



Universidade Federal do Amazonas
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Programa de Pós-Graduação em Informática

Geração Semi-automática de Extratores de Dados da Web Considerando Contextos Fracos

Daniel Pereira de Oliveira

Manaus – Amazonas
Março de 2006

Daniel Pereira de Oliveira

Geração Semi-automática de Extratores de Dados da Web Considerando Contextos Fracos

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Departamento de Ciência da Computação da Universidade Federal do Amazonas, como requisito parcial para obtenção do Título de Mestre em Informática. Área de concentração: Recuperação da Informação.

Orientador: Prof. Dr. Altigran Soares da Silva - UFAM/PPGI

Daniel Pereira de Oliveira

Geração Semi-automática de Extratores de Dados da Web Considerando Contextos Fracos

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Departamento de Ciência da Computação da Universidade Federal do Amazonas, como requisito parcial para obtenção do Título de Mestre em Informática. Área de concentração: Recuperação da Informação.

Banca Examinadora

Prof. Dr. Altigran Soares da Silva
Departamento de Ciência da Computação – UFAM/PPGI

Prof. Alberto H. F. Laender, Ph.D.
Departamento de Ciência da Computação – UFMG/DCC

Prof. Dr. Edleno Silva de Moura
Departamento de Ciência da Computação – UFAM/PPGI

Prof. João Marcos Bastos Cavalcanti, Ph.D.
Departamento de Ciência da Computação – UFAM/PPGI

Manaus – Amazonas
Março de 2006

Para minha família.

Não é o poder que corrompe o homem, mas é o
homem quem corrompe o poder.

Ulisses Guimarães

Resumo

Hoje em dia a Web se apresenta como o maior repositório de informações da humanidade. Contudo, essa imensa gama de informação é formada principalmente por conteúdo textual e necessariamente requer interpretação humana para se tornar útil. Por outro lado, existe uma grande quantidade de páginas na Web que são, na verdade, formadas por um conjunto implícito de objetos. Isso ocorre, por exemplo, em páginas oriundas de sites de catálogos on-line, bibliotecas digitais e comércio eletrônico em geral. A extração desse conteúdo e a identificação da estrutura dos objetos disponíveis permite uma forma mais sofisticada de processamento além da tradicional navegação por hiperlinks e consultas por palavras-chave. A tarefa de extrair dados de páginas Web é executada por programas chamados extratores ou *wrappers*.

Neste trabalho propomos uma nova abordagem para o desenvolvimento de extratores. Nessa abordagem o usuário se restringe a fornecer exemplos de treinamento para os atributos que constituem os objetos de interesse. Baseado nesses exemplos, são gerados automaticamente padrões para extrair dados inseridos em contextos similares àqueles fornecidos como exemplos. Em seguida, esses dados são automaticamente organizados segundo uma estrutura plausível.

Nosso método de geração de padrões de extração utiliza técnicas herdadas de soluções para o problema do alinhamento múltiplo de seqüências. O método é capaz de produzir padrões de extração que podem ser facilmente transformados em expressões regulares. A tarefa de inferir uma estrutura plausível para os objetos extraídos é realizada pelo algoritmo HotCycles, que foi previamente proposto e que foi revisto e ampliado neste trabalho. O algoritmo constrói um grafo de adjacências para esses dados, e realiza nele, uma análise estrutural em busca de padrões que indiquem construtores estruturais como tuplas e listas. A partir de tais construtores, é associado um tipo aninhado aos dados que foram extraídos da página.

Experimentos realizados em 21 coleções de páginas reais da Web demonstram a viabilidade do método de extração de valores atômicos, obtendo um desempenho superior a 94% e utilizando no máximo 10 exemplos de treinamento por atributo. O algoritmo HotCycles foi capaz de inferir uma estrutura plausível para os objetos em todas as coleções utilizadas. Seu desempenho combinado com o método de extração de valores atômicos chegou a 97% de estruturas corretamente inferidas com a utilização também até 10 exemplos por atributo.

A combinação desses dois métodos demonstrou-se extremamente viável. Os altos índices de estruturas corretamente inferidas juntamente com os elevados índices de precisão e revocação do processo de extração demonstram que esta é sem dúvida uma abordagem promissora.

Abstract

In the current days, the Internet has become the largest information repository available. However, this huge variety of information is mostly represented in textual format and it necessarily requires human intervention to be effectively used. On the other hand, there exists a large set of Web pages that are in fact composed of collections of implicit data objects. For instance, on-line catalogs, digital libraries and e-commerce Web sites in general. Extracting the contents of these pages and identifying the structure of the data objects available allow for more sophisticated forms of processing besides hyperlink browsing and keyword-based searching. The task of extracting data from Web pages is usually executed by specialized programs called *wrappers*.

In the present work we propose and evaluate a new approach to the wrapper development problem. In this approach, the user is only responsible for providing examples for the atomic items that constitute the objects of interest. Based on these examples, our method automatically generates expressions for extracting other atomic items similar to those presented as example and infers a plausible and meaningful structure to organize them.

Our method for generating extraction expression uses techniques inherited from solutions for the multiple string alignment problem. The method is able to produce good extraction expressions that can be easily encoded as regular expressions. Inferring a meaningful structure for the objects whose atomic values were extracted is the task of the HotCycles algorithm, that were previously proposed and which we have revised and extended in this work. The algorithm assembles an adjacency graph for these atomic values, and executes a structural analysis over this graph, looking for patterns that resemble structural constructs such as tuples and lists. From such constructs, a complex object type can be assigned to the extracted data.

The experiments carried out using 21 collections of real Web pages have demonstrated the feasibility of our extraction method, reaching 94% of effectiveness using no more than 10 examples for each attribute. The HotCycles algorithm was able to infer a meaningful structure for the objects present in all used collections. Its effectiveness, combined with our atom extraction method, reached 97% of structures correctly inferred, also using no more than 10 examples per attribute.

The association of these two methods has demonstrated to be extremely feasible. The high number of correctly inferred structures together with the high precision and recall values of the extraction process demonstrates that this new approach is indeed a promising one.

Sumário

1	Introdução	1
1.1	Abordagem Proposta	6
1.2	Trabalhos Relacionados	8
1.3	Principais Contribuições	12
1.4	Organização da Dissertação	12
2	Extração dos Valores Atômicos	14
2.1	Conceitos Básicos	15
2.1.1	Problema do Alinhamento Múltiplo de Seqüências	16
2.1.2	Valor de Similaridade de um Alinhamento	17
2.1.3	Técnica do Alinhamento Múltiplo Progressivo	18
2.2	Identificação dos Delimitadores	19
2.3	Alinhamento dos Delimitadores	20
2.4	Esquema de Pontuação	22
2.5	Posicionamento dos Delimitadores	23
2.6	Solução por Programação Dinâmica	25
2.7	Generalização das Expressões de Consenso	26
2.8	Processo de Inferência	27
2.9	Geração da Expressão Regular	30
2.10	Estratégia de Extração dos Valores Atômicos	33
2.11	Um Exemplo Completo	34
3	Montagem dos Objetos	38
3.1	Grafo de Adjacências	39

3.2	Ciclos Dominantes	40
3.3	Ciclos Estruturantes	43
3.4	Substituição das Ocorrências do Ciclo Estruturante	48
3.5	Algoritmo HotCycles	49
3.6	Um Exemplo Completo	51
4	Resultados Experimentais	55
4.1	Metodologia de Avaliação	55
4.1.1	Coleções Utilizadas	57
4.1.2	Métrica de Avaliação	57
4.2	Experimentos de Extração	59
4.2.1	Resultado Geral	59
4.2.2	Experimentos com a Coleção DBLP	63
4.3	Experimentos de Montagem	69
4.3.1	Desempenho Isolado	70
4.3.2	Desempenho Combinado	70
4.3.3	Resultado Geral	71
4.3.4	Experimentos com a Coleção APSA	74
4.4	Resultado Final	75
5	Conclusões e Trabalhos Futuros	78
A	Detalhamento dos Experimentos com as Coleções Utilizadas	80
	Referências Bibliográficas	121

Lista de Figuras

1.1	Página de livro na Amazon.com	2
1.2	Atributos selecionados e XML correspondente	3
1.3	Delimitadores explícitos e implícitos	4
1.4	Estrutura dos objetos na página	5
1.5	Processo de geração e execução de um extrator	7
2.1	Alinhamento múltiplo de DNA	16
2.2	Alinhamento múltiplo de exemplos	17
2.3	Processo de identificação dos delimitadores	20
2.4	Um exemplo de alinhamento de contextos	21
2.5	Exemplo de delimitadores	22
2.6	Exemplo de uma árvore guia	29
2.7	Delimitadores válidos e inválidos	33
2.8	Estratégia de extração	34
2.9	Página do <i>site</i> APSA	35
2.10	Primeiro alinhamento dos exemplos	36
2.11	Segundo alinhamento de exemplos	37
2.12	Último alinhamento e geração da expressão final	37
2.13	Expressão de extração final e máscara dos dados	37
3.1	Página de catálogo da Amazon.com	40
3.2	Grafo de adjacências para uma página da Amazon.com	40
3.3	Contagem de ocorrências de um ciclo num grafo de adjacências	42
3.4	Casos de generalização do ciclo dominante	46
3.5	Utilização dos fatores E e F	48

3.6	Grafo de adjacências inicial das páginas de autores do <i>site</i> DBLP	52
3.7	Montagem do objetos da página de autores da DBLP - Passo 01	52
3.8	Montagem do objetos da página de autores da DBLP - Passo 02	53
3.9	Montagem do objetos da página de autores da DBLP - Último Passo	53
4.1	Página de autores do <i>site</i> DBLP	63
4.2	Padrão de extração para o atributo Autor	64
4.3	Padrão de extração para o atributo Título	65
4.4	Padrão de extração para o atributo Local de Publicação	66
4.5	Padrão de extração para o atributo Volume	66
4.6	Padrão de extração para o atributo Páginas	67
4.7	Padrão de extração para o atributo Ano	67
4.8	Resultados com a coleção DBLP por atributo sem realimentação	67
4.9	Resultados com a coleção DBLP por atributo com realimentação	68
4.10	Resultados com a coleção DBLP por atributo com realimentação preguiçosa	68
4.11	Resultados gerais obtidos na coleção DBLP	69
4.12	Página do site da APSA	74
4.13	Resultados obtidos na coleção APSA	75
4.14	Resultado geral do método de inferência de estrutura	76
A.1	Resultados obtidos na coleção do All Politics	82
A.2	Resultados obtidos na coleção APSA	84
A.3	Resultados obtidos na coleção do CDDB	86
A.4	Resultados obtidos na coleção do CDNOW	88
A.5	Resultados obtidos na coleção do IMDB	90
A.6	Resultados obtidos na coleção Loteria	92
A.7	Resultados obtidos na coleção Monster	94
A.8	Resultados obtidos na coleção PubMed	96
A.9	Resultados obtidos na coleção Restaurants	98
A.10	Resultados obtidos na coleção DNC	100
A.11	Resultados obtidos na coleção Film.com	102
A.12	Resultados obtidos na coleção PharmWeb	104

A.13 Resultados obtidos na coleção News.com	106
A.14 Resultados obtidos na coleção Sigmod	108
A.15 Resultados obtidos na coleção Sony	110
A.16 Resultados obtidos na coleção VLDB	112
A.17 Resultados obtidos na coleção Wine	114
A.18 Resultados obtidos na coleção Yahoo-Movie	116
A.19 Resultados obtidos na coleção Yahoo-Telephone	118
A.20 Resultados obtidos na coleção Zagat	120

Lista de Tabelas

2.1	Exemplos selecionados para o atributo artigo	35
2.2	Matriz de distâncias inicial para os exemplos do atributo artigo	36
2.3	Matriz de distâncias após a primeira generalização	36
4.1	Coleções utilizadas nos experimentos	58
4.2	Resultado geral do método de extração de atributos atômicos	60
4.3	Tempo médio de geração dos padrões de extração (em milisegundos)	62
4.4	Estruturas inferidas corretamente pelo método sobre as bases de referência	71
4.5	Percentual de estruturas corretamente inferidas por número de exemplos	72

Capítulo 1

Introdução

A *World Wide Web* tornou-se o maior e mais diversificado repositório de informação já disponível na história da humanidade. Porém, como a maior parte de seu conteúdo encontra-se em forma textual (artigos, livros, publicações científicas, etc.) este se torna praticamente inaproveitável sem interpretação humana. A falta inerente de estrutura dificulta o processamento automático da informação tornando seu acesso limitado ao processo de navegação nos *hiperlinks* ou através de busca por palavras-chave.

Acontece porém, que uma grande parte da Web apresenta páginas que contêm dados semi-estruturados, ou seja, dados que possuem uma estrutura ou *esquema*, mesmo que este não seja explicitamente declarado. Tipicamente, essas páginas são geradas dinamicamente a partir de consultas a um banco de dados e pela aplicação de um *template* HTML em seguida. Podemos considerar essas páginas como sendo repositórios de objetos implícitos, os quais podem ser identificados, extraídos e manipulados independentemente. Exemplos comuns são as páginas de *sites* de comércio eletrônico, anúncios, catálogos on-line, etc.

Uma característica comum a essas páginas é que, apesar de não apresentarem uma estrutura explicitamente declarada, esta é imediatamente identificada seja pela disposição e formatação estilística dos dados, pelo tipo da informação em si ou pelo contexto sintático em que está inserida. Podemos verificar isso através das páginas contendo dados sobre livros disponíveis no *site* da Amazon.com, conforme apresentado na Figura 1.1. Nela identificamos um esquema implícito formado por um título, lista de autores, preço do livro, disponibilidade e assim por diante.

Ter acesso aos dados implícitos nessas páginas é de grande utilidade, pois possibilita a realiza-

Software Engineering (7th Edition) (International Computer Science Series)by [Ian Sommerville](#)[Publisher: learn how customers can search inside this book.](#)[Share your own customer images](#)**List Price:** ~~\$99.68~~**Price:** **\$94.62** & This item ships for **FREE with Super Saver Shipping.** [See details](#)**You Save:** **\$4.98 (5%)****Availability:** Usually ships within 24 hours from Amazon.com**Want it delivered Thursday, January 20?** Order it in the next 1 hour and 58 minutes, and choose **One-Day Shipping** at checkout. [See details](#)[102 used & new](#) from **\$43.00****Edition:** Hardcover

Figura 1.1: Página de livro na Amazon.com

ção de consultas complexas sobre a informação. Além disso, o acesso a estes dados é fundamental para os sistemas de integração da informação e para os chamados *Agentes Web* que são programas que utilizam Web para a automatização de tarefas específicas, como verificação de horários de vôos, checagem de correio eletrônico, consulta de manchetes em jornais eletrônicos, etc. A navegação por *hyperlinks* e a busca por palavras-chave não são suficientes para as páginas que apresentam conteúdo semi-estruturado. Assim uma forma mais sofisticada de acesso é necessário, o que só pode ser alcançado pela extração e estruturação do seu conteúdo.

Com a introdução da linguagem XML, que possui uma forma estruturada para representar os dados, procurou-se superar algumas limitações da linguagem HTML que é voltada para a apresentação. O XML promoveu a separação entre os dados e sua formatação, fornecendo um modo padrão para a troca de informações. Seu uso, contudo, ainda é restrito principalmente a aplicações *business-to-business*. Uma discussão aprofundada sobre essas linguagens é apresentada em [Light, 1997].

Devido a larga popularidade e facilidade de uso, o HTML já é uma linguagem consolidada na Web e continuará sendo por algum tempo seu formato padrão de publicação. Isso faz com que o problema da extração de dados dessas páginas continue de grande importância prática.

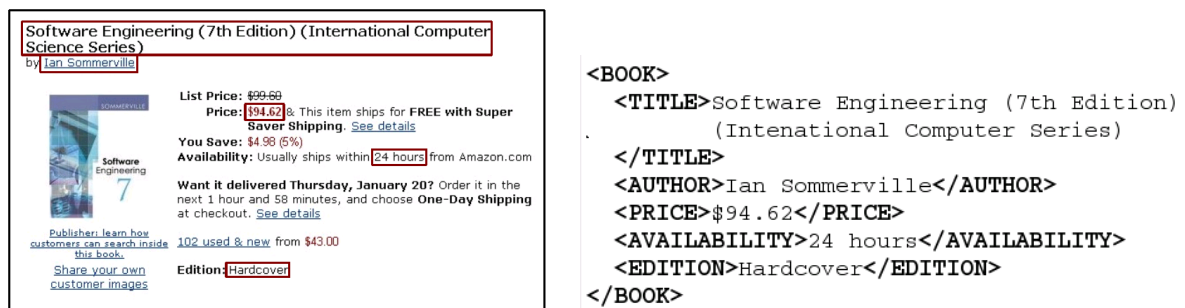
A extração de dados de páginas HTML é normalmente realizada usando programas chamados extratores ou *wrappers*, que localizam e extraem dados das páginas e os transformam em um formato adequado que possa ser processado de maneira automatizada tal como tabelas relacionais ou arquivos XML [Arasu and Garcia-Molina, 2003, Baumgartner et al., 2001,

Crescenzi et al., 2001, Liu et al., 2000, Laender et al., 2002b, Sahuguet and Azavant, 2001]. Os dados de interesse são localizados através da identificação de padrões que aparecem no documento junto com esses dados, tais como formatação, disposição, tipo e contexto sintático.

Uma forma de encarar o problema de extração de dados de páginas da Web consiste em dividi-lo em duas etapas:

- localização e extração dos valores atômicos que constituem os objetos de interesse;
- montagem dos objetos de acordo com alguma estrutura plausível.

Considerando novamente a Figura 1.1, um extrator, que seguisse os passos descritos acima, localizaria, inicialmente, os valores atômicos constituintes dos objetos, tais como: título, autores, preço, etc. Em seguida combinaria esses dados segundo uma estrutura plausível de forma a reconstituir o objeto complexo, conforme apresentado na Figura 1.2.



Software Engineering (7th Edition) (International Computer Science Series)
by Ian Sommerville

List Price: ~~\$99.99~~
Price: **\$94.62** & This item ships for **FREE** with Super Saver Shipping. [See details](#)
You Save: \$4.98 (5%)
Availability: Usually ships within **24 hours** from Amazon.com

Want it delivered Thursday, January 20? Order it in the next 1 hour and 58 minutes, and choose **One-Day Shipping** at checkout. [See details](#)

102 used & new from \$43.00
Edition: **Hardcover**

```
<BOOK>
  <TITLE>Software Engineering (7th Edition)
    (Intenational Computer Series)
  </TITLE>
  <AUTHOR>Ian Sommerville</AUTHOR>
  <PRICE>$94.62</PRICE>
  <AVAILABILITY>24 hours</AVAILABILITY>
  <EDITION>Hardcover</EDITION>
</BOOK>
```

Figura 1.2: Atributos selecionados e XML correspondente

O desenvolvimento de extratores para páginas HTML apresenta grandes desafios, a maior parte oriunda do fato que o formato HTML é destinado à visualização pelo usuário. O que ocorre é que normalmente os itens de dados aparecem misturados junto com elementos de marcação e *hyperlinks*, além de texto irrelevante. Com isso, boa parte dos dados de interesse nas páginas Web estão inseridas em contexto fraco.

Dizemos que um contexto é pobre ou fraco quando ele é formado por delimitadores variáveis, cujo conteúdo muda conforme o dado de interesse. Além de variáveis esses delimitadores podem inclusive estar ausentes. Ao contrário, um contexto é forte quando é formado por delimitadores estáticos que sempre estão presentes junto ao dado de interesse.

A Figura 1.3(a) apresenta exemplos de valores atômicos ocorrendo em uma página com contexto forte, todas as instâncias dos valores atômicos estão inseridas entre delimitadores

estáticos cujos valores são sempre fixos. Diferentemente da Figura 1.3(b), são apresentados valores atômicos inseridos em contexto fraco. Percebemos que nesse caso o valor dos delimitadores é variável, tal como o número e a cidade do endereço. O complemento *Beverly Center* figura como um valor opcional.

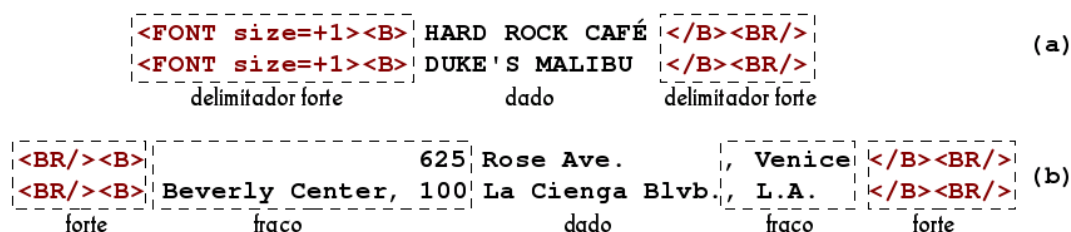


Figura 1.3: Delimitadores explícitos e implícitos

Outro desafio para a geração de extratores está relacionado a estrutura dos objetos. Os objetos nas páginas da Web possuem uma estrutura que deve ser respeitada pelo extrator. Essa estrutura está implícita nas páginas e na maioria das vezes apresenta variações e elementos opcionais.

Tipicamente, os objetos nas páginas da Web podem ser representados por tipos aninhados [da Silva et al., 2002]. Esses tipos aninhados são formados pela combinação de três elementos:

1. elemento atômico - um atributo existente na página HTML;
2. lista - uma seqüência de elementos de mesmo tipo;
3. tupla - uma seqüência de elementos de tipos distintos.

Na Figura 1.4 é apresentada uma página contendo informações sobre filmes em exibição em um determinado cinema do *site* Yahoo!¹. Cada um dos cinemas apresentados nas páginas desse *site* pode ser representado por um objeto que possui sua estrutura indicada pelo tipo aninhado nessa figura.

As abordagens iniciais para geração de extratores eram baseadas na detecção manual do *template* HTML e na codificação de regras de transformação utilizando linguagens de propósito geral como *Perl* e *Java*. O desenvolvimento manual de extratores sofre principalmente de dois problemas: o esforço inicial de programação e os custos de manutenção, uma vez que essas páginas tendem a modificar sua estrutura com certa freqüência [Laender et al., 2002b].

¹<http://br.movies.yahoo.com>

Centerplex Shopping Lapa

Endereço: Rua Catão, 72
Shoppings Centers
São Paulo, SP
Telefone: (11) 3675-4681

Todos os filmes neste Cinema:

Sala	Lugares	Filme
1	290	Wallace e Gromit - A Batalha dos Vegetais Horários: 13h(a) (dub.)
1	290	Jogos Mortais 2 Horários: 15h(a) / 17h / 19h / 21h10
2	151	A Lenda do Zorro Horários: 13h30(a) / 16h / 18h30 / 21h

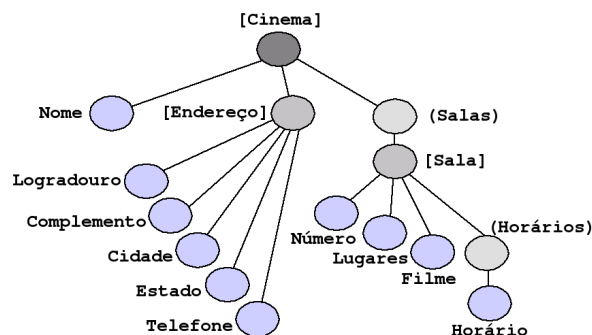


Figura 1.4: Estrutura dos objetos na página

Mais tarde, diversas técnicas automáticas e semi-automáticas foram propostas. Uma característica das técnicas de extração existentes é a dificuldade para tratar dados inseridos em contextos fracos.

Os métodos automáticos de geração [Arasu and Garcia-Molina, 2003, Crescenzi et al., 2001] e os métodos supervisionados [Laender et al., 2002b, Adelberg, 1998, Baumgartner et al., 2001, Sahuguet and Azavant, 2001] não apresentam bom suporte para delimitadores fracos. Esse métodos extraem como uma unidade atômica tudo que não for fortemente delimitado ou geram regras disjuntas sendo necessário uma para cada possível variação dos delimitadores. Alguns métodos de indução apresentam reduzido suporte a esse tipo de delimitador [Muslea et al., 2001].

Outra característica importante de uma ferramenta de extração é o suporte a tipos aninhados, uma vez que a maior parte dos objetos codificados nas páginas HTML não apresenta uma estrutura plana. Nos métodos supervisionados, o usuário deve fornecer essa estrutura, o que consome tempo [Laender et al., 2002b, Baumgartner et al., 2001, Adelberg, 1998, Sahuguet and Azavant, 2001]. Nos automáticos, ela é dependente da disposição dos elementos na página, o que na maioria das vezes não é satisfatório [Crescenzi et al., 2001, Arasu and Garcia-Molina, 2003].

Neste trabalho, desenvolvemos uma nova abordagem de geração semi-automática de extractores onde o usuário fornece exemplos de treinamento para os atributos a serem extraídos das páginas. Com base nesses exemplos, padrões de extração, são automaticamente inferidos mesmo quando a informação contextual é pobre. Em seguida, os objetos são reconstituídos de acordo com uma estrutura que é automaticamente inferida pela análise da distribuição dos valores atômicos extraídos.

Em nossa abordagem, os exemplos de treinamento selecionados pelo usuário são transformados em padrões de extração por meio de progressivos alinhamentos e generalizações. Diferentemente de outras abordagens que requerem uma interação intensa com o usuário, seja pelo processo de especificação visual do extrator, seja pela definição de uma estrutura para os dados, a nossa proposta requer apenas a seleção de alguns exemplos de treinamento, tarefa essa que pode ser enormemente facilitada pela utilização de uma interface gráfica.

A montagem dos objetos é automática, realizada através da análise de padrões na distribuição dos valores atômicos extraídos, realizada pelo algoritmo HotCycles [da Silva, 2002] que é capaz de gerar estruturas aninhadas multi-nível, representando mais fielmente os objetos contidos nas páginas que as estruturas geradas por métodos completamente automáticos onde não conseguem distinguir o que é realmente relevante ao usuário refletindo mais a estrutura da página do que a dos objetos de interesse.

Experimentos realizados com 21 coleções de páginas demonstram a viabilidade do nosso método de extração que com poucos exemplos de treinamento é capaz de alcançar níveis de desempenho superiores a 94% de eficácia. Estes resultados, aliados a capacidade de inferir estruturas plausíveis para montar os objetos sem a utilização de nenhuma informação estrutural prévia faz com que o método aqui descrito se constitua numa abordagem inovadora e promissora para a geração de extratores para páginas da Web.

1.1 Abordagem Proposta

Nesta seção apresentaremos uma visão geral da abordagem proposta. Cada uma das etapas que constituem nosso método de extração de dados será detalhadamente discutida posteriormente.

Na literatura são descritos diferentes métodos para a extração de dados de páginas da Web. Pode-se utilizar a estrutura do documento HTML para extrair a informação, seja através da análise da estrutura das árvores DOM² [Crescenzi et al., 2001, Liu et al., 2000, Sahuguet and Azavant, 2001], seja através da decomposição hierárquica da página [Adelberg, 1998, Muslea et al., 2001]. É possível utilizar também o contexto local que envolve os dados de interesse [Laender et al., 2002b]. Algumas abordagens combinam esses métodos para alcançar melhores resultados como em [Baumgartner et al., 2001].

²Document Object Model - Modelo de objeto que representa um documento

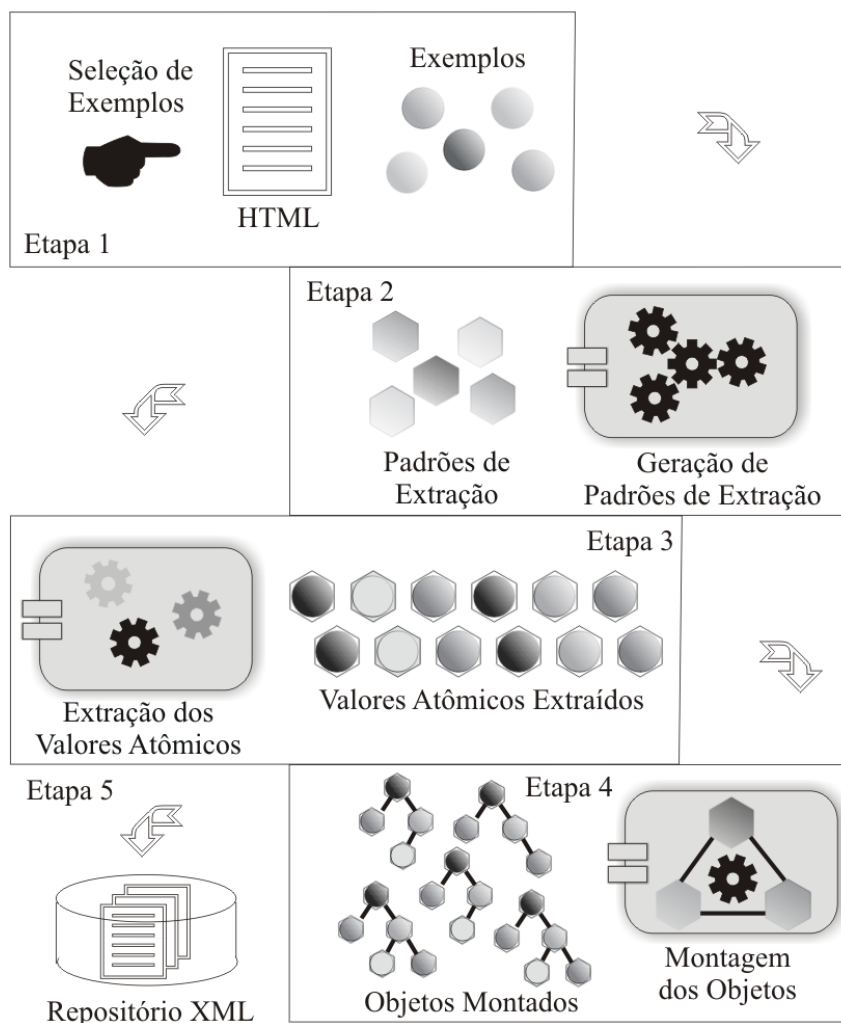


Figura 1.5: Processo de geração e execução de um extrator

Em nossa abordagem, utilizamos expressões regulares para extrair a informação. Não fazemos nenhuma consideração a respeito da estrutura do documento HTML, utilizamos somente o contexto local como evidência para a geração das expressões regulares.

De modo a facilitar o processamento das páginas HTML realizamos uma normalização prévia do arquivo a ser processado, transformando as páginas HTML em documentos XHTML bem formados. Essa normalização visa padronizar a geração de expressões regulares de modo a considerar que *tags* HTML ocorrerão sempre em caixa alta e o valor de seus atributos estarão sempre entre aspas. Nossa abordagem de geração semi-automática de extratores baseada em exemplos pode ser resumida de acordo com as etapas ilustradas na Figura 1.5.

Na Etapa 1 o usuário deve fornecer um conjunto de exemplos de treinamento para os atributos que deseja extrair da página. Para cada exemplo fornecido são extraídos o sufixo e o prefixo que

são em seguida transformados em listas de delimitadores.

Após a identificação dos delimitadores, iniciamos, na Etapa 2, a procura por alguma informação contextual comum para os exemplos fornecidos. Nessa busca por padrões contextuais semelhantes, realizamos progressivos alinhamentos dos exemplos de treinamento em busca de uma boa distribuição dos delimitadores o que nos permitirá realizar posteriormente a geração de um padrão de extração.

Durante o processo de alinhamento de exemplos, atribuímos a cada par de delimitadores graus de similaridade de modo a possibilitar o casamento aproximado pelo algoritmo de alinhamento de seqüências [Gusfield, 1997]. Utilizamos em nossa abordagem uma técnica de alinhamento múltiplo progressivo [Feng and Doolittle, 1987] que gera consensos entre os exemplos fornecidos e que vão sendo refinados com novos exemplos de treinamento de forma a generalizar um padrão de extração.

Depois dos sucessivos alinhamentos, passamos para a Etapa 3, onde transformamos o padrão de extração gerado em uma expressão regular que é então avaliada para se recuperar os dados nas páginas de interesse. A extração dos valores atômicos é realizada por um interpretador de expressão regular padrão que seqüencialmente extrai a informação que casa com o padrão fornecido.

Uma vez que todos os valores atômicos foram extraídos, inicia-se a Etapa 4 onde ocorre a montagem dos objetos complexos de acordo com a estrutura que vai sendo inferida pelo algoritmo HotCycles [da Silva, 2002]. O algoritmo procura, na lista de valores atômicos extraídos, padrões que identifiquem elementos estruturais como listas ou tuplas e vai, dessa forma, reconstituindo os objetos.

Finalmente, na Etapa 5 temos os objetos devidamente montados e o arquivo XML com o resultado da extração gerado, finalizando portando o processo de geração e execução do extrator.

1.2 Trabalhos Relacionados

Diferentes abordagens têm sido propostas para o problema da geração de extratores para páginas da Web. Tais abordagens baseiam-se em diversas técnicas como linguagens declarativas [Baumgartner et al., 2001], análise estrutural [Crescenzi et al., 2001, Liu et al., 2000], aprendizagem de máquina [Hsu and Dung, 1998, Kushmerick, 2000, Freitag and Kushmerick, 2000],

processamento de linguagem natural [Soderland, 1999, Muslea et al., 2001], modelagem de dados [Adelberg, 1998, Laender et al., 2002b] e ontologias [Embley et al., 1999]. Uma boa introdução sobre ferramentas de extração de dados é apresentada em [Laender et al., 2002a].

Lixto [Baumgartner et al., 2001] e W4F [Sahuguet and Azavant, 2001] são sistemas para especificação de extratores. Lixto fornece uma interface gráfica onde o usuário pode selecionar partes de interesse no documento HTML e associar filtros aos padrões gerados. O processo de especificação consiste na geração hierárquica de padrões que representam a informação de interesse. Toda a especificação é traduzida para uma linguagem interna baseada em lógica, *ELog* (Extraction Logic) e um interpretador dessa linguagem trata de realizar a extração baseado-se no programa *ELog* fornecido. É possível definir regras específicas para a geração de arquivos XML com os dados extraídos.

W4F adota as mesmas idéias, utilizando porém uma linguagem hierárquica chamada HEL (HTML Extraction Language) que é baseada na estrutura DOM³ da página. O suporte da interface gráfica em bem menor que o do Lixto, sendo que, na maioria das vezes a especificação do extrator tem que ser manualmente refinada.

Outras ferramentas que também utilizam essa abordagem supervisionada são NoDoSE [Adelberg, 1998] e DEByE [Laender et al., 2002b]. No NoDoSE o usuário deve fornecer um modelo dos dados a serem extraídos do documento e, em seguida, utilizando uma interface gráfica, deve decompor hierarquicamente o documento, mapeando regiões de interesse para o modelo definido. O sistema então tenta inferir a gramática do documento baseado-se nos exemplos fornecidos pelo usuário. Por fim é possível especificar como os dados extraídos serão exportados.

A abordagem utilizada na ferramenta DEByE consiste na seleção de exemplos dos valores atômicos e na especificação da estrutura dos objetos utilizando o conceito de tabelas aninhadas. O processo de extração se dá pela geração de expressões regulares para identificação dos valores atômicos utilizando apenas o seu contexto local. Em seguida é realizada a montagem do objetos de acordo com a estrutura definida pelas tabelas aninhadas.

Utilizando algumas idéias introduzidas pela ferramenta DEByE, foi proposto um arcabouço para o desenvolvimento de extratores de valores atômicos que possibilita a utilização e combinação de diferentes heurísticas de acordo com as necessidades de cada aplicação

³Document Object Model

[de Castro Reis et al., 2002]. Tal abordagem foi utilizada com sucesso para a realização de algumas aplicações reais [Calado et al., 2002].

Todas as ferramentas discutidas até agora demandam um elevado grau de interação com o usuário. Além disso, o suporte para delimitadores fracos é inapropriado, requerendo especificação manual por parte do usuário. Por outro lado, temos ainda a necessidade do fornecimento da estrutura aninhada dos dados, seja na especificação do tipo aninhado, na decomposição da página ou na própria especificação visual do extrator.

Ferramentas de indução de extratores estão mais relacionadas ao nosso trabalho. Em geral elas utilizam exemplos de treinamento de modo a inferir regras de extração que reconheçam instâncias em contextos similares àqueles que foram fornecidos como exemplos. A ferramenta WIEN [Kushmerick, 2000] depende, assim como a nossa, de exemplos dos elementos a serem extraídos. O método utilizado nessa ferramenta assume que as páginas respeitam uma estrutura pré-determinada: um delimitador inicial, um corpo constituído por um conjunto de tuplas e então um delimitador final. Esse método utiliza apenas delimitadores fortes que aparecem junto aos dados e não é capaz de extrair dados de páginas com componentes opcionais ou fora de ordem. Estruturas aninhadas também não são suportadas.

SoftMealy [Hsu and Dung, 1998] e Stalker [Muslea et al., 2001] também inferem extratores, generalizando exemplos através de técnicas de aprendizado de máquina. As regras de extração do SoftMealy são mais gerais que as do WIEN, além de suportarem componentes opcionais ou fora de ordem. Seus principais problemas são: a necessidade de um exemplo independente para cada possível permutação dos elementos a serem extraídos e a falta de suporte para tipos aninhados.

Stalker é sem dúvida a mais expressiva das técnicas de inferência de extratores descrita na literatura e a que apresenta melhores resultados. Nela o processo de extração segue uma abordagem hierárquica, suportando variações assim como tipos aninhados uma vez que cada componente é extraído independentemente por um processo de decomposição *top-down*. Stalker usa consideravelmente menos exemplos que SoftMealy ou WIEN, porém para seu funcionamento é essencial a definição da estrutura do documento pelo usuário, que deve fornecer além de exemplos dos valores atômicos, exemplos da estrutura dos objetos, de forma a construir o chamado *embedded catalog*. Tal característica restringe consideravelmente a possibilidade de automatização da seleção de exemplos conforme técnicas descritas em [Kushmerick, 2000].

Em nossa abordagem, o usuário se restringe a tão somente fornecer exemplos dos elementos atômicos constituintes do documento. No caso da Figura 1.1 temos o título, os autores, o preço e assim por diante. O suporte a variações na estrutura se deve ao fato dos padrões de extração serem independentes um dos outros. Já o suporte a tipos aninhados se deve à utilização do algoritmo HotCycles, que infere uma estrutura plausível para montar os objetos de acordo com a distribuição dos valores atômicos extraídos, não sendo necessária qualquer definição de estrutura por parte do usuário.

Mais recentemente, novas abordagens têm proposto a geração completamente automática de extratores. Na ferramenta RoadRunner [Crescenzi et al., 2001], o problema de geração de extratores é encarado como um problema de inferência de gramática dado exemplos positivos. Cada página é considerada como um exemplo positivo de uma gramática regular que vai sendo progressivamente inferida de acordo com a análise das incompatibilidades estruturais encontradas durante o alinhamento dos marcadores HTML. Essas incompatibilidades permitem ao algoritmo a descoberta de características estruturais tais como tuplas, listas e opcionais. Essa abordagem considera somente delimitadores fortes (marcadores HTML) e a estrutura gerada para os dados respeita a mesma estrutura do documento. Além disso, por ser uma abordagem automática, tende a extrair elementos que não são relevantes ao usuário (menus, datas, etc.).

Outras abordagens para a geração automática de extratores utilizando distância de edição em árvores têm sido propostas para a utilização em domínios específicos [Reis et al., 2004]. Essas abordagens consideram um documento HTML como uma árvore e identificam estruturas similares de forma a agrupar páginas semelhantes e encontrar uma representação genérica para sua estrutura e assim extrair os dados.

Seguindo a linha dos métodos automáticos, temos ainda o trabalho desenvolvido por Arasu & Molina [Arasu and Garcia-Molina, 2003] que se baseia no conceito de *classes de equivalências*. Apesar de ser mais flexível que a ferramenta RoadRunner, possibilitando delimitadores textuais, não possui suporte para delimitadores fracos e sofre do mesmo problema com relação à estrutura dos dados extraídos e extração de dados irrelevantes.

1.3 Principais Contribuições

A idéia de se encarar o problema de extração de valores atômicos como um problema de inferência de expressões regulares não é nova. Porém, a utilização do alinhamento múltiplo de seqüências para propor uma melhor distribuição dos delimitadores durante a generalização das padrões de extração se configura numa abordagem inovadora. Apesar do problema do alinhamento múltiplo constituir por si só um desafio, ele apresenta inúmeros algoritmos aplicáveis que são adequados à nossa modelagem do problema de extração de dados.

A utilização de uma função de similaridade que considera o casamento aproximado entre os delimitadores permite que o problema da existência de contexto fraco seja adequadamente tratado. A incorporação de evidências relacionadas ao domínio dos dados também amplia a expressividade do método de extração de valores atômicos.

Durante o processo de inferência de padrões de extração é necessário que sejam fornecidos exemplos de treinamento dos atributos que se deseja extrair. Acontece, porém, que normalmente na Web esses valores atômicos apresentam-se relacionados, formando uma estrutura que pode ser representada por um tipo aninhado. Normalmente essa estrutura é fornecida pelo usuário.

Em nossa abordagem, aplicamos e estendemos o algoritmo HotCycles [da Silva, 2002] que é capaz de inferir automaticamente estruturas plausíveis para agregar esses valores atômicos. Assim o usuário não precisa mais especificar manualmente a estrutura dos objetos. O algoritmo considera a disposição dos valores atômicos existentes no documento de modo a identificar a ocorrência de padrões que indiquem a existência de listas ou tuplas.

A principal contribuição deste trabalho é, portanto, o desenvolvimento de um novo método de extração de dados que trata convenientemente o problema do contexto fraco, onde o usuário se limita a fornecer exemplos das unidades de informação de interesse de forma a possibilitar a inferência das expressões de extração, onde a estrutura aninhada dos objetos é automaticamente inferida pela análise de padrões na distribuição dos valores atômicos extraídos.

1.4 Organização da Dissertação

No Capítulo 2 é apresentado em detalhes o método de inferência de expressões regulares para os valores atômicos constituintes das páginas HTML. Será discutida a utilização de algoritmos de

alinhamento de seqüências durante a inferência dos padrões de extração para os valores atômicos, bem como o processo de extração dos dados de interesse.

O Capítulo 3 é reservado para o estudo detalhado do método de montagem dos objetos extraídos de acordo com uma estrutura plausível que vai sendo automaticamente gerada pelo algoritmo HotCycles.

No Capítulo 4 são apresentados os resultados experimentais obtidos pelos métodos que constituem nossa abordagem. A metodologia de avaliação é apresentada bem como os resultados dos experimentos para cada um dos métodos discutidos. Uma avaliação conjunta dos métodos que compõem nossa abordagem é também apresentada de modo a verificar sua viabilidade como um todo.

Finalmente, no Capítulo 5, são apresentadas as conclusões obtidas após a concepção, prototipação e avaliação da nossa abordagem de extração, onde apresentamos também as possibilidades de extensão e trabalhos futuros relacionadas.

Capítulo 2

Extração dos Valores Atômicos

A primeira etapa do processo de extração de valores atômicos assume que um conjunto de exemplos foi previamente selecionado por um usuário. Esses exemplos servirão para a inferência de padrões de extração utilizadas pelo extrator para recuperar os valores atômicos nas páginas da Web. Com os valores atômicos extraídos inicia-se a segunda etapa da abordagem, onde esses elementos até então dispersos, são combinados de modo a constituir um objeto complexo.

O processo de geração do extrator envolve uma série de passos que são executados progressivamente até que se obtenha um padrão de extração que incorpore informações contextuais de todos os exemplos de treinamento fornecidos para um atributo. As fases necessárias à geração desses padrões de extração são abordadas neste capítulo.

A seguir formalizamos algumas definições que já foram intuitivamente introduzidas no capítulo anterior e que serão utilizadas na descrição do nosso método de extração de valores atômicos.

Definição 1 *Seja H uma página HTML. Um **valor atômico** A é representado por um par $\langle \tau, v \rangle$ onde τ é o nome de um atributo e v é uma cadeia de caracteres de H que representa uma instância de τ .*

Considerando a Figura 1.1 é possível, por exemplo, identificar os seguintes atributos: **título**, **autor** e **preço**. Para cada um desses atributos existe um valor atômico que o representa, respectivamente: **Software Engineering**, **Ian Sommerville** e **94.62**.

Definição 2 *Seja $A = \langle \tau, v \rangle$ um valor atômico ocorrendo em uma página HTML H . Definimos $\epsilon = \langle p, v, s \rangle$ como sendo um **exemplo de treinamento** onde p é um prefixo de v e s um sufixo*

de v na página H .

Algoritmo 1: Algoritmo para geração de um padrão de extração

Data: ex é um conjunto de exemplos de treinamento para o atributo τ

```

1 begin
2   contextos  $\leftarrow \emptyset$  ;
3   foreach exemplo de treinamento  $\epsilon \in ex$  do
4     | contextos  $\leftarrow$  contextos  $\cup$  identificarDelimitadores ( $\epsilon$ ) ;
5   exp  $\leftarrow$  inferirExpressão (contextos) ;
6   regex  $\leftarrow$  toRegEx (exp) ;
7 end
```

O Algoritmo 1 apresenta uma visão geral dos passos necessários para a geração dos padrões de extração que irão constituir um extrator para um atributo τ . Nas Linhas 3 e 4 realizamos a identificação e separação dos delimitadores que constituem o prefixo e o sufixo dos exemplos de treinamento fornecidos, esse processo é discutido na Seção 2.2. Em seguida, na Linha 5, realizamos a inferência do padrão de extração através de alinhamentos progressivos dos delimitadores que circundam os exemplos de treinamento. A ordem desses alinhamentos respeita uma política que considera primeiro os contextos mais semelhantes. Esse passo do algoritmo será discutido nas Seções 2.3 a 2.8.

Uma vez que o padrão de extração foi gerado é necessário transformá-lo em uma expressão regular para que esta possa ser então avaliada no momento da extração, o que será discutido na Seção 2.9. Essa expressão regular será utilizada pelo extrator. O processo de extração dos valores atômicos existentes nas páginas será apresentado na Seção 2.10.

Inicialmente apresentamos na Seção 2.1 alguns conceitos básicos necessários à discussão da primeira etapa de nossa abordagem. Muitos desses conceitos estão relacionados ao problema de alinhamento de seqüências de DNA estudado na área de biologia computacional e serão aqui adaptados.

2.1 Conceitos Básicos

Nas seções a seguir são apresentadas a definição clássica do problema do alinhamento múltiplo de seqüências e da principal técnica utilizada na literatura para resolver este problema que serve de base para o nosso método de geração de padrões de extração.

2.1.1 Problema do Alinhamento Múltiplo de Seqüências

O problema do alinhamento múltiplo é uma generalização natural do problema do alinhamento entre duas seqüências [Gusfield, 1997]. Este problema deve ser primeiramente definido uma vez que se constitui na forma mais básica do problema do alinhamento múltiplo.

Sejam S_1 e S_2 duas seqüências. Um **alinhamento** dessas duas seqüências é obtido pela inserção de espaços “-”, também conhecidos como **gaps**, em quaisquer posições ao longo de S_1 e S_2 . Ao final desse processo $|S_1| = |S_2|$ e podemos então posicionar as duas seqüências resultantes uma embaixo da outra, de tal forma que cada elemento ou espaço em uma seqüência esteja alinhado a um único elemento ou a um único espaço na outra seqüência [Gusfield, 1997]. Além disso não é permitido que um espaço de uma seqüência esteja alinhado com um espaço em outra seqüência.

Um **alinhamento múltiplo** de $k > 2$ seqüências $\{S_1, S_2, \dots, S_k\}$ é uma generalização do alinhamento entre duas seqüências. Espaços são inseridos em posições arbitrárias ao longo de quaisquer das k seqüências, de tal forma que as seqüências resultantes possuam um mesmo tamanho l . As seqüências podem então ser dispostas em k linhas e l colunas, de tal forma que cada elemento ou espaço de cada seqüência está em uma única coluna [Gusfield, 1997]. Além disso não é permitido que uma coluna seja constituída apenas por espaços. A Figura 2.1 apresenta um exemplo de alinhamento múltiplo entre seqüências de DNA.



Figura 2.1: Alinhamento múltiplo de DNA

Devemos considerar entretanto que o processo de alinhamento visa inserir espaços nas seqüências de forma a facilitar a visualização do relacionamento formado entre as seqüências que foram alinhadas. Em nosso caso específico tal alinhamento deve facilitar o processo de comparação entre os delimitadores de constituem o prefixo e o sufixo de um exemplo de treinamento de forma a possibilitar a geração de um padrão de extração.

Conforme apresentado na Figura 2.2 percebemos que o alinhamento múltiplo dos delimitadores que constituem o prefixo e o sufixo dos exemplos de treinamento para um atributo τ qualquer

```

...</B> (      1988      ) . <<dados>>, <I>      Psychoanal. Rev      </I>, ...
...</B> (      1974      ) . <<dados>>, <I> J. Amer. Psychoanal. Assn      </I>, ...
...      (<B> <I>1999</I> </B> ) . <<dados>>, <I>      Child      Analysis </I>, ...
...</B> (<B> <I>1999</I> </B> ) . <<dados>>, <I>      Psychoanal. Books      </I>, ...

```

```

...</B>? (<B>?<I>?\d+</I>?</B>?) . <<mask>> , <I>[\w+\.\s]*? \w+      \.?\w+      \.??</I>, ...

```

onde <<mask>> = ((? : [^<>] | (? : </?I\b[^<>]*>) | (? : </?B\b[^<>]*>)) +) ?

Figura 2.2: Alinhamento múltiplo de exemplos

facilita a visualização dos relacionamentos entre dado de interesse e contexto de extração. Isso pode ser utilizado para a geração de um padrão de extração.

2.1.2 Valor de Similaridade de um Alinhamento

O valor de similaridade entre duas seqüências S_1 e S_2 e o alinhamento ótimo relacionado, pode ser calculado por programação dinâmica [Gusfield, 1997] através da resolução da relação de recorrência da Equação 2.1.

$$s[i, j] = \begin{cases} 0 & \text{se } i = 0 \text{ ou } j = 0 \\ \max(s[i-1, j-1] + \text{sim}(x_i, y_j), s[i-1, j], s[i, j-1]) & \text{se } i, j > 0 \end{cases} \quad (2.1)$$

Se S_1 e S_2 possuem tamanho m e n respectivamente, o valor do alinhamento ótimo é dado por $s[m, n]$. Esse valor, assim como o cálculo de todos os subproblema, pode ser obtido em $O(mn)$ [Gusfield, 1997]. Conforme apresentado em [Leiserson et al., 2001] utilizando-se apontadores durante a resolução dos subproblemas é possível reconstruir a solução ótima calculada através da programação dinâmica.

Percebemos pela Equação 2.1 que para realizarmos o alinhamento entre duas seqüências é necessário a definição de um esquema de pontuação apropriado que reflita o relacionamento entre cada par de elementos alinhados. Dependendo do que se deseja alinhar, esquemas de pontuação específicos podem ser utilizados [Gusfield, 1997]. Em nosso caso desejamos alinhar os delimitadores dos prefixos e sufixos dos exemplos de treinamento, para tanto, devemos definir um esquema de pontuação que reflita o relacionamento entre esses delimitadores.

2.1.3 Técnica do Alinhamento Múltiplo Progressivo

Assim como existe um método de programação dinâmica para se alinhar otimamente duas seqüências de acordo com um esquema de pontuação definido, é possível generalizar o método para calcular o alinhamento ótimo entre k seqüências. Estender o algoritmo de programação dinâmica para considerar alinhamento múltiplo de k seqüências implica em um algoritmo de complexidade $O(n^k)$, considerando seqüências de tamanho n , que é inviável na prática.

Por esse motivo diversos métodos heurísticos de alinhamento múltiplo de seqüência têm sido propostos com o objetivo de realizar alinhamentos mais rápidos sem que isso implique em degradação na qualidade dos resultados. A maioria desses métodos, assim como o nosso, se baseia na estratégia do alinhamento progressivo [Feng and Doolittle, 1987].

A proposta do alinhamento progressivo é realizar sucessivos alinhamentos entre os pares de seqüências. O método realiza inicialmente o alinhamento entre duas seqüências que foram escolhidas. Em seguida uma nova seqüência é escolhida e alinhada com o primeiro alinhamento realizado, ou então, um novo par de seqüências é escolhido e alinhado. Esse processo se repete até que todas as seqüências fornecidas tenha sido alinhadas [Durbin et al., 1998].

Muitos métodos de alinhamento múltiplo são baseados em estratégias progressivas. A mais representativa dessas estratégias foi proposta por Feng e Doolittle [Feng and Doolittle, 1987]. Nessa estratégia é construída uma **árvore guia** que determina a ordem com que cada par de seqüências será alinhado. Essa estratégia é formada pelas seguintes etapas:

- **Montagem da Matriz de Distâncias.** É o passo inicial dessa estratégia, onde cada par de seqüências é alinhado com o objetivo de gerar uma matriz de distâncias com o valor de similaridade de cada par de seqüências.
- **Geração da Árvore Guia.** Uma árvore binária é gerada a partir dos valores de similaridade contidos na matriz de distância de forma a determinar a ordem com que os alinhamentos serão progressivamente realizados. Diversos métodos podem ser utilizados para a construção da árvore guia, tais como UPGMA [Smith and Waterman, 1981] ou *Neighbor-Joining* [Saiton and Nei, 1987]. Os nós internos da árvore guia representam alinhamentos múltiplos parciais, as folhas representam as seqüências originais. O nó raiz representa o alinhamento múltiplo final. Os nós mais afastados da raiz representam os pares de seqüências mais similares [Durbin et al., 1998].

- **Alinhamento Progressivo.** Uma vez determinada a ordem dos alinhamentos pela árvore guia, podemos realizar progressivamente os alinhamentos. Primeiramente as seqüências mais similares são alinhadas. Em seguida, as duas próximas seqüências mais similares são alinhadas ou uma nova seqüência é introduzida no alinhamento múltiplo das seqüências anteriores. Esse processo continua até todas as seqüências tenha sido alinhadas, dando origem ao alinhamento múltiplo final.

A técnica de alinhamentos progressivos utiliza uma política gulosa onde uma vez introduzido um espaço, este não pode ser removido por nenhum alinhamento subsequente, todos os espaços introduzidos são preservados até o alinhamento múltiplo final. A taxa de erro introduzida pelo alinhamento progressivo em cada etapa tende a diminuir no caso de seqüências mais similares, e aumentar quando alinhamos seqüências mais divergentes [Feng and Doolittle, 1987].

Dessa forma determinar a ordem dos alinhamentos é de fundamental importância no processo de resolução progressiva do alinhamento múltiplo. O ideal é que seqüências mais similares sejam alinhadas primeiramente, deixando para o final as mais divergentes de forma a reduzir o erro introduzido pelo processo heurístico de solução.

2.2 Identificação dos Delimitadores

O primeiro passo para a geração dos padrões de extração é a identificação dos delimitadores que envolvem os exemplos de treinamento selecionados pelo usuário. Cada exemplo possui um par de afixos, que definem o contexto sintático onde o mesmo está inserido. Em nosso trabalho, o tamanho desses afixos é pré-definido como sendo formado pelos n caracteres mais próximos ao dado de interesse.

Inicialmente esses afixos são cadeias de caracteres no formato HTML, é necessário realizar a identificação e separação dos possíveis delimitadores que irão compor o padrão de extração. Para tanto, identificamos dois tipos de delimitadores:

- **Delimitadores de Marcação** - que representam os elementos HTML existentes no documento, assim como seus atributos;
- **Delimitadores de Texto** - que representam o conteúdo visível para o usuário, ou seja, as cadeias de caracteres contidas entre os marcadores HTML.

Aos delimitadores de marcação associamos identificadores, que são os nomes dos respectivos marcadores HTML, e uma lista de pares que correspondem aos atributos¹ e valores que podem estar presentes. Aos delimitadores de texto por sua vez, são associados um conteúdo que é justamente o valor da cadeia de caracteres que ele representa e o alfabeto a que essa cadeia de caracteres pertence. Consideramos aqui os seguintes alfabetos: dígitos, letras maiúsculas, letras em geral e caracteres de pontuação(.,!()[]{}~\$%£#...).

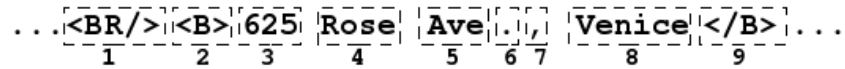
...


Figura 2.3: Processo de identificação dos delimitadores

A Figura 2.3 exemplifica os delimitadores identificados após esse processo. Nela é possível identificar 9 delimitadores. Os delimitadores 1,2 e 9 são delimitadores de marcação, e o restante são delimitadores de texto. Em relação aos delimitadores de texto temos que o alfabeto do delimitador 3 é “dígitos”, os delimitadores 4, 5 e 8 possuem o alfabeto “letras”, e para os delimitadores 6 e 7 utilizamos o alfabeto “ponto” e “vírgula” respectivamente. Usando esta política de identificação, os exemplos de treinamento são transformados em exemplos delimitados.

Definição 3 *Seja $\epsilon = \langle p, v, s \rangle$ um exemplo de treinamento para um atributo τ qualquer, definimos $\varepsilon = \langle p_t, v, s_t \rangle$ como sendo o **exemplo delimitado** de ϵ , onde $p_t = \text{delimitadores}(p)$, $s_t = \text{delimitadores}(s)$ e delimitadores é uma função de identificação que opera como mostrado acima.*

Realizamos esse processo de identificação de delimitadores para todos os exemplos que foram selecionados pelo usuário para determinado atributo, para, em seguida, dar início à inferência do padrão de extração, como descrito a seguir.

2.3 Alinhamento dos Delimitadores

O processo de inferência realiza inicialmente o alinhamento dos exemplos delimitados. Esse alinhamento prévio permite que delimitadores semelhantes sejam alinhados e delimitadores não

¹Neste contexto, nos referimos aos atributos do marcador HTML

semelhantes sejam afastados, de maneira a permitir a produção de um padrão de extração através da generalização dos exemplos alinhados.

Sejam $\varepsilon_1 = \langle p_1, v_1, s_1 \rangle$ e $\varepsilon_2 = \langle p_2, v_2, s_2 \rangle$ dois exemplos delimitados para um dado atributo τ . Podemos informalmente definir um **alinhamento de contextos** como sendo a disposição dos delimitadores de p_1 e p_2 bem como os de s_1 e s_2 de forma que tal alinhamento torne possível o processo de comparação entre os dois exemplos.

Figura 2.4: Um exemplo de alinhamento de contextos

A Figura 2.4 exemplifica a noção de alinhamento². Nela podemos identificar dois exemplos delimitados alinhados. Apenas o prefixo e o sufixo dos exemplos são exibidos. Percebemos pela figura que o alinhamento entre dois exemplos de treinamento facilita a sua comparação, que é necessária durante o processo de generalização do padrão de extração.

Após o alinhamento dos exemplos é possível classificar os delimitadores de acordo com sua frequência nos contextos alinhados, como definido a seguir.

Definição 4 *Sejam t_1 e t_2 um par de delimitadores alinhados. Dizemos que t_1 e t_2 são **delimitadores fortes** sss $t_1 = t_2$.*

Definição 5 *Sejam t_1 e t_2 um par de delimitadores alinhados. Dizemos que t_1 e t_2 são **delimitadores fracos** sss $dom(t_1) = dom(t_2)$ e $t_1 \neq t_2$, onde dom é o alfabeto do delimitador (dígitos, letras, caracteres de pontuação, etc) no caso de delimitadores de texto ou o identificador de elemento HTML no caso de delimitador de marcação.*

Definição 6 *Seja t um delimitador qualquer. Dizemos que t é um **delimitador opcional** sss t não estiver alinhado com nenhum outro delimitador.*

Para ilustrar estes conceitos, considere o exemplo apresentado na Figura 2.5. Nesta figura os delimitadores identificados como fortes são representados por um quadrado, os identificados como fracos são representados por um círculo e os delimitadores opcionais identificados por um triângulo. O delimitador $$ é forte, porém o alinhamento possui também delimitadores

²Trecho de página extraída de <http://www.apsa.org>

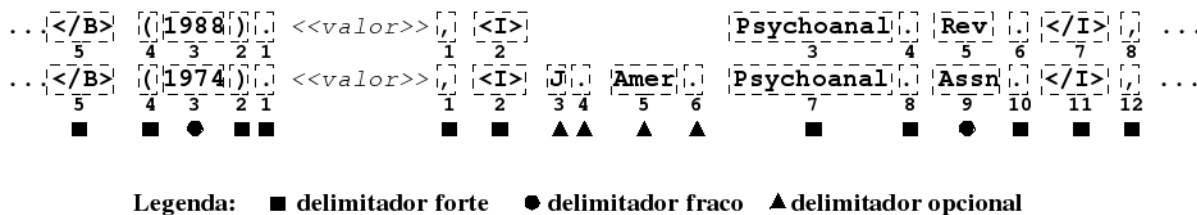


Figura 2.5: Exemplo de delimitadores

fracos que devem ser considerados ao generalizar um padrão de extração. Um delimitador forte aparece em todos os exemplos e é fixo tal como `</B>`. Delimitadores fracos são aqueles que têm importância semântica como `1988` e `1974` ou `Assn` e `Rev`. Eles não são fixos porque variam dependendo do contexto, mas são obrigatórios. Finalmente há também alguns delimitadores que são opcionais como `J` e `Amer`.

Uma forma de encontrar o melhor alinhamento entre dois exemplos de treinamento é reduzir esse problema ao problema de otimização conhecido como *alinhamento entre duas seqüências*, que pode ser resolvido eficientemente usando programação dinâmica [Gusfield, 1997]. O objetivo aqui é fornecer um bom alinhamento entre os exemplos, que será em seguida usado para generalizar um padrão de extração.

2.4 Esquema de Pontuação

Conforme exposto na Seção 2.1 a definição de um esquema de pontuação para o algoritmo de alinhamento de seqüências é fundamental. Em nosso caso, tal esquema deve refletir o relacionamento existente entre os delimitadores que constituem os prefixos e os sufixos dos exemplos de treinamento.

Considere dois delimitadores de marcação. Se ambos forem exatamente iguais o valor de similaridade entre eles é bastante alto. Se possuírem valores de atributos diferentes o valor de similaridade ainda é alto, porém se possuírem atributos diferentes esse valor começa a decair. Em se tratando de delimitadores que representam marcadores HTML distintos, o valor de similaridade deve refletir uma penalização.

A mesma idéia se aplica à similaridade entre delimitadores de texto. Se ambos possuírem o mesmo conteúdo a similaridade é alta, se possuírem o mesmo alfabeto ainda sim temos uma similaridade razoável. Porém, se apresentarem alfabetos disjuntos, o valor de similaridade deve

indicar uma penalização. O valor de similaridade entre delimitadores de texto e delimitadores de marcação deve também indicar uma penalização.

O Algoritmo 2 utiliza as idéias apresentadas nessa seção. Ele recebe como parâmetros dois delimitadores que estão sendo comparados e mais dois índices que indicam a posição desses delimitadores em relação ao dado que se deseja extrair e retorna como resultado o valor de similaridade entre os dois delimitadores.

A idéia do valor de similaridade é refletir quão parecidos são dois delimitadores. Marcadores HTML se apresentam como delimitadores fortes e o alinhamento de tais delimitadores deve ser priorizado. Cadeias de caracteres iguais também formam bons delimitadores e portanto devem ser também priorizados. O Algoritmo 2 apresenta um esquema de pontuação que reflete essas idéias.

Nas Linhas 2 a 11 é tratado o caso em que ambos os delimitadores são marcadores. Inicialmente é verificado se os dois delimitadores representam um mesmo marcador HTML. Nesse caso é realizado um novo teste para verificar se os dois delimitadores são iguais, atribuindo o valor 5 para o alinhamento, ou se ambos, apesar de diferentes, possuem os mesmos atributos, quando então atribuímos o valor 2. Caso os delimitadores de marcação tenham atributos diferentes, penalizamos o alinhamento com o valor -1. Se os delimitadores de marcação indicarem marcadores HTML diferentes, é atribuída a penalização de -1.

As Linhas 12 a 18 consideram o caso do alinhamento entre delimitadores de texto. Três possibilidades podem ocorrer: para delimitadores iguais atribuímos o valor 5, delimitadores com o mesmo alfabeto utilizamos o valor 1 e penalizamos delimitadores com alfabetos disjuntos com -1. Delimitadores de tipos diferentes também são penalizados com o valor -1.

A função de similaridade `sim` do algoritmo de alinhamento, também chamada de função de *pontuação* ou *score*, considera portanto os domínios dos delimitadores que estão sendo alinhados de modo a possibilitar o casamento aproximado entre eles.

2.5 Posicionamento dos Delimitadores

Um outro fator que consideramos relevante quando alinhamos seqüências de delimitadores, é a posição relativa dos mesmos nas seqüências. Verificamos que os delimitadores mais próximos ao valor a ser extraído apresentam-se como delimitadores mais expressivos do que aqueles mais

Algoritmo 2: Função de similaridade

Data: T_a e T_b dois delimitadores sendo alinhados
Data: p_a e p_b são as posições dos delimitadores em relação ao dado de interesse
Data: score é o valor de similaridade

```

1 begin
2   if  $T_a$  e  $T_b$  são delimitadores de marcação then
3     if  $T_a$  e  $T_b$  possuem o mesmo identificador e os mesmo marcadores de tag then
4       if conteúdo de  $T_a$  = conteúdo de  $T_b$  then
5         | score  $\leftarrow$  5;
6       else if o nome dos atributos de tag de  $T_a$  são os mesmos de  $T_b$  then
7         | score  $\leftarrow$  4;
8       else
9         | score  $\leftarrow$  -1;
10    else
11      | score  $\leftarrow$  -1;
12  else if  $T_a$  e  $T_b$  são delimitadores de texto then
13    if conteúdo de  $T_a$  = conteúdo de  $T_b$  then
14      | score  $\leftarrow$  5;
15    else if domínio de  $T_a$  = domínio de  $T_b$  then
16      | score  $\leftarrow$  1;
17    else
18      | score  $\leftarrow$  -1;
19  else
20    | score  $\leftarrow$  -1;
21  return score ;
22 end

```

distantes, logo a função de similaridade deve considerar também a distância em que determinado alinhamento de delimitadores ocorreu. Delimitadores mais próximos devem ser priorizados.

Um alinhamento de delimitadores que ocorre mais próximo ao valor a ser extraído possui importância maior do que o alinhamento de delimitadores que está mais distante. Essa característica é utilizada na função de similaridade que recebe como parâmetros os dois delimitadores sendo alinhados e as posições de cada delimitador em relação ao dado de interesse.

As posições dos delimitadores são utilizadas para calcular um peso baseando-se na distância relativa entre os delimitadores que estão sendo alinhados e na distância desse alinhamento até o dado de interesse. Quanto mais distante está o alinhamento do dado de interesse e quanto maior for a distância relativa entre os delimitadores, menor é o peso que lhe é atribuído. Quanto mais próximo estiver o alinhamento maior será o seu peso. Essa proximidade é então considerada

pela função de similaridade.

2.6 Solução por Programação Dinâmica

Tomando por base a Equação 2.1 e considerando a existência de apenas $\Theta(mn)$ subproblemas distintos, podemos usar a técnica de programação dinâmica [Leiserson et al., 2001] para calcular as soluções de maneira *bottom-up*. Isso é possível pois o problema do alinhamento entre duas seqüências apresenta a propriedade da sub-estrutura ótima [Gusfield, 1997].

O Algoritmo 3 utiliza duas seqüências de delimitadores $X = \langle x_1, \dots, x_m \rangle$ e $Y = \langle y_1, \dots, y_n \rangle$. Ele armazena os valores de similaridade $s[i, j]$ na tabela **score** além de manter a tabela **path** para simplificar o processo de reconstrução da solução ótima.

Algoritmo 3: Alinhamento de seqüência usando programação dinâmica

Data: X, Y são duas seqüências de delimitadores
Data: score é a matriz que armazenará a similaridade total do alinhamento encontrado para cada subproblema
Data: path é a matriz que armazenará as decisões tomadas em cada subproblemas
Data: match é a variável que armazenará o valor da similaridade entre dois delimitadores
Data: gapX e gapY são as variáveis que irão armazenar os valores calculados para o alinhamentos entre delimitadores e espaços

```

1 begin
2   for i ← 1 to size (X) do
3     | score [i][0] ← 0
4   for i ← 1 to size (Y) do
5     | score [0][i] ← 0
6   for i ← 1 to size (X) do
7     for j ← 1 to size (Y) do
8       match ← score [i-1][j-1] + sim (pegarDelimitador (X, i), pegarDelimitador
9         (Y, j), i, j) ;
10      gapX ← score [i][j-1] ;
11      gapY ← score [i-1][j];
12      score [i][j] ← max (match, gapX, gapY);
13      if score [i][j] = match then
14        | path [i][j] ← “Delimitadores Similares”;
15      else if score [i][j] = gapX then
16        | path [i][j] ← “Inserir Espaço em X”;
17      else
18        | path [i][j] ← “Inserir Espaço em Y”;
19    end
20  end
21 end

```

Na Linha 8 é calculado o caso de casamentos exatos ou aproximados entre os delimitadores,

conforme a função de similaridade apresentada na Seção 2.4. Nas Linhas 9 e 10 são verificados os casos de utilização de espaços nas seqüências X e Y respectivamente.

Verificamos no Algoritmo 3 a existência de dois laços de repetição aninhados que são executados conforme o tamanho das duas seqüências que estão sendo alinhadas. Isso indica uma complexidade computacional na ordem de $O(mn)$. Se consideramos que as duas seqüências possuem tamanhos semelhantes temos que a complexidade do algoritmo de alinhamento é da ordem de $O(n^2)$. A tabela `path` criada durante o cálculo da solução ótima pode ser utilizada para se reconstruir um alinhamento ótimo, segundo técnicas descritas em [Leiserson et al., 2001].

2.7 Generalização das Expressões de Consenso

Após o alinhamento dos delimitadores que constituem os prefixos e os sufixos dos exemplos de treinamento, inicia-se o processo de generalização de uma expressão que irá representar o consenso entre os dois exemplos alinhados. Essa etapa é necessária pois o nosso objetivo é gerar um padrão de extração capaz de extrair valores atômicos que aparecem em contextos similares através de uma expressão regular. O alinhamento dos exemplos de treinamento é na verdade utilizado para viabilizar o processo de generalização.

Para cada conjunto de delimitadores alinhado é gerada uma expressão capaz de representar os delimitadores envolvidos no alinhamento. A essa expressão é atribuído uma indicação de seu tipo (fixo, obrigatório ou opcional) de acordo com a frequência dos delimitadores nas seqüências alinhadas. As regras de generalização para cada conjunto de delimitadores alinhados são as seguintes:

1. **Alinhamento de delimitadores de marcação:** Neste caso deve-se verificar se os delimitadores de marcação são idênticos ou se existe variação nos atributos HTML, dependendo desta situação, atribui-se a expressão gerada a indicação de “valor fixo”, quando o conteúdo dos marcadores é exatamente igual, ou “valor obrigatório” quando os valores dos atributos dos marcadores HTML são diferentes.
2. **Alinhamento de delimitadores de texto:** Aqui verifica-se se as cadeias de caracteres possuem o mesmo valor implicando numa expressão com a indicação de “valor fixo”. Quando os delimitadores de texto possuem o mesmo alfabeto para os seus dados, é atribuído a indicação “valor obrigatório” para a expressão.

3. **Alinhamento de delimitadores (texto ou marcação) com espaços:** Nesse caso temos o aparecimento de delimitadores opcionais no contexto. A expressão gerada para esse alinhamento possui a indicação de “valor opcional”, indicando a possibilidade de sua ausência no momento da extração.

Ao término do processo de generalização, uma expressão de consenso é produzida pela concatenação das expressões unitárias que foram geradas para cada conjunto de delimitadores alinhado. Essa expressão pode então ser novamente refinada pela utilização de um novo exemplo. Neste caso, a expressão de consenso é instanciada formando o que chamamos de **exemplo de consenso** que é então realinhado com o novo exemplo fornecido e em seguida generalizado.

2.8 Processo de Inferência

Foi apresentado na Seção 2.7 o processo de geração de uma expressão de consenso a partir da utilização de dois exemplos delimitados. Acontece porém que o processo ideal para geração de um padrão de extração envolve o alinhamento múltiplo dos delimitadores que formam o contexto dos exemplos de treinamento, ou seja, os prefixos e os sufixos.

O alinhamento múltiplo de contextos é uma generalização natural para o alinhamento de dois contextos. Conforme apresentado na Seção 2.1 vimos que esse problema pode ser solucionado em $O(mn)$ por uma estratégia de programação dinâmica [Gusfield, 1997]. Estender o algoritmo de programação dinâmica para considerar alinhamento múltiplo de k contextos implica em um algoritmo de complexidade $O(n^k)$, considerando contextos de tamanho n . Tal fato nos levou à adoção de uma estratégia para realização progressiva dos alinhamentos [Feng and Doolittle, 1987] onde a cada iteração combinamos um par de exemplos de treinamento, um exemplo de treinamento e uma expressão de consenso ou um par de expressões de consenso de maneira a cobrir progressivamente todos os exemplos fornecidos.

Após o alinhamento de um par de exemplos, é iniciada a generalização de uma nova expressão de consenso que combina as informações contidas nos dois exemplos anteriores. Como foi visto anteriormente, no processo de generalização, percorremos as seqüências alinhadas e, para cada par de delimitadores que foram alinhados, generalizamos uma expressão única capaz de identificar esses dois delimitadores.

A ordem com que realizamos os progressivos alinhamentos influenciam no resultado final. Se combinarmos inicialmente exemplos muito diferentes, temos a tendência de banalizar a expressão de consenso resultante. Porém, se escolhermos inicialmente exemplos mais similares, conseguimos um processo de generalização que reduz o erro introduzido, o que leva a geração de uma expressão de consenso bem mais precisa.

A política utilizada para a seleção dos exemplos ou consensos mais similares baseia-se na geração de uma árvore guia para determinar a ordem com que os alinhamentos serão realizados [Durbin et al., 1998]. A montagem de nossa árvore guia é uma variação da técnica descrita em [Smith and Waterman, 1981]. Inicialmente é criada uma matriz de distâncias com as similaridades de cada par de alinhamento possível. Feito isso, é possível localizar o par de exemplos mais similar. Esse par de exemplos é então alinhado e em seguida generalizado em uma expressão de consenso que incorpora as informações contidas nos exemplos alinhados.

Os dois exemplos que foram utilizados são removidos da matriz de distâncias dando lugar a nova expressão de consenso que é então inserida nesta matriz. Em seguida é calculado o valor de similaridade para cada alinhamento dessa nova expressão com os exemplos já existentes. O processo se repete até que exista apenas uma única expressão de consenso que formará o padrão de extração utilizado pelo extrator. Esse processo é descrito pelo Algoritmo 4.

Algoritmo 4: Algoritmo de alinhamento progressivo

Data: L é uma lista que contém os exemplos atômicos para um determinado atributo
Data: $score$ é uma matriz que mantém as similaridades entre as expressões parciais

```

1 begin
2   for  $i \leftarrow 0$  to  $size(L)$  do
3     for  $j \leftarrow i + 1$  to  $size(L)$  do
4        $score_{i,j} \leftarrow$  score do alinhamento entre a  $i$ -ésima e a  $j$ -ésima expressões parciais;
5   while  $size(L) > 1$  do
6      $(i, j) =$  índices do maior valor na matriz  $score$  ;
7      $consenso \leftarrow$  expressão de consenso entre o  $i$ -ésima e o  $j$ -ésima exemplos ou expressões de consenso;
8     remove ( $L, i$ );
9     remove ( $L, j$ );
10    adicionar ( $L, consenso$ );
11    Recalcular o valor de similaridade para a expressão de consenso recém criada e os outros exemplos e expressões de consenso já existentes;
12 end
```

Nas Linha 2 a 4 é criada a matriz de distâncias que contém o valor de similaridade para cada

par de alinhamento entre os exemplos fornecidos. Nas Linhas 5 a 11 o padrão de extração vai sendo progressivamente gerado. Inicialmente, na Linha 6, verifica-se qual par de alinhamentos apresenta a maior similaridade, em seguida uma expressão de consenso baseada nesse par de alinhamentos é gerada na Linha 7.

Nas Linhas 8 e 9 removemos as duas seqüências utilizadas para a geração da expressão de consenso e adicionamos a nova expressão à lista de exemplos. Note-se que agora existem não só exemplos delimitados na lista mas também expressões de consenso parciais. Finalmente, na Linha 11 atualizamos a matriz de distâncias e recalculamos o novo valor de similaridade entre os exemplos antigos e a expressão consenso gerada. Esse processo se repete até que só exista uma única expressão de consenso, que será transformada em uma expressão regular que pode ser utilizada para extrair os dados de interesse.

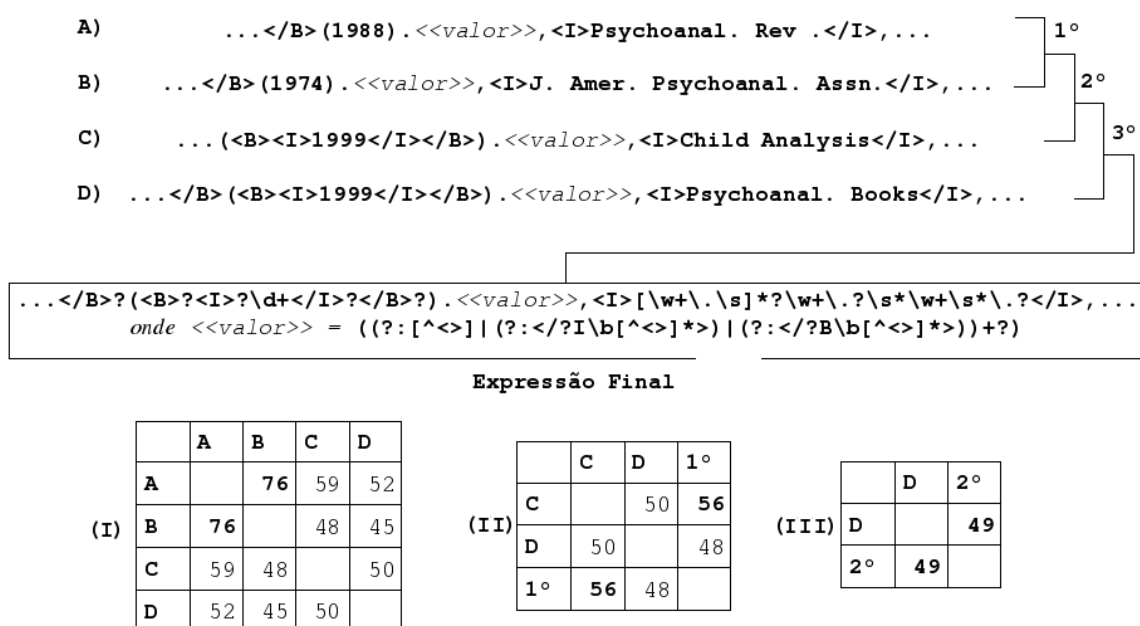


Figura 2.6: Exemplo de uma árvore guia.

A ordem com que os diferentes exemplos são combinados dão origem ao que se chama **árvore guia** [Durbin et al., 1998]. A Figura 2.6 exemplifica este processo. Nela temos os exemplos A,B,C,D, a matriz de distâncias inicial é representada em (I), onde verifica-se que as seqüências mais similares são A e B (76). Uma nova expressão de consenso é gerada a partir dessas seqüências e a matriz é atualizada (II). Em seguida verifica-se que o par de seqüências mais similar é formado por C e a expressão de consenso 1 gerada no passo anterior. Uma nova expressão de

consenso é então gerada e a matriz atualizada. Por fim, restam apenas duas seqüências que são então alinhadas e generalizadas resultando a expressão de consenso final.

2.9 Geração da Expressão Regular

Depois que a expressão de consenso está suficientemente refinada, podemos gerar uma expressão regular que será usada para efetivamente extrair os dados da página HTML. Durante a geração da expressão regular, quando forem encontrados delimitadores fracos e opcionais são utilizados o alfabeto dos seus dados, isto é, dígito, letra, pontuação, etc., no caso de se tratarem de delimitadores de texto. Se forem delimitadores de marcação, utilizamos seu próprio conteúdo generalizando apenas os valores dos possíveis atributos.

Primeiramente, todos os delimitadores opcionais que aparecem no expressão de consenso antes de um delimitador fraco ou forte são identificados. Procuramos assegurar, sempre que possível, que nenhum delimitador fraco ou opcional faça parte do alfabeto dos dados de interesse. O processo de geração da expressão regular inicia-se sempre da extremidade mais próxima ao valor a ser extraído na expressão de consenso.

Conforme apresentado anteriormente, verificamos que a expressão de consenso é constituída pela concatenação das expressões individuais geradas a partir de cada par de delimitadores alinhado, que são representadas por um delimitador e um estado (fixo, obrigatório ou opcional). O estado dos delimitadores é considerado durante a geração da expressão regular da seguinte forma:

- **Para os obrigatórios:** concatenar seu domínio na expressão regular;
- **Para os fixos:** concatenar seu conteúdo;
- **Para os de texto opcionais:** se o alfabeto do delimitador não fizer parte dos delimitadores opcionais que aparecem na expressão de extração antes de um delimitador forte ou fraco, coletar seu domínio e introduzir na classe de caracteres opcionais que será concatenada à expressão regular quando alguma expressão obrigatória ou fixa for encontrada, caso contrário adicionamos seu conteúdo como um padrão opcional na expressão regular;
- **Para os de marcação opcionais:** se o nome do marcador HTML representado pelo delimitador não fizer parte dos delimitadores opcionais que aparecem na expressão de

extração antes de um delimitador forte ou fraco, introduzimos o marcador na mesma classe de caracteres anterior, caso contrário adicionamos seu conteúdo como um padrão opcional na expressão regular.

Uma máscara de extração para os dados a serem extraídos também é gerada, de maneira a contemplar os diferentes tipos de dados que constituem os dados de interesse. Ela é formada pela união de todos os domínios dos tokens que constituem os dados de interesse. Por exemplo, se temos os dados de interesse **Beatles** e **25 Junke Box** a máscara gerada será $([w \setminus d \setminus s]^+)$. Se a variedade de domínios existentes for acima de 5 como nos dados **Beatles**, **RPM #25!**, **Y&I Now?** ou um único dado contiver mais que 5 tokens como no dado **Are you ready for rock tonight?**, geramos uma máscara genérica para os dados $(.*?)$.

As idéias aqui discutidas são implementadas no Algoritmo 5. Na Linha 2 a variável **limitadores** é instanciada com o conjunto de todos os delimitadores opcionais que aparecem antes de um delimitador obrigatório ou fixo. Na Linha 3 é definida uma variável que será utilizada para a criação da classe de caracteres dos delimitadores opcionais. Nas Linhas 4 a 28 temos a geração do prefixo da expressão regular, onde realizamos a transformação para cada delimitador do prefixo da expressão de consenso.

Na Linha 5 temos o critério de parada para a transformação, de modo a impedir a geração de expressões regulares muito grandes e de avaliação lenta. Se isso não ocorresse, mesmo após identificados delimitadores fortes para os dados o processo de transformação continuaria a introduzir padrões opcionais para representar os delimitadores fracos. Padrões opcionais em expressões regulares indicam a existência de variantes que devem ser consideradas durante o casamento da expressão, a procura por essas variantes tornam o processo de avaliação da expressão regular mais lento.

Na Linha 7 tratamos o caso de delimitadores fixos, quando então concatenamos seu conteúdo junto à expressão regular que vai sendo gerada na variável *regex*. Devemos, contudo, verificar, conforme a Linha 8, se algum delimitador opcional foi previamente processado, caso em que é necessário concatenar a classe de caracteres representadas por esses delimitadores (Linha 11). Reiniciamos então a variável que armazena os delimitadores opcionais na Linha 12 e incrementamos a quantidade de delimitadores obrigatórios já processados na Linha 13.

As Linhas 14 a 20 tratam o caso de delimitadores obrigatórios. O processamento é semelhante ao destinado a delimitadores fixos, a única diferença é que no lugar de se utilizar o conteúdo do

Algoritmo 5: Algoritmo de Geração da Expressão Regular

Data: A é uma expressão de consenso

```

1 begin
2   /*Geração do prefixo da expressão */
3   limitadores ← getDelimitadoresImplicitos (getPrefix (A)) ;
4   opcionais ← ∅;
5   foreach delimitador in getPrefix (A) do
6     if (delimitadoresObrigatórios > 4 and contaDelimitadores > 10) or
7       (delimitadoresObrigatórios > 6) or (contaDelimitadores >
8         MAXDELIMITADORES) then
9       break ;
10    if delimitador is FIXO then
11      if opcionais = ∅ then
12        regex ← dados (delimitador))+regex ;
13      else
14        regex ← dados (delimitador))+opcionaisToRegex (opcionais)+regex ;
15      opcionais ← ∅;
16      delimitadoresObrigatórios ++ ;
17    else if delimitador is OBRIGATÓRIO then
18      if opcionais = ∅ then
19        regex ← domínio (delimitador))+regex ;
20      else
21        regex ← domínio (delimitador))+opcionaisToRegex (opcionais)+regex ;
22      opcionais ← ∅;
23      delimitadoresObrigatórios ++ ;
24    else
25      if (domínio (delimitador) not in limitadores) or delimitadoresObrigatórios > 0)
26        then
27        opcionais ← opcionais ∪ domínio (delimitador);
28      else
29        regex ← "["+getCharClass (delimitador))+"]"*+opcionaisToRegex
30          (opcionais)+regex ;
31        opcionais ← ∅;
32        remover (limitadores, delimitador);
33    contaDelimitadores ++ ;
34  /*Máscara para extração dos dados */
35  concatenar (regex, getDataDom (A));
36  /*Geração do sufixo é simétrica */
37  return regex;
38 end

```

token, utilizamos apenas o seu domínio.

As Linhas 21 a 27 consideram o caso em que o delimitador foi marcado como opcional. Neste caso temos que considerar duas possibilidades. Na Linha 22 se algum delimitador obrigatório

ou fixo já foi processado ou se o delimitador atual não pertencer aos limitadores, simplesmente acrescentamos o alfabeto do delimitador na classe de caracteres opcionais. Caso contrário na Linha 25 concatenamos na expressão regular o alfabeto do delimitador seguido pela classe de caracteres dos delimitadores opcionais anteriores e removemos esse delimitador da lista de limitadores.

Na Linha 29 a máscara de extração para dados é gerada e concatenada ao prefixo da expressão regular. Em seguida inicia-se o processamento do sufixo da expressão que é simétrico ao do prefixo. Após ter terminado este processo, uma expressão regular que considera delimitadores fracos e opcionais é gerada.

Existe, contudo, um requisito para o método. O alfabeto dos delimitadores fracos e opcionais deve ser disjunto do alfabeto dos dados de interesse, caso contrário a precisão poderá ser comprometida.

Roberto Carlos, Fábio Junior, KLB, Daniela Mercury (a)
delimitadores válidos

Roberto Carlos e Fábio Junior e KLB e Daniela Mercury (b)
delimitadores inválidos

Figura 2.7: Delimitadores válidos e inválidos

A Figura 2.7 demonstra esta restrição. Em (a) é apresentada uma lista de cantores delimitados pelo caractere “,” cujo alfabeto é disjunto do alfabeto relacionado aos dados de interesse ($([\backslash w\backslash s]^+)$). Em (b) temos o caso em que o delimitador compartilha parte do alfabeto dos dados. Aqui o problema é evidente, o método não conseguirá distinguir o que é delimitador opcional e o que é dado de interesse.

2.10 Estratégia de Extração dos Valores Atômicos

Uma vez que uma expressão apropriada é gerada, podemos usar qualquer avaliador de expressão regular para extrair os dados. Como a expressão criada representa delimitadores fracos que podem ser compartilhados por diferentes instâncias dos atributos de interesse, é necessário reinicializar a avaliação após cada cadeia de caracteres encontrada.

Considere por exemplo uma lista de cantores. Para representar um atributo de interesse (cantor) dentro dessa lista é necessário utilizar os outros dados de interesse como delimitadores

fracos ou opcionais. Neste caso, para recuperar todas as instâncias desse atributo, precisamos remover o valor encontrado e reiniciar a avaliação desde o início da última sequência que foi casada. A Figura 2.8 exemplifica essa estratégia.

```

<B><I>Melhores: Roberto Carlos, Fábio Junior, KLB, Daniela Mercury</I></B>
<B><I>Melhores:           , Fábio Junior, KLB, Daniela Mercury</I></B>
<B><I>Melhores:           ,           , KLB, Daniela Mercury</I></B>
<B><I>Melhores:           ,           ,           , Daniela Mercury</I></B>
<B><I>Melhores:           ,           ,           ,           </I></B>

<B><I>Melhores: [\w\s\, ]*?([\w\s]+)[\w\s\, ]*?</I></B>
                    expressão regular

```

Figura 2.8: Estratégia de extração

Como podemos perceber no exemplo da Figura 2.8, os delimitadores opcionais são representados pelo domínio $([\wedge\s\,]^*?)$ possuem intersecção com o domínio da máscara dos dados que estão sendo extraídos $([\wedge\s]^+)$. Isso acontece tipicamente quando manipulamos uma lista de elementos, onde cada elemento é ele próprio um dado de interesse. Dessa forma extraímos cada instância do atributo e reavaliamos a expressão regular sobre a mesma lista cuja instância anteriormente identificada foi removida, possibilitando a extração das outras instâncias ainda não identificadas.

2.11 Um Exemplo Completo

Para exemplificar o funcionamento do nosso método, vamos considerar páginas de resultados de busca no *site* da American Psychoanalytic Association que contém dados sobre publicações na área de psicanálise. Nessas páginas é possível identificar, conforme Figura 2.9, a existência de dois tipos de objeto: livros e publicações em periódicos, os atributos existentes são **autores**, **ano**, **livro**, **cidade**, **editora**, **artigo**, **periódico**, **edição** e **páginas**.

As páginas deste site apresentam um grande desafio ao processo de extração pois os objetos ali encontrados possuem uma estrutura interna de difícil caracterização devido ao fato de possuírem muitos delimitadores fracos na definição dos contextos de extração. Os métodos automáticos não são capazes de reconhecer nessas páginas a estrutura dos objetos existentes, tratando cada um deles como um elemento indivisível.

Vamos considerar neste exemplo o atributo **artigo**. Esse atributo é encontrado em objetos que identificam publicações em periódicos. Nas instâncias do atributo **artigo** existe uma vasta

American Psychoanalytic Association

Jourlit Bookrev Search Result

Your Search Found 2,755 Matches
Showing Matches 1 to 50.

Show More

New Search

Revise Search

Abel, T. (1978). Yukio Mishima - a *psychoanalytic* interpretation, *J. Amer. Acad. Psychoanal.*, 6:403-424

Abend, S., Porder, M. & Willick, M. (1983). *Borderline Patients: Psychoanalytic Perspectives*. Kris Study Group of the New York Psychoanalytic Institute., New York: Int. Univ. Press.

Figura 2.9: Página do *site* APSA

gama de possibilidades dos dados aí contidos, podendo ser utilizados quaisquer tipos de caractere. Além disso, existe também a possibilidade de alguma palavra vir marcada como <I> ou .

Os prefixos desses elementos são formados por um delimitador obrigatório que é ano de publicação. Acontece que esse delimitador pode opcionalmente vir acompanhados dos marcadores <I> ou . Após o título do artigo aparece, separado por uma vírgula e em itálico, o atributo *periódico*. Essas informações de contexto podem ser utilizadas como delimitadores fixos pelo padrão de extração.

Num.	Exemplo
01	... (1988). <I>The</I> question <I>of</I> lay analysis: sixty years later, <I>Psychoanal. Rev.</I>,...
02	... (1974). Panel: <I>The</I> analyst's emotional life during work., <I>J. Amer. Psychoanal. Assn.</I>,...
03	... (<I>1999</I>). Review of <I>Illness and culture in the postmodern age.</I>, <I>Psychoanal. Books</I>,...
04	...(<I>1999</I>). 'The principal task of child analysis' in retrospect., <I>Child Analysis</I>,...

Tabela 2.1: Exemplos selecionados para o atributo artigo

Inicialmente selecionamos 4 exemplos de treinamento para esse atributo (Tabela 2.1) que serão utilizados para generalizar um padrão de extração. Em seguida calculamos a matriz de

distâncias com os valores dos alinhamentos para cada par de exemplos e escolhemos o par de exemplos mais similar para dar início ao processo de generalização.

Ex.	01	02	03	04
01	0.0	827.0	709.1667	565.0
02	0.0	0.0	534.0	659.0
03	0.0	0.0	0.0	672.0
04	0.0	0.0	0.0	0.0

Tabela 2.2: Matriz de distâncias inicial para os exemplos do atributo artigo

Pela Tabela 2.2 percebemos que os exemplos 01 e 02 são os que apresentam maior similaridade, devendo, portanto, ser generalizados primeiro. A primeira etapa desse processo consiste em realizar a identificação dos delimitadores desses exemplos de modo a permitir o seu subsequente alinhamento, como mostrado na Figura 2.10.

```

...</B> (1988) . <<valor>>, <I>          Psychoanal. Rev .</I>, ...
...</B> (1974) . <<valor>>, <I> J. Amer.    Psychoanal. Assn.</I>, ...
-----
...</B> (1988) . <<valor>>, <I> J. Amer.    Psychoanal. Rev .</I>, ...
  ■   ■   ●   ■   ■           ■   ■   ▲▲   ▲   ▲           ■   ■   ●   ■   ■   ■

```

Figura 2.10: Primeiro alinhamento dos exemplos

Após o alinhamento dos exemplos, inicia-se o processo de generalização para cada par de delimitadores alinhado, resultando em uma expressão de consenso que incorpora informações relativas aos dois exemplos. Em seguida é recalculado a matriz de distâncias para a expressão de consenso gerada, conforme a Tabela 2.3.

Ex.	01	02	03
01	0.0	672.0	707.1667
02	0.0	0.0	765.0
03	0.0	0.0	0.0

Tabela 2.3: Matriz de distâncias após a primeira generalização

Dando prosseguimento ao processo de generalização, um novo par de exemplos é escolhido. Agora os mais similares são o exemplo 04 e a expressão de consenso recém criada. É realizado então o alinhamento entre essas seqüências conforme apresentado na Figura 2.11.

Por fim, restam apenas duas seqüências. Todo o processo se repete para a geração de uma nova expressão que será justamente a expressão consenso final, que será transformada em uma expressão regular que poderá ser utilizada para se extrair os valores atômicos das páginas.

...	(	<I> 1999 </I>		).	<<valor>> ,	<I>	Child	Analysis	</I> ,	...
...		(	1988	)	<<valor>> ,	<I> J. Amer.	Psychoanal.	Rev	</I> ,	...
	■	■	●			■	■	▲	▲	▲

...		(	<I> 1988 </I>		).	<<valor>> ,	<I> J. Amer.	Psychoanal.	Rev	</I> ,	...
	■	■	▲	●	▲	▲	■	■	▲	▲	▲

Figura 2.11: Segundo alinhamento de exemplos

...		(	<I> 1999 </I>		).	<<valor>> ,	<I>	Psychoanal.	Books	</I> ,	...
...		(	<I> 1988 </I>		).	<<valor>> ,	<I> J. Amer.	Psychoanal.	Rev	</I> ,	...
	■	■	▲	●	▲	▲	■	■	▲	▲	▲

...		(	<I> 1988 </I>		).	<<valor>> ,	<I> J. Amer.	Psychoanal.	Rev	</I> ,	...
	■	■	▲	●	▲	▲	■	■	▲	▲	▲

Figura 2.12: Último alinhamento e geração da expressão final

O que está faltando na expressão da Figura 2.12 é a geração da expressão regular que é apresentada na Figura 2.13

```
...</B>?(<B>?<I>?\d+</I>?</B>?). <<valor>>, <I>[\w+\.\s]*?\w+\.\.?s*\w+s*\.\.?</I>, ...
onde <<valor>> = ((?:[^\<>]|(?:</?I\b[^\<>]*>)|(?:</?B\b[^\<>]*>))+?)
```

Figura 2.13: Expressão de extração final e máscara dos dados

Após a geração da expressão regular, esta é utilizada para extrair os valores atômicos que estão localizados nas páginas. Nesse processo de geração semi-automática nenhuma outra informação é fornecida, apenas exemplos de treinamento para os valores atômicos, diferentemente de como acontece com outros métodos que se utilizam não só dos exemplos, mas também de informações adicionais fornecidas pelo usuário, tal como estrutura dos objetos nas páginas.

Capítulo 3

Montagem dos Objetos

Após a localização e extração dos valores atômicos usando as técnicas descritas no capítulo anterior, é necessário reconstituir os objetos de acordo com uma estrutura plausível. Normalmente, nos métodos convencionais [Baumgartner et al., 2001, Muslea et al., 2001, Laender et al., 2002b, Sahuguet and Azavant, 2001], essa estrutura é fornecida pelo usuário. Porém, em nossa abordagem, utilizamos um método automático para inferir a estrutura aninhada dos objetos que é capaz de combinar adequadamente os valores atômicos. Esse método, inicialmente proposto em [da Silva, 2002], foi revisto e ampliado neste trabalho e é aqui apresentado em detalhes.

No método, primeiramente é construído um grafo de adjacências para os valores atômicos que foram extraídos, considerando para tanto a posição relativa em que estes ocorreram dentro do documento HTML. Em seguida, é feita uma busca nesse grafo a procura de padrões estruturais freqüentes, detectáveis por meios de ciclos no grafo, que se assemelhem a construtores como listas ou tuplas. Os ciclos mais freqüentes no grafo, chamados de **ciclos dominantes**, são utilizados para identificar pistas que descrevem a estrutura dos objetos.

Os ciclos dominantes passam por um processo de generalização, em seguida todas as ocorrências desses ciclos são substituídas por um novo nodo no grafo que representa parte da estrutura aninhada que foi identificada. Com a substituição dos nodos dos níveis mais baixos da estrutura aninhada, podemos dar seqüência à identificação do próximo nível da estrutura dos objetos. Assim, os objetos vão sendo recursivamente reconstruídos até que não existam mais elementos a serem montados.

3.1 Grafo de Adjacências

O grafo de adjacências que utilizamos é um grafo direcionado, gerado a partir da lista de ocorrências dos valores atômicos extraídos. Neste grafo, o conjunto de vértices corresponde exatamente ao conjunto dos atributos definidos para o documento através de exemplos dados pelo usuário. As arestas, por sua vez, indicam uma relação de precedência entre as instâncias dos atributos na página. Cada aresta tem associado um peso que nada mais é do que a quantidade de ocorrências em que as instâncias dos atributos relacionados à aresta se seguiram na página [da Silva, 2002].

Definição 7 Considere o conjunto $\{\langle\tau_1, s_1\rangle, \dots, \langle\tau_n, s_n\rangle\}$ formado por valores atômicos extraídos de uma página HTML H , onde τ_k é o atributo associado a string s_k . Definimos uma **lista de ocorrências** $L = \langle l_1, l_2, \dots, l_n \rangle$ como sendo uma lista ordenada de pares $l_i = \langle\tau_i, \rho_i\rangle$, onde τ_i é o nome do atributo do valor atômico extraído s_i e ρ_i é a posição desse valor na página H . Chamamos cada l_i de uma ocorrência em G . Dizemos também que $l_i \prec l_j$, ou seja, l_i imediatamente precede l_j sss $j = i + 1$.

Definição 8 Seja L uma lista de ocorrências de uma página HTML H . Um grafo de adjacências para H é um grafo $G = \langle V, A, l \rangle$, onde $V = \{\tau_i \mid \exists p_i \text{ onde } \langle\tau_i, p_i\rangle \in L\}$ é o conjunto de vértices de G e corresponde aos nomes dos atributos em L (um vértice para cada atributo); $A = \{\langle\tau_i, \tau_j\rangle \mid \exists p_i, p_j \text{ onde } \langle\tau_i, p_i\rangle, \langle\tau_j, p_j\rangle \in L \text{ e } \langle\tau_i, p_i\rangle \prec \langle\tau_j, p_j\rangle\}$ é o conjunto de arestas de G tal que existe uma aresta se pelo menos uma instância do atributo τ_i precede outra instância do atributo τ_j em L ; e $l(\langle\tau_i, \tau_j\rangle) = |\{\langle\langle\tau_i, p_i\rangle, \langle\tau_j, p_j\rangle\rangle \in L \times L \mid \langle\tau_i, p_i\rangle \prec \langle\tau_j, p_j\rangle\}|$ é uma função que rotula as arestas de A com o número de instâncias do atributo τ_i que imediatamente precedem instâncias de τ_j em L .

Considere a página de catálogo de livros do site Amazon.com apresentada na Figura 3.1. A Figura 3.2 apresenta um grafo gerado para essa página. Nesse grafo verificamos que 50 instâncias do atributo **Título** são encontrados que precedem outras instâncias do atributo **Autor**. Verificamos também que instâncias do atributo **Autor** precedem 16 vezes outras instâncias desse mesmo tipo, e assim por diante.

O Algoritmo HotCycles [da Silva, 2002] reconstitui os objetos de acordo com uma política *bottom-up*. A cada iteração o algoritmo analisa o grafo de adjacências numa tentativa de com-

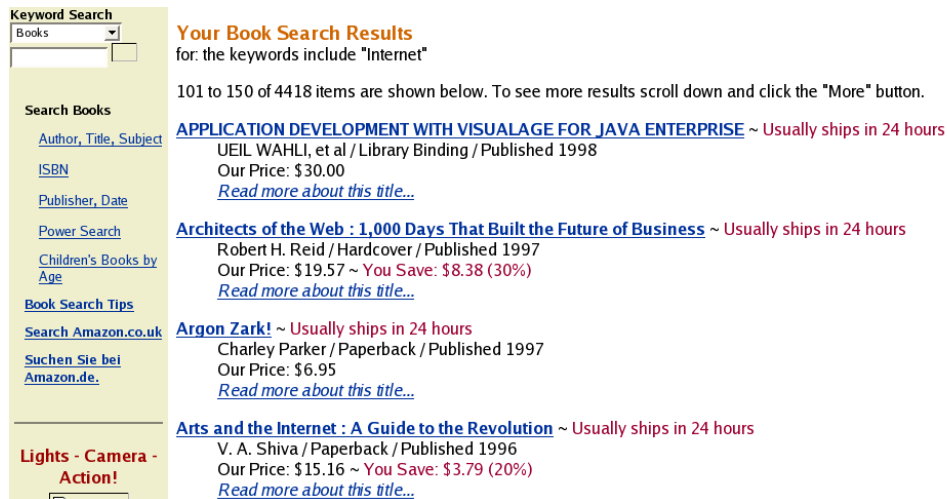


Figura 3.1: Página de catálogo da Amazon.com

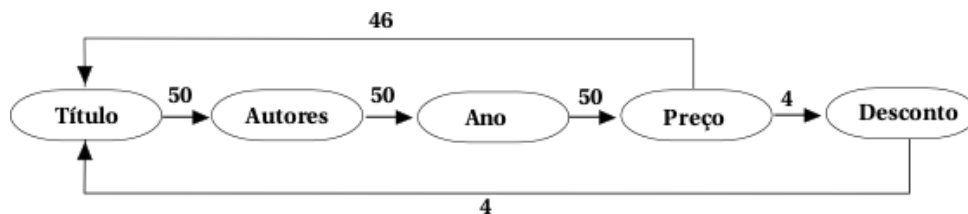


Figura 3.2: Grafo de adjacências para uma página da Amazon.com

binar os elementos existentes em estruturas mais complexas. Isso é realizado através da identificação de ciclos no grafo. O objetivo é encontrar ciclos altamente frequentes no grafo que indicam a existência de uma estrutura plausível para se combinar os elementos.

Em nossa proposta estendemos a proposição original do algoritmo HotCycles de modo a fornecer um suporte mais adequado a estruturas que apresentem elementos opcionais. Realizamos isso através da generalização dos ciclos dominantes, identificando ciclos mais amplos que representam de modo mais adequado a estrutura dos objetos. Esses ciclos são chamados de **ciclos estruturantes**. A seguir é descrito o processo para a identificação de ciclos dominantes, para em seguida discutirmos os ciclos estruturantes.

3.2 Ciclos Dominantes

O primeiro passo para encontrar os ciclos dominantes ocorre após a montagem do grafo de adjacências. Nessa fase identificamos todos os ciclos existentes nesse grafo. Durante este processo, ciclos semelhantes, ou seja, ciclos formados pelos mesmos atributos, podem ser identificados.

Dado um determinado ciclo no grafo, dizemos que existe um tipo complexo formado por todos os atributos que ocorrem no ciclo. Esse conceito é formalizado na Definição 9.

Definição 9 *Seja $\varrho = (\tau_1, \tau_2, \dots, \tau_{n-1}, \tau_n, \tau_1)$ um ciclo em um grafo de adjacências G . Dizemos que $\gamma(\varrho) = \langle \tau_1, \tau_2, \dots, \tau_{n-1}, \tau_n \rangle$ é o **tipo plano** correspondente ao ciclo ϱ em G*

Considere o grafo de adjacências apresentado na Figura 3.2, nele podemos identificar o ciclo **Título, Autor, Ano, Preço, Título**. Porém, é possível identificar também o ciclo **Autor, Ano, Preço, Título, Autor**. Neste caso é necessário determinar qual a ordem de precedência entre os atributos que constituem os vértices do ciclo, o que pode ser realizado pela verificação da disposição de vértices que apresenta a maior frequência na lista de ocorrências. No exemplo da Figura 3.2 identificaríamos a disposição **Título, Autor, Ano, Preço, Título** como a mais frequente das disposições possíveis para este ciclo.

Além do tipo plano de um ciclo, também é necessário conhecer os atributos na vizinhança do ciclo. Para tanto usamos os conceitos de atributos de entrada e atributos de saída do ciclo.

Definição 10 *Seja ϱ um ciclo de um grafo de adjacências G gerado a partir de uma lista de ocorrências L . Chamamos de **atributos de entrada** de ϱ o conjunto de atributos $\{\tau_1, \dots, \tau_n\}$ tal que para todo τ_i , existe um $\langle \tau_i, p_i \rangle$ que precede alguma ocorrência do ciclo ϱ em L e **atributos de saída** o conjunto de atributos $\{\tau_1, \dots, \tau_n\}$ tal que para todo τ_i , existe um $\langle \tau_i, p_i \rangle$ que sucede alguma ocorrência do ciclo ϱ em L .*

Uma vez determinados todos os ciclos existentes no grafo de adjacências, é realizada uma contagem para verificar o número de ocorrências de cada ciclo. Na proposta original do método [da Silva, 2002] a identificação do ciclo mais frequente é realizada através da contagem associada as arestas do grafo de adjacências. A contagem das ocorrências considerando apenas as arestas do grafo de adjacências é imprecisa, pois o grafo representa uma simplificação da disposição original dos valores atômicos na lista de ocorrências.

Nesse novo processo de contagem é considerado como uma ocorrência toda sequência de valores cujos nomes dos atributos correspondem exatamente à mesma sequência de vértices no ciclo. Além disso é necessário ainda que o valor atômico que sucede tal sequência na lista de ocorrências seja:

- um valor atômico cujo tipo já tenha ocorrido na sequência (indicando uma provável repetição da sequência e portanto do tipo plano correspondente);

- um valor atômico cujo tipo seja igual a algum dos atributos de saída do último vértice do ciclo (indicando a última ocorrência do tipo plano em uma provável lista), ou;
- fim da lista (indicando também a última ocorrência do tipo plano em uma provável lista).

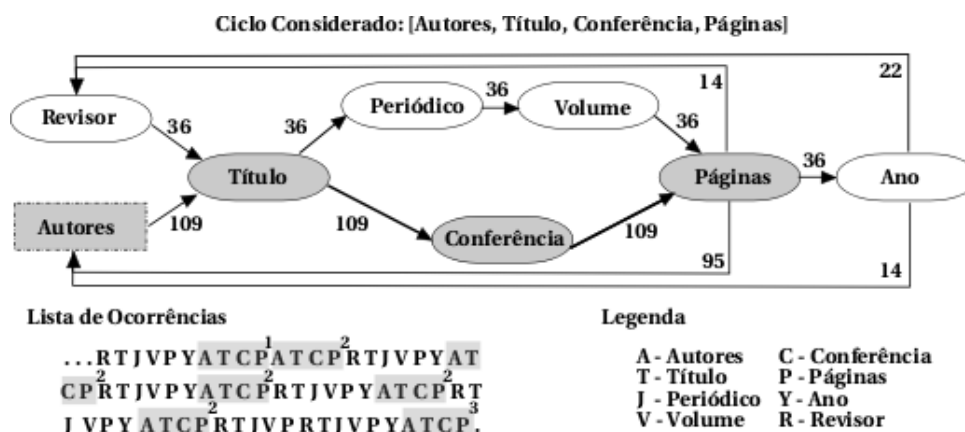


Figura 3.3: Contagem de ocorrências de um ciclo num grafo de adjacências

A Figura 3.3 representa um grafo de adjacências de um conjunto fictício de páginas contendo uma lista de artigos em conferências e em periódicos. Para exemplificar o processo de contagem de ocorrências, considere o ciclo formado pelos vértices **Autores**, **Título**, **Conferência**, **Páginas**. Na lista de ocorrências deste grafo podemos identificar as diversas ocorrência do ciclo em questão.

Na Figura 3.3 as ocorrências são classificadas de acordo com um índice que corresponde a cada uma das condições descritas anteriormente. O índice 1 identifica o caso de repetição do ciclo, o índice 2 o caso em que o ciclo é o último de uma lista ou o caso em que o ciclo ocorre sozinho, finalmente o índice 3 identifica o caso em que o ciclo ocorre no fim da lista.

Definição 11 *Seja ϱ um ciclo em um grafo de adjacências G gerado a partir de uma lista de ocorrências L . Dizemos que $F(\varrho)$ é uma função que indica a quantidade de ocorrências do ciclo ϱ na lista de ocorrências L .*

Assim, podemos finalmente redefinir o conceito de ciclo dominante.

Definição 12 *Seja $\{S_1, \dots, S_n\}$ o conjunto com todos os ciclos de G , onde G é um grafo de adjacências gerado a partir de uma lista de ocorrências L . Chamamos de **ciclo dominante** o ciclo S_i cujo $F(S_i) = \max(F(S_1), \dots, F(S_n))$*

3.3 Ciclos Estruturantes

A utilização do ciclo dominante nos fornece um bom indício para a identificação do nível mais interno da estrutura aninhada dos objetos. Contudo, é necessário verificar a possibilidade de generalização desse ciclo, de maneira a identificar possíveis elementos opcionais no tipo plano relacionado.

Tomando como exemplo o grafo de adjacências da Figura 3.2, identificamos o ciclo **Título, Autor, Ano, Preço** como sendo o mais freqüente. Contudo, percebemos a existência do atributo **Desconto** que figura como atributo opcional. É necessário portanto generalizar o ciclo dominante para que considere também elementos desse tipo.

Definição 13 *Seja L uma lista de ocorrências e um grafo de adjacências G construído a partir de L . Dizemos que um ciclo ϱ de G é um **ciclo viável** se $F(\varrho) > 1$.*

Definição 14 *Seja S um ciclo de um grafo de adjacências G construído a partir de uma lista de ocorrências L . Dizemos que um ciclo S_A de G é um **ciclo ampliado** de S , se S_A for viável, contiver todos os vértices de S e pelo menos mais um outro vértice qualquer.*

Um ciclo dominante é progressivamente ampliado para um ciclo maior, isto é, ciclos que possuam mais atributos, de acordo com regras de generalização de procuram identificar elementos opcionais no tipo plano correspondente. Tal processo é essencial pois nem sempre todas as instâncias de um dado tipo contêm todos os átomos possíveis. Além disso, o processo de extração de átomos pode falhar em identificar todos os valores existentes nas páginas HTML, fazendo com que o grafo de adjacências apresente anomalias. O objetivo desse processo é identificar o tipo alvo mais próximo do ideal para agrupar os valores atômicos.

Um ciclo dominante é comparado com todos os seus ciclos ampliados existentes no grafo de adjacências com o objetivo de realizar a generalização do ciclo dominante para um desses ciclos. Uma vez ampliado o ciclo, o processo é reiniciado sendo o ciclo recém-generalizado comparado por sua vez com todos os seus ciclos ampliados existentes no grafo. Esse processo termina quando não é mais possível generalizar o ciclo considerado.

Um ciclo dominante é generalizado para um ciclo ampliado de acordo com as seguintes regras:

Caso 1: Valores opcionais existentes no interior do tipo alvo. Neste caso percebemos a existência de um ciclo ampliado que envolve o ciclo dominante e que possui as seguintes características:

- Contém todos os vértices do ciclo dominante;
- É um ciclo viável; e
- O primeiro e o último vértice desse ciclo ampliado correspondem respectivamente ao primeiro e ao último vértice do ciclo dominante.

Caso 2: Valores opcionais existentes nas extremidades do tipo alvo. Duas possibilidades são identificadas: valores opcionais à direita do tipo alvo e valores opcionais à esquerda do tipo alvo. Aqui existe também um ciclo mais amplo envolvendo o ciclo dominante, contudo é necessário verificar se tal situação indica realmente a existência de um valor opcional no tipo alvo, ou se tal configuração representa na verdade um novo nível dentro da estrutura aninhada. Para tanto, de acordo com as duas possibilidades, devemos verificar dois sub-casos para esse ciclo ampliado:

Caso 2a: Opcionais à direita

- Contém todos os vértices do ciclo dominante;
- É um ciclo viável;
- O primeiro vértice do ciclo ampliado é igual ao primeiro vértice do ciclo dominante;
- O tipo do elemento mais próximo ao início da lista de ocorrências existente no ciclo dominante é o mesmo que o elemento mais próximo ao início da lista de ocorrências existente no ciclo ampliado.
- Os atributos de entrada do ciclo ampliado contêm todos os atributos de entrada do ciclo dominante;
- Os atributos de saída do ciclo dominante contêm todos os atributos de saída do ciclo ampliado ou vice-versa (indicando que ambos os ciclos ocorrem em contextos semelhantes), ou a quantidade de ocorrências do super-ciclo seja maior que um fator $E\%$ do número de ocorrências do ciclo dominante. Isso indica que os valores se adequam melhor à situação de valor opcional, do que a um novo nível na estrutura aninhada.

Caso 2b: Opcionais à esquerda

- Contém todos os vértices do ciclo dominante;
- É um ciclo viável;
- O último vértice do ciclo ampliado é igual ao último vértice do ciclo dominante;
- O tipo do elemento mais próximo ao final da lista de ocorrências existente no ciclo dominante é o mesmo que o elemento mais próximo ao final da lista de ocorrências existente no ciclo ampliado.
- Os atributos de saída do ciclo ampliado contêm todos os atributos de saída do ciclo dominante;
- Os atributos de entrada do ciclo dominante contêm todos os atributos de entrada do ciclo ampliado ou vice-versa (indicando que ambos os ciclos ocorrem em contextos semelhantes), ou a quantidade de ocorrências do super-ciclo seja maior que um fator $E\%$ do número de ocorrências do ciclo dominante. Isso indica que os valores se adequam melhor a situação de valor opcional, do que a um novo nível na estrutura aninhada.

Caso 3: Valores opcionais existentes em diversas posições do tipo alvo. Neste caso o tipo alvo apresenta muitos valores opcionais ou o método de extração de átomos apresentou um grau de revocação ruim. Devemos decidir se os valores atômicos do ciclo ampliado em questão devem ser encarados como valores opcionais, ou se constituem um novo nível na estrutura aninhada. O ciclo dominante será generalizado se o ciclo ampliado:

- Contém todos os vértices do ciclo dominante;
- É um ciclo viável;
- A quantidade de ocorrências desse ciclo ampliado seja maior que um fator $F\%$ do número de ocorrências do ciclo dominante. Isso indica que os valores se adequam melhor à situação de valor opcional, do que a um novo nível na estrutura aninhada.

A Figura 3.4 exemplifica cada um dos casos anteriormente elencados. Devido ao espaço necessário para se apresentar a lista de ocorrência para cada um dos grafos de adjacências dessa figura, optamos por suprimir essa informação. Os pesos associados às arestas do grafo indicam

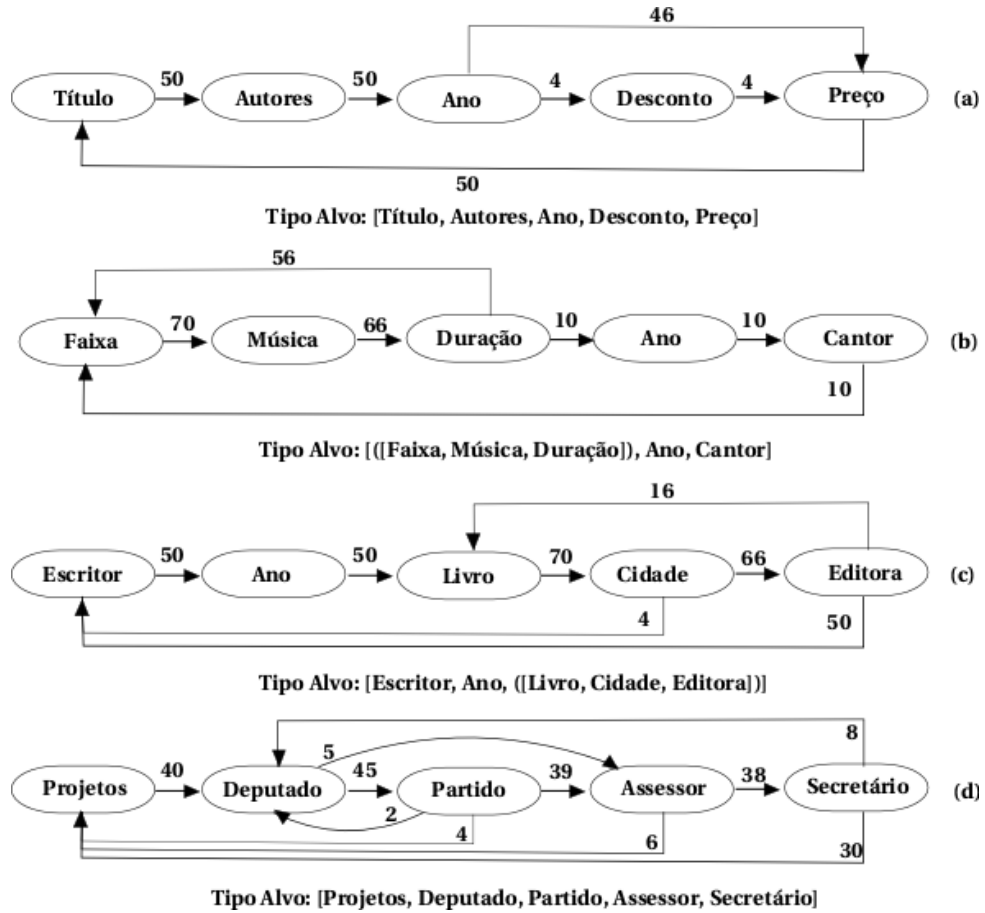


Figura 3.4: Casos de generalização do ciclo dominante

a frequência com que os atributos ocorrem na página e, conforme dito anteriormente, representa uma simplificação da distribuição dos atributos na página, não sendo utilizados para a contagem das ocorrências de cada ciclo.

Em (a) temos a ocorrência do Caso 1, o ciclo dominante Título, Autores, Ano, Preço cuja ocorrência é 46 deve ser generalizado para o ciclo ampliado Título, Autores, Ano, Desconto, Preço.

Um exemplo de aplicação do Caso 2a é apresentado em (b). Dado o ciclo dominante Faixa, Música, Duração cuja frequência é 70, devemos decidir se este deve ou não ser generalizado para o ciclo Faixa, Música, Duração, Ano, Cantor cuja ocorrência é 10. Conforme as regra do Caso 2a verificamos que os atributos de entrada não são compatíveis e o número de ocorrências do ciclo ampliado não é próximo o suficiente de 70 para que se justifique a ampliação. Isso indica que uma estrutura aninhada representa melhor os objetos.

Em (c) temos que decidir se o ciclo Livro, Cidade, Editora cuja ocorrência é 66 deve ou

não ser generalizado para o ciclo **Escritor, Ano, Livro, Cidade, Editora** cuja ocorrência é 35. Aplicamos aqui a regra do Caso 2b e verificamos que os atributos de saída não são compatíveis nem tão pouco que a frequência 35 seja próxima o suficiente de 66¹. Isso também indica que uma estrutura aninhada representa melhor os objetos e portanto decidimos não generalizar o ciclo.

Em (d) percebemos a existência de muitas variantes, pois o tipo alvo em questão apresenta vários atributos opcionais. Devemos agora decidir se o ciclo dominante **Deputado, Partido** cuja ocorrência é 45 deve ser ou não generalizado para o ciclo ampliado **Projetos, Deputado, Partido, Assessor, Secretário** cuja ocorrência é 30. Pela aplicação da regra do Caso 3 concluímos pela não generalização deste ciclo. Porém a existência de outros ciclos ampliados nos permite realizar a generalização do ciclo dominante através de outro caminho.

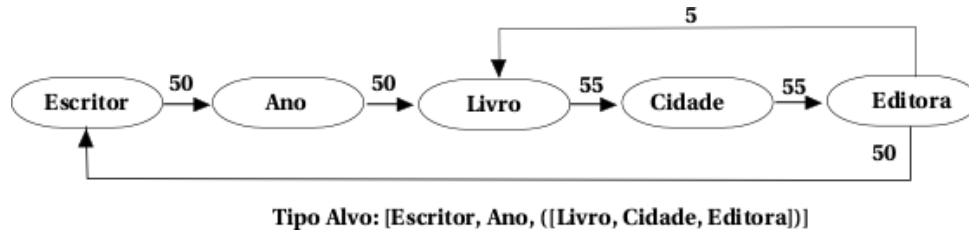
Verificamos ainda a existência do ciclo **Projetos, Deputado, Partido** cujos atributos de entrada e saída são compatíveis com o ciclo dominante **Deputado, Partido**, permitindo portanto a generalização desse ciclo através da regra do Caso 2b. Verificamos também a existência do ciclo **Projetos, Deputado, Partido, Assessor**, cujos atributos de entrada e saída são compatíveis com os do ciclo recém ampliado **Projetos, Deputado, Partido**. Aplicando a regra do Caso 2a realizamos essa nova generalização.

Após ampliarmos o ciclo dominante **Deputado, Partido** para o ciclo **Projetos, Deputado, Partido, Assessor** verificamos ser possível agora a generalização desse ciclo para o ciclo **Projetos, Deputado, Partido, Assessor, Secretário** pela aplicação da regra do Caso 2a. Finalmente chegamos com esse processo ao ciclo **Projetos, Deputado, Partido, Assessor, Secretário**.

A idéia de se utilizar os fatores E e F é permitir que ciclos dominantes sejam generalizados para ciclos mais amplos mesmo que estes se apresentem em contextos diferentes. Isso se explica pois algumas estruturas apesar de aninhadas podem ser representadas como estruturas planas quando o número de ocorrências da estrutura mais interna é muito próximo do número de ocorrências da estrutura externa.

Considere por exemplo o grafo de adjacências da Figura 3.5. Apesar do tipo alvo definido, verificamos que o número de ocorrências da estrutura mais interna representada por **Livro, Cidade, Editora** é muito próximo ao número de ocorrências da estrutura externa **Escritor**,

¹Em nossos experimentos utilizamos os valores para $E = 80\%$ e $F = 90\%$

Figura 3.5: Utilização dos fatores E e F

Ano, Livro, Cidade, Editora. Nesse caso Livro e Cidade apesar de representarem uma estrutura aninhada são identificados como elementos opcionais.

Como foi visto, o processo de ampliação do ciclo dominante é realizado em etapas onde em cada passo um novo ciclo ampliado é encontrado. Com a progressiva generalização do ciclo dominante identificamos uma parte da estrutura do objeto complexo relacionado ao grafo de adjacências. Os ciclos dominantes ampliados sugerem, portanto, estruturas complexas para agrupar os átomos. Estes ciclos são o que chamamos de **ciclos estruturantes** ou **HotCycles**.

Uma vez identificado o nível mais interno da estrutura aninhada, podemos então realizar a substituição desses elementos de forma a reconstituir esse nível da estrutura dos objetos.

3.4 Substituição das Ocorrências do Ciclo Estruturante

Como vimos na seção anterior, os ciclos estruturantes são responsáveis por identificar a estrutura mais interna do tipo aninhado dos objetos complexos. Nosso objetivo, contudo, não se restringe à identificação somente desse nível da estrutura complexa, é necessário dar prosseguimento ao processo de identificação da estrutura completa dos objetos.

Para tanto realizamos a substituição do nível mais interno que foi identificado e em seguida damos prosseguimento à identificação do próximo nível da estrutura seguindo, portanto, um processo de reconstrução **bottom-up** dos objetos, começando com as estruturas mais internas e chegando até a estrutura final dos objetos.

Uma vez definidas partes da estrutura pela generalização dos ciclos dominantes, deve-se realizar a substituição na lista de ocorrência das instâncias do tipo plano correspondente ao ciclo estruturante. Nessa substituição, considera-se a possibilidade da existência de elementos opcionais, conforme considerado durante a ampliação do ciclo dominante. Dessa maneira é possível encontrar elementos que podem estar ausentes no tipo plano do ciclo estruturante.

Definição 15 *Seja $\gamma(\varrho) = \langle \tau_1, \tau_2, \dots, \tau_{n-1}, \tau_n \rangle$ o tipo plano correspondente ao ciclo ϱ em G . Dizemos que $\gamma_f(\varrho) = \langle \tau_1?, \tau_2?, \dots, \tau_{n-1}?, \tau_n? \rangle$ é o **tipo plano expandido** correspondente a ϱ , que nada mais é que o mesmo tipo $\gamma(\varrho)$ com a possibilidade de elementos opcionais.*

A ordem em que os elementos devem aparecer na lista de ocorrências é definida pela disposição dos vértices no ciclo dominante. Cada ocorrência detectada é substituída por um elemento que representa o tipo encontrado. Uma ocorrência do tipo plano expandido de um ciclo é detectada quando:

- existe uma seqüência de elementos cujos tipos são os mesmos definidos pelos vértices do ciclo e estes elementos respeitam a mesma disposição dos vértices, podendo haver opcionais. Alguns elementos podem, portanto, estar ausentes, mas não fora de ordem;
- na posição que antecede essa seqüência na lista de ocorrências temos um elemento que pertença aos atributos de entrada do ciclo;
- na posição que sucede essa seqüência na lista de ocorrências temos um elemento que pertença aos atributos de saída do ciclo ou nessa posição nos deparemos com o final da lista.

Detecção de Listas

A substituição dos ciclos estruturantes abrange as estruturas que constituem as tuplas. Mas temos ainda aquelas estruturas que são formadas por iteração (listas). Listas são facilmente detectadas através da localização de auto-ciclos no grafo de adjacências, ou seja, arestas cujos vértices de entrada e saída são os mesmos, indicando uma repetição desse elemento. Neste caso cada seqüência desses mesmos elementos é substituída por um novo elemento que representa a repetição.

3.5 Algoritmo HotCycles

O Algoritmo 7, denominado originalmente HotCycles [da Silva, 2002] aplica as idéias apresentadas nesse capítulo para reconstruir os objetos complexos. Os objetos são montados inicialmente pelas estruturas mais internas do tipo aninhado, através da identificação e ampliação dos ciclos dominantes. O algoritmo utiliza uma técnica de montagem *bottom-up* que recursivamente

vai montando os objetos. Uma vez identificado um ciclo estruturante é realizada a substituição das instâncias do tipo plano expandido correspondente, de modo a possibilitar a identificação do próximo nível na estrutura dos objetos. O algoritmo termina sua execução quando todos os objetos já estão devidamente montados.

Algoritmo 6: Algoritmo de análise de adjacências

Data: L é uma lista de ocorrências
Data: G é o grafo de adjacências de L
Data: S é uma lista de ciclos de L
Data: C_D é o ciclo dominante
Data: C é o ciclo atual ou de trabalho
Data: C_E é o ciclo estruturante

```

1 begin
2   while possuiElementosNãoMontados ( $L$ ) do
3      $G \leftarrow$  gerarGrafoAdjacências ( $L$ );
4     foreach ciclo simples  $[v,v]$  in  $G$  do
5       Substitua todas as seqüências  $\langle v, p_i \rangle, \dots, \langle v, p_{i+k} \rangle (k > 0)$  por um único
       elemento  $\langle (v), p_i \rangle$  em  $L$ 
6      $G \leftarrow$  gerarGrafoAdjacências ( $L$ );
7      $S \leftarrow$  ciclos ( $G, L$ );
8      $C_D \leftarrow$  próximo ( $S$ );
9      $C \leftarrow C_D$ ;
10    while ( $|S| > 0$  and existeValorRemanescente ( $L, C_D$ )) do
11       $C_E \leftarrow$  ampliarCiclo ( $C, S$ );
12      if (cicloViável ( $C_E$ )) then
13        Substitua todas as seqüências compatíveis com  $C_E$   $\langle v_1, p_i \rangle, \dots, \langle v_n,$ 
         $p_{i+n} \rangle$  em  $L$  por um único elemento  $\langle (v_1, \dots, v_n), p_i \rangle$  em  $L$ .
14      if ( $|S| > 0$ ) then
15         $C \leftarrow$  próximo ( $S$ )
16      else
17         $C \leftarrow null$ ;
18 end
```

Na repetição que inicia na Linha 2 do algoritmo HotCycles a função **possuiElementosNãoMontados** verifica se existem elementos ainda “soltos” na lista de ocorrências e enquanto detectar tal situação continua o processo de montagem dos objetos complexos. Na Linha 3 é gerado o grafo de adjacências G relativo à lista de ocorrências L . Em seguida, na Linha 4, o algoritmo verifica a existência de auto-ciclos e, na Linha 5, para cada auto-ciclo detectado, substituem-se todas as seqüências de valores cujos atributos são equivalentes ao vértice em questão, acrescentando em L um novo elemento que representa uma lista desses valores.

Em seguida, na Linha 6 um novo grafo de adjacências G é gerado, agora sem nenhum auto-

ciclo, pois estes já foram previamente substituídos. Na Linha 7 são identificados todos os ciclos existentes em G através da função `ciclos` que retorna todos os ciclos de G ordenados de maneira decrescente com relação ao número de ocorrências de cada ciclo em L . C_D representa o ciclo dominante.

O algoritmo inicia então o processo de generalização do ciclo de trabalho C que inicialmente é o próprio ciclo dominante. Na Linha 10 enquanto existirem ciclos e valores remanescentes em L cujos tipos sejam equivalentes ao tipo de algum vértice de C_D o processamento desse nível continua. Isso é necessário devido à possibilidade de existência de tuplas de estruturas diferentes mas que possuem alguns atributos em comum, como é o caso do exemplo das páginas de autores do *site* DBLP², apresentado em detalhes na Seção 3.6, onde existem duas estruturas para os objetos e que possuem atributos em comum.

A função `ampliarCiclo` na Linha 11 é responsável pela processo de generalização do ciclo dominante conforme definido da Seção 4.3. Em seguida nas Linhas 12 e 13 verifica-se se o ciclo gerado é de fato viável, quando então é realizada a substituição das ocorrências em L .

Na Linha 14 o ciclo de trabalho C é então atualizado para o próximo ciclo existente em S continuando o processo de substituição de ocorrências. Quando não houver mais em L nenhuma instância dos atributos utilizados no ciclo, uma nova iteração é iniciada de maneira a identificar o próximo nível na estrutura aninhada.

3.6 Um Exemplo Completo

Para exemplificar o funcionamento do algoritmo `HotCycles`, vamos considerar as páginas de autores da DBLP, que contêm informações sobre produção bibliográfica de diversos pesquisadores na área de computação. Nessas páginas é possível identificar a existência de dois tipos de objeto: publicações em conferências e publicações em periódicos. Os atributos existentes são `Autor`, `Título`, `Periódico`, `Conferência`, `Edição`, `Páginas` e `Ano`.

Esses tipos de objeto podem ser encarados de duas formas: a primeira considera que esses tipos de objetos são disjuntos, ou seja, existem dois tipos específicos de objetos representados pelas estruturas `[(Autor), Título, Conferência, Páginas]` e `[(Autor), Título, Periódico, Volume, Páginas, Ano]`. Se por outro lado consideramos `Conferência`

²<http://dblp.uni-trier.de>

e Periódico como um mesmo tipo atômico Local de Publicação temos então a estrutura [(Autor), Título, Local de Publicação, Volume, Páginas, Ano] onde Volume e Ano figuram como elementos opcionais. Para esse exemplo de caso iremos considerar esses tipos de objeto como diferentes.

Após a extração dos valores atômicos constituintes dessas páginas, conforme descrito no capítulo anterior, é necessário combinar esses átomos em uma estrutura plausível de maneira a reconstituir os objetos complexos existentes nas páginas, sendo essa a tarefa do algoritmo HotCycles.

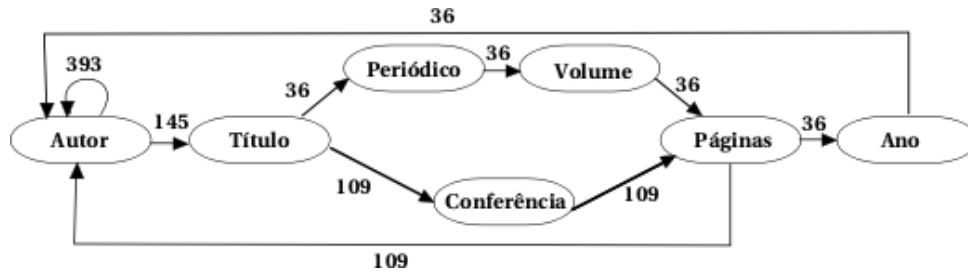


Figura 3.6: Grafo de adjacências inicial das páginas de autores do *site* DBLP

Inicialmente é calculado o grafo de adjacências correspondente à lista de ocorrências dos valores atômicos extraídos das páginas conforme pode ser visualizado na Figura 3.6. Em seguida, o algoritmo procura por auto-ciclos que possam identificar listas de elementos e localiza o vértice **Autor** que possui um auto-ciclo de frequência 393 sendo então substituído na lista de ocorrências. O algoritmo recalcula o grafo de adjacências conforme apresentado na Figura 3.7.

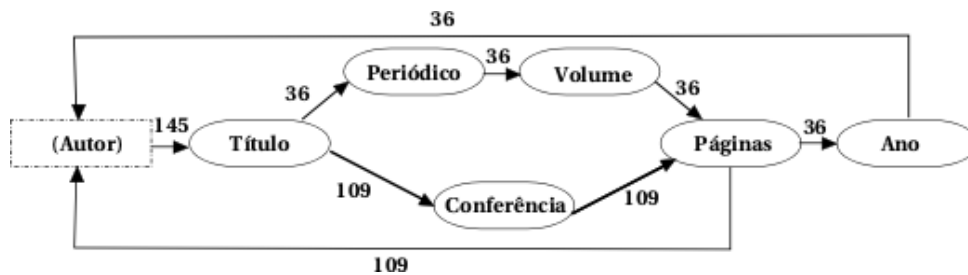


Figura 3.7: Montagem do objetos da página de autores da DBLP - Passo 01

Agora é possível identificar a existência de quatro ciclos:

- C_1 [(Autor), Título, Conferência, Páginas]
- C_2 [(Autor), Título, Conferência, Páginas, Ano]
- C_3 [(Autor), Título, Periódico, Volume, Páginas, Ano]
- C_4 [(Autor), Título, Periódico, Volume, Páginas]

O algoritmo então verifica qual ciclo é o mais freqüente e identifica que C_1 ocorre 109 vezes na lista de ocorrências. Em seguida tenta-se ampliar este ciclo para o ciclo C_2 . Porém ao se verificar a viabilidade de C_2 constata-se que o tipo plano definido por este ciclo não é encontrado na lista de ocorrências. Utiliza-se portanto o tipo definido por C_1 na substituição dos valores na lista de ocorrências.

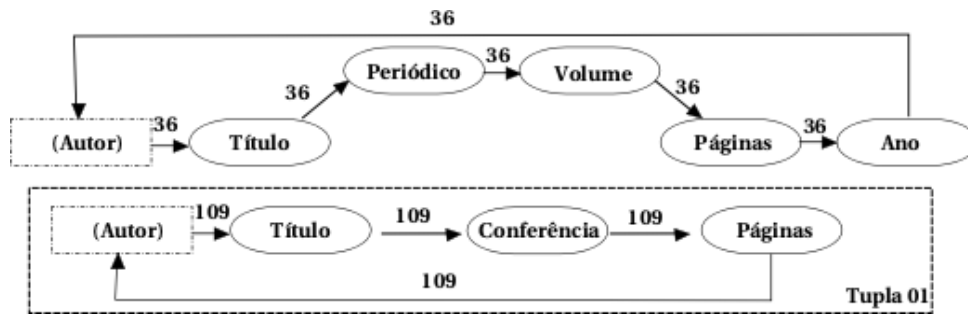


Figura 3.8: Montagem do objetos da página de autores da DBLP - Passo 02

Em seguida o ciclo C_3 é identificado conforme pode ser verificado na Figura 3.8. Ao se iniciar o processo de generalização do ciclo verifica-se que não existem mais outros ciclos disponíveis, utiliza-se portanto o ciclo C_3 na substituição dos valores na lista de ocorrências. Após a substituição desse ciclo, todos os valores atômicos já se encontram combinados e o algoritmo termina sua execução.

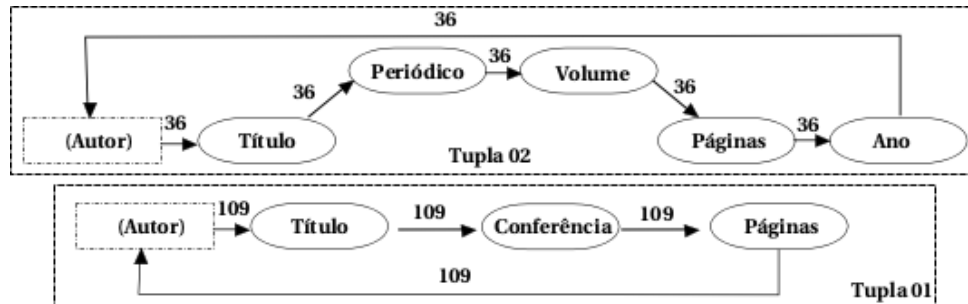


Figura 3.9: Montagem do objetos da página de autores da DBLP - Último Passo

Finalmente, a Figura 3.9 demonstra os dois ciclos estruturantes encontrados, C_1 e C_3 , onde cada ciclo possui associado um tipo plano expandido que define um nível na estrutura dos objetos extraídos das páginas. As estruturas dos objetos montados são, portanto, duas: $[(\text{Autor}), \text{Título}, \text{Conferência}, \text{Páginas}]$ e $[(\text{Autor}), \text{Título}, \text{Periódico}, \text{Volume}, \text{Páginas}, \text{Ano}]$.

Intuitivamente o algoritmo descobre pistas de construtores estruturais que não foram declara-

dos. Como será demonstrado em nossos experimentos no próximo capítulo, essas pistas são efetivas para a identificação de estruturas plausíveis para os objetos.

Capítulo 4

Resultados Experimentais

Neste capítulo apresentamos os resultados experimentais obtidos com o método de extração que foi descrito nos capítulos anteriores. Iniciamos com a apresentação da metodologia de avaliação adotada para mensurar a eficácia da abordagem proposta.

Os experimentos foram realizados para cada uma das etapas que constituem o método. Primeiramente, apresentamos os resultados do método de geração de padrões de extração, que é capaz de localizar valores atômicos que constituem os atributos dos objetos presentes na página HTML. Em seguida, apresentamos os resultados do algoritmo HotCyles que é responsável por remontar os objetos segundo uma estrutura plausível. No final do capítulo discutimos os resultados obtidos.

4.1 Metodologia de Avaliação

Um dos principais problemas na avaliação de algoritmos de extração de dados é a ausência de uma coleção de referência amplamente aceita e difundida. Assim, a comparação direta entre métodos fica em grande parte prejudicada devido a essa falta de padronização. Por isso, em nossos experimentos, procuramos selecionar coleções de páginas Web representativas, sendo que algumas delas foram utilizadas nos experimentos de outros métodos de extração publicados na literatura. Para o nosso trabalho elaboramos um processo de avaliação que é descrito a seguir.

Escolhemos inicialmente um conjunto de páginas que pertençam a uma mesma classe. Manualmente identificamos os atributos de interesse nas páginas e selecionamos alguns exemplos para execução do método de extração. Em seguida, refinamos manualmente o resultado dessa

extração removendo ou adicionando valores atômicos que não foram corretamente extraídos. Ao final desse processo temos um conjunto de dados classificados que representa o resultado de uma extração perfeita onde todos os valores estão corretamente identificados pelo seu conteúdo, página ao qual pertence e posição absoluta na página. A este conteúdo chamamos **coleção de referência**.

Quando a coleção de referência está montada, iniciamos a avaliação do método de extração fornecendo exemplos de treinamento e conferindo o resultado da extração de maneira automática de acordo com as seguintes políticas de seleção de exemplos:

1. **Execução sem realimentação:** são escolhidos inicialmente 2 exemplos aleatórios da coleção de referência. Em seguida é gerado um padrão de extração e o método executado. Verifica-se a qualidade da extração e, se ainda existirem valores que não foram corretamente extraídos, um novo exemplo é selecionado aleatoriamente sendo então utilizado para refinar o padrão de extração corrente. Esse processo se repete até que o número de exemplos sorteados atinja 10 ou que todos os valores sejam extraídos. Essa política simula um usuário que não dispõe de uma interface com recurso de realimentação para a execução do método.
2. **Execução com realimentação:** são escolhidos inicialmente 2 exemplos aleatórios da coleção de referência. Em seguida é gerado um padrão de extração e o método executado. Verifica-se a qualidade da extração e, se ainda existirem valores que não foram corretamente extraídos, um novo exemplo é selecionado dentre aqueles que ainda não foram cobertos pelo padrão de extração corrente. Esse processo se repete até que o número de exemplos sorteados atinja 10 ou que todos os elementos sejam extraídos. Essa política simula um usuário que possui uma interface que lhe proporciona o resultado da execução do método permitindo que sejam fornecidos novos exemplos que são de fato relevantes para a generalização do padrão de extração.
3. **Execução com realimentação e seleção preguiçosa de exemplos:** são utilizados inicialmente os 2 exemplos de treinamento da coleção de referência, seguindo a mesma ordem de ocorrência dos valores atômicos nas páginas HTML. O padrão de extração é então gerado e o método executado. Verifica-se a qualidade da extração e, se ainda existirem valores que não foram corretamente extraídos, utiliza-se o primeiro valor atômico da coleção de

referência não coberto pelo padrão de extração corrente. Esse processo se repete até que o número de exemplos selecionados atinja 10 ou que todos os valores sejam extraídos. Essa política simula um usuário que vai selecionando novos exemplos de maneira seqüencial de acordo com o resultado da extração apresentado pelo método.

Como as políticas 1 e 2 envolvem seleção aleatória de exemplos, executamos cada experimento que usa essas políticas 20 vezes. Devido a peculiaridade da política 3, realizamos sua execução uma única vez considerando como valores iniciais aqueles contidos no início da primeira página catalogada na coleção de referência.

Cada uma dessas políticas de seleção de exemplos apresenta resultados ligeiramente distintos. Quando utilizamos a realimentação percebemos os melhores resultados da nossa abordagem, uma vez que cada exemplo fornecido é de fato relevante, pois um novo exemplo é fornecido somente se este não foi ainda coberto pela expressão de extração atual. Esses resultados são apresentados na Seção 4.2.

4.1.1 Coleções Utilizadas

Foram escolhidas 21 fontes de páginas HTML, que foram classificadas manualmente seguindo os passos descritos anteriormente. Após a classificação dessas fontes e a montagem das respectivas coleções de referência, foram realizados os experimentos de extração de acordo com as políticas de seleção de exemplos discutidas. Nas seções seguintes são apresentados resultados sumarizados das extrações realizadas sobre estas coleções. Para cada uma das coleções são apresentadas características detalhadas no Apêndice A, assim como gráficos com os resultados experimentais da execução do método de extração.

Na Tabela 4.1, para cada coleção utilizada é identificado o número de atributos (**Atts**), o nível de aninhamento (**Nest**), a quantidade de atributos opcionais (**Opt**), a quantidade de listas existentes no tipo aninhado (**List**), bem como as referências a outros trabalhos em que as mesmas foram utilizadas.

4.1.2 Métrica de Avaliação

Para avaliar os resultados dos nossos experimentos utilizamos uma métrica bastante utilizada em trabalhos da área de recuperação da informação, a medida F_1

Coleção	Atts	Nest	Opt	List	Outros Trabalhos
All Politics	4	1	0	0	[Kushmerick, 2000, Muslea et al., 2001]
Amer. Psc. Ass.	9	1	1	1	
CDDb	6	2	0	1	
CD-NOW	4	1	0	0	[Laender et al., 2002b]
DBLP	7	2	2	1	[Laender et al., 2002b]
IMDB	7	2	0	1	
LOTERIA	11	2	0	2	
Monster	4	1	1	0	[Laender et al., 2002b]
PubMed	7	2	2	1	
Restaurants	7	2	3	1	[Muslea et al., 2001, Crescenzi et al., 2001]
DNC	4	1	0	0	[Kushmerick, 2000, Muslea et al., 2001]
Film.com	6	1	0	0	[Kushmerick, 2000, Muslea et al., 2001]
PharmWeb	15	1	4	0	[Kushmerick, 2000, Muslea et al., 2001]
News.com	4	1	0	0	[Kushmerick, 2000, Muslea et al., 2001]
Sigmod	7	3	0	2	
Sony	8	2	0	1	
VLDB	6	3	1	2	
Wine	6	2	2	1	[Crescenzi et al., 2001]
Yahoo-Phone	3	1	0	0	[Kushmerick, 2000, Muslea et al., 2001]
Yahoo-Movie	10	3	0	2	
Zagat	8	2	3	1	[Muslea et al., 2001]

Tabela 4.1: Coleções utilizadas nos experimentos

[Baeza-Yates and Ribeiro-Neto, 1999]. Essa métrica é definida como a média harmônica da precisão e da revocação:

$$F_1 = \frac{2 * rev * prc}{rev + prc} \quad (4.1)$$

onde revocação (*rev*) é uma medida que representa a porcentagem de valores atômicos existentes na coleção de referência que o método foi capaz de extrair. Precisão (*prc*), por sua vez, é uma medida que representa a porcentagem dos valores atômicos corretamente extraídos pelo método.

A medida F_1 foi utilizada nesse trabalho, pois ela é capaz de sumarizar a qualidade do processo de extração de átomos para um determinado número de exemplos utilizados. Consideramos a precisão e a revocação como medidas igualmente importantes, pois, em métodos de

extração de dados é interessante extrair a maior quantidade possível de objetos e ao mesmo tempo esses objetos devem possuir qualidade suficiente para permitir um processamento mais sofisticado da informação.

Para cada coleção traçamos curvas da medida F_1 em relação ao número de exemplos fornecidos para o método de extração, de maneira a demonstrar o comportamento do método conforme novos exemplos atômicos são fornecidos.

4.2 Experimentos de Extração

Nessa seção são discutidos os resultados obtidos com o método de extração nas várias coleções escolhidas para execução dos testes. Apresentamos aqui somente os resultados sumarizados das coleções utilizadas, com exceção da coleção DBLP para a qual mostraremos o desempenho de extração para cada um das políticas de seleção de exemplos consideradas.

No Apêndice A são apresentados os grafos comparativos da medida F_1 descrevendo o desempenho nas coleções utilizadas para cada uma das políticas de seleção de exemplos consideradas. Lá também são apresentados os gráfico da medida F_1 x número de exemplos para cada um dos atributos que constituem cada coleção de referência.

4.2.1 Resultado Geral

A Tabela 4.2 apresenta o desempenho geral do método de extração de átomos para cada uma das coleções utilizadas. Nessa tabela, para cada coleção é identificado o maior valor da média harmônica para as políticas de seleção de exemplos sem realimentação (**F-SR**), com realimentação (**F-CR**) e com realimentação preguiçosa (**F-CRP**) assim como a quantidade média de exemplos necessários para se obter um desempenho superior a 90% com cada política.

Percebemos na Tabela 4.2 que o método apresenta resultados acima de 94% de eficácia na extração de elementos atômicos com o fornecimento de no máximo 10 exemplos para cada atributo. Na prática, a maior parte dos atributos requerem não mais que 6 exemplos para a correta especificação do padrão de extração, conforme os gráficos apresentados no Apêndice A.

Verificamos que em alguns casos, mesmo com o fornecimento de 10 exemplos, não foi possível obter uma extração ideal. Acreditamos que isso se deve ao fato de utilizamos um algoritmo aproximado durante o processo de inferência de expressões. Conforme discutido no Capítulo 2,

Coleção	Melhor Resultado			Num. Ex. $F_1 > 0.9$		
	F-SR	F-CR	F-CRP	F-SR	F-CR	F-CRP
All Politics	0.936	0.996	0.999	7	5	6
APSA	0.957	0.997	0.998	4	3	4
CDDb	0.991	1	1	4	3	5
CD-NOW	0.963	1	1	3	2	2
DBLP	0.904	0.981	0.995	10	6	7
IMDB	0.985	1	1	2	2	2
LOTERIA	0.991	1	1	2	2	2
Monster	0.977	1	1	2	2	3
PubMed	0.928	0.989	0.996	6	5	4
Restaurants	0.881	0.942	0.795	10	5	10
DNC	0.994	1	1	2	2	2
Film.com	0.996	1	1	3	2	3
PharmWeb	0.993	1	1	2	2	2
News.com	0.987	1	1	3	2	3
Sigmod	0.991	1	1	2	2	2
Sony	0.966	1	1	3	3	2
VLDB	0.939	0.975	1	8	5	7
Wine	0.936	0.988	0.99	5	4	5
Yahoo-Phone	0.991	1	1	3	2	2
Yahoo-Movie	0.98	1	1	4	3	3
Zagat	0.957	0.997	0.999	6	4	4

Tabela 4.2: Resultado geral do método de extração de atributos atômicos

o método ideal para se alinhar k exemplos atômicos envolveria um alinhamento múltiplo com um custo de $O(n^k)$. Em nosso método, a cada alinhamento que realizamos geramos uma expressão de consenso que é realinhada com o próximo exemplo. Conforme novos exemplos são utilizados, há a tendência de nos afastarmos do alinhamento múltiplo ideal.

Em cada alinhamento progressivo, utilizamos o estado dos delimitadores (fixo, obrigatório e opcional) como marcadores para reduzir esse afastamento natural que ocorre devido à utilização de um método aproximado. Este fato associado à característica de relevância dos exemplos atômicos selecionados com a política de realimentação (feedback) fazem com que esta política se adeque melhor ao nosso método, pois com ela menos exemplos são necessários, reduzindo os erros incorporados pelo método de inferência de expressões.

Os resultados dos experimentos com as coleções DNC, Film.com, PharmWeb, News.com e Yahoo-Telephone utilizadas em [Kushmerick, 2000] e [Muslea et al., 2001] mostram que o nosso método foi capaz de inferir extratores com 100% de eficácia, utilizando as políticas com realimentação. Dessas cinco coleções, três obtiveram também um resultado de 100% nesses outros trabalhos. Além disso, em [Muslea et al., 2001] é necessário o fornecimento de exem-

plos para a definição da estrutura complexa dos objetos para cada uma dessas coleções e em [Kushmerick, 2000] é necessário um número muito maior de exemplos.

A coleção AllPolitics apresentou um índice de desempenho de 99,6%, utilizando a política de realimentação. Essa coleção não foi corretamente extraída em [Kushmerick, 2000] e em [Muslea et al., 2001] os atributos que envolviam contexto fraco chegaram a índices de 91% com a utilização de 10 exemplos.

As coleções Restaurants e Zagat são utilizadas em [Muslea et al., 2001]. Em nosso método conseguimos alcançar um índice de desempenho médio superior a 94% e 99% respectivamente, considerando o endereço de cada restaurante como um objeto complexo constituído por vários atributos. Ao contrário, em [Muslea et al., 2001] os endereços da coleção Restaurants são considerados como elementos atômicos, enquanto que na coleção Zagat a decomposição do endereço é incompleta considerando os atributos **número** e **rua** como um único elemento atômico.

As coleções DBLP, CD-NOW e Monster foram utilizadas em [Laender et al., 2002b]. Essa ferramenta utiliza-se de informações extras durante a geração dos padrões de extração e não considera contextos fracos. Na coleção CD-NOW e Monster, ambos os métodos apresentam um desempenho de 100%. Na coleção DBLP, a ferramenta DEByE gera regras disjuntas para cada uma possível variação na posição das instâncias do atributo **autor**. Em nosso método consideramos apenas um padrão de extração para cada atributo.

Na coleção Wine obtivemos um desempenho superior a 98%, conseguindo separar os atributos que constituem o nome, o local e a safra dos vinhos listados. Em [Crescenzi et al., 2001] esses atributos são podem ser extraídos separadamente, além do fato de extrair dados desnecessários, o que é típico nos métodos automáticos.

A Tabela 4.3 apresenta o tempo médio de geração de padrões de extração para cada uma das bases utilizadas. Nesses experimentos foi utilizado um computador com processador AMD XP2500+, com 512 MB de memória principal. Na tabela, o tempo de geração é apresentado em milisegundos e se refere ao processo de geração progressiva dos padrões de extração conforme a técnica apresentada na Seção 2.8. O tempo médio total de geração de extratores para cada coleção pode ser obtido pela multiplicação do valor na Tabela 4.3 pelo número de atributos que constituem os objetos de cada coleção.

Cada exemplo de treinamento possui um par de afixos (prefixo e sufixo), que definem o contexto sintático onde o mesmo está inserido. Em nossos experimentos, o tamanho desses afixos é

Coleção	Número de Delimitadores	Número de Exemplos Utilizados								
		2	3	4	5	6	7	8	9	10
All Politics	53	12	29	63	99	152	222	295	378	472
APSA	78	33	101	212	340	516	751	925	1180	1455
CDDDB	33	7	16	30	50	73	103	135	174	219
CDNOW	23	8	13	20	37	47	67	84	106	135
DBLP	56	11	30	61	100	154	223	300	384	487
IMDB	31	5	14	27	47	71	98	131	168	210
Loteria	30	10	19	31	51	75	103	136	174	215
Monster	25	6	18	24	34	46	61	85	107	129
Pubmed	43	8	22	46	78	115	163	217	281	352
Restaurants	75	24	79	165	274	412	574	771	984	1227
DNC	48	13	33	66	109	166	236	308	398	507
Film.com	72	20	65	137	235	358	505	680	874	1095
PharmWeb	33	5	15	30	50	75	104	137	175	219
News.com	54	12	33	68	118	176	253	335	438	547
Sigmod	31	6	15	28	46	68	96	126	163	201
Sony	31	7	14	29	47	71	99	135	173	214
VLDB	53	12	33	66	112	171	241	282	332	388
Wine	26	6	11	21	34	49	70	95	122	152
Yahoo-Movie	37	8	22	41	67	100	141	187	244	303
Yahoo-Telephone	75	25	77	164	272	401	561	753	945	1164
Zagat	48	9	26	53	91	137	192	252	326	401

Tabela 4.3: Tempo médio de geração dos padrões de extração (em milissegundos)

pré-definido como sendo formado pelos 400 caracteres mais próximos ao dado de interesse. Essa cadeia de caracteres é processada para a identificação dos delimitadores de texto e dos delimitadores de marcação. O número médio de delimitadores existentes em cada afixo é apresentado na segunda coluna da Tabela 4.3. Utilizamos ainda um limite máximo de 80 delimitadores que são considerados durante o processo de alinhamento das seqüências.

O número médio de delimitadores identificados nos afixos dos exemplos de treinamento de cada coleção é variável, pois reflete a estrutura do contexto sintático dos atributos de cada coleção. Se o contexto é formado principalmente por marcadores HTML com vários atributos ou por palavras muito grandes, temos a tendência de identificar, nessa janela fixa de 400 caracteres, um número menor de delimitadores, ao contrário, se o contexto é formado por inúmeros caracteres de pontuação, palavras pequenas, ou mesmo marcadores HTML curtos como ou <I>, temos a tendência de identificar um número maior de delimitadores.

Além desse número inicial de delimitadores identificados, conforme realizamos novos alinhamentos, espaços ou *gaps* são inseridos nas seqüências, o que tende a aumentar número de

delimitadores na expressão de consenso. Quanto maior o número de delimitadores considerados durante os alinhamentos, maior é o tempo necessário para se realizar o alinhamento dos contextos. A Tabela 4.3 apresenta o tempo médio de geração de um padrão de extração para cada uma das coleções estudadas.

4.2.2 Experimentos com a Coleção DBLP

Para dar uma idéia mais precisa do funcionamento do nosso método de extração, apresentaremos nesta subseção os resultados experimentais obtidos com as páginas de autores do *site* DBLP. Esse *site* contém uma lista de artigos publicados em conferências e em periódicos. Os objetos de interesse são formado pelos seguintes atributos: **autor**, **título**, **conferência**, **periódico**, **volume**, **páginas** e **ano**. A Figura 4.1 apresenta um exemplo dessas páginas.

John Adams	
List of publications from the DBLP Bibliography Server - FAQ	
Ask others: ACM - CiteSeer - CSB - Google - HomePageSearch	
2004	
4	EE Aidan Feeney, John Adams, Lara Webber, Michael Ewbank: Individual Differences in Graphical Reasoning. Diagrams 2004: 271-285
1996	
3	EE Avril Smith, John Adams, Geoff Tagg: Available Bit Rate - A New Service for ATM. Computer Networks and ISDN Systems 28(5): 635-640 (1996)
2	John Adams: Controversy: Rejoinder: Independent One-Time Passwords. Computing Systems 9(3): 255-256 (1996)
1974	
1	John Adams, John Gary: Compact Representation of Contour Plots for Phone Line Transmission. Commun. ACM 17(6): 333-336 (1974)

Figura 4.1: Página de autores do *site* DBLP

As páginas de autor do *site* DBLP apresentam inúmeros desafios ao processo de extração. As listas de autores são formadas por elementos de texto e por elementos que representam *hyperlinks* o que dificulta a definição do valor atômico correspondente. Além disso, somente o primeiro autor apresenta um contexto de extração definido, os outros são delimitados apenas por elementos opcionais. O contexto de extração dos outros atributos é definido por elementos de texto e delimitadores fracos o que dificulta o processo de extração.

Os objetos que estão dispostos nessas páginas constituem uma lista de publicações em conferências e em periódicos, onde cada publicação está localizada como uma célula de um elemento

HTML <TABLE>.

Abordagens automáticas [Arasu and Garcia-Molina, 2003, Crescenzi et al., 2001] são capazes de detectar cada uma dessas publicações, porém, não são capazes de identificar a estrutura interna de cada um desses objetos, extraindo a informação como um único elemento atômico. Outros métodos de inferência de extratores [Hsu and Dung, 1998, Kushmerick, 2000, Muslea et al., 2001] fazem suposições sobre a estrutura das páginas ou requerem que sejam fornecidos exemplos da estrutura dos objetos. Além disso, na maioria dos casos, simplesmente não são capazes de recuperar os valores atômicos em contextos fracos. Em nossa abordagem necessitamos apenas a seleção de alguns exemplos para que o método possa reconhecer os valores atômicos existentes nas páginas e em seguida combinar esses valores em uma estrutura plausível que represente os objetos.

Conforme discutido no exemplo do Capítulo 2, existem duas formas de se representar os objetos localizados nas páginas da DBLP. Para este experimento consideraremos que os objetos localizados nas páginas do DBLP são representados pela a estrutura [(Autor), Título, Local de Publicação, Volume, Páginas, Ano] onde Volume e Ano figuram como elementos opcionais. Passaremos a discutir cada um desses atributos.

Autor

Cada publicação possui associada uma lista de autores, esses autores ocorrem no início da descrição de uma publicação e podem ser constituídos por *links* (elementos <A HREF>) ou simplesmente por cadeias de caracteres. Cada autor é separado um do outro através do uso de vírgulas. Para a extração de cada autor em separado faz-se necessário a utilização de delimitadores fracos capazes de representar outros autores na lista enquanto os delimitadores fortes não são localizados.

Os delimitadores fortes desse tipo atômico são constituídos por elementos HTML que definem campos de tabelas <TD> e pelo caracter “:” que delimita o final da lista de autores. O padrão de extração gerado para este atributo é apresentada na Figura 4.2.

```
<TD>\s*(?:[\. \p{L}\p{Lu}\, \s]|(?:</?A\b[^\>]*>))*?[\, \s]*
  ((?:[\. \p{L}\p{Lu}\s]|(?:</?A\b[^\>]*>))+
\s*[\, \s]*(?:[\. \p{L}\p{Lu}\s]|(?:</?A\b[^\>]*>))*?:\s*
```

Figura 4.2: Padrão de extração para o atributo Autor

Título

No título de uma publicação existe vasta gama de possibilidades podendo ser utilizado qualquer tipo de caracter, logo o domínio dos dados não será de grande valia para a definição de uma expressão de extração. Contudo percebemos que um título ocorre sempre após a lista de autores, conseqüentemente, logo após o caracter “.” que pode ser utilizado como um delimitador forte, seguido de delimitadores fracos que representam parte da lista de autores.

O delimitador final do título é um link HTML que representa o periódico ou conferência onde o trabalho foi publicado. Utilizamos portanto o elemento `<A href="[ "]*">` como delimitador obrigatório e o domínio dos nomes de periódicos ou conferências como delimitador fraco, seguido por um numeral representando o ano da conferência ou a edição do periódico e pelo elemento `</A>`. O padrão de extração gerado para este atributo é apresentada na Figura 4.3.

```
\, \s*(?: [\.\p{L}\p{Lu}\, \s] | (?:</?A\b[^\>]*>))*?\p{L}+\s*
(?: [\.\p{L}\p{Lu}\, \s] | (?:</?A\b[^\>]*>))*?: \s*([^\>]+)
\s*<A href="[ "]*>\s*[\.\&(\p{L})\/\-\; \p{Lu}\s]*?\d+\s*</A>\s*
```

Figura 4.3: Padrão de extração para o atributo Título

Local de Publicação

O atributo local de publicação dependendo do caso pode representar um periódico ou uma conferência. Devido à vasta possibilidade de nomes possíveis para os locais de publicação não podemos fazer nenhuma restrição ao domínio dos seus dados. Sabemos contudo que um elemento deste tipo ocorre dentro de um *link* HTML, logo podemos utilizar a tag `<A href="[ "]*">` como delimitador forte, seguido por delimitadores fracos e opcionais que representam o título de um artigo.

Como delimitador terminal temos o respectivo marcador `</A>` seguido de delimitadores fracos que representam o intervalo de páginas onde o trabalho foi publicado, assim como o volume no caso do local de publicação se tratar de um periódico. O padrão é finalizada com delimitadores fortes que indicam fim de célula `</TD>`.

Na verdade poderíamos considerar os atributos `periódico` e `conferência` como sendo dois atributos distintos, assim como fizemos no exemplo do Capítulo 2, contudo, neste caso seria possível identificar dois tipos complexos ao invés de um só como ocorre quando consideramos

estes atributos representado um local de publicação. O padrão de extração gerado para este atributo é apresentada na Figura 4.4.

```
\:\s*[(\:\p{L})]\-\p{Lu}\s]*?\p{L}+\s*[\.\?]\pLu\s]*?<A href="[^\"]*">\s*
([\^<>]+)
\s*</A>\s*[(\:\p{L})]\-\p{Lu}\s]*?</TD>\s*
```

Figura 4.4: Padrão de extração para o atributo Local de Publicação

Volume

Este elemento indica o volume da edição em que foi publicado o trabalho em um periódico. Como é formado apenas por dígitos e pelo caracter “-”, o domínio dos dados é de grande valia durante a definição do padrão de extração. O prefixo desse elemento é definido pelo caracter “(” e pelo título do periódico que ocorre sempre como um *link* HTML, por isso usamos `</A>` como delimitador forte seguido por um numeral que representa o número de edição do periódico e um delimitador fraco que representa o nome do periódico. O sufixo é definido pela sequência “):” seguida de um numeral que representa o início do intervalo de páginas onde o trabalho foi publicado. O padrão de extração gerada para este atributo é apresentada na Figura 4.5.

```
\.\s*(?:[\.\pL\s]|(?:</?A\b[\^<>]*>))*?\d+\s*</A>\s*\s*
([\-\d]+)
\s*\)\s*:\s*\d+\s*\-\s*
```

Figura 4.5: Padrão de extração para o atributo Volume

Páginas

Este elemento representa o intervalo de páginas onde o trabalho foi publicado, seja num periódico ou em anais de uma conferência. O domínio dos dados pode ser utilizado para facilitar a extração. O prefixo é definido pelo delimitador forte representado pelo caracter “:” seguido por um elemento opcional que representa o atributo volume e finalmente pelo marcador `</A>` seguido por um numeral que representa o ano da conferência ou a edição do periódico. O sufixo é definido pelo ano que depende do tipo de publicação e pelos delimitadores fortes `</TD>` e `</TR>`. O padrão de extração gerado para este atributo é apresentada na Figura 4.6.

Ano

Finalmente, o último atributo de interesse das páginas do *site* DBLP, o atributo ano é

```

\d+\s*</A>\s*[\(\)\-]\d\s]*?:\s*
([\-]\d)+
\s*[\(\s]*?[\)]\d\s]*?</TD>\s*</TR>\s*

```

Figura 4.6: Padrão de extração para o atributo Páginas

representado por um numeral apenas, sendo seu domínio utilizado para definir a máscara de extração para os dados. O prefixo é determinado pelo caracter “(” seguido pelo intervalo de páginas. Como sufixo temos os delimitadores fortes que indicam fim de célula </TD> e fim de linha </TR> de uma tabela. O padrão de extração gerada para este atributo é apresentada na Figura 4.7.

```

\d+\s*\-]\s*\d+\s*\(\s*
([\d]+)
\s*\)\s*</TD>\s*</TR>\s*

```

Figura 4.7: Padrão de extração para o atributo Ano

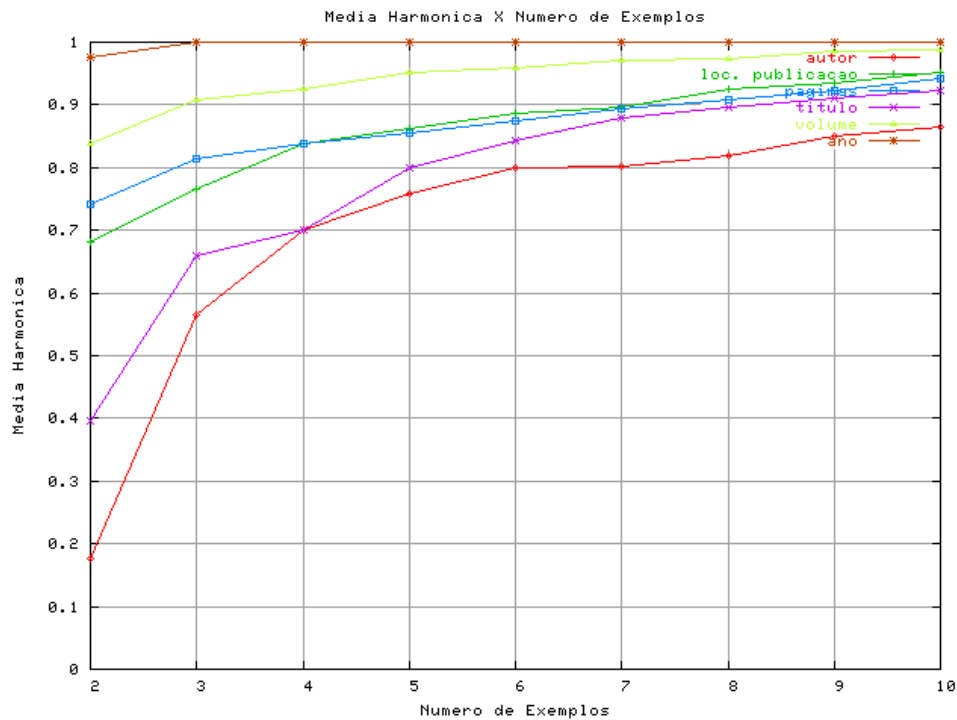


Figura 4.8: Resultados com a coleção DBLP por atributo sem realimentação

Nos gráficos das Figuras 4.8, 4.9 e 4.10 verificamos o desempenho do método de extração para todos os atributos em cada uma das políticas de seleção de exemplos. Percebemos no

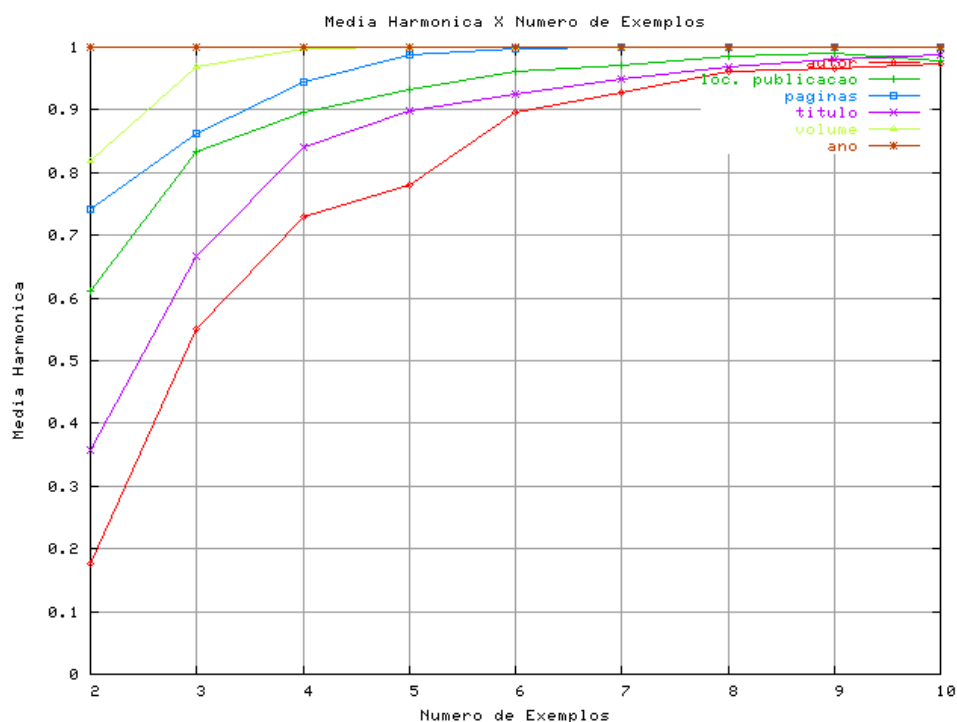


Figura 4.9: Resultados com a coleção DBLP por atributo com realimentação

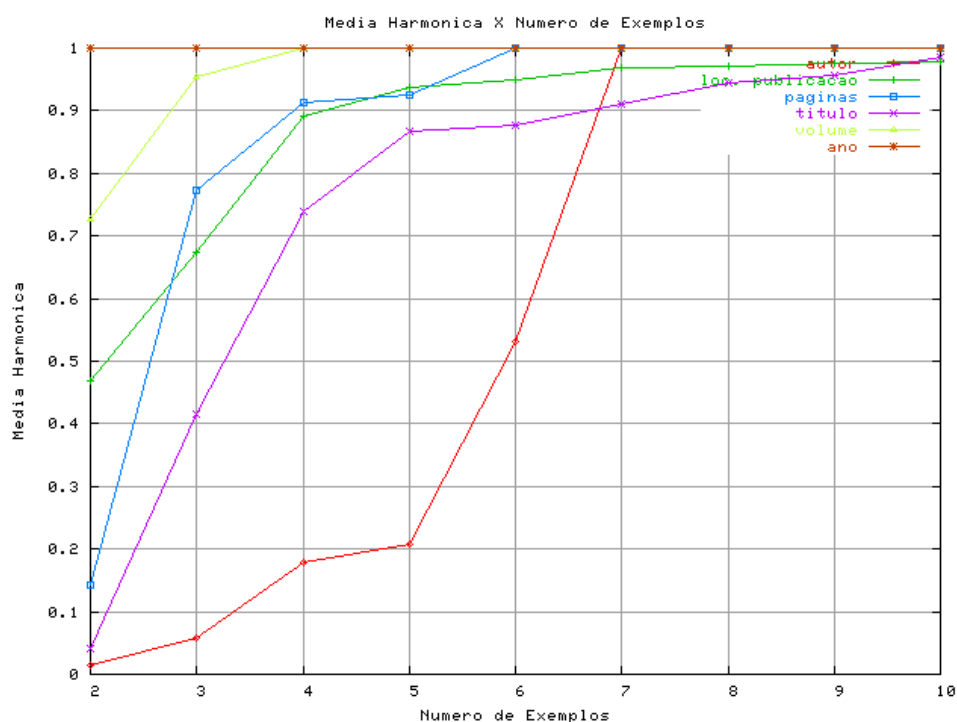


Figura 4.10: Resultados com a coleção DBLP por atributo com realimentação preguiçosa

gráfico da Figura 4.11 a influência da política de seleção de exemplos nos resultados da extração dos elementos.

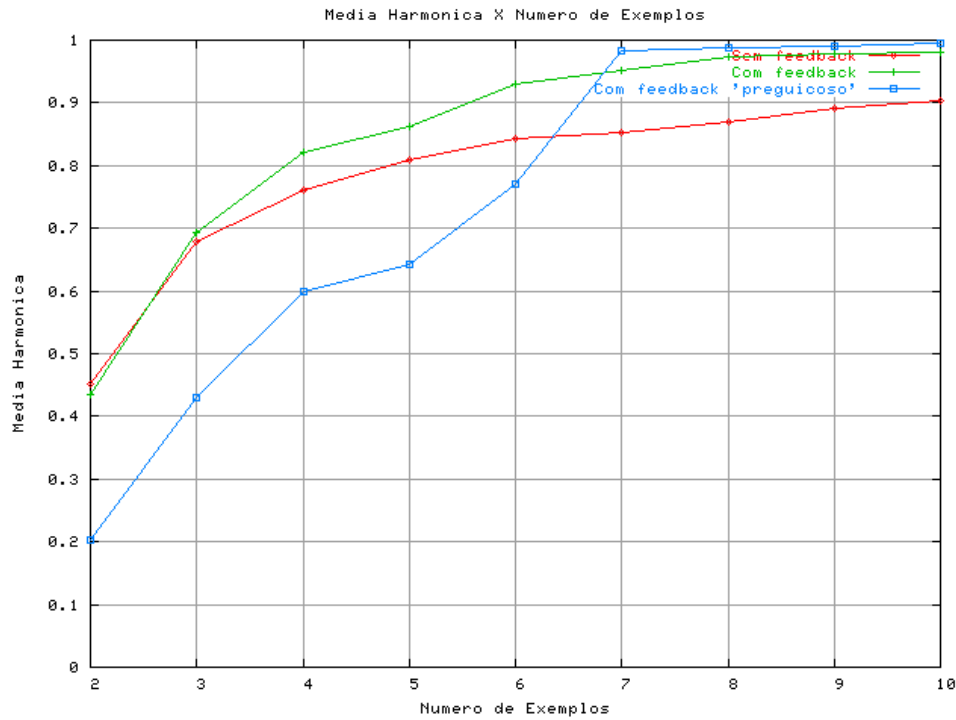


Figura 4.11: Resultados gerais obtidos na coleção DBLP

A utilização de realimentação permite que as expressões consigam melhorar seu desempenho mais rapidamente necessitando menos exemplos do que quando comparadas com as geradas pela política de seleção aleatória. Isso acontece pois com a utilização de realimentação cada novo exemplo é de fato representativo possuindo características ainda não incorporadas à expressão regular.

As tendências verificadas nos resultados com a coleção DBLP se repetem em cada uma das coleções testadas, os gráficos de desempenho por atributos para a política de seleção de exemplos com realimentação são os que apresentam melhores resultados. O uso de realimentação é justamente a política que se adequa melhor à nossa abordagem. O gráfico comparativo de desempenho com as três políticas de seleção de exemplos estudadas, bem como o gráfico por atributos utilizando realimentação são apresentados para cada uma das coleções no Apêndice A.

4.3 Experimentos de Montagem

Nesta seção apresentaremos os resultados obtidos nos testes do algoritmo HotCycles que realizamos sobre as coleções que foram catalogadas. Inicialmente, o método de montagem foi executado sobre a coleção de referência, ou seja, utilizando o resultado de um processo de ex-

tração ideal de valores atômicos. Em seguida, realizamos novos experimentos sobre os resultados da execução do nosso método de geração de padrões de extração, o que nos permitiu avaliar o resultado global de nosso método de geração de extratores.

4.3.1 Desempenho Isolado

Nesse experimento, foram utilizadas as coleções que foram manualmente catalogadas de modo a verificar a viabilidade do método de inferência para um processo ideal de extração de valores atômicos. Os resultados mostram que o método de montagem inferiu corretamente uma estrutura plausível para os objetos em todas as coleções de referência. A Tabela 4.4 apresenta para cada uma dessas coleções a estrutura inferida pelo método.

4.3.2 Desempenho Combinado

Verificamos que o algoritmo de montagem de objetos complexos é efetivo quando consideramos uma extração perfeita. Iremos agora demonstrar os resultados obtidos pela utilização do método de montagem sobre os resultados do nosso método de extração de atributos atômicos de maneira a estudar a viabilidade do método como um todo.

Para esse experimento foram consideradas como estruturas válidas somente aquelas iguais às da Tabela 4.4 com tolerância de erro em apenas um tipo atômico identificado como lista. Por exemplo, com relação ao tipo alvo `[produtor, ([ano, tipo, origem, descrição, preço])]` a estrutura `[produtor, ([ano, tipo, origem, (descrição), preço])]` é tolerada, já a estrutura `[produtor, ([ano, (tipo), origem, (descrição), preço])]` não é considerada correta.

Serão apresentados a seguir os resultados obtidos pela utilização do algoritmo HotCycles nos resultados da extração de valores atômicos para a política de seleção de exemplos com realimentação. Posteriormente discutiremos com detalhes os resultados obtidos para as páginas da coleção APSA. Os resultados serão apresentados em forma de tabela indicando o desempenho percentual da quantidade de estruturas corretamente inferidas pela quantidade de exemplos fornecidos.

Fonte	Estrutura Inferida
All Politics Search Results	[título, mês, dia, ano]
American Psychoanalytic Assoc.	[autores, ano, livro, cidade, editora] [autores, ano, artigo, jornal, edição, páginas]
CDDB	[cantor, ano, título, rótulo, ([posição, faixa])]
CDNOW	[nome, tipo, preço, desconto]
DBLP	[(autor), título, local de publicação, volume, páginas, ano]
IMDB	[década, tipo de ranking, ([filme, ano, média, posição, votos])]
Loteria Esportiva	concurso, data, ([número do jogo, gols do primeiro, primeiro time, segundo time, gols do segundo]), ([número de acertos, ganhadores, prêmio]), estimativa de prêmio
Monster	[data, lugar, emprego, país]
PUBMED	[(autor), título, jornal, data, edição, páginas, PID]
LA Weekly Restaurants	[restaurante, número, logradouro bairro, telefone, descrição, (cartão)]
DNC	[arquivo, fonte, tamanho, tipo]
Film.com	[título, resumo, url, tamanho, data, hora]
PharmWeb	[faculdade, universidade, responsável, endereço, complemento, cidade, cep, estado, país, fone, fax, url, update dia, update hora, update pers]
News.COM	[título, mês, dia, ano]
Sigmod	[mês, ano, número, volume, ([seção, ([autores, artigos])])]
Sony	[ano, álbum, artista, referência, selo, tipo, ([música, faixa])]
VLDB	[ano, cidade, país, ([seção, ([autor), artigo])])]
Wine Producers	[produtor, ([ano, tipo, origem, descrição, preço])]
Yahoo-Movie	[cinema, cidade, complemento, estado, logradouro, telefone, ([sala, lotação, filme, (horário)])]
Yahoo-Telephone	[nome, endereço, telefone]
Zagat Restaurants	[nome, cozinha, ([número, logradouro, complemento, estado, fone]), descrição]

Tabela 4.4: Estruturas inferidas corretamente pelo método sobre as bases de referência

4.3.3 Resultado Geral

A Tabela 4.5 apresenta o desempenho geral do método de montagem de objetos para cada uma das coleções catalogadas de acordo com a utilização da política de seleção de exemplos

Coleção	Número de Exemplos Fornecidos								
	2	3	4	5	6	7	8	9	10
All Politics	15%	50%	75%	85%	95%	100%	100%	100%	100%
APSA	0%	5%	15%	20%	40%	60%	65%	90%	90%
CDDDB	50%	75%	95%	100%	100%	100%	100%	100%	100%
CD-NOW	10%	50%	90%	100%	100%	100%	100%	100%	100%
DBLP	0%	15%	40%	60%	100%	95%	100%	95%	95%
IMDB	5%	10%	50%	95%	100%	100%	100%	100%	100%
Loteria	0%	35%	100%	100%	100%	100%	100%	100%	100%
Monster	50%	85%	100%	100%	100%	100%	100%	100%	100%
PubMed	35%	35%	60%	70%	80%	80%	85%	90%	95%
Restaurants	0%	40%	90%	80%	100%	100%	100%	100%	100%
DNC	10%	100%	100%	100%	100%	100%	100%	100%	100%
Film.com	60%	90%	100%	100%	100%	100%	100%	100%	100%
PharmWeb	5%	90%	100%	100%	100%	100%	100%	100%	100%
News.com	85%	95%	95%	100%	100%	100%	100%	100%	100%
Sigmod	25%	100%	100%	100%	100%	100%	100%	100%	100%
Sony	30%	25%	55%	85%	100%	100%	100%	100%	100%
VLDB	0%	0%	5%	25%	30%	50%	65%	65%	75%
Wine	95%	100%	95%	90%	90%	95%	100%	100%	100%
Yahoo-Movie	0%	90%	100%	100%	100%	100%	100%	100%	100%
Yahoo-Phone	60%	100%	100%	100%	100%	100%	100%	100%	100%
Zagat	0%	0%	5%	40%	85%	80%	95%	100%	100%

Tabela 4.5: Percentual de estruturas corretamente inferidas por número de exemplos

com realimentação. Essa tabela indica o percentual de estruturas corretamente inferidas pelo algoritmo para cada uma das coleções de acordo com o número de exemplos de treinamento fornecidos.

Foram utilizadas nesse experimento os resultados obtidos nos experimentos da seção anterior considerando a política de seleção de exemplos com realimentação, onde para cada coleção existente foram executados 20 experimentos de extração. Com base nos valores atômicos extraídos em cada um desses experimentos executamos o algoritmo HotCycles para realizar a montagem dos objetos complexos. Em seguida verificamos se as estruturas obtidas durante as montagens correspondem à estrutura definida como estrutura alvo e, dessa forma, construímos a Tabela 4.5.

Na Tabela 4.5 é possível verificar que o algoritmo HotCycles é capaz de inferir, para a maioria das coleções, 100% de estruturas plausíveis com a utilização de apenas 6 exemplos. O algoritmo HotCycles identifica padrões na distribuição dos valores atômicos extraídos e a partir desses padrões, identifica uma estrutura para se agrupar esses valores. Quando o desempenho do

processo de extração está baixo, o suporte para a geração do grafo de adjacências fica prejudicado e isso se reflete na produção de estruturas desconexas. Conforme pode ser verificado na Tabela 4.5, quando são utilizados 2 ou 3 exemplos o percentual de estruturas plausíveis é muito baixo.

Conforme o desempenho no resultado do processo de extração melhora, através do fornecimento de novos exemplos de treinamento, observamos uma tendência de aumento percentual nas estruturas plausíveis inferidas. Em 4 coleções, contudo, não obtivemos um desempenho ideal, mesmo com a utilização de 10 exemplos.

Em alguns casos nas coleções DBLP e PubMed o ciclo estruturante não foi completamente generalizado. Isso aconteceu em casos onde o ciclo a ser ampliado continha todos os vértices do grafo de adjacências. Nesses casos não se pode utilizar o critério de atributos de entrada e atributos de saída, uma vez que todos os atributos fazem parte do ciclo, restando somente o critério de número de ocorrências que nesses casos não foi suficiente.

Na coleção VLDB em alguns casos a estrutura aninhada não foi corretamente inferida devido a falhas no processo de extração de valores atômicos. Alguns valores atômicos incorretamente extraídos foram suficientes para interferir na geração do grafo de adjacências e fazer com que o algoritmo HotCycles falhasse na detecção da estrutura aninhada. A coleção APSA será discutida na próxima seção.

Nas abordagens automáticas [Crescenzi et al., 2001, Arasu and Garcia-Molina, 2003] a estrutura inferida para os objetos reflete a estrutura das páginas, o que nem sempre é suficiente. A ferramenta WIEN [Kushmerick, 2000] faz várias considerações sobre as estruturas possíveis para as páginas o que restringe sua aplicação. Em Stalker [Muslea et al., 2001] é essencial a definição de exemplos para a estrutura complexa existente de maneira a construir o chamado “*embedded catalog*”. Em DEByE [Laender et al., 2002b], Lixto [Baumgartner et al., 2001] e NoDoSE [Adelberg, 1998] o usuário deve explicitamente fornecer a estrutura dos objetos.

Em nosso método nada disso é necessário, pois uma estrutura plausível latente indicada pela própria distribuição dos valores atômicos extraídos é identificada e utilizada para acomodar esses átomos de maneira automática. Percebemos através da Tabela 4.5 que o algoritmo HotCycles conseguiu inferir um alto índice de estruturas plausíveis para as coleções estudadas chegando a um percentual de 100% de estruturas corretamente inferidas para 17 das 21 coleções. Quanto maior o número de exemplos de treinamento melhores são os resultados do método de extração de valores atômicos e conseqüentemente melhor é o suporte para o método de montagem de

objetos.

4.3.4 Experimentos com a Coleção APSA

Nesta seção apresentaremos os resultados experimentais obtidos com as páginas do *site* APSA¹. Será apresentado e discutido o gráfico que contém a porcentagem de estruturas corretamente inferidas de acordo com o número de exemplos fornecidos para o método de extração de valores atômicos.

Páginas do site APSA são formadas por uma lista de livros e de artigos publicados em periódicos. Os objetos de interesse são representados por duas estruturas: [autores, ano, livro, cidade, editora] e [autores, ano, artigo, jornal, edição, páginas]. A Figura 4.12 apresenta um exemplo dessas páginas.

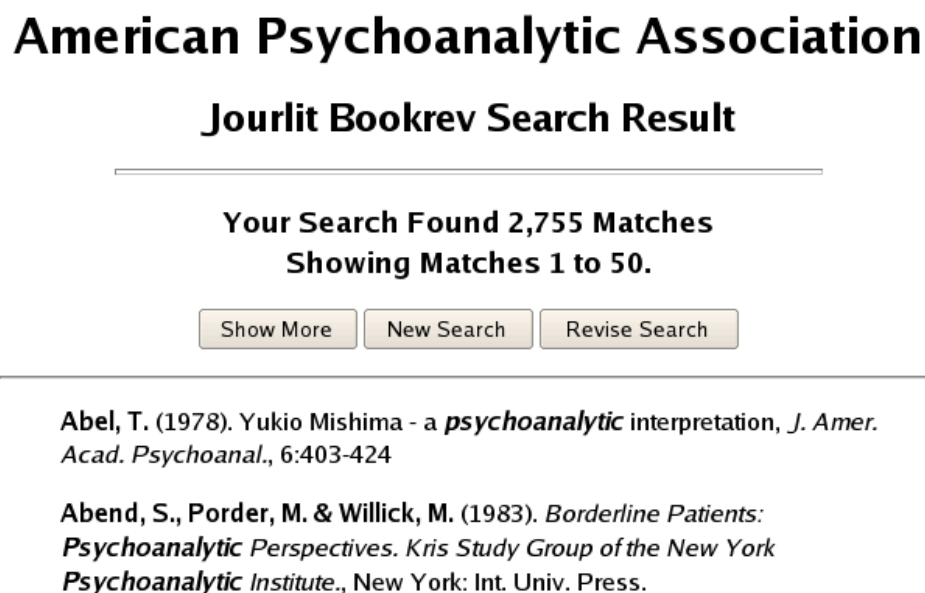


Figura 4.12: Página do site da APSA

As páginas do *site* APSA apresentam desafios interessantes para o processo de extração. A utilização de elementos de marcação HTML no interior dos valores atômicos, bem como a presença de delimitadores fracos entre os atributos dos objetos fazem com que o processo de extração seja bastante dificultado.

A Figura 4.13 indica o percentual de estruturas corretamente inferidas para a coleção de acordo com a quantidade máxima de exemplos fornecidos para os atributos. Para dois exemplos

¹<http://www.apsa.org>

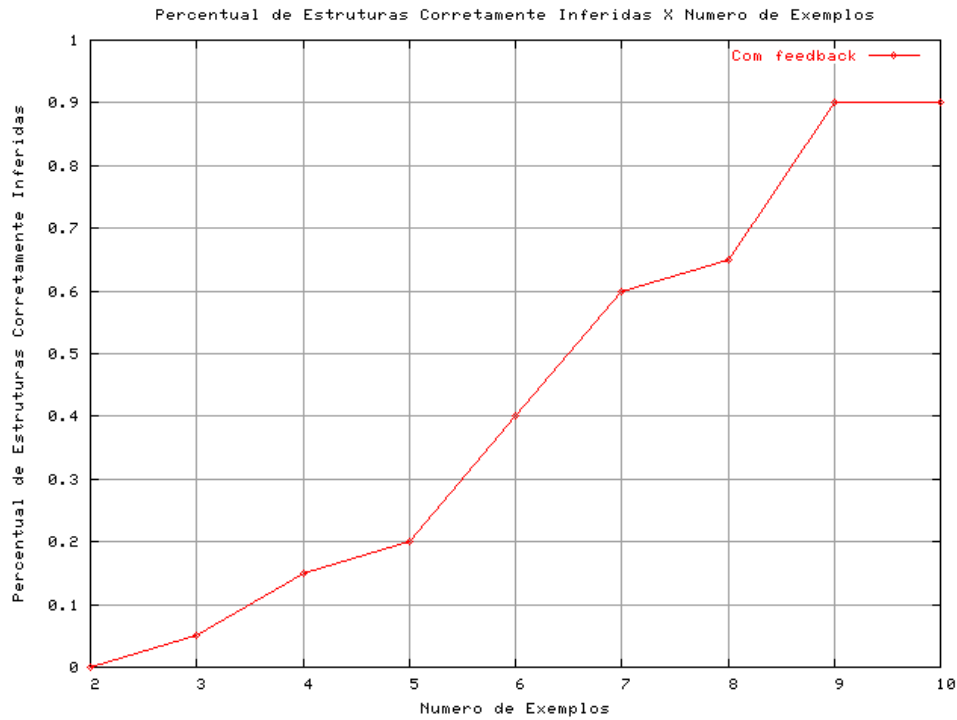


Figura 4.13: Resultados obtidos na coleção APSA

por atributo o método de inferência de estrutura não foi capaz de identificar nenhuma estrutura corretamente. A quantidade de estruturas corretamente inferidas vai aumentando progressivamente conforme novos exemplos são fornecidos. Por exemplo, para 5 exemplos temos um acerto de 20% nas estruturas definidas com 7 exemplos temos 60%. O percentual máximo de estruturas corretamente inferidas para essa coleção chegou a 90% com a utilização de no máximo 9 exemplos por atributo.

Nessa coleção, o algoritmo HotCycles não foi capaz de identificar em alguns casos a disposição correta dos atributos dentro da tupla. A utilização da disposição mais frequente do ciclo estruturante nesses casos não foi suficiente para determinar corretamente a ordem de precedência entre os atributos. Em alguns casos também ocorreu a incorreta generalização de ciclos estruturantes pela ampliação para ciclos com frequências baixas. Nestes casos, o critério de viabilidade do ciclo, $F(\varrho) > 1$ não foi suficiente para descartar ciclos inviáveis.

4.4 Resultado Final

A Figura 4.14 apresenta um gráfico com desempenho percentual médio do método de inferência de estrutura utilizando os resultados da extração de valores atômicos nas 21 coleções utilizadas

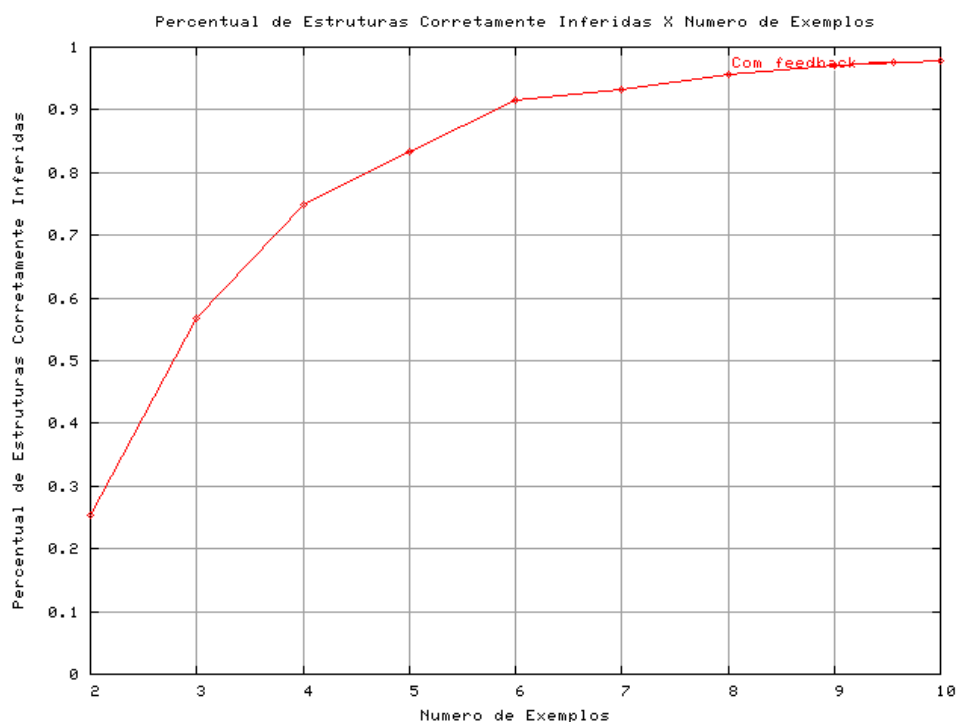


Figura 4.14: Resultado geral do método de inferência de estrutura

que foram apresentados na Tabela 4.5.

Nesse gráfico podemos acompanhar o progressivo aumento no número de estruturas corretamente inferidas conforme são fornecidos novos exemplos. Isso ocorre pois quanto mais exemplos representativos são fornecidos para o algoritmo de geração de padrões de extração, melhor é o resultado do processo de extração de valores atômicos e com isso melhor é o suporte do grafo de adjacências correspondente à lista de ocorrências.

Percebemos aqui a grande dependência do método de inferência de estrutura em relação ao método de extração dos átomos, que é o limitador principal, pois não se pode esperar que a qualidade da montagem dos objetos extraídos seja melhor que a qualidade do processo de extração dos valores atômicos.

O algoritmo HotCycles apresenta uma boa estabilidade em relação às estruturas inferidas sendo capaz de identificar estruturas plausíveis mesmo quando valores atômicos estão ausentes. A possibilidade de erros por parte do extrator de valores atômicos não foi considerada, sendo portanto reduzida a tolerância a falhas. Se o método de extração de atributos não for confiável e sua precisão estiver baixa o processo de inferência de estrutura fica prejudicado. Se a estrutura dos objetos permitir variação na posição dos atributos o método também não é capaz de montar

corretamente os objetos.

O algoritmo de geração de padrões de extração apresenta seu melhor desempenho quando exemplos representativos ainda não selecionados são fornecidos, conforme os experimentos utilizando a política de seleção de exemplos com realimentação. A incorporação desses dois métodos, o de extração e o de montagem, em uma ferramenta gráfica com suporte à seleção de exemplos com realimentação apresenta-se como uma alternativa extremamente viável e promissora.

Capítulo 5

Conclusões e Trabalhos Futuros

Neste trabalho desenvolvemos uma nova abordagem para o problema de geração de extratores de dados para páginas da Web. Nessa abordagem o usuário se restringe a fornecer alguns exemplos de treinamento para os atributos que constituem os objetos de interesse de maneira a possibilitar a geração de padrões de extração eficientes e permitir a posterior montagem dos objetos extraídos.

A estratégia semi-automática de geração de extratores de dados permite que apenas os dados realmente relevantes ao usuário sejam extraídos, diferentemente das abordagens completamente automáticas que misturam os objetos de interesse com elementos desnecessários constituintes das páginas.

A idéia de considerar o contexto de cada exemplo fornecido pelo usuário como um conjunto de delimitadores e de se encarar o problema de inferência de padrões de extração como um problema de alinhamento múltiplo de seqüências envolvendo esses contextos demonstrou-se bastante efetiva na prática. Tal abordagem trata de maneira conveniente o problema de geração de padrões de extração envolvendo contextos fracos, alcançando elevados índices de desempenho em todas as coleções testadas.

A utilização desse método de geração de padrões de extração com uma política de seleção de exemplos com realimentação apresentou os melhores resultados, alcançando índices de desempenho superior a 94%. Nesse sentido o apoio de uma interface gráfica que facilite o processo de seleção de exemplos bem como a visualização dos resultados por parte do usuário é fundamental.

O algoritmo HotCycles demonstrou-se bastante efetivo na inferência automática de estruturas para se organizar os valores atômicos extraídos, conseguindo inferir 100% de estruturas

plausíveis em 17 coleções testadas. É interessante salientar que a única evidência utilizada pelo algoritmo é a disposição dos elementos extraídos das páginas. Nenhum conhecimento prévio sobre os objetos é necessário para se inferir corretamente estruturas plausíveis.

Como trabalhos futuros podemos elencar três vertentes. Primeiramente, temos a necessidade de uma interface com o usuário que dê suporte a nova abordagem, trabalho este que já está sendo desenvolvido. Em seguida temos o desafio de introduzir no algoritmo HotCycles algum modo de controle e correção de possíveis falhas do processo de extração de átomos de maneira que durante a montagem dos objetos somente aqueles considerados corretos sejam reconstruídos. Por último a realização de novos testes com novas coleções de páginas, com extensão dos experimentos considerando realimentação preguiçosa, que é o que mais se assemelha ao comportamento de um usuário padrão, além da avaliação das curvas de precisão e revocação do resultados dos objetos montados pelo algoritmo HotCycles.

A combinação dos dois métodos aqui descritos mostrou-se extremamente viável, conforme demonstrado pelos experimentos utilizando o HotCycles nos resultados de extrações realizadas através de padrões de extração gerados pelo método baseado em alinhamentos progressivos de contextos. Os altos índices de estruturas corretamente inferidas juntamente aos elevados índices de precisão e revocação do processo de extração demonstram que esta é sem dúvida uma abordagem promissora.

Apêndice A

Detalhamento dos Experimentos com as Coleções Utilizadas

Neste apêndice são apresentados os resultados dos experimentos de extração de valores atômicos para as 21 coleções utilizadas. Para cada uma das coleções utilizadas será exibida uma figura que representa parte de uma página daquela coleção, bem como uma tabela que resume suas características. Nela é possível identificar a quantidade de páginas da coleção, o número de objetos existentes e a quantidade de instâncias de cada um dos atributos que compõem os objetos nas páginas bem como uma breve descrição sobre o conteúdo das páginas da coleção.

A estrutura dos objetos que constituem cada coleção também é apresentada. Neste trabalho utilizamos a seguinte notação para representar os elementos que constituem um tipo aninhado:

1. elemento atômico - representado pelo próprio identificador do atributo;
2. lista - representado por `()`;
3. tupla - representado por `[]`.

Para cada coleção são exibidos os resultados obtidos durante os experimentos. Dois gráficos são apresentados: o primeiro contém os resultados da extração de valores atômicos para cada atributo existente na coleção, de acordo com a política de seleção de exemplos com realimentação, o segundo trata-se de um gráfico comparativo do desempenho do método de extração de valores atômicos para cada uma das políticas de seleção de exemplos utilizadas. Neles é possível verificar que os melhores resultados foram alcançados utilizando as políticas que implementam algum tipo de realimentação de exemplos.

AllPolitics

AllPolitics Search Results

[home](#) | [news](#) | [in-depth](#) | [analysis](#) | [what's new](#) | [bulletin boards](#) | [contents](#) | [search](#)

Results for terms gore:

1. [AllPolitics - The Celluloid Senator - August 4, 1997](#)
2. [AllPolitics - Quotes From Budget Pact - July 30, 1997](#)
3. [AllPolitics - FDCH Transcript - JULY 29, 1997](#)
4. [AllPolitics - FDCH Transcript - JULY 29, 1997](#)
5. [AllPolitics - Sights & Sounds - July 28, 1997](#)
6. [AllPolitics - Lott Renews Call For Independent Counsel - July 28, 1997](#)
7. [AllPolitics - Campaign Fundraising Hearings: Connect The Dots - July 28, 1997](#)
8. [AllPolitics - Clinton Unveils Plan To Clean Up Lake Tahoe - July 26, 1997](#)
9. [AllPolitics - Friday July 25, 1997](#)
10. [AllPolitics - FDCH Transcript - JULY 25, 1997](#)

Número de Páginas	4
Número de Objetos	200
Número de Atributos	título=198, mês=200, dia=200, ano=200
Descrição Resumida:	Lista de artigos sobre política do site All Politics
Estrutura dos Objetos:	[título, mês, dia, ano]

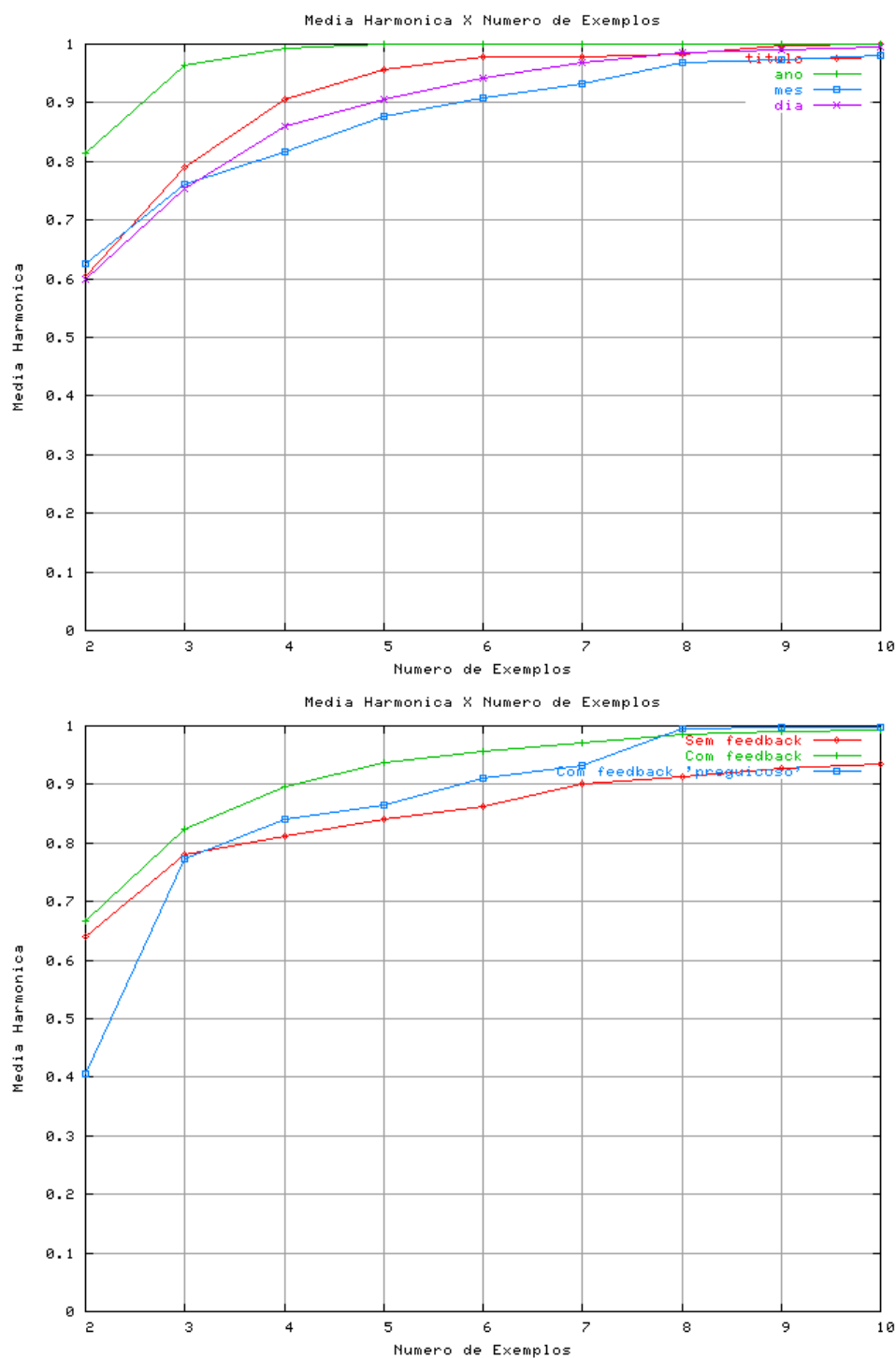


Figura A.1: Resultados obtidos na coleção do All Politics

APSA

American Psychoanalytic Association

Jourlit Bookrev Search Result

Your Search Found 2,755 Matches

Showing Matches 1 to 50.

[Show More](#)[New Search](#)[Revise Search](#)

Abel, T. (1978). Yukio Mishima - a *psychoanalytic* interpretation, *J. Amer. Acad. Psychoanal.*, 6:403-424

Abend, S., Porder, M. & Willick, M. (1983). *Borderline Patients: Psychoanalytic Perspectives*. Kris Study Group of the New York Psychoanalytic Institute., New York: Int. Univ. Press.

Número de Páginas 6

Número de Objetos 300

Número de Atributos: ano=300, artigo=257, autores=298, cidade=43, edição=257, editora=43, jornal=257, livro=43, páginas=257

Descrição Resumida: Publicações na American Psychoanalytic Association.

Estrutura dos Objetos: [autores, ano, livro, cidade, editora]
[autores, ano, artigo, jornal, edição, páginas]

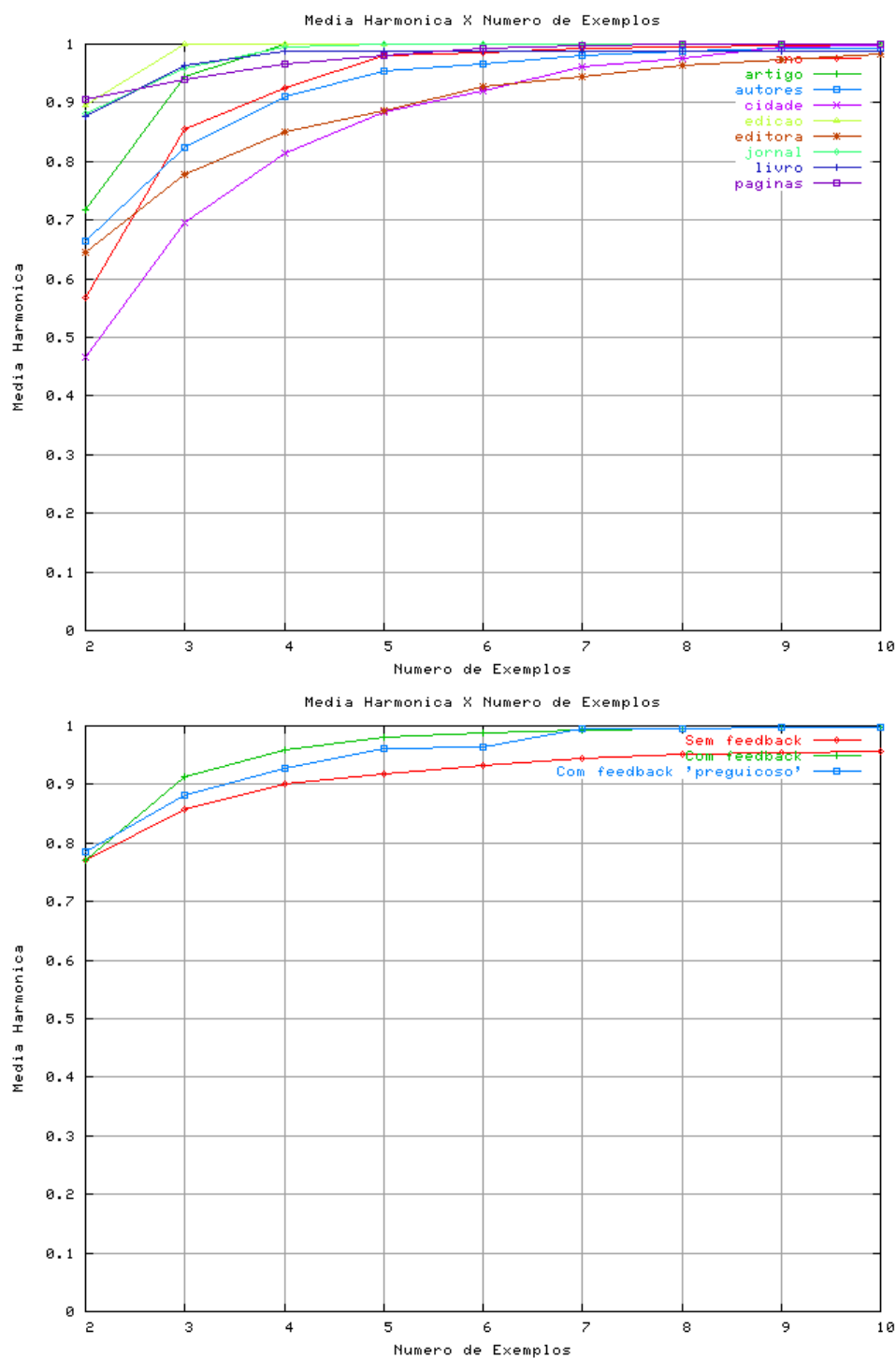


Figura A.2: Resultados obtidos na coleção APSA

CDDDB

Elton John / Greatest Hits

Label: Polydor

Year: 1974

[Music Search](#)[Submit A Change](#)[Digital Top Ten Index](#)[How Does It Work](#)[Products using Gracenote®](#)[Content Partner Program](#)**Track Title****Downloads
Available**

1. Your Song
2. Daniel
3. Honky Cat
4. Goodbye Yellow Brick Road
5. Saturday Night's Alright For Fighting
6. Rocket Man
7. Bennie And The Jets
8. Candle In The Wind

Número de Páginas 15

Número de Objetos 15

Número de Atributos ano=15, cantor=15, faixa=190, label=15, posição=190,
título=15

Descrição Resumida: Base de dados com informações sobre cds de áudio

Estrutura dos Objetos: [cantor, ano, título, rótulo, ([posição, faixa])]

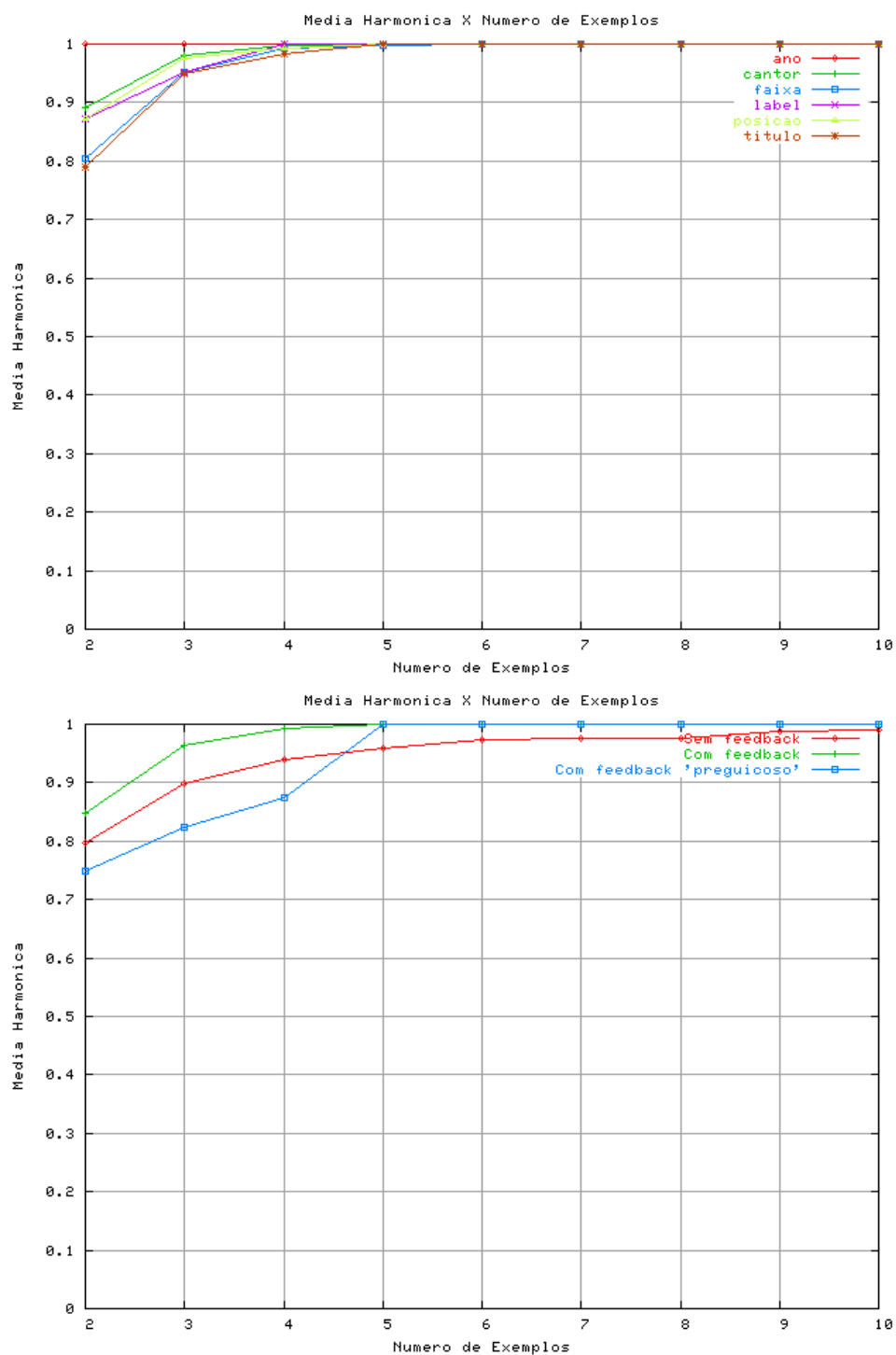


Figura A.3: Resultados obtidos na coleção do CDDB

CDNOW

CDnow Home
[Shipping to Europe or the Middle East?](#)

[Rock/Pop](#)
[Jazz/Blues](#)
[Urban/Electronic](#)
[Classical](#)
[Country/Folk](#)
[World/New Age](#)
[Children](#)

[MTV CD Lounge](#)
[VH1 Music Shop](#)

Main Navigation Menu
 Artist

Album Advisor™ - Discover music based on your taste.

Hundreds of our best-selling albums now 30% off!

Stock up on your favorite music.

	List Price	Add to Cart
Garth Brooks Double Live	\$26.97	\$18.88
Garth Brooks Hits	\$15.97	\$11.18
Garth Brooks Limited Series Box Set	\$44.97	\$31.48
Foxy Brown Chyna Doll	\$16.97	\$11.88

Número de Páginas	8
Número de Objetos	219
Número de Atributos	desconto=219, nome=219, preço=219, tipo=219
Descrição Resumida:	Lista de preços de cds de áudio.
Estrutura dos Objetos:	[nome, tipo, preço, desconto]

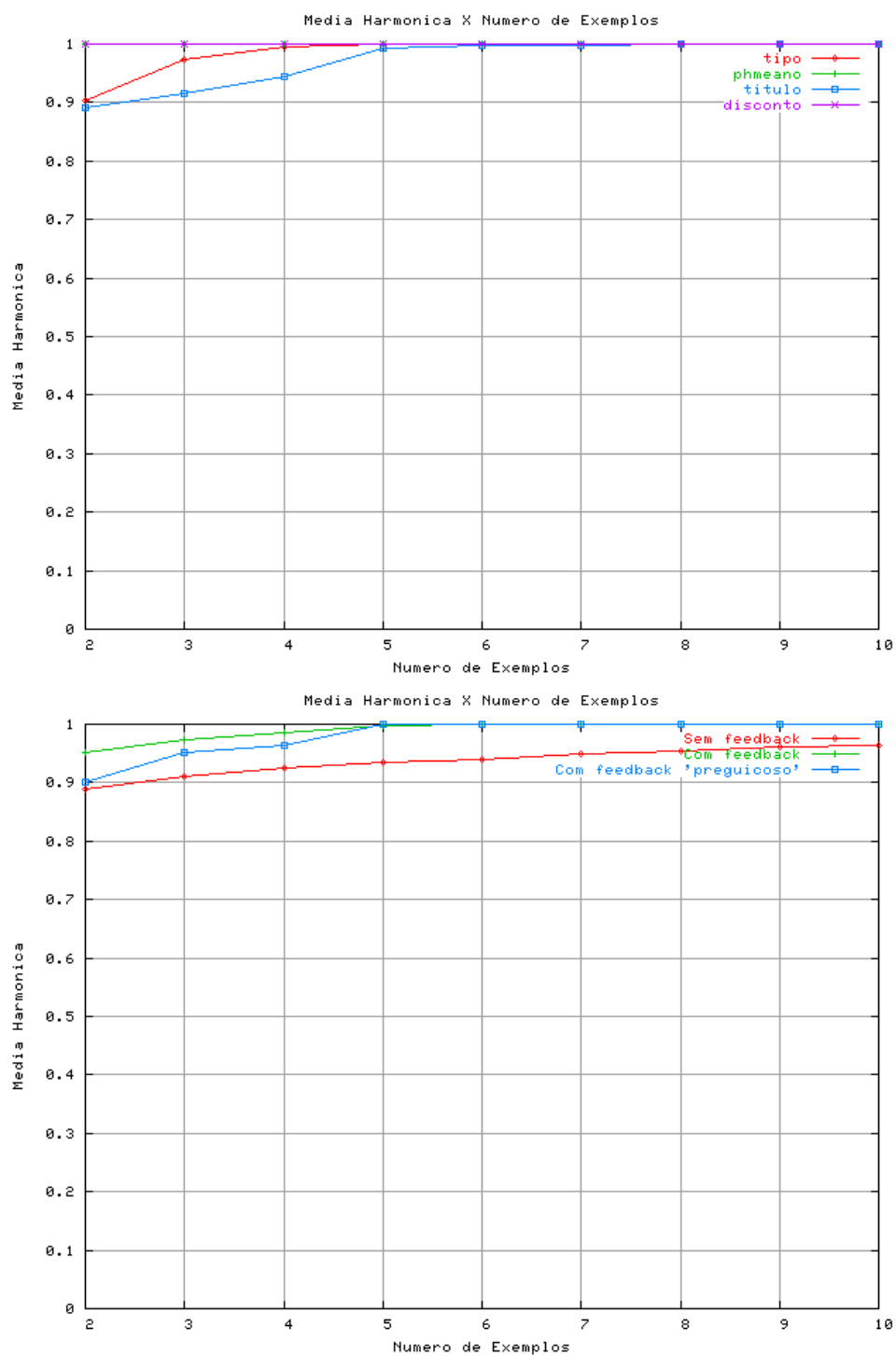


Figura A.4: Resultados obtidos na coleção do CDNOW

IMDB

Top Rated "1920s" Titles

Rank	Rating	Title	Votes
1.	8.5	The Big Parade (1925)	482
2.	8.4	The Wind (1928/I)	599
3.	8.3	Metropolis (1927)	13,595
4.	8.3	The General (1927)	7,463
5.	8.3	Faust (1926)	931
6.	8.3	Nibelungen: Siegfried, Die (1924)	420
7.	8.2	Passion de Jeanne d'Arc, La (1928)	4,128
8.	8.2	The Last Command (1928)	196
9.	8.2	The Gold Rush (1925)	7,252

Número de Páginas 6

Número de Objetos 11

Número de Atributos ano=350, década=11, posição=350, média=350, tipo de ranking=11, filme=350, votos=350

Descrição Resumida: Rankings dos melhores e piores filmes por década

Estrutura dos Objetos: [década, tipo de ranking, ([filme, ano, média, posição, votos])]

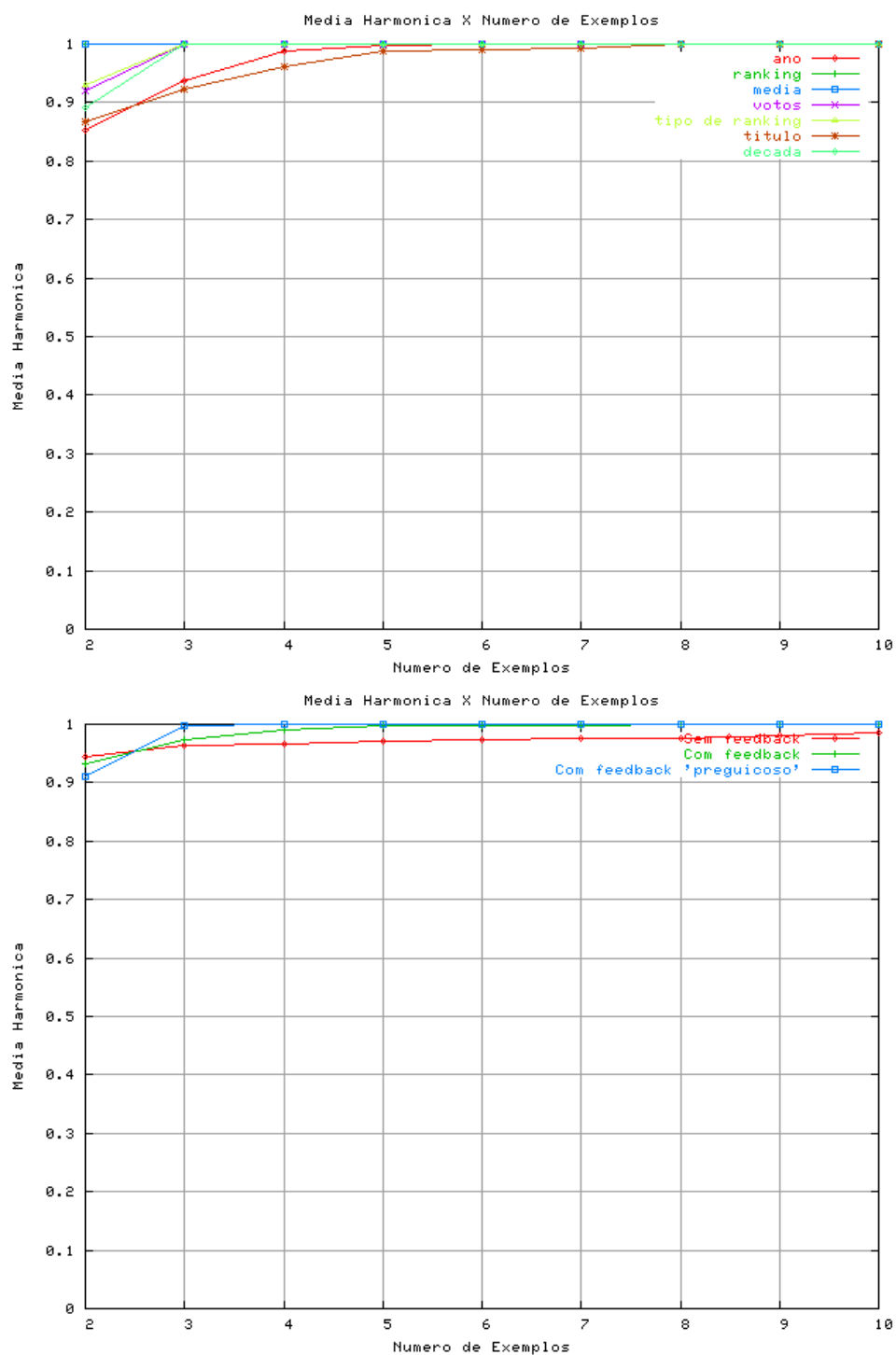


Figura A.5: Resultados obtidos na coleção do IMDB

Loterias

Loteca 					
Concurso 165 Data: 30/05/2005					
jogo	1	time	x	time	2
1	0	FLAMENGO/RJ	x	FLUMINENSE/RJ	0
2	2	PALMEIRAS/SP	x	SANTOS/SP	1
3	1	SÃO PAULO/SP	x	CRUZEIRO/MG	1
4	2	BOTAFOGO/RJ	x	ATLÉTICO/PR	0
5	0	ATLÉTICO/MG	x	CORINTHIANS/SP	1
6	1	CORITIBA/PR	x	JUVENTUDE/RS	2
7	1	BRASILIENSE/DF	x	PARANÁ/PR	0

Número de Páginas 20

Número de Objetos 20

Número de Atributos concurso=20, data=20, estimativa de prêmio=20, gols do primeiro=280, gols do segundo=280, número do jogo=280, prêmio=60, número de acertos=60, ganhadores=60, primeiro time=280, segundo time=280

Descrição Resumida: Páginas com resultados da loteria esportiva

Estrutura dos Objetos: concurso, data, ([número do jogo, gols do primeiro, primeiro time, segundo time, gols do segundo]), ([número de acertos, ganhadores, prêmio]), estimativa de prêmio

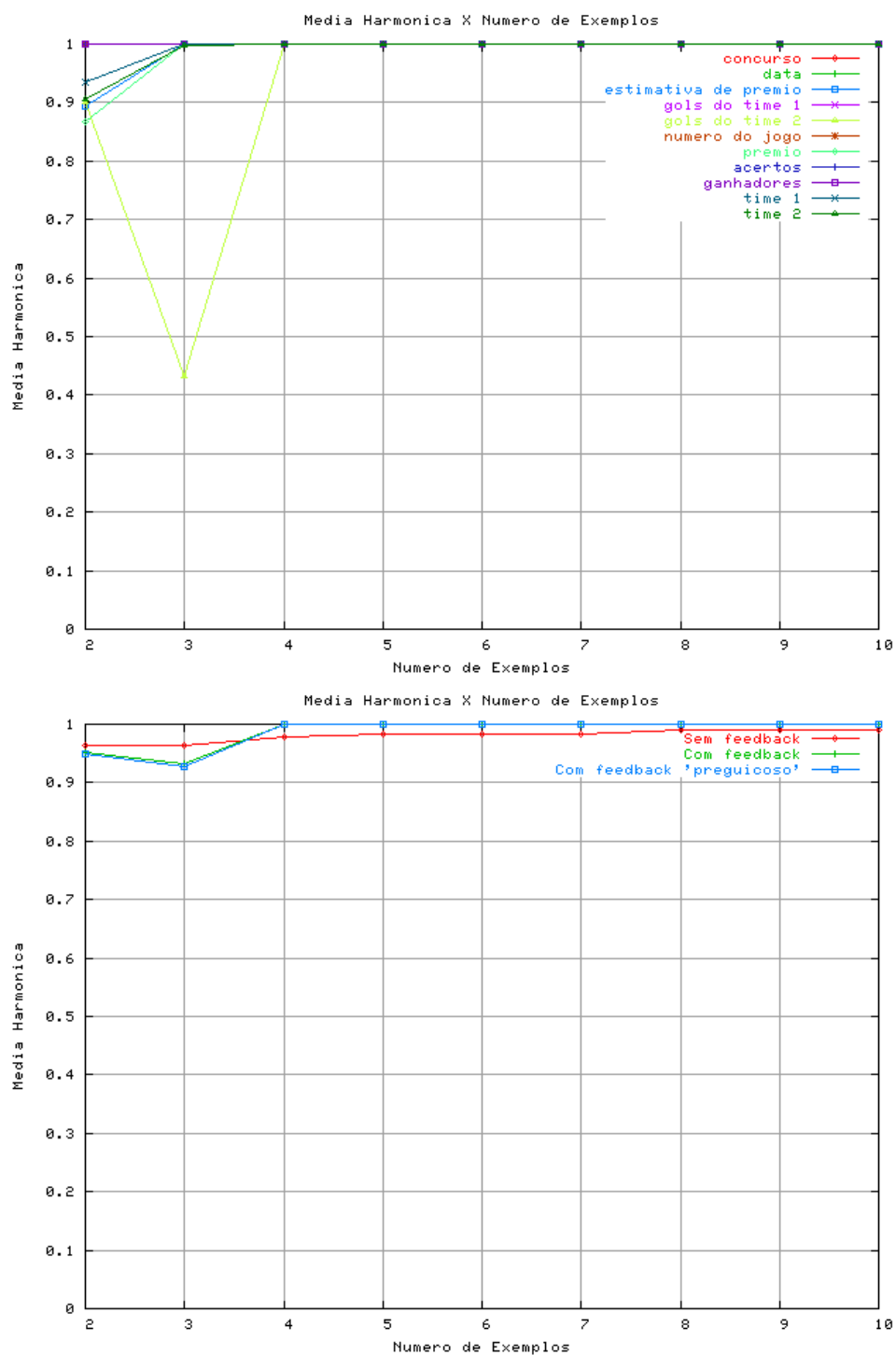


Figura A.6: Resultados obtidos na coleção Loteria

Monster

Monster Technology Jobs 1 to 50 of more than 1,000

Date	Location	Job Title	Company
Jun 8	US-TX-fort worth	Web Developer	StarTech Staffing
Jun 8	US-TN-Nashville	Programmer Analyst	OAO
Jun 8	US-CA-Sacramento	Lotus Notes/Domino Developer	Dynamic Staffing
Jun 7	US-CA-San Francisco	Development Manager	LookSmart
Jun 7	US-VA-Falls Church	Internet Consultant	AppNet, Inc.
Jun 7	US-IL-Chicago	OpenStep Opportunity	Technisource
Jun 7	US-CO-Broomfield	Oracle Database Administrator	Level3 Communications
Jun 7	US-CA-San Francisco	Programmer/Analyst - COBOL	Boeing
Jun 7	US-NY-New York City	SR. NETWORK PLANNER	Forsoft
Jun 7	South Korea	Senior Software Developer	O1 Inc
Jun 7	US-CT-Wallingford	project lead	Maxim Group
Jun 7	US-GA-Atlanta	Network Verification Testers	Butler International
Jun 7	US-IL-Chicago	Internet/Intranet Developer	Hall Kinion
Jun 7	US-CA-Los Angeles	Project Manager	Search West

Número de Páginas 8

Número de Objetos 400

Número de Atributos lugar=394, data=400, emprego=400, país=400

Descrição Resumida: Páginas com anúncio de empregos

Estrutura dos Objetos: [data, lugar, emprego, país]

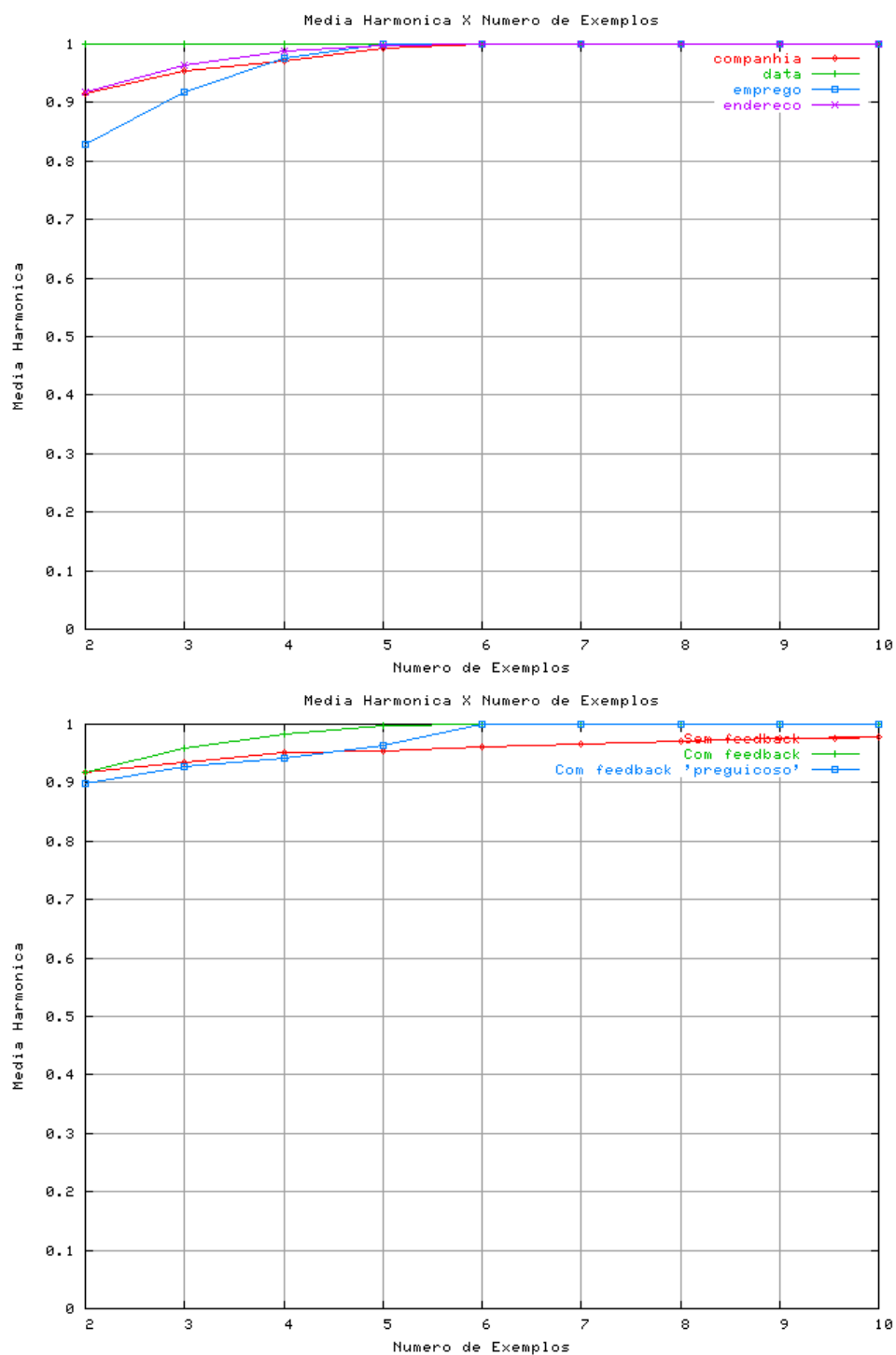


Figura A.7: Resultados obtidos na coleção Monster

PubMed

☐ **1:** [Wang D, Zhang W, Bakhai A.](#) [Related Articles, Links](#)

Abstrac Comparison of Bayesian model averaging and stepwise methods for model selection in logistic regression.

Stat Med. 2004 Oct 25;23(22):3451-3467 [Epub ahead of print]

PMID: 15505893 [PubMed - as supplied by publisher]

☐ **2:** [Shao Q, Zhou X.](#) [Related Articles, Links](#)

Abstrac A new parametric model for survival data with long-term survivors.

Stat Med. 2004 Oct 25;23(22):3525-3543 [Epub ahead of print]

PMID: 15505885 [PubMed - as supplied by publisher]

☐ **3:** [Cook NR, Rosner BA, Chen W, Srinivasan SR, Berenson GS.](#) [Related Articles, Links](#)

Abstrac Using the area under the curve to reduce measurement error in predicting young adult blood pressure from childhood measures.

Stat Med. 2004 Oct 25;23(22):3421-3435 [Epub ahead of print]

PMID: 15505884 [PubMed - as supplied by publisher]

Número de Páginas	3
Número de Objetos	60
Número de Atributos	PID=60, autores=251, data=60, edição=53, jornal=60, páginas=53, título=60
Descrição Resumida:	Páginas com listas de publicações na área médica
Estrutura dos Objetos:	[(autor), título, jornal, data, edição, páginas, PID]

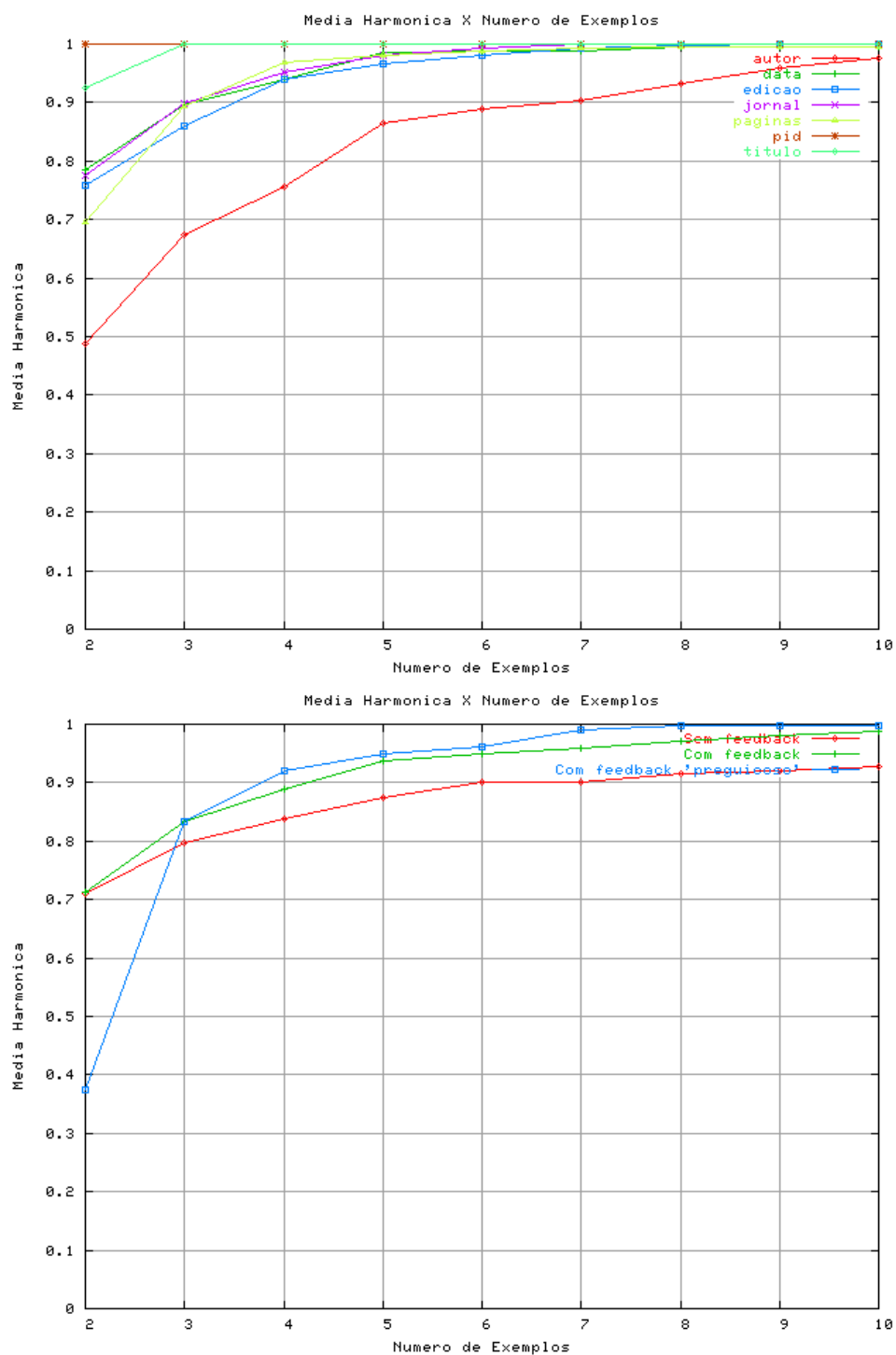


Figura A.8: Resultados obtidos na coleção PubMed

Restaurants

LA Restaurants

Search Criteria: Location: **Any** Cuisine: **Coffeehouse**

GOURMET COFFEE WAREHOUSE

671 Rose Ave., Venice

(310) 392-6479

The Gourmet Coffee Warehouse is a necessity for the caffeine-addicted epicurean and, with wholesale prices, a relief for the pocketbook. The airy space swirls with the spicy aromas of arabica beans culled from fields around the world and fluid-bed roasted on the premises. Gourmet Coffee Warehouse serves not only patrons but the city at large, offering everything from half-pound to 5-pound bags (at those wholesale prices) to home connoisseurs, quality restaurants and offices. Open seven days. No alcohol; delivery. MC, V.

Número de Páginas	27
Número de Objetos	174
Número de Atributos	bairro=164, cartão=559, descrição=174, logradouro=171, número=174, restaurante=174, telefone=174
Descrição Resumida:	Páginas com listas de restaurantes do L.A. Weekly
Estrutura dos Objetos:	[restaurante, número, logradouro bairro, telefone, descricao, (cartão)]

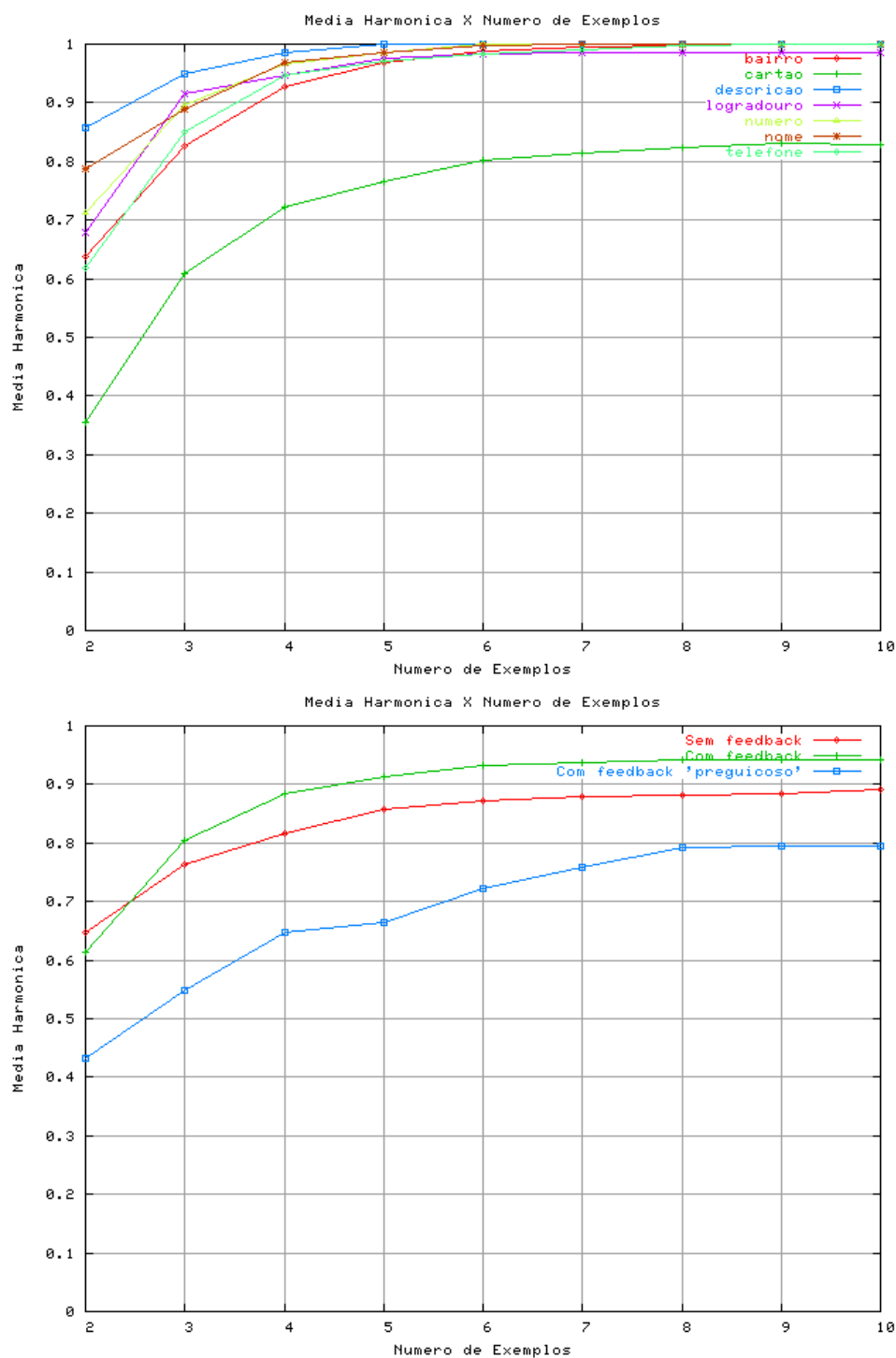


Figura A.9: Resultados obtidos na coleção Restaurants

DNC

Here is the result of your search using the keyword(s) "abortion":

- 1: [w960910.html](#)
Score: 1000, Size: 2 kbytes, Type: HTML file
- 2: [index.html](#)
Score: 904, Size: 3 kbytes, Type: HTML file
- 3: [choice.html](#)
Score: 857, Size: 2 kbytes, Type: HTML file
- 4: [w960104c.html](#)
Score: 714, Size: 3 kbytes, Type: HTML file

Número de Páginas	10
Número de Objetos	327
Número de Atributos	arquivo=327, fonte=327, tamanho=327, tipo=327
Descrição Resumida:	Páginas com resultados de buscas
Estrutura dos Objetos:	[arquivo, fonte, tamanho, tipo]

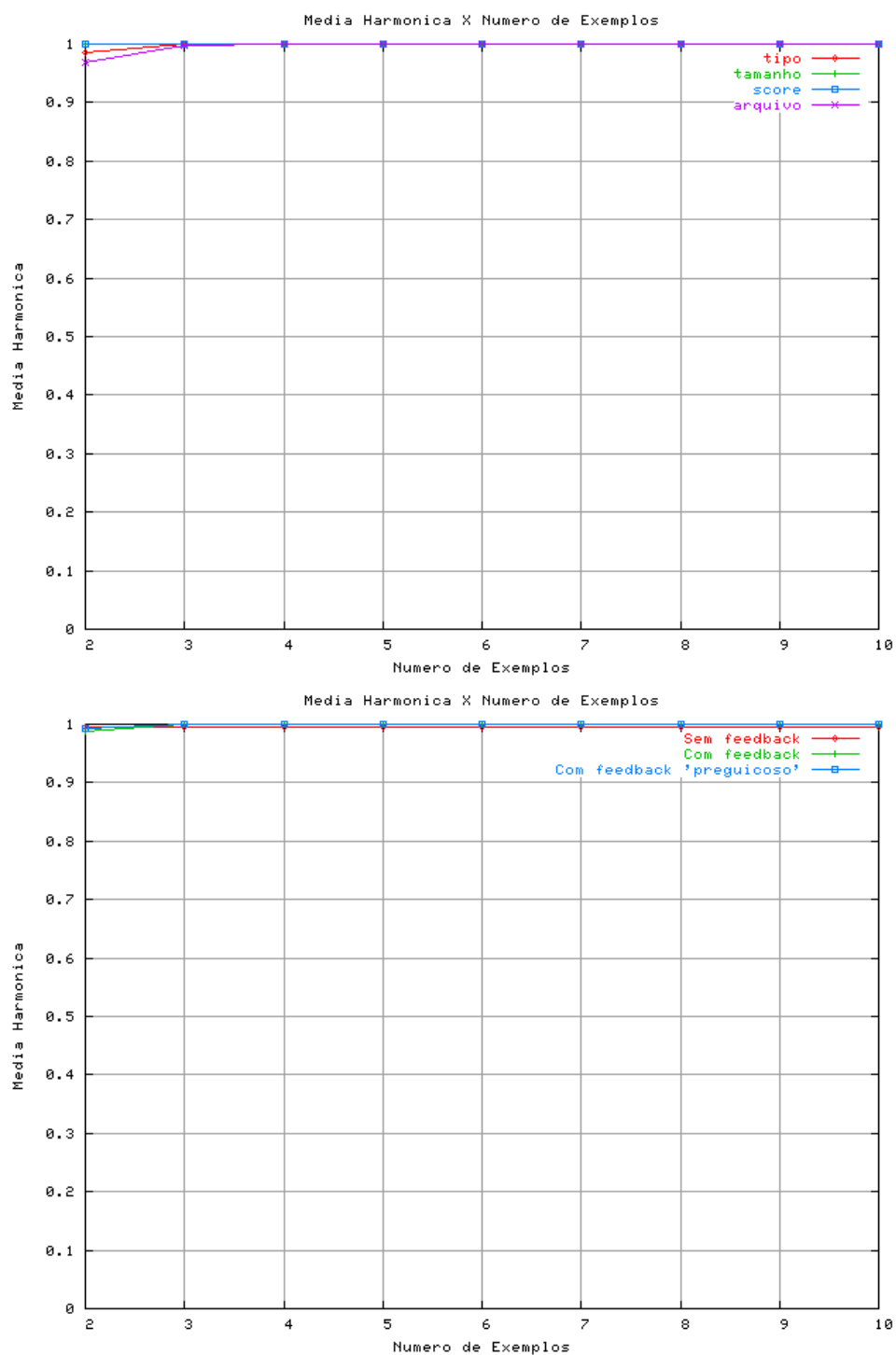


Figura A.10: Resultados obtidos na coleção DNC

Film.com

1. Ingmar Bergman TV THIS MONTH

Abstract: Ingmar Bergman. TVNow TV Schedule Software and Schedules. Film.com TV Appearance Main Page. Film.com Main Page. TVNow Home Page. Channels covered by TVNow as of Dec. 1, 1996.. TV for May. This month's Ingmar Bergman television schedule. This TV schedule was created with the easy to use.

<http://www.film.com/ht/tvnow/archive/arched0601/bergman.htm> - size 6,828 bytes - 4/28/97 5:38:56 PM GMT

2. Ingmar Bergman TV THIS MONTH

Abstract: Ingmar Bergman. TVNow TV Schedule Software and Schedules. Film.com TV Appearance Main Page. Film.com Main Page. TVNow Home Page. Channels covered by TVNow as of May 1, 1997.. TV for June. This month's Ingmar Bergman television schedule. This TV schedule was created with the easy to use.

<http://www.film.com/ht/tvnow/archive/june/bergman.htm> - size 8,599 bytes - 5/28/97 4:16:18 PM GMT

Número de Páginas	10
Número de Objetos	174
Número de Atributos	resumo=174, data=174, tamanho=174, time=174, título=174, url=174
Descrição Resumida:	Páginas com resenhas de filmes
Estrutura dos Objetos:	[título, resumo, url, tamanho, data, hora]

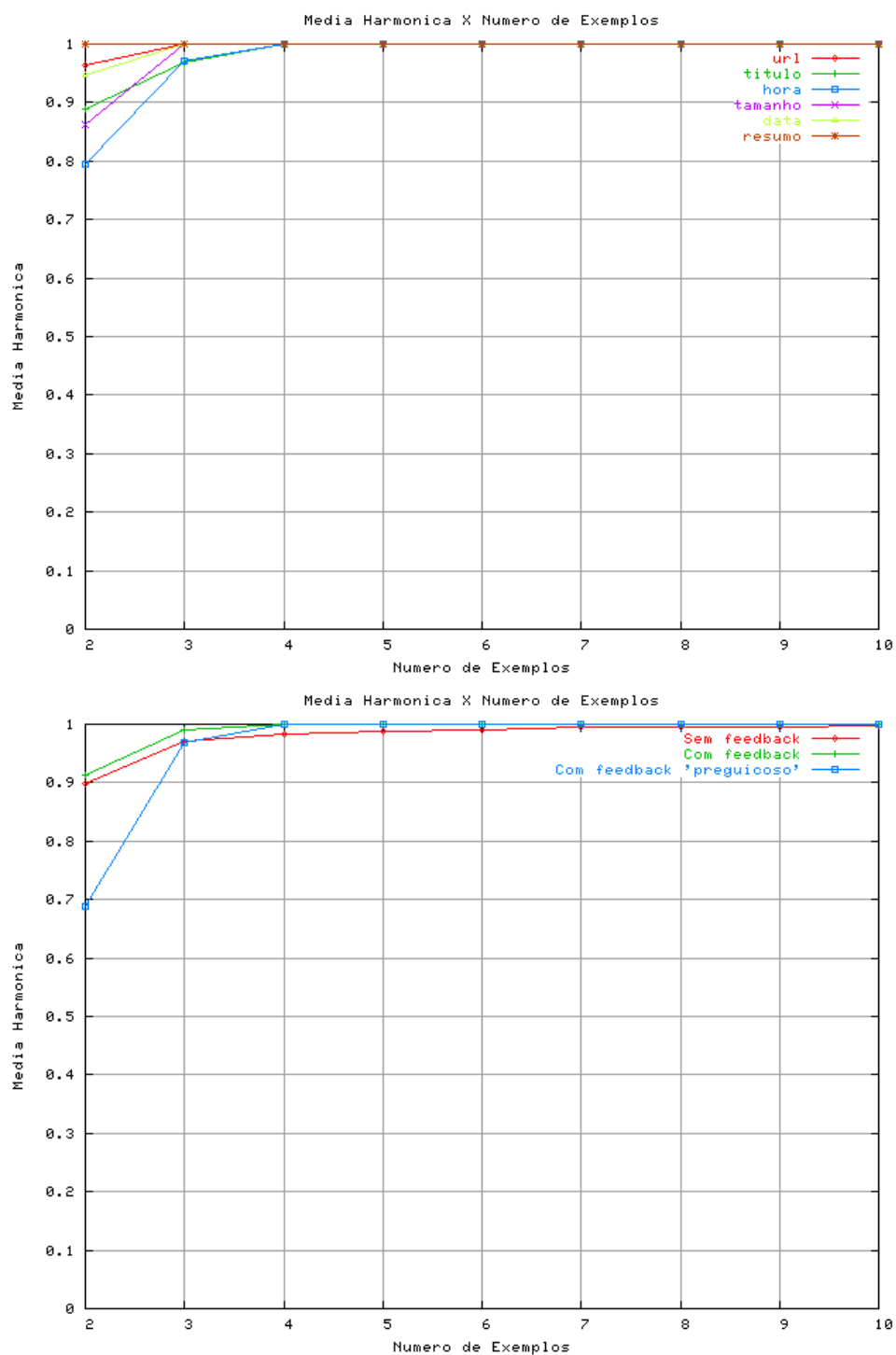


Figura A.11: Resultados obtidos na coleção Film.com

PharmWeb

Faculty	School of Pharmacy
University	University of Washington
Attention	Dr. Sidney D. Nelson, Dean
Address	Room H364 Health Sciences Centre
	Box 357631
City	Seattle
ZIP	WA 98195-7631
Province	Washington
Country	USA

Número de Páginas 10

Número de Objetos 26

Número de Atributos endereço=26, responsável=25, cidade=26, complemento=6, país=26, faculdade=24, fax=26, fone=26, estado=26, universidade=26, update dia=26, update pers=26, update hora=26, url=15, cep=26

Descrição Resumida: Páginas com informações sobre faculdades de farmácia

Estrutura dos Objetos: [faculdade, universidade, responsável, endereço, complemento, cidade, cep, estado, país, fone, fax, url, update dia, update hora, update pers]

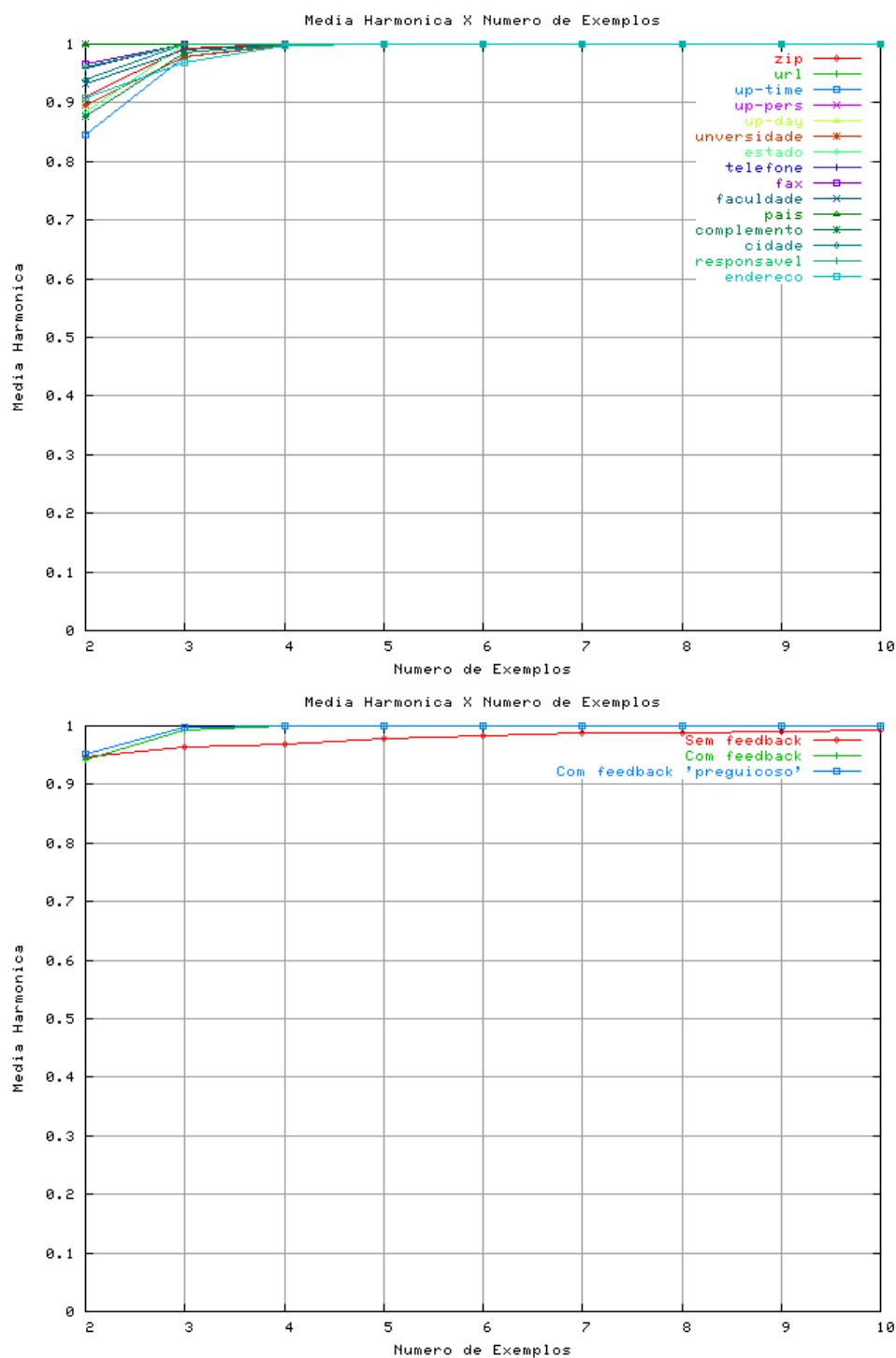


Figura A.12: Resultados obtidos na coleção PharmWeb

News.com

[Join now FREE!](#)**MENU**[Front Door](#)[The Net](#)[Computing](#)[Intranets](#)[Business](#)[CNET Radio](#)[Perspectives](#)[Newsmakers](#)[Rumor Mill](#)

Your search found 285 news items.

Showing items 1 through 25

- * [McAfee expands security business](#), July 30, 1997
- * [Adviser to Paul Allen resigns](#), July 28, 1997
- * [Target One to gather data for Web ads](#), July 28, 1997
- * [PROMO: NEWS.COM's most popular stories](#), July 25, 1997
- * [MS: Windows 98 no blockbuster](#), July 24, 1997
- * [Vendors unfazed by Win 98 delay](#), July 24, 1997
- * [Microsoft looks to year 2000](#), July 24, 1997
- * [Gates, Ballmer tout corporate strategy](#), July 24, 1997
- * [Microsoft jabs Java](#), July 23, 1997
- * [Memphis is Windows 98](#), July 23, 1997
- * [Microsoft plans African expansion](#), July 23, 1997

Número de Páginas 10

Número de Objetos 218

Número de Atributos dia=218, mês=218, título=218, ano=218

Descrição Resumida: Páginas com listas de resultado de busca para notícias

Estrutura dos Objetos: [título, mês, dia, ano]

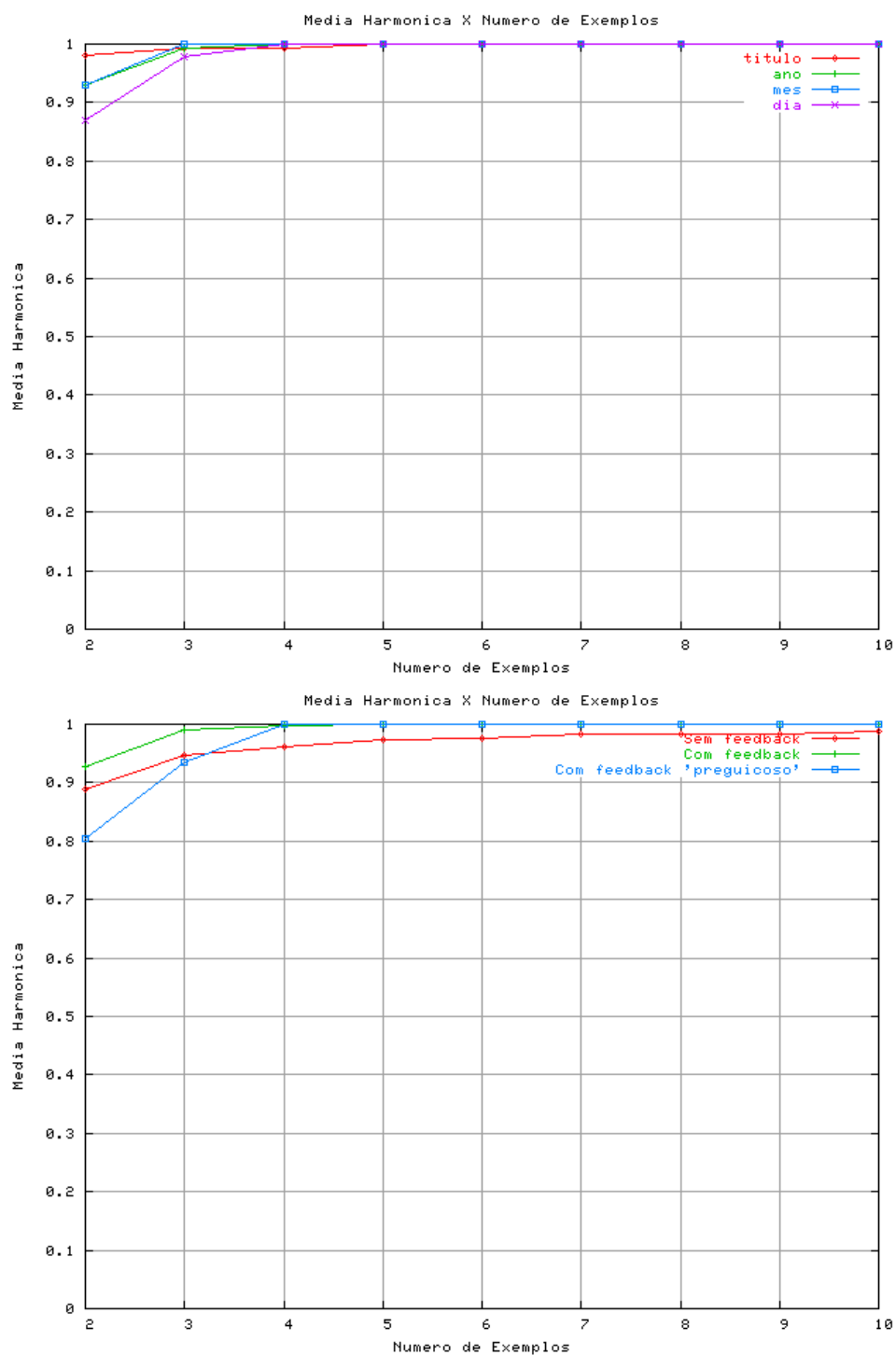


Figura A.13: Resultados obtidos na coleção News.com

Sigmod

Special Section on Scientific Workflows

Guest Editors' Introduction to the Special Section on Scientific Workflows

B. Ludaescher and C. Goble

Available in: [PDF](#)

Integrating Databases and Workflow Systems

S. Shankar, A. Kini, D.J. DeWitt and J. Naughton

Available in: [PDF](#)

An Approach for Pipelining Nested Collections in Scientific Workflows

T.M. McPhillips and S. Bowers

Available in: [PDF](#)

Número de Páginas	10
Número de Objetos	10
Número de Atributos	ano=10, artigo=209, autores=201, mês=10, número=10, sessão=83, volume=10
Descrição Resumida:	Páginas de conferências da SIGMOD
Estrutura dos Objetos:	[mês, ano, número, volume, ([seção, ([autores, artigos]])]]

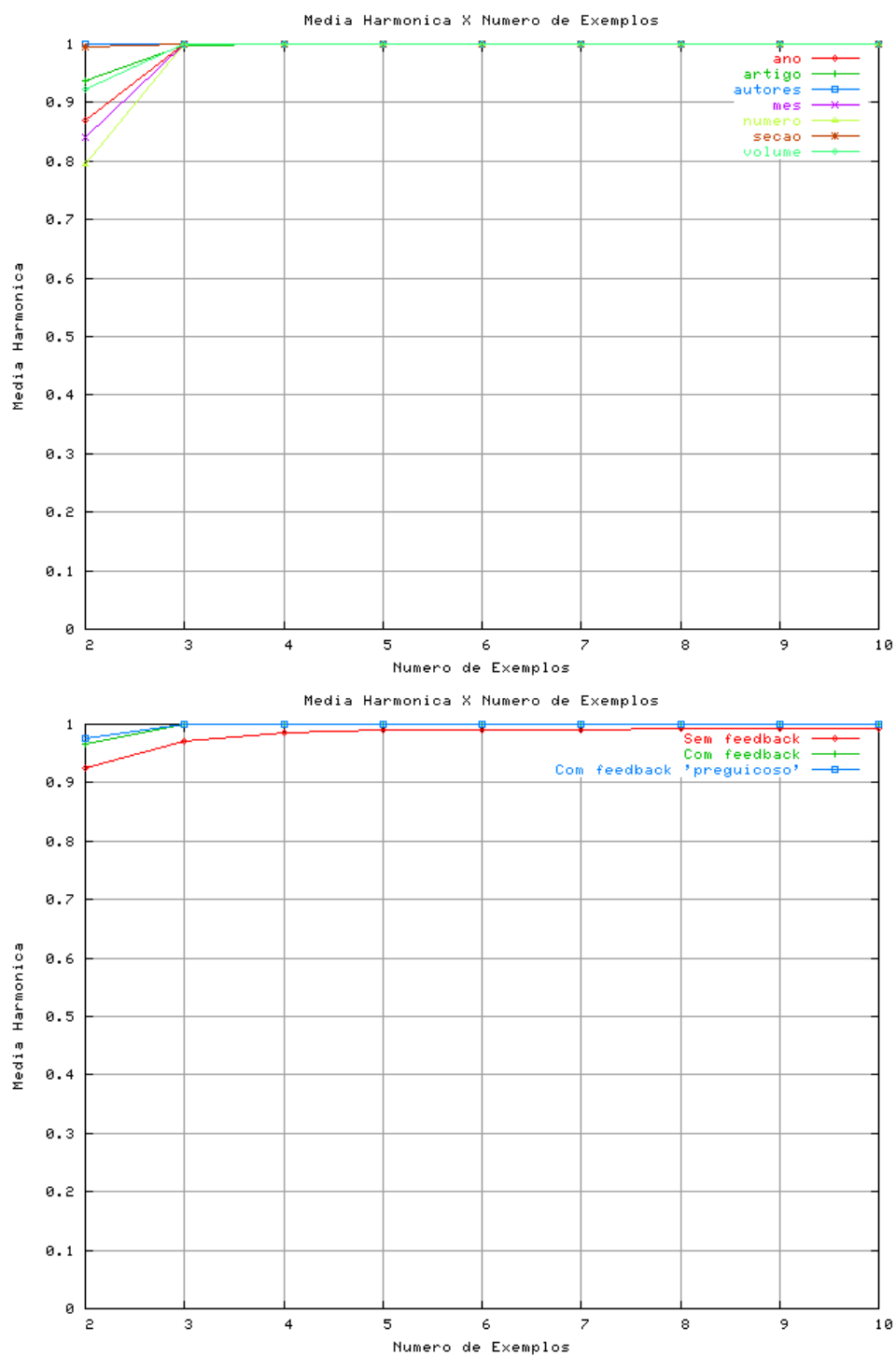


Figura A.14: Resultados obtidos na coleção Sigmod

Sony

**ADRIANA
CALCANHOTTO**
ADRIANA PARTIMPIM /
CD
Ano: 2004
Referência: 82876592092
Selo: ARIOLA

Faixas:

- 01** - lição de baião
- 02** - oito anos
- 03** - lig-lig-lig-lé
- 04** - fico assim sem você 
- 05** - canção da falsa tartaruga (the mock turtle's story)
- 06** - formiga bossa nova
- 07** - ciranda da bailarina
- 08** - ser de sagitário
- 09** - borboleta
- 10** - saiba

Número de Páginas 20

Número de Objetos 20

Número de Atributos álbum=20, ano=20, artista=20, música=298, faixa=298, referência=20, selo=20, tipo=20

Descrição Resumida: Páginas de cds da Sony Music

Estrutura dos Objetos: [ano, álbum, artista, referência, selo, tipo, ([música, faixa])]

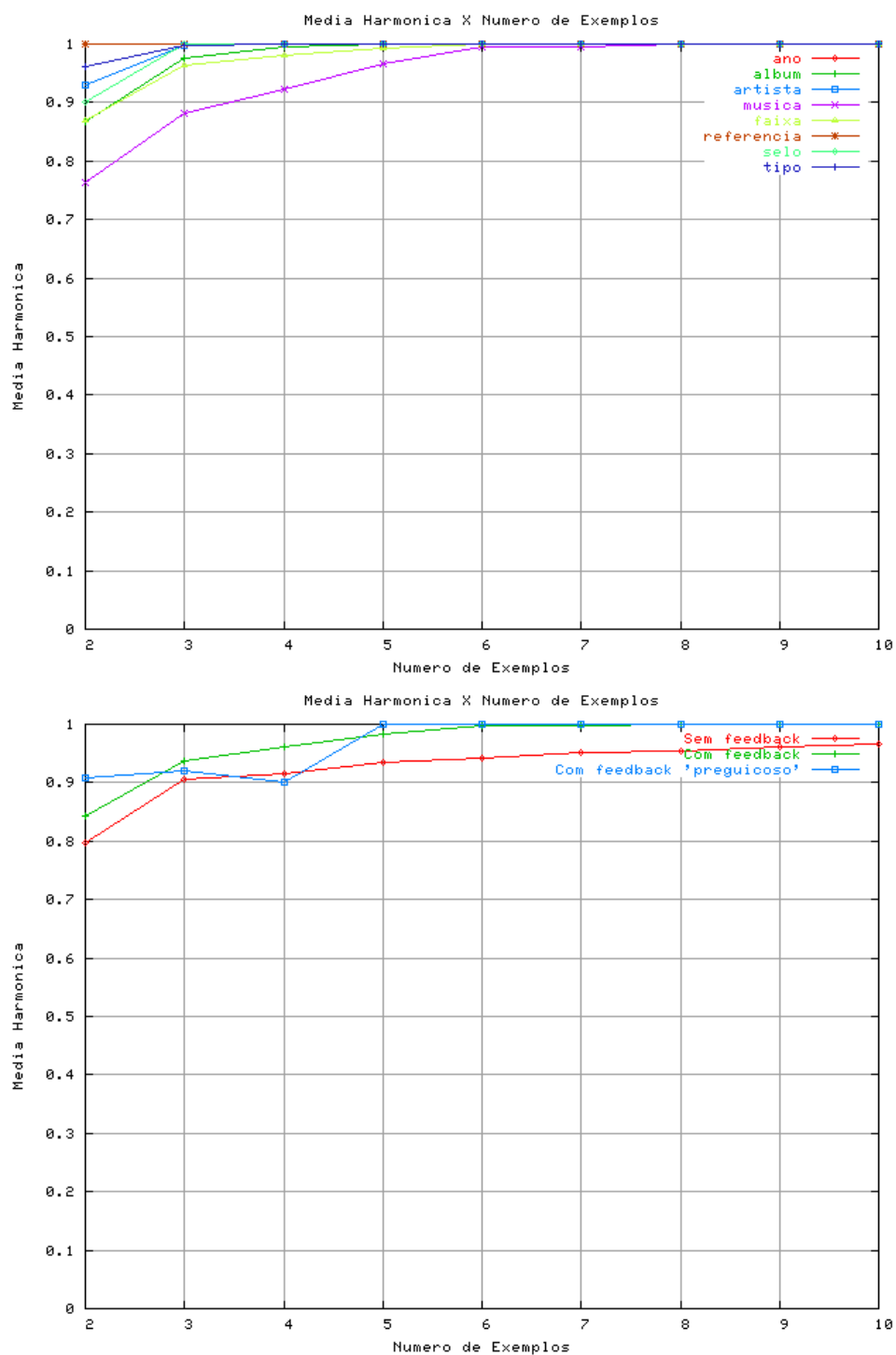


Figura A.15: Resultados obtidos na coleção Sony

VLDB

VLDB 1995

Invited Talks

- ♦ Henry F. Korth:
The Double Life of the Transaction Abstraction: Fundamental Principle and Evolving System Concept. 2-6,
Electronic Edition
- ♦ Rudolf Bayer:
Document Management as a very large Database Application. 7-10,
Electronic Edition
- ♦ Stuart E. Madnick:
From VLDB to VMLDB (Very MANY Large Data Bases): Dealing with Large-Scale Semantic Heterogeneity. 11-16,
Electronic Edition

Número de Páginas 7

Número de Objetos 7

Número de Atributos ano=7, artigo=426, autor=1132, cidade=7, país=7,
seção=81

Descrição Resumida: Páginas do VLDB com todos os trabalhos publicados nas
respectivas seções

Estrutura dos Objetos: [ano, cidade, país, ([seção, ((autor), artigo)])]

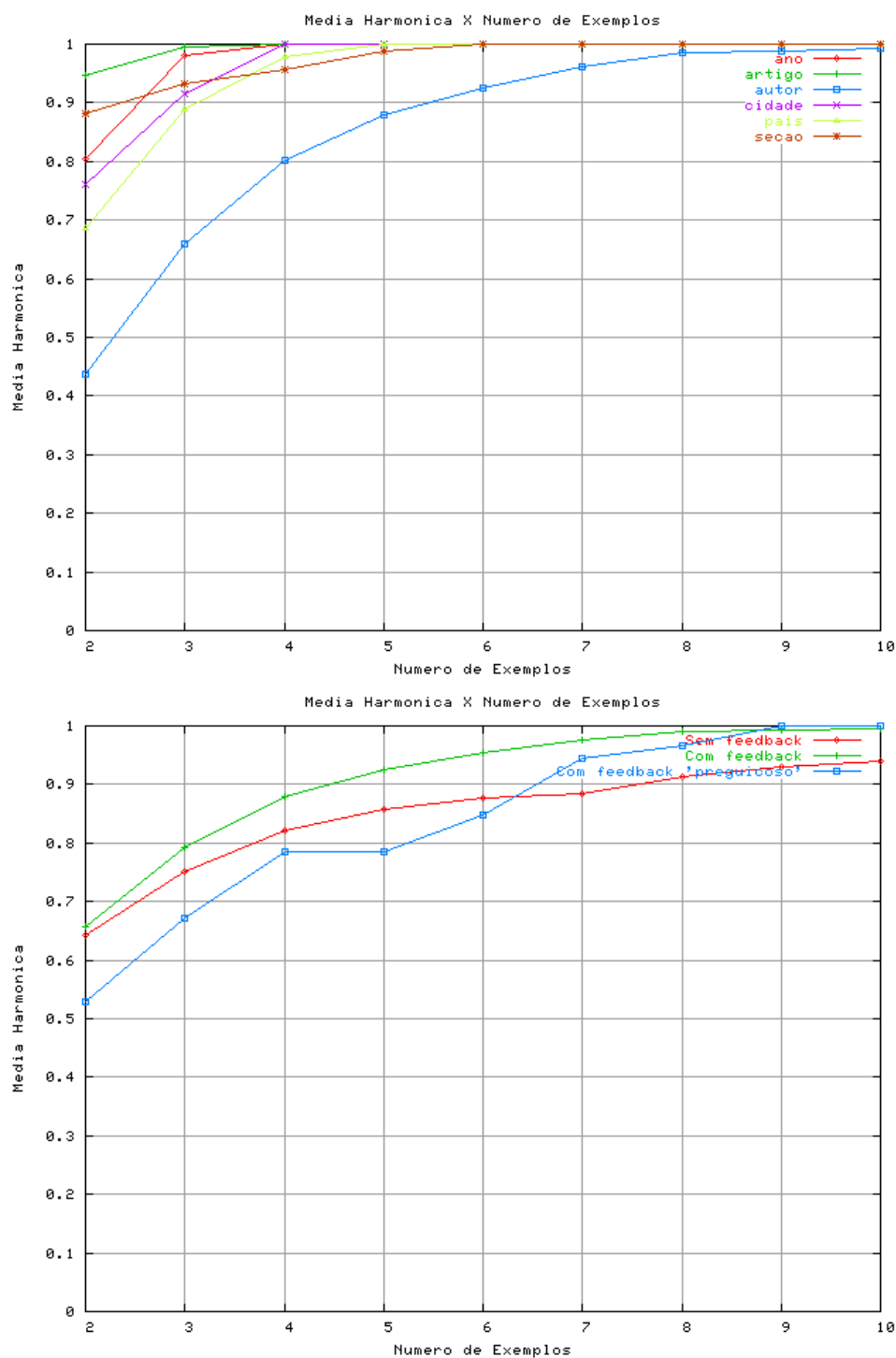


Figura A.16: Resultados obtidos na coleção VLDB

Wine

Page 1 of 5 1 2 3 4 5 |

Barberani

1998 Barberani Orvieto Classico **\$13.00**
Castagnolo, Umbria, Italy

A lovely, dry vineyard-designated Orvieto Classico from one of the region's best producers. Delicious flavors of fig, green melon, pear, and light herb offer a refreshing respite when you've had one too many oaky whites.

[Add to
My Wish
List](#)

1998 Barberani Grechetto, Italy **\$19.95**

Grechetto is the most distinctive and interesting of the five grape varieties used in Orvieto Classico. This wine offers a nice clean, refreshing taste of melon, pear, and anise. Pairs beautifully with fresh trout, calamari, or Ceasar salad.

[Add to
My Wish
List](#)

Número de Páginas 10

Número de Objetos 55

Número de Atributos descrição=88, origem=84, preço=88, produtor=55,
 tipo=88, ano=80

Descrição Resumida: Páginas com listas de vinhos organizados por produtores

Estrutura dos Objetos: [produtor, ([ano, tipo, origem, descrição, preço])]

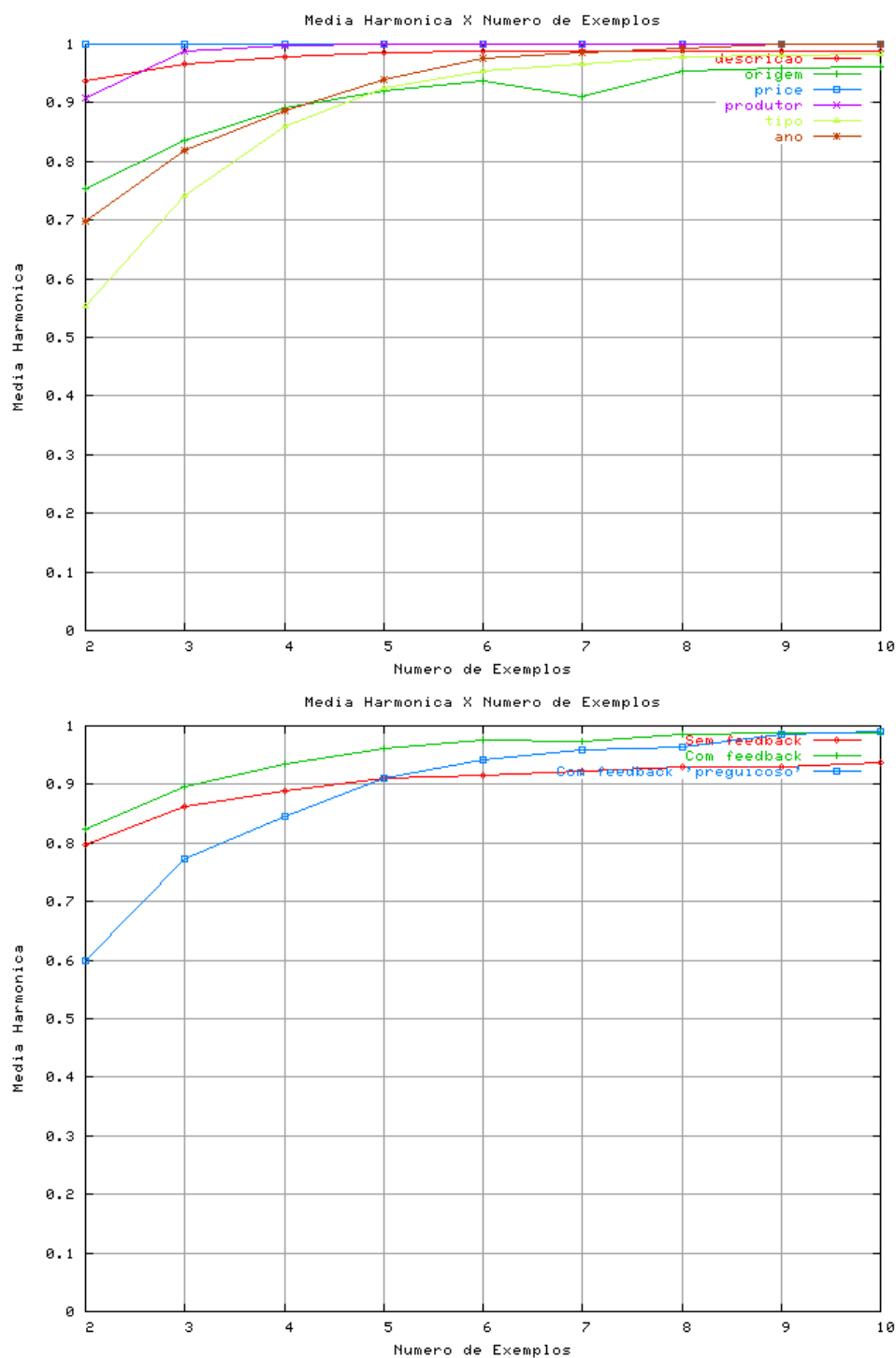


Figura A.17: Resultados obtidos na coleção Wine

Yahoo-Movie**Centerplex Shopping Lapa**

Endereço: Rua Catão, 72
Shoppings Centers
São Paulo, SP
Telefone: (11) 3675-4681

Todos os filmes neste Cinema:

Sala	Lugares	Filme
1	290	Wallace e Gromit - A Batalha dos Vegetais Horários: 13h(a) (dub.)
1	290	Jogos Mortais 2 Horários: 15h(a) / 17h / 19h / 21h10
2	151	A Lenda do Zorro Horários: 13h30(a) / 16h / 18h30 / 21h

Número de Páginas 19

Número de Objetos 19

Número de Atributos cidade=19, complemento=19, estado=19, logradouro=19, telefone=19, horário=436, cinema=19, filme=118, lotação=118, sala=118

Descrição Resumida: Páginas de filmes em cartaz de vários cinemas

Estrutura dos Objetos: [cinema, cidade, complemento, estado, logradouro, telefone, ([sala, lotação, filme, (horario)])]

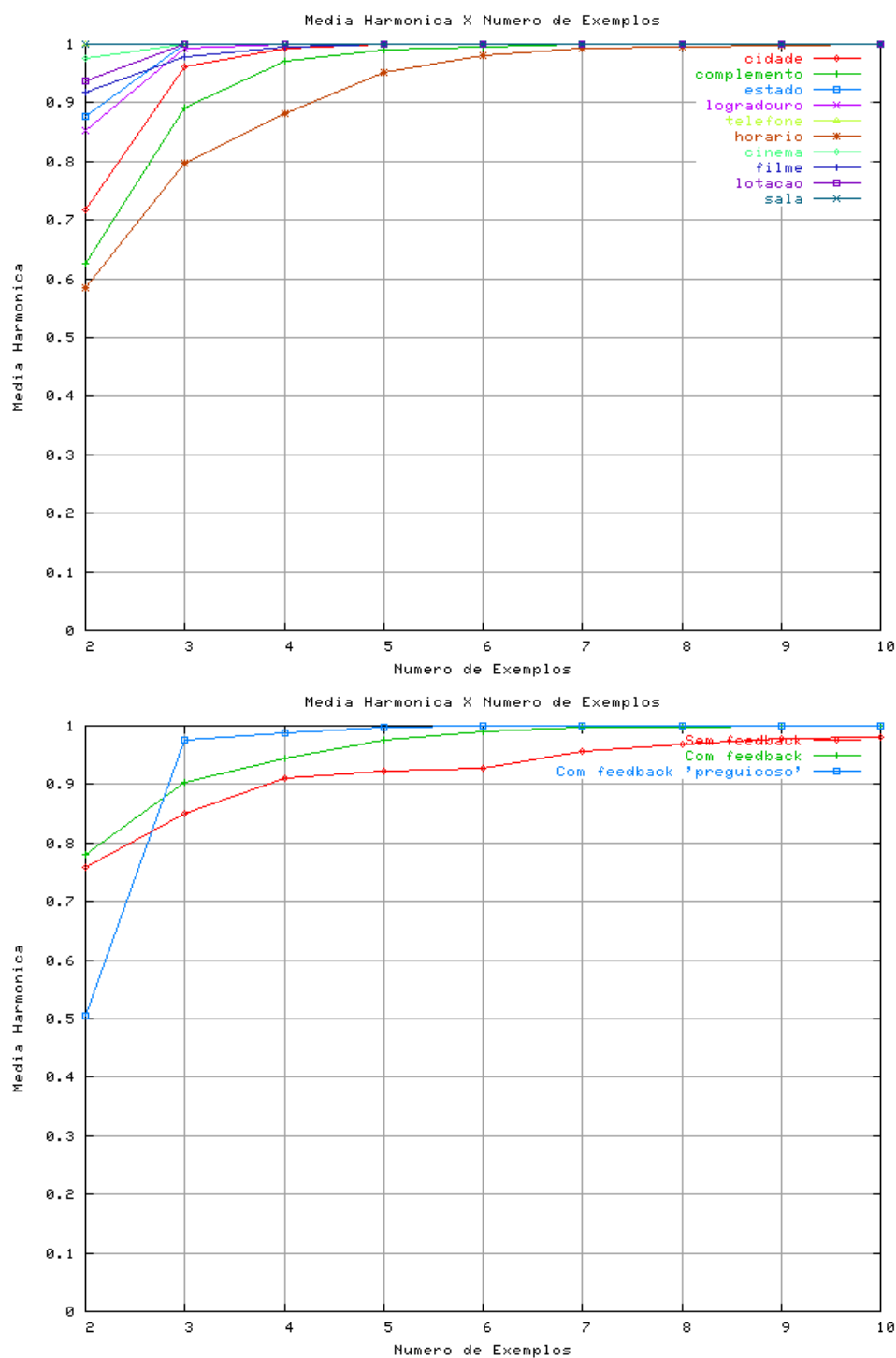


Figura A.18: Resultados obtidos na coleção Yahoo-Movie

Yahoo-Telephone

[Displaying matches 1-10 of 200 | [Next 10](#)]

Name	Address	Phone
K Aab	Bullhead City,AZ 86430	(520)704-0213
Bruce M & K Aaberg	2022 Nw 21st Pl, Portland,OR 97210-2302	(503)222-5794
Krystyn & K Aaberg	1155 O St, Gering,NE 69341-2845	(308)436-2450
Michael & K Aaberg	975 Cnty Rd 327, Silt,CO 81652-9778	(970)876-2167
K Aaboe	San Diego,CA 92138	(619)574-1603
K I Aaby	620 9th Av N, Fargo,ND 58102-3626	(701)298-8840

Número de Páginas	10
Número de Objetos	100
Número de Atributos	endereço=100, nome=100, telefone=100
Descrição Resumida:	Páginas de lista telefonica
Estrutura dos Objetos:	[nome, endereço, telefone]

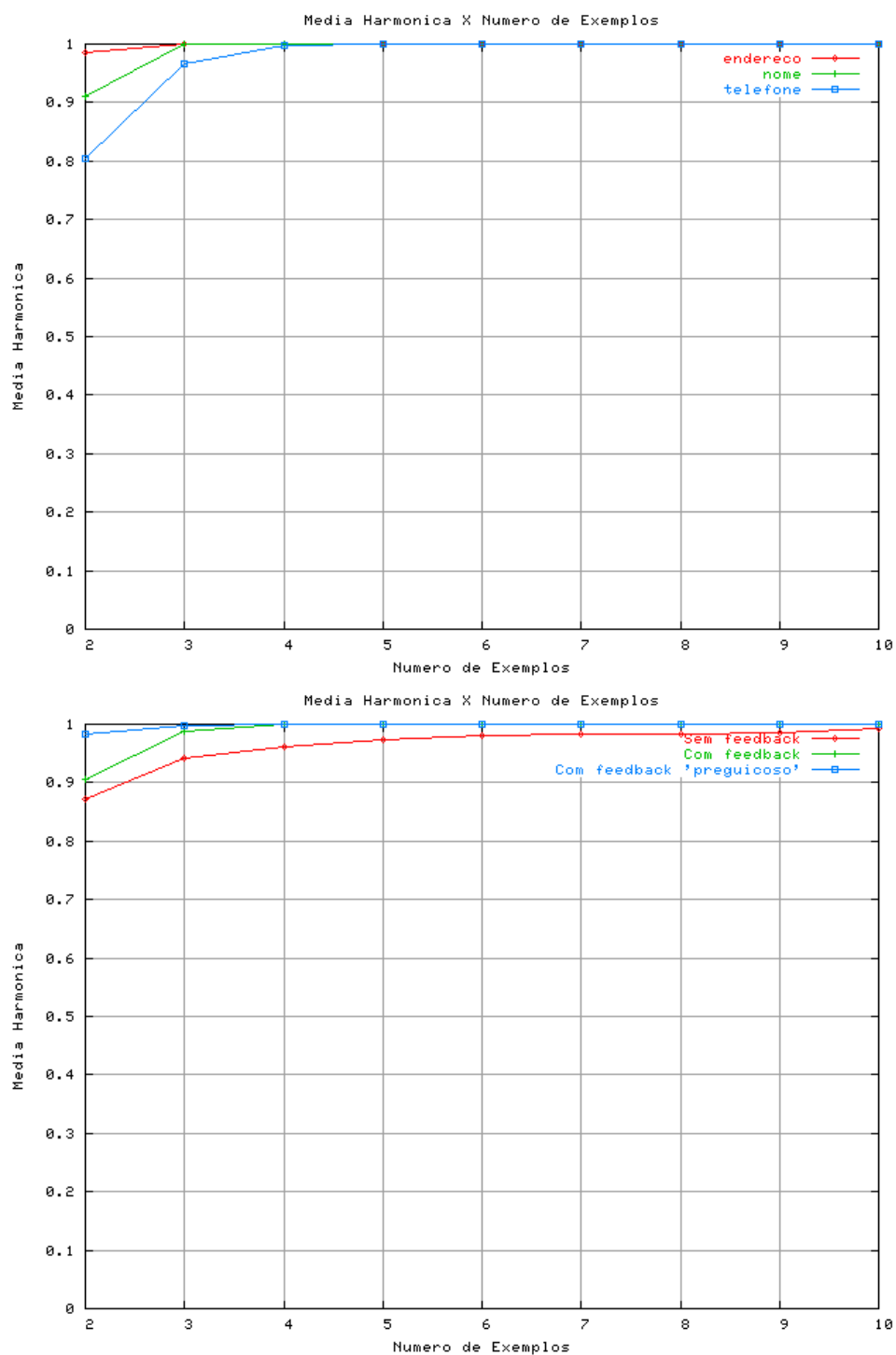


Figura A.19: Resultados obtidos na coleção Yahoo-Telephone

Zagat**Koo Koo Roo**

Chicken

8393 W. Beverly Blvd., LA 213-655-9045

17136 Ventura Blvd., Encino 818-906-2649

11066 Santa Monica Blvd., West LA 310-444-9852

262 S. Beverly Dr., Beverly Hills 310-274-3121

11650 San Vicente Blvd., Brentwood 310-207-3232

3225 Glencoe Ave., Villa Marina Shopping Ctr., Marina del Rey

310-305-8100

What began as one shop selling skinless flame-broiled chicken has expanded into a full-service poultry and salad chain offering "delicious" birds, "fab turkey sandwiches", "great side dishes" and more stuff you'll "want every day"; admirers say you "can't beat" it for a "healthy", fast-food "bargain."

[Back to Alphabetical List](#)

Número de Páginas	91
Número de Objetos	91
Número de Atributos	complemento=17, cozinha=91, descrição=91, logradouro=151, nome=91, número=150, fone=148, estado=151
Descrição Resumida:	Páginas com listas de restaurantes
Estrutura dos Objetos:	[nome, cozinha, ([número, logradouro, complemento, estado, fone]), descrição]

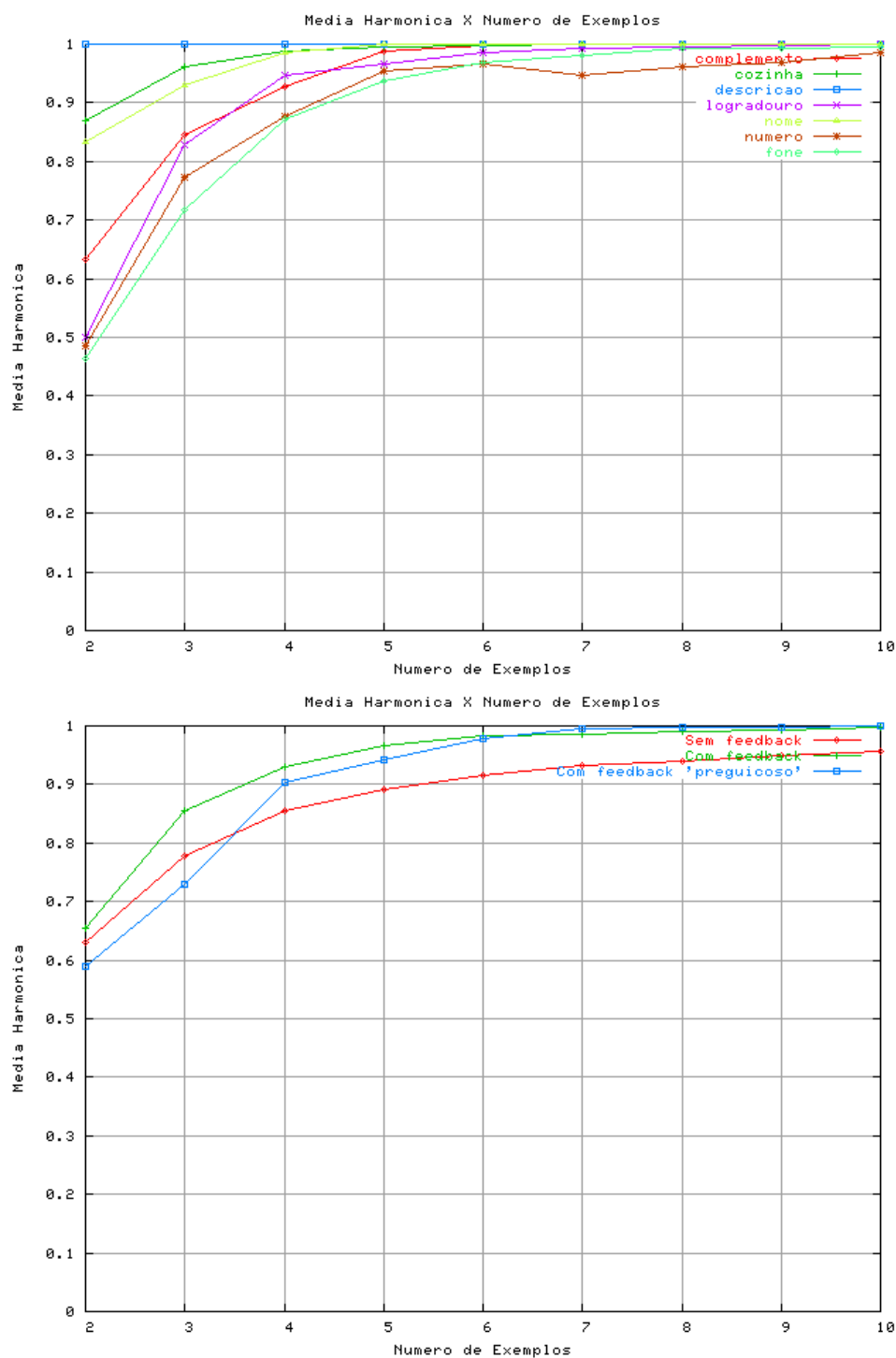


Figura A.20: Resultados obtidos na coleção Zagat

Referências Bibliográficas

- [Adelberg, 1998] Adelberg, B. (1998). NoDoSE: A tool for semi-automatically extracting structured and semistructured data from text documents. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 283–294.
- [Arasu and Garcia-Molina, 2003] Arasu, A. and Garcia-Molina, H. (2003). Extracting structured data from web pages. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 337–348.
- [Baeza-Yates and Ribeiro-Neto, 1999] Baeza-Yates, R. A. and Ribeiro-Neto, B. A. (1999). *Modern Information Retrieval*. ACM Press / Addison-Wesley.
- [Baumgartner et al., 2001] Baumgartner, R., Flesca, S., and Gottlob, G. (2001). Visual web information extraction with lixto. In *Proceedings of the 27th International Conference on Very Large Data Bases*, pages 119–128.
- [Calado et al., 2002] Calado, P. P., da Silva, A. S., Ribeiro-Neto, B., Laender, A. H. F., Lage, J. P., Reis, D. C., Roberto, P. A., Vieira, M. V., Gonçalves, M. A., and Fox, E. A. (2002). Web-dl: an experience in building digital libraries from the web. In *Proceedings of the 11th international conference on Information and knowledge management*, pages 675–677.
- [Crescenzi et al., 2001] Crescenzi, V., Mecca, G., and Merialdo, P. (2001). Roadrunner: Towards automatic data extraction from large web sites. In *Proceedings of 27th International Conference on Very Large Data Bases*, pages 109–118.
- [da Silva, 2002] da Silva, A. S. (2002). *Web Data Extraction Based on User Examples*. PhD thesis, Department of Computer Science, Federal University of Minas Gerais.

- [da Silva et al., 2002] da Silva, A. S., Filha, I. M. R. E., Laender, A. H. F., and Embley, D. W. (2002). Representing and querying semistructured web data using nested tables with structural variants. In *International Conference on the Entity Relationship Approach*, pages 135–151.
- [de Castro Reis et al., 2002] de Castro Reis, D., Araujo, R. B., da Silva, A. S., and Ribeiro-Neto, B. A. (2002). A framework for generating attribute extractors for web data sources. In *Proceedings of the 9th International Symposium on String Processing and Information Retrieval*, pages 210–226.
- [Durbin et al., 1998] Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological sequence analysis*. Cambridge University Press.
- [Embley et al., 1999] Embley, D. W., Campbell, D. M., Jiang, Y. S., Liddle, S. W., Kai Ng, Y., Quass, D., and Smith, R. D. (1999). Conceptual-model-based data extraction from multiple-record web pages. *Data and Knowledge Engineering*, 31(3):227–251.
- [Feng and Doolittle, 1987] Feng, D.-F. and Doolittle, R. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, 25:351–360.
- [Freitag and Kushmerick, 2000] Freitag, D. and Kushmerick, N. (2000). Boosted wrapper induction. In *American Association for Artificial Intelligence Conference*, pages 577–583.
- [Gusfield, 1997] Gusfield, D. (1997). *Algorithms on Strings, Trees and Sequences*. Cambridge University Press.
- [Hsu and Dung, 1998] Hsu, C.-N. and Dung, M.-T. (1998). Generating finite-state transducers for semi-structured data extraction from the web. *Information Systems*, 23(8):521–538.
- [Kushmerick, 2000] Kushmerick, N. (2000). Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence Journal*, 118(1-2):15–68.
- [Laender et al., 2002a] Laender, A., Ribeiro-Neto, B., Silva, A., and Teixeira, J. (2002a). A brief survey of web data extraction tools. *SIGMOD Record*, 31(2):84–93.
- [Laender et al., 2002b] Laender, A. H. F., Ribeiro-Neto, B. A., and da Silva, A. S. (2002b). DEByE - data extraction by example. *Data and Knowledge Engineering*, 40(2):121–154.

- [Leiserson et al., 2001] Leiserson, C. E., Stein, C., Rivest, R. L., and Cormen, T. H. (2001). *Introduction to algorithms*. MIT Press.
- [Light, 1997] Light, R. (1997). *Presenting XML*. SAMS.
- [Liu et al., 2000] Liu, L., Pu, C., and Han, W. (2000). XWRAP: An XML-enabled wrapper construction system for web information sources. In *International Conference on Data Engineering*, pages 611–621.
- [Muslea et al., 2001] Muslea, I., Minton, S., and Knoblock, C. (2001). Hierarchical wrapper induction for semistructured information sources. *Autonomous Agents and Multi-Agent Systems*, 4(1/2):93–114.
- [Reis et al., 2004] Reis, D. D. C., Golgher, P. B., Laender, A. H., and da Silva, A. S. (2004). Automatic web news extraction using tree edit distance. In *Proceedings of the 13th International Conference on World Wide Web*, pages 502–511.
- [Sahuguet and Azavant, 2001] Sahuguet, A. and Azavant, F. (2001). Building intelligent web applications using lightweight wrappers. *Data and Knowledge Engineering*, 36(3):283–316.
- [Saiton and Nei, 1987] Saiton, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology Evolution*, 4:406–425.
- [Smith and Waterman, 1981] Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197.
- [Soderland, 1999] Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3):233–272.