem quatro tipos de fontes de dados que podem ser usadas: (1) Dados com atributos numéricos representados em um formato vetorial, (2) Dados de distância representados por

meio de uma matriz, (3) um conjunto compactado de textos em formato ASCII ou UNICODE e (4) um conjunto de textos resultante de uma busca na Internet. Após a fonte de dados ser selecionada, é necessário definir a técnica de projeção que será usada para visualizar os dados. Cinco técnicas são disponibilizadas com o intuito de diminuir a dimensionalidade dos dados: LSP, ProjClus, Sammon *Mapping* e PCA, que foram apresentadas na Seção 2.2, além do IDMAP (Minghim *et al.*, 2006) que é uma combinação de *Fastmap* (Seção 2.2.7) com o *Force Scheme*. Para melhorar a qualidade dos mapas gerados por essas técnicas é possível aplicar a técnica *Force Scheme* (Seção 2.2.8). No caso da criação de uma projeção ser iniciada a partir de um conjunto de textos, as duas estratégias de cálculo de similaridade (vetorial e NCD) estão disponíveis.

A representação visual é na forma de um grafo, onde cada vértice representa um elemento do conjunto de dados. As arestas indicam o relacionamento entre os vértices a partir dos vizinhos mais próximos na projeção (espaço $\mathbb{R}^2$) ou a partir dos vizinhos mais próximos com base no espaço n-dimensional ($\mathbb{R}^n$). É possível também realizar uma triangulação dos pontos resultates. Finalmente, o PEx incorpora varias funcionalidades para ajudar o usuário nas tarefas de exploração visual e interação com o conjunto de textos. Essas funcionalidades permitem entre outras coisas:

- Criar rótulos para a identificação de um conjunto de textos no mapa (Lopes *et al.*, 2007).
- Visualizar o conteúdo de um texto ou grupo de textos, assim como de seus vizinhos simplesmente ao clicar sobre um vértice.
- Colorir vértices no mapa conforme a freqüência de um termo ou conjunto de termos procurados pelo usuário.
- Coordenar duas projeções diferentes. Por exemplo, ao selecionar algum vértice ou grupo de vértices em uma projeção eles serão ressaltados em outra projeção para permitir sua localização.
- Selecionar uma área da projeção e analisá-la conforme dois criterios: ampliá-la, de modo a obter uma análise em maiores detalhes (visão global); ou cortá-la para uma análise em separado da área (visão local).
- Criar e salvar múltiplos mapas ao mesmo tempo.
- Criar ou modificar a cor e tamanho dos vértices no mapa por meio da associação de um valor escalar a eles. O usuário pode mudar o valor ao selecionar um vértice em questão e computar sua similaridade com os demais vértices, ou executar o algoritmo de agrupamento k-*means*. Outra opção é importar um arquivo de escalares.

- Explorar os dados em diferentes níveis de detalhamento. O usuário pode navegar realizando uma exploração geral do mapa e gradualmente identificar informação como relevante ou não, de acordo a seu interesse.

Essas funcionalidades permitem que o usuário seja guiado na aquisição do conhecimento sobre o conjunto de dados em questão. A seguir é apresentada a análise do sistema PEx em termos de sua implementação.

4.3.2 Estrutura interna do sistema PEx

O PEx é uma ferramenta implementada em Java e organizada estruturalmente em pacotes. Cada pacote representa um conjunto de classes e interfaces que estão relacionadas para um fim específico. Na Figura 4.2 é ilustrado o diagrama geral de pacotes do sistema PEx, os quais serão delineados a seguir.

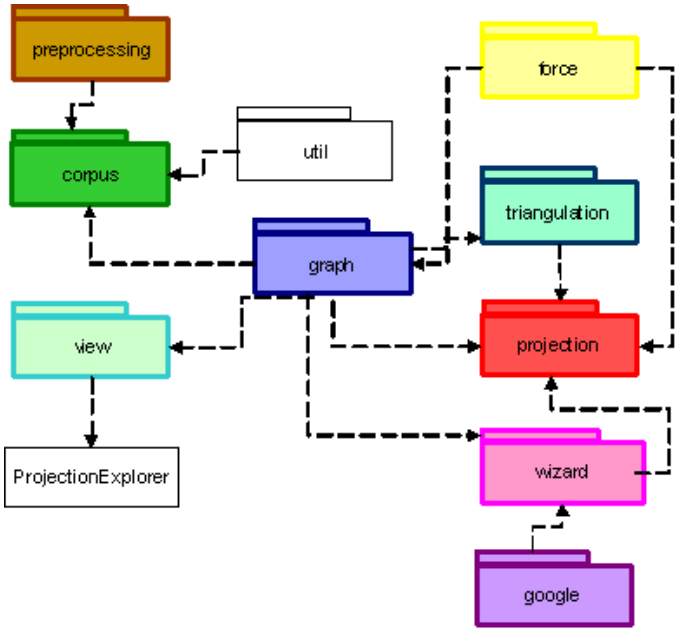


Figura 4.2: Diagrama geral de pacotes no PEx.

O pacote ***corpus*** é composto de classes que permitem importar dados textuais. O pacote ***preprocessor*** contém as classes *stopwords*, *PorterStemmer* e *ngram*, que são responsáveis por realizar as etapas de representação de cada texto como um vetor espacial. Essas classes são gerenciadas pela classe *Preprocessor*, que é responsável por gerar a matriz de textos x termos. As classes principais dos dois pacotes e seus relacionamentos são ilustrados na Figura 4.3.

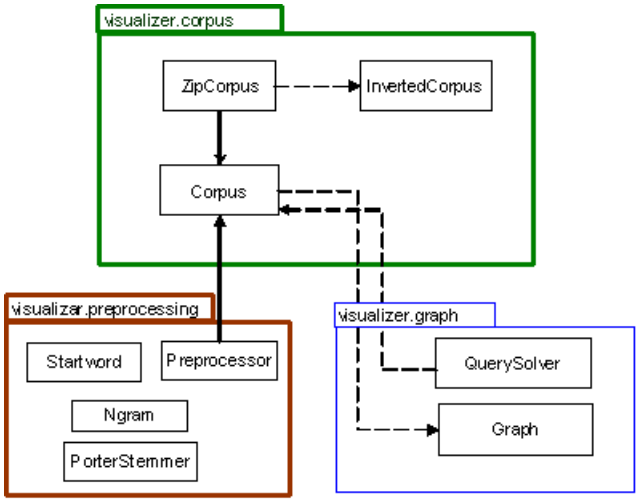


Figura 4.3: Diagrama das principais classes dos pacotes ***corpus*** e **preprocessing** do PEx.

O pacote ***projection*** abrange os mecanismos necessários para criar a matriz de similaridades e aplicar técnicas de projeção na construção de um mapa de textos. Como mostrado na Figura 4.4, a matriz de similaridades é definida no sub-pacote ***distance*** como conseqüência de usar alguma métrica de distância implementadas nas classes *Euclidean*, *CityBlock*, *FastMapSimilarity*, *Kolmogorov*. As técnicas de projeção são implementadas nos sub-pacotes: **idmap**, **lsp**, **pca**, **projclus** e **sammon**, cada um correspondente a uma técnica específica. Considerando que o vetor de textos gerado pela classe *Preprocessor* no pacote **preprocessing** pode ter um grande número de elementos (atributos), pode-se escolher uma técnica que permita reduzir a dimensionalidade dos atributos, como *PCA*, *k-means* e *fastmap*, que estão agrupadas no pacote **dimensionreduction**. O gerenciamento da construção do mapa ou grafo de textos, assim como o controle de suas propriedades, é realizado pela classe *GraphBuilder*. Além disso, essa classe instancia classes do pacote ***triangulation***, responsável por determinar as arestas que ligam os vértices. Para tal fim foram implementadas a classe *Delaunay*, que constrói a triangulação DelaunayEdges; a classe *KNNEdges*, que determina os k vizinhos mais próximos no espaço original dos dados; e a classe *CNNEdges* que possibilita a obtenção dos vizinhos entre os elementos do mesmo grupo.

O pacote ***graph*** é constituído por um conjunto de classes responsáveis pela visualização e exploração do mapa de textos apresentado em forma de um grafo. Em linhas gerais, nesse pacote se criam os componentes *GUI* (*Graphical User Interface*). As classes que criam tais componentes fazem parte do pacote *Swing* do Java (*javax::swing*). Para a representação gráfica dos objetos é usada a biblioteca *Java.awt.graphics*, que permite gerar gráficos em duas dimensões. As funcionalidades para interagir com o grafo, tais como a seleção de vértices em uma área limitada, a criação de rótulos e a coordenação, entre outras, são implementadas no

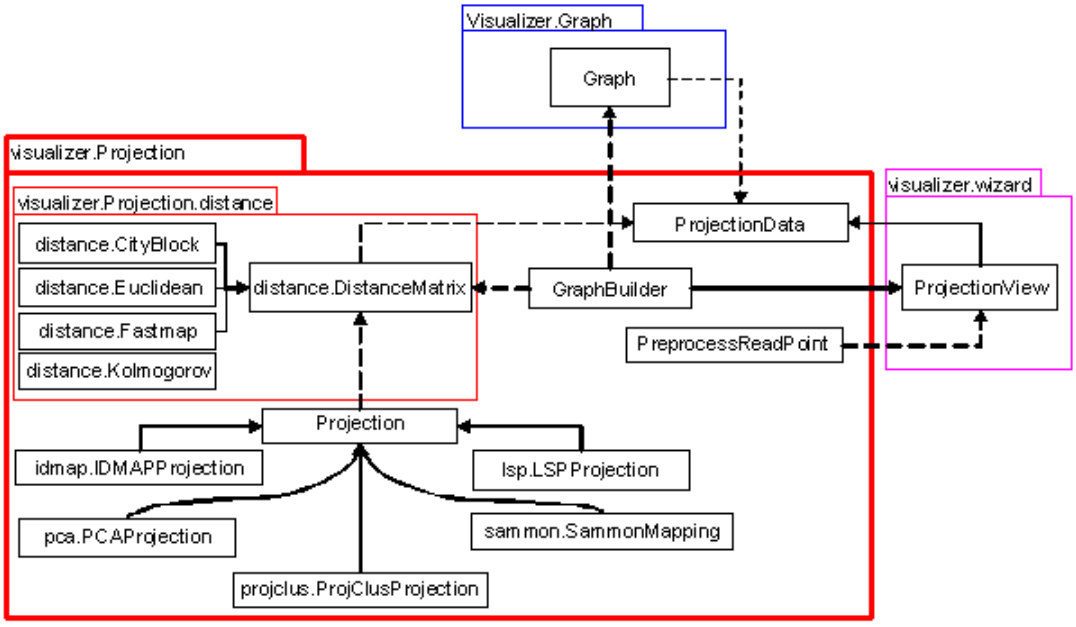


Figura 4.4: Diagrama das principais classes do pacote ***projection***.

sub-pacote **graph.view**. A interação dessas funcionalidades na superfície de desenho do grafo é realizada a partir de botões implementados com funções de escuta de eventos (*event handlers*), eventos esses que transmitem às funções informação do tipo: que botão foi carregado, onde clicou (posição *x*,*y*), arraste do *mouse*, e outros, e que são gerenciados pela classe *GraphPanel*, que por sua vez herda as funcionalidades da janela de classe *JPanel* (pertencente à biblioteca *Java.swing.JPanel*). Além disso, outra classe importante dentro desse pacote é a classe *Graph*, que permite a manipulação das propriedades do grafo. Outro pacote diretamente relacionado com o pacote ***graph*** é o ***force***, encarregado de aplicar um algoritmo baseado em molas para reestruturar a posição final dos vértices. A representação gráfica das principais classes e suas relações é apresentada na Figura 4.5.

O pacote ***view*** é constituído por classes encarregadas da implementação de janelas, por meio das quais pode-se realizar a configuração do sistema, possibilitando o controle dos diversos parâmetros utilizados para gerar um mapa de textos. Estas janelas são definidas como objetos JDialog (*Java.swing.JDialog*), que permitem criar caixas de diálogo para que o usuário possa manipular informação de entrada e saída, como por exemplo, importar escalares de um arquivo externo, exportar um novo conjunto de textos considerando alguma técnica de projeção, modificar a lista de *stopwords*, analisar os parâmetros para o corte de *Lunh*, entre outros. Nesse pacote, também estão incluídas as classes que gerenciam a manipulação de cores, agrupadas no sub-pacote **view.color** e as classes de manipulação de rótulos, agrupadas no sub-pacote **view.label**. O diagrama das principais classes é ilustrado na Figura 4.6.

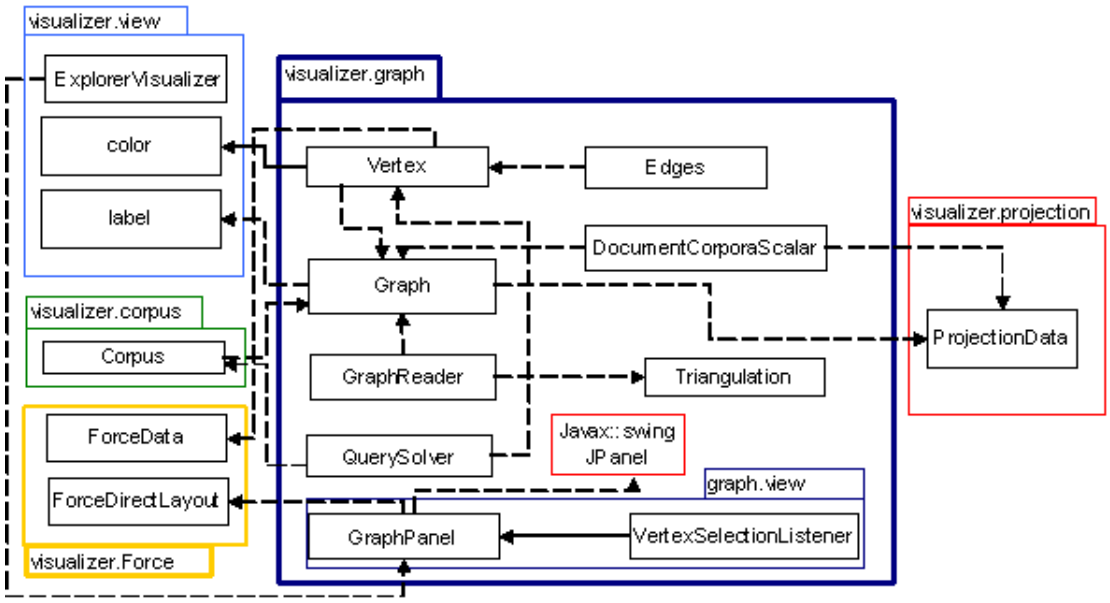


Figura 4.5: Diagrama das principais classes do pacote ***graph*** e ***force***.

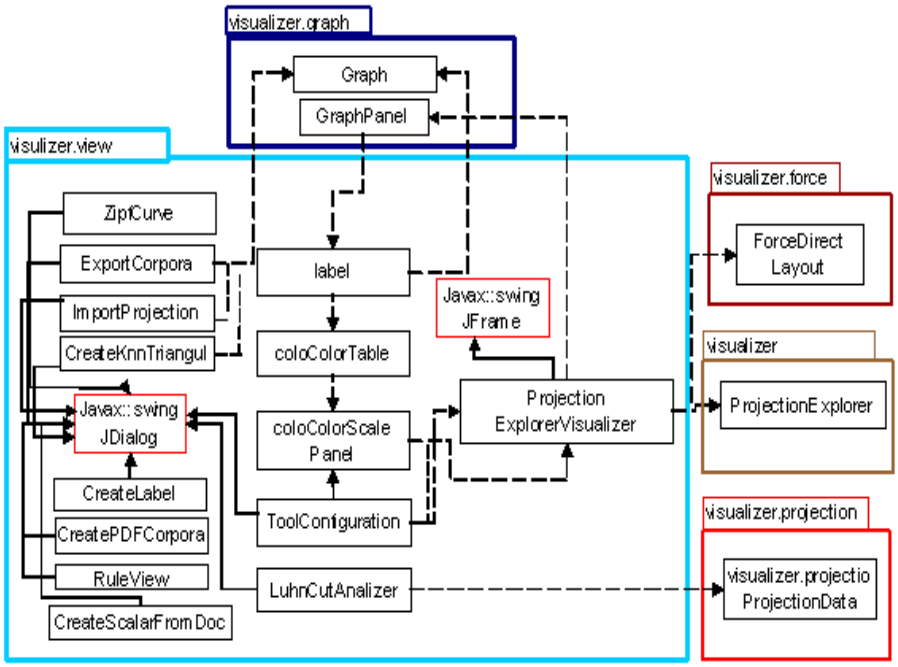


Figura 4.6: Diagrama das principais classes do pacote **view**.

O pacote ***wizard*** é formado por um conjunto de classes que guiam o usuário na construção de uma projeção. Essas janelas herdam as funcionalidades da classe *JPanel* (Java::swing::JPanel), que possuem a capacidade de conter outros objetos dentro de uma janela especificando sua área de atuação. Como exemplos podemos mencionar a janela que permite fixar os parâmetros

na etapa de pré-procesamento dos textos, a janela que permite configurar as características adicionais das técnicas de projeção e as janelas que permitem ajustar os parâmetros conforme a técnica de projeção desejada. A representação gráfica das principais classes do pacote ***wizard***, aparece na Figura 4.7.

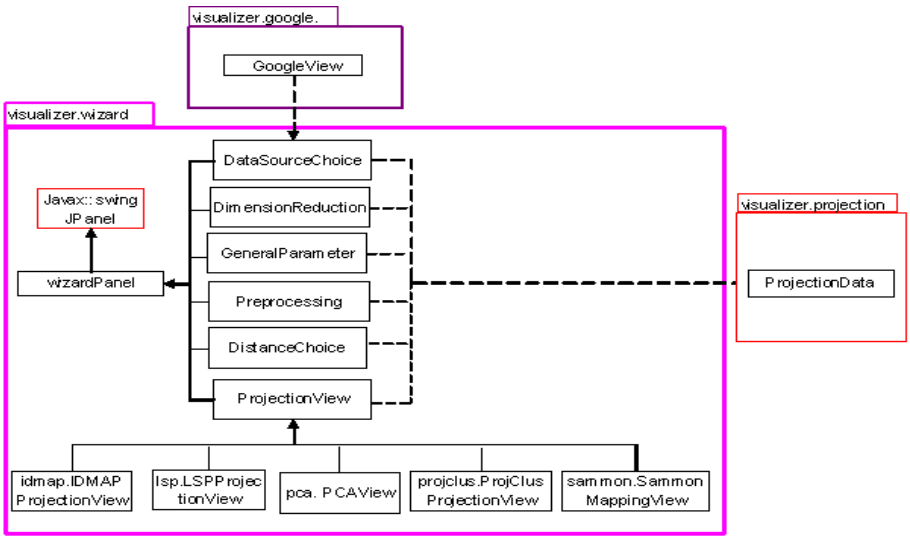


Figura 4.7: Diagrama das principais classes do pacote **wizard**.

Após a análise do sistema em estudo e da definição do processo realizado pela técnica NJ para construir mapas de textos, foram detectados os módulos que deram suporte à inserção do algoritmo *Neighbor Joining* e sua correspondente representação visual na estrutura do sistema PEx. Os detalhes dessa extensão são apresentados na próxima seção.

4.4 Implementação e processo de construção da árvore filogenética

Para demonstrar que árvores filogenéticas podem fornecer uma visão alternativa na construção e exploração de conjuntos de textos, foi necessário realizar a implementação da técnica de filogenia NJ em três etapas: (1) implementação do algoritmo de geração da árvore filogenética *Neighbor Joining* (NJ), (2) implementação do algoritmo de desenho de árvores e (3) adaptação das funções de interação próprias do PEx às árvores.

Para realizar a primeira etapa, de acordo com a forma descrita na Seção 2.3.1, o NJ recebe como dado de entrada uma matriz de similaridades e gera como saída uma árvore sem raiz. Pela análise feita na seção anterior, sabe-se que o PEx possui duas formas de gerar essa matriz de similaridades, a primeira é por meio do passo de pré-processamento para visualizar textos a

partir do espaço vetorial e a segunda é por meio de métricas diretas de cálculo de similaridades como NCD. Essa matriz é a entrada para o algoritmo NJ.

A estrutura interna da árvore está implementada dentro do pacote **projection** do PEx e sua organização é definida no sub-pacote NJ, formado principalmente pelas classes *NJNode* e *NeighborJoining*. Na classe *NJNode*, são definidas as propriedades dos nós da árvore, como tipo de nó, posição em coordenas *x* e *y*, distâncias, entre outros. Na classe *NeighborJoining* é implementada a função da construção da árvore conforme o Algoritmo 1 na Seção 2.3.2. A estrutura interna da árvore está contida em um vetor, no qual cada elemento faz referência a um nó. Essa estrutura foi escolhida principalmente porque o acesso aos índices do vetor é realizado de forma direta, eliminando assim a necessidade de implementação de funções de busca. Uma vez que o número de nós na árvore é definido, operações de eliminação e adição de elementos não serão realizados.

Para compatibilizar as estruturas árvore e grafo no sistema PEX, os nós da árvore com suas respectivas propriedades são armazenados na classe *vertex* e a relação entre eles (ramos) na classe *edges* do pacote ***graph***. A organização do algoritmo NJ e as relações com as classes do PEx são apresentadas na Figura 4.8.

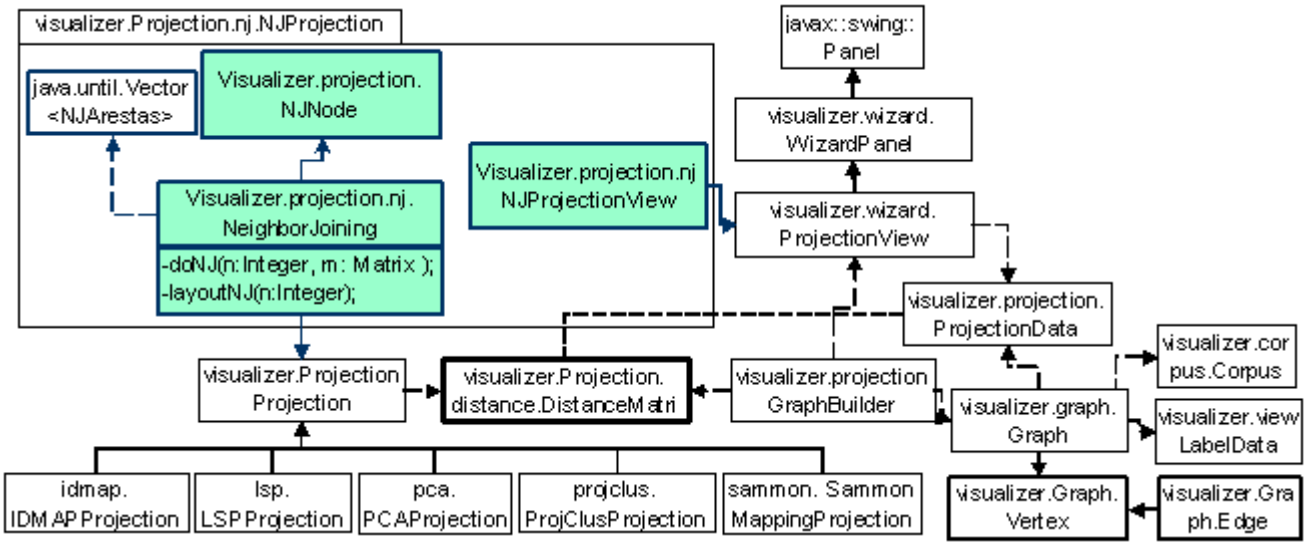


Figura 4.8: Organização do pacote NJ no contexto do PEx.

Dado que a técnica NJ foi implementada como uma extensão do sistema PEx, os recursos de interação com o mapa de textos descritos na Seção 4.3.1 se adaptam como funções de interação na árvore. Elas são descritas a seguir.

4.4.1 Descrição das funcionalidades de interação

Uma vez construída a árvore filogenética, a segunda etapa de desenvolvimento foi implementar a visualização e a interação. Para tal fim, foram implementados: 1) o algoritmo de desenho de árvores filogenéticas que reflete o grau de similaridade entre os textos e 2) funções de interação com o mapa.

O algoritmo de **desenho de árvores** é baseado no *layout* radial (Bachmaier *et al.*, 2005), que o diferencia de outros *layouts* (circular, *phylogram* e *cladogram*) por tentar preservar ao máximo a distância entre os ramos e porque as arestas se estendem radialmente afastando-se da raiz.

A idéia principal da construção da árvore usando *layout* radial é atribuir a cada sub-árvore *T(v)* uma porção angular proporcional ao número total de folhas, tal como ilustrado na Figura 4.9. A porção angular (α) que corresponde ao nó *v* é dividida entre seus filhos w_1 e w_2 e as arestas se estendem ao longo do ângulo bissector da porção angular que corresponde a cada nó. O comprimento das arestas é determinado pela distância (δ) entre os nós.

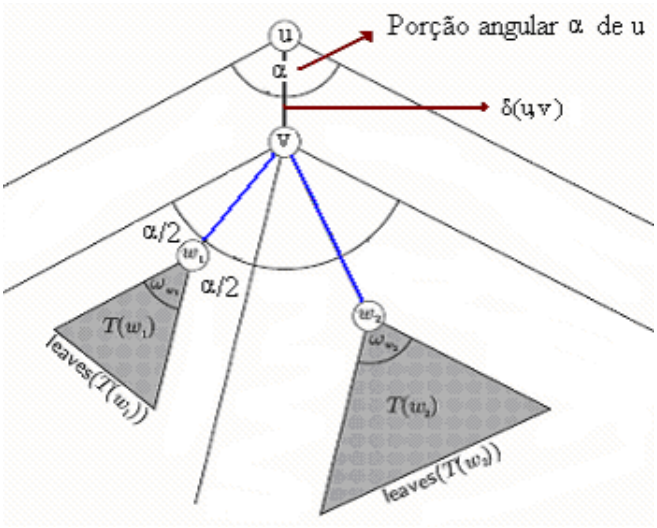


Figura 4.9: Representação angular do *layout* radial (Bachmaier *et al.*, 2005).

No algoritmo descrito por Bachmaier (Bachmaier *et al.*, 2005), uma árvore filogenetica é denotada por $T = (V, E, \delta)$ com $n = |V|$ vértices, $m = |E|$ arestas e distâncias positivas das arestas ($\delta : E \rightarrow \mathbb{R}^+$). O Algoritmo 2 começa percorrendo duas vezes todos os nós da árvore, iniciando pela raiz (*root*(T)) [14]. A primeira realiza uma varredura pós-ordem, na qual é determinado o número de folhas (l_v) de cada sub-árvore (T_v). Na segunda, a varredura é feita

[14]Como a técnica NJ gera uma árvore sem raiz, o nó central da árvore é usado como raiz.

em pré-ordem, na qual um filho *w* de um vértice interno *v* é localizado a uma distância $\delta(v, w)$ na porção do ângulo bissector reservada para *w*. A construção é feita em tempo linear.

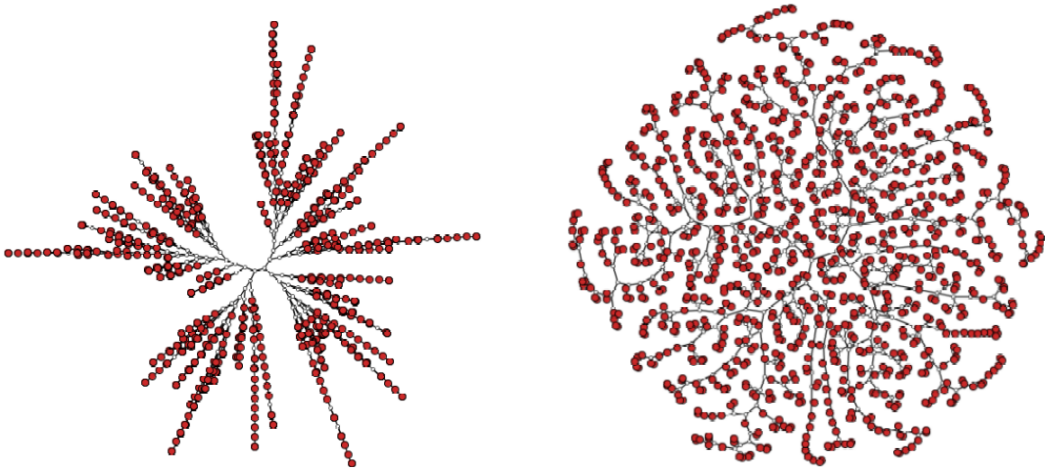
Entrada: árvore $T = (V, E, \delta)$
Dados: vetor de vértices *l* (número de folhas na sub-árvore), ω (tamanho da porção angular), τ (porção do ângulo), $deg(v)$ (número de arestas incidentes em um vértice), η (variável de acumulação da porção angular).
Saída: coordenadas x_v para todo $v \in V$
begin
$postorder_traversal(root(T))$
$x_{root(T)} \leftarrow (0, 0)$
$\omega_{root(T)} \leftarrow 2\Pi$
$\tau_{root(T)} \leftarrow 0$
$preorder_traversal(root(T))$
end

procedure $postorder_traversal(vertice\ v)$
se $deg(v) = 1$ **então**
senão
$lv \leftarrow 1$
fim
$lv \leftarrow 0$
para cada $w \in filho(v)$ **faça**
$postorder_traversal(\omega)$
$l_v \leftarrow l_v + l_\omega$
fim

procedure $preorder_traversal(vertice\ v)$
se $v \neq root(T)$ **então**
$u \leftarrow pai(v)$
$x_v \leftarrow x_u + \delta(u, v) \cdot (\cos(\tau_v + \frac{\omega_v}{2}), \sin(\tau_v + \frac{\omega_v}{2}))$
fim
$\eta \leftarrow \tau_v$
para cada $w \in filho(v)$ **faça**
$\omega_w \leftarrow \frac{l_w}{l_{root(T)}} \cdot 2\Pi$
$\tau \leftarrow \eta$
$\eta \leftarrow \eta + \omega_w$
$preorder_traversal(w)$
fim

Algorithm 2: *Layout* radial (Bachmaier *et al.*, 2005).

A Figura 4.10(a) apresenta o resultado da aplicação do *layout* radial sobre a estrutura da árvore filogenética. O mapa é composto por 1.000 textos contendo notícias de jornal, no qual as relações de similaridade entre textos podem ser determinadas claramente pelos ramos. Após a primeira apresentação no *layout*, pode-se usar o algoritmo de desenho de grafos baseado em molas incluído na ferramenta PEx para expandir a árvore e melhorar sua distribuição no plano (Figura 4.10(b)).



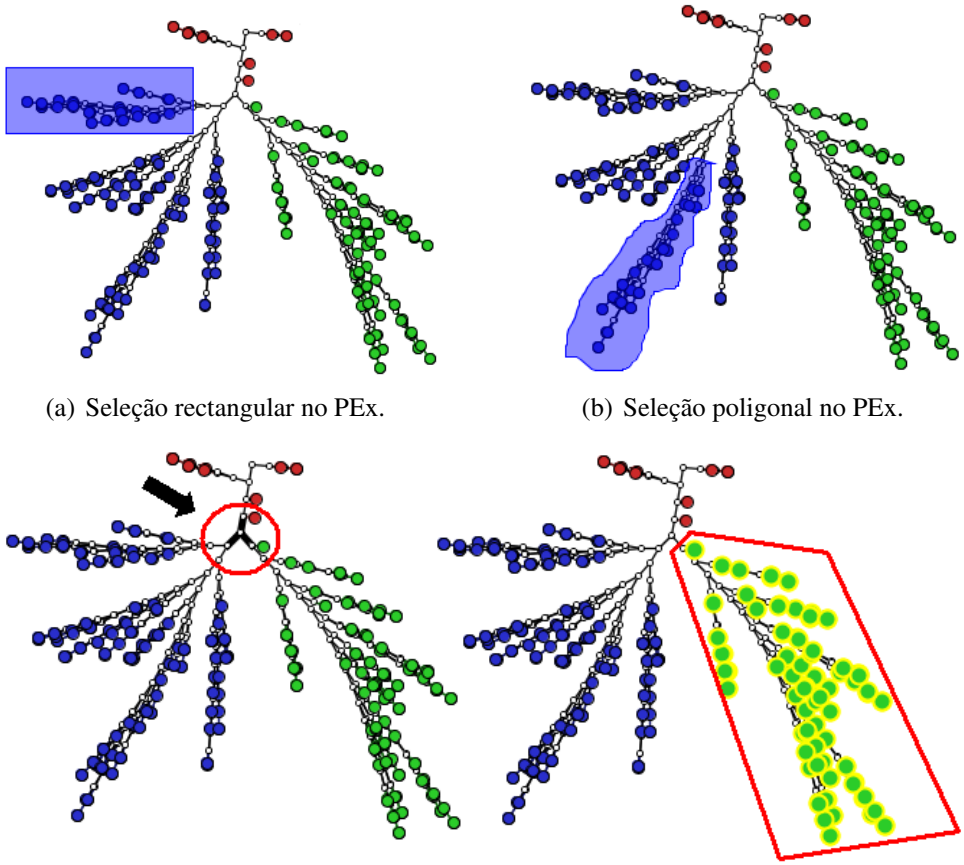
(a) *Layout* radial implementado. (b) *Layout* baseado em molas do PEx.

Figura 4.10: Visualização da árvore usando os dois *layouts* no PEx.

No que se refere às **funções de interação da árvore**, no PEx existem vários componentes funcionais que trabalham com a visualização, exploração e ações de interação do usuário com um conjunto de textos.

As funções interativas do PEx são implementadas geralmente considerando a seleção de vértices (textos) no plano de visualização. Essa ação é feita selecionando uma região retangular ou poligonal (Figura 4.11), mas para o caso das árvores é preciso adaptar outra forma de seleção e, conseqüentemente, foi adicionada uma terceira, que permite selecionar diretamente todos os nós de um ramo. Essa ação é realizada em dois passos: 1) clicando no nó pai do nó raiz do ramo que será selecionada e 2) clicando no nó filho que se tornará na raiz do ramo selecionado (Figura 4.11(c)).

A última forma de seleção foi estendida para seu uso em funcionalidades já implementadas no PEx e que foram adaptadas para seu funcionamento na árvore. Por exemplo, para permitir que o usuário identifique o conteúdo comum presente nos textos que formam um ramo foi usada a funcionalidade de geração de rótulo previamente implementada no PEx. Na Figura 4.12, o retângulo amarelo é um exemplo de rótulo para o ramo selecionado.



(a) Seleção rectangular no PEx.

(b) Seleção poligonal no PEx.

(c) Seleção do ramo implementada.

Figura 4.11: Formas de selecionar nós na árvore.

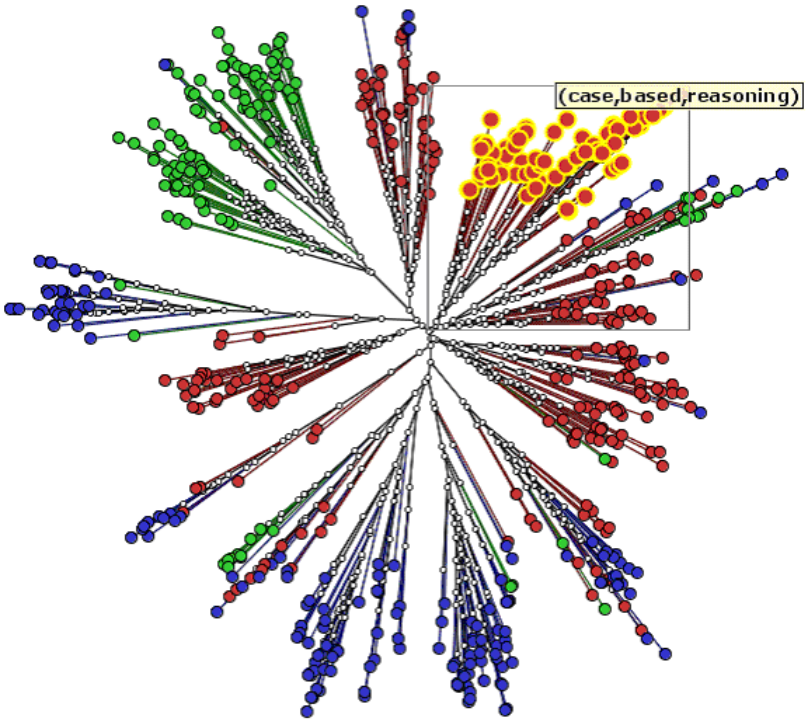
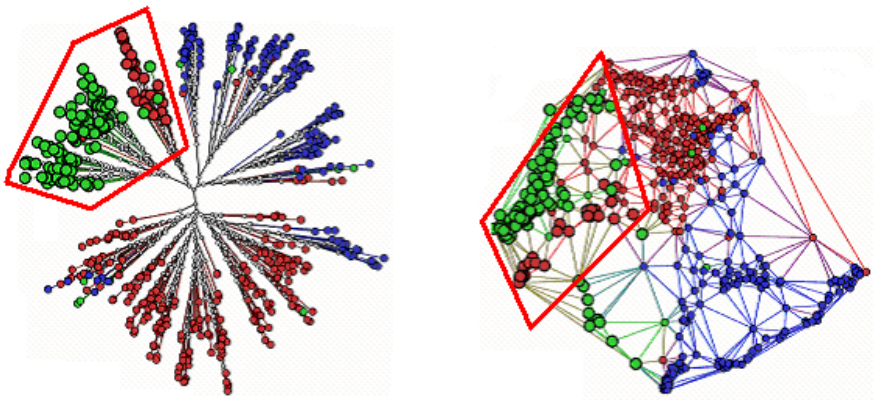


Figura 4.12: Criação do rótulo do ramo no conjunto de textos CBR+ILP+IR.

No caso do usuário necessitar de uma análise mais detalhada de uma parte da árvore, ele pode selecionar uma área da árvore, ampliá-la e tratá-la como uma árvore em uma janela separada. De igual modo, as funcionalidades de visualização de conteúdo dos textos, deslocamento dos vértices selecionados e coordenação de múltiplas visões foram adaptadas de acordo com os requisitos da árvore. Por exemplo, na Figura 4.13 é ilustrada a coordenação entre uma árvore e uma projeção. Uma vez que um ramo é selecionado na janela NJ (Figura 4.13(a)), a seleção é refletida na janela da projeção (Figura 4.13(b)) e conseqüentemente, os textos que suportam alguma consulta em NJ são destacados na projeção e vice-versa.



(a) Janela de visualização NJ.

(b) Janela de visualização da projeção LSP.

Figura 4.13: Coordenação de múltiplas visões aplicado no conjunto de textos CBR+ILP+IR.

Além de adaptar as funcionalidades já implementadas no PEx para seu uso com a técnica NJ, foi adicionada a possibilidade de colorir os ramos ou conjunto de nós da árvore. Tais nós são selecionados de acordo com algum critério estabelecido pelo usuário, no qual cada cor está associada a um valor escalar do nó, como apresentado na Figura 4.14 .

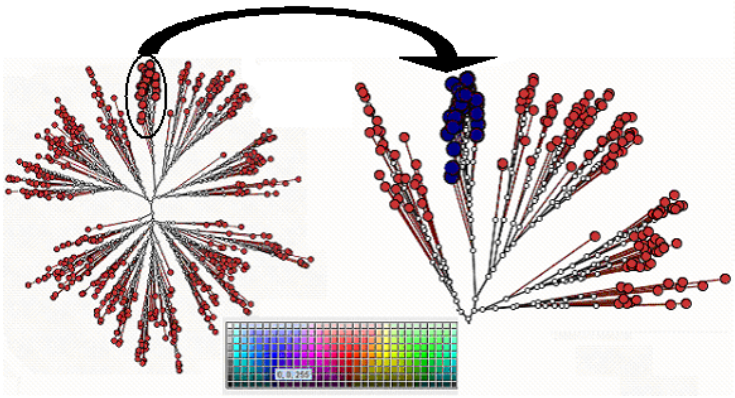


Figura 4.14: Coloração do ramo como resultado de selecionar um ramo do conjunto de textos CBR+ILP+IR.

Outra funcionalidade implementada foi identificar textos vizinhos com base nas ligações existentes entre os nós conforme com a estrutura da árvore. O número de nós vizinhos a ser mostrado é selecionado pelo usuário, como apresentado na Figura 4.15.

Finalmente, as características próprias da árvore na geração do mapa criado são armazenadas em formato XML para reutilização em análises futuras.

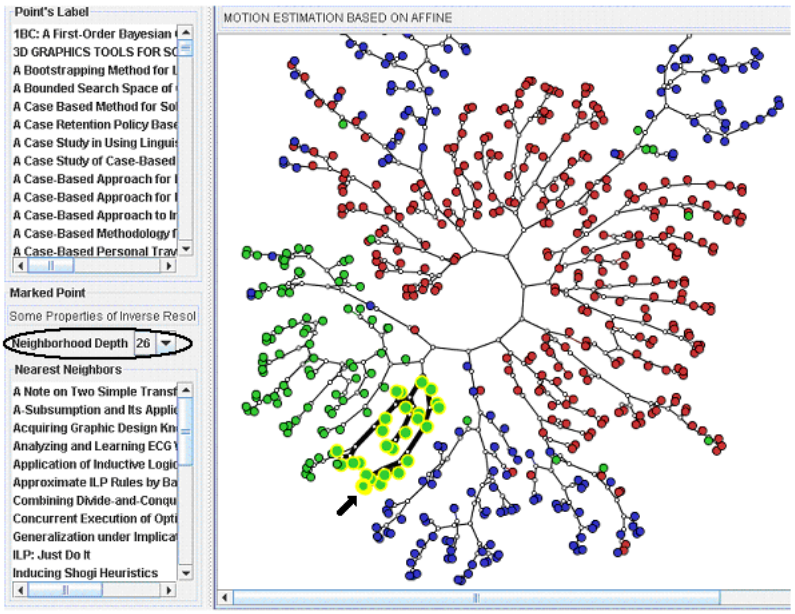


Figura 4.15: Ligação de 26 textos vizinhos ao vértice circundado com uma linha grossa amarela.

Uma vez definidos o processo, a estrutura e funcionalidades aplicáveis na construção de árvores filogenéticas em mapas de textos, um exemplo detalhado de aplicação é apresentado na próxima seção.

4.5 Exemplo de aplicação para visualizar mapas de textos usando filogenia

Para deixar claro como são realizadas a visualização e interação no processo de construção das árvores filogenéticas, a Figura 4.16 explica o processo geral descrito anteriormente, agora destacando a nova técnica NJ de mapeamento de textos no sistema PEx.

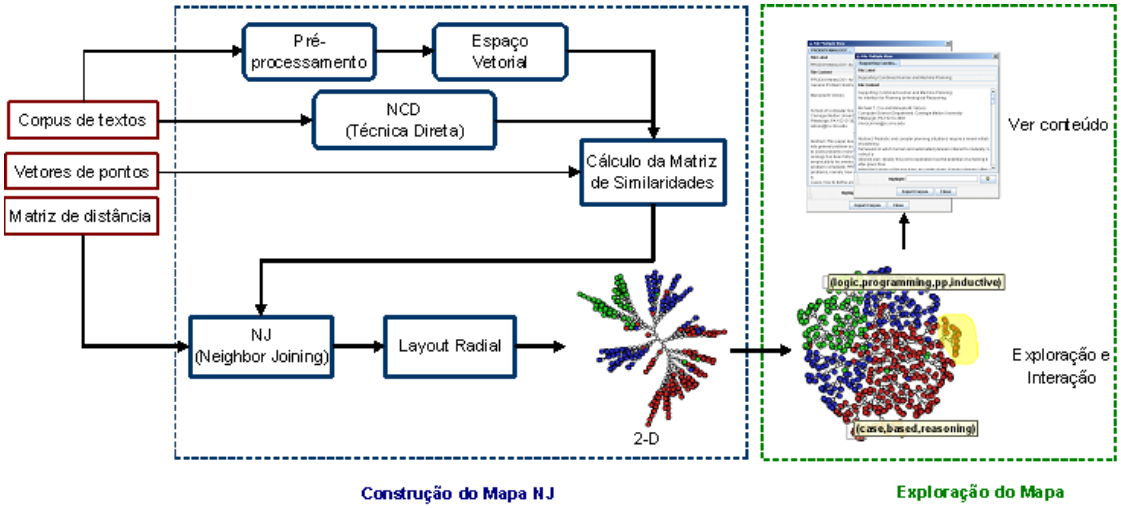


Figura 4.16: Processo de construção de mapas de textos empregando a técnica NJ na ferramenta PEx.

Para mostrar o desempenho da técnica NJ e a capacidade de exploração e interação da ferramenta, o conjunto de textos *News* foi utilizado. Ele contém 2.684 notícias coletadas entre os dias 5 e 7 de abril de 2006 do *website* de quatro agências de notícias (BBC, CNN, *Reuters* e *Associated Press*). Esse conjunto de textos é caracterizado por conter muitas notícias de conteúdo similar ou redundante, além de serem muito curtas em média. O mapa foi gerado usando título e conteúdo das notícias. O cálculo da matriz de similaridades foi realizada a partir do vetor de representação de textos, que depois de realizar o passo de pré-processamento ficou com 1.042 dimensões. O corte de *Luhn* inferior foi de 12. A Figura 4.17 mostra a árvore gerada pelo *layout* radial na janela principal do sistema PEx.

Para que a exploração de um mapa de textos seja feita de forma a facilitar a extração de conhecimento de um conjunto de textos, é importante que ferramentas de visualização forneçam ao usuário: 1) uma visão geral dos tópicos que estão sendo tratados no conjunto de textos e 2) mecanismos adequados para encontrar informação útil e realizar a análise dos dados em maior profundidade. No caso da ferramenta PEx, o usuário pode navegar e procurar informação na estrutura global da árvore e gradualmente fazer a distinção entre ramos relevantes ou irrelevantes. Assim, por exemplo, na Figura 4.17 é apresentado o primeiro *layout* que reflete a topologia inteira da árvore, onde os textos estão distribuídos aparentemente em ramos densos. O usuário pode inferir, em primeira instância, a possível tendência que segue o conjunto *news* na formação de grupos. Logo após, pode-se concentrar em ramos específicos e nas relações existentes entre um grupo de notícias que pertencem a um ramo ou grupos de ramos. Para maior detalhe, pode concentra-se também nas relações existentes entre notícias individuais.

Para o usuário determinar que tipo de informação está contida em cada nó e ramo do mapa é preciso que ele possa identificar os tópicos essenciais dentro de um texto ou coleçãp de textos.

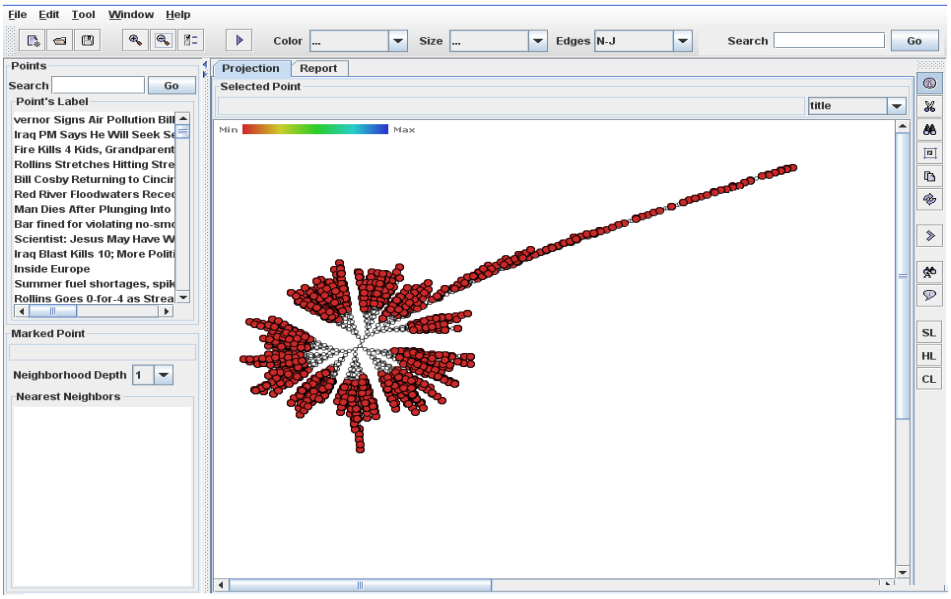
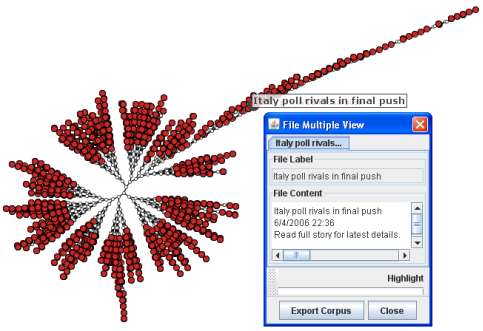


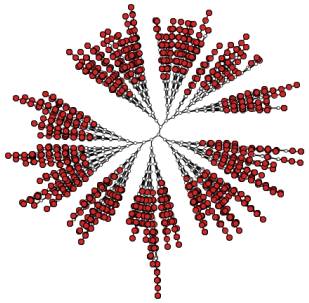
Figura 4.17: Representação visual da árvore do conjunto de textos *News* na janela principal do PEx.

Uma opção implementada no PEx é criar rótulos baseados no conteúdo dos textos que pode ser usada conforme as necessidades do usuário, tais como:

- Se o objetivo do usuário é identificar nós individualmente, basta passar o *mouse* pelo nó e o rótulo aparecerá automaticamente. Caso o usuário precise consultar o conteúdo do nó, isso pode ser feito por meio de um clique duplo sobre ele (Figura 4.18(a)). Se o conteúdo é de interesse para o usuário, ele pode selecionar os nós e destacá-los. Caso contrário, ele poderia, por exemplo, suprimí-los. Assim, a Figura 4.18(b) mostra o mapa de *News* em que são removidos os nós que estão posicionados no ramo mais longo. Esses nós possuíam título mas seu conteúdo era apenas "*read full story for latest details*".



(a) Identificação de um nó ao mostrar seu rótulo e visualização de seu conteúdo ao executar um duplo clique sobre o conjunto *news*.



(b) Mapa *news* ampliado depois de suprimir o ramo sem ramificações.

Figura 4.18: Processo de identificar nós individualmente.

- Se o usuário necessita identificar os principais tópicos por ramos, a funcionalidade de rótulo por regiões deve ser ativada de modo a selecionar um grupo ou ramo de interesse. O rótulo do ramo é formado pelos termos com mais elevada covariância do grupo de nós selecionados. Na Figura 4.19, cada ramo é identificado por seus tópicos principais. Assim, por exemplo, o rótulo (*china,stones*) identifica o ramo da região (A) e os nós que contém esses termos são destacados de forma a permitir sua identificação visual. Os nós coloridos em azul são mais similares conforme o tema identificado no ramo e os nós tendendo para vermelho são menos similares. O ramo da região (B) foi selecionado aplicando a funcionalidade seleção por ramo e o ramo da região (C) aplicando seleção poligonal.

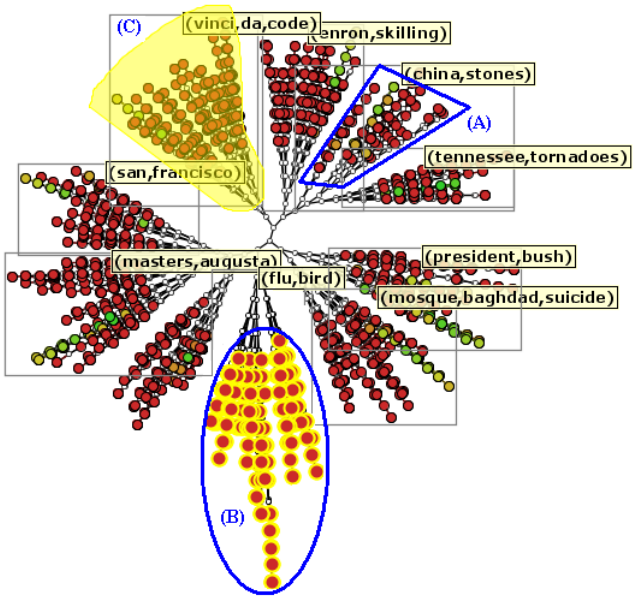


Figura 4.19: Exploração da árvore por ramos.

- Alternativamente, se o usuário deseja identificar automaticamente os tópicos principais discutidos em um conjunto de textos, o algoritmo de agrupamento *k-means* pode ser executado. Por exemplo, na Figura 4.20 foi executado o algoritmo *k-means* sobre o conjunto *news* projetado. As principais notícias identificadas foram: processo de autoria a respeito do lançamento do polemico filme *Da Vinci Code*; processo legal contra o administrador da energética norte-americana *Enron Corp.* por sua possível participação na falência da empresa; discurso de um novo projeto de lei para imigração na pauta do senado dos Estados Unidos; ataques no Iraque; aparição de uma ave morta no Reino Unido, gerando alarme em torno da enfermidade *flu bird*; e inicio da temporada do torneio de golfe *Masters Augusta*.

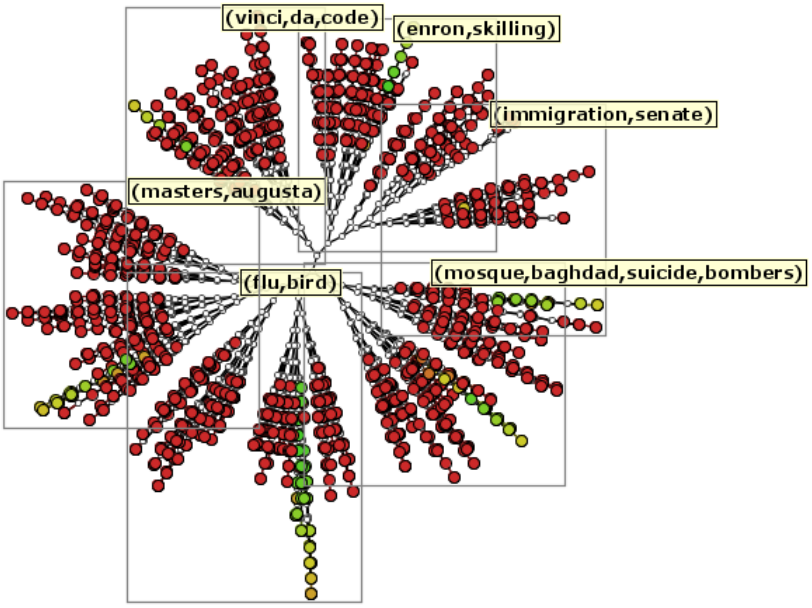


Figura 4.20: Exibição de rótulos depois da aplicação do algoritmo de agrupamento *k-means* sobre o conjunto *news*.

Todas essas funcionalidades são úteis quando não se tem conhecimento prévio da informação contida em um conjunto de textos, apoiando o usuário a determinar que assunto está sendo tratado dentro dos nós no mapa, facilitando assim seu processo de aprendizagem.

Para realizar buscas com um maior nível de detalhamento, a exploração sobre o conteúdo presente nos nós pode ser realizada gradualmente, de modo a melhorar a compreensão das relações existentes entre os textos. No PEx existem varias opções por explorar sub-ramos, como:

- Encontrar informação contida em ramos para explorar sub-ramos. Por exemplo, no ramo com rótulo (*mosque, baghdad, suicide, bombers*) da Figura 4.20, pode-se encontrar sub-ramos com informação específica a respeito de outros tópicos de notícias. Assim, na Figura 4.21 é apresentada uma hierarquia de notícias novas por meio da exploração de sucessivos sub-ramos. Por exemplo, o ramo representativo do tópico (*mosque, baghdad, suicide*) é dividido em dois sub-ramos, onde foram encontrados novos tópicos referentes a: protestos no Nepal contra o rei Gyanendra e ataques suicidas em mesquitas em Bagdá. O sub-ramo que ainda conserva o rótulo (*mosque, baghdad, suicide*) é dividido em dois sub-grupos de notícias: a detenção do ministro Hamas em Israel e ataques em Bagdá. Se descermos mais um nível, um novo grupo de notícias é encontrado: um frustrado ataque terrorista que iria acontecer nas eleições gerais na Itália.

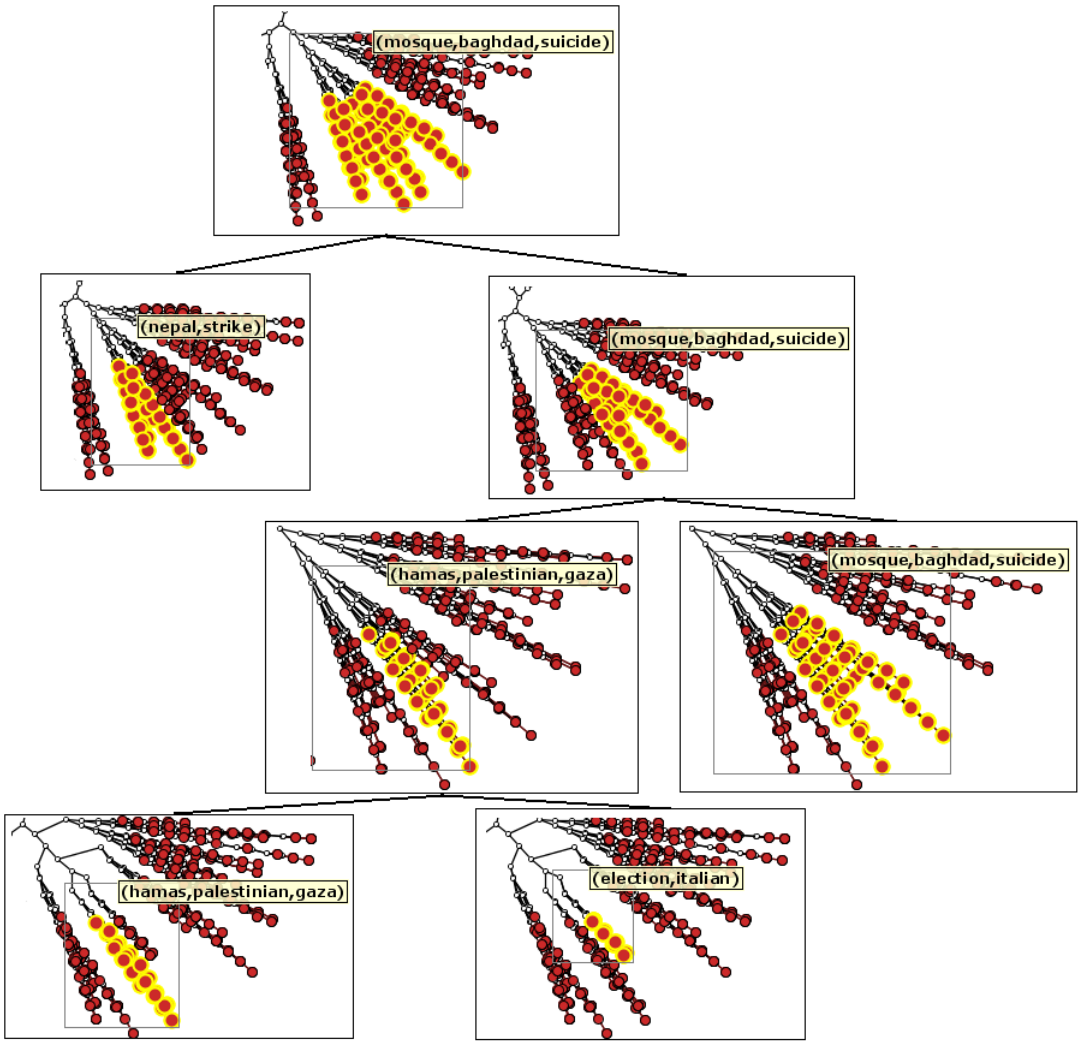


Figura 4.21: Exploração de informação por ramos usando principalmente funcionalidades de ampliação, seleção de ramos e deslocamentos de ramos. As notícias são posicionadas nos ramos conforme o cálculo da similaridade entre nós.

- Encontrar informação pela freqüência de ocorrência de um ou mais termos, onde os nós são coloridos no mapa de acordo a essa freqüência. A busca de informação por meio dessa funcionalidade é apresentada na Figura 4.22. Na figura, os nós são coloridos pela freqüência do termo *president*. No ramo da região (**A**) se observa principalmente que existe maior ocorrência do termo *president*, que parecem estar localizadas em dois sub-ramos. Uma exploração mais detalhada confirma que os textos que tratam do tema "investigação de vazamento de informação de inteligência sobre Iraque na Casa Branca" foram agrupados no sub-ramo (1) e os textos que tratam sobre "eleições presidenciais no Perú" foram agrupados no sub-ramo (2).

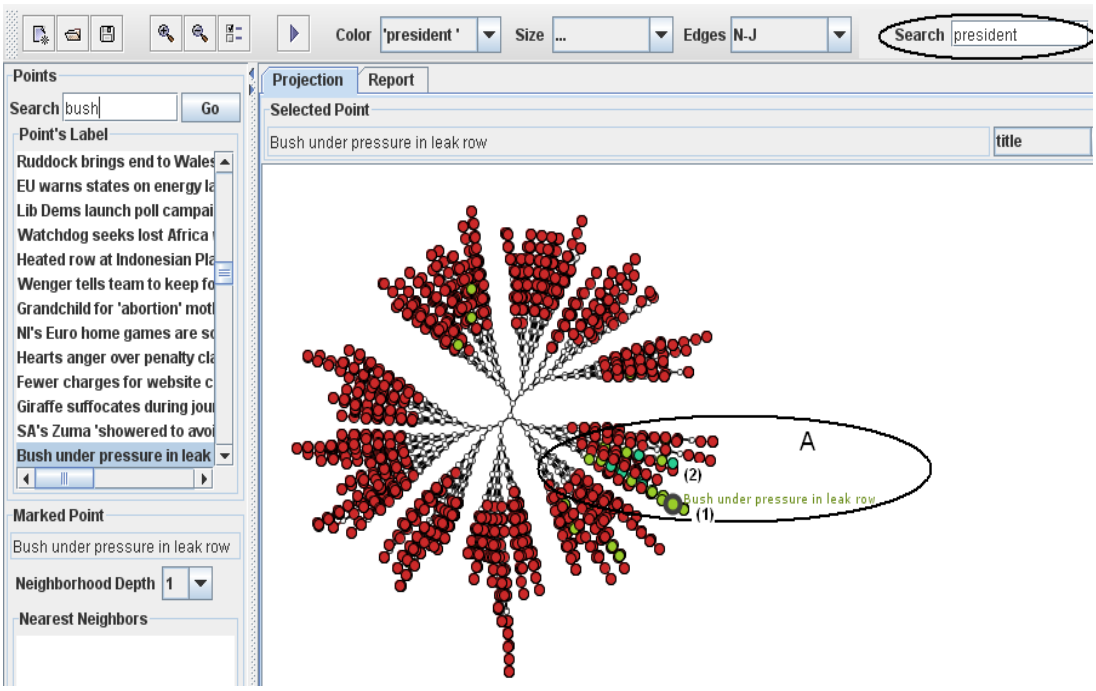
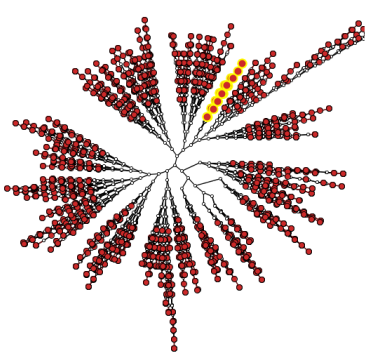


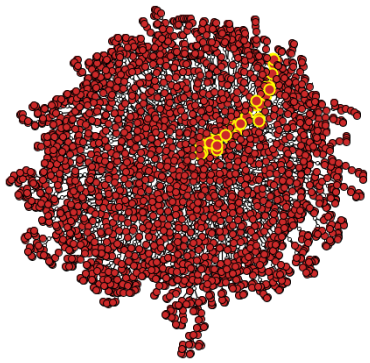
Figura 4.22: Busca de informação no conjunto *news* coloridas de acordo à freqüência do termo *president*.

- Explorar as relações de similaridade entre os nós por meio da aplicação do algoritmo de molas para o desenho de grafo do PEX. Esse algoritmo permite reestruturar a posição dos nós gerados pelo *layout* radial, de forma a expandir os ramos da árvore, minimizando os cruzamentos entre arestas. Exemplos ilustrativos no próximo capítulo demonstram a utilidade do layout baseado em molas na exploração de textos.

  É importante destacar que a representação da topologia da árvore apresenta comportamentos diferentes dependendo do algoritmo utilizado: *layout* radial ou de molas. Por exemplo, nas Figuras 4.23 e 4.24 são apresentadas duas visões do mapa do conjunto *news* ao aplicar os *layouts* radial e de molas no PEx. Na árvore das Figuras 4.23(a) e 4.24(a) o usuário mantém uma visão global da topologia inteira da árvore. As relações entre os nós podem ser inferidas em nível de grupos. Mas, em contrapartida, existe demasiada sobreposição entre os nós. A árvore das Figuras 4.23(b) e 4.24(b) fornece uma visão detalhada das relações de similaridade entre os nós, principalmente no nível de relações entre sub-ramos e no nível de relações entre nós individuais. Embora a sobreposição dos vértices tenha diminuído, observa-se uma perda da organização dos nós na estrutura global da árvore. É importante salientar que o usuário pode apoiar sua exploração nas funcionalidades de ampliação, redução e corte das áreas selecionadas nos dois mapas, facilitando ao máximo o processo de exploração em relação à sobreposição.

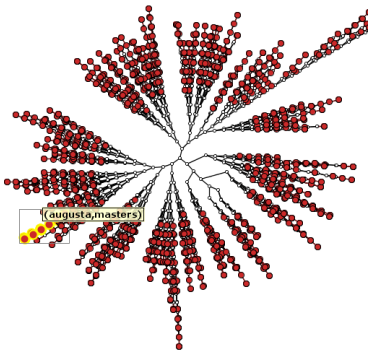


(a) A idéia da similaridade entre nós do ramo selecionado em relação à árvore inteira é mantida no *layout* radial.

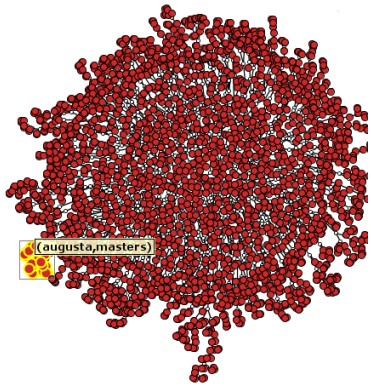


(b) A idéia da similaridade entre nós do ramo selecionado e a árvore inteira é perdida no *layout* de molas.

Figura 4.23: Representação topológica da árvore para o conjunto *news* gerada por meio do sistema PEx.



(a) A análise individual dos nós no mapa é difícil de ser realizada no *layout* radial pela sobreposição dos nós.



(b) A análise individual dos nós é mais fácil de se realizar no *layout*de molas.

Figura 4.24: Representação topológica da árvore para o conjunto *news* no PEx.

Finalmente, para compreender as relações de vizinhança entre os nós, é usada a funcionalidade *Neighborhood depth* no PEx, que foi modificada para seu uso em árvores. Se um clique simples for executado sobre um nó, seus vizinhos são identificados de acordo com a trajetória que segue o nó no processo de formação da árvore. Na Figura 4.25, os 20 nós vizinhos do nó com rótulo "*Tornadoes Sweep Through Tennessee*" são destacados no mapa. Na sub-janela *Nearest Neighbors*, uma lista com os rótulos dos nós vizinhos que estão relacionados com o final de temporada de tornados em Estados Unidos é mostrada.

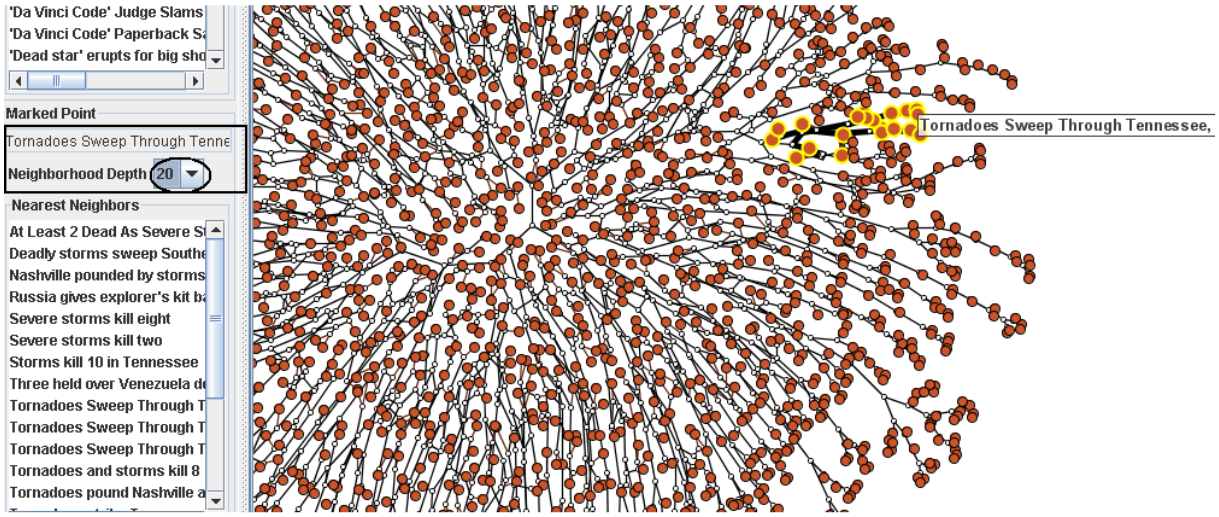


Figura 4.25: Nos vizinho do nó "Tornadoes Sweep Through Tennessee" para o conjunto de textos News.

Um estudo das relações de similaridade que existem entre os nós da árvore é ilustrado em maior detalhe no próximo capítulo.

4.6 Considerações finais

Neste capítulo foi analisada a estrutura interna do PEx, por meio da descrição de como são realizados o processo de construção de mapas e sua representação visual, e como os diversos recursos implementados apóiam na exploração e interação com textos. Uma vez entendido o esquema de implementação do PEx foi possível acoplar à sua estrutura o algoritmo de construção de árvores filogenéticas *Neighbor Joining*.

O próximo capítulo mostra os resultados alcançados visando verificar o desempenho da técnica NJ e sua validade na construção de mapas de textos.

CAPÍTULO

5

Discussão de Resultados e Conclusões

5.1 Considerações Iniciais

Como mencionado no Capítulo 2, os resultados obtidos até o momento pelas técnicas de redução de dimensionalidade na construção de mapas de textos são bons, no entanto ainda existem problemas para serem solucionados. Dois desses problemas referem-se ao posicionamento de uma fração de pontos em vizinhanças inadequadas e a dificuldade na compreensão do relacionamento entre pontos próximos, que são muitas vezes posicionados uns sobre outros em mapas de tamanho moderado.

Em vista disso, os objetivos deste capítulo são dois. Primeiro, tentar demonstrar que a técnica NJ baseada na construção de árvores filogenéticas é aplicável na construção de mapas de textos, permitindo tornar claras as relações de similaridade refletidas pela medida utilizada sobre um conjunto de textos, delimitando áreas de conhecimento por conteúdo e identificando textos que pertencem a essas áreas e suas possíveis subáreas. Segundo, avaliar os resultados da técnica NJ com os resultados de projeção e, para isso, nosso estudo é focalizado sob três pontos de vista: 1) comparação visual, 2) cálculo de precisão dos pontos e 3) tempo de construção do mapa.

Na Seção 5.2 é analisada a aplicabilidade da técnica NJ na construção de mapas de textos em termos de resultados. Na Seção 5.3 é analisado o comportamento do NJ em relação às técnicas de projeção. Na parte final são apresentadas as conclusões obtidas durante o desenvolvimento do trabalho, apontando algumas melhorias que podem ser realizadas em trabalhos futuros.

5.2 Análise dos resultados

Para a análise do desempenho da técnica NJ foram realizados vários testes sobre conjuntos de dados, especialmente com textos, com o intuito de validar a capacidade do NJ em construir mapas de textos de várias fontes. Dessa forma, vários mapas contendo informação sobre artigos científicos, mensagens de grupos de discussão, notícias de jornal, entre outros, foram construídos. Nesse contexto, mapas de textos devem ser entendidos como um conjunto de pontos que estão localizados no plano de visualização, no qual pontos que se mostram próximos têm conteúdo mais similar do que pontos que se mostram distantes (Paulovich e Minghim, 2006a). A tabela 2.1 apresenta um resumo dos conjuntos de dados utilizados e suas respectivas características para mostrar a validade do NJ.

Tabela 5.1: Conjunto de dados utilizados para mostrar a validade da técnica NJ.

Conjunto de dados	Tipo	No. de Dados
CBR+ILP+IR+SON	artigos científicos	680
KDVis	artigos científicos	1.624
INFOVIS04	artigos científicos	515
ALL	artigos científicos	2.819
MESSAGES	mensagens de discussão	300
NEWS	notícias curtas	2.684
Quadruped Mammals	numérico	10.000

Considerando esses mapas, avaliaremos a capacidade do NJ de ser aplicado na construção e exploração de mapas de dados multidimensionais de conteúdo similar. Para isso, a análise da técnica NJ é guiada sobre três princípios: 1) capacidade de agrupar textos em regiões bem definidas, 2) capacidade da técnica NJ em formar ramos com textos de conteúdo similar 3) capacidade de posicionar textos de conteúdo similar o mais próximo possível e 4) capacidade de trabalhar com diversos tipos de textos e dados.

Para validar esses princípios, usaremos como exemplo o mapa da Figura 5.1 gerado ao aplicar a técnica NJ ao conjunto de dados CBR+ILP+IR+SON, que inclui artigos que foram

extraídos de revistas científicas e de buscas na Internet. Cada artigo contém seu respectivo título, autores, resumo e referem-se a quatro diferentes tópicos ou áreas: *Case-based Reasoning* (*CBR*), *Inductive Logic Programming* (*ILP*), *Sonification* (*SON*) e *Information Retrieval* (*IR*). Os pontos no mapa foram coloridos de acordo com o tópico ao qual pertencem para facilitar sua análise. Salientamos que para o cálculo da matriz de similaridades entre os textos foi usado o modelo de espaço vetorial e a distância baseada no cosseno. O corte de *Luhn* inferior foi de 11, com 2.767 atributos.

De acordo com o mapa de textos CBR+ILP+IR+SON da Figura 5.1, podemos inferir que a técnica NJ distribui os nós na superfície de visualização separando bem os quatro tópicos tratados conforme a pré-classificação realizada no conjuntos de textos, o que mostra que os nós de assuntos similares são colocados na mesma área e separados das outras, e que conseqüentemente a árvore é capaz de separar em subárvores artigos de conteúdo similar. O mapa apresenta alguns nós que parecem estar mal posicionados. Isto ocorre pelo fato da classificação dos artigos ser realizada manualmente de acordo com a fonte de extração do texto. Portanto, um artigo que foi classificado em um determinado grupo, também pode ser considerado parte de outro por possuir conceitos que pertencem às diferentes áreas de classificação.

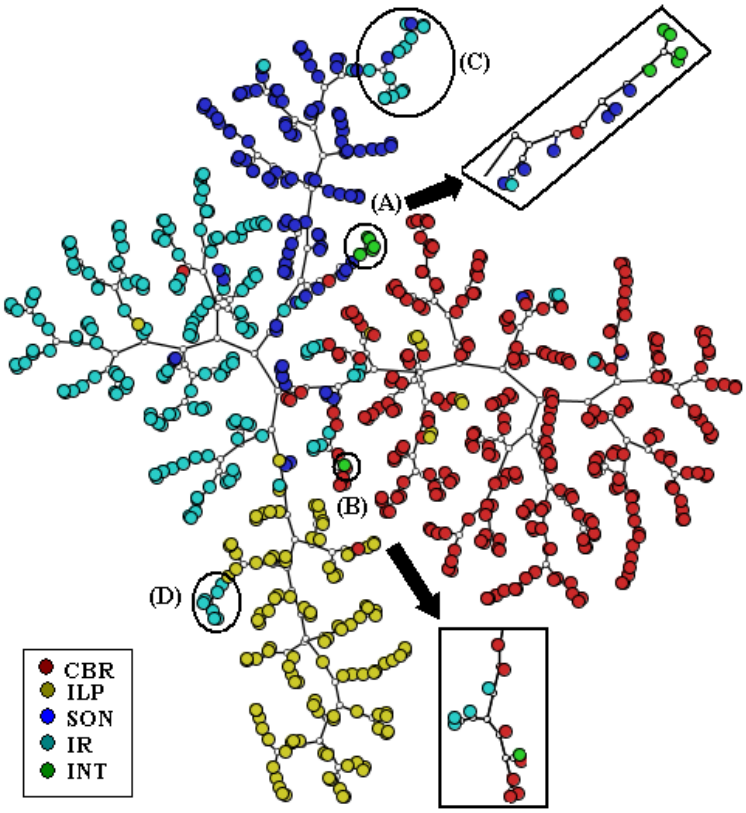


Figura 5.1: Mapa de textos CBR+ILP+IR+SON. A técnica NJ é capaz de agrupar textos baseados em conteúdo em regiões bem definidas.

De forma a mostrar o segundo princípio, isto é, capacidade da técnica NJ em formar ramos com textos de conteúdo similar, no conjunto de dados CBR+ILP+IR+SON foram adicionados 6 artigos; seus rótulos começam com INT e a cor mapeada é verde. O tópico principal para cinco deles é SON (sonificação), o sexto artigo é *outlier* [15] e trata de um tema diferente daqueles das demais classes (*image segmentation*). Como ilustrado na Figura 5.1, a região marcada como **(A)** mostra que os cinco artigos novos que possuem como tema comum a evolução do sistema DSVOL (*Distributed Sound for Volumes*) formam parte do mesmo ramo e são colocados bastante perto. Destacamos que o nó de cor vermelha (*CBR*), que faz parte do mesmo ramo e que pareceria estar mal posicionado, trata de modelos de visualização 3D, tema que é discutido pelos cinco artigos novos. A região marcada como **(B)** identifica a posição do *outlier*. Outro exemplo que permite validar que NJ separa bem os textos em ramos com diferente conteúdo, circunda os nós marcados como **(C)** e **(D)**. No caso da região marcada como **(C)**, nós que segundo a classificação deveriam estar posicionados na área *IR* e estão localizados na área de SON estão mal posicionados.

Realizando uma análise mais exaustiva, os artigos mal posicionados discutem assuntos sobre recuperação de informação de áudio "*audio information retrieval*", que é mais similar ao conteúdo presente em artigos do grupo *SON*. Essa análise foi feita aplicando a funcionalidade de rótulo por ramo como mostrado na Figura 5.2. No caso da região **(D)**, acontece o mesmo que na região **(C)** ou seja, o ramo que está formado na maioria por artigos do tópico *ILP* contém um grupo de artigos classificados como *IR*. Ao analisar o conteúdo desses artigos vemos que tratam de situações de aprendizado *ILP* que consideram apenas os algoritmos que satisfazem algum critério de qualidade para validação de aprendizado, conhecido como modelo *PAC-learning*. Portanto o conteúdo é mais similar aos artigos *ILP*.

Outro exemplo que confirma a capacidade da técnica NJ em definir regiões considerando áreas comuns por conteúdo, é mostrado na Figura 5.3, onde o mapa construído para o conjunto de textos KDVis é formado por 1.624 artigos científicos obtidos de um repositório na Internet[16] e que inclui textos sobre *Bibliographic Coupling* (BC), *Co-citation Analysis* (SC), *Milgrams* (MG) e *Information Visualization* (IV). Nesse caso, foi usada a distância NCD para determinar o grau de correlação entre os artigos.

Com esses exemplos, podemos sugerir que os ramos na árvore são formados por nós que possuem conteúdo similar.

[15] Um *outlier* é um objeto que possui atributos que são incomuns da maioria de outros objetos do conjunto de dados.

[16] http://ella.slis.indiana.edu/ katy/outgoing/hitcite/bc,sc,mb,iv.txt

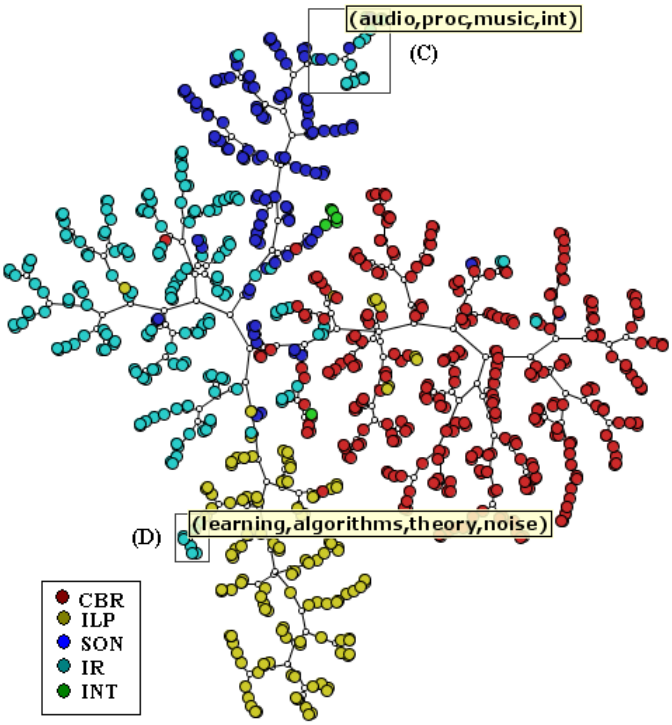


Figura 5.2: Rótulo por dois ramos do conjunto CBR+ILP+IR+SON. A função de rótulo permite mostrar coerência entre a posição dos textos e seu conteúdo.

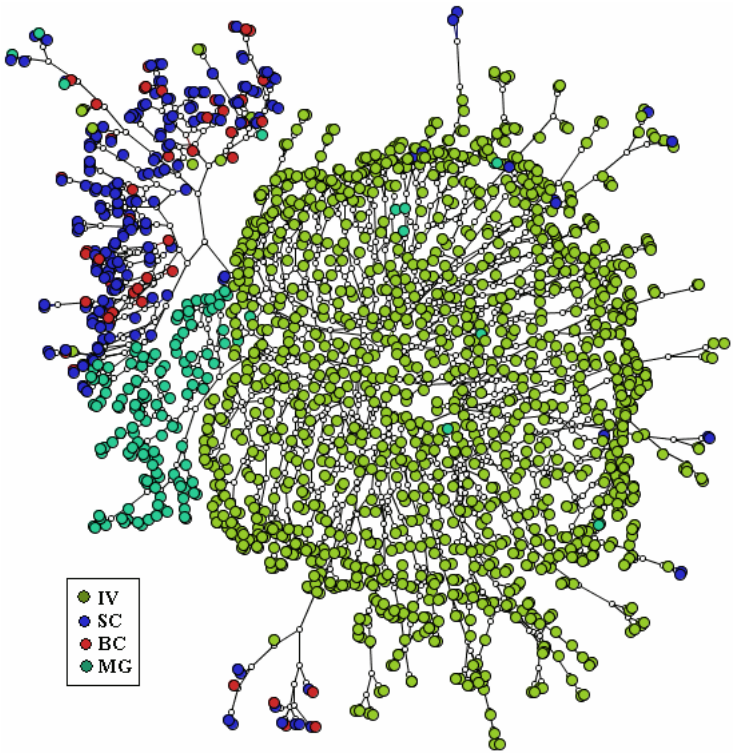


Figura 5.3: Exemplo do mapa gerado para o conjunto de artigos científicos KDVIS.

Para o terceiro princípio, isto é, capacidade de posicionar textos de conteúdo similar o mais próximo possível, seguiremos apoiados na análise do conteúdo existente nos cinco artigos definidos como novos e que são altamente correlacionados. Fazendo um exame minucioso no seu conteúdo, verificamos que os artigos, além de tratar do mesmo tema, foram escritos pelos mesmos autores. A Figura 5.4 mostra a região ampliada do ramo onde esses artigos foram posicionados. Nessa ampliação é possível verificar que os nós folhas da região **(A)** e **(B)**, que corresponde aos cinco artigos novos, estão posicionados nos extremos do ramo. Os artigos da região **(A)** tratam sobre a validade de aplicar técnicas de som nas visualizações, sendo que o artigo "*An Interaction Model ...*" que é mais recente, referencia o artigo "*User Evaluations of Interactive*". Portanto, é comprovado um alto grau de similaridade entre eles. O mesmo caso acontece na região **(B)**. O quinto artigo, o mais antigo e intitulado "*Sonification to support visualization*" , e referenciado por um artigo de cada grupo, pode ser chamado de pai dos outros artigos.

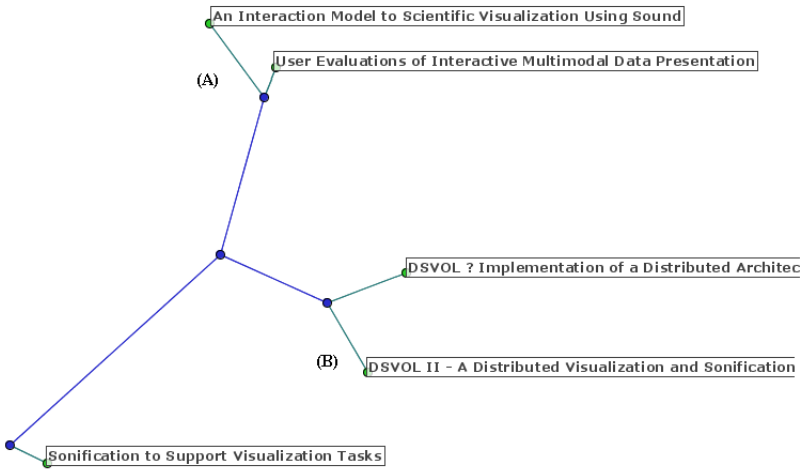


Figura 5.4: Ampliação de uma parte do ramo do conjunto CBR+ILP+IR+SON, para mostrar a similaridade entre os nós filhos.

Considerando esses exemplos, fica claro que a técnica NJ posiciona nós similares o mais próximo possível e que os nós que aparecem nos extremos do ramo apresentam um alto grau de similaridade. Logo, a informação contida nas sub-árvores ou ramos pode ser abstraída em diferentes níveis de detalhe.

Conseqüentemente, como mencionado acima, é possível verificar que a técnica NJ pode ser empregada na construção, exploração e interpretação de mapas de conjuntos de textos. Um aspecto importante a destacar é que os resultados do NJ dependem fortemente da matriz de similaridades, e sempre que as medidas de distância são capazes de produzir uma boa distinção por conteúdo a técnica NJ é capaz de ligar apropriadamente textos vizinhos.

Como exemplos adicionais, na Figura 5.6, na Seção 5.3.1 é ilustrado o mapa de textos para o conjunto *Messages*, que contém mensagens de três grupos de discussão e cujos dados foram obtidos de um repositório da internet [17]. Nesse caso, novamente a NJ consegue separar os pontos por temas de discussão e conseqüentemente por proximidade de conteúdo. A medida de similaridade utilizada neste caso foi baseada no cosseno, com corte de *Luhn* inferior de 157.

Além de comprovar a efetividade da técnica NJ em relação a tipos de dados textuais, também foi possível demonstrar a aplicabilidade da técnica em conjuntos de dados não textuais, como é o caso do conjunto *Quadruped Mammals* (Gennari *et al.*, 1989) formado por 10.000 elementos, composto das classes: *dog*, *cats*, *horses* e *giraffes* com 72 atributos e o formato de entrada são vetores em um espaço multidimensional. A construção do mapa para esse conjunto de dados é ilustrada na Figura 5.5

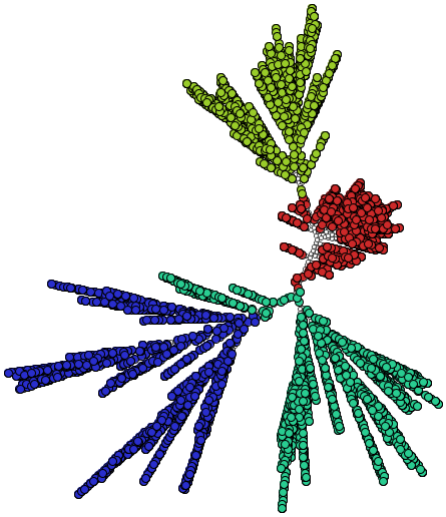


Figura 5.5: Mapa construído para o conjunto de dados *Quadruped Mammals*.

Uma vez demonstrada a efetividade e aplicabilidade da técnica NJ, o passo seguinte é determinar seu desempenho comparado com os resultados das técnicas de projeção de dados multidimensionais, o que será tratado na próxima seção.

5.3 Comparação da técnica NJ com técnicas de Projeção

As técnicas de projeção se tornaram cada vez mais úteis no processo de extração de conhecimento, representando uma boa opção para explorar conjuntos de dados multidimensionais

[17] http://kdd.ics.uci.edu

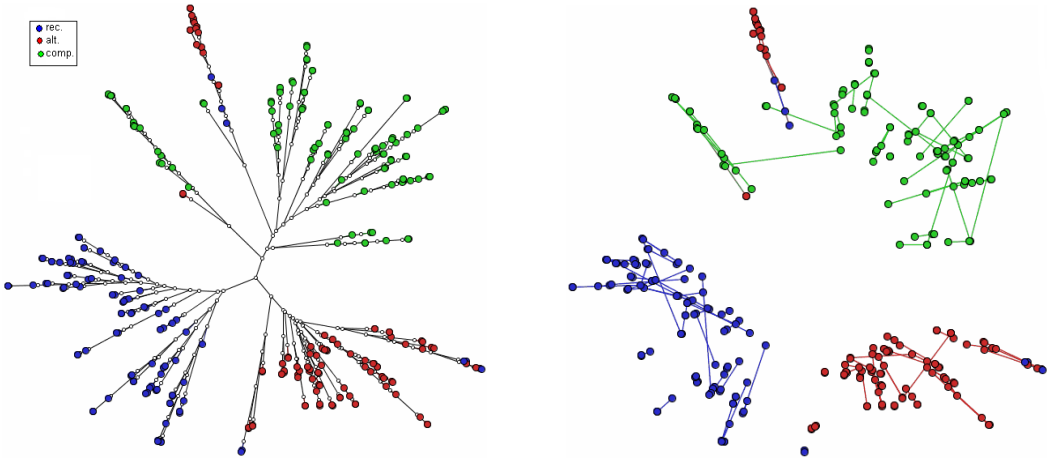
no plano de visualização, de forma a facilitar o entendimento das relações existentes entre os dados. Dado seu comprovado sucesso (Paulovich e Minghim, 2006b), (Paulovich *et al.*, 2006) e em razão de compartilhar características em comum com a técnica NJ é possível estabelecer resultados de comparação entre elas, principalmente no que diz respeito ao comportamento visual dos dados, ao posicionamento dos dados no plano de visualização e ao tempo de processamento.

5.3.1 Comparação visual entre as técnicas NJ e projeção

Para o caso da comparação visual entre NJ e projeções, baseamos nosso estudo na estrutura dos dados exibidos no plano e na forma como apresentam seus resultados.

No que se refere à estrutura dos dados, embora ambas as técnicas reflitam uma mesma representação básica (os nós no mapa representam os textos e as arestas revelam o grau de similaridade entre eles) os resultados devem ser interpretados considerando conceitos diferentes. Para projeções textos similares são colocados na mesma área de vizinhança no plano, já para técnicas de filogenia, devem ser colocados no mesmo ramo. Para ilustrar esse comportamento, na Figura 5.6(a) apresentamos o mapa resultante de aplicar a técnica de filogenia NJ sobre o conjunto de textos *Messages* de 300 mensagens de discussão e na Figura 5.7 o mapa gerado pela técnica de projeção LSP para o mesmo conjunto de dados. Delas podemos inferir que o mapa gerado por NJ é definido pela topologia da árvore e as relações de similaridade entre os textos são definidas pela trajetória que se segue através dos ramos, considerando seu tamanho. No caso da técnica LSP, observamos claramente que é possível separar os textos em três grupos que representam as regiões onde os pontos estão posicionados muito próximos, isto é, que exibem alta densidade, oferecendo uma ótima visão global, mas localmente é difícil relacionar textos vizinhos porque praticamente estão posicionados uns sobre outros, dificultando a percepção de relacionamento entre eles.

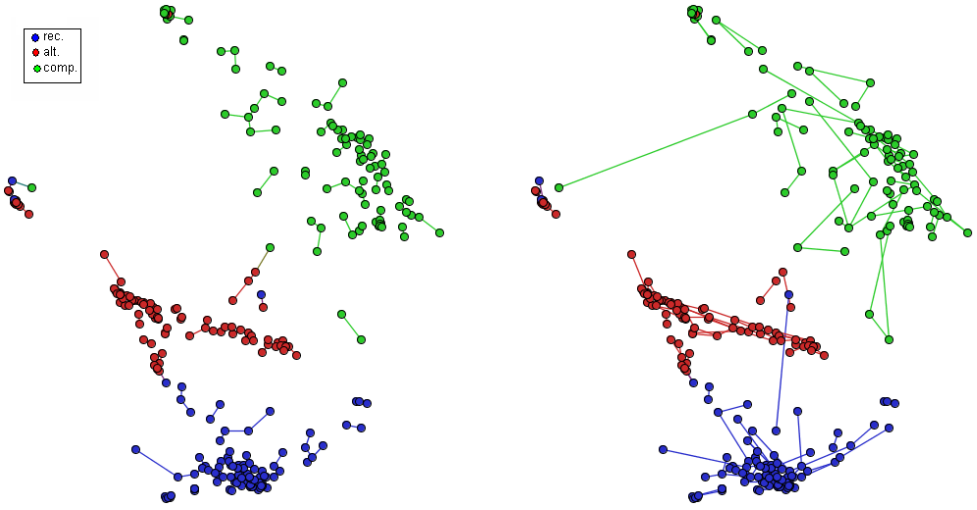
Dos mapas gerados por ambas as técnicas e em relação à pre-classificação dos dados, podemos supor que os pontos no mapa LSP estão melhor posicionados que no mapa gerado pela técnica NJ. Mas um exame em relação às conexões em $\mathbb{R}^n$, como mostrado na Figura 5.6(b) e na Figura 5.7(b), mostra que NJ teve maior grau de consistência em relação ao espaço original que a técnica LSP, porque existe menos cruzamento entre arestas que ligam os nós de grupos diferentes.



(a) Representação da construção árvore filogenética.

(b) Relações de vizinhanças em $\mathbb{R}^n$.

Figura 5.6: Mapa gerado pela técnica NJ para o conjunto *news* no PEx.



(a) Relações de vizinhança em $\mathbb{R}^2$.

(b) Relações de vizinhanças em $\mathbb{R}^n$.

Figura 5.7: Mapa gerado pela técnica de projeção LSP para o conjunto *news* no PEx.

Análises detalhadas de ambas as técnicas foram feitas sobre outro mapa gerado a partir de um conjunto de dados mais complexo e homogêneo, no qual os dados lidam com conceitos e técnicas semelhantes e a classificação em forma manual foi difícil de se realizar. Nos referimos ao conjunto de textos infovis2004, composto por 515 artigos que foram publicados durante 10 anos na Conferência IEEE *Information Visualization*. O mapa resultante para a técnica NJ é apresentado na Figura 5.8 e para projeções na Figura 5.9.

Para validar uma vez mais o desempenho da técnica NJ em relação às técnicas de projeções, os nós serão coloridos pela freqüência de ocorrência dos termos no mapa. Na região **(A)** da Figura 5.8 os nós estão coloridos considerando a ocorrência da expresão termo "*image retrieval*". Pode-se notar que os artigos que tratam desse assunto estão localizados em um mesmo ramo quase sem ramificações. Os nós que estão no extremo do ramo guardam relações por referência e por ano de publicação e, dessa forma, as relações de similaridade entre eles são muito altas. Já para o caso de projeções, região **(A)** da Figura 5.9, não fica clara a relação de vizinhança entre os artigos, embora pareçam estar localizados razoavelmente perto. Um segundo exemplo é mostrado na região **(B)** baseado na busca de ocorrências com o termo "*web*". Observamos que os artigos na técnica NJ foram posicionados no mesmo ramo e tratam de um tema em comum: visualização de dados recuperados de sítios *web*. De novo é mostrado o alto grau de similaridade dos artigos que pertencem ao mesmo ramo. Na projeção, os artigos ficaram mais dispersos na região **(B)** (Figura 5.9). Com o intuito de procurar informação sobre áreas mais gerais, aplicamos a funcionalidade do PEx de criar rótulos por grupos usando o algoritmo de agrupamento *k-means*. Conforme a informação obtida dos agrupamentos gerados e seus respectivos rótulos procuramos os artigos que possuem como área de interesse "*human interface*", assunto amplamente abordado em visualização de informação, e obtivemos como resultado que esse tema se encontram em maior freqüência no ramo da região **(C)**. No caso de projeções, esses artigos foram posicionados um sobre os outros em uma área densa, onde em primeira instância não se pode inferir as relações existentes entre os mesmos (região **(C)** em projeções). Funcionalidades de ampliação ou corte deveria ser usadas para explorar essas áreas.

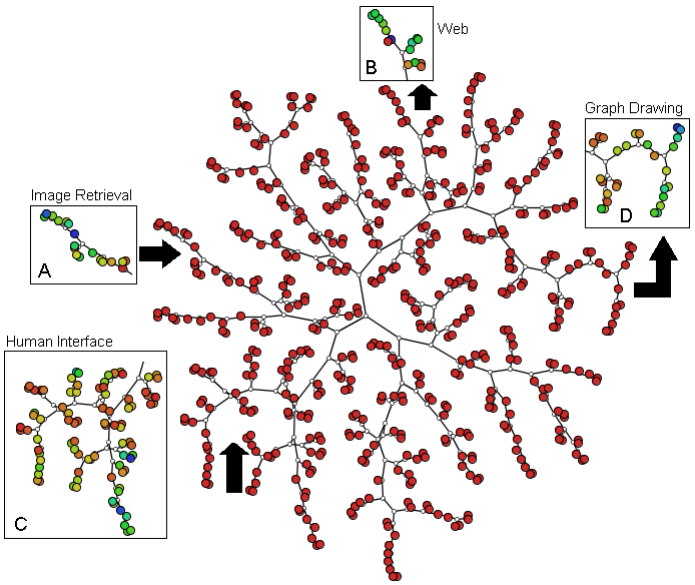


Figura 5.8: Construção do mapa de textos InfoVis com a técnica NJ.

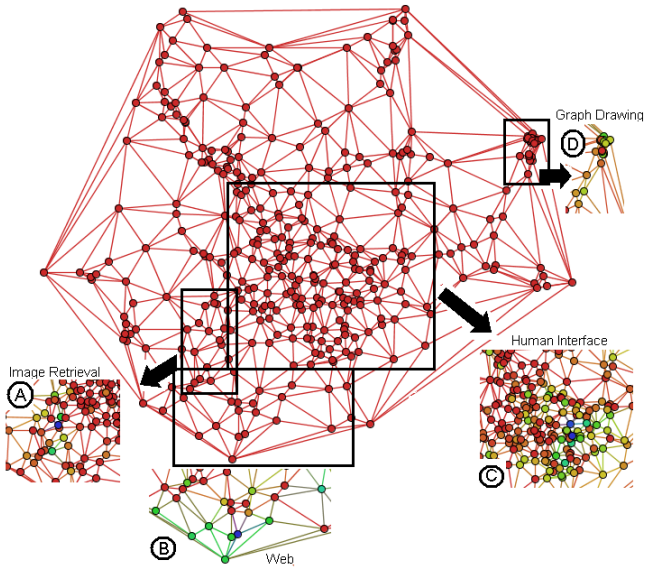


Figura 5.9: Construção do mapa de textos InfoVis com a técnica de projeção LSP.

A seguir, exploramos as regiões densas das projeções e analisamos como essas se refletem na árvore. Um exemplo é mostrado na região **(D)** da Figura 5.9, na qual os pontos são coloridos pela ocorrência dos termos "graph drawing", mostrando claramente que existe um agrupamento bem definido na projeção. Esse mesmo agrupamento foi coordenado na árvore e os termos foram posicionados no ramo da região (D) da NJ; outra vez a árvore posiciona os pontos em um mesmo ramo, além de proporcionar maior conhecimento das relações existentes entre eles.

Conseqüentemente conclui-se que filogenia adiciona maior informação sobre o conjunto de textos que as projeções, não só porque permite aproveitar a relação hierárquica inerente das árvores, mas também porque permite obter uma visão global e local dos dados, tornando mais clara a percepção das relações de similaridade entre textos ao refinar regiões densas existentes nas projeções. Também é demonstrado que textos similares pertencem ao mesmo ramo e que os textos que possuem um alto grau de similaridade ficam nos extremos dos mesmos. Se um ramo apresenta ramificações significa que dentro de uma área geral existem áreas específicas. Já as projeções conseguem delimitar claramente regiões separadas por conteúdo, permitindo definir possíveis agrupamentos.

Outra pauta que marca a comparação visual entre a técnica NJ e projeções é o posicionamento dos pontos no plano, que será tratado a seguir.

5.3.2 Comparação entre NJ e projeções relativas ao posicionamento dos pontos no mapa de visualização

Apesar das técnicas de projeção e NJ terem aspectos em comum que as relaciona, as duas técnicas são distintas no modo como os dados são interpretados. O cálculo de precisão de um ponto referente a seu posicionamento em vizinhanças de conteúdo similar também difere.

No caso das projeções, a distribuição dos pontos no plano depende da direção em que eles são projetados, isto é, em que eixo do espaço projetado os pontos conseguem manter alta correlação com o intuito de conseguir que eles pertençam a vizinhanças adequadas e permitir que o usuário possa deduzir os grupos formados no mapa. No caso de árvores filogenéticas as relações de vizinhanças ficam determinadas pelos ramos e embora a posição destes possa ficar invertida, a relação de similaridade entre as folhas ainda permanece igual, permitindo ao usuário memorizar as relações existentes entre os nós.

Então, uma posibilidade para o cálculo da precisão do *layout* gerado por ambas as técnicas é tomar os k vizinhos mais próximos de um ponto no plano e conferir que proporção deles pertence à mesma classe. O cálculo desses vizinhos é diferente para cada técnica. No caso de projeções, o cálculo é realizado em função da distância entre os pontos no plano de projeção (Paulovich *et al.*, 2006). Para árvores filogenéticas, a precisão é determinada pela trajetória que um nó segue durante seu processo evolutivo enquanto a árvore é formada. Especificamente, a trajetória de um nó é composta por seus nós ancestrais e por seus nós descendentes, relações que estão refletidas nos ramos da árvore. Portanto, os k-vizinhos de um nó são definidos pelos nós que participam em seu processo de evolução e o cálculo é definido pela porcentagem dos k nós vizinhos que pertencem à mesma classe do nó consultado.

A precisão dos nós da árvore filogenética NJ em relação a classe a que pertencem é determinada em duas fases: percorrendo a árvore segundo o critério de vizinhança para NJ e calculando quantos desses vizinhos pertencem à mesma classe.

O trajeto da árvore é realizado respeitando a ordem em que um par de nós folha foram unidos a um nó interno. É importante destacar que cada nó interno X possui grau três, e com a finalidade de respeitar a ordem de criação dos nós, as primeiras duas posições indicam o par de nós vizinhos que dão origem ao nó X e a terceira posição indica o nó com quem X é unido. Assim, o percorrido dos k vizinhos mais próximos a um nó v é realizado como segue:

1. Visitar o nó ancestral X imediato de v.
2. Caminhar por todos os nós descendentes do primeiro vizinho mais próximo de X.
3. Caminhar por todos os nós descendentes do segundo vizinho mais próximo de X.

4. Caminhar por todos os nós ascendentes do terceiro vizinho mais próximo de X.

A árvore é percorrida até obter a seqüência $(v_1, v_2, ..., v_k)$ dos k nós folhas vizinhas a v. O processo é realizado para todos os nós folhas da árvore.

O cálculo dos k vizinhos é determinado pela porcentagem de nós que que pertencem à mesma classe.

No que diz respeito às projeções, o cálculo da seqüência dos k nós vizinhos de um determinado nó v é realizado aplicando a distância Euclideana de v a todos os nós, segundo uma ordem crescente. O cálculo da porcentagem de precisão é a mesma que foi aplicada para árvores.

Segundo os critérios estabelecidos acima para calcular o posicionamento correto dos pontos no plano de visualização, a Figura 5.10 mostra os resultados de avaliar a precisão dos mapas gerados pela técnica NJ na Figura 5.11(a) comparado com técnicas de projeção IDMAP (b), LSP (c), *Sammon Mapping* (d) e PCA (e) para o mesmo conjunto de dados e aplicando a distância baseada no cálculo do cosseno, tomando como amostra os 30 vizinhos mais próximos.

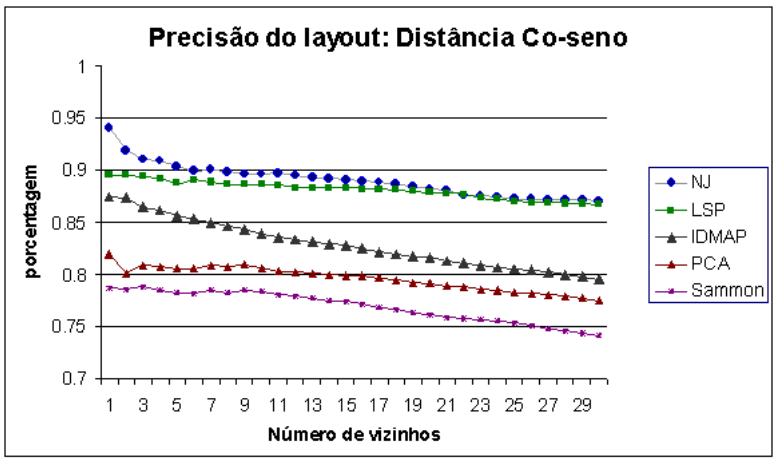


Figura 5.10: Precisão na construção visual do mapa para o conjunto de textos CBR-ILP-IR-SON usando NJ e Projeções, para distâncias baseadas no cosseno.

Para comprovar que a precisão visual dos pontos no *layout* é aplicável não só aos tipos de distância usados na representação vetorial dos textos, mas também em distâncias de cálculo direto, como NCD, foram testados vários conjuntos de dados. A Figura 5.12 mostra o resultado de um deles aplicando a distância NCD sobre o conjunto de dados *All* que reúne um total de 2.684 artigos provenientes dos conjuntos de textos cbr-ilp-ir-son, kdvis e infovis. Os mapas de visualização são mostrados na Figura 5.13, demonstrando que a técnica NJ é capaz de posicionar os pontos em conjuntos de dados heterogêneos e homogêneos independentemente do cálculo da distância.

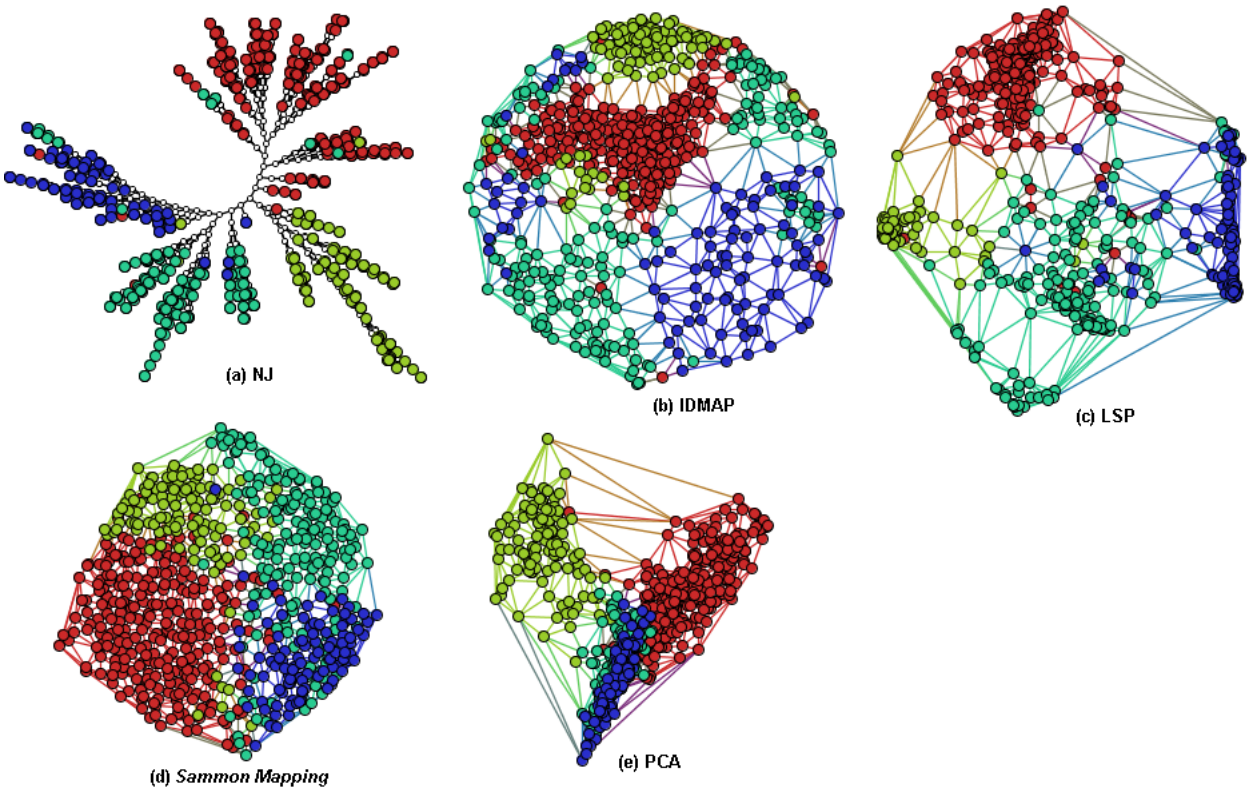


Figura 5.11: Mapas de textos CBR-ILP-IR-SON. Técnica NJ (a), IDMAP (b), LSP (c), Sammon mapping (d) e PCA (e).

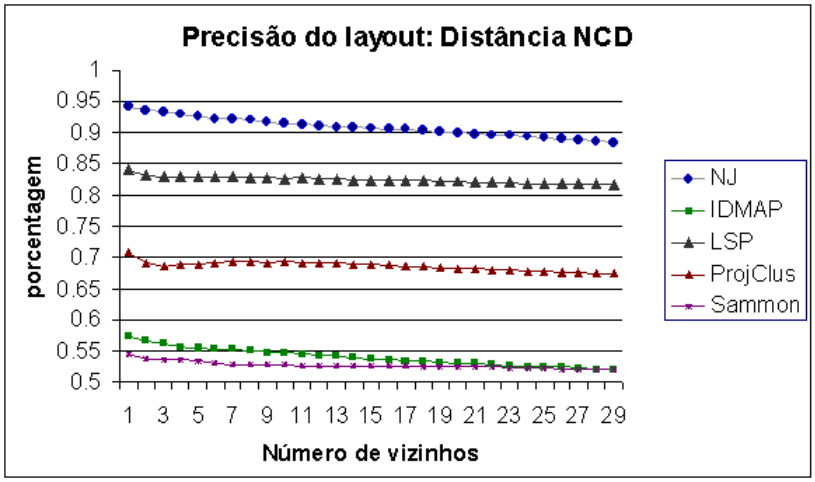


Figura 5.12: Precisão na construção visual do mapa para o conjunto de textos ALL usando NJ e Projeções, para distâncias diretas NCD.

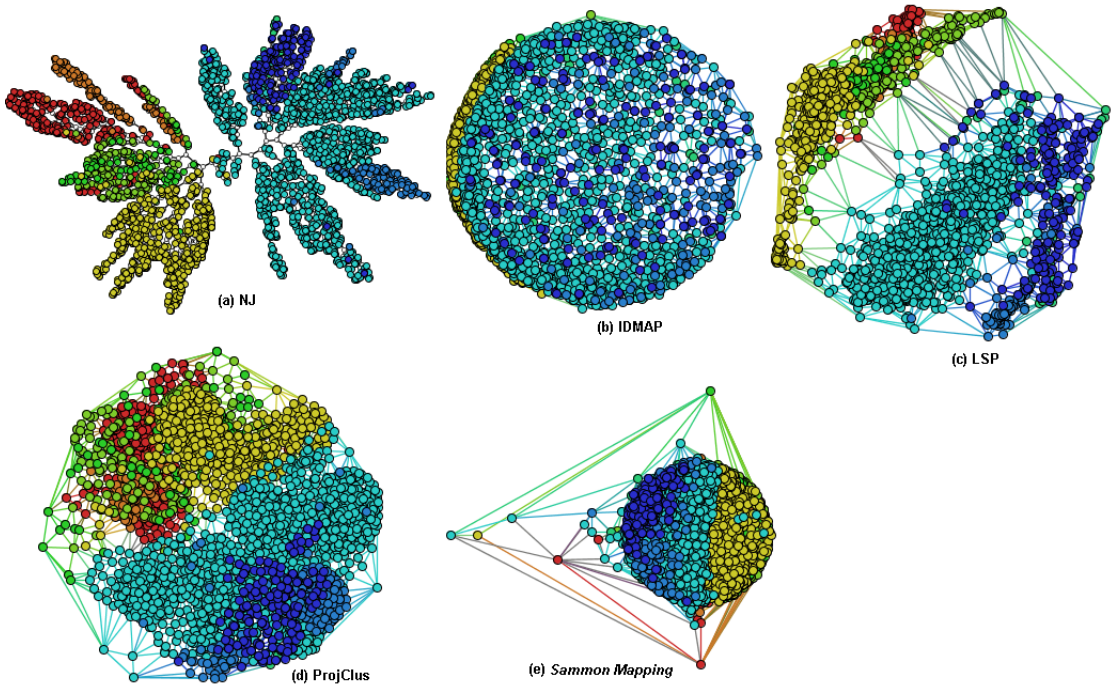


Figura 5.13: Mapas de textos *ALL*, Técnica (a) NJ , (b) IDMAP, (c)LSP, (d) ProjClus (e) Sammon *mapping*.

Esses resultados indicam que NJ possui a maior porcentagem de acerto quando se trata dos primeiros vizinhos mais próximos, o que confirma a alta similaridade dos nós folhas posicionados nos extremos dos ramos. O valor vai decrescendo conforme aumenta o número de vizinhos, possivelmente porque o conteúdo nos nós pode tratar de assuntos menos específicos, mas ainda assim é melhor que algumas das técnicas de projeção. No que se refere às técnicas de projeção, o resultado do cálculo da precisão do *layout* sugere que LSP é mais consistente e estável que as outras técnicas de projeção, mas que em conjunto apresentam uma porcentagem alta, entre 90% e 80%, de acerto para o melhor caso.

Esses testes indicam que tanto a técnica NJ quanto as projeções são capazes de identificar relações de similaridade entre os textos por região e que essas relações são refletidas durante a exploração de informação no mapa de visualização.

A seguir é analisado o tempo de processamento de ambas as técnicas.

5.3.3 Tempo de execução

O grande problema da técnica NJ é sua complexidade computacional ($O(n^3)$), comparada com técnicas de projeção com complexidade menor $O(n^{3/2})$ ou $O(n\sqrt{n})$. Conseqüentemente, a maior diferença entre ambas as técnicas é seu tempo de execução e NJ se torna inapropriado

para grandes conjuntos de dados que precisam ser analizados em tempo curto. A tabela No. 5.2 apresenta o tempo em segundos que a técnica NJ gastou para gerar mapas de textos em um computador Pentium 4 de 3.2 GHz. A medida do tempo é determinada pelo processo de construção da estrutura da árvore e pelo processo de posicionamento dos pontos no *layout*. O tempo de cálculo da matriz de similariades não foi considerado. Já o tempo de processamento para as projeções, como a LSP é inferior a um segundo (Paulovich *et al.*, 2006), (Paulovich *et al.*, 2007b).

Tabela 5.2: Tempo gasto para construir um mapa de textos (em segundos).

Data Set	No. de Dados	NJ	Layout	Total
MESSAGES	300	0,35	0,31	0,66
INFOVIS04	515	1,83	0,45	2,28
CBR+ILP+IR+SON	680	4,55	0,52	5,17
KDVis	1.624	66,20	1,26	67,46
NEWS	2,684	359,63	1,70	361,33
ALL	2,819	454,66	2,17	456,83

5.4 Conclusões

Após da análise dos resultados, concluímos primeiramente que a técnica NJ foi bem sucedida para construir árvores de textos considerando relações de similaridade por conteúdo. Como os testes computacionais indicam, a árvore resultante é capaz separar textos por conteúdo similar, tornando claras as relações visuais entre os textos. Textos similares são colocados nos ramos da árvore e nós colocados nos extremos possuem maior grau de similaridade que nós colocados distantes, próximos ao centro da árvore. A terminologia e as relações vertidas em filogenia, como ascendentes e descendentes, ou pai e filhos, pode ser aplicável em textos. Essas relações estão refletidas no posicionamento dos textos nos ramos, permitindo o usuário inferir relações de hierarquia entre os textos seguindo a trajetória dos mesmos. É importante destacar que os resultados da NJ estão limitados pelo cálculo da matriz de similaridades. Se as distâncias nessa matriz são capazes de gerar boas distinções entre textos por conteúdo, a técnica NJ é capaz de relacionar apropriadamente textos similares.

A mais forte limitação da técnica NJ é seu elevado custo computacional, sobretudo para aplicações em tempo real de conjuntos de textos grandes.

A principal contribuição deste trabalho é a introdução de uma nova técnica visual, no campo da visualização de informação, para explorar conjuntos de textos. Ela é orientada a refletir relações de similaridade baseadas em conteúdo com precisão elevada.

Para validar o desempenho de nossa técnica foram usadas comparações com técnicas de projeção. Muitas dessas técnicas alcançaram um desempenho satisfatório para mapear grandes conjuntos de dados multidimensionais. Mas pela natureza dos mecanismos de processamento que aplicam, elas lidam principalmente com dois problemas: dados que deveriam ser colocados próximos no mapa de visualização, são colocados distantes e dados com alto grau de correlação são sobrepostos prejudicando a distinção individual e em grupos. Em relação a essas desvantagens, a técnica NJ foi comparada com a técnica de projeção LSP e obtivemos os seguintes resultados (ver também (Cuadros *et al.*, 2007)):

- Visualmente, a técnica NJ alternativamente pode partir de uma visão local ou global da árvore. Apoiada na natureza hierárquica da mesma, os nós são agrupados por similaridade em ramos, que podem ser agrupados com seus ancestrais mais similares a outros ramos e assim permitir ao usuário uma interação de contração e expansão. Projeções não refletem muitas vezes uma estrutura clara presente nos textos de uma vizinhança. As relações de vizinhança determinam claramente a existência de possíveis agrupamentos de textos similares. Mas em regiões muito densas é difícil determinar sub-agrupamentos. Portanto, NJ adiciona informação útil sobre o conjunto de textos, refinando regiões densas presentes nas projeções.

- Em relação ao posicionamento dos pontos no plano, embora a porcentagem de acerto em ambas as técnicas seja boa, entre 80% e 95%, NJ é melhor quando se trata dos primeiros vizinhos mais próximos.

- Em relação ao tempo de processamento, a complexidade computacional da técnica NJ é muito elevada ($O(n^3)$) em comparação às técnicas de projeção mais rápidas. Para LSP o custo computacional é $O(n\sqrt{n})$.

A técnica NJ e as técnicas de projeção resumidamente se diferenciam por:

- Projeções usam técnicas geométricas para visualizar mapas de dados, e que em NJ são substituídas pela topologia da árvore.

- NJ reflete relações de similaridade mais precisas que projeções.

- A técnica NJ não precisa da intervenção do usuário para introduzir parâmetros, o que não acontece com projeções.

- Uma vez gerada a matriz de similariades a técnica NJ produz uma única visão topológica da árvore. Projeções podem produzir varias visualizações.

Finalmente, ambas as técnicas são importantes métodos para visualizar e explorar informação, tentando preservar no máximo as relações de similaridades entre os textos como definido no espaço original e formam informações que complementam na interpretação de uma coleção de textos.

A continuidade deste trabalho pode ser realizada em duas direções: a primeira é tratar de melhorar o tempo computacional para execuções rápidas em tempo real e a segunda é melhorar a escalabilidade e estabilidade visual da técnica. Em relação à primeira, existem algoritmos com versões mais rápidas do NJ, testes sobre esses deverão ser feitos para comprovar seu funcionamento com textos. Para a segunda, deveria se implementar uma plataforma especial para explorar e interagir com árvores filogenéticas e adicionar funcionalidades às já existentes no PEx como: layout de expansão da árvore, múltiplas visões, contração e expansão imediata da árvore, entre outras. Uma técnica mista de projeção e árvores de similidades se encontra em estudo.

Referências Bibliográficas

BACHMAIER, C.; BRANDES, U.; SCHLIEPER, B. Drawing phylogenetic trees. In: *Proceedings of Intl. Symp. Algorithms and Computation: ISAAC '05*, 2005, p. 1110–1121.

BECKS, A.; SEELING, C.; MINKENBERG, R. Benefits of document maps for text access in knowledge management: a comparative study. In: *Proceedings of the 2002 ACM symposium on Applied computing: SAC '02*, New York, NY, USA: ACM Press, 2005, p. 621–626.

BERKHIN, P. *Survey of clustering data mining techniques*. Relatório Técnico, Accrue Software, San Jose, CA, 2002.

BINGHAM, J.; SUDARSANAM, S. Visualizing large hierarchical clusters in hyperbolic space. *Bioinformatics*, v. 16, n. 7, p. 660–661, 2000.

BORNER, K.; CHEN, C.; BOYACK, K. Visualizing knowledge domains. *Annual Review of Information Science and Technology*, v. 37, n. 1, 2003.

BOYACK, K.; WYLIE, B.; DAVIDSON, G. Domain visualization using vxinsight for science and technology management. *Journal of the American Society for Information Science and Technology*, v. 53, n. 9, p. 764–774, 2002.

BRYANT, D. On the uniqueness of the selection criterion in neighbor-joining. *Journal of Classification*, v. 22, n. 1, p. 3–15, 2005.

CARRIZO, S. Phylogenetic trees: an information visualisation perspective. In: *Proceedings of the second conference on Asia-Pacific bioinformatics: APBC '04*, Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2004, p. 315–320.

CHEN, C. Generalised similarity analysis and pathfinder network scaling. *Interacting with Computers*, v. 10, n. 2, p. 107–128, 1998.

CHEN, C. *Information visualization: Beyond the horizon*. Springer-Verlag New York, 2004.

CHOI, J.; JUNG, H.; KIM, H.; CHO, H. Phylodraw: a phylogenetic tree drawing system. *Bioinformatics*, v. 16, n. 11, p. 1056–1058, 2000.

COX, T.; COX, M. *Multidimensional scaling*. Second ed. Chapman & Hall/CRC, 2000.

CROW, V.; WISE, J.; THOMAS, J.; LANTRIP, D.; POTTIER, M.; SCHUR, A. Multidimensional visualization and browsing for intelligence analysis. In: *Proceedings of Graphics And Visualization Conference: GVIZ '94.*, WA: Richland, 1994.

CUADROS, A.; PAULOVICH, F.; MINGHIM, R.; TELLES, G. Point placement by phylogenetic trees and its application to visual analysis of document collections (conditionally accepted). IEEE Computer Society Press, 2007.

DEERWESTER, S.; DUMAIS, S.; LANDAUER, T.; FURNAS, G.; HARSHMAN, R. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, v. 41, n. 6, p. 391–407, 1990.

DING, C. A probabilistic model for dimensionality reduction in information retrieval and filtering. In: *Proceedings of 1st SIAM Computational Information Retrieval Workshop*, Raleigh, NC, 2000.

FALOUTSOS, C.; LIN, K. Fastmap: A fast algorithm for indexing, datamining and visualization of traditional and multimedia databases. In: CAREY, M. J.; SCHNEIDER, D. A., eds. *Proceedings of International Conference on Management of Data: ACM SIGMOD*, San Jose, California: ACM Press: New York, 1995, p. 163–174.

FELSENSTEIN, J. *Inferring phylogenies*. Sinauer Associates, 2003.

FRUCHTERMAN, T.; REINGOLD, E. Graph drawing by force-directed placement. *Software - Practice and Experience.*, v. 21, n. 11, p. 1129–1164, 1991.

GENNARI, J.; LANGLEY, P.; FISHER, D. Models of incremental concept formation. *Journal of Artificial Intelligence*, v. 40, n. 1-3, p. 11–61, 1989.

GRANITZER, M.; WOLFGANG, K.; SABOL, S.; ANDREWS, K.; KLIEBER, W. Evaluating a system for interactive exploration of large, hierarchically structured document repositories. In: *Proceedings of the IEEE Symposium on Information Visualization: INFOVIS'04*, Los Alamitos, CA, USA: IEEE Computer Society, 2004, p. 127–134.

GUSFIELD, D. *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge, England: Cambridge University Press, 1997.

HERMAN, I.; MELANÇON, G.; MARSHALL, M. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, v. 6, n. 1, p. 24–43, 2000.

HUANG, S.; WARD, M.; RUNDENSTEINER, E. Exploration of dimensionality reduction for text visualization. In: *Proceedings of the Coordinated and Multiple Views in Exploratory Visualization: CMV'05*, Washington, DC, USA: IEEE Computer Society, 2005, p. 63–74.

JOLLIFFE, I. *Principal component analysis*. Springer-Verlag, 1986.

KEIM, D. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, v. 8, n. 1, p. 1–8, 2002.

KOHONEN, T. Exploration of very large databases by self-organizing maps. In: *Proceedings of the IEEE International Conference on Neural Networks*, 1997, p. 1–6.

LOPES, A.; MINGHIM, R.; MELO, V.; PAULOVICH, F. Mapping texts through dimensionality reduction and visualization techniques for interactive exploration of document collections. In: ERBACHER, R. F.; ROBERTS, J. C.; GRÖHN, M. T.; BORNER, K., eds. *Proceedings of SPIE-IS&T Electronic Imaging - Visualization and Data Analysis 2006*, San Jose, California: SPIE, 2006, p. 60600T.

LOPES, A.; PINHO, R.; F.PAULOVICH; MINGHIM, R. Visual text mining using association rules. *Computers & Graphics*, v. 31, n. 3, p. 316–326, 2007.

LUHN, H. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, v. 2, n. 2, p. 159–165, 1968.

MINGHIM, R.; PAULOVICH, F.; LOPES, A. Content-based text mapping using multi-dimensional projections for exploration of document collections. In: ERBACHER, R. F.; ROBERTS, J. C.; GRÖHN, M. T.; BORNER, K., eds. *Proceedings of SPIE-IS&T Electronic Imaging Visualization and Data Analysis 2006*, San Jose, California: SPIE, 2006, p. 60600S.

MUNZNER, T.; GUIMBRETIRE, F.; TASIRAN, S.; ZHANG, L.; ZHOU, Y. Treejuxtaposer: scalable tree comparison using focus+context with guaranteed visibility. *ACM Transactions on Graphics*, v. 22, n. 3, p. 453–462, 2003.

PAGE, R. Treeview : An application to display phylogenetic trees on personal computers. *Computer Applications in the Biosciences*, v. 12, p. 357–358, 1996.

PAULOVICH, F.; MINGHIM, R. *Técnicas geométricas para análise visual de dados - integrando mineração e visualizações*. Exame de qualificação, ICMC-USP, 2006a.

PAULOVICH, F.; MINGHIM, R. Text map explorer: a tool to create and explore document maps. In: *Proceedings of the conference on Information Visualization: IV '06*, Washington, DC, USA: IEEE Computer Society, 2006b, p. 245–251.

PAULOVICH, F.; NONATO, G.; MINGHIM, R.; LEVKOWITZ, H. Least square projection: a fast high precision multidimensional projection technique and its application to document mapping. *IEEE Transactions on Visualization and Computer Graphics (accept for publication)*, 2007a.

PAULOVICH, F.; NONATO, L.; MINGHIM, R. Visual mapping of text collections through a fast high precision projection technique. In: *Proceedings of the conference on Information Visualization: IV '06*, Washington, DC, USA: IEEE Computer Society, 2006, p. 282–290.

PAULOVICH, F.; DE OLIVEIRA, M.; MINGHIM, R. The projection explorer: A flexible tool for projection-based multidimensional visualization. In: *Proceedings of XX Brazilian Symposium on Computer Graphics and Image Processing – SIBGRAPI 2007 (to appear)*, Belo Horizonte, Brazil: IEEE Computer Society Press, 2007b.

PENNOCK, K.; LANTRIP, D. A landscape representation of themes in text. In: *Proceedings of the Symposium on Advanced Intelligence Processing and Analysis*, Washington DC: Office of Research and Development: In AIPA Steering Group (Eds.), 1995, p. 47.

PINHO, R.; DE OLIVEIRA, M. *Mineração visual de conjuntos de documentos: regras de associação e identificação de tópicos em mapas*. Exame de qualificação, ICMC-USP, 2006.

PORTER, M. An algorithm for suffix striping. *Program*, v. 14, n. 3, p. 130–137, 1980.

RASMUSSEN, M.; KARYPIS, G. *gcluto: an interactive clustering, visualization, and analysis system*. Technical Report CSE/UMN TR 04-021, Univ. of Minnesota, Dep. of Computer Science and Engineering, 2004.

REID, E.; CHEN, H. Mapping the contemporary terrorism research domain. Duluth, MN, USA: Academic Press, Inc., 2007, p. 42–56.

ROHRER, R.; SIBERT, J.; EBERT, D. The shape of shakespeare: Visualizing text using implicit surfaces. In: *Proceedings of the 1998 IEEE Symposium on Information Visualization: INFOVIS '98*, Washington, DC, USA: IEEE Computer Society, 1998, p. 121–129.

ROST, U.; BORNBERG-BAUER, E. Treewiz: interactive exploration of huge trees. *Bioinformatics*, v. 18, n. 1, p. 109–114, 2002.

SAITOU, N.; NEI, M. The neighbor joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, v. 4, n. 4, p. 406–425, 1987.

SALEMI, M.; VANDAMME, A. *The phylogenetic handbook. a practical approach to dna and protein phylogeny*. Cambridge: Cambridge University Press, 2003.

SALTON, G.; WONG, A.; YANG, C. S. A vector space model for automatic indexing. *Commun. ACM*, v. 18, n. 11, p. 613–620, 1975.

SAMMON, J. A nonlinear mapping for data structure analysis. v. C-18, n. 5, p. 401–409, 1969.

SETUBAL, J.; MEIDANIS, J. *Introduction to computational molecular biology*. PWS Publishing Co., 1997.

SHNEIDERMAN, B. The eyes have it: A task by data type taxonomy for information visualizations. In: *Proceedings of the 1996 IEEE Symposium on Visual Languages: VL '96*, Washington, DC, USA: IEEE Computer Society, 1996, p. 336.

SPENCE, R. *Information visualization*. Addison Wesley, 2001.

SWOFFORD, D.; SULLIVAN, J. Phylogeny inference based on parsimony and other methods using PAUP*. Theory and practice. In: SALEMI, M.; VANDAMME, A. M., eds. *The phylogenetic handbook. A practical approach to DNA and protein phylogeny*, UK: Cambridge University Press, p. 160–206, 2003.

TAN, P.; STEINBACH, M.; KUMAR, V. *Introduction to data mining*. Addison-Wesley, 2006.

TEJADA, E.; MINGHIM, R.; NONATO, L. On improved projection techniques to support visual exploration of multidimensional data sets. *Information Visualization Journal*, v. 2, n. 4, p. 218–231, 2003.

TELLES, G.; MINGHIM, R.; PAULOVICH, F. Normalized compression distance for visual analysis of document collections. *Computer & Graphics, Special Issue on Visual Analytics*, v. 31, n. 3, p. 327–337, 2007.

WEIPPL, E. Visualizing content based relations in texts. In: *Proceedings of the 2nd Australasian conference on User interface: AUIC '01*, Washington, DC, USA: IEEE Computer Society, 2001, p. 34–41.

WEISS, S.; INDURKHYA, N.; ZHANG, T.; DAMERAU, F. *Text Mining: predictive methods for analyzing unstructured information*. Springer, 2004.

WISE, J. The ecological approach to text visualization. *Journal of the American Society for Information Science*, v. 50, n. 13, p. 1224–1233, 1999.

ZMASEK, C.; EDDY, S. Atv: display and manipulation of annotated phylogenetic trees. *Bioinformatics*, v. 17, n. 4, p. 383–384, 2001.