CAPÍTULO
10

Visualização de Texto e Documentos

Dispomos agora de enormes recursos de informação;de bibliotecas a arquivos de e-mail, a todas as facetas de aplicativos executados na World Wide Web.A visualização é uma grande ajuda na análise desses dados.Podemos visualizar coisas de muitas maneiras diferentes, como um blog, um wiki, um feed do Twitter, bilhões de palavras, uma coleção de artigos ou uma biblioteca digital.Como as visualizações dependem de tarefas, podemos ver quais tarefas são necessárias para lidar com texto, documentos ou objetos baseados na web.Para textos e documentos, as tarefas mais óbvias são

pesquisando uma palavra, frase ou tópico.Para dados parcialmente estruturados, podemos procurar relações entre palavras, frases, tópicos ou documentos.Para coleções estruturadas de textos ou documentos, a principal tarefa geralmente é procurar padrões e valores discrepantes no texto ou nos documentos.

Neste capítulo, nos concentramos nas tarefas de visualização relacionadas ao texto e nas diversas abordagens para a análise visual do texto.

10.1 Introdução

Definimos uma coleção de documentos como um corpus (corpora plural).Lidamos com objetos dentro de corpora.Esses objetos podem ser palavras, frases, parágrafos, documentos ou até mesmo coleções de documentos.Podemos até considerar imagens e vídeos.Freqüentemente, esses objetos são considerados atômicos no que diz respeito à tarefa, análise e visualização.Textos e documentos são muitas vezes minimamente estruturados e podem ser ricos em atributos e metadados, especialmente quando focados em um domínio de aplicação específico.Por exemplo, os documentos têm um formato e muitas vezes incluem metadados sobre o documento (ou seja, autor, data de criação, data de modificação, comentários, tamanho).Sistemas de recuperação de informação são usados para consultar corpora, o que requer calcular a relevância de

um documento em relação a uma consulta.Isto requer pré-processamento de documentos e interpretação da semântica do texto.

Podemos calcular estatísticas sobre documentos.Por exemplo, o número de palavras ou parágrafos, ou a distribuição ou frequência das palavras, podem ser usados para autenticidade do autor.Existem parágrafos que repetem as mesmas palavras ou frases?Também podemos identificar relações entre parágrafos ou documentos dentro de um corpus.Por exemplo, poder-se-ia perguntar: “Que documentos se relacionam com a propagação da gripe?”Esta não é uma consulta simples;não é simplesmente

procurando a palavra gripe.Por que?Podemos ainda procurar conexões naturais

ou relações entre vários documentos.Que clusters existem?Eles representam temas dentro do corpus?A similaridade pode ser definida em

termos de citações, autorias comuns, tópicos e assim por diante.

10.2 Níveis de representações de texto

Definimos três níveis de representação de texto: lexical, sintático e semântico.Cada um exige que convertamos o texto não estruturado em alguma forma de dados estruturados.

Nível lexical.O nível lexical preocupa-se em transformar uma sequência de caracteres em uma sequência de entidades atômicas, chamadas tokens.Os analisadores lexicais processam a sequência de caracteres com um determinado conjunto de regras em uma nova sequência de tokens que pode ser usada para análise posterior.Os tokens podem incluir caracteres, n-gramas de caracteres, palavras, radicais de palavras, lexemas, frases ou n-gramas de palavras, todos com atributos associados.Muitos tipos de regras podem ser usados para extrair tokens, sendo os mais comuns máquinas de estados finitos definidas por expressões regulares.

Nível sintático.O nível sintático trata da identificação e marcação (anotação) da função de cada token.Atribuímos várias tags, como posição da frase ou se uma palavra é um substantivo, palavrão, adjetivo, modificador pendente ou conjunção.Os tokens também podem ter atributos como serem singulares ou plurais ou sua proximidade com outros tokens.Tags mais ricas incluem data, dinheiro, local, pessoa, organização e hora (Figura 10.3).O processo de extração dessas anotações é chamado

reconhecimento de entidade nomeada (NER).A riqueza e a grande variedade de modelos de linguagem e gramáticas (generativas, categóricas, de dependência, probabilísticas e funcionalistas) produzem uma ampla variedade de abordagens.

Nível semântico.O nível semântico abrange a extração de significado e relações entre conhecimentos derivados da estrutura.

identificadas no nível sintático.O objetivo deste nível é definir uma interpretação analítica do texto completo dentro de um contexto específico, ou mesmo independente do contexto.

10.3 O modelo de espaço vetorial

A computação de vetores de termos é uma etapa essencial para muitas técnicas de visualização e análise de documentos e corpus.No modelo de espaço vetorial [356], um vetor de termo para um objeto de interesse (parágrafo, documento ou coleção de documentos) é um vetor em que cada dimensão representa o peso de uma determinada palavra naquele documento.Normalmente, para limpar o ruído, palavras irrelevantes (como “o” ou “a”) são removidas (filtragem) e palavras que compartilham um radical de palavra são agregadas (lematização) [192].

O pseudocódigo abaixo conta ocorrências de tokens únicos, excluindo pare as palavras.Supõe-se que a entrada seja um fluxo de tokens gerados por um analisador léxico para um único documento.A variável de termos contém um hashtable que mapeia termos exclusivos para suas contagens no documento.

```
Count-Terms(tokenStream) 1 termos ← ∅ inicializa termos em uma tabela hash vazia.
2   para cada token t em tokenStream
3       faça se t não for uma palavra de parada
4           incremente (ou inicialize para 1) termos[t]
5   termos de devolução
```

Podemos aplicar o pseudocódigo ao texto a seguir.

```
Há muita controvérsia sobre a segurança dos alimentos geneticamente
modificados.Os defensores da biotecnologia dizem frequentemente que os
riscos são exagerados.''Houve 25.000 testes de testes geneticamente
culturas modificadas no mundo, agora, e nem um único incidente, ou
qualquer coisa perigosa nestas libertações'', disse um porta-voz da
Adventa Holdings, uma empresa de biotecnologia do Reino Unido.Durante a
campanha presidencial de 2000, o então candidato George W. Bush disse que
“estudo após estudo não mostrou nenhuma evidência de perigo”. E o
secretário de Agricultura do governo Clinton, Dan Glickman, disse que
“teste após teste científico rigoroso” havia provado o segurança dos
produtos geneticamente modificados.
```

O parágrafo contém 98 tokens de string, 74 termos e 48 termos quando as palavras irrelevantes são removidas.Aqui está um exemplo do termo vetor que seria gerado pelo pseudocódigo:

genetically	said	safety	engineered	study	test	great	deal	controversy	foods
3	3	2	2	2	2	1	1	1	1

10.3.1 Calculando Pesos

Este modelo de espaço vetorial requer um esquema de ponderação para atribuir pesos aos termos de um documento.Existem muitos desses métodos, o mais conhecido dos quais é o termo frequência inversa de frequência de documento (tf-idf) [355].Seja Tf (w) a frequência do termo ou número de vezes que a palavra w ocorreu no documento, e seja Df (w) a frequência do documento (número de documentos que contêm a palavra).Seja N o número de documentos.Nós definimos Tf Idf (w) como

$$\text{Tf Idf (w)} = \text{Tf (w)} * \log \left(\frac{N}{\text{Df (w)}} \right).$$

Esta é a importância relativa da palavra no documento, que corresponde à nossa visão intuitiva da importância das palavras.Uma palavra é mais importante quanto menos documentos ele aparecer (menor Df), bem como se aparecer várias vezes em um único documento de destino (maior Tf).Dito de outra forma, estamos mais interessados em palavras que aparecem frequentemente em um documento, mas não frequentemente na coleção.Tais palavras são intuitivamente mais importantes, pois diferenciam, separam ou classificam palavras.A Figura 10.1 mostra vetores de termos para um grupo de documentos usando pesos tf-idf.

id	men	entered	bank	charlotte	missiles	masks	aryan	guns	witnesses	reported	silver	suv	august
seg1.txt	0.239441	0	0.153457	0.195243	0	0.237029	0	0.195243	0.237029	0.140004	0.195243	0.237029	0
seg13.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg14.txt	0	0.192197	0	0	0	0	0	0	0	0	0	0	0.172681
seg15.txt	0	0	0	0	0	0	0	0	0	0	0	0	0.149652
seg16.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg17.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg18.txt	0	0.158432	0	0	0	0	0	0	0	0	0	0	0
seg19.txt	0	0	0	0.197255	0	0	0	0	0	0.141447	0	0	0.155038
seg2.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg20.txt	0	0.234323	0	0	0	0	0	0	0	0	0	0	0
seg21.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg22.txt	0	0	0	0	0.139629	0	0.127389	0	0	0	0	0	0
seg23.txt	0	0	0	0	0	0	0	0	0	0.180656	0	0	0
seg24.txt	0	0	0	0	0	0	0.117966	0	0	0.117966	0	0	0
seg25.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg26.txt	0	0	0	0	0	0	0	0	0	0	0	0	0
seg27.txt	0	0	0.235418	0	0	0	0.214781	0	0	0	0	0	0
seg28.txt	0	0	0	0	0.151753	0	0	0	0	0	0	0	0
seg29.txt	0	0	0	0	0	0	0.129852	0	0	0	0	0	0.142329
seg3.txt	0	0	0	0	0.18432	0	0	0	0	0	0	0	0
seg30.txt	0.078262	0	0	0	0	0	0	0	0	0	0	0	0
seg31.txt	0	0	0.213409	0	0	0	0.194701	0	0	0	0	0	0
seg32.txt	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 10.1. Uma ilustração de vetores de termos para muitos documentos, contendo seus valores tf-idf.

O pseudocódigo a seguir calcula vetores tf-idf para cada documento em uma determinada coleção de documentos.Ele usa a função Count-Terms no exemplo de pseudocódigo anterior.A primeira seção percorre todos os documentos, computando e armazenando frequências de termos e frequências de documentos.A segunda seção calcula os vetores tf-idf para cada documento e os armazena em uma tabela.

```
Compute-TfIdf(documents) 1 termF requências ← ∅ Procura tabelas de contagem
de termos para nomes de documentos.
2  reqüências documentF ← ∅ Conta os documentos nos quais ocorre um termo.
3  uniqueT erms ← ∅ A lista de todos os termos exclusivos.
4  para cada documento d em documentos
5          do docN ame ← Name(d) Extrai o nome do documento.
6              tokenStream ← Tokenize(d) Gera fluxo de token de documento.
7              termos ← Count-Terms(tokenStream) Conta as frequências dos termos.
8              termF requências[docN ame] ← termos Armazena as frequências dos termos.
9              para cada termo t em chaves (termos)
10                 incremente (ou inicialize para 1) documentF requencies[t]
11                     termos únicos ← termos únicos ∪ t
12
13 tf Idf V ectorTable ← ∅ Procura vetores tf-idf para nomes de documentos.
14 n ← Comprimento(documentos)
15 para cada nome de documento docN ame em chaves (requências termF)
16         faça tf Idf Vector ← crie uma matriz zerada de comprimento Length (uniqueT erms)
17             termos ← termoF frequências[docN ame]
18             para cada termo t em chaves (termos)
19                 faça tf ← termos[t]
20                     df ← requisitos do documentoF[t]
21                     tf Idf ← tf ∗ log(n/df )
22                     vetor tf Idf [índice de t em termos únicos] ← tf Idf
23             tf Idf V ectorTable[docN ame] ← tf Idf V ector
24 retornar tf Idf V ectorT capaz
```

10.3.2 Lei de Zipf

As distribuições normal e uniforme são as que estamos mais familiarizados.A distribuição da lei de potência é comum hoje em dia com os grandes tamanhos de dados que encontramos, que refletem fenômenos escaláveis.O economista Vilfredo Pareto afirmou que a receita de uma empresa é inversamente proporcional à sua posição – uma lei de potência clássica, que resulta na famosa regra 80-20, em que 20% da população detém 80% da riqueza.

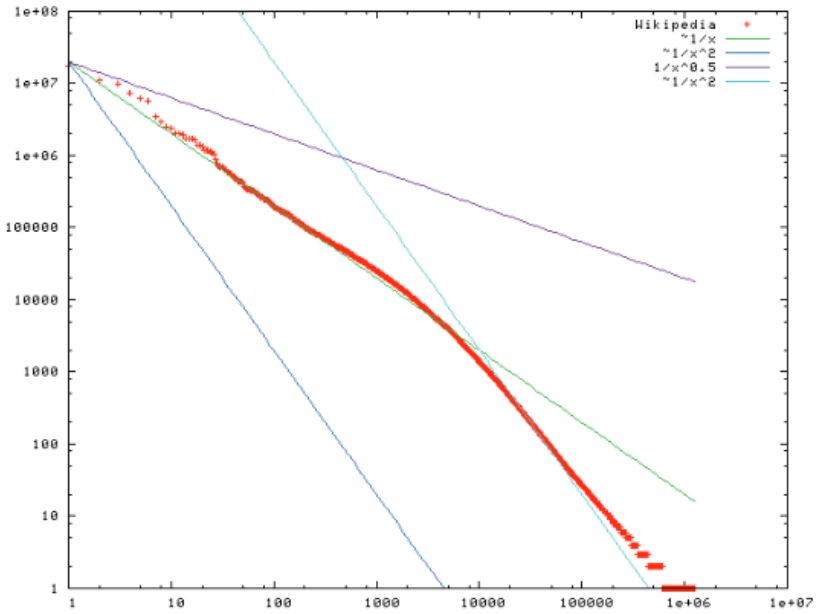

Figura 10.2. A distribuição de termos na Wikipedia, um exemplo da Lei de Zipf em ação.A frequência do termo está no eixo y e a classificação da frequência está no eixo x.

O lingüista de Harvard, George Kingsley Zipf, declarou a distribuição de palavras em corpora de linguagem natural usando uma distribuição discreta de lei de potência chamada distribuição Zipfiana.A Lei de Zipf [490] afirma que em um ambiente natural típico documento linguístico, a frequência de qualquer palavra é inversamente proporcional a sua posição na tabela de frequência.Traçar a curva Zipf em uma escala log-log produz uma linha reta com inclinação de -1 (ver Figura 10.2).

Uma implicação imediata da Lei de Zipf é que um pequeno número de palavras descreve a maioria dos conceitos-chave em pequenos documentos.Existem inúmeros exemplos de resumo de texto que permitem uma descrição completa com apenas algumas palavras.

10.3.3 Tarefas usando o modelo de espaço vetorial

O modelo de espaço vetorial, quando acompanhado de alguma métrica de distância, permite um para realizar muitas tarefas úteis.Podemos usar tf-idf e o modelo de espaço vetorial para identificar documentos de interesse particular.Por exemplo, o modelo de espaço vetorial, com o uso de alguma métrica de distância, nos permitirá responder

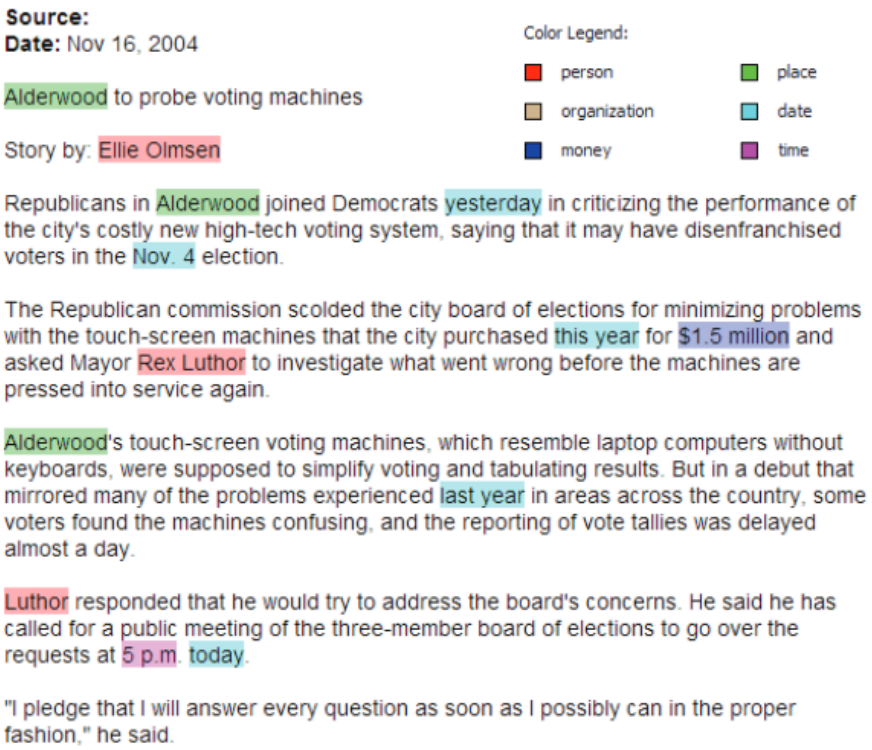

Figura 10.3. Uma visualização de documento na qual as entidades nomeadas são destacadas, codificadas por cores por tipo de entidade.

questões como quais documentos são semelhantes a um documento específico, quais documentos são relevantes para uma determinada coleção de documentos ou quais documentos são mais relevantes para uma determinada consulta de pesquisa – tudo isso encontrando os documentos cujos vetores de termos são mais semelhantes ao determinado documento, o vetor médio de uma coleção de documentos ou o vetor de uma consulta de pesquisa.

Outra tarefa indireta é como ajudar o usuário a entender um todo

corpus.O usuário pode estar procurando padrões ou estruturas, como os temas principais de um documento, clusters e a distribuição de temas por meio de uma coleção de documentos.Isso geralmente envolve a visualização do corpus em um layout bidimensional ou a apresentação ao usuário de um gráfico de conexões entre documentos ou entidades para navegar.O pipeline de visualização mapeia bem a visualização de documentos: obtemos os dados (corpus), transformamos-os em vetores, depois executamos algoritmos baseados nas tarefas de interesse (ou seja, similaridade, pesquisa, agrupamento) e geramos as visualizações.

10.4 Visualizações de documentos únicos

Aqui apresentamos diversas visualizações de um único documento de texto, retirado do conjunto de dados do VAST Contest 2007.

Figura 10.4. Uma visualização de nuvem de tags gerada pelo serviço gratuito tagCrowd.com [396].O tamanho da fonte e o escurecimento são proporcionais à frequência da palavra no documento.

10.4.1 Nuvens de palavras

Nuvens de palavras (Figura 10.4), também conhecidas como nuvens de texto ou nuvens de tags, são layouts de tokens brutos, coloridos e dimensionados de acordo com sua frequência em um único documento.Nuvens de texto e suas variações, como Wordle (Figura 10.5), são exemplos de visualizações que usam apenas vetores de frequência de termos e algum algoritmo de layout para criar a visualização.

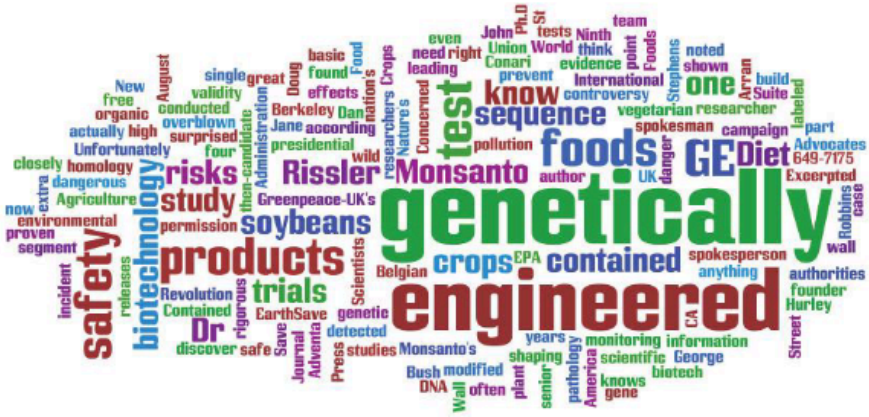

Figura 10.5. Uma visualização Wordle gerada pelo serviço gratuito wordle.net [118].O tamanho do texto corresponde à frequência da palavra no documento.

10.4.2 Árvore de palavras

A visualização WordTree [450] é uma representação visual de ambas as frequências dos termos, bem como de seu contexto (Figura 10.6).O tamanho é usado para representar a frequência do termo ou frase.A raiz da árvore é uma palavra ou frase de interesse especificada pelo usuário, e os ramos representam os vários contextos nos quais a palavra ou frase é usada no documento.

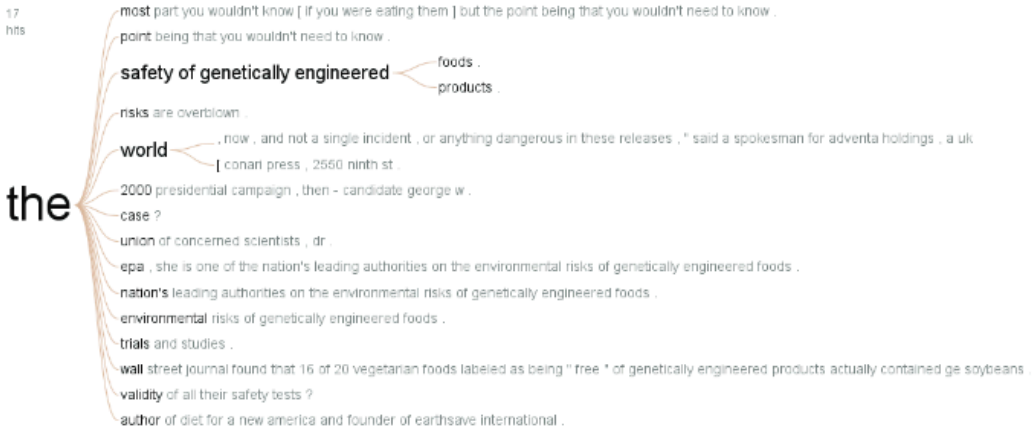

Figura 10.6. Uma visualização WordTree gerada pelo serviço gratuito ManyEyes [196]. O os ramos da árvore representam os vários contextos que seguem uma palavra ou frase raiz no documento.

10.4.3 Arco de texto

Podemos estender a representação da distribuição de palavras exibindo a conectividade.Existem várias maneiras pelas quais as conexões podem ser calculadas.

TextArc [312] é uma representação visual de como os termos se relacionam com as linhas de texto em que aparecem (Figura 10.7).Cada palavra do texto é desenhada em ordem em torno de uma elipse como pequenas linhas com um ligeiro deslocamento no início.Como em uma nuvem de texto, as palavras que ocorrem com mais frequência são desenhadas em tamanho maior e mais claro.Palavras com frequências mais altas são desenhadas dentro da elipse, puxadas por suas ocorrências no círculo (semelhante ao RadViz).O usuário é capaz de destacar o texto subjacente com sondagem e animar a “leitura” do texto, visualizando o fluxo do texto por meio de termos conectados relevantes.

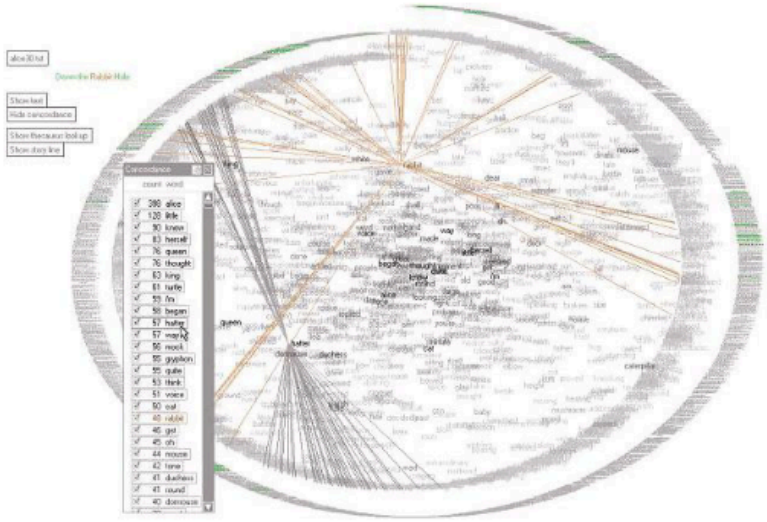

Figura 10.7. Uma visualização TextArc que usa o texto completo de Alice no País das Maravilhas. Palavras que ocorrem uniformemente em todo o documento são posicionados no centro da tela, enquanto as palavras que aparecem apenas em seções específicas estão localizadas mais próximas da circunferência.(Imagem de [312], c© 2002 IEEE.)

10.4.4 Diagramas de Arco

Diagramas de arco [451] são uma visualização focada em exibir repetições em texto ou qualquer sequência.Subsequências repetidas são identificadas e conectadas por arcos semicirculares.A espessura dos arcos representa o comprimento da subsequência e a altura dos arcos representa a distância entre as subsequências.A Figura 10.8 mostra o Minueto em Sol Maior de Bach, visualizando

Figura 10.8. Uma visualização do diagrama de arco do Minueto em Sol Maior de Bach.Sequências repetidas são conectadas por arcos semicirculares.(Imagem de [451], c© 2002 IEEE.)

o padrão clássico de um minueto.Contém duas partes, cada uma composta por um

passagem longa tocada duas vezes.As partes estão frouxamente relacionadas, como mostra o feixe de arcos finos que conectam as duas partes principais.A sobreposição dos dois arcos principais mostra que o final da primeira passagem é igual ao início da segunda.

10.4.5 Impressão digital de literatura

A impressão digital da literatura é um método de visualização de características usadas para caracterizar

caracterizar texto [222].Em vez de calcular apenas um valor de recurso ou vetor para todo o texto (isso é o que normalmente é feito), calculamos uma sequência de valores de recurso por texto e os apresentamos ao usuário como uma impressão digital característica do documento.Isto permite ao usuário "olhar para dentro" do documento e analisar o desenvolvimento dos valores ao longo do texto.Além disso, as informações estruturais do documento são utilizadas para visualizá-lo em diferentes níveis de resolução.A impressão digital da literatura foi aplicada a um problema de atribuição de autoria para mostrar o poder de discriminação das medidas padrão que são assumidas para capturar o estilo de escrita de um autor (ver Figura 10.9).

10.5 Visualizações de coleção de documentos

Na maioria dos casos de visualizações de coleções de documentos, o objetivo é colocar documentos semelhantes próximos uns dos outros e distantes uns dos outros.Este é um problema minimax e normalmente O(n2).Calculamos a semelhança entre todos os pares de documentos e determinamos um layout.As abordagens comuns são layouts de mola de gráfico, escalonamento multidimensional, agrupamento (k-means, hierárquico, maximização de expectativa (EM), vetor de suporte) e mapas auto-organizados.Apresentamos diversas visualizações de coleções de documentos, como mapas auto-organizados, mapas de cluster e paisagens temáticas.

10.5.1 Mapas auto-organizáveis

Um mapa auto-organizado (SOM) [248] é um algoritmo de aprendizagem não supervisionado que usa uma coleção de nós tipicamente 2D, onde os documentos serão localizados.Cada nó possui um vetor associado da mesma dimensionalidade dos vetores de entrada (os vetores de documento) usados para treinar o mapa.Inicializamos os nós SOM, normalmente com pesos aleatórios.Escolhemos um vetor aleatório dos vetores de entrada e calculamos sua distância de cada nó.Nós ajustamos o

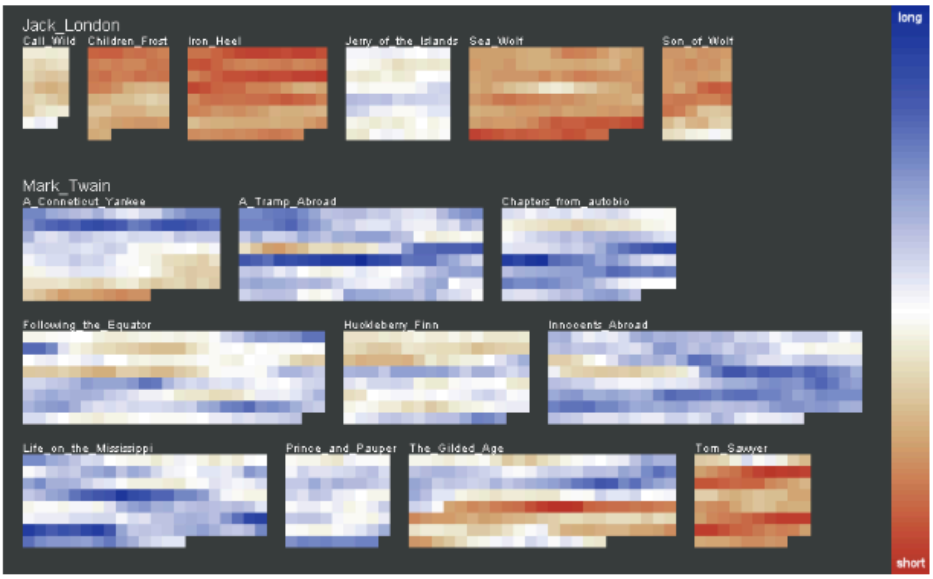

Figura 10.9. Técnica de impressão digital de literatura.Aqui, a impressão digital da literatura é usada para analisar a capacidade de diversas medidas textuais de discriminar entre autores.Cada pixel representa um bloco de texto e os pixels são agrupados em livros.A cor é mapeado para o valor do recurso, neste caso para o comprimento médio da frase.Se uma medida for capaz de discriminar entre os dois autores, os livros da primeira linha (escritos por Londres) são visualmente separados dos restantes livros (escritos por Mark Twain).(Imagem de [222], c© 2007 IEEE.)

pesos dos nós mais próximos (dentro de um determinado raio), tornando cada um mais próximo do vetor de entrada, com os pesos mais altos correspondendo ao nó selecionado mais próximo.À medida que iteramos pelos vetores de entrada, o raio fica menor.Um exemplo do uso de SOMs para dados de texto é mostrado na Figura 10.10 [454], que mostra um milhão de documentos coletados de 83 grupos de notícias.

10.5.2 Paisagens temáticas

Themescapes são resumos de corpora usando paisagens 3D abstratas nas quais altura e cor são usadas para representar a densidade de documentos semelhantes.O exemplo mostrado na Figura 10.11 do Pacific Northwest National Labs [407] representa artigos de notícias visualizados como uma paisagem temática.As montanhas mais altas representam temas frequentes no corpus documental (a altura é proporcional ao número de documentos relativos ao tema).

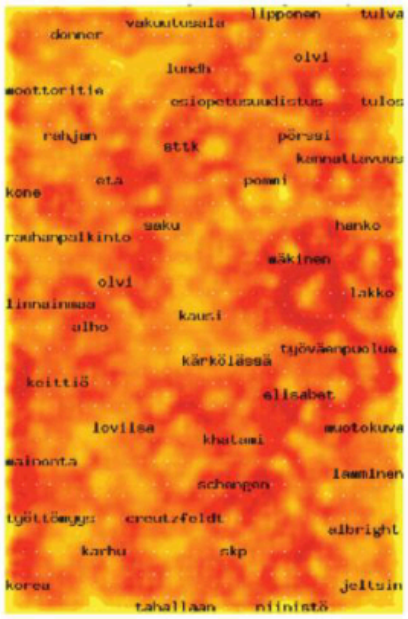

Figura 10.10. Um layout de mapa auto-organizado (SOM) de boletins de notícias finlandeses.Os rótulos mostram as áreas temáticas e a cor representa o número de documentos, com áreas claras contendo mais [454].

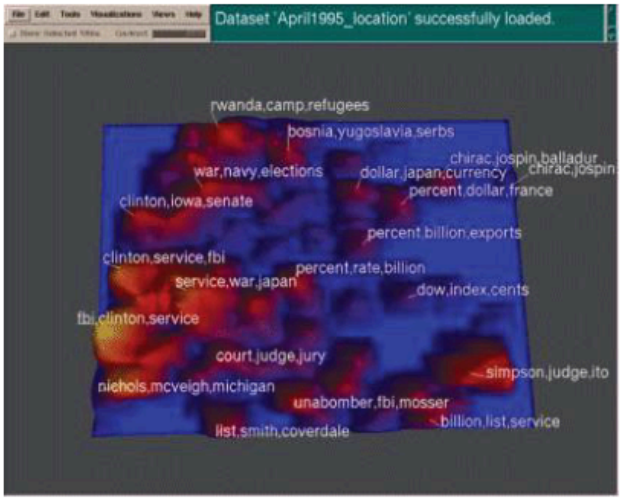

Figura 10.11. Um themescape do PNNL que usa altura para representar a frequência dos temas nas reportagens.(Imagem reimpressa de [407] com permissão da Springer Science and Business Media.)

10.5.3 Cartões de documentos

Os cartões de documentos são uma visualização compacta (Figura 10.12) que representa a semântica-chave do documento como uma mistura de imagens e termos-chave importantes, semelhantes às cartas em um jogo de trunfos [400].Os termos-chave são
extraído usando uma abordagem avançada de mineração de texto baseada em uma extração automática da estrutura do documento.As imagens e suas legendas são ex-
são tratadas por meio de uma heurística gráfica e as legendas são utilizadas para uma ponderação semi-semântica da imagem.Além disso, o histograma de cores da imagem é usado para classificar imagens em classes (classe 1: fotografia/imagem renderizada, classe 2: diagrama/esboço/gráfico, classe 3: tabela) e mostrar pelo menos um representante de cada classe não vazia.

10.6 Visualizações de texto estendidas

Aqui investigamos diversas técnicas de visualização de texto que envolvem metadados ou que vão além das típicas visualizações baseadas em termos vetoriais.

10.6.1 Visualização de Software

Eick et al.desenvolveu uma ferramenta de visualização chamada SeeSoft [108] que visualiza estatísticas para cada linha de código (ou seja, idade e número de modificações, programador, datas).Na Figura 10.13, cada coluna representa um arquivo de código-fonte com a altura representando o tamanho do arquivo.Se o arquivo for maior que
na tela, ele continua na próxima coluna.Na representação clássica da SeeSoft
representação, cada linha representa uma linha de código.Como o número de linhas é muito grande para uma linha, cada linha de código é representada por um pixel na linha.Isso aumenta o número de linhas que podem ser exibidas.A cor é usada para representar a contagem de chamadas.Quanto mais vermelha for uma linha, mais frequentemente ela será chamada e, portanto, é um ponto importante.Uma linha azul raramente é chamada.A cor pode ser usada para representar outros parâmetros, como hora da última modificação ou número de modificações.Com uma tela de 1K × 1K, o SeeSoft é capaz de exibir até 50.000 linhas de código.Esta figura contém 52 arquivos com 15.255 linhas de código.O arquivo selecionado é file1.c, um bloco de código com visualização ampliada da linha 408.

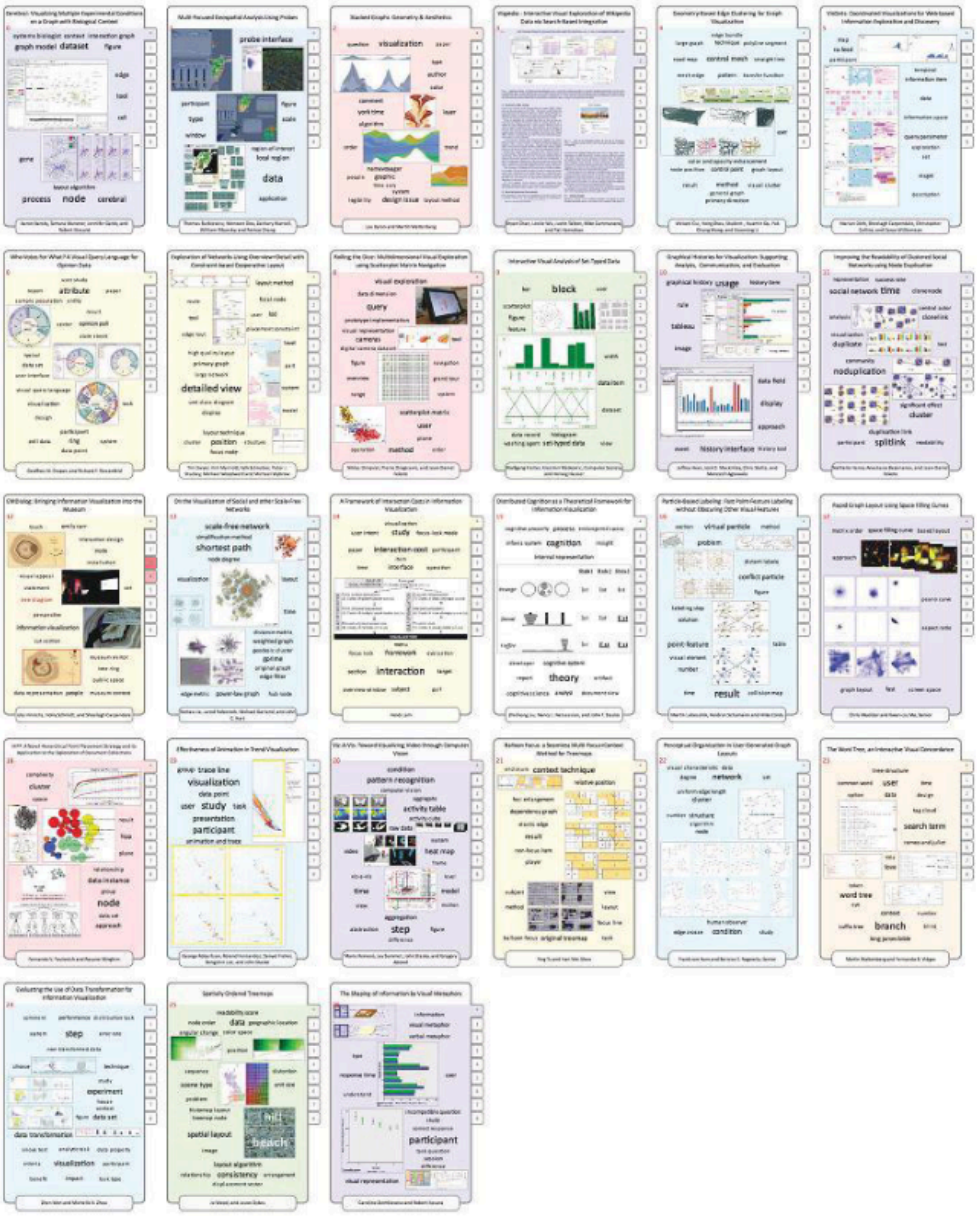

Figura 10.12. O corpus de procedimentos do IEEE InfoVis 2008, representado por uma matriz de fichas de documentos.A frequência do termo em cada página é mostrada no lado direito da ficha do documento (quanto mais vermelho, maior a frequência, como pode ser visto no primeiro documento da linha três) [400].

Figura 10.13. A visualização do software SeeSoft. Retângulos representam arquivos de código-fonte. O os tamanhos dos retângulos em cada coluna correspondem ao comprimento do arquivo de código-fonte e a cor de cada linha representa os parâmetros relacionados à modificação.(Imagem de [108], c© 1992 IEEE.)

10.6.2 Visualização do resultado da pesquisa

Marti Hearst desenvolveu uma visualização simples de resultados de consulta fundamentalmente semelhante aos displays de pixel de Keim [232], chamada TileBars [178], que exibe uma série de estatísticas relacionadas a termos, incluindo frequência e distribuição de termos, comprimento do documento, classificação baseada em termos e força da classificação.Cada documento do conjunto de resultados é representado por um retângulo, onde a largura indica o comprimento relativo do documento e os quadrados empilhados correspondem aos segmentos de texto (ver Figura 10.14).Cada linha da pilha representa um

conjunto de termos de consulta, e o quadrado escuro indica a frequência

de termos entre os termos correspondentes.Títulos e as primeiras palavras de

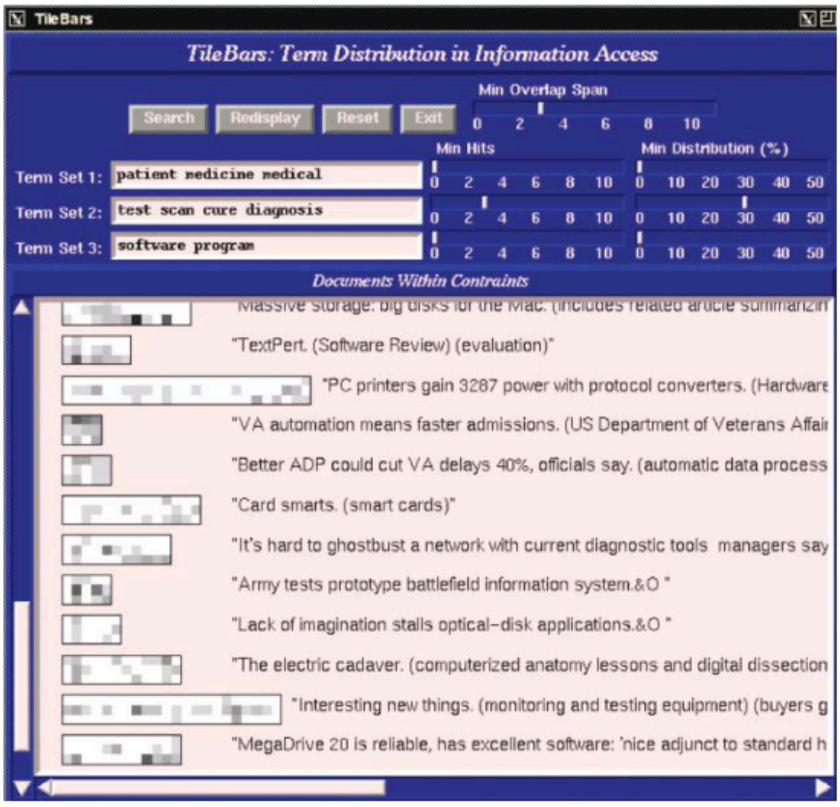

Figura 10.14. A visualização do resultado da consulta TileBars.Cada retângulo grande indica um documento e cada quadrado dentro do documento representa um segmento de texto.Quanto mais escuro o bloco, mais frequente será o conjunto de termos de consulta.(Imagem de [178], c© 1995 Addison-Wesley.)

o documento aparece próximo ao seu TileBar.Cada retângulo grande indica um documento e cada quadrado dentro do documento representa um segmento de texto.Quanto mais escuro o bloco, mais frequente é o conjunto de termos de consulta.Isso produz uma representação compacta e fornece feedback sobre a estrutura do documento, refletindo o comprimento relativo do documento, a frequência do termo da consulta e a distribuição do termo da consulta.

10.6.3 Visualizações de coleção de documentos temporais

ThemeRiver [173], também chamado de gráfico de fluxo, é uma visualização de mudanças temáticas em uma coleção de documentos ao longo do tempo (Figura 10.15).Esta visualização-

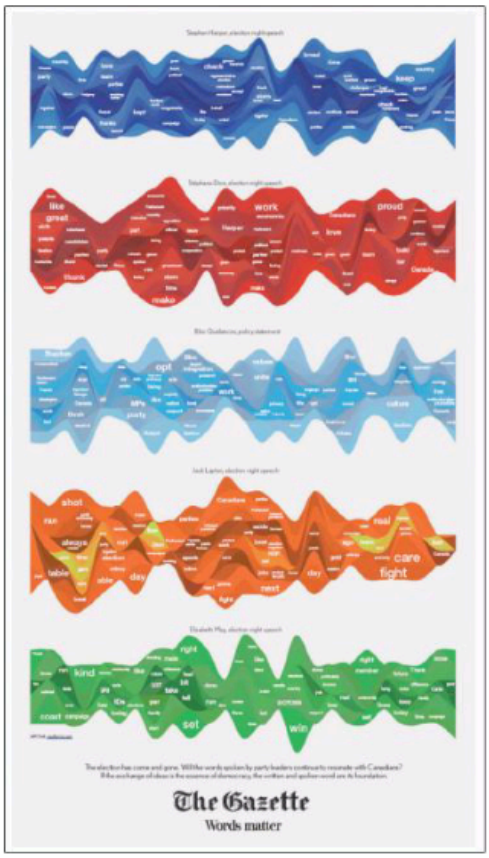

Figura 10.15. Um gráfico de fluxo (ThemeRiver), representando os discursos da noite eleitoral de vários candidatos diferentes para uma eleição canadense.(Imagem de [173], c© 2002 IEEE.)

A situação pressupõe que os dados de entrada progridem ao longo do tempo.Os temas são representados visualmente como faixas horizontais coloridas cuja espessura vertical em um determinado local horizontal representa sua frequência em um determinado momento.

Jigsaw é uma ferramenta para visualizar e explorar corpora de texto [155].A visualização de calendário do Jigsaw posiciona os objetos do documento em um calendário com base nas entidades de data identificadas no texto.Quando o usuário destaca um documento, as entidades que ocorrem nesse documento são exibidas (ver Figura 10.16).
Wanner et al.desenvolveu uma ferramenta de análise visual para conduzir análises de sentimento semiautomáticas de grandes feeds de notícias [440].Embora a ferramenta recupere e analise automaticamente feeds RSS em relação a resultados positivos e

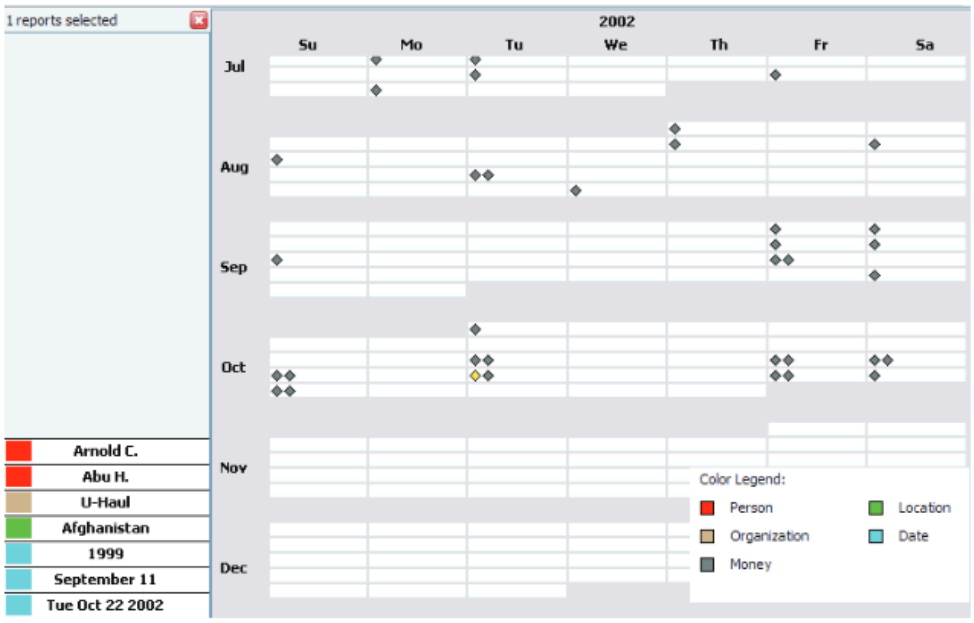

Figura 10.16. Os artigos de notícias são apresentados com a visualização do calendário Jigsaw, com base nas entidades de data extraídas.(Imagem de [155], c© 2007 IEEE.)

palavras de opinião negativa, a análise de notícias mais exigente para encontrar tendências, detectar peculiaridades e colocar eventos em contexto é deixada para o especialista humano.Conforme mostrado na Figura 10.17, cada notícia é representada por um objeto visual e plotado em um eixo de tempo horizontal de acordo com seu horário de publicação.A forma e a cor de um item revelam informações sobre a categoria a que pertence, e seu deslocamento vertical indica se ele tem uma conotação positiva (deslocamento para cima) ou negativa (deslocamento para baixo).

10.6.4 Representando Relacionamentos

Jigsaw [155] também inclui uma visualização de gráfico de entidade (Figura 10.18), na qual o usuário pode navegar em um gráfico de entidades e documentos relacionados.Em quebra-cabeça, entidades estão conectadas aos documentos em que aparecem.A visualização do gráfico Jigsaw não mostra toda a coleção de documentos, mas permite ao usuário expandir gradativamente o gráfico selecionando documentos e entidades de interesse (ver Figura 10.19).

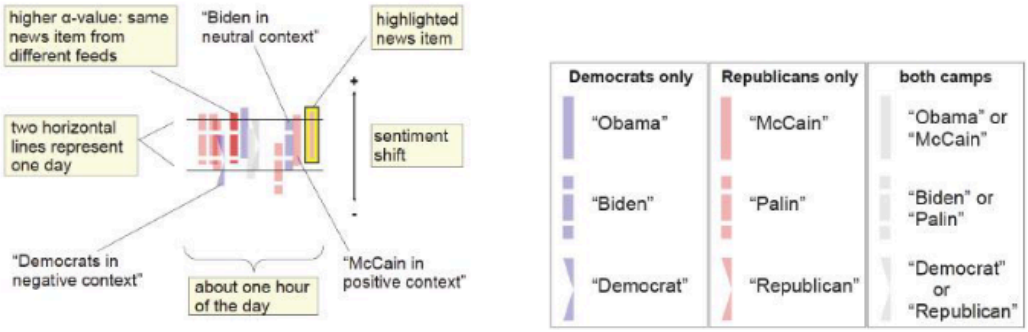

Figura 10.17. Uma visualização de análise de sentimento [440].As notícias são plotadas ao longo do tempo eixo.A forma e a cor mostram a qual categoria um item pertence, e a posição vertical depende da pontuação de sentimento determinada automaticamente de um item.Os objetos visuais que representam notícias são pintados de forma semitransparente para tornar os itens sobrepostos mais facilmente distinguíveis.

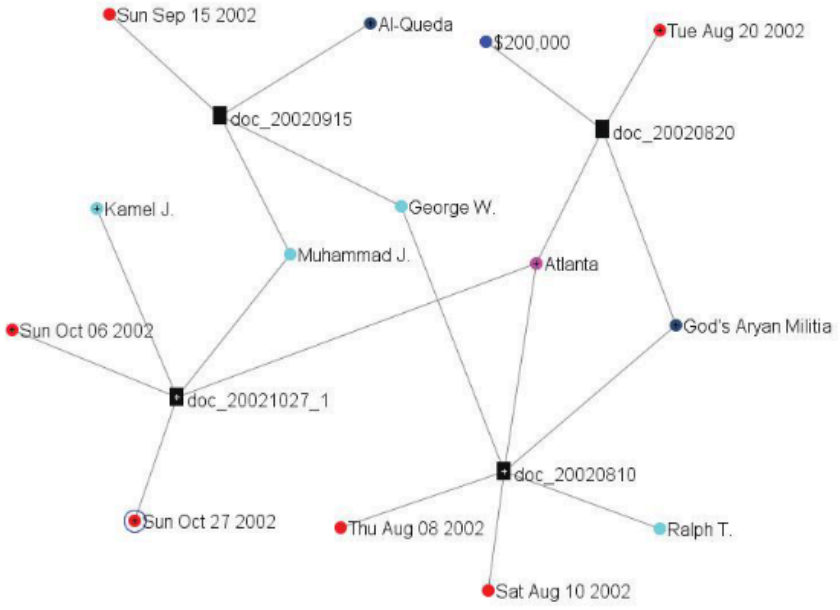

Figura 10.18. A visualização do gráfico Jigsaw, representando conexões entre entidades nomeadas e documentos.(Imagem de [155], c© 2007 IEEE.)

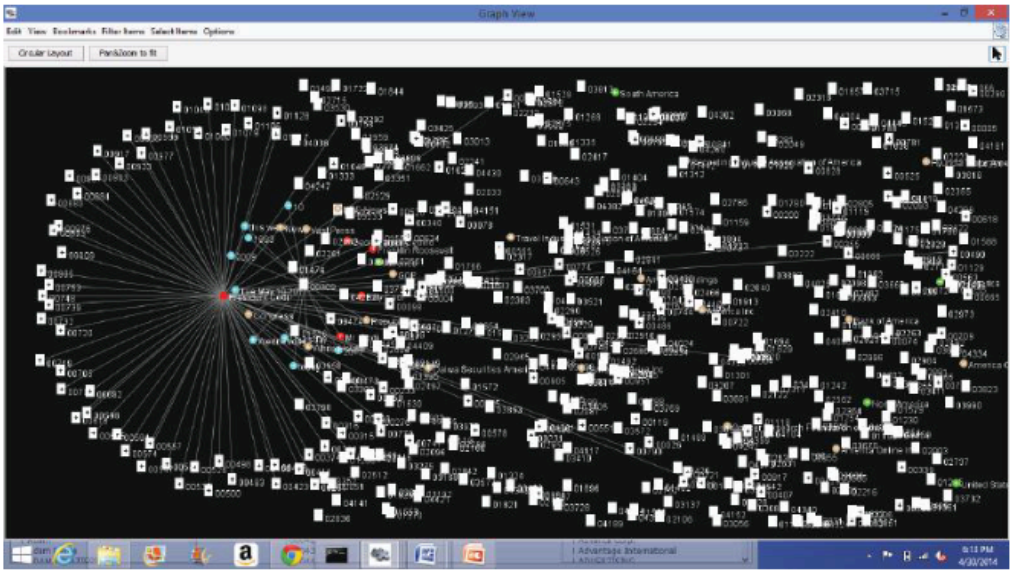

Figura 10.19. Uma visualização de gráfico agrupado no Jigsaw que filtra documentos com entidades específicas.Passar o mouse sobre uma entidade identifica dados sobre o documento.As cores representam valores de token.

A visualização de lista Jigsaw é uma alternativa à visualização gráfica, pois permite ao usuário explorar relacionamentos entre vários tipos de entidades e documentos.Conforme mostrado na Figura 10.20, quando o usuário seleciona itens de interesse, a visualização de lista desenha linhas de conexão mostrando seus relacionamentos.

10.7 Resumo

Neste capítulo, exploramos as abordagens computacionais fundamentais para transformar texto não estruturado em dados estruturados adequados para visualização e análise.Introduzimos visualizações como nuvens de texto e árvores de palavras para encontrar temas e padrões em documentos únicos.Visualizações como SOMs, exibições de mapas e paisagens temáticas são úteis para visualizar coleções de documentos.Para uma análise mais aprofundada de coleções de documentos com relacionamentos complexos e características temporais, pesquisamos brevemente diversas visualizações, como gráficos de nós, ThemeRiver e Calendar View.

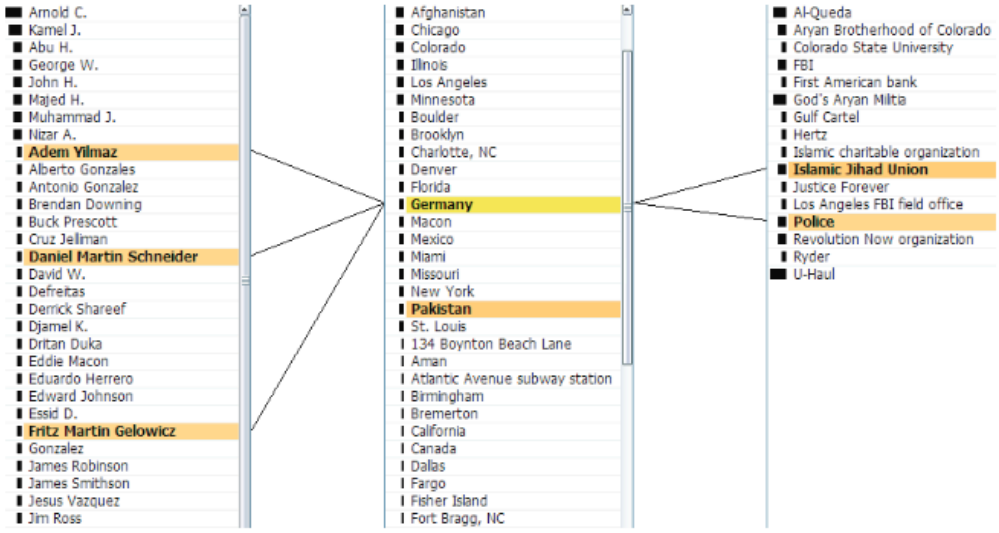

Figura 10.20. A visualização de lista Jigsaw, exibindo as conexões entre pessoas (esquerda), lugares (centro) e organizações (direita).(Imagem de [155], c© 2007 IEEE.)

10.8 Leituras Relacionadas

Uma maravilhosa coleção de artigos originados de uma reunião em 2005 para discutir o estado da arte no processamento de informações visuais e descrever a integração da análise de texto e visualização pode ser encontrada no livro Visual Data Mining: Theory, Techniques and Tools for Visual Analytics, editado por Simoff, Bohlen e Mazeika [379].Mais detalhes sobre mineração e análise de texto podem ser encontrados no livro de Feldman e Sanger, The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data [122].O livro cobre todo o pipeline de descoberta de conhecimento, incluindo visualização.Marti Hearst tem um ótimo livro intitulado Search User Interfaces [179], que inclui um capítulo muito relevante sobre visualização de informações para análise de texto;o livro também está disponível on-line em http://searchuserinterfaces.com/book/.

10.9 Exercícios

1. Dê exemplos dos cálculos sugeridos necessários para o documento análise para as seguintes aplicações:

   (a) identificar plágio, (b) determinar artigos que discutam um tema específico,

(c) selecionar um restaurante chinês com boas críticas, (d) qualquer outro de sua escolha.

2. Quais são algumas vantagens e desvantagens das nuvens de tags?

3. Selecione um documento de sua preferência e gere uma nuvem de tags.

4. Faça uma pesquisa na web em busca de repositórios de texto disponível publicamente corpora.Recupere dois ou três e analise-os em termos de quais problemas eles poderiam ser usados para resolver.Em que formato eles estão?Que pré-processamento é necessário para implementar as visualizações fornecidas neste capítulo?

5. Repita o processo acima, usando um jornal como fonte.Que tipo de dados você pode extrair do jornal?Quais são os tipos de dados?Que conjuntos de dados você poderia obter processando as informações do jornal?Tente projetar pelo menos um conjunto de dados para cada seção do jornal.

6. As técnicas deste capítulo podem ser usadas com notícias televisivas. Como?

7. Procure Weka (http://www.cs.waikato.ac.nz/ml/weka/), uma biblioteca de mineração de dados de código aberto e uma coleção de ferramentas que podem ser usadas como um mecanismo computacional para pré-processamento de texto para visualização.

10.10 Projetos

1. Escreva um programa que determine a distribuição de palavras em um documento mento.

2. Usando o procedimento acima, calcule o tf-idf para esse mesmo documento.

3. Escreva um programa que gere uma nuvem de palavras.

4. Uma tarefa comum ao lidar com dados é dividi-los em categorias, como baixo, médio e alto.Escreva um programa que leia um documento e divida as palavras em três classes: simples, complexas e aquelas intermediárias.

5. Implemente o pseudocódigo deste capítulo em uma seção de texto, digamos, um de seus relatórios ou em um dos conjuntos de dados menores, semelhantes ao VAST, disponíveis no site do livro.

6. Explore a Lei de Zipf em alguns documentos.

7. Baixe e instale o Weka e use-o em um dos conjuntos de dados menores do tipo VAST disponíveis no site do livro ou, se você for ambicioso, em um dos conjuntos de dados VAST.