UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Extração automática de dados de páginas HTML utilizando alinhamento em dois níveis

André de Souza PEDRALHO

Manaus - Amazonas
Julho de 2011

André de Souza PEDRALHO

Extração automática de dados de páginas HTML utilizando alinhamento em dois níveis

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 de Informação.*

Orientador: Dr. Altigran Soares DA SILVA - UFAM/PPGI

André de Souza PEDRALHO

Extração automática de dados de páginas HTML utilizando alinhamento em dois níveis

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 de Informação.*

Banca Examinadora

Dr. Altigran Soares DA SILVA
Departamento de Ciência da Computação - UFAM/PPGI

D.Sc. João Marcos Bastos CAVALCANTI
Departamento de Ciência da Computação - UFAM/PPGI

Ph.D. Mirella M. MORO
Departamento de Ciência da Computação - UFMG

Manaus - Amazonas
Julho de 2011

Ficha Catalográfica

CATALOGAÇÃO REALIZADA PELA BIBLIOTECA CENTRAL DA UFAM

Pedralho, André de Souza
P371e Extração automática de dados de páginas HTML utilizando alinhamento em dois níveis / André de Souza Pedralho. - Manaus: UFAM, 2011.
62 f.: il. color.

Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas. 2011.
Orientador: Prof. Dr. Altigran Soares da Silva.

1. Recuperação da Informação 2. Sites da Web 3. Sistemas de recuperação da informação I. Silva, Altigran Soares da (Orient.) II. Universidade Federal do Amazonas III. Título

CDU 004.78(043.3)

André de Souza PEDRALHO

A conclusão deste trabalho não seria possível sem a colaboração, incentivo e apoio de algumas pessoas muito importantes, às quais dedico estes resultados.

Agradeço a meus pais pelo incentivo, suporte e cobranças durante toda minha vida estudantil. Sem eles, não teria conseguido nem mesmo ingressar em um programa de pós-graduação.

Agradeço a Gisele por estar ao meu lado em todos os momentos, compreendendo minhas necessidades e me apoiando em todas as situações. Sem ela, não teria conseguido terminar este trabalho.

Agradeço aos meus amigos por todo o incentivo durante estes anos de estudo e trabalho.

E agradeço aos colegas de trabalho, por compreenderem minha situação de estudante de pós-graduação.

Agradecimentos

Resumo

Existe uma grande quantidade de informação na World Wide Web em páginas compostas por objetos similares. Web sites de comércio eletrônico e catálogos online, em geral, são exemplos destes repositórios de dados. Apesar destes dados serem apresentados em porções de texto semi-estruturados, são projetados para serem interpretados e utilizados por humanos e não processados por máquinas. A identificação destes objetos em páginas Web é feita por aplicações externas chamadas extratores ou *wrappers*.

Neste trabalho propomos e avaliamos um método automático para o problema de extrair e estruturar registros e valores de seus atributos presentes em páginas Web ricas em dados. O método utiliza um *Algoritmo de Alinhamento de Árvores* para encontrar nestas páginas exemplos de registros que correspondem a objetos de interesse. Em seguida, o método gera expressões regulares para extrair objetos similares aos exemplos dados usando o *Algoritmo de Alinhamento de Múltiplas Sequências*. Em um passo final, o método decompõe os registros em sequências de texto aplicando a expressão regular criada e formatações e delimitadores comuns, com o intuito de identificar os valores dos atributos dos registros. Experimentos utilizando uma coleção composta por 128 páginas Web de diferentes domínios demonstram a viabilidade do nosso método de extração. O método foi avaliado em relação à identificação de blocos de código HTML que contêm os registros e quanto à extração dos registros e dos valores de seus atributos. Obtivemos precisão de 83% e revocação de 80% na extração de valores de atributos. Estes valores significam um ganho na precisão de 43,37% e na revocação de 68,75%, em relação a propostas similares.

PALAVRAS-CHAVE: extração de dados Web, alinhamento em dois níveis, distância de edição de árvores, geração automática de extratores.

Abstract

There is a huge amount of information in the World Wide Web in pages composed by similar objects. E-commerce Web sites and on-line catalogs, in general, are examples of such data repositories. Although this information usually occurs in semi-structured texts, it is designed to be interpreted and used by humans and not processed by machines. The identification of these objects in Web pages is performed by external applications called extractors or *wrappers*.

In this work we propose and evaluate an automatic approach to the problem of generating wrappers capable of extracting and structuring data records and the values of their attributes. It uses the *Tree Alignment Algorithm* to find in the Web page examples of objects of interest. Then, our method generates regular expressions for extracting objects similar to the examples given using the *Multiple Sequence Alignment Algorithm*. In a final step, the method decomposes the objects in sequences of text using the regular expression and common formats and delimiters, in order to identify the value of the attributes of the data records. Experiments using a collection composed by 128 Web pages from different domains have demonstrated the feasibility of our extraction method. It is evaluated regarding the identification of blocks of HTML source code that contain data records and regarding record extraction and the value of its attributes. It reached a precision of 83% and a recall of 80% when extracting the value of attributes. These values mean a gain in precision of 43.37% and in recall of 68.75% when compared to similar proposals.

KEYWORDS: Web Data extraction, two-level alignment, tree edit distance, automatic Wrapper generation.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Capítulo 1

Introdução

Existe na *World Wide Web* uma grande quantidade de informação semi-estruturada disponível nas chamadas páginas ricas em dados. Estas páginas são geradas a partir de resultados de consultas em banco de dados ou máquinas de busca e inseridos em estruturas HTML pré-definidas com o intuito de serem interpretadas por humanos. Web sites de comércio eletrônico, bibliotecas digitais e máquinas de busca são exemplos de aplicações que geram páginas ricas em dados. Estas páginas disponibilizam registros contendo dados sobre objetos tais como produtos, anúncios, personalidades, filmes, páginas Web, etc. O objetivo deste trabalho é desenvolver um método automático para identificar, extrair, estruturar estes dados sem intervenção humana. Os dados resultantes deste processo podem ser utilizados para permitir a execução de consultas estruturadas, mineração de dados, disseminação, etc.

As páginas Web ricas em dados são projetados para serem interpretados e utilizados por humanos e não processados por máquinas. Os objetos representados nestas páginas possuem estrutura textual implícita, podendo ocorrer até mesmo em sequências de texto puro, sem delimitadores separando as informações. A estrutura dos registros que representam implicitamente estes objetos é definida pelo posicionamento dos dados no texto da página, pela formatação utilizada na sua apresentação, ou pelo contexto textual onde está inserido. Por esta razão, dizemos que estes objetos são semi-estruturados [Laender et al., 2002].

Conceitualmente, os registros são compostos por campos ou *atributos*. No Web site *amazon.com*, representado na Figura 1.1, os livros contêm os atributos título, autores e preço, por exemplo. No Web site *monster.com*, representado na Figura 1.2, os registros são posicionados em formato de tabela e os valores dos atributos dispostos nas colunas da mesma. Já na Figura 1.3, os registros são resultados da consulta usando o termo "cars" na máquina de busca *google.com* e os valores dos atributos são dispostos de forma a serem interpretados por humanos, sem delimitadores textuais ou estruturais.

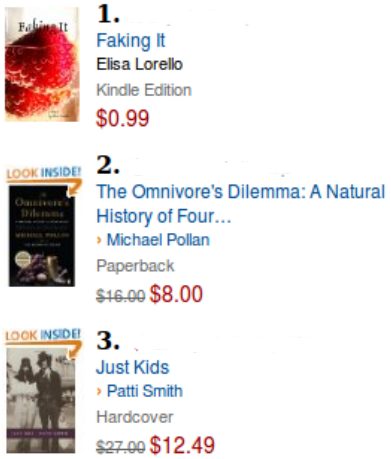

Figura 1.1: Lista de livros de uma página gerada por uma busca no site *amazon.com*. São apresentados três registros contendo os valores dos atributos título, autor, preço, etc.

Jun 8	US-TX-Fort Worth	Web Developer	StarTech Staffing
Jun 8	US-TN-Nashville	Programmer Analyst	OAO
Jun 8	US-CA	Lotus Notes/Domino Developer	Dynamic Staffing
Jun 7	US-CA-San Francisco	Development Manager	LookSmart
Jun 7	US-VA-FallsChurch	Internet Consultant	AppNet, Inc.
Jun 7	US-IL-Chicago	OpenStep Opportunity	Technisource
Jun 7	US-CO	Oracle Database Administrator	Level 3 Communic
Jun 7	US-CA-San Francisco	Programmer/Analyst - COBOL	Boeing

Figura 1.2: Lista de ofertas de emprego obtida em uma página do Web site *monster.com*. Os registros contêm os valores dos atributos data, país, estado, cidade, ocupação e empresa

A extração de valores de atributos correspondentes a objetos em páginas ricas em dados é essencial para aplicações que necessitam utilizar informações presentes nestas páginas de forma estruturada, como coletores de dados e máquinas de consulta es-

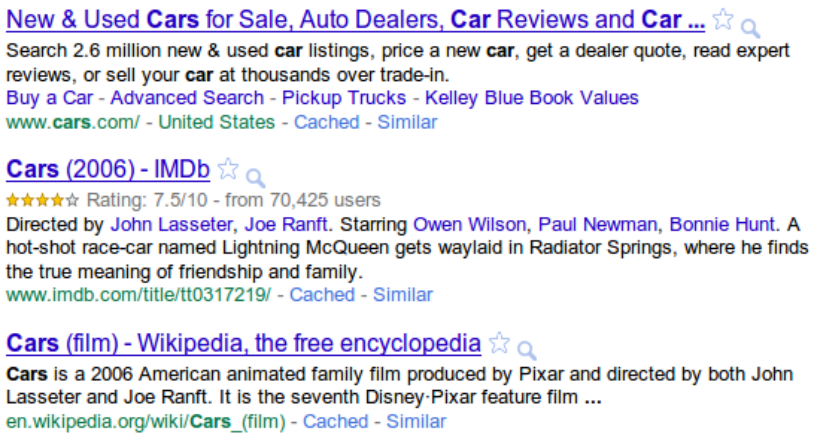

Figura 1.3: Resultados da busca pelo termo “cars” em *google.com*

truturadas. O problema de extração é complexo devido à diversidade do conteúdo e formatação visual e estrutural utilizadas na representação. Os exemplos das Figuras 1.1, 1.2 e 1.3 mostram três formas distintas de apresentação dos objetos. Programas utilizados para extração de dados correspondentes a objetos representados em páginas Web e para o mapeamento dos mesmos em formatos estruturados e padronizados como banco de dados relacionais ou documentos XML são chamadas *wrappers* ou *extratores* [Laender et al., 2002, Liu et al., 2003, Zhao et al., 2005].

A Tabela 1.1 mostra o resultado esperado da extração dos dados dos três livros representados na páginas da Figura 1.1. Todos os registros possuem valores para os atributos título, autor, tipo de edição e preço. Os dois últimos registros possuem também um valor para o atributo preço promocional. A identificação dos valores dos atributos por um humano pode ser feita através da identificação das diferentes cores e tamanhos das fontes, pelas quebras de linha entre os valores de atributos ou pelas tags HTML do código fonte da página, por exemplo.

Título	Autor	Tipo de edição	Preço	Preço promocional
Faking It	Elisa Lorello	Kindle Edition	$0.99	
The Omnivore’s...	Michael Pollan	Paperback	$16.00	$8.00
Just Kids	Patti Smith	Hardcover	$27.00	$12.49

Tabela 1.1: Valores de atributos dos registros correspondentes aos objetos representados na página da Figura 1.1

1.1 Método proposto

Nesta dissertação apresentamos o método MAIt - **M**ore **A**bout **I**t, um novo método para geração automática de extratores. O método foi desenvolvido para gerar extratores capazes de identificar e extrair os registros e valores de atributos independentemente da formatação visual ou estrutural da página que os contém. Os dados identificados são dispostos em formatação XML, viabilizando o processamento das informações por computadores sem auxílio humano. O método MAIt explora a padronização apresentada na estruturação do código HTML, das árvores DOM e do conteúdo textual dos registros para extraí-los e para identificar os valores de seus atributos. Além de descrever o método em detalhes, avaliamos sua eficiência através de experimentos realizados com coleções de páginas Web reais.

O processo de extração de dados empregado pelo MAIt utiliza porções de código HTML contendo registros como exemplos para geração de expressões regulares. Estas são capazes de identificar outras porções de código contendo registros do mesmo Web site. São então identificados elementos textuais comuns a todas as porções e, através destes, sequências de texto que contêm os valores dos atributos. A identificação dos padrões textuais entre as porções de código HTML é feita através de algoritmos de alinhamento de múltiplas sequências textuais [Needleman e Wunsch, 1970, Pereira e Silva, 2006].

Por ser um processo de extração automático, os exemplos utilizados para criação de expressões regulares são encontrados sem intervenção humana. Para isso, o método MAIt faz uso da estrutura das árvores DOM onde os registros são armazenados. Dado que os registros representam objetos de uma mesma classe, as sub-árvores da árvore DOM da página que os contêm são similares. Assim, o método MAIt identifica na árvore DOM um conjunto de sub-árvores similares e assume que estas representam os registros. A similaridade das sub-árvores da árvore DOM da página é identificada através de algoritmos de alinhamento de árvores [Valiente, 2001, Reis et al., 2004]. A padronização da estrutura do código HTML destas sub-árvores torna possível a criação de uma expressão regular que as represente.

Além de identificar porções de código HTML com estrutura idêntica àquelas dadas como exemplo, a expressão regular também é capaz de extrair os campos de texto que contêm os valores dos atributos de cada registro. Os valores dos atributos são extraídos fazendo-se uso de delimitadores textuais e tipos básicos comumente encontrados em páginas ricas em dados.

Como dito, o método MAIt faz uso da padronização da estrutura das árvores DOM, do código HTML e do conteúdo textual dos registros para extrair registros e valores de atributos de páginas ricas em dados. O processo de extração de dados utilizado no MAIt é descrito em mais detalhes no Capítulo 3.

1.2 Contribuições

Apesar de os algoritmos de alinhamento de árvores e de alinhamento de múltiplas sequências serem problemas desafiadores, eles oferecem a solução para o problema de extração de dados em páginas Web. A utilização de algoritmos de alinhamento de árvores na extração de elementos de páginas Web não é uma inovação do método proposto. Entretanto, a aplicação deste método na identificação de porções da página de interesse contendo os dados de interesse para geração de uma expressão regular capaz de identificar estes dados em todas as páginas do mesmo Web site é inovador.

Outros autores têm proposto métodos capazes de extrair elementos de páginas Web. Entretanto, devido à diversidade de formatação dos registros em diferentes Web sites, os outros métodos não são capazes de identificar os dados em uma grande variedade de Web sites. O método MAIt considera que todos os registros de um Web site possuem estruturas similares entre si, o que permite identificá-los. Ao identificar de forma automática esta similaridade, o método é automaticamente adaptado a diferentes Web sites.

O nosso método não faz uso de informações externas à página Web para ter os elementos extraídos. Ele também não faz uso de interação externa ou humana para encontrar exemplos de registros ou para gerar uma expressão regular utilizando estes exemplos. Ele é completamente automático e utiliza somente as informações disponíveis na página de

interesse.

Além disso, é capaz de identificar em uma sequência textual os valores de um ou mais atributos. Este problema é abordado em outros trabalhos específicos para este tema, mas o método MAIt trata esta situação durante o processo de extração dos valores dos atributos.

Assim, a principal contribuição deste trabalho é a criação de um gerador completamente automático de extratores capazes de identificar e extrair objetos e os valores de seus atributos de páginas Web ricas em dados utilizando informações disponíveis na página e mapear o resultado do processo de extração em documentos XML estruturados.

1.3 Organização

O texto desta dissertação é organizado como se segue. O Capítulo 2 descreve outros trabalhos cujo objetivo é a extração de dados em páginas Web ricas em informações. A metodologia, os resultados, as fraquezas e os pontos fortes de cada um dos métodos são discutidos. No Capítulo 3 o método proposto neste trabalho é apresentado em mais detalhes. Todos os passos necessários para a identificação dos elementos a serem extraídos são mostrados. Os algoritmos implementados são explicados e as premissas e as soluções utilizadas são definidas. Os experimentos realizados para verificar a eficácia do método proposto e comparações com métodos relacionados são descritas no Capítulo 4. A metodologia e os parâmetros utilizados nos experimentos também são apresentados. Finalmente, no Capítulo 5 são apresentadas as conclusões e os possíveis trabalhos a serem desenvolvidos a partir do método MAIt.

Capítulo 2

Trabalhos relacionados

Vários métodos e ferramentes têm sido propostos com o intuito de resolver o problema de extração de dados de páginas Web. Um estudo sobre os trabalhos desenvolvidos sobre este tema é apresentado em [Laender et al., 2002]. De forma geral, os diversos trabalhos na literatura lidam com este problema de formas distintas. Em [Liu et al., 2003] são realizadas comparações entre o conteúdo textual de páginas de exemplo. Em [Reis et al., 2004] e [Dalvi et al., 2009] é utilizado Alinhamento de Árvores DOM de páginas de exemplo. Em [Zhao et al., 2005], os autores propõem a utilização de características visuais dos dados a extrair. Em [Pereira e Silva, 2006] é proposto um algoritmo que utiliza Alinhamento Múltiplo de Sequências para gerar extratores de dados. Neste capítulo apresentamos estes métodos, enfocando seus pontos positivos e comparando-os com o método proposto neste trabalho.

O método proposto em [Liu et al., 2003] utiliza comparação do conteúdo textual da página na identificação do conteúdo a ser extraído e se mostra bastante eficiente no que se propõe. Entretanto, este método se restringe a extrair dados contidos em elementos HTML relacionados a tabelas e formulários, como “TABLE”, “FORM”, “TR”, “TD”, etc. O método é chamado de *MDR - Mining Data Records in Web Pages*. A premissa do MDR é de que os dados a serem extraídos pertencem a uma região da página, são formatados por *tags* HTML similares e contidos em nodos adjacentes e de mesmo pai na árvore DOM da página. A identificação dos registros é feita formando-se sequências textuais similares do conteúdo dessas *tags*. Este modelo poderia ser expandido para qualquer

elemento HTML, entretanto, o alto custo computacional da identificação dos registros o tornaria inviável, já que são feitas combinações de várias sequências textuais até que um padrão seja encontrado. Assim, reduzindo-se os casos para dados contidos em tabelas e formulários a quantidade de combinações diminui substancialmente.

Outro método de extração encontrado na literatura foi proposto por [Reis et al., 2004]. Utilizando Distância de Edição de Árvores [Selkow, 1977, Valiente, 2002], assim como em nosso trabalho, este método identifica e extrai informações relevantes de páginas Web. Entretanto, diferentemente do MAIt, o foco de [Reis et al., 2004] é a extração de textos contidos em páginas de notícias. Para isso, a distância de edição entre as árvores DOM de todas as páginas de interesse é calculado e as sub-árvores idênticas são descartadas, pois são elementos que se repetem entre as diversas páginas, como menus, temas, publicidades e âncoras, restando os textos das notícias. Em nosso método calculamos a distância de edição entre todas as sub-árvores da árvore DOM da página de interesse através do algoritmo proposto por [Valiente, 2001]. Este algoritmo nos possibilita encontrar sub-árvores similares e identificar elementos semelhantes na página de interesse, no caso os blocos de dados contendo os registros a serem extraídos posteriormente.

O extrator descrito em [Miao et al., 2009] também considera a árvore DOM na identificação de registros. De acordo com [Miao et al., 2009], os registros são contidos em sub-árvores acessíveis por caminhos de *tags* idênticos desde a raiz da árvore DOM até o nodo raiz da sub-árvore que contém cada um dos registros. Assim, o método é capaz de identificar os registros presentes na página de interesse, de forma automazida. Entretanto, este extrator é incapaz de identificar atributos e seus respectivos valores.

O método proposto em [Dalvi et al., 2009] também utiliza técnicas de Alinhamento de Árvore [Valiente, 2001] para extração de valores de atributos e registros em páginas Web. A principal premissa deste trabalho é de que os blocos de dados que contêm os registros sofrem alterações em sua composição frequentemente, tornando seus extratores obsoletos. A solução proposta por [Dalvi et al., 2009] é inferir possíveis composições de cada bloco de dados, utilizando versões prévias do mesmo. A técnica de Alinhamento de Árvores é aplicada na identificação das modificações atribuídas ao bloco de dados

entre suas diversas versões. Desta forma, um extrator capaz de identificar os valores dos atributos e os registros é gerado, mesmo que os blocos de dados que os contêm sofram alterações não significativas com o passar do tempo. A geração do extrator só é possível através de exemplos de valores de atributos a serem identificados. Estes exemplos são providos através de interação humana, o que torna este método não automático. Além disso, são necessárias versões de diferentes períodos de uma dada página com o intuito de inferir estocasticamente possíveis formatos dos blocos de dados contendo os registros e atributos.

O trabalho descrito em [Pereira e Silva, 2006] utiliza técnicas de Alinhamento Múltiplo de Sequências [Gusfield, 1997] para gerar extratores. Ele considera que os valores dos atributos são contidos em sequências de texto e *tags* HTML que possuem um padrão quando comparados a atributos equivalentes de outros registros. O padrão considera que essas sequências podem ser divididas em três partes: (1) os valores dos atributos, textos que variam de registro em registro, (2) um prefixo e (3) um sufixo, que são equivalentes entre si, quando comparados em diferentes registros. No método de [Pereira e Silva, 2006] é necessário que exemplos de atributos com seus prefixos e sufixos sejam selecionados manualmente. Estes exemplos são processados sucessivamente pelo Algoritmo de Alinhamento Múltiplo de Sequências que, através de técnicas de programação dinâmica, gera uma sequência contendo os elementos que se repetem nos exemplos e *gaps*, caso contrário. A sequência final é transformada em uma expressão regular capaz de identificar os valores de outros atributos equivalentes àqueles contidos nos exemplos dados, tornando viável a extração dos registros que os contêm.

O método que mais se aproxima do MAIt que utilizamos para comparar nossos resultados experimentais foi proposto em [Zhao et al., 2005]. O ViNTs - **V**isual **i**nformation a**N**d **T**ag **s**tructure - faz uso de informações visuais e da estrutura de *tags* do código HTML da página de interesse para identificar os registros contidos na mesma. Ele pressupõe que os dados de interesse possuem uma formatação padrão: os atributos são agrupados e os registros separados por uma linha em branco, posicionados em uma área distinta, grande e central da página. Além disso, os atributos podem ser textos, âncoras, âncoras com

texto, âncoras iniciados por um numeral, textos iniciados por um numeral, âncoras com texto iniciados por um numeral ou uma linha iniciada pela *tag* HTML “HR”. Estes atributos possuem um posicionamento padronizado nos registros, podendo estar aninhados ou deslocados entre si. Com essas informações, vários conjuntos de elementos são formados, sendo que somente um destes contém os registros a serem extraídos. Alguns parâmetros são usados na escolha do conjunto correto, aquele que contém os registros a serem extraídos: área visual ocupada por todos os elementos do conjunto; distância do centro da área visual do conjunto até o centro da página; número de itens do conjunto; número médio de caracteres por item do conjunto. Identificado o conjunto de registros, recupera-se o caminho destes até a raiz da árvore DOM e elementos textuais que os delimitam. Com este caminho, é possível identificar em outras páginas similares à atual o conjunto de registros e com os delimitadores, separá-los. Um dos pontos positivos do nosso método em relação ao métodos ViNTs é a identificação de valores de atributos contidos em sequências de texto. Nestas, na maioria dos casos, não existe diferenciação visual entre os valores dos atributos, o que inviabiliza a identificação correta dos mesmo pelo ViNTs.

O trabalho de [He et al., 2007] é uma evolução do método proposto por [Zhao et al., 2005]. O objetivo do novo método é otimizar a identificação de valores de atributos contidos em sequências de texto visualmente similares. Para isso, são criados grupos de sequências de texto equivalentes e calculada a distância de edição entre termos de sequências equivalentes, considerando modificações visuais e de tipo dos dados envolvidos. Grandes valores de distância de edição indicam que as sequências textuais devem ser divididas em duas partes, às quais são posteriormente comparadas. Formam-se, assim, grupos de termos semelhantes, os quais representam valores dos mesmos atributos.

O método proposto por [Liu et al., 2010], assim como o de [Zhao et al., 2005] e o de [He et al., 2007], utiliza informações visuais na identificação de registros e valores de atributos. Entretanto, no ViDE - **Vi**sion-based **D**ata **E**xtractor - a árvore DOM da página não é levado em consideração como nos outros métodos. Neste trabalho, os registros são identificados pelo seu padão de posicionamento, tamanho e fonte e atributos visuais de

conteúdos vizinhos. Os valores dos atributos são identificados através de sua ordem de apresentação nos registros e de textos estáticos que não representam atributos.

O método de [He et al., 2007] não pode ser comparado ao MAIt devido à não disponibilidade de detalhes implementacionais e de algoritmos por parte de seus autores. Já o método ViDE não foi utilizado em nossos experimentos por ter sido publicado na literatura recentemente. Assim, o método ViNTs, proposto por [Zhao et al., 2005] foi utilizado nos experimentos deste trabalho descrito no Capítulo 4. O motivo da escolha do ViNTs em detrimento dos outros métodos de extração descritos anteriormente se deu por este não necessitar de interação humana e pelo fato de que em [Zhao et al., 2005] este já ser comparado com o MDR [Liu et al., 2003].

O método MAIt possui pontos diferenciais positivos quando comparado aos trabalhos relacionados apresentados. Dentre estes, pode-se destacar que nosso método é completamente automático, não necessitando de intervenção humana no processo de identificação de registros e valores de atributos. Além disso, nosso método não se restringe a extrair registros contidos em tipos de *tags* HTML ou formatações visuais específicos.

Capítulo 3

O Método MAIt

Neste capítulo, apresentamos os detalhes do método MAIt - **M**ore **A**bout **It**. Como já descrito, o objetivo deste método é gerar de forma automática extratores capazes de identificar registros e valores de seus atributos que ocorrem em páginas ricas em dados. Estas informações são contidas em trechos do código fonte HTML das páginas chamados *blocos de dados*. Conceitualmente, cada bloco de dados contém um único registro e, da mesma forma, cada registro pertence a um único bloco de dados.

O processo de geração de extratores proposto neste trabalho consiste em gerar uma expressão regular capaz de identificar no código fonte HTML os blocos de dados e, a partir destes, os registros e os valores dos atributos. A aplicação do método MAIt em páginas ricas em dados é possível pelo fato destas apresentarem características típicas de páginas geradas automaticamente, como conteúdo e estrutura pré-definidos.

Em resumo, o método MAIt pode ser divido em três fases:

1. Identificação de exemplos de blocos de dados.

2. Geração de uma expressão regular capaz de identificar os blocos de dados.

3. Extração de blocos de dados, registros e valores de atributos.

O restante desde Capítulo é organizado como se segue. Na Seção 3.1, apresentamos características e propriedades das páginas ricas em dados exploradas pelo método

MAIt no processo de geração de extratores. Na Seção 3.2, introduzimos nosso método de geração de extratores mostrando, em linhas gerais, a aplicação de algoritmos de alinhamento de árvores e de sequências de texto neste processo. Nas seções seguintes, detalhamos o processo de geração de extratores, desde a identificação de exemplos de blocos de dados na Seção 3.3 e do padrão do conteúdos dos mesmos na Seção 3.4 até a extração de registros e valores de atributos na Seção 3.5.

3.1 Características de Páginas Ricas em Dados

Páginas ricas em dados pertencentes a um mesmo Web site apresentam propriedades importantes exploradas pelo método MAIt. Por pertencerem ao mesmo site, estas páginas possuem áreas, temas e textos em comum, como cabeçalhos, menus, rodapés, áreas de propaganda, etc. Como consequência, a árvore DOM dessas páginas possuem várias sub-árvores idênticas. Visualmente, entretanto, essas páginas possuem áreas que as diferenciam entre si, onde são dispostos os objetos de interesse deste trabalho, os registros. Estes são armazenados em sub-árvores irmãs contidas nestas áreas.

A Figura 3.1 mostra um exemplo de árvore DOM de uma suposta página rica em dados. Todas as outras páginas pertencentes ao mesmo Web site da página de exemplo possuem a estrutura composta pelos nodos “D”, “F”, “G”, “H” e “I”, que podem ser cabeçalhos, rodapés, menus, áreas de propaganda, etc. Neste exemplo, os blocos de dados são representados pelas sub-árvores irmãs iniciadas nos nodos “E”, contidas na área representada pelo nodo “F”.

Outra propriedade de páginas ricas em dados é que os registros implicitamente representados nos blocos de dados são instâncias de uma mesma classe de objetos. Os registros são diferenciados entre si pelos valores de seus atributos, já que possuem as mesmas características. A semelhança entre os registros é refletida nas sub-árvores que os contêm. Desta forma, as sub-árvores contendo os registros de um mesmo tipo tendem a apresentar estruturas similares entre si, variando apenas o conteúdo e a formatação do texto nas suas folhas, onde são armazenados os valores dos atributos.

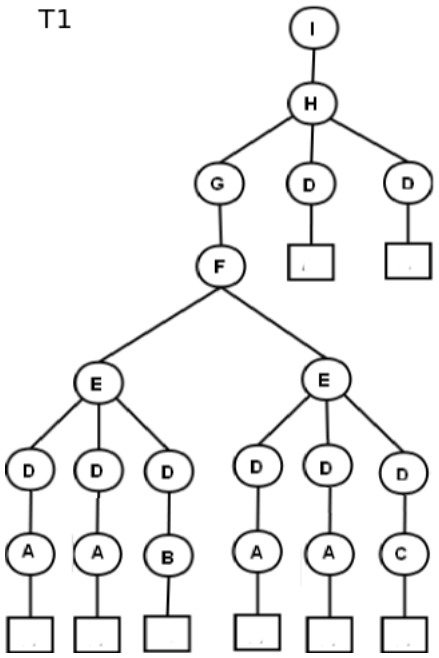

Figura 3.1: As sub-árvores iniciadas nos nodos “E” contêm os blocos de dados a serem extraídos

A similaridade entre as sub-árvores que contêm blocos de dados torna possível a utilização do Algoritmo de Alinhamento de Árvores proposto em [Valiente, 2002] na identificação das mesmas. Para isso, é preciso ignorar, para fins de alinhamento das sub-árvores, os nodos que armazenam ou modificam visualmente o conteúdo textual da página. Isto fará com que as sub-árvores que contêm os blocos de dados se tornem isomórficas e sejam identificadas como tais pelo Algoritmo de Alinhamento de Árvores.

Na árvore da Figura 3.1, por exemplo, para que o Algoritmo de Alinhamento de Árvores identifique as sub-árvores iniciadas nos nodos “E” como isomórficas, é necessário ignorar os nodos “B” e “C”. Estes serão ignorados se representarem *tags* HTML de formatação textual ou que alterem visualmente o valor do atributo sem alterar sua semântica, como “B”, “I”, “BR”, “FONT”, “H1”, “H2”, “A”, etc.

A Figura 3.2 mostra três blocos de dados representando registros com informações sobre livros do Web site *amazon.com*. Os valores do atributo autor são visualmente diferentes em cada registro. No primeiro livro, o autor *Elisa Lorello* não possui uma âncora para referência externa e, portanto, não está sublinhado. Já os valores do atributo autor dos dois últimos livros estão destacados. Apesar das diferenças visuais, os três valores repre-

sentam o mesmo atributo autor: *Elisa Lorello*, *Michael Pollan* e *Patti Smith*. As âncoras apresentadas em dois dos autores são representadas pela *tag* "A" e a presença deste nodo torna as sub-árvores dos registros estruturalmente diferentes entre si. Ao ignorar este nodo durante o processo de alinhamento das sub-árvores, estas tornam-se isomórficas, viabilizando a identificação dos blocos de dados de interesse.

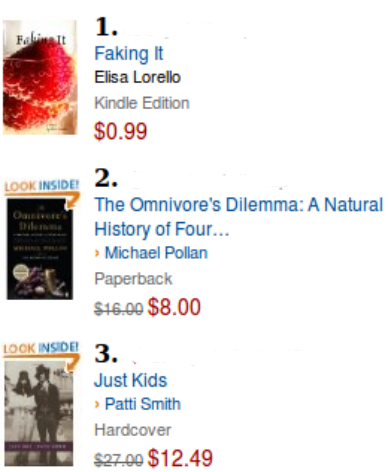

Figura 3.2: Lista de livros do Web site *amazon.com*: os valores do atributo autor são visualmente diferentes entre cada um dos registros

Além da forma, outra característica das sub-árvores que contêm blocos de dados aplicada na identificação de registros é o seu conteúdo. Registros são representações de objetos, compostos por atributos e armazenados na árvore DOM da página de forma a possibilitar a interpretação da informação por um humano. Estas características são utilizadas pelo MAIt na diferenciação de uma lista de seleção de uma lista de registros, como as mostradas na Figura 3.3, por exemplo. Ambas possuem as propriedades previamente descritas: assim como os registros, os itens da lista de seleção pertencem a sub-árvores irmãs e isomórficas. O método MAIt considera a quantidade de nodos e de informação textual contida em cada sub-árvore para selecionar o conjunto de sub-árvores irmãs e isomórficas. Desta forma, as sub-árvores que contêm os blocos de dados são aquelas que possuem maior número de nodos e maior quantidade de texto.

Assim, é possível identificar blocos de dados contendo registros em páginas Web geradas à partir de consultas em banco de dados ou máquinas de busca. Estes blocos de dados são utilizados como exemplos para geração de uma expressão regular capaz de

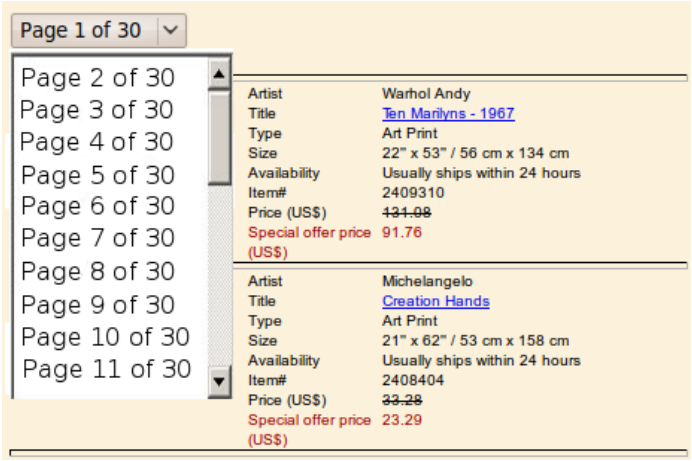

Figura 3.3: Página Web contendo uma longa lista de seleção e apenas dois registros

extrair registros e valores de atributos de páginas do mesmo Web site. Na Seção 3.3 a aplicação destas premissas é explicada em mais detalhes.

3.2 Processo de Extração

O processo de extração de dados envolve, além da estrutura das sub-árvores que os contêm, o conteúdo de seu código HTML. Como dito anteriormente, os registros de páginas de um Web site representam instâncias de objetos de uma mesma classe. Como consequência, estes registros possuem formatação visual semelhantes, propriedade que é refletida tanto na estrutura das sub-árvores que os contêm quanto no código HTML de seus blocos de dados. Estes padrões tornam possível a criação de expressões regulares capazes de identificar os blocos de dados.

O método MAIt utiliza uma adaptação do Algoritmo de Alinhamento de Sequências descrito em [Pereira e Silva, 2006] na identificação do padrão textual dos blocos de dados previamente encontrados pelo Algoritmo de Alinhamento de Árvores. Baseado no Algoritmo de Alinhamento de Múltiplas Sequências [Gusfield, 1997], o algoritmo consiste em dividir sequências de texto em segmentos de tipos pré-definidos e alinhar seus termos equivalentes ou similares. O resultado do alinhamento das várias sequências de texto é uma expressão formada pelos termos comuns a todas as sequências separados por um *gap*, representando o padrão da composição textual das sequências de entrada. No método MAIt, esta expressão é gerada através do alinhamento dos blocos de dados. Os

blocos de dados são divididos e alinhados em comentários HTML, *tags* HTML, símbolos HTML em geral, datas, números, endereços ou URLs, endereços de *e-mail*, símbolos de moedas, pontuações e palavras em geral. A expressão criada representa o padrão da composição textual dos blocos de dados alinhados e são transformadas em expressões regulares capazes de identificar outros blocos de dados pertencentes ao mesmo Web site.

Utilizando o padrão da composição textual dos blocos de dados, também é possível encontrar sequências textuais onde possivelmente são armazenados os valores dos atributos a serem extraídos. Como os valores dos atributos não se repetem em todos os registros, é possível inferir que estes estão contidos em sequências de texto que variam em cada bloco de dados. Desta forma, supõe-se que os *gaps* da expressão formada no alinhamento dos blocos de dados representem segmentos de texto que contêm os valores de um ou mais atributos, os quais são delimitados por sequências comuns a todos os blocos.

O processo de geração de expressões regulares a partir de blocos de dados utilizando o Algoritmo de Alinhamento de Múltiplas Sequências será detalhado na Seção 3.4.

Páginas Web contendo registros são geradas a partir de consultas em banco de dados ou máquinas de busca. O formato e o conteúdo dos objetos variam de acordo com a origem da consulta, gerando estilos diferentes, como os mostrados na Figura 3.4. O objetivo do método MAIt é identificá-los independentemente de sua origem.

Através do padrão textual dos blocos de dados e da expressão regular gerada, é possível identificar segmentos de texto que não se repetem em todos os blocos de dados. Estes campos de texto contêm valores de um ou mais atributos que compõem os registros e são, por isso, doravante chamados de *sequências de valores*. Na Figura 3.4(a), por exemplo, a segunda coluna da tabela de ofertas de emprego é formada por sequências de valores de três atributos para cada registro ou linha da tabela. Neste caso, as sequências são equivalentes, por estarem igualmente posicionadas entre elementos que se repetem em todos os registros da página. A sequência “US-CA-San Francisco” contém os valores “US”, “CA” e “San Francisco”, que representam os valores dos atributos país, estado e cidade de um registro de oferta de emprego.

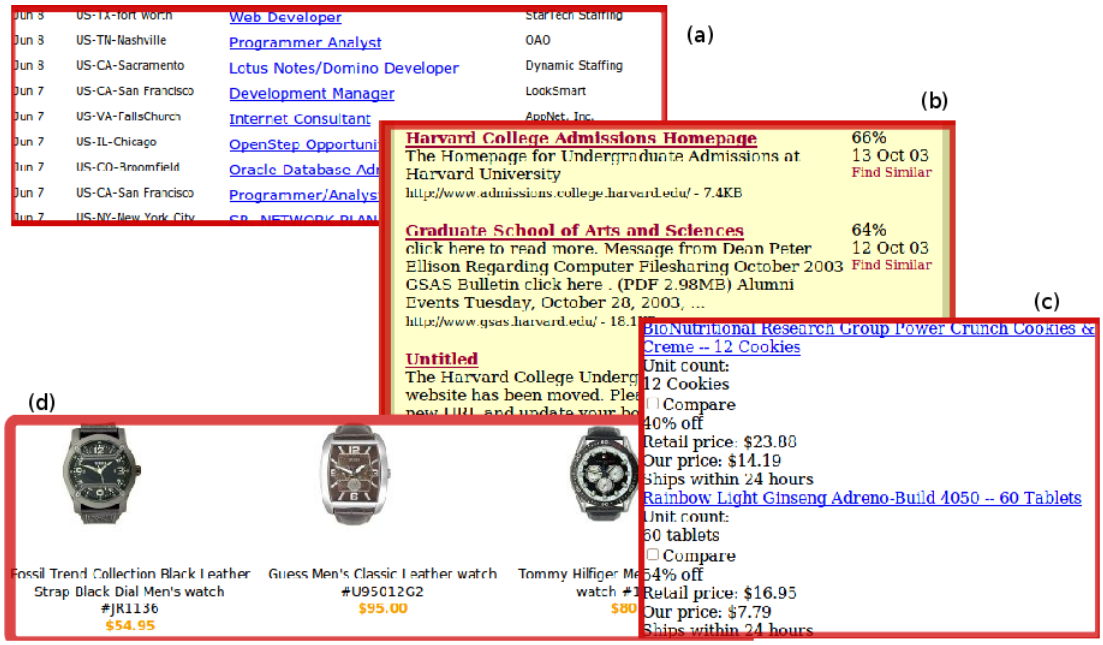

Figura 3.4: Quatro Web sites contendo estilos de objetos de domínios diferentes: (a) ofertas de emprego, (b) páginas Web, (c) remédios e (d) relógios

As sequências de valores equivalentes possuem características que permitem a definição da quantidade de valores de atributos contidos em cada registro. O método MAIt assume que sequências de valores equivalentes contêm a mesma quantidade de valores de atributos, os quais são igualmente posicionados entre si. Porém, é considerada a possibilidade de um ou mais atributos não possuir valor.

Através de observações feitas nas coleções utilizadas nos experimentos descritos no Capítulo 4, foi possível identificar propriedades das sequências de valores aplicáveis na divisão das mesmas em valores de atributos. A primeira propriedade é referente ao comprimento textual da sequência. Sequências de texto com mais de 60 caracteres tendem a ser textos descritivos do objeto em questão e, por isso, não devem ser divididas. Na Figura 3.4(b) o texto em Inglês "The Homepage for Undergraduate Admission at Harvard University" é um exemplo desta situação. Esta sequência representa o valor de um único atributo. Então, as propriedades que se seguem são aplicáveis a sequências mais curtas, onde possivelmente existem valores de mais de um atributo, como na sequência "US-CA-San Francisco", que contém os valores de três atributos.

Em sequências de valores com menos de 60 caracteres é possível encontrar datas,

números, endereços ou URLs e endereços de *e-mail*. Além de representarem valores de atributos, estes campos podem delimitar os valores de outros atributos, assim como símbolos de pontuação em geral. A Figura 3.5 mostra um registro do Web site *american.com*. O bloco de dados contendo este registro possui a sequência de texto *http://www.american.edu/spa/admissionsgrad.html - 10.0KB - American University's Web Site*. Esta sequência é divisível em três partes, pois contém um endereço ou URL e dois delimitadores - os hifens.

Esta propriedade dos valores dos atributos, como dito, foi identificada através de observações feitas nas páginas das coleções utilizadas nos experimentos descritos no Capítulo 4. Experimentos adicionais para validação da mesma podem ser feitas, porém, com o valor fixo em 60 caracteres obtivemos valores satisfatórios de eficiência, como será demonstrado posteriormente.

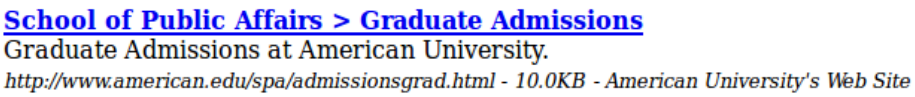

Figura 3.5: Registro extraído do Web site *american.edu* com uma sequência de texto composta por um endereço ou URL, tamanho e origem da página

Como dito, sequências de valores equivalentes contêm o mesmo número de valores de atributos. Se analisadas separadamente, de acordo com as propriedades apresentadas, não é possível definir a quantidade de divisões a serem feitas nas sequências e extrair corretamente seus valores. Então, para definir a quantidade de valores de atributos esperados em sequências de valores equivalentes, o método MAIt as divide e considera o número de divisões que ocorre na maioria das sequências equivalentes.

Definido o número de divisões a serem aplicados em sequências de valores equivalentes, os valores dos atributos são identificados, finalizando o processo de extração em páginas ricas em dados.

3.3 Identificação de Blocos de Dados

Definidas as premissas para extração de páginas ricas em dados nas Seções 3.1 e 3.2, torna-se possível descrever os algoritmos utilizados no desenvolvimento do MAIt. O processo de extração de dados do MAIt é dividido em três fases. A primeira dessas fases é a identificação de blocos de dados que, como já descrito, são trechos de código HTML que contêm todos os valores dos atributos de um registro.

Em nosso método, assumimos que blocos de dados contendo registros que representam objetos de uma mesma classe estão contidos em sub-árvores similares pertencentes à árvore DOM das páginas ricas em dado fornecidas como entrada. Assim, para encontrar estas sub-árvores similares, utilizamos em nosso método o Algoritmo de Alinhamento de Árvores proposto por Valiente [Valiente, 2002]. Este algoritmo é baseado no conceito de Distância de Edição de Árvores [Selkow, 1977, Valiente, 2001] e é bastante adequado para o nosso problema, pois seu objetivo é encontrar sub-árvores de uma dada árvore que sejam isomórficas entre si. O algoritmo agrupa estas sub-árvores em *classes de equivalência*, de forma que sub-árvores isomórficas pertençam à mesma classe. Como consequência, os blocos de dados contidos nas sub-árvores são também agrupados e a classe dos objetos representados pelos registros, cujos dados estão nos blocos de dados, é recuperada de forma implícita.

No entando, deve ser observado que, de formal geral, é esperado que mais de uma classe de equivalência seja encontrada pelo algoritmo. Desta forma, é necessário identificar qual das classes encontradas contêm os registros de interesse para uma aplicação. Embora existam várias formas de realizar essa escolha, inclusive com o apoio do usuário, em nosso trabalho essa escolha é feita com base em uma pontuação atribuída a cada classe, de tal forma que a classe com a maior pontuação é utilizada. A pontuação é um valor proporcional ao número de nodos contidos na sub-árvore em questão e à soma do comprimento da união de todos as porções de texto da mesma, atendendo às propriedades esperadas dos blocos de dados.

Entrada: L lista de árvores DOM de páginas de um Web site.

```
1 início
2     para cada Árvore A ∈ L faça
3         A ← limpaÁrvore(A);
4         C ← extraiClasses(A);
5         C_I ← ClasseDeInteresse(C);
6         N ← BlocosCandidatos(C_I,A);
7         N_B ← extraiBlocos(N,M);
8     fim
9 fim
```

Algoritmo 3.1: Identificação de Blocos de Dados

Remoção de Elementos Irrelevantes das Árvores DOM

No primeiro passo para identificar as sub-árvores contendo blocos de dados, na Linha 3 do Algoritmo 3.1, são removidos os nodos que modificam visualmente os valores dos atributos e, consequentemente, a estrutura das sub-árvores dos blocos de dados. As sub-árvores com raízes “HEAD”, “STYLE”, “IMG” e “INPUT”, também, são removidas por não serem relevantes para o propósito de extração de dados. Da mesma forma, os nodos “B”, “I”, “CENTER”, “A”, “H1”, “H2”, dentre outros, são removidos por alterarem a estrutura esperada da árvore. Neste último caso, somente o nodo é removido, a sub-árvore iniciada a partir deste é mantida. Este passo é obrigatório, já que os elementos removidos podem fazer com que o Algoritmo de Alinhamento de Árvores classifique uma sub-árvore contendo um bloco de dados em uma classe de equivalência não esperada no próximo passo, devido às diferenças na estrutura das sub-árvores.

O processo de remoção de informações irrelevantes de uma árvore DOM é exemplificado na Figura 3.6. Esta figura mostra duas árvores T1 e T2, onde a última é gerada com a remoção de nodos e sub-árvores desnecessárias da primeira. Na árvore T2, é possível verificar visualmente a formação de sub-árvores isomórficas, que não ocorriam em T1.

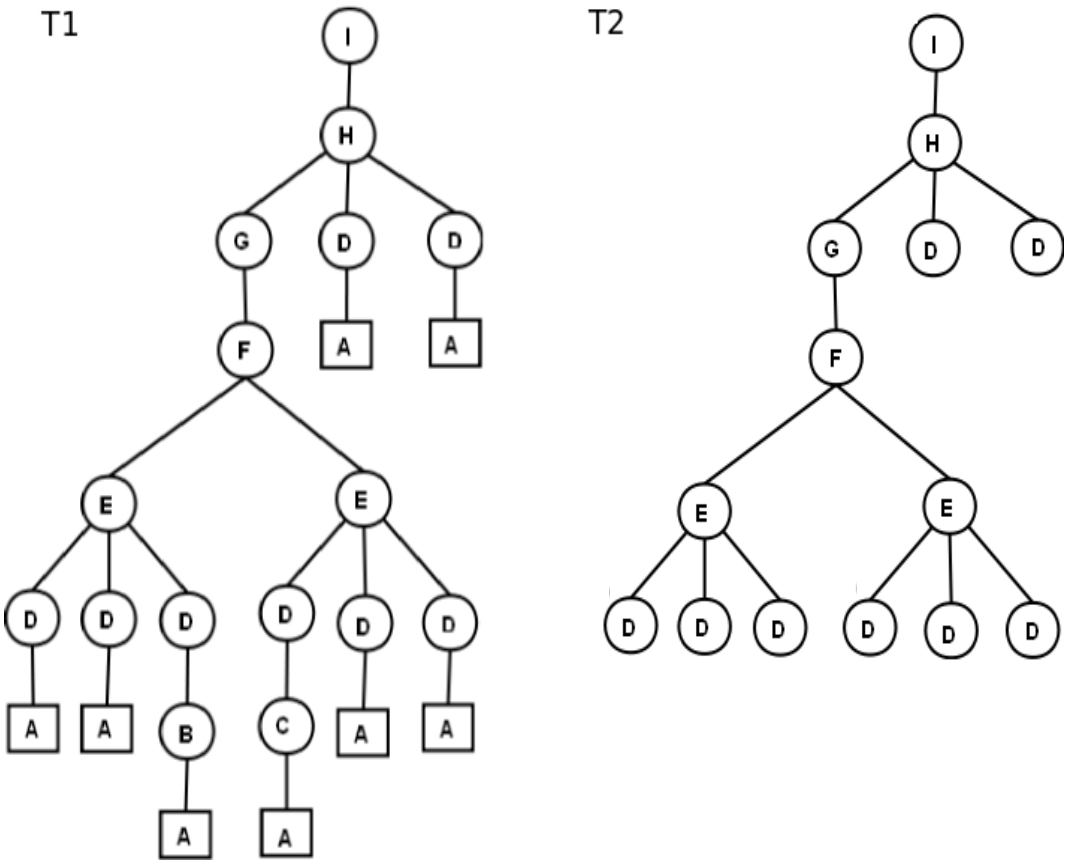

Figura 3.6: T1 e sua versão sem nodos desnecessários, T2

Extração de Classes de Equivalência

Na Linha 4 do Algoritmo 3.1, utilizamos o Algoritmo de Alinhamento de Árvores descrito em [Valiente, 2002]. Este algoritmo recebe como entrada a árvore DOM da página e calcula a distância de edição - número de modificações, remoções ou adições - necessárias para tornar suas sub-árvores idênticas. Este algoritmo é do tipo *bottom-up* e utiliza a distância de edição das sub-árvores para calcular a distância de edição da árvore que as contém. Sub-árvores idênticas possuem distância de edição nula e, portanto, são rotuladas com a mesma classe de equivalência.

Este algoritmo é normalmente utilizado na identificação da similaridade entre as árvores de duas ou mais páginas. Duas páginas são similares se suas raízes forem rotuladas na mesma classe de equivalência. A Figura 3.7 mostra o resultado do alinhamento de duas árvores genéricas. As sub-árvores isomórficas são destacadas com a mesma cor.

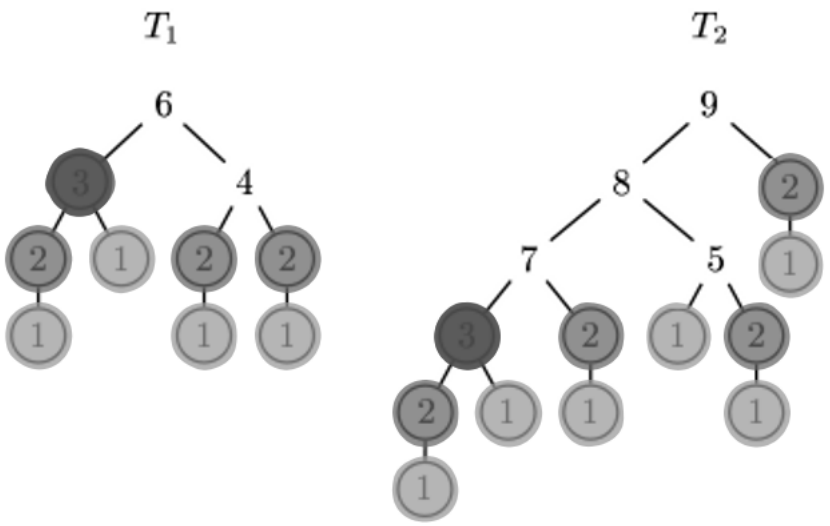

Figura 3.7: Classes de equivalência das sub-árvores das árvores T1 e T2

O objetivo deste passo no método MAIt é agrupar as sub-árvores de uma ou mais páginas de um Web site de acordo com sua similaridade. Estas são agrupadas em uma mesma classe de equivalência e são candidatas a conter os blocos de dados, registros e valores dos atributos.

Identificação da Classes de Equivalência de Interesse

Na Linha 5 do Algoritmo 3.1, o Algoritmo 3.2 é utilizado na localização da classe de equivalência de maior pontuação. Este algoritmo calcula a pontuação de cada classe de equivalência para, então, definir aquela com maior pontuação. A pontuação de uma classe de equivalência é o somatório da pontuação dos nodos rotulados nesta classe.

O Algoritmo 3.2 utiliza uma lista para armazenar a pontuação de cada classe de equivalência. Na Linha 4 a pontuação de um dado nodo é calculada e armazenada na posição da lista correspondente à sua classe de equivalência e, entre as Linhas 5 e 7, a classe de equivalência de maior pontuação é salva para uso posterior.

Entrada: A é a árvore DOM da página de interesse.

Saída: C_I é a classe de equivalência com maior pontuação.

Dados: L é uma lista das pontuações de uma classe de equivalência.

```
1  início
2      para cada nodo em A faça
3          classe ← classeDoNodo(nodo);
4          L[classe] ← L[classe] + pontuaçãoDoNodo(nodo);
5          se L[classe] > maior_pontuação_encontrada então
6              maior_pontuação_encontrada ← L[classe];
7              classe_com_maior_pontuação ← classe;
8          fim
9      fim
10 fim
```

Algoritmo 3.2: Encontra a Classe de Equivalência de Interesse

Extração de Blocos de Dados

Dentre as sub-árvores agrupadas na classe de equivalência de maior pontuação, algumas contêm blocos de dados. Então, na Linha 6 do Algoritmo 3.1, a lista contendo os nodos raízes destas sub-árvores é dada como entrada para o Algoritmo 3.3, responsável por encontrar as sub-árvores que contêm blocos de dados.

Os nodos raízes das sub-árvores que contêm blocos de dados possuem um ascendente em comum. O Algoritmo 3.3 consiste em agrupar os nodos irmãos da lista de entrada e calcular a pontuação dos nodos por grupo. Cada grupo é representado pelo nodo pai de seus nodos, os quais são armazenados em uma lista de nodos. Esta lista armazena em cada posição, o somatório da pontuação dos nodos de um dado grupo. O grupo de nodos com maior pontuação é aquele que contém os blocos de dados. O Algoritmo 3.3 retorna o nodo pai dos nodos que contém os blocos de dados.

Além disso, o Algoritmo 3.3 calcula a pontuação média dos nodos dos blocos de dados. A quantidade de nodos por grupo é calculada na Linha 4 e na Linha 11 a média

aritmética da pontuação dos nodos de maior pontuação é calculada.

Entrada: N: lista dos nodos rotulados na classe de equivalência de interesse.

Saída: N_B é a lista de nodos filhos do nodo com maior pontuação.

Saída: M é a média da pontuação dos nodos raízes de sub-árvores que contêm blocos de dados.

```
1  início
2      para cada nodo em N faça
3          nodo_pai ← paiDoNodo(nodo);
4          D[nodo_pai] ← D[nodo_pai] + 1;
5          P[nodo_pai] ← P[nodo_pai] + pontuaçãoDoNodo(nodo);
6          se P[nodo_pai] > maior_pontuação_encontrada então
7              maior_pontuação_encontrada ← P[nodo_pai];
8              nodo_com_maior_pontuação ← nodo_pai;
9          fim
10     fim
11     M ← P[nodo_com_maior_pontuação] / D[nodo_com_maior_pontuação];
12 fim
```

Algoritmo 3.3: Extração de Blocos de Dados.

O último passo do Algoritmo 3.1 é a identificação de blocos de dados contidos em nodos rotulados em diferentes classes de equivalência. São candidatos os nodos irmãos daqueles pertencentes ao grupo de maior pontuação identificado previamente pelo Algoritmo 3.3.

O Algoritmo 3.4 é responsável pela identificação dos novos nodos. As sub-árvores que contêm blocos de dados, de modo geral, têm conteúdo e formatação semelhantes e, por consequência, pontuações próximas. Desta forma, o Algoritmo 3.4 considera os nodos com pontuação 25% menor ou maior que a média calculada no Algoritmo 3.3 como nodos que contêm blocos de dados.

Com este último passo, algumas sub-árvores de diferentes classes de equivalência são adicionadas à lista de sub-árvores que contêm blocos de dados. A árvore iniciada a

partir do terceiro nodo *C* na Figura 3.1 é um exemplo de sub-árvore adicionada à lista neste passo. Esta sub-árvore iniciada no nodo *C* é ignorada nos passos anteriores por não ser classificada com a mesma classe de equivalência das sub-árvores irmãs.

Entrada: F uma lista dos filhos do nodo previamente definido.
Entrada: M é a média da pontuação dos nodos raízes das sub-árvores que contêm blocos de dados.
Saída: L é uma lista contendo os nodos raízes das sub-árvores que contêm registros.

```
1  início
2  |  P ← M * 0.85%;
3  |  T ← M * 1.25%;
4  |  para cada nodo em F faça
5  |  |  pontuação_do_nodo = pontuaçãoDoNodo(nodo);
6  |  |  se P < pontuação_do_nodo < T então
7  |  |  |  adiciona o nodo em L;
8  |  |  fim
9  |  fim
10 fim
```

Algoritmo 3.4: Extração de Blocos de Dados Adicionais.

3.4 Identificação de Padrões no Conteúdo de Blocos de Dados

Na seção anterior, a lista de sub-árvores que contêm os blocos de dados é formada. A partir de cada bloco de dados é possível extrair os valores dos atributos, que juntos formam registros. Esta extração é possível utilizando uma expressão regular capaz de encontrar todos os blocos de dados identificados como de interesse no Web site. O Algoritmo de Alinhamento de Múltiplas Sequências [Gusfield, 1997] é usado na geração desta expressão regular. Os blocos de dados previamente identificados são segmentados e as informações comuns a todos eles são alinhadas.

A segmentação dos blocos de dados é feita em comentários HTML, *tags* HTML, símbolos HTML, datas, números, endereços ou URLs, endereços de e-mail, símbolos de moedas, pontuações e palavras em geral.

A Figura 3.8 mostra dois blocos de dados do Web site *monster.com*, o qual é parcialmente mostrado na Figura 3.9. Ambos os blocos são formados pela mesma sequência de texto, diferenciando-se apenas pela informação em negrito. De acordo com as definições

apresentadas na Seção 3.1, os valores dos atributos são encontrados em sequências de texto que não se repetem em todos os blocos de dados e são delimitados por sequências comuns a todos os blocos, as chamadas sequências de valores. Então, as informações em negrito contêm os valores dos atributos a serem identificados.

1. <TR><TD><FONT FACE="Verdana">**Jun 8**</FONT></TD><TD><FONT FACE="Verdana"> **US-TN-Nashville** </FONT></TD><TD><FONT FACE="Verdana"> **Programmer Analyst** </FONT></TD><TD><FONT FACE="Verdana"> **OAO** </FONT></TD></TR>

2. <TR><TD><FONT FACE="Verdana">**Jun 7**</FONT></TD><TD><FONT FACE="Verdana"> **US-IL-Chicago** </FONT></TD><TD><FONT FACE="Verdana"> **OpenStep Opportunity** </FONT></TD><TD><FONT FACE="Verdana"> **Technisource** </FONT></TD></TR>

Figura 3.8: Dois blocos de dados do Web site *monster.com*

Figura 3.9: Trechos de uma página Web do site *monster.com*

Algoritmo de Alinhamento de Múltiplas Sequências

O Algoritmo de Alinhamento de Múltiplas Sequências [Gusfield, 1997] é uma generalização do algoritmo de Alinhamento de Duas Sequências [Needleman e Wunsch, 1970], o qual é originalmente aplicado na descoberta de regiões similares entre duas cadeias de proteínas ou nucleotídeos.

O alinhamento de duas sequências consiste na inserção de *gaps* em qualquer posição das sequências respeitando 3 regras:

1. um elemento pode ser alinhado com um outro elemento semelhante ou com um *gap*;

2. dois *gaps* não podem ser alinhados;

3. o comprimento de ambas as sequências devem ser iguais após o alinhamento;

A Tabela 3.1 sumariza um exemplo de aplicação do Algoritmo de Alinhamento de Duas Sequências. A sequências 1 (CNERSKAFSCPS) e 2 (CNQCGKAFAQHS) são alinhadas e o resultado (CN—KAF—S) é apresentado. Os *gaps* são representados por hifens.

Sequência 1	C	N	E	R	S	K	A	F	S	C	P	S
Sequência 2	C	N	Q	C	G	K	A	F	A	Q	H	S
Resultado	**C**	**N**	**-**	**-**	**-**	**K**	**A**	**F**	**-**	**-**	**-**	**S**

Tabela 3.1: Exemplo de alinhamento de duas sequências genéricas

Com pequenas alterações, é possível adaptar o Algoritmo de Alinhamento de Duas Sequências para trabalhar com sequências de textos segmentados. Esta alteração é necessária no processamento de alinhamento do conteúdo de blocos de dados. A Tabela 3.2 apresenta resumidamente o alinhamento dos blocos de dados de *monster.com* apresentados na Figura 3.8.

<TD>	Jun 8	</TD><TD>	US-TN-Nashville	</TD><A href="...">	Programmer Analyst	</A>	...
<TD>	Jun 7	</TD><TD>	US-IL-Chicago	</TD><A href="...">	OpenStep Opportunity	</A>	...
<TD>	**-**	**</TD><TD>**	**-**	**</TD><A href="...">**	**-**	**</A>**	**...**

Tabela 3.2: Alinhamento de sequências de dois blocos de dados do Web site *monster.com*

Como dito, o Algoritmo de Alinhamento de Múltiplas Sequências é uma generalização do Algoritmo de Alinhamento de Duas Sequências. O objetivo é alinhar elementos similares de todas as sequências envolvidas, adicionando *gaps* nas posições onde os elementos são diferentes. Como demonstração, o resultados do alinhamento das quatro sequências (1) ATGCGGT, (2) AGCCGT, (3) TGCGT e (4) ATCGGT são mostradas na Tabela 3.3.

Durante o processo de alinhamento, os *gaps* adicionados são separados por segmentos de texto comuns a todas as sequências. Esta propriedade é utilizada pelo MAIt

Sequência 1	A	T	G	-	C	G	G	T
Sequência 2	A	-	G	C	C	G	-	T
Sequência 3	A	T	G	C	C	G	G	T
Sequência 4	A	T	G	-	C	G	G	T
Resultado	**A**	**-**	**G**	**-**	**C**	**G**	**-**	**T**

Tabela 3.3: Alinhamento de quatro sequências genéricas

na identificação dos valores dos atributos. Os segmentos comuns a todos os blocos são alinhados e os *gaps* formados representam os termos que diferenciam os blocos entre si. Então, como dito na Seção 3.1, os *gaps* adicionados nas sequências representam as sequências de valores a serem identificadas.

Geração da Expressão Regular

O resultado do alinhamento dos blocos de dados é utilizado na geração da expressão regular capaz de reconhecer os blocos de dados alinhados. Entretanto, esta expressão é capaz de reconhecer, também, outros blocos de dados não identificados durante o processo de alinhamento de árvores, já que os blocos de dados estão contidos em sub-árvores com estruturas padronizadas, mas não necessariamente idênticas e, por isso, não identificadas durante o alinhamento.

A Figura 3.8 mostra dois blocos de dados extraídos de *monster.com*. Os segmentos de texto em negrito contêm os valores de atributos a serem extraídos. Eles são delimitados por segmentos comuns aos dois blocos de dados, como esperado. No processo de alinhamento de sequências, estes segmentos são transformados em *gaps*, já que não se repetem em todos os blocos.

A geração da expressão regular se dá a partir do resultado do alinhamento dos blocos de dados. O primeiro objetivo da expressão regular é identificar outros blocos de dados do mesmo Web site, por isso, esta preserva as informações constantes em todos os blocos de dados utilizados no alinhamento. O segundo objetivo é identificar campos de texto candidatos a conter os valores dos atributos, os segmentos de valores. Como estes coincidem com os *gaps* do padrão textual gerado no processo de alinhamento de sequências, os *gaps* são transformados em grupos diferenciados na expressão regular. No passo final da criação da expressão regular estes grupos, quando adjacentes, são aglutinados em um

único grupo.

A Tabela 3.4 mostra a expressão regular gerada a partir das sequências da Tabela 3.1. Na terceira linha, é mostrado o resultado do alinhamento de múltiplas sequências, com os segmentos comuns a todas as sequências. Na quarta linha, os *gaps* da terceira linha são transformados em grupos diferenciados do restante das informações constantes aos blocos. E na última linha, a expressão regular é formada com a aglutinação destes grupos, quando adjacentes.

Sequência 1	C	N	E	R	S	K	A	F	S	C	P	S
Sequência 2	C	N	Q	C	G	K	A	F	A	Q	H	S
Resultado	C	N	-	-	-	K	A	F	-	-	-	S
Expressão regular parcial	C	N	(.*)	(.*)	(.*)	K	A	F	(.*)	(.*)	(.*)	S
Expressão regular final	**CN(.*)KAF(.*)S**											

Tabela 3.4: Exemplo de alinhamento de sequências e geração da expressão regular.

O mesmo processo se aplica à criação da expressão regular usada para extrair os blocos de dados e suas respectivas sequências de valores de atributos. Na Figura 3.10 é representada a expressão regular gerada através do alinhamento dos blocos de dados de *monster.com*.

<TR><TD><FONT FACE="Verdana"> **(.*)** </FONT></TD><TD><FONT FACE="Verdana"> **(.*)** </FONT></TD><TD><FONT FACE="Verdana"> **(.*)** </FONT></TD><TD><FONT FACE="Verdana"> **(.*)** </FONT></TD></TR>

Figura 3.10: Expressão regular criada a partir dos blocos de dados de *monster.com*

Desta forma, utilizando alinhamento de árvores e de múltiplas sequências textuais é possível criar uma expressão regular capaz de extrair blocos de dados e sequências de texto contendo os valores de seus atributos. Na próxima seção, os valores dos atributos são identificados nestas sequências.

3.5 Extração de Valores de Atributos e Registros

Usando a expressão regular criada no passo anterior, é possível extrair os blocos de dados de interesse de uma página Web. A mesma expressão é capaz de identificar campos

de texto do bloco de dados que, possivelmente, contêm valores de atributos a serem extraídos.

O passo final do método MAIt consiste na identificação dos registros representando os objetos da página de interesse. Registros são caracterizados e diferenciados pelos valores de seus atributos. O método MAIt identifica os registros através dos valores de seus atributos.

Como dito, os valores dos atributos são contidos nas sequências textuais que diferenciam os blocos de dados. Estas sequências são identificadas pela expressão regular gerada previamente. A heurística para localização dos valores de atributos nestes campos de texto considera o tamanho do campo de texto e o posicionamento do mesmo e dos valores encontrados entre eles.

A essência da heurística consiste em determinar a quantidade de atributos contidos em sequências de valores equivalentes. Duas sequências de valores são consideradas equivalentes se são igualmente posicionadas em relação a elementos comuns a todos os registros. Por exemplo, as sequências de valores da primeira coluna da tabela de empregos de *monster.com* na Figura 3.9 são equivalentes. Estes campos de texto são posicionados entre as mesmas estruturas HTML em todos os registros de emprego.

Sequências de valores equivalentes contêm os valores da mesma quantidade de atributos. Porém, alguns destes atributos podem não ter valor em alguns dos registros. A quantidade de atributos contidas em uma sequência de texto é definida pelo número de divisões a maioria das sequências equivalentes podem sofrer. Aquelas contidas na segunda coluna da Figura 3.9 podem ser divididas em três, por exemplo. Estas sequências são relacionadas à localização e contêm os atributos país, estado e cidade em todos os registros. Em dois dos registros o atributo cidade não possui valor mas, na maioria dos casos, as sequências podem ser dividas em três partes. Assim, a quantidade de atributos por sequência de texto equivalente é definida.

Na Figura 3.11 existem dois registros: o primeiro do Web site *amercoll.edu* e o segundo de *monster.com*. Os registros de *amercoll.edu* contêm três atributos: um título,

uma URL e um resumo da página representada neste registro. Os registros de *monster.com* contêm seis atributos: uma data, um país, um estado, uma cidade, um título de emprego e uma empresa. Este cenário será utilizado no exemplo a seguir.

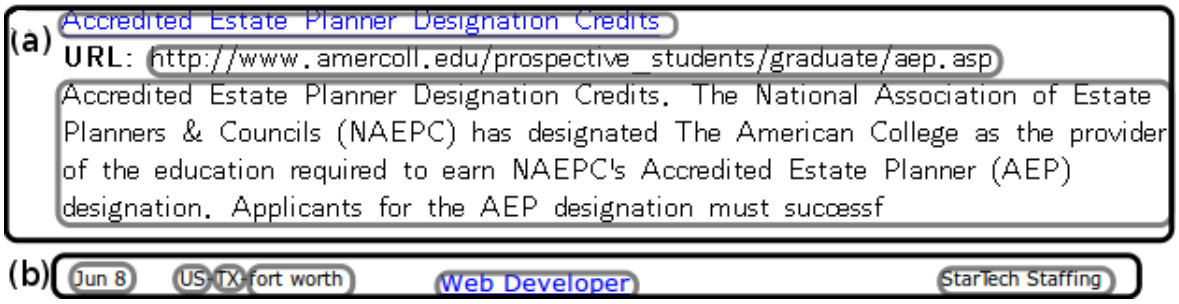

Figura 3.11: Exemplos de registros dos Web sites *amercoll.edu* (a) e *monster.com* (b) apresentando diferentes formatos de atributos

O primeiro passo na definição da quantidade de partes em que uma sequência textual pode ser dividida é a verificação do tamanho da sequência de valores. Como dito na Seção 3.1, sequências com mais de 60 caracteres não devem ser divididas. O segundo passo é aplicado a sequências mais curtas e consiste em dividi-las em delimitadores comumente encontrados em sequências de texto. Além disso, datas, números, endereços ou URLs, endereços de e-mail e textos em geral separados por estes delimitadores são possíveis valores de atributos.

Nos registros de *monster.com*, na Figura 3.9, as sequências de valores relativas à localização são divisíveis em três partes. Estes campos de texto possuem menos de 60 caracteres e possuem hifens que são divisores comumente encontrados em textos. Então, “US-TN-Nashville” e “US-CA-Sacramento” são divididos e alinhados como valores de atributos “US” e “US”, “TN” e “CA” e “Nashville” e “Sacramento”. Os campos representando títulos de emprego e empresa não são divididos, já que na maioria dos casos não são verificadas ocorrências de datas, números, endereços ou URLs ou endereços de e-mail.

Os registros de *amercoll.edu*, como os mostrados na Figura 3.11(a), são submetidos ao mesmo processo. A URL do segundo atributo não é dividida, por ser um dos tipos especiais comumente encontrados em valores de atributos. O mesmo ocorre com a descrição da página representada neste registro. Por conter mais de 60 caracteres, este

campo de texto não é dividido.

Com a identificação dos valores dos atributos em um bloco de dados, é possível extrair o registro que os contém. Cada bloco de dados contém os valores de atributos de um único registro, assim, para cada bloco, os valores dos atributos são alinhados e os registros identificados.

Dessa forma, com a identificação dos registros e os valores de seus atributos, chega ao fim o processo de extração proposto no MAIt. O Algoritmo de Alinhamento de Árvores é utilizado na identificação de exemplos de blocos de dados da página de interesse. Estes exemplos são usados na geração de uma expressão regular capaz de identificar os blocos de dados do site a que a página pertence. A expressão também é capaz de identificar sequências textuais contendo os valores dos atributos a serem extraídos. Estas sequências são divididas em delimitadores comumente encontrados em textos e os valores dos atributos identificados. Os registros são, então, formados pelo conjunto dos valores de atributos encontrados em cada bloco de dados.

O Capítulo 4 detalha os experimentos realizados e as avaliações de eficácia do método MAIt.

Capítulo 4

Experimentos

Este capítulo descreve os resultados de experimentos realizados para avaliar a eficácia do método de extração proposto nos capítulos anteriores. A metodologia para avaliação, as coleções de Web sites e os resultados alcançados também são apresentados.

Os experimentos consistem em identificar os blocos de dados, os registros e os valores de seus atributos. Para efeitos comparativos, além do nosso método, os mesmos experimentos são executados utilizando o método ViNTs apresentado em [Zhao et al., 2005] e brevemente descrito no Capítulo 2. Ambos os métodos são avaliados com respeito à extração dos blocos de dados, dos valores dos atributos e dos registros. Os experimentos utilizando o método ViNTs foram executados na ferramenta disponível na página dos autores[1].

Para execução dos experimentos foi criada uma coleção de páginas Web contendo dados a serem extraídos. Metade da coleção é composta de páginas geradas a partir de consultas em máquinas de busca e foram originalmente utilizadas nos experimentos de avaliação do método ViNTs, em [Zhao et al., 2005]. A outra metade é composta por páginas provenientes de domínios variados, incluindo sites de músicas, filmes, livros, remédios, vinhos e empregos e foram utilizadas nos experimentos de avaliação dos métodos propostos por [Pereira e Silva, 2006] e por [Crescenzi et al., 2001]. Desta forma, esperamos que nossos experimentos não beneficiem ou prejudiquem a avaliação dos métodos

[1] http://www.data.binghamton.edu:8080/vints/

em questão. Na próxima seção, serão dados mais detalhes sobre a coleção de dados.

As métricas para avaliação dos experimentos são as difundidas precisão e revocação [Baeza-Yates e Ribeiro-Neto, 1999]. A primeira mensura a quantidade de respostas corretas em relação ao total de respostas retornadas e é representada pela fórmula da Equação 4.1. Já a Revocação é a quantidade de respostas corretas em relação ao total de respostas esperadas ou relevantes, cuja fórmula está representada na Equação 4.2. A aplicação destas métricas na avaliação do problema proposto será detalhada nas próximas seções.

$$Precisao = \frac{|\{respostas\ relevantes\} \cap \{respostas\ retornadas\}|}{|\{respostas\ retornadas\}|} \tag{4.1}$$

$$Revocacao = \frac{|\{respostas\ relevantes\} \cap \{respostas\ retornadas\}|}{|\{respostas\ relevantes\}|} \tag{4.2}$$

4.1 Bases Utilizadas

As páginas Web utilizadas nos experimentos são compostas por objetos implícitos cuja estrutura apresenta um certo grau de regularidade. Elas fazem parte de 32 diferentes Web sites, sendo que 16, doravante chamadas de *coleção Search*, foram também utilizadas originalmente nos experimentos para avaliação do ViNTs [Zhao et al., 2005]. As 16 restantes, doravante chamadas de *coleção Mixed*, são páginas representativas já utilizadas nos experimentos dos métodos de extração propostos por [Pereira e Silva, 2006] e por [Crescenzi et al., 2001].

Ao total, 15383 valores de atributos dos 3402 objetos de 128 páginas de diferentes domínios devem ser extraídos. As Tabela 4.1 e 4.2 listam todas as bases utilizadas e o número de registros e atributos a serem extraídos de cada uma delas. As tabelas também mostram os números mínimos e máximos de atributos em cada base, por exemplo: na base *allgame.com*, existem 150 registros com no mínimo 3 atributos e no máximo 5, em um total de 623 valores de atributos.

Web site	Registros	Atributos		Valores
		Mínimo	Máximo	
allgame.com	150	3	5	623
allmovie.com	393	3	3	1179
allmovie.com (2)	400	3	3	1200
allmusic.com	125	3	3	375
allpolitics.com	150	2	2	300
amazon.com	75	9	12	862
amazon.com (2)	36	5	10	306
cdnow.com	90	4	4	360
imdb.com	170	4	4	680
monster.com	150	4	6	869
ncbi.nlm.nih.gov (PubMed)	60	7	8	440
terra.com.br/loterias/loteca	42	4	4	168
vitacost.com	259	5	8	1842
watchzone.com	111	6	6	666
wine.com	30	5	6	171
yahoo.com/search/people	30	3	3	90
TOTAL	**2271**	-		**10131**

Tabela 4.1: Coleção Mixed de Web sites a serem extraídos

A Tabela 4.1 apresenta informações sobre os Web sites da coleção Mixed. De cada Web site foram aleatoriamente coletadas 3 páginas, contendo um total de 10131 valores de atributos organizados em 2271 registros. Os 16 Web sites desta coleção possuem estruturas diversificadas como descrito a seguir:

- *allgame.com*, *allmovie.com*, *allmovie.com (2)*, *allmusic.com*, *cdnow.com*, *imdb.com*, *monster.com*, *terra.com.br/loterias/loteca* e *yahoo.com/search/people* possuem registros organizados em formas de tabelas.
- *allpolitics.com* possui uma lista enumerada com textos curtos e data de publicação.
- *amazon.com*, *amazon.com (2)*, *vitacost.com*, *watchzone.com* e *wine.com* possuem registros em forma convencional de produtos em sites de venda.
- *PubMed* tem formatação de resultados de máquinas de busca convencional, com o título da página de destino e um pequeno texto a descrevendo.

A Tabela 4.2 apresenta informações sobre os Web sites da coleção Search. Como dito anteriormente, estes Web sites também foram utilizados nos experimentos executa-

Web site	Registros	Atributos		Valores
		Mínimo	Máximo	
alltheweb.com	41	4	5	200
amercoll.edu	49	3	3	152
american.edu	50	6	6	301
atlanticuc.edu	50	5	5	260
atu.edu	116	5	6	694
bu.edu	125	6	6	750
campbellsville.edu	50	3	3	151
clemson.edu	50	6	6	300
csuchico.edu	50	3	6	254
csudh.edu	50	3	6	255
fairfield.edu	50	5	5	250
franklin.edu	125	2	2	250
harvard.edu	50	6	6	310
metacrawler.com	100	3	3	300
mit.edu	75	6	7	524
search.excite.com	100	3	3	300
TOTAL	**1131**	-		**5252**

Tabela 4.2: Coleções Search de Web sites a serem extraídos

dos com o método ViNTs [Zhao et al., 2005]. As bases desta coleção foram obtidas do site dos autores[2]. Para cada Web site da coleção são disponibilizadas 5 páginas, totalizando 5252 valores de atributos organizados em 1131 registros. Os 16 Web sites desta coleção são todos eles sites de busca, de modo que suas páginas contêm os resultados retornados a partir de consultas de um usuário. Desta forma, os registros da coleção Search contêm o título da página sugerida, uma amostra do seu conteúdo textual (*snippets*) e algumas informações opcionais, como data de alteração e tamanho da página. É importante salientar que, apesar de conterem basicamente os mesmos atributos e seguirem a mesma estrutura, cada um dos 16 Web sites geram páginas de resposta com formatações diferentes entre si.

Como será descrito na próxima seção, para a avaliação dos métodos de extração é necessário possuir um conjunto resposta constituído, para cada página, dos seus valores de atributos e dos registros formados por eles, assim como os blocos de dados que contêm essas informações. Como este conjunto resposta não foi disponibilizado pelos trabalhos previamente publicados na literatura, os 15383 valores de atributos, 3402 blocos de da-

[2]http://idke.ruc.edu.cn/news/2008/dataset.htm

dos e 128 páginas foram manualmente identificados para que pudéssemos usá-los como gabaritos em nossa avaliação.

O uso combinado de páginas das coleções Search e Mixed se mostrou uma forma coerente de avaliar os métodos MAIt e ViNTs, como será descrito nas próximas seções. A aplicação das métricas de precisão e revocação na avaliação dos experimentos será descrita na seção que se segue.

4.2 Métricas de avaliação

Como previamente mencionado, utilizamos as métricas de precisão e revocação para avaliar a corretude ou acurácia dos métodos utilizados neste experimento. Os conceitos de precisão e revocação são aplicados na medição da quantidade de respostas corretas em relação ao total de respostas encontradas e esperadas, respectivamente. A fórmula matemática geral para o cálculo da precisão está representada na Equação 4.1 e da revocação na Equação 4.2. Essas fórmulas foram aplicadas em nossos experimentos para que pudéssemos comparar os métodos de extração em questão, em relação a 3 pontos:

1. identificação correta dos blocos de dados que contêm os registros e seus atributos.

2. identificação correta dos registros, dados os atributos encontrados.

3. identificação correta dos atributos.

De acordo com os conceitos, para cada ponto de avaliação é necessário definir as respostas esperadas e compará-las às respostas encontradas. Desta forma, os valores, registros e blocos de dados manualmente identificados previamente são considerados as respostas esperadas em cada um dos pontos de avaliação. Por exemplo, para avaliar o quão correto foi a identificação dos blocos de dados, comparamos os valores do conjunto resposta com os valores encontrados pelo extrator a ser avaliado, formando o conjunto de blocos de dados identificados corretamente. De posse dos 3 conjuntos - (1) blocos esperados, (2) blocos encontrados e (3) blocos corretamente encontrados - é possível calcular

a precisão e a revocação do extrator em questão em relação à identificação de blocos de dados.

Nas próximas seções, apresentamos os resultados da avaliação dos métodos ViNTs e MAIt em relação à extração de blocos de dados, registros e valores de atributos.

4.3 Avaliação da extração de blocos de dados

Como explicado na seção anterior, de posse do conjunto de respostas esperadas na extração de blocos de dados e do conjunto de blocos encontrados, podemos calcular a precisão e a revocação de cada um dos métodos extratores em relação à identificação de blocos de dados.

Avaliamos os métodos ViNTs e MAIt em relação à extração de blocos de dados para cada Web site e de modo geral. Cada Web site tem seus conjuntos de respostas ou blocos esperadas e de blocos encontrados, o que torna viável a avaliação por site. Para a avaliação geral, consideramos a união dos conjuntos de respostas esperadas e encontradas de todos os sites.

Como descrito no Capítulo 3, o processo de geração de extratores do MAIt consiste em fases sucessivas. Durante este processo, é aplicada a técnica de alinhamento de árvores na identificação de um conjunto inicial de blocos de dados. Estes são utilizados como exemplos para geração de uma expressão regular capaz de identificar o conjunto final de blocos de dados. Quanto maior for o número de blocos de dados usados como exemplos, maior será a quantidade de blocos de dados identificados pela expressão regular gerada. Isto ocorre porque a expressão regular é capaz de identificar, além dos blocos usados como exemplo para sua geração, outros blocos similares a eles.

Considerando o conjunto de páginas de interesse e seus blocos de dados, o método MAIt será avaliado quanto à identificação destes blocos de três formas:

1. MAIt 1: serão considerados apenas os blocos de dados encontrados pela técnica de alinhamento de árvores, sem geração e uso de uma expressão regular.

2. MAIt 2: serão considerados os blocos de dados identificados por uma expressão regular gerada utilizando os blocos de dados de apenas uma das páginas de interesse.

3. MAIt 3: serão considerados os blocos de dados identificados por uma expressão regular gerada utilizando os blocos de dados de todas as páginas de interesse.

Os resultados alcançados na avaliação por Web site do método ViNTs e do método MAIt nas três formas acima são mostrados nas Tabelas 4.3 e 4.4. A primeira tabela descreve os resultados obtidos usando a coleção Mixed e a segunda os resultados obtidos usando a coleção Search. Com os experimentos, foi possível constatar que o método ViNTs teve um melhor desempenho ao extrair os blocos das bases utilizadas em seus experimentos em [Zhao et al., 2005] do que com as bases da coleção Mixed. Este fato corrobora com nossa metodologia de uso de páginas diversificadas nos experimentos, incluindo as coleções Mixed e Search.

A avaliação geral está na Tabela 4.5. Nesta, mostramos apenas o resultado obtido em MAIt 3, que se mostrou mais eficiente na avaliação por site de acordo com as Tabelas 4.3 e 4.4. Calculamos, também, o ganho obtido por MAIt em relação ao método ViNTs no âmbito da extração de blocos de dados de modo geral. Os valores obtidos neste cálculo mostram que nosso método teve um ganho de 40,00% na precisão e 67,05% na revocação em relação ao ViNTs, ou seja, encontramos mais blocos de dados do conjunto de respostas esperadas.

Nas seções seguintes são descritos os resultados das avaliações das extrações dos registros e dos valores de seus atributos.

4.4 Avaliação da extração de registros

A avaliação da extração de registros é medida pela eficiência do extrator em identificar todos os valores atributos contidos nos blocos de dados. As Tabelas 4.6 e 4.7 sumarizam os resultados obtidos na avaliação dos métodos extratores em relação à identificação de registros.

	MAIt 1		MAIt 2		MAIt 3		ViNTs	
Coleção Mixed	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
allgame.com	1.00	0.96	0.97	0.95	1.00	1.00	0.97	0.97
allmovie.com	1.00	0.99	1.00	1.00	1.00	1.00	0.95	0.95
allmovie.com (2)	1.00	0.74	0.90	0.68	0.90	0.68	0.97	0.97
allmusic.com	1.00	0.94	0.90	0.90	0.90	0.90	0.99	0.99
allpolitics.com	1.00	0.93	1.00	1.00	1.00	1.00	0.13	0.13
amazon.com	1.00	0.92	0.96	0.92	1.00	1.00	0.87	0.87
amazon.com (2)	0.71	0.47	0.63	0.69	0.61	0.69	0.97	0.97
cdnow.com	1.00	0.97	1.00	1.00	1.00	1.00	1.00	1.00
imdb.com	1.00	0.84	0.88	0.88	0.88	0.88	0.91	0.81
monster.com	1.00	0.96	1.00	1.00	1.00	1.00	0.99	0.99
ncbi.nlm.nih.gov (PubMed)	1.00	0.95	0.93	0.42	1.00	1.00	1.00	1.00
terra.com.br/loterias/loteca	1.00	0.86	1.00	1.00	1.00	1.00	0.00	0.00
vitacost.com	1.00	0.98	0.98	0.98	0.99	0.99	0.00	0.99
watchzone.com	1.00	0.95	0.83	0.59	1.00	1.00	0.00	0.00
wine.com	1.00	0.87	1.00	1.00	1.00	1.00	0.96	0.83
yahoo.com/search/people	1.00	0.80	1.00	1.00	1.00	1.00	0.00	0.00

Tabela 4.3: Resultado da avaliação da extração de blocos de dados da coleção Mixed

	MAIt 1		MAIt 2		MAIt 3		ViNTs	
Coleção Search	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
alltheweb.com	1.00	0.64	0.67	0.52	0.90	0.90	0.78	0.95
amercoll.edu	1.00	0.90	0.89	0.84	0.89	0.84	0.90	0.90
american.edu	1.00	0.70	1.00	1.00	1.00	1.00	0.84	0.84
atlanticuc.edu	1.00	0.80	0.72	0.26	1.00	1.00	0.92	0.92
atu.edu	1.00	0.87	1.00	1.00	1.00	1.00	0.97	0.97
bu.edu	1.00	0.88	1.00	1.00	1.00	1.00	0.99	0.99
campbellsville.edu	0.98	0.84	0.77	0.72	0.84	0.84	0.90	0.90
clemson.edu	1.00	0.73	0.96	1.00	0.96	1.00	1.00	1.00
csuchico.edu	1.00	0.62	0.89	0.80	0.89	0.80	1.00	1.00
csudh.edu	1.00	0.68	0.91	0.84	1.00	1.00	0.98	0.98
fairfield.edu	1.00	0.80	1.00	1.00	1.00	1.00	1.00	1.00
franklin.edu	1.00	0.90	1.00	1.00	1.00	1.00	1.00	1.00
harvard.edu	1.00	0.70	1.00	1.00	1.00	1.00	1.00	1.00
metacrawler.com	1.00	0.85	1.00	1.00	1.00	1.00	1.00	1.00
mit.edu	1.00	0.80	0.80	0.60	1.00	1.00	0.88	0.88
search.excite.com	0.98	0.83	0.98	0.98	0.98	0.98	1.00	1.00

Tabela 4.4: Resultado da avaliação da extração dos blocos de dados da coleção Search

	MAIt		ViNTs		Ganhos	
	P	**R**	**P**	**R**	**P**	**R**
Resultado Geral	0.90	0.88	0.54	0.29	40.00%	67.05%

Tabela 4.5: Resultado geral da avaliação da extração dos blocos de dados

O cálculo da precisão e da revocação da extração de um registro considera, respectivamente, a média aritmética da precisão e da revocação da extração dos valores dos atributos deste registro. Em outras palavras, a precisão da extração de um registro será dada pela média dos valores de precisão da extração de todos os seus atributos. A revocação por registro é calculada de forma similar.

Para calcular os valores da precisão por Web site, como os demonstrados nas Tabelas 4.6 e 4.7, faz-se a média aritmética da precisão de todos os registros do site em questão. O cálculo da revocação por Web site se dá de forma semelhante.

Como esperado, houve grande diferença no resultado da avaliação do método ViNTs quanto à extração dos registros da coleção Search e à extração dos registros da coleção Mixed. Entretanto, de acordo com os valores de precisão e revocação calculados, nosso método se mostrou mais adequado na extração de registros de modo geral.

	MAIt		**ViNTs**	
Coleção Mixed	**P**	**R**	**P**	**R**
allgame.com	0.62	0.62	0.00	0.00
allmovie.com	0.98	0.98	0.00	0.00
allmovie.com (2)	0.97	0.97	0.00	0.00
allmusic.com	0.99	0.99	0.00	0.00
allpolitics.com	0.74	0.74	0.00	0.00
amazon.com	0.46	0.53	0.32	0.20
amazon.com (2)	0.48	0.38	0.25	0.13
cdnow.com	1.00	1.00	0.00	0.00
imdb.com	0.96	0.96	0.00	0.00
monster.com	0.94	0.95	0.00	0.00
ncbi.nlm.nih.gov (PubMed)	0.45	0.45	0.25	0.12
terra.com.br/loterias/loteca	0.91	0.91	0.00	0.00
vitacost.com	0.65	0.65	0.00	0.00
watchzone.com	1.00	1.00	0.13	0.03
wine.com	0.67	0.33	0.32	0.18
yahoo.com/search/people	1.00	1.00	0.00	0.00

Tabela 4.6: Resultado da avaliação da extração de registros dos Web sites da coleção Mixed, de acordo com a identificação de seus atributos

4.5 Avaliação da extração de valores de atributos

Assim como na avaliação da extração dos blocos de dados, na avaliação da extração dos valores de atributos foram calculados os valores da precisão e da revocação por Web site

	MAIt		ViNTs		Ganhos	
Coleção Search	**P**	**R**	**P**	**R**	**P**	**R**
alltheweb.com	0.80	0.80	0.62	0.53	22.50%	33.75%
amercoll.edu	0.89	0.89	0.97	0.97	-8.99%	-8.99%
american.edu	0.67	0.58	0.57	0.67	14.93%	-15.52%
atlanticuc.edu	0.72	0.58	0.62	0.40	13.89%	31.03%
atu.edu	0.98	0.98	0.68	0.80	30.61%	18.37%
bu.edu	0.98	0.98	0.77	0.77	21.43%	21.43%
campbellsville.edu	0.85	0.85	0.95	0.95	-11.76%	-11.76%
clemson.edu	1.00	1.00	0.65	0.65	35.00%	35.00%
csuchico.edu	0.52	0.45	0.26	0.17	50.00%	62.22%
csudh.edu	0.75	0.64	0.28	0.19	62.67%	70.31%
fairfield.edu	0.56	0.27	0.67	0.40	-19.64%	-48.15%
franklin.edu	1.00	1.00	1.00	1.00	0.00%	0.00%
harvard.edu	0.78	0.72	0.83	0.83	-6.41%	-15.28%
metacrawler.com	0.66	0.66	0.65	0.65	1.52%	1.52%
mit.edu	0.93	0.93	0.64	0.55	31.18%	40.86%
search.excite.com	0.50	0.50	0.62	0.62	-24.00%	-24.00%

Tabela 4.7: Resultado da avaliação da extração de registros dos Web sites da coleção Search, de acordo com a identificação de seus atributos

e de forma geral. Para o cálculo geral, consideramos todos os atributos de todos os Web sites, constituindo um conjunto de respostas esperadas com todos os valores de atributos de todos os sites a serem identificados e um conjunto de valores de atributos encontrados por cada um dos métodos avaliados.

A Tabela 4.8 mostra o resultado da avaliação geral da extração de valores de atributos, que é calculada sem considerar o Web site ou a página de origem de cada atributo. Nosso método extrator teve um ganho de 43,37% de precisão e 68,75% de revocação sobre o ViNTs, o que mostra que identificamos mais valores de atributos do conjunto de valores esperados.

	MAIt		ViNTs		Ganhos	
	P	**R**	**P**	**R**	**P**	**R**
Resultado Geral	0.83	0.80	0.47	0.25	43.37%	68.75%

Tabela 4.8: Resultado geral da avaliação da extração dos valores dos atributos

Devido à grande quantidade de informação, as tabelas detalhando a avaliação da extração de valores de atributos por Web site estão no Anexo A. Nesta avaliação calculamos as taxas de acerto para cada atributo de um dado site. Por exemplo, na Tabela A.1 é

mostrado o resultado obtido na extração da base *allgame.com*, onde é esperado que sejam identificados 5 atributos por registro. Para calcular a precisão e a revocação da extração do primeiro atributo, formam-se dois conjuntos: o primeiro contendo todos os valores esperados do primeiro atributo de todos os registros e o segundo com todos os valores do primeiro atributo encontrados pelo método de extração em avaliação. Com base nesses conjuntos é possível encontrar a quantidade de valores do primeiro atributo foram corretamente encontrados pelo extrator em avaliação e, assim, calcular as taxas de precisão e revocação. O mesmo procedimento é repetido para os demais atributos.

Na Tabela 4.9 são mostradas, para cada base da coleção Mixed, a média aritmética da precisão e da revocação da extração dos valores dos atributos. Na Tabela 4.10 são mostrados os valores das médias aritméticas da precisão e da revocação da extração dos valores dos atributos das bases da coleção Search.

	MAIt		**ViNTs**	
Coleção Mixed	**P**	**R**	**P**	**R**
allgame.com	0.63	0.63	0.00	0.00
allmovie.com	0.98	0.98	0.00	0.00
allmovie.com (2)	0.98	0.98	0.00	0.00
allmusic.com	1.00	1.00	0.00	0.00
allpolitics.com	0.77	0.77	0.00	0.00
amazon.com	0.74	0.56	0.16	0.19
amazon.com (2)	0.49	0.38	0.14	0.10
cdnow.com	1.00	1.00	0.00	0.00
imdb.com	0.98	0.98	0.00	0.00
monster.com	0.96	0.97	0.00	0.00
ncbi.nlm.nih.gov (PubMed)	0.54	0.48	0.13	0.13
terra.com.br/loterias/loteca	0.94	0.94	0.00	0.00
vitacost.com	0.74	0.66	0.00	0.00
watchzone.com	1.00	1.00	0.01	0.03
wine.com	0.50	0.33	0.31	0.14
yahoo.com/search/people	1.00	1.00	0.00	0.00

Tabela 4.9: Resultado da avaliação da extração de atributos dos Web sites da coleção Mixed

Com algumas exceções, nosso método se mostrou mais eficiente na identificação de atributos quando comparados ao ViNTs. Este último, principalmente na coleção Mixed, apresentou baixas precisão e revocação, assim como na extração dos blocos de dados descrita anteriormente.

	MAIt		ViNTs		Ganhos	
Coleção Search	**P**	**R**	**P**	**R**	**P**	**R**
alltheweb.com	0.94	0.94	0.81	0.94	13.82%	0.00%
amercoll.edu	0.97	0.91	0.99	0.99	-2.06%	-8.79%
american.edu	0.75	0.61	0.67	0.67	10.66%	-9.83%
atlanticuc.edu	0.59	0.59	0.44	0.40	25.42%	32.20%
atu.edu	0.99	0.99	0.74	0.80	25.25%	19.19%
bu.edu	1.00	1.00	0.80	0.80	20.00%	20.00%
campbellsville.edu	0.93	0.93	0.97	0.97	-4.30%	-4.30%
clemson.edu	1.00	1.00	0.71	0.71	29.00%	29.00%
csuchico.edu	0.58	0.51	0.10	0.17	82.75%	66.66%
csudh.edu	0.73	0.73	0.09	0.18	87.67%	75.34%
fairfield.edu	0.48	0.26	0.46	0.41	4.16%	-57.69%
franklin.edu	1.00	1.00	1.00	1.00	0.00%	0.00%
harvard.edu	0.94	0.78	0.83	0.83	11.70%	-6.41%
metacrawler.com	0.74	0.74	0.74	0.74	0.00%	0.00%
mit.edu	0.99	0.99	0.53	0.56	46.46%	43.43%
search.excite.com	0.35	0.51	0.71	0.71	-1.02%	-39.21%

Tabela 4.10: Resultado da avaliação da extração de atributos dos Web sites da coleção Search

4.6 Discussão dos resultados obtidos

De modo geral, o método MAIt apresentou melhor eficiência tanto na precisão quanto na revocação em relação ao método ViNTs em todos os experimentos realizados. Entretando, aspectos específicos de alguns Web sites tornaram a diferença de eficiência entre os dois métodos, ainda maior.

Durante o processo de extração de blocos de dados, o método ViNTs identificou elementos incorretos como sendo aqueles candidatos a conter os registros nos Web sites *terra.com.br/loterias/loteca* e *watchzone.com*. Nestes casos, o posicionamento dos elementos em posição de destaque na página e área ocupada pelos mesmos na mesma foram responsáveis por induzir o método ViNTs a considerá-los como blocos de dados. Em outros casos, como em *allpolitics.com* e *vitacost.com* os blocos de dados foram identificados como se contivessem elementos que não fazem parte dos registros. Isto ocorreu por estes elementos serem visualmente parte do registro mas, na verdade, não identificarem os mesmos. E em *yahoo.com/search/people*, nenhum resultado foi obtido.

No processo de extração de registros e de valores de atributos, o método ViNTs

se mostrou bastante ineficiente em vários casos. Alguns casos, como consequência da identificação incorreta dos blocos de dados, em outros, pela pequena diferenciação visual entre os valores dos atributos, como aqueles contidos em uma mesma sequência de texto. A incorreta identificação destes, tem como consequência a baixa eficiência na identificaçãos dos registros.

Capítulo 5

Conclusão e Trabalhos Futuros

Neste trabalho apresentamos o método MAIt - **M**ore **A**bout **It**, um gerador de extratores de dados de páginas ricas em dados. Ao contrário da maioria dos trabalhos anteriores, nosso método não necessita de interação humana e não é restrito à formatação e disposição dos dados nas páginas. Em nossa abordagem fazemos uso da padronização das estruturas do código HTML, das árvores DOM e do conteúdo textual dos registros para extrai-los e para identificar os valores de seus atributos. Primeiramente, o método MAIt utiliza alinhamento de árvores para identificar as sub-árvores da árvore DOM da página de interesse que contêm os registros. Em um segundo momento, as porções de código HTML dessas sub-árvores são processadas, de modo a se definir o padrão de seu conteúdo através de alinhamentos de múltiplas sequências de texto. O padrão é utilizado na criação de uma expressão regular capaz de identificar os registros e os campos contendo os valores dos atributos. Finalmente, os valores dos atributos são encontrados utilizando delimitadores e tipos de dados comumente encontrados em registros.

O método MAIt difere de outros métodos publicados na literatura por não restringir a origem das páginas de interesse. Os registros podem representar objetos disponíveis em catálogos de compras, listagens ou páginas retornadas em máquinas de busca, por exemplo.

Os experimentos realizados utilizando o método MAIt demonstram sua eficácia e aplicabilidade. O método foi avaliado em relação à identificação de blocos de código

HTML que contêm os registros e quanto à extração dos registros e dos valores de seus atributos. Obtivemos precisão de 83% e revocação de 80% na extração de valores de atributos. Estes valores significam um ganho na precisão de 43,37% e na revocação de 68,75%, em relação ao método ViNTs.

A versatilidade do método MAIt, em relação à origem das páginas ricas em dados, também foi verificada nos experimentos realizados. Nestes, utilizamos páginas compostas por registros de estilos visuais variados, incluindo tabelas, listagens, catálogos de compras e páginas de resultados de buscas convencionais. Com estes resultados, corroboramos a hipótese de que através da padronização da estrutura visual, textual e das árvores DOM das páginas ricas em dados é possível identificar registros e valores de atributos.

Como trabalhos futuros, pretendemos desenvolver uma ferramenta integrada com navegadores Web, de forma a tornar viável a extração das informações de páginas ricas em dados durante a navegação. Desta forma, poderemos difundir o método MAIt e permitir a personalização e utilização do mesmo em situações diversas.

Além disso, pretendemos utilizar os dados extraídos pelo MAIt em processos de rotulamento. Com a integração do extrator gerado por MAIt com um método de rotulamento é possível otimizar sistemas de análise de dados, máquinas de busca e de metabusca.

Referências Bibliográficas

[Baeza-Yates e Ribeiro-Neto, 1999] Baeza-Yates, R. A. e Ribeiro-Neto, B. (1999). *Modern Information Retrieval*.

[Crescenzi et al., 2001] Crescenzi, V., Mecca, G., Merialdo, P., Roma, U., Università, T., Università, B., e Tre, R. (2001). Roadrunner: Towards automatic data extraction from large web sites. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 109–118.

[Dalvi et al., 2009] Dalvi, N., Bohannon, P., e Sha, F. (2009). Robust web extraction: an approach based on a probabilistic tree-edit model. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, SIGMOD '09, pages 335–348.

[Gusfield, 1997] Gusfield, D. (1997). *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge Univ. Press.

[He et al., 2007] He, H., Meng, W., Zhao, H., e Yu, C. (2007). Annotating structured data of the deep web. In *Proceedings of the 23rd International Conference on Data Engineering*, pages 376–385.

[Laender et al., 2002] Laender, A. H. F., Ribeiro-Neto, B. A., da Silva, A. S., e Teixeira, J. S. (2002). A brief survey of web data extraction tools. *SIGMOD Record*, 31:84–93.

[Liu et al., 2003] Liu, B., Grossman, R., e Zhai, Y. (2003). Mining data records in web pages. In *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 601–606.

[Liu et al., 2010] Liu, W., Meng, X., e Meng, W. (2010). Vide: A vision-based approach for deep web data extraction. *IEEE Transactions on Knowledge and Data Engineering*, 22:447–460.

[Miao et al., 2009] Miao, G., Tatemura, J., Hsiung, W.-P., Sawires, A., e Moser, L. E. (2009). Extracting data records from the web using tag path clustering. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 981–990.

[Needleman e Wunsch, 1970] Needleman, S. B. e Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

[Pereira e Silva, 2006] Pereira, D. O. e Silva, A. S. (2006). Geração semi-automática de extratores de dados web considerando contextos fracos. Dissertação de Mestrado, Universidade Federal do Amazonas, Instituto de Ciências Exatas, Departamento de Ciência da Computação.

[Reis et al., 2004] Reis, D. C., Golgher, P. B., Silva, A. S., e Laender, A. F. (2004). Automatic web news extraction using tree edit distance. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 502–511.

[Selkow, 1977] Selkow, S. (1977). The tree-to-tree editing problem. *Information Processing Letters*, 6:184–186.

[Valiente, 2001] Valiente, G. (2001). An efficient bottom-up distance between trees. In *Proceedings of the 8th International Symposium of String Processing and Information Retrieval*, SPIRE '01, pages 212–219.

[Valiente, 2002] Valiente, G. (2002). Tree edit distance and common subtrees. *Research Report LSI-02-20-R*.

[Zhao et al., 2005] Zhao, H., Meng, W., Wu, Z., Raghavan, V., e Yu, C. (2005). Fully automatic wrapper generation for search engines. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 66–75.

Apêndice A

Experimentos

Base	**MAIt**		**ViNTs**	
allgame.com	**P**	**R**	**P**	**R**
1	0.86	0.86	0.00	0.00
2	1.00	1.00	0.00	0.00
3	0.00	0.00	0.00	0.00
4	0.34	0.34	0.00	0.00
5	0.97	0.97	0.00	0.00
Média	**0.63**	**0.63**	**0.00**	**0.00**

Tabela A.1: Resultado da avaliação da extração dos valores dos atributos da base *allgame.com*

Base	**MAIt**		**ViNTs**	
allmovie.com	**P**	**R**	**P**	**R**
1	1.00	1.00	0.00	0.00
2	0.97	0.97	0.00	0.00
3	0.97	0.97	0.00	0.00
Média	**0.98**	**0.98**	**0.00**	**0.00**

Tabela A.2: Resultado da avaliação da extração dos valores dos atributos da base *allmovie.com*

Base	**MAIt**		**ViNTs**	
allmovie.com (2)	**P**	**R**	**P**	**R**
1	1.00	1.00	0.00	0.00
2	0.99	0.99	0.00	0.00
3	0.94	0.94	0.00	0.00
Média	**0.98**	**0.98**	**0.00**	**0.00**

Tabela A.3: Resultado da avaliação da extração dos valores dos atributos da base *allmovie.com (2)*

Base	**MAIt**		**ViNTs**	
allmusic.com	**P**	**R**	**P**	**R**
1	0.99	0.99	0.00	0.00
2	1.00	1.00	0.00	0.00
3	1.00	1.00	0.00	0.00
Média	**1.00**	**1.00**	**0.00**	**0.00**

Tabela A.4: Resultado da avaliação da extração dos valores dos atributos da base *allmusic.com*

Base	**MAIt**		**ViNTs**	
allpolitics.com	**P**	**R**	**P**	**R**
1	0.81	0.81	0.00	0.00
2	0.73	0.73	0.00	0.00
Média	**0.77**	**0.77**	**0.00**	**0.00**

Tabela A.5: Resultado da avaliação da extração dos valores dos atributos da base *allpolitics.com*

Base	**MAIt**		**ViNTs**	
amazon.com	**P**	**R**	**P**	**R**
1	0.76	1.00	0.19	0.56
2	0.95	0.95	0.00	0.00
3	1.00	0.91	0.00	0.00
4	1.00	1.00	0.00	0.00
5	1.00	0.01	0.00	0.00
6	1.00	0.09	1.00	0.04
7	0.00	0.00	0.43	0.99
8	1.00	0.01	0.28	0.72
9	0.51	1.00	0.00	0.00
10	0.99	1.00	0.00	0.00
11	0.70	0.71	0.00	0.00
12	0.00	0.00	0.00	0.00
Média	**0.74**	**0.56**	**0.16**	**0.19**

Tabela A.6: Resultado da avaliação da extração dos valores dos atributos da base *amazon.com*

Base	**MAIt**		**ViNTs**	
amazon.com (2)	**P**	**R**	**P**	**R**
1	0.72	0.72	0.00	0.00
2	0.00	0.00	1.00	0.03
3	1.00	1.00	0.00	0.00
4	1.00	0.83	0.00	0.00
5	1.00	1.00	0.00	0.00
6	0.00	0.00	0.00	0.00
7	0.12	0.08	0.00	0.00
8	0.08	0.03	0.00	0.00
9	0.00	0.00	0.29	0.94
10	1.00	0.17	0.11	0.03
Média	**0.49**	**0.38**	**0.14**	**0.10**

Tabela A.7: Resultado da avaliação da extração dos valores dos atributos da base *amazon.com (2)*

Base	**MAIt**		**ViNTs**	
cdnow.com	**P**	**R**	**P**	**R**
1	1.00	1.00	0.00	0.00
2	1.00	1.00	0.00	0.00
3	1.00	1.00	0.00	0.00
4	1.00	1.00	0.00	0.00
Média	**1.00**	**1.00**	**0.00**	**0.00**

Tabela A.8: Resultado da avaliação da extração dos valores dos atributos da base *cdnow.com*

Base	**MAIt**		**ViNTs**	
imdb.com	**P**	**R**	**P**	**R**
1	1.00	1.00	0.00	0.00
2	0.92	0.92	0.00	0.00
3	1.00	1.00	0.00	0.00
4	1.00	1.00	0.00	0.00
Média	**0.98**	**0.98**	**0.00**	**0.00**

Tabela A.9: Resultado da avaliação da extração dos valores dos atributos da base *imdb.com*

Base	**MAIt**		**ViNTs**	
monster.com	**P**	**R**	**P**	**R**
1	1.00	1.00	0.00	0.00
2	1.00	1.00	0.00	0.00
3	0.98	0.98	0.00	0.00
4	0.85	0.86	0.00	0.00
5	1.00	1.00	0.00	0.00
6	0.95	0.96	0.00	0.00
Média	**0.96**	**0.97**	**0.00**	**0.00**

Tabela A.10: Resultado da avaliação da extração dos valores dos atributos da base *monster.com*

Base	**MAIt**		**ViNTs**	
ncbi.nlm.nih.gov (PubMed)	**P**	**R**	**P**	**R**
1	1.00	1.00	0.00	0.00
2	0.98	0.98	1.00	1.00
3	0.62	0.62	0.00	0.00
4	0.62	0.62	0.00	0.00
5	0.94	0.50	0.00	0.00
6	0.18	0.08	0.00	0.00
7	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00
Média	**0.54**	**0.48**	**0.13**	**0.13**

Tabela A.11: Resultado da avaliação da extração dos valores dos atributos da base *ncbi.nlm.nih.gov (PubMed)*

Base	**MAIt**		**ViNTs**	
terra.com.br/loterias/loteca	**P**	**R**	**P**	**R**
1	1.00	1.00	0.00	0.00
2	0.88	0.88	0.00	0.00
3	0.88	0.88	0.00	0.00
4	1.00	1.00	0.00	0.00
Média	**0.94**	**0.94**	**0.00**	**0.00**

Tabela A.12: Resultado da avaliação da extração dos valores dos atributos da base *terra.com.br/loterias/loteca*

Base	**MAIt**		**ViNTs**	
vitacost.com	**P**	**R**	**P**	**R**
1	0.69	0.68	0.00	0.00
2	0.46	0.42	0.00	0.00
3	0.27	0.28	0.00	0.00
4	1.00	0.01	0.00	0.00
5	0.50	0.98	0.00	0.00
6	0.96	0.94	0.00	0.00
7	1.00	0.98	0.00	0.00
8	1.00	0.98	0.00	0.00
Média	**0.74**	**0.66**	**0.00**	**0.00**

Tabela A.13: Resultado da avaliação da extração dos valores dos atributos da base *vitacost.com*

Base	**MAIt**		**ViNTs**	
watchzone.com	**P**	**R**	**P**	**R**
1	1.00	1.00	0.07	0.15
2	1.00	1.00	0.00	0.00
3	1.00	1.00	0.00	0.00
4	1.00	1.00	0.00	0.00
5	1.00	1.00	0.00	0.00
6	1.00	1.00	0.00	0.00
Média	**1.00**	**1.00**	**0.01**	**0.03**

Tabela A.14: Resultado da avaliação da extração dos valores dos atributos da base *watchzone.com*

Base	**MAIt**		**ViNTs**	
wine.com	**P**	**R**	**P**	**R**
1	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00
5	1.00	1.00	0.89	0.80
6	1.00	1.00	0.00	0.00
1	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00
4	1.00	0.03	1.00	0.03
5	0.97	0.97	0.83	0.83
6	1.00	1.00	0.00	0.00
Média	**0.50**	**0.33**	**0.31**	**0.14**

Tabela A.15: Resultado da avaliação da extração dos valores dos atributos da base *wine.com*

Base	**MAIt**		**ViNTs**	
yahoo.com/search/people	**P**	**R**	**P**	**R**
1	1.00	1.00	0.00	0.00
2	1.00	1.00	0.00	0.00
3	1.00	1.00	0.00	0.00
Média	**1.00**	**1.00**	**0.00**	**0.00**

Tabela A.16: Resultado da avaliação da extração dos valores dos atributos da base *yahoo.com/search/people*

Base	**MAIt**		**ViNTs**	
alltheweb.com	**P**	**R**	**P**	**R**
1	0.94	0.94	0.81	0.94
2	0.86	0.86	0.90	0.86
3	0.92	0.92	0.72	0.76
4	1.00	1.00	0.51	0.36
5	1.00	1.00	0.00	0.00
Média	**0.94**	**0.94**	**0.59**	**0.58**

Tabela A.17: Resultado da avaliação da extração dos valores dos atributos da base *alltheweb.com*

Base	**MAIt**		**ViNTs**	
amercoll.edu	**P**	**R**	**P**	**R**
1	1.00	0.94	0.98	0.98
2	1.00	0.94	1.00	1.00
3	0.91	0.86	1.00	1.00
Média	**0.97**	**0.91**	**0.99**	**0.99**

Tabela A.18: Resultado da avaliação da extração dos valores dos atributos da base *amercoll.edu*

Base	**MAIt**		**ViNTs**	
american.edu	**P**	**R**	**P**	**R**
1	0.00	0.00	1.00	1.00
2	0.48	0.48	1.00	1.00
3	1.00	1.00	0.00	0.00
4	1.00	0.20	0.00	0.00
5	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00
Média	**0.75**	**0.61**	**0.67**	**0.67**

Tabela A.19: Resultado da avaliação da extração dos valores dos atributos da base *american.edu*

Base	**MAIt**		**ViNTs**	
atlanticuc.edu	**P**	**R**	**P**	**R**
1	1.00	1.00	1.00	1.00
2	1.00	1.00	0.92	0.92
3	0.94	0.94	0.30	0.06
4	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00
Média	**0.59**	**0.59**	**0.44**	**0.40**

Tabela A.20: Resultado da avaliação da extração dos valores dos atributos da base *atlanticuc.edu*

Base	**MAIt**		**ViNTs**	
atu.edu	**P**	**R**	**P**	**R**
1	0.98	0.98	0.88	1.00
2	0.99	0.99	0.97	0.97
3	1.00	1.00	0.82	0.81
4	0.99	0.99	0.00	0.00
5	1.00	1.00	0.93	1.00
6	1.00	1.00	0.82	1.00
Média	**0.99**	**0.99**	**0.74**	**0.80**

Tabela A.21: Resultado da avaliação da extração dos valores dos atributos da base *atu.edu*

Base	**MAIt**		**ViNTs**	
bu.edu	**P**	**R**	**P**	**R**
1	1.00	1.00	1.00	1.00
2	0.98	0.98	0.99	0.99
3	1.00	1.00	0.80	0.80
4	1.00	1.00	0.00	0.00
5	0.99	0.99	0.99	0.99
6	1.00	1.00	1.00	1.00
Média	**1.00**	**1.00**	**0.80**	**0.80**

Tabela A.22: Resultado da avaliação da extração dos valores dos atributos da base *bu.edu*

Base	**MAIt**		**ViNTs**	
campbellsville.edu	**P**	**R**	**P**	**R**
1	1.00	1.00	1.00	1.00
2	0.88	0.88	0.90	0.90
3	0.90	0.90	1.00	1.00
Média	**0.93**	**0.93**	**0.97**	**0.97**

Tabela A.23: Resultado da avaliação da extração dos valores dos atributos da base *campbellsville.edu*

Base	**MAIt**		**ViNTs**	
clemson.edu	**P**	**R**	**P**	**R**
1	1.00	1.00	1.00	1.00
2	1.00	1.00	1.00	1.00
3	1.00	1.00	0.26	0.26
4	1.00	1.00	0.00	0.00
5	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00
Média	**1.00**	**1.00**	**0.71**	**0.71**

Tabela A.24: Resultado da avaliação da extração dos valores dos atributos da base *clemson.edu*

Base	**MAIt**		**ViNTs**	
csuchico.edu	**P**	**R**	**P**	**R**
1	0.34	0.24	0.47	1.00
2	0.73	0.66	0.10	0.04
3	0.95	0.84	0.00	0.00
4	0.48	0.44	0.00	0.00
5	0.00	0.00	0.00	0.00
6	0.96	0.86	0.00	0.00
Média	**0.58**	**0.51**	**0.10**	**0.17**

Tabela A.25: Resultado da avaliação da extração dos valores dos atributos da base *csuchico.edu*

Base	**MAIt**		**ViNTs**	
csudh.edu	**P**	**R**	**P**	**R**
1	1.00	1.00	0.46	1.00
2	0.92	0.96	0.10	0.06
3	0.88	0.88	0.00	0.00
4	0.62	0.62	0.00	0.00
5	0.00	0.00	0.00	0.00
6	0.98	0.94	0.00	0.00
Média	**0.73**	**0.73**	**0.09**	**0.18**

Tabela A.26: Resultado da avaliação da extração dos valores dos atributos da base *csudh.edu*

Base	**MAIt**		**ViNTs**	
fairfield.edu	**P**	**R**	**P**	**R**
1	1.00	0.06	1.00	1.00
2	1.00	0.94	1.00	1.00
3	0.40	0.32	0.30	0.06
4	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00
Média	**0.48**	**0.26**	**0.46**	**0.41**

Tabela A.27: Resultado da avaliação da extração dos valores dos atributos da base *fairfield.edu*

Base	**MAIt**		**ViNTs**	
franklin.edu	**P**	**R**	**P**	**R**
1	1.00	1.00	1.00	1.00
2	1.00	1.00	1.00	1.00
Média	**1.00**	**1.00**	**1.00**	**1.00**

Tabela A.28: Resultado da avaliação da extração dos valores dos atributos da base *franklin.edu*

Base	**MAIt**		**ViNTs**	
harvard.edu	**P**	**R**	**P**	**R**
1	1.00	0.02	1.00	1.00
2	0.66	0.66	1.00	1.00
3	1.00	0.98	1.00	1.00
4	1.00	1.00	0.00	0.00
5	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00
Média	**0.94**	**0.78**	**0.83**	**0.83**

Tabela A.29: Resultado da avaliação da extração dos valores dos atributos da base *harvard.edu*

Base	**MAIt**		**ViNTs**	
metacrawler.com	**P**	**R**	**P**	**R**
1	0.98	0.98	0.95	0.95
2	0.97	0.97	1.00	1.00
3	0.27	0.27	0.27	0.27
Média	**0.74**	**0.74**	**0.74**	**0.74**

Tabela A.30: Resultado da avaliação da extração dos valores dos atributos da base *metacrawler.com*

Base	**MAIt**		**ViNTs**	
mit.edu	**P**	**R**	**P**	**R**
1	0.96	0.96	0.80	0.91
2	0.96	0.96	0.92	0.95
3	1.00	1.00	0.19	0.04
4	1.00	1.00	0.00	0.00
5	1.00	1.00	0.00	0.00
6	1.00	1.00	0.87	1.00
7	1.00	1.00	0.95	1.00
Média	**0.99**	**0.99**	**0.53**	**0.56**

Tabela A.31: Resultado da avaliação da extração dos valores dos atributos da base *mit.edu*

Base	**MAIt**		**ViNTs**	
search.excite.com	**P**	**R**	**P**	**R**
1	0.00	0.00	0.92	0.92
2	0.56	0.56	0.98	0.98
3	0.48	0.96	0.23	0.23
Média	**0.35**	**0.51**	**0.71**	**0.71**

Tabela A.32: Resultado da avaliação da extração dos valores dos atributos da base *search.excite.com*