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

Rotulagem Automática de Dados Anônimos Extraídos da Web

Marco Aurélio da Silva Sevalho

Manaus – Amazonas
Maio de 2007

Marco Aurélio da Silva Sevalho

Rotulagem Automática de Dados Anônimos Extraídos da Web

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 e Banco de Dados.

Orientador: Prof. João Marcos Bastos Cavalcanti, Ph.D.
Co-orientador: Prof. Dr. Altigran Soares da Silva

Marco Aurélio da Silva Sevalho

Rotulagem Automática de Dados Anônimos Extraídos da Web

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 e Banco de Dados.

Banca Examinadora

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

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

Prof. Marco Antônio Casanova, Ph.D.
Departamento de Ciência da Computação – PUC-Rio

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

Manaus – Amazonas
Maio de 2007

Aos meus pais pela dedicação de toda uma vida.

A minha namorada pelo amor e companheirismo.

Agradecimentos

Primeiramente à *Deus* pela conclusão de mais esta etapa na minha vida.

Aos meus pais *Horácio* e *Ana Maria Sevalho* pela compreensão e apoio diante da ausência temporária.

Em especial a grande dupla de orientadores que tive, os meus sinceros agradecimentos aos professores doutores *João Marcos* e *Altigran*. Obrigado pela oportunidade e por terem propiciado uma formação exemplar que jamais será esquecida. Obrigado também à importante colaboração oferecida pelo professor Dr. *Denilson Barbosa*.

À minha namorada *Cláudia Suzany* pela motivação e pelo maravilhoso convívio durante a jornada.

Aos colegas da Secretaria Municipal de Finanças, em especial à *Laura Takahashi*. Seu apoio foi essencial.

Aos meus amigos que sempre me ajudaram a tornar as coisas menos difíceis.

Aos colegas do mestrado pelo convívio e pela ajuda mútua.

Resumo

Nos últimos anos tem sido desenvolvido um considerável número de trabalhos sobre extração automática de dados da Web. Entretanto estes trabalhos são capazes de capturar apenas a estrutura implícita dos dados e falham em atribuir qualquer semântica aos mesmos.

Embora os extratores automáticos de dados da Web geralmente consigam identificar com elevada acurácia os valores de cada atributo nos dados extraídos, eles não conseguem atribuir rótulos significativos a estes atributos. Isso acarreta numa séria limitação para o uso destes métodos num contexto de integração de dados, já que eles requerem uma considerável intervenção humana na rotulagem dos dados extraídos.

Este trabalho trata do problema de encontrar rótulos descritivos para um conjunto relacional de dados anônimos tais quais produzidos por extratores de dados da Web no estado da arte. Uma abordagem de rotulagem automática simples mas altamente efetiva é proposta nesta dissertação. A atribuição de rótulos é feita a partir de um modelo probabilístico que computa a *afinidade* entre atributos anônimos e rótulos candidatos. O modelo é povoado por um método que envia consultas especulativas para máquinas de busca na Web. Também é proposto um método automático e bastante efetivo para encontrar rótulos candidatos na Web para o conjunto de dados anônimos. Até onde se sabe, esta é a primeira abordagem totalmente automática para rotular dados extraídos da Web que não necessita dos documentos de onde os dados foram extraídos.

Ao final são apresentados resultados de experimentos extensivos feitos com 8 diferentes domínios de aplicação mostrando que os métodos atingem alta acurácia mesmo quando poucas consultas são enviadas a uma máquina de busca.

Palavras-chave: Extração de Dados da Web. Rotulagem Semântica de Dados.

Abstract

There has been considerable work on automatic extraction of data from the Web in recent years. While existing methods are capable of capturing the *structure* of the underlying data, they fail to attach any *semantics* to such data.

In particular, although they can usually identify the values of each attribute in the extracted data with high accuracy, they do not assign meaningful labels to those attributes. This poses a severe limitation for using these methods in a data integration setting, as they require considerable human intervention in labeling the extracted data.

In this work, we consider the problem of finding descriptive labels for anonymous, structured data sets, such as those produced by state-of-the-art Web wrappers. We propose a simple yet highly effective and fully automatic method for labeling anonymous data extracted from the Web. A probabilistic model computes the *affinity* between anonymous attributes and candidate labels, and describes a method that uses standard Web search engines for populating the model. Also, we propose an effective and fully automatic method for finding candidate labels for a given anonymous data set. To the best of our knowledge, this is the first fully automatic method for labeling extracted Web data that does not rely on mining the HTML pages containing the data.

We present the results of extensive experiments carried out with data from 8 different domains, showing that our methods achieve high accuracy even with very few search engine accesses.

Keywords: Web data extraction. Data Labeling.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A Web é uma vasta, embora desorganizada, fonte de informações apresentadas nas mais variadas formas. Diversas aplicações para a Web têm sido propostas envolvendo as tarefas de coleta, indexação, busca de documentos, extração de dados, integração de esquemas, etc. O problema de extração de dados na Web tem sido tratado na literatura por inúmeras abordagens que se utilizam de diferentes técnicas [11]. Dessa forma, vários métodos de extração automática e semi-automática têm sido propostos [1, 4, 22] para identificar coleções de dados na Web e reorganizar estes dados num formato adequado para integração com outras aplicações. Entretanto, estes métodos são capazes de reconhecer apenas a estrutura dos dados implícitos, mas não sua semântica. Tais ferramentas produzem conjuntos de dados com objetos *anônimos* (ou seja, sem rótulos descritivos associados a eles). Isto limita severamente sua usabilidade em aplicações de integração de dados, uma vez que estes objetos anônimos fazem necessária uma intervenção manual considerável por parte do usuário.

Em face dessa limitação, outros autores [2, 5, 18] propõem métodos para rotulagem de dados anônimos gerados por extratores automáticos de dados da Web. Em geral, estes métodos atuam buscando termos com alguma formatação distintiva dentro dos documentos de onde os dados foram extraídos. Entretanto, apesar deste tipo de abordagem alcançar altos índices de acurácia (os autores de [2] relatam até 90% de acurácia nos seus experimentos), é possível apontar duas falhas: a ausência de rótulos para todos os valores

de dados apresentados e a restrição ao uso dos rótulos utilizados nos documentos.

Com relação ao primeiro caso, observa-se que nem todos os documentos da Web contêm rótulos descritivos para todos os seus atributos. Um exemplo disso pode ser visto na descrição de um livro na Figura 1.1.

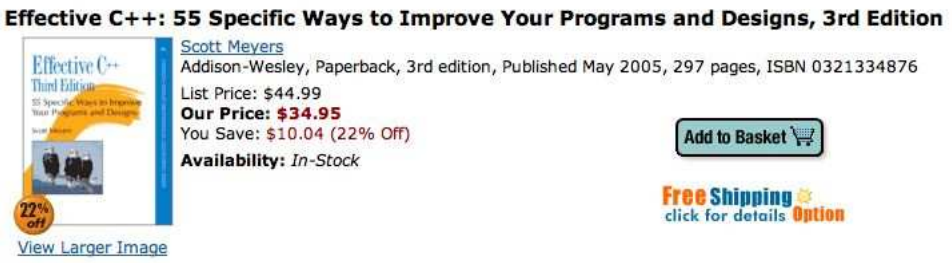

Figura 1.1: Descrição de um livro extraída de `http://www.bookpool.com`.

Este é um exemplo típico do trecho de um documento da Web que é alvo de extração. Apesar dele possuir alguns rótulos descritivos, boa parte dos rótulos está ausente. Mais especificamente, o trecho em questão descreve valores de um objeto com 13 atributos distintos, dos quais 8 não possuem rótulos: título (*Effective C++: 55 Specific Ways to Improve Your Programs and Designs*), autor (*Scott Meyers*), editora (*Addison-Wesley*), formato (*Paperback*), edição (*3rd edition*), número de páginas (*297 pages*), percentual de desconto (*22% Off*) e frete (*Free Shipping*); enquanto que apenas 5 possuem algum rótulo: código ISBN (*ISBN 0321334876*), preço de estoque (*List Price: $44.99*), preço promocional (*Our Price: $34.95*), valor de desconto (*You Save: $10.04*) e disponibilidade (*Availability: In-Stock*). Vale ainda destacar neste exemplo a existência de um atributo não textual (o valor do atributo frete é representado através de uma imagem) e de um atributo com rótulo sem formatação distintiva (é o caso de ISBN, que não possui o delimitador ":", como nos demais casos). Em ambas as situações, haveria problemas em aplicar alguma das abordagens de rotulagem existentes. No primeiro caso, não haveria como atribuir rótulos, pois eles não existem. Já a outra situação (ausência de formatação condicional), mostra ser um desafio para qualquer abordagem atual de rotulagem.

Uma segunda falha identificada nos métodos existentes, é que estes limitam-se ao

uso dos rótulos definidos pelos autores dos documentos extraídos da Web. Dessa forma, se estes autores escolherem rótulos pouco significativos, os dados extraídos não terão rótulos suficientemente descritivos. Restringir os rótulos aos escolhidos pelo autor do documento possui outra desvantagem. Considere o cenário onde se queira integrar dados de um documento da Web com um banco de dados local. Neste caso, seria necessário extrair os dados do documento, extrair seus rótulos e encontrar correspondências entre o esquema extraído (estrutura e rótulos) e o esquema do banco de dados local usando uma ferramenta para reconhecimento de padrões de casamento [16]. Entretanto, uma abordagem mais efetiva e potencialmente melhor seria usar os nomes dos atributos no esquema do banco de dados local para rotular os dados extraídos da Web e, aí então, integrá-los.

Este trabalho propõe uma abordagem nova e altamente efetiva para selecionar e atribuir de forma automática, a partir da Web, rótulos semanticamente representativos a conjuntos de dados anônimos extraídos da Web. Os resultados experimentais obtidos com esta abordagem demonstram bons níveis de revocação na identificação de rótulos candidatos e elevada acurácia na atribuição destes rótulos.

1.1 Abordagem Proposta

Um conjunto de dados anônimos é uma coleção estruturada de dados onde os rótulos descritivos para objetos semelhantes estão ausentes. Por exemplo, a Figura 1.2 mostra o trecho de um conjunto (relacional) de dados anônimos sobre carros gerado pela ferramenta de extração automática *RoadRunner* [4][1].

Este conjunto de dados é um exemplo da saída produzida por boa parte das ferramentas de extração no estado da arte [2]. O exemplo mostra que os dados possuem um esquema $R(A, B\,C\,D)$ bem definido, embora semanticamente frágil. Além disso, ambos atributos têm domínios bem definidos (A contém marcas de carros, B contém tipos de carros, C

[1] `http://www.dia.uniroma3.it/db/roadRunner/amazon/cars/data.html`

[2] Algumas ferramentas produzem relações aninhadas codificadas em formato XML que podem ser facilmente convertidas em um formato relacional

R	A	B	C	D
	Acura	Coupe	2000	NSX
	Audi	Sedan	2001	A4
	Volvo	Wagon	2001	V70 XC
	...			

Figura 1.2: Trecho de um conjunto de dados anônimos $R(A, B, C, D)$ sobre carros extraído pela ferramenta *RoadRunner*.

ano e D modelo). O problema de rotulagem, do qual é tratado nesta dissertação, consiste em encontrar rótulos descritivos para um conjunto relacional de dados anônimos.

A solução para o problema de rotulagem pode ser visto como um processo em dois passos: (1) encontrar um bom conjunto de rótulos candidatos e (2) encontrar a melhor combinação de associações entre rótulos e atributos. Este trabalho propõe uma abordagem composta de dois métodos totalmente automáticos para executar os passos 1 e 2, de acordo com a Figura 1.3.

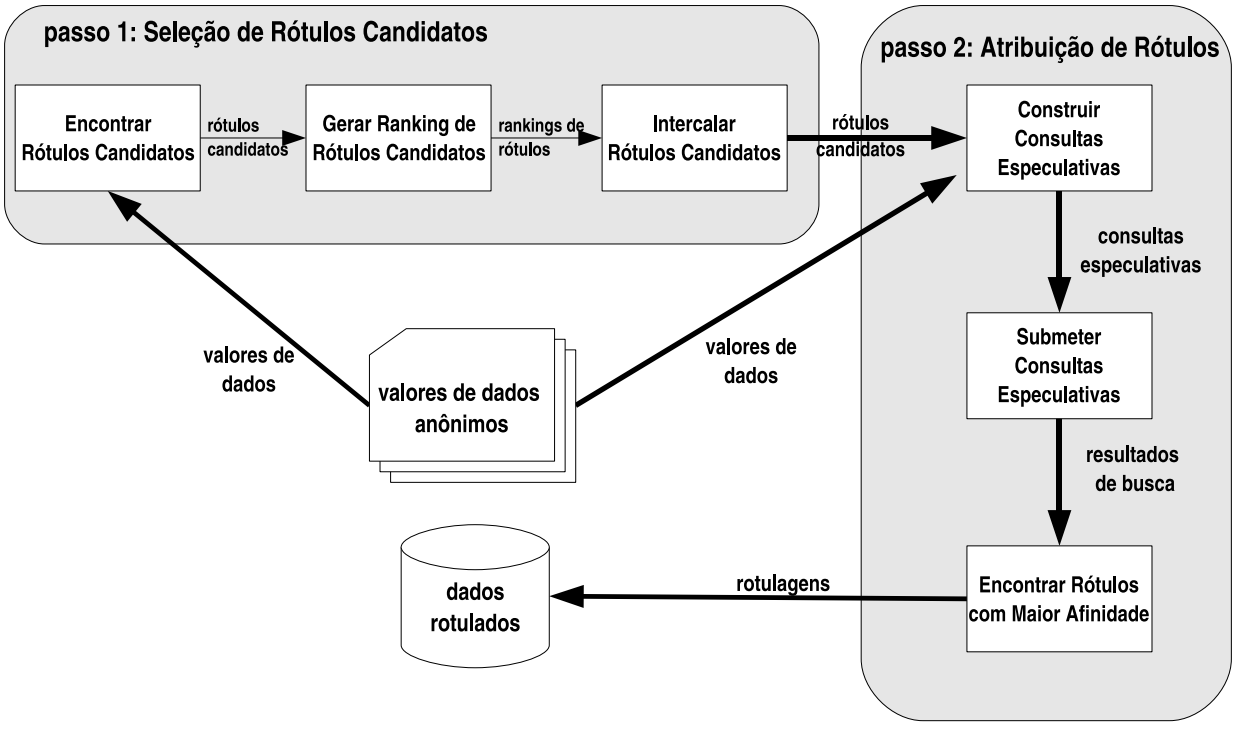

Figura 1.3: Visão geral da abordagem proposta

A abordagem considera a existência de um conjunto de *valores de dados anônimos* não rotulados que precisa de um rótulo descritivo para cada um de seus atributos anônimos.

Assume-se que estes dados anônimos tenham sido gerados a partir de algum processo automático (ou semi-automático) de extração de dados.

O *passo 1* corresponde ao método que realiza a *Seleção de Rótulos Candidatos*, ou seja, rótulos potencialmente representativos aos atributos anônimos. A entrada deste método são *valores de dados* do conjunto de dados anônimos e sua saída é uma lista de *rótulos candidatos* para este conjunto. Para identificar rótulos candidatos, este método busca em documentos da Web por certos padrões de texto comumente usados para descobrir instâncias de classes de objetos [10]. Estes padrões são usados para obter termos que ocorrem com freqüência e próximos a eles. Os termos que melhor atendem esses critérios são selecionados como rótulos candidatos.

O *passo 2* corresponde ao método que realiza a *Atribuição de Rótulos*. Essa atribuição é feita com a escolha dos rótulos candidatos que têm maior afinidade com os valores de dados anônimos. As entradas para este método são *valores de dados* anônimos e o conjunto de *rótulos candidatos*, definidos no passo anterior. Sua saída é um conjunto de *rotulagens* para os atributos anônimos, ou seja, os dados anônimos agora rotulados. A atribuição de rótulos é baseada num modelo probabilístico simples que leva em consideração a *afinidade* entre um conjunto de valores (de um atributo anônimo) e um conjunto de rótulos potencialmente representativos para seu domínio (os rótulos candidatos). As probabilidades são estimadas em função da contagem do número de respostas para uma *consulta especulativa* submetida a uma máquina de busca na Web de uso geral. Essencialmente, consultas especulativas formulam a hipótese que determinado termo é um rótulo válido para um dado atributo anônimo. Para avaliar a viabilidade dessa hipótese, o método usa uma máquina de busca na Web como uma espécie de oráculo. As etapas intermediárias deste método serão explicas no Capítulo 4.

É importante separar a abordagem nestes dois passos pois é possível encontrar rótulos candidatos de diversas maneiras. Por exemplo, eles podem ser oferecidos por um usuário, caso este esteja integrando dados extraídos da Web com algum banco de dados local (neste caso, os rótulos poderiam ser os nomes dos atributos do esquema do banco de dados).

Alternativamente, os rótulos candidatos poderiam ser extraídos semi-automaticamente de alguma fonte de dados. Por exemplo, alguém poderia buscar os rótulos candidatos nos documentos de onde os dados foram extraídos, como observado em [2, 18]. Dessa forma, apesar de haver diversas maneiras de encontrar rótulos candidatos, como será visto no Capítulo 3, a abordagem apresentada neste trabalho oferece um método totalmente automático para realizar esta tarefa. Este capítulo explica detalhadamente seu funcionamento e apresenta um estudo sobre o algoritmo que o implementa. Já no Capítulo 4 são oferecidos mais detalhes sobre os conceitos envolvidos no método, sobre o processo de rotulagem e sobre o algoritmo utilizado.

1.2 Principais Contribuições

Este trabalho apresenta uma nova abordagem automática que identifica e atribui rótulos semanticamente representativos a um conjunto de dados anônimos extraídos da Web. Ela se destaca dentre as demais por não precisar das páginas de onde os dados são extraídos e por utilizar rótulos obtidos na Web com forte representatividade para os domínios dos dados anônimos.

O método de seleção de rótulos candidatos apresenta uma técnica bastante eficaz que encontra termos potencialmente representativos para os atributos anônimos em documentos da Web. Já o método de rotulagem introduz uma técnica simples, que também envia consultas para a Web, para encontrar associações semânticas entre rótulos candidatos e atributos anônimos a partir do cálculo de sua afinidade. Em geral, estas técnicas garantem soluções com alta acurácia para algumas fragilidades identificadas nos métodos de rotulagem existentes na literatura. Além de contribuir na melhoria dos métodos automáticos de extração de dados, a abordagem também auxilia tarefas complementares à extração, tal como a integração de dados. Outra vantagem da abordagem proposta é que ela pode atuar de forma complementar às abordagens existentes baseadas em heurísticas locais.

Os resultados de experimentos feitos com 52 atributos pertencentes a 8 diferentes domínios mostram níveis de acurácia na atribuição de rótulos acima de 90% em domínios

considerados populares na Web e níveis de revocação acima de 85% na seleção de rótulos candidatos.

1.3 Organização da Dissertação

Este trabalho está organizado da seguinte forma: No Capítulo 2 são discutidos os trabalhos relacionados. No Capítulo 3 é apresentado o primeiro passo da abordagem proposta referente ao método de seleção os rótulos candidatos. O Capítulo 4 apresenta o método referente ao passo seguinte da abordagem que realiza a rotulagem dos conjuntos de dados anônimos. No Capítulo 5 é apresentada a metodologia de avaliação e os resultados experimentais tanto do método de seleção quanto do método de rotulagem são discutidos. Finalmente, o Capítulo 6 apresenta as conclusões e trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

Este capítulo apresenta alguns trabalhos importantes relacionados com temas da abordagem de rotulagem proposta nesta dissertação. Inicialmente na Seção 2.1 são identificados trabalhos sobre extração automática de dados da Web. Em seguida, na Seção 2.2 são revisado alguns trabalhos que tratam da rotulagem de dados anônimos extraídos da Web. Finalmente, na Seção 2.3 são apresentadas algumas abordagens de gerenciamento de dados na Web que utilizam técnicas de busca para gerar informação a partir das respostas obtidas, tal como é feito neste trabalho com as consultas especulativas.

2.1 Extração de Dados na Web

Atualmente existe uma vasta quantidade de trabalhos relacionados com extração de dados na Web. Laender em [11] apresenta um *survey* sobre as várias abordagens encontradas na literatura. Tais abordagens, segundo [11], baseiam-se em diferentes técnicas, tais como: linguagens declarativas, análise estrutural, aprendizagem de máquina, processamento de linguagem natural, modelagem de dados e ontologias. Ainda de acordo com este trabalho, uma ferramenta de extração pode ser classificada de acordo com seu grau de automação em: manual, semi-automática e automática. A abordagem de rotulagem apresentada nesta dissertação é complementar aos dois últimos tipos. Principalmente às ferramentas automáticas. Dessa forma, nesta seção são discutidos alguns importantes métodos que

requerem pouca ou nenhuma intervenção humana no processo de extração [1, 3, 4, 13, 18, 22, 23].

Crescenzi *et al.* [4] propõe a ferramenta *RoadRunner*, uma abordagem para geração automática de *wrappers* baseada na estrutura de documentos no formato HTML[1]. Seu trabalho é apresentado como um problema de inferência de gramática, dado um conjunto de exemplos positivos de entrada (documentos HTML fornecidos pelo usuário). O processo de extração transforma os documentos em árvores que representam a hierarquia dos marcadores (*tags*) HTML. A partir dessas árvores, a ferramenta gera regras de extração baseadas nas diferenças estruturais descobertas durante o alinhamento dos marcadores. Apesar desta abordagem promover um alto grau de automação, ela é exclusiva para estruturas HTML e é fortemente dependente do uso consistente de marcadores. Além disso, esta ferramenta está sujeita à extração de elementos não relevantes (menus, *templates*, propaganda, etc.), que causam ruídos no resultado da extração e afetam qualquer aplicação que use estes dados.

Também seguindo uma abordagem automática, Arasu e Garcia-Molina [1] apresentam o algoritmo *ExAlg* que toma como entrada um conjunto de páginas geradas a partir de *templates*. Um *template* é um conjunto comum de características de formato e *layout* que ocorrem num conjunto de páginas HTML produzidas por um programa ou *script*, que gera dinamicamente páginas com conteúdo em formato HTML. O algoritmo infere a estrutura implícita destes *templates*, que é usadada como base para a extração dos valores das páginas de entrada. Esta abordagem introduz um novo método para identificação de *templates* e seus resultados experimentais, conduzidos com 45 coleções de páginas, indicam uma acurácia média em torno de 80%. Além disso, esta abordagem apresenta melhorias em alguns aspectos da abordagem de Crescenzi *et al.* [4]. Entretanto, assim como a ferramenta *RoadRunner* [4], esta abordagem também sofre de problemas relacionados com a dependência de marcadores HTML e com as interferências decorrentes da extração de dados irrelevantes. Vale destacar, por fim, que ambas as abordagens não se preocupam

[1] *Hipertext Markup Language*

em atribuir rótulos significativos aos dados extraídos.

Omni [3] é um outro método para extração automática de dados em documentos da Web. Seu processo de extração consiste em 3 etapas: preparação do documento da Web para extração, localização dos objetos de interesse e extração desses objetos. Durante este processo o método combina uma série de heurísticas para obter melhorias nos seus resultados. As heurísticas analisam algumas propriedades na estrutura das tags, tais como: o desvio-padrão da ocorrência de tags, o padrão de ocorrência de pares de tags, tags que separam sub-árvores HTML, etc. Em experimentos conduzidos com 2000 páginas do 50 Web sites de comércio eletrônico, os autores relatam 100% de acurácia (*i.e.* ausência de falsos positivos) e níveis altos de revocação (entre 93% a 98%).

DeLa [18] é um método automático para extração de dados em páginas da Web geradas dinamicamente (ou seja, que fazem parte da chamada *Deep Web* ou *Hidden Web*). A extração é feita através da indução de expressões regulares a partir da estrutura de marcadores HTML de um conjunto de páginas geradas dinamicamente por um mesmo *template*. Para inferir os *wrappers* de forma eficiente, o método utiliza um algoritmo para detecção de áreas ricas em dados. Os *wrappers* gerados são utilizados para extrair dados numa tabela (semelhante ao seu esquema implícito). Nesta tabela são aplicadas várias heurísticas para separar corretamente valores de dados aninhados e transformá-los em novos atributos. Experimentos conduzidos com páginas de 27 sites da Web de 3 diferentes domínios mostram uma revocação de 100% na fase de geração dos wrappers e 95% de acurácia na fase de separação dos atributos. Além de extrair e alinhar dados de paginas da Web, a próxima seção mostra de que forma *Dela* realiza a rotulagem desses dados.

Em outra linha de trabalho, Liu *et al.* [13] e Zhao *et al.* [23] estudam o problema de extração automática de registros a partir dos resultados de consultas feitas a máquinas de busca na Web. O objetivo destes trabalhos é a melhoria dos resultados de ferramentas de meta-busca.

Embora se diferenciem em muitos aspectos, todos estes métodos têm como foco a estrutura, e não a semântica, dos dados contidos nos documentos extraídos da Web.

Dessa forma, todas estas ferramentas produzem somente conjuntos de dados *anônimos*, tal como discutido anteriormente.

2.2 Rotulagem de Dados Anônimos Extraídos da Web

Com os avanços das abordagens automáticas de extração de dados da Web, novos trabalhos foram desenvolvidos com o objetivo de atribuir rótulos os dados anônimos produzidos por *wrappers* [2]. Outros trabalhos recentes passaram a oferecer recursos de rotulagem como uma etapa adicional em seus processos de extração automática [5, 18]. Nesta seção são apresentadas alguns trabalhos que apresentam iniciativas de rotulagem de dados anônimos extraidos da Web. Mesmo assim, como será visto, ainda há pontos em aberto nestas iniciativas que o trabalho apresentado nesta dissertação visa contemplar.

Apesar do foco deste trabalho ser em métodos de rotulagem *genéricos* (*i.e.*, métodos não limitados a um domínio específico), é importante destacar que existem métodos de extração de dados em domínios específicos bastante eficientes. Por exemplo, Reis *et al.* [5] desenvolveu uma abordagem automática para extrair e rotular artigos de notícias da Web. Seu médoto é baseado no conceito de distância de edição em árvores. Neste trabalho a rotulagem é realizada como uma epata complementar ao processo de extração e seu objetivo específico é encontrar atribuições para passagens de texto que correspondam a um *título* ou a um *corpo* de uma notícia extraída da Web. Apesar de usar uma heurística simples, esta estratégia de rotulagem mostrou-se bastante efeitva em experimentos com mais de 4000 páginas de 35 fontes de dados distintas.

São conhecidos ainda outros dois métodos de rotulagem automática de dados anônimos extraídos da Web: *DeLa* (*Data Extraction and Label Assignment*) [18] e *Labeller* [2], que serão discutidos a seguir.

A ferramenta *DeLa* [18], já citada na seção anterior, além de extrair valores de dados através da reconstrução do esquema implícito do banco de dados a da análise da estrutura

de páginas geradas dinamicamente, também associa rótulos aos atributos anônimos do esquema reconstruído.

Esta abordagem de atribuição de rótulos é baseada no seguinte conjunto de heurísticas: (1) há similaridade entre os rótulos dos campos de um formulário HTML de consulta e os valores de dados extraídos de páginas geradas a partir de consultas neste formulário; (2) há presença de rótulos nos cabeçalhos de estruturas tabulares de documentos HTML de onde os valores de dados são extraídos; (3) há presença de rótulos nas proximidades dos valores de dados extraídos; (4) a formatação dos valores de dados pode dar indícios sobre seus rótulos, por exemplo: a presença de "@" dentro de um valor pode sugerir "*e-mail*" como um rótulo plausível. Os autores relatam resultados experimentais mostrando que a combinação dessas quatro heurísticas levam a uma acurácia na atribuição de rótulos próxima de 90%. Além disso, assim como o trabalho aqui apresentado, *Dela* também utiliza uma técnica de envio de consultas para Web em uma de suas heurísticas. Esta técnica será comentada na próxima seção.

Labeller [2] é uma extensão da ferramenta *RoadRunner* [4] que usa um conjunto de termos identificados dentro de páginas da Web para rotular dados anônimos extraídos de forma automática dessas mesmas páginas. *Labeller* usa um conjunto de heurísticas baseadas na posição entre determinados elementos gráficos de uma página da Web renderizada (tal como exibida em um navegador da Web), para encontrar as melhores associações entre rótulos e valores de dados. Estas heurísticas capturam estilos de projeto de páginas da Web relacionados com a distância e o alinhamento entre componentes gráficos que representam valores de dados e seus respectivos rótulos. Por exemplo, o método baseia-se no fato que um rótulo é geralmente colocado acima ou à esquerda do seu valor correspondente, numa localização não muito distante dele. Os autores oferecem resultados experimentais indicando alta acurácia (em torno de 90%), o que fica próximo dos resultados relatados pela ferramenta *DeLa* [18].

Uma importante limitação dos métodos *DeLa* e *Labeller* é sua forte dependência em encontrar os rótulos nos documentos da Web de onde os dados foram extraídos. Observe

que isso limita a escolha de rótulos para somente aqueles usados pelo autor do conteúdo do documento. Além disso, tal como discutido anteriormente, essas abordagens falham se algum dos rótulos estiver ausente no documento da Web de onde foi feita a extração (tal como ilustra a Figura 1.1). O método de rotulagem apresentado nessa dissertação, por outro lado, não sofre de nenhum desses problemas.

2.3 Uso de Consultas Especulativas na Web

Nesta seção é demonstrado que a idéia do uso de busca na Web para ajuda em tarefas de gerenciamento de dados na Web não é nova.

Por exemplo, Goldman e Windom [9] exploram máquinas de busca na Web para melhorar os resultados de consultas SQL[2] em um banco de dados relacional. Sua abordagem, denominada WSQ (*Web Supported database Queries*) permite que usuários escrevam consultas SQL que automaticamente executam buscas na Web e combinam seus resultados com os dados estruturados de um banco de dados local.

Outro tema comum na literatura é o uso de busca na Web como meio de encontrar mapeamentos entre esquemas implícitos em formulários de interface da *Hidden Web*. Wang *et al.* [19], apresenta o problema de identificação de correspondências semânticas entre diferentes esquemas implícitos em bancos de dados na Web. Este problema é equivalente ao de encontrar mapeamentos de esquemas, só que aplicado a bancos de dados na *Hidden Web*. A solução proposta por Wang envolve o envio de consultas através de formulários de interface. As respostas a estas consultas são analisadas para inferir correspondências entre os elementos da interface de busca e das páginas da Web resultantes. Observe que, mesmo usando uma técnica para envio de consultas na Web, esta ainda limita-se a encontrar os rótulos nas páginas resultantes das buscas feitas nas interfaces de consulta.

Em [21] Wu *et al.* apresenta a ferramenta *WebIQ*. Nela os resultados de consultas feitas a uma máquina de busca são usados para validar associações entre valores de dados e rótulos de formulários de interface de bancos de dados da Web. Para obter estes resul-

[2] *Structured Query Language*

tados, *WebIQ* formula consultas usando rótulos de atributos (encontrados no formulário), instâncias candidatas (obtidas numa etapa anterior) e alguns padrões léxico-sintáticos de validação. Um escore de validação é calculado para cada resposta obtida da máquina de busca e os valores de dados com maior escore são selecionados como válidos. Ao final, é feito um mapeamento entre rótulos de diferentes formulários de interfaces associados com um mesmo conjunto de instâncias de valores de dados.

Nestes dois últimos trabalhos citados, observa-se que ambos abordam o problema do mapeamento de esquemas de banco de dados com o conhecimento prévio sobre a semântica dos conjuntos de dados. No caso do método proposto nesta dissertação, há somente conjuntos de dados anônimos. Dessa forma, embora a idéia geral seja a mesma, este trabalho a explora de forma diferente e com propósitos distintos.

Por fim, o modelo probabilístico usado nesta dissertação, que estima a afinidade entre valores de atributos e rótulos, mostrou-se similar ao utilizado pelo algoritmo PMI-IR, proposto por Turney [17]. Este algoritmo é usado para encontrar sinônimos entre palavras da língua inglesa através do cálculo de suas similaridades. A acurácia dessas similaridades, tal como observado pelo autor, depende do tamanho da coleção de documentos no sistema de recuperação de informação utilizado (no caso desta dissertação, toda a Web). Além disso, os resultados de acurácia obtidos nos experimentos do Capítulo 5 são bastante similares aos obtidos por Turney.

Capítulo 3

Método de Seleção de Rótulos Candidatos

Este capítulo apresenta os fundamentos, conceitos e técnicas envolvidas no funcionamento do método de seleção de rótulos candidatos, além do algoritmo que o implementa. O capítulo está estruturado da seguinte maneira. A Seção 3.1 dá uma visão geral do método com o enunciado das hipóteses que o fundamentam. Em seguida, a Seção 3.2 explica a técnica de uso de padrões léxico-sintáticos, que serve de base para as etapas do método. A Seção 3.3 apresenta as etapas necessárias à geração dos rótulos candidatos. Finalmente, a Seção 3.4 define o algoritmo que implementa o método de seleção de rótulos candidatos juntamente com algumas simplificações e otimizações.

3.1 Visão Geral

Como visto na Seção 1.1, o objetivo do método de seleção de rótulos candidatos é gerar uma boa lista de rótulos para um conjunto de dados anônimos tal que possam ser usados no passo seguinte (atribuição de rótulos).

A construção desse método é feita a partir de algumas hipóteses. Dado um atributo A_i pertencente a uma relação anônima R, assume-se que:

(1) *rótulos expressivos para* A_i *provavelmente ocorrem em documentos da Web que*

também contêm (alguns) valores de A_i. Por exemplo, a Figura 3.1 apresenta a descrição de um objeto extraído de um site de comércio eletrônico de relógios. Nela pode ser observado um conjunto de valores de atributos pertencentes ao domínio de relógios ("Armani", "AR0536", "Men", etc). Para todos estes valores existe um rótulo significativo associado ("Brand", "Model", "Size", etc). De forma similar, documentos de outros *sites* da Web que também comercializam relógios provavelmente também possuem rótulos expressivos para os valores apresentados.

Figura 3.1: Descrição de um relógio extraída de `http://www.watchzone.com`.

(2) *nestes documentos, os rótulos provavelmente ocorrem próximos aos valores de* A_i. Por exemplo, em páginas de comércio eletrônico e de catálogos de produtos é comum que os rótulos descritivos estejam localizados no cabeçalho de estruturas tabulares, à esquerda ou acima dos seus respectivos valores (além de estarem associados com alguma formatação distintiva ou algum separador explícito). Um exemplo disso também é observado na Figura 3.1, onde os rótulos estão todos localizados à esquerda de seus valores, formatados em negrito e separados pelo caracter ":".

(3) *esta próximidade expressa uma relação de hiponímia*[1] *entre o rótulo e o valor.*

[1] Relação semântica que estabelece a subordinação ou pertencimento a um nível semântico (ou classe) inferior.

Ainda com relação à Figura 3.1, percebe-se que existe tal relação semântica entre: *Brand*←*Armani*, *Model*←*AR0536* e *Size*←*Men*.

(4) *esta situação é frequente na Web, de modo que estes documentos são provavelmente coletados por máquinas de busca populares*. De fato, o exemplo em questão, apesar de ter sido retirado a partir da máquina de busca *Google*, também pôde ser facilmente encontrado a partir do *Yahoo!* e do *MSN*.

3.2 Padrões Léxico-Sintáticos

Em [8, 21] observa-se que é possível descobrir, com alta acurácia, instâncias de uma dada classe de objetos similares enviando determinados tipos de consultas a uma máquina de busca na Web. Esta técnica consiste em buscar documentos na Web com um determinado padrão (*pattern*) léxico-sintático p. Estes *padrões* expressam relacionamentos entre termos, particularmente *hiponímias*, e neste contexto costumam ser chamados de *Hearst patterns* [21], em alusão ao trabalho de Hearst [10]. Por exemplo, o *padrão* "NP_1 *such as* $(\mathsf{NP}_2)*$" pode ser usado para encontrar uma ou mais instâncias NP_2 de uma classe NP_1. Para encontrar nomes de cidades usando essa técnica (aplicada à língua inglesa), busca-se por documentos que contenham a expressão "*cities such as*". A intuição é que os termos que ocorrem com maior freqüência após a expressão sejam nomes de cidades. De fato, de acordo com a Figura 3.2, os termos que ocorrem logo após a expressão "*cities such as*" são nomes de cidades ("St David's in Wales and Wells in England", "Shafer, Minnesota" e "Babylon, Athens, Rome, London, Madrid").

Dessa forma, considerando que Hearst [10] comprova experimentalmente que sua técnica funciona bem quando usada em grandes *corpus*[2] e que [8, 21] demonstram que esta técnica também pode ser aplicada na Web com bons resultados. O problema de encontrar rótulos candidatos para um conjunto de atributos pode ser considerado como um problema inverso ao identificado acima e, com isso, ser parcialmente investigado uti-

[2]Conjuntos de dados lingüísticos criteriosamente coletados sobre determinado assunto.

Figura 3.2: Consulta submetida ao *Yahoo!* usando o pattern "NP_1 *such as* $(\mathsf{NP}_2)*$'.

lizando uma abordagem similar. Mais especificamente, o método de seleção de rótulos candidatos usa a técnica *Hearst Patterns* para descobrir *hiperonímias*[3] entre vocábulos na língua inglesa.

Os padrões léxico-sintáticos utilizados no método são mostrados na Tabela 3.1. A pri-

Padrão	Estrutura	Exemplo
backward	"L *such as* v_k"	"*Film such as Before Sunrise*"
permutation	"$Ls \leftarrow v_1,\ldots, v_n$"	"*Bands: Beatles, Pink Floyd*"
forward	"v_k *is a* L"	"*Jazz is a Genre*"
value	"$L \leftrightarrow v$"	"*Brand: Armani*"

Tabela 3.1: *Hearst Patterns* usados no método de Seleção de Rótulos Candidatos.

meira coluna desta tabela identifica cada um dos 4 tipos de padrões utilizados (*backward, permutation, forward e value*). A segunda coluna mostra a estrutura de cada um destes padrões, onde $v_1,\ldots, v_n$ representa um conjunto de valores anônimos de um atributo A_i, v_k é um valor anônimo qualquer de A_i e L e Ls representam, respectivamente, um rótulo no singular e no plural. A última coluna mostra exemplos da instanciação destes padrões. A Seção 3.3 mostra detalhes da utilização estes padrões para encontrar rótulos candidatos.

Antes de entrar em detalhes, entretanto, é importante discutir porque o uso exclusivo da técnica de *Hearst patterns* falha na busca por bons rótulos (e por esta razão, o passo 2 da abordagem torna-se necessário). Primeiro, tal como mencionado anteriormente,

[3]Relação estabelecida entre um termo de sentido mais genérico e outro de sentido mais específico.

observa-se que o uso de *Hearst patterns* é particularmente adequado para encontrar hiponímias [10]. Ou seja, para encontrar termos semanticamente mais genéricos que um determinado termo. Assim, aplicando esta técnica é provável que se encontre "Paris" como sendo uma cidade, já que "Paris" é uma hiponímia popular para o termo "cidade". Entretanto, a palavra "paris" pode possuir significados diferentes, dependendo do contexto no qual é usada. De fato, no momento que este texto foi escrito - quando submetida a consulta "such as Paris" - o termo "Paris" não foi encontrado como referência direta à cidade francesa em nenhuma das 5 primeiras ocorrências de documentos retornados pelas máquinas de busca Google, Yahoo! e MSN. As respostas que mais se aproximam do desejado ocorrem em apenas dois, de quinze documentos, e referem-se a "Paris" com o termo *location* (localização), que é uma hipernímia de "city". Segundo, caso os valores usados para encontrar as associações não contenham termos populares na Web, as máquinas de busca costumam produzir respostas vazias, ou seja, sem documento relevante algum. Neste situação, não é possível aferir o quanto pode ser acurado um método baseado em *Hearst patterns*. Por essas razões, para promover uma rotulagem eficaz torna-se necessário aplicar um método adicional que avalie a afinidade do conjunto rótulos candidatos encontrados no passo 1 com os valores de dados anônimos. As etapas necessárias para gerar o conjunto de rótulos candidatos são apresentadas na Seção 3.3 e o método que examina a afinidade entre rótulos candidatos e valores de atributos anônimos é explicado no Capítulo 4.

3.3 As Etapas do Método

Esta seção apresenta as etapas do método de seleção de rótulos candidatos, tal como ilustrado no *passo 1* da Figura 3.3.

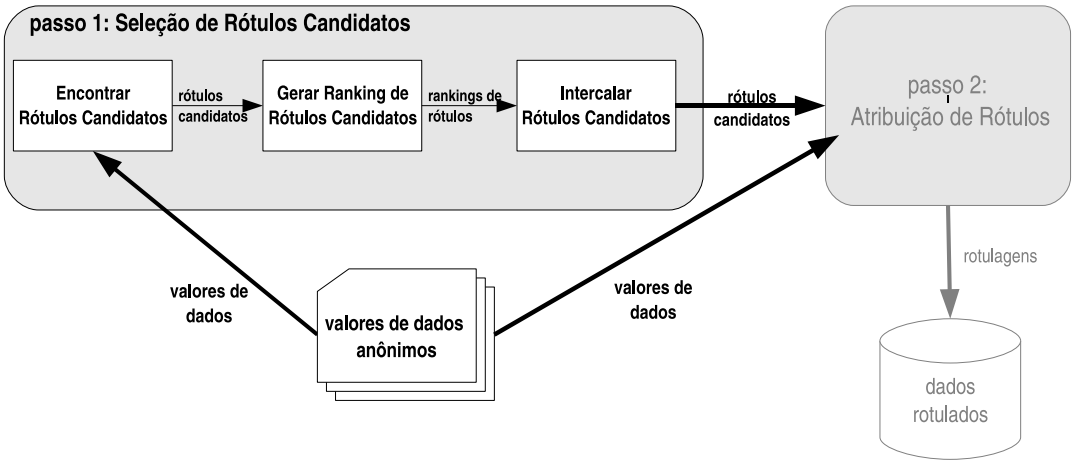

Figura 3.3: Método de Seleção de Rótulos Candidatos

A primeira etapa (subseção 3.3.1) aplica diferentes estratégias para identificar rótulos candidatos em documentos da Web. A segunda etapa (Seção 3.3.2) explica como avaliar a representatividade destes rótulos para formar rankings de rótulos candidatos para cada atributo anônimo. Finalmente, a terceira etapa (Seção 3.3.3) explica como gerar uma lista final de rótulos candidatos para o conjunto de dados anônimos.

3.3.1 Encontrando Rótulos Candidatos

A primeira etapa consiste em encontrar *rótulos candidatos* a partir de documentos da Web usando *valores de dados* de um atributo anônimo. O método utiliza estes valores para instanciar os padrões léxico-sintáticos apresentados na seção anterior. As consultas geradas por estes padrões são submetidas a uma máquina de busca na Web e os documentos respondidos são usados para encontrar os rótulos candidatos.

Definição 3.1 Dado um *padrão* p e um valor v de um atributo anônimo A_j. Uma *expressão de consulta* é definida como $q(p,v)$. Onde q é uma instanciação do padrão p

usando o valor $v \in A_j$

Dessa forma, uma expressão de consulta q é submetida a uma máquina de busca na Web que retorna um conjunto de documentos contendo q. Os rótulos candidatos são termos que ocorrem próximos à q nestes documentos, e o padrão p utilizado na expressão de consulta determina onde buscar os termos mais adequados. De modo geral, um rótulo candidato pode ser encontrado *à esquerda* da expressão de consulta, *à direita* ou em *ambos os sentidos*. Baseado nessa idéia, o método considera três estratégias para encontrar rótulos candidatos: (1) a estratégia *forward search* busca rótulos candidatos *à direita* da expressão de consulta; (2) de maneira inversa, a estratégia *backward search* busca por rótulos candidatos *à esquerda* da expressão de consulta; e (3) a estratégia de busca *bidirectional search* é uma combinação das duas anteriores, ou seja, busca por rótulos que apareçam antes ou depois da frase de consulta.

A Tabela 3.2 mostra a relação entre os tipos de expressões de consulta e as estratégias de busca correspondentes .

$q(p,v)$	estratégia	região de busca
"*such as* v"	*backward search*	termos antes da expressão de consulta
"v_1, v_2"'	*backward search*	termos antes da expressão de consulta
"v *is* "	*forward search*	termos após a expressão de consulta
"v"	*bidirectional search*	todos os termos na resposta

Tabela 3.2: *Patterns* para consulta usados no método de Seleção de Rótulos Candidatos.

A primeira coluna da Tabela 3.2 mostra as expressões de consulta correspondentes ao padrões léxico-sintáticos da Tabela 3.1. A segunda coluna mostra os nomes das estratégias de busca utilizadas e a última coluna identifica a região onde é feita a busca por rótulos candidatos. Observa-se que as expressões que usam os *padrões* do tipo *backward* e *permutation* (linhas 1 e 2) adotam a estratégia de busca do tipo *backward search*, enquanto a que usa o *padrão* do tipo *forward* (linha 3) adota a estratégia do tipo *forward search*. Já no padrão do tipo *value* (linha 4) a estratégia *bidirectional search* é aplicada. Observa-se ainda que para construir uma expressão de consulta usando o padrão *permutation* são usados dois valores anônimos distintos. Os experimentos feitos com essa técnica indica

que o padrão do tipo *permutation* é mais efetivo com atributos categóricos do que com outros tipos. Nota-se ainda que apesar dos padrões léxico-sintáticos usados no método serem similares aos usados em trabalhos anteriores (p.ex.,[8, 21]), o objetivo estabelecido aqui é exatamente o oposto ao desses autores: ou seja, encontrar hiperonímias ao invés de hiponímias.

Como mencionado anteriormente, este método considera como rótulos candidatos somente substantivos na língua inglesa. Para isso, utiliza a biblioteca *Java WordNet Library* (JWNL)[4] para a identificar substantivos e reconhecer variações sintáticas aplicando regras de inflexão. Ou seja, para o método aqui proposto, rótulos candidatos são substantivos no singular, representados na sua forma canônica de acordo com a *WordNet*[5]. Como também foi visto, estes substantivos ocorrem em documentos da Web próximos a certas *expressões de consulta*. Entretanto, como simplificação, o método restinge o espaço de busca do documento ao *snippet* [6] respondido pela máquina de busca, ao invés de usar o documento inteiro. Fazendo isso, os rótulos candidatos ficam restritos aos termos que ocorrem próximos à expressão de consulta no *snippet* gerado pela máquina de busca. Mais especificamente, seja S o conjunto dos *snippets* referentes aos documentos respondidos pela máquina de busca a uma expressão de consulta $q(p, v)$, que usa o padrão p instanciado com o valor v de um atributo anônimo A_j. De acordo com o padrão p aplicado o método encontra um conjunto de substantivos que são avaliados na etapa posterior de acordo com um escore numérico.

3.3.2 Gerando Rankings de Rótulos

Na segunda etapa do método, cada rótulo candidato encontrado na etapa anterior é avaliado de acordo com sua representatividade para o domínio do atributo anônimo.

A Figura 3.4 ilustra as duas últimas etapas do processo de seleção de rótulos candidatos para um conjundo de dados com dois atributos anônimos (A_i,A_j).

[4] `http://jwordnet.sourceforge.net`.

[5] `http://wordnet.princeton.edu`.

[6] *Snippets* são pequenos fragmentos de texto de documentos da Web retornados como parte das respostas de uma máquina de busca onde geralmente encontram-se os termos usado na consulta.

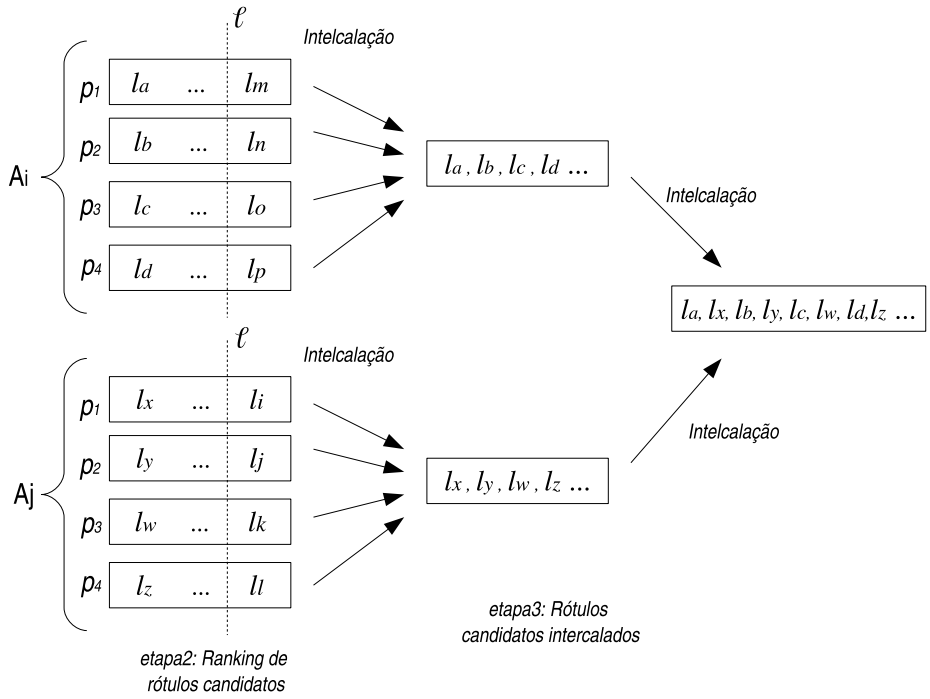

Figura 3.4: Etapas 2 e 3 do método de seleção de rótulos candidatos para os atributos anônimos A_i e A_j

A Figura 3.4 considera que a etapa 1 do método tenha encontrado um conjunto de rótulos para cada um dos quatro padrões léxico-sintáticos (p_1, p_2, p_3, p_4) aplicados aos valores dos atributos anônimos (A_i,A_j). Na segunda etapa, estes rótulos são ordenados na forma de um ranking de acordo com seu grau de representatividade para o domínio do atributo anônimo. Os critérios que avaliam esta representatividade são baseados nos requisitos para um bom rótulo candidato (definidos na primeira etapa). Ou seja, para que um termo seja um bom candidato a descritor de um atributo ele deve deve ocorrer com freqüência e o mais próximo possível da expressão de consulta nos documentos respondidos pela máquina de busca na Web. Estes critérios são implementados na forma de um *escore* numérico chamado de *fator de coincidência*, que é associado a cada rótulo candidato.

Mais especificamente, seja um valor anônimo v e um termo t no conjunto de *snippets*

S, o *fator de coincidência* de t e v em S é dado pela equação 3.1,

$$\alpha(t, v, S) = \sum_{s_i \in S_t} \frac{W_i}{d(t, v, s_i)^2} \tag{3.1}$$

onde S_t é o subconjunto dos *snippets* $s_i \in S$ que contêm o termo t na região de busca do rótulos (veja Tabela 3.2), W_i é a quantidade de termos no *snippet* e $d(t, v, s_i)$ é a distância (ou seja, o número de termos) entre t e v neste *snippet*. Observa-se que este escore é inversamente proporcional ao quadrado da distância do rótulo ao valor usado na expressão de consulta, onde a distância é medida a partir do número de termos entre eles. Dessa forma, esta estratégia de escore para escolha de rótulos candidatos é fortemente amparada em rótulos que ocorrem com freqüência em documentos respondidos por máquinas de busca e que frequentemente aparecem próximos à S. Intuitivamente, a Equação 3.1 captura o desejo por um bom rótulo, tal como discutido na seção anterior: ela favorece termos que aparecem em muitos *snippets*, mas penaliza termos "distantes" da expressão de consulta no *snippets*.

O cálculo do fator de coincidência para cada rótulo candidato gera listas ordenadas baseado no seu valor (ou seja, os *rankings* de rótulos candidatos vistos na Figura 3.4). Por razões práticas, limita-se cada uma destas listas para um máximo de ℓ rótulos candidatos. (Em todos os experimentos relatados no Capítulo 5 foi usado $\ell = 10$).

3.3.3 Intercalando Listas de Rótulos Candidatos

Após a formação dos *rankings* em cada atributo anônimo, o método intercala estas listas mantendo somente os ℓ primeiros rótulos rankeados dentre todas as listas individuais (vide Figura 3.4). Finalmente, as listas resultantes da intercalação são unificadas em uma lista final que contêm rótulos candidatos para todos os atributos do conjunto de dados anônimos.

Basicamente há dois motivos para intercalar as listas rótulos, ao invés de aplicá-las individualmente a cada atributo anônimo no método de rotulagem. Primeiro, tem sido observado em [10] que diferentes *padrões* são úteis na descoberta de diferentes tipos de

relacionamentos entre palavras. Assim, combinar os rótulos identificados pelos diferentes *padrões* é melhor do que usá-los isoladamente. Segundo, foi observado experimentalmente que algumas listas de rótulos não continham termos representativos para o seu domínio, enquanto que outras além de trazer termos representativos para o seu domínio, também traziam bons rótulos candidatos para domínios correlatos. Dessa forma, a intercalação das listas de rótulos garante uma cobertura maior aos atributos do conjunto de dados anônimos. Ou seja, há uma possibilidade menor que determinado atributo fique sem pelo menos um rótulo candidato.

3.4 Algoritmo de Seleção de Rótulos Candidatos

A Figura 3.5 mostra o algoritmo para seleção de rótulos candidatos referente ao passo 1 da abordagem proposta na Seção 1.1.

O algoritmo toma como entrada uma relação anônima R e oferece como saída um conjunto L de rótulos candidatos para R. O algoritmo itera sobre um conjunto de atributos de R (linha 1) e para cada atributo A_i, seleciona k valores (linha 3) para serem usados nas consultas a serem submetidas para cada um dos quatro *patterns* (linhas 6–17). As consultas submetidas (linha 7) são as feitas a partir dos padrões léxico-sintáticos da Tabela 3.1 e os *snippets* retornados são coletados na linha 8.

Após coletar os *snippets*, calcula-se o *fator de coincidência* (α_{cur}) para cada termo t (normalizado pelo WordNet, tal como discutido acima) encontrado em S com relação à expressão de consulta que usa o valor v (linha 10). O algoritmo acumula o chamado *fator de coincidência máximo* ($\alpha_{max}[t, v]$), obtido com todos os *patterns* (linhas 11–15). Após processar todos os valores v randomicamente selecionados para o atributo A_i, o algoritmo calcula um escore ($score[A_i, t]$) para cada termo t encontrado com relação a A_i, como sendo a soma de todos os fatores de coincidência máximo para t (linhas 18–24). O processamento termina pela atribuição de L_i aos ℓ termos com maior escore (linha 15).

A análise do algoritmo *CanditadeLabelSelection* (Figura 3.5) é simples e direta. Sua

Algoritmo *CanditadeLabelSelection*

início

Entrada: relação anônima $R(A_1, \ldots, A_n)$
Saída: conjunto L de rótulos candidatos para R

1 **para cada** $A_i \in \{A_1, \ldots, A_n\}$ **faça**
2 $T_i \leftarrow \emptyset$;
3 $V \leftarrow k$ valores distintos randomicamente selecionados de A_i in R;
4 **para cada** $v \in V$ **faça**
5 $T \leftarrow \emptyset$;
6 **para cada** *pattern* $p \in \{fwd, bwd, val, perm\}$ **faça**
7 $q \leftarrow$ query(p,v);
8 $S \leftarrow$ conjunto dos m top snippets resultantes de q;
9 **para cada** t *encontrado em snippets de* S **faça**
10 $\alpha_{cur} \leftarrow \alpha(t, v, S)$;
11 **se** $t \notin T$ **então**
12 $T \leftarrow T \cup \{t\}$;
13 $\alpha_{max}[t, v] \leftarrow 0$;
14 **fim**
15 $\alpha_{max}[t, v] \leftarrow \max\{\alpha_{max}[t, v]; \alpha_{curr}\}$;
16 **fim**
17 **fim**
18 **para cada** $t \in T$ **faça**
19 **se** $t \notin T_i$ **então**
20 $T_i \leftarrow T_i \cup \{t\}$;
21 $score[A_i, t] \leftarrow 0$;
22 **fim**
23 $score[A_i, t] \leftarrow score[A_i, t] + \alpha_{max}[t, v]$;
24 **fim**
25 $L_i \leftarrow$ t termos in T_i com top ℓ $score[A_i, t]$ valores
26 **fim**
27 $L \leftarrow L_1 \cup \ldots \cup L_n$
28 **fim**

fim

Figura 3.5: O Algoritmo para *Seleção de Rótulos Candidatos*

execução é determinada pelos seguintes parâmetros: k, o número de amostras de cada atributo anônimo (linha 3); e a quantidade de m *snippets* recuperados de cada consulta (linha 8). Seu tempo de execução é limitado por $O(n \cdot m \cdot k)$, onde n é o número de atributos anônimos no conjunto de dados.

O valor de k precisa ser alto o suficiente para assegurar que uma amostra representativa do atributo possa ser usada. Mas ao mesmo tempo não pode ser muito alto, a fim

de evitar uma quantidade excessiva de consultas à maquina de busca. Inversamente, dado que os *snippets* são ordenados pela máquina busca de acordo com a relevância dos seus documentos correspondentes, valores baixos de m poderiam indicar rótulos mais significativos. Entretanto, o uso de valores baixos para m requer que mais consultas sejam submetidas para que seja obtido um conjunto representativo de rótulos candidatos. Em todos os experimentos relatados nesta dissertação, tanto m quanto k foram ajustados para 10. Este aspecto será discutido com mais detalhes no Capítulo 5. Um último parâmetro, ℓ, determina a quantidade de rótulos candidatos que serão selecionados para cada atributo anônimo. Obviamente, o valor de ℓ tem um impacto não somente na acurácia da abordagem de rotulagem como um todo, mas também no custo do método de atribuição de rótulos (reveja a Figura 1.3). Os experimentos iniciais conduzidos indicam que $\ell = 10$ é uma boa escolha para este parâmetro.

Uma última observação na implementação do algoritmo de seleção de rótulos candidatos diz respeito à eliminação de *stop words*[7], pontuação e qualquer outro caracter especial. Os experimentos indicam que a realização deste passo acarreta em pouco efeito na acurácia do método, mas reduz o uso da memória, já que esta passa a manter quantidades menores palavras.

Este capítulo apresentou algumas hipóteses e conceitos que fundamentam o método de seleção de rótulos candidatos. Além disso, explicou-se como os padrões léxico-sintáticos podem ser utilizados para construir expressões de consultas que retornam um conjunto de documentos relevantes quando submetidas para Web. A partir dos *snippets* desses documentos da Web foi mostrado onde e como identificar rótulos candidatos. Finalmente, foi definido um algoritmo simples e eficaz que encontra um conjunto de rótulos candidatos para o conjunto de dados anônimos.

[7]Uma *stop word* no contexto desse método é qualquer palavra não identificado pelo *WordNet* como sendo um substantivo.

Capítulo 4

Método de Rotulagem Especulativa

Este capítulo apresenta o segundo passo da abordagem de rotulagem, responsável pela atribuição dos rótulos candidatos (obtidos no passo 1) a um conjunto de dados anônimos extraídos da Web. Para fazer isso, a abordagem oferece um método automático baseado num modelo probabilístico simples que realiza as atribuições de rótulos enviando determinados tipos de consultas para uma máquina de busca na Web.

O capítulo inicia com uma introdução ao problema de associação entre rótulos candidatos e atributos anônimos (Seção 4.1). Na Seção 4.2 é feita a apresentação do modelo probabilístico proposto. Em seguida, na Seção 4.3, é explicado como é feito o processo de rotulagem a partir de três etapas: (1) a construção de consultas especulativas, (2) envio de consultas especulativas para a Web e (3) o cálculo da afinidade entre rótulos e atributos. Finalmente a Seção 4.4 apresenta o algoritmo de rotulagem usado e analisa os desafios na sua implementação.

4.1 Definição do Problema

Assumindo que seja dada a seguinte relação

$$R(A_1, A_2, \ldots, A_n)$$

com n atributos anônimos $A_1, \ldots, A_n$, onde cada A_j pertence a um domínio D_j. Assumindo também que seja dada uma instância de R com t tuplas, onde todos os domínios são disjuntos entre si. Dado um conjunto $L = \{l_1, \ldots, l_m\}$ com m *rótulos candidatos* ($m > n$), o objetivo do método consiste em associar para cada A_j um rótulo $l_i \in L$ tal que l_i seja o *melhor* descritor para o atributo A_j. Observe que este problema é equivalente ao de encontrar a correspondência de peso máximo em grafos bipartidos [1], onde o conjunto de vértices são as colunas em R e os rótulos em L, respectivamente, e o peso de cada aresta (A_j, l_i) indica quão bem l_i descreve A_j.

Existem dois grandes desafios em encontrar uma boa rotulagem. Primeiro, observe que existem

$$\binom{m}{n} = \frac{m!}{(m-n)!n!}$$

formas diferentes de rotular n atributos com m rótulos candidatos, o que torna o problema intratável na prática. (Os melhores algoritmos para o problema de correspondência em grafos bipartidos com pesos executam em tempo polinomial ao tamanho do grafo [20]; assim, usando-os diretamente poderia exigir um tempo exponencial no contexto aqui apresentado). Segundo, é necessário definir uma forma de medir o quão bem uma rotulagem $R \to L^n$ descreve os (domínios dos) atributos em R. Este trabalho usa uma estratégia gulosa na atribuição dos rótulos aos atributos, ou seja, para cada atributo é associado um rótulo de forma isolada aos demais, e uma vez que este rótulo é associado a um atributo ele não é mais considerado como candidato para outros atributos.

4.2 Modelo Probabilístico

O método usa um modelo probabilístico simples que avalia a *afinidade* de um rótulo candidato com um atributo anônimo. Ou seja, ele mede a probabilidade de um rótulo descrever bem um dado atributo. Mais precisamente, define-se $P(l_i|A_j)$, que é a probabilidade do rótulo l_i descrever bem o atributo A_j, como a métrica que avalia a rotulagem de

[1] um grafo bipartido é um tipo especial de grafo onde o conjunto de vértices pode ser dividido em dois conjuntos disjuntos U e V tal que nenhuma aresta possui ligação com vértices de um mesmo conjunto.

um atributo anônimo. Como simplificação, assume-se que a probabilidade de um rótulo ser uma boa escolha para um atributo é independente de outros atributos. A noção de afinidade no contexto do método leva em consideração a incerteza em saber o quanto um rótulo descreve bem um domínio.

Observe que

$$P(l_i|A_j) = \frac{P(A_j|l_i)P(l_i)}{P(A_j)}.$$

Assim, é necessário estimar $P(A_j|l_i)$ e $P(l_i)$ para calcular a afinidade entre l_i e A_j. Analisando $P(A_j)$, constata-se que ele é somente um fator de normalização e que pode ser ignorado por razões práticas. Já a probabilidade $P(l_i)$, intuitivamente, captura a preferência do usuário pelo rótulo l_i *independente* de sua afinidade com qualquer outro atributo. Para os propósitos deste trabalho, define-se $P(l_i)$, que é a probabilidade *a priori* de l_i ser um bom rótulo para algum atributo, como a freqüência relativa de l_i dentre a lista de rótulos candidatos obitida no passo 1 da abordagem.

4.3 O Processo de Rotulagem

O método de rotulagem atribui rótulos representativos a um conjunto de dados anônimos através de um processo em três etapas: (1) construção de consultas especulativas; (2) submissão dessas consultas para uma máquina de busca na Web; e (3) cálculo dos rótulos que possuem maior *afinidade* com os valores usados nas consultas especulativas. As etapas deste processo são vistas no *passo 2* da Figura 4.1

A intuição por trás da técnica de consultas especulativas é a seguinte. Suponha que a atribuição de um rótulo l_i para um atributo A_j seja considerada melhor que a de um rótulo l_k. Dado um documento da Web d contendo informações de alta qualidade sobre um valor de A_j, as chances de d referir-se a l_i são maiores que as de d referir-se a l_k. Uma interpretação complementar desse modelo é assumir que cada documento da Web sobre determinado atributo A_j (por exemplo, um artista específico, caso o domínio de A_j for

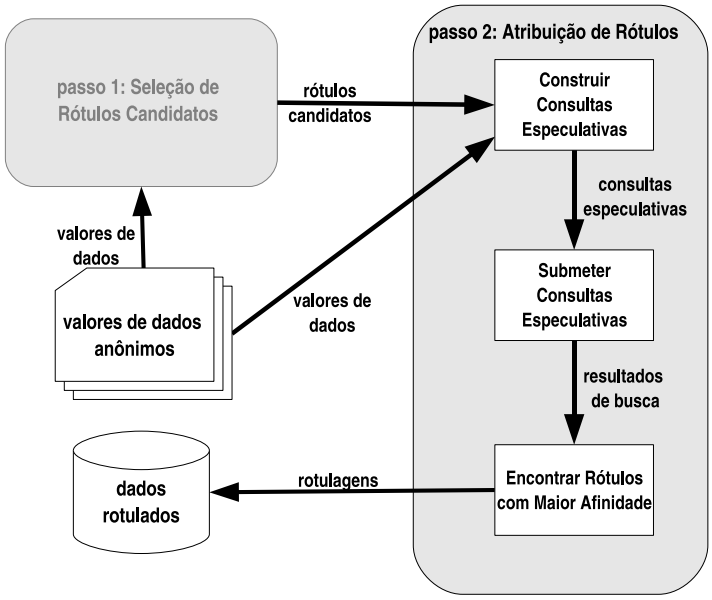

Figura 4.1: Método de Seleção de Rótulos Candidatos

nomes de artistas), seja um "especialista" na entidade representada por este valor. Dessa forma, a quantidade de documentos na resposta a uma consulta especulativa representa a quantidade de especialistas que consideram o rótulo em questão um bom descritor para o valor do atributo. Ou seja, para cada consulta especulativa submetida, o método de rotulagem precisa considerar apenas a *quantidade de documentos* (*Document Count*) respondidos pela máquina de busca. Isso simplifica o processo, pois não é necessário verificar o conteúdo desses documentos. Mais precisamente, define-se:

Definição 4.1 Dado um rótulo candidato l_i e um valor v_x de um atributo anônimo A_j, uma *Consulta Especulativa* q, é uma expressão conjuntiva definida como $l_i \wedge v_x$ que testa a probabilidade do rótulo l_i e do valor $v_x \in A_j$ ocorrerem juntos em um documento da Web.

Definição 4.2 *Document Count* de uma consulta especulativa q, definido como $DC(q)$, é a quantidade de documentos relevantes para q de acordo com uma determinada máquina de busca na Web.

Definição 4.3 Dada uma relação anônima $R(A_1, \ldots, A_n)$ e um conjunto de rótulos candidatos $L = \{l_1, \ldots, l_m\}$, a *Afinidade Rótulo-Atributo (Label-Attribute Affinity – LAA)*

entre A_j *e* l_i*, indicada por* $LAA(A_j, l_i)$*, é definida como*

$$LAA(A_j, l_i) = P(A_j|l_i) = \left(\frac{1}{|[A_j]|}\right) \sum_{x=1}^{|[A_j]|} \frac{DC(l_i \wedge v_x)}{\sum_{y=1}^{m} DC(l_y \wedge v_x)}$$

onde $[A_j]$ é o domínio ativo [2] de A_j e $v_x \in [A_j]$.

Com relação à Definição 4.1, observa-se que uma consulta especulativa nada mais é que uma indicação afirmando que determinado rótulo l_i é um bom descritor para o atributo A_j. Na Definição 4.3, observa-se que o cálculo de LAA é restrito ao *domínio ativo* de cada atributo; ou seja, valores duplicados são ignorados. Intuitivamente, isso evita enviesar a afinidade de um atributo baseada na freqüência relativa dos valores de seu domínio ativo. Já com relação à Definição 4.2, observa-se que o cálculo de $DC(q)$ pode levar ao uso intenso da máquina de busca na Web, que é uma operação custosa em termos de tempo de execução, mesmo com uma conexão rápida à Internet. Entretanto, comprova-se experimentalmente no Capítulo 5, que apenas uma pequena amostra do domínio ativo de cada atributo anônimo é suficiente para chegar a bons resultados. Dessa forma, ao invés de usar $t \cdot m$ consultas especulativas para calcular $P(A_j|l_i)$, somente uma amostra das tuplas no conjunto de dados anônimo é necessária. Ou seja, não é necessário enviar muitas consultas especulativas para definir a afinidade entre um rótulo e um atributo.

Para entender melhor o funcionamento e a intuição por trás da técnica de consultas especulativas, considere o seguinte exemplo que realiza as 3 etapas do processo de rotulagem. Seja $R(A_1, A_2)$ um conjunto de dados anônimos sobre música, $L = \{artist, title, album\}$ um conjunto de rótulos candidatos e r uma instância de R com $k = 3$ tuplas, de acordo com a Figura 4.2.

Por exemplo, se $l_i = artist$ e $v_x = Miles\ Davis$, a consulta especulativa correspondente faz a hipótese que Miles Davis seja um artista. Similarmente, se $l_i = album$ e $v_x = John\ Coltrane$, a consulta correspondente lança a hipótese que John Coltrane seja um álbum.

[2] O domínio ativo de um atributo é o conjunto de valores distintos do atributo que são usados na instância atual da banco de dados

R	A_1	A_2
	Miles Davis	Kind of Blue
	John Coltrane	A Love Supreme
	John Coltrane	My Favorite Things

Figura 4.2: Um conjunto de dados anônimos sobre música contendo a relação $R(A_1, A_2)$.

A primeira questão que surge é como formular tal hipótese usando o paradigma atual de busca baseada em palavras que é adotado em todas as principais máquinas de busca na Web. Com base nos valores do conjuto de dados anônimos da Figura 4.2, a Tabela 4.1 mostra uma comparação do número de resultados de três tipos de consultas especulativas; todas elas foram realizadas usando a máquina de busca Google em 03 de junho de 2006.

		tipos de consultas especulativas		
rótulo (l)	valor (v)	expressão	$l + v$	todas palavras
artist	Miles Davis	39.900	5.980.000	12.000.000
title	Miles Davis	220	1.940.000	16.300.000
album	Miles Davis	9.230	6.140.000	9.170.000
artist	John Coltrane	21.700	2.910.000	3.710.000
title	John Coltrane	83	1.670.000	2.720.000
album	John Coltrane	493	3.340.000	4.180.000
artist	Kind of Blue	7	425.000	25.600.000
title	Kind of Blue	117	539.000	46.700.000
album	Kind of Blue	17.900	885.000	17.900.000
artist	A Love Supreme	36	213.000	9.000.000
title	A Love Supreme	60	158.000	16.800.000
album	A Love Supreme	497	318.000	5.610.000

Tabela 4.1: Número de documentos respondidos correspodendo a diferentes tipos de consultas especulativas.

As duas primeiras colunas da Tabela 4.1 mostram uma combinação entre alguns valores $v \in r$ e cada rótulo candidato de L. A terceira coluna (expressão) mostra os valores de *Document Count (DC)* para consultas especulativas formadas pela *expressão exata* do tipo "l v". Ou seja, consultas feitas no Google do tipo "*with the exact phrase*" [3]. Por exemplo, a primeira linha demonstra que a consulta "artist Miles Davis" obteve DC=39.900. A quarta coluna ($l + v$) mostra valores de DC para consultas especulativas formadas pela *expressão conjuntiva* do tipo "$l \wedge v$". Ou seja, consultas feitas no Google do tipo "*with*

[3] `http://www.google.com/advanced_search`

all of the words". Por exemplo, a segunda linha demonstra que a consulta "title" AND "Miles Davis" teve DC=1.940.000. Finalmente, a última coluna (todas palavras) mostra os valores de DC para consultas formadas pela *expressão disjuntiva* do tipo "$l \vee v$". Ou seja, consultas feitas no Google do tipo "*with at least one of the words*". Por exemplo, a terceira linha demonstra que a consulta "album" OR "Miles" OR "Davis" teve DC=9.170.000.

A Tabela 4.2 mostra a *afinidade* (LAA) calculada de acordo com a Definição 4.3 entre os atributos anôninos (A_1 e A_2) e os rótulos candidatos em L, usando os valores de DC da Tabela 4.1.

	tipos de consultas especulativas		
afinidade	expressão	$l + v$	todas palavras
$P(A_1\|artist)$	**0.8911**	0.3964	**0.3350**
$P(A_1\|title)$	0.0043	0.1744	0.3457
$P(A_1\|album)$	0.1046	**0.4292**	0.3193
$P(A_2\|artist)$	0.0305	0.2695	0.3160
$P(A_2\|title)$	0.0538	0.2604	**0.5826**
$P(A_2\|album)$	**0.9156**	**0.4701**	0.1014

Tabela 4.2: Afinidade entre Rótulos e Atributos.

A primeira coluna representa a probabilidade de um rótulo anônimo l_i ser um bom descritor para um atributo anônimo A_j, ou seja, $P(A_j|l_i)$. Nas demais colunas são mostrados os valores calculados de LAA para cada tipo de consulta especulativa (expressão, $l + v$, todas palavras). Os valores destacados em negrito são os que alcançaram o maior escore de LAA para o atributo, ou seja, os que indicam maior afinidade entre rótulo e atributo. Mais especificamente, como se pode observar na segunda coluna, usando a consulta especulativa do tipo "expressão" pode-se alcançar uma rotulagem muito boa (ambas as atribuições estão corretas) com apenas duas tuplas do conjunto de dados anônimos. Na terceira coluna, o uso de consultas do tipo "$l + v$" atinge somente uma atribuição correta de rótulo (que A_2 contém álbuns), embora o faça com uma baixa confiança. Já, o uso da abordagem "todas palavras", na quarta coluna, oferece uma rotulagem menos precisa ainda para a atribuição correta de A_1. Mesmo que alguém questione que o domínio de A_1 contenha títulos de álbuns, ainda assim, a rotulagem do tipo *expressão* possui uma

confiança bem maior. Uma razão para que o uso da abordagem "todas palavras" tenha um desempenho fraco é que consultas especulativas construídas dessa forma tendem a ser muito genéricas; dessa maneira, constata-se o motivo de seus valores de DC serem tão altos. Nos experimentos realizados foi descoberto que as abordagens "$l + v$" e "expressão" atingem altos valores de DC para hipóteses avaliadas como *corretas*. Entretanto, "$l + v$" não faz boa distinção entre hipóteses *incorretas*. Isto acontece porque documentos da Web com alta autoridade sobre determinado domínio tendem a possuir muitos dos rótulos candidatos. Por exemplo, um documento com a discografia do artista Miles Davis provavelmente terá todos os rótulos para o exemplo em questão.

Conclui-se que o tipo de consulta especulativa "expressão" é o que melhor evidencia as hipóteses de rotulagem corretas, além de ser o mais adequado para destacar hipóteses incorretas. Dessa forma, todos os resultados experimentais discutidos no Capítulo 5 foram obtidos usando consultas especulativas deste tipo.

4.4 O Algoritmo de Rotulagem Especulativa

A Figura 4.3 mostra o algoritmo de rotulagem baseada em consultas especulativas para rotular relações anônimas.

O algoritmo funciona da seguinte maneira. Ele itera sobre todos os atributos da relação anônima R, atribuindo rótulos para cada um deles, um de cada vez de forma independente dos outros (ou seja, numa estratégia gulosa). Uma vez que um rótulo do conjunto de rótulos candidatos L é atribuído a um atributo, ele torna-se indisponível para qualquer outro atributo (linha 17). Observe que para encontrar um rótulo para um atributo A_i, é necessário ignorar valores duplicados para A_i; caso contrário valores populares poderiam desvirtuar a rotulagem. A rotulagem é realizada obtendo k amostras de valores distintos de um atributo A_i pertencente a uma relação R (linha 3).

Para cada valor v é calculado o $DC(l_j \wedge v)$ para cada rótulo candidato l_j ainda disponível (Linhas 4–10). O algoritmo usa $LVC[i, j]$ para manter a soma da contagem de documentos (DC) de todas as consultas especulativas usando um rótulo l_j e um valor v

Algoritmo *SpeculativeQueryLabeler*

```
    início
        Entrada: relação anônima R(A_1, ..., A_n)
                 conjunto L de rótulos candidatos para L = {l_1, ..., l_m}
        Saída: rotulagem de R
1       para cada A_i ∈ {A_1, ..., A_n} faça
2           b ← 1; sum ← 0;
3           V ← k valores distintos randomicamente selecionados de A_i in R;
4           para cada v ∈ V faça
5               para cada l_j ∈ L faça
6                   s ← speculative query l_j ∧ v;
7                   LVC[A_i, l_j] += DC(s);
8                   sum += LVC[A_i, l_j];
9               fim
10          fim
11          para cada L_j ∈ L faça
12              LAA[A_i, l_j] = LVC[A_i, l_j]/sum;
13              se (P(l_j) · LAA[A_i, l_j] > P(l_b) · LAA[A_i, l_b]) então
14                  b ← j;
15              fim
16          fim
17          L ← L − {l_b};
18          result = result ∪ {(A_i, l_b)};
19      fim
20      retorna result;
    fim
```

Figura 4.3: O Algoritmo de *Rotulagem Baseada em Consultas Especulativas*

do atributo A_i.

Quando este passo é completado, o algoritmo calcula a afinidade entre A_i e cada rótulo l_j através do valor de $LAA[A_i, l_j]$ (linha 12). Durante este processo mantém registro do rótulo com a maior afinidade com o atributo corrente (isso é feito na variável b, nas linhas 13–15). Na linha 17 o melhor rótulo é removido do conjunto de rótulos candidatos e a saída é construída na linha 18.

A análise do algoritmo de rotulagem especulativa é a seguinte. Seja n o número de atributos numa relação anônima, m o número de rótulos candidatos e k o número de amostras tomadas de valores de um dado atributo A_i. É fácil observar que, no pior caso, o algoritmo executa no tempo $O(n \cdot m \cdot k)$. Entretanto, uma característica interessante do

método é que seu algoritmo pode atingir uma alta acurácia mesmo quando uma pequena amostra de valores de dados é usada. Isso é importante porque reduz a quantidade de consultas especulativas a serem submetidas na Linha 6. Realmente, tal como discutido anteriormente, os resultados dos experimentos atingiram boa acurácia com menos de 10 valores de amostra por atributo.

Obviamente, quanto menos consultas especulativas forem usadas para rotular um atributo, melhor. Nos experimentos realizados, o tempo de resposta para executar uma consulta especulativa usando o serviço da Web para consultas do *Yahoo!* [4] foi inferior a 1.5 segundos, mas com uma alta variância.

Este capítulo apresentou um método que realiza a rotulagem de dados anônimos enviado consultas para a Web para estimar a afinidade entre um conjunto de rótulos candidatos e valores de atributos anônimos. Além disso, explicou o funcionamento de um algoritmo que implementa o método. No próximo capítulo são mostrados os resultados experimentais que validam este método como uma solução eficaz para o problema de rotulagem.

[4] `http://developer.yahoo.com/search/index.html`

Capítulo 5

Experimentos

Esta seção apresenta resultados experimentais que confirmam a viabilidade da abordagem e avaliam a efetividade dos dois métodos propostos. Experimentos realizados com 8 domínios de aplicação mostram uma acurácia do método de rotulagem acima de 90% para domínios populares e acima de 80% para domínios não-populares. A avaliação do método de seleção de rótulos candidatos demonstra níveis de revocação superiores a 85%. Já quando avaliados conjuntamente, os métodos mostram uma acurácia média acima de 80%.

O capítulo está dividido da seguinte forma: a Seção 5.1 explica a composição dos experimentos e identifica os elementos e os procedimentos comuns adotados em cada experimento; a Seção 5.2 descreve os experimentos que avaliam o isoladamente o funcionamento do método de rotulagem; a Seção 5.3 descreve os experimentos que avaliam o método de seleção de rótulos candidatos e, finalmente, a Seção 5.4 mostra os resultados alcançados pelos dois métodos atuando conjuntamente.

5.1 Configuração dos Experimentos

Para testar a eficácia da abordagem em diferentes cenários foram usados 8 conjuntos de dados anônimos de diferentes domínios de aplicação totalizando 52 atributos. Os conjuntos de dados usados nos experimentos estão listados na Tabela 5.1. O objetivo

coleção de páginas		extração		l
domínio	site	t	a	
livros	www.amazon.com	900	8	15
jogos	www.allgame.com	500	5	21
medicamentos	www.vitacost.com	615	7	22
filmes	www.allmovie.com	700	5	20
música	www.allmusic.com	600	4	17
posters	www.postershop.com	510	8	14
times	www.fifaworldcup.com	32	10	18
relógios	www.watchzone.com	550	5	18

Tabela 5.1: Conjuntos de dados utilizados nos experimentos acompanhados do número correspondente de tuplas (t), atributos (a) e rótulos candidatos (l).

na seleção de atributos foi de maximizar sua diversidade. Dessa forma, foram utilizados conjuntos de dados de diferentes domínios e selecionados diversos tipos de atributos: com muitos valores distintos, com poucos valores distintos, com valores numéricos, com valores textuais, etc. Para levar em consideração a possibilidade de interferência de processos de extração de dados, utilizou-se a ferramenta de extração semi-automática DESANA [6] para extrair dados de bancos de dados da Web. A Figura 5.1 exibe uma amostra de tuplas de três dos conjuntos de dados utilizados nos experimentos juntamente com os rótulos candidatos considerados para cada conjunto.

Os experimentos foram divididos em três partes. Na primeira parte (Seção 5.2) são apresentados um conjunto de experimentos que estudam o comportamento do método de rotulagem usando rótulos oferecidos manualmente. Na segunda parte (Seção 5.3) é estudada a eficácia do método de seleção de rótulos candidatos. Finalmente, na terceira parte (Seção 5.4), são relatados os experimentos que avaliam os dois métodos trabalhando em conjunto.

Todos os resultados de acurácia são dados através da comparação da rotulagem produzida pelo algoritmo com uma avaliação manual feita por especialista. Para a submissão de consultas especulativas foram realizados experimentos com as APIs (*Application Program Interfaces*) de busca tanto do Google, quanto do Yahoo!, obtendo resultados similares. Entretanto todos os resultados relatados nesta dissertação foram gerados usando o Serviço Web de busca do Yahoo!.

A1	A2	A3	A4	...
Structured Computer Organization	Andrew S. Tanenbaum	Hardcover	June 15, 2005	...
Fundamentals of Database Systems	Ramez Elmasri, Shamkant B. Navathe	Hardcover	July 23, 2003	...
Software Requirements	Karl E. Wiegers	Paperback	February 26, 2003	...
Modern Information Retrieval	Ricardo Baeza-Yates, Berthier Ribeiro-Neto	Paperback	May 15, 1999	...

Amostra do conjunto de dados de livros. Somente quatro dos oito atributos são mostrados. Rótulos candidatos: author, buy new, date, used & new from, format, isbn, language, list price, publication date, publisher, price, rate, subject, title and type.

A1	A2	A3	A4	A5
Romance	5 Stars	1965	Doctor Zhivago	David Lean
Comedy	5 Stars	1936	Modern Times	Charles Chaplin
Action	4.5 Stars	1981	Raiders of the Lost Ark	Steven Spielberg
Epic	4 Stars	1960	Spartacus	Stanley Kubrick

Amostra do conjunto de dados sobre filmes. Rótulos Candidatos: actor, box office, certification, company, country, director, directed by, distributor, film, genre, language, movie, release, rating, rank, starring, theatrical run, title, votes, year.

A1	A2	A3	A4	A5
Armani	AR5447	Ladies	Stainless steel bracelet	$195.00
Seiko	SDWG32	Men	Stainless steel Butterfly clasp	$400.00
Longines	L51580966	Petite	Stainless Steel Bracelet	$1700.00
Casio	DW5600E1V	Men	Plastic, black	$70.00

Amostra do conjunto de dados sobre relógios. Rótulos Candidatos: band, brand, category, condition, description, display, gender, features, material, movement, model, price, price range, savings amount, size, style, title, type.

Figura 5.1: Amostras de tuplas de três conjuntos de dados com os rótuos candidatos.

5.2 Eficácia da Rotulagem Especulativa

Esta seção apresenta três experimentos que avaliam o funcionamento do método de rotulagem. O primeiro experimento (Subseção 5.2.1) verifica a eficácia da medida LAA – *Label-Attribute Affinity* em determinar o rótulo correto para um atributo anônimo. O segundo experimento (Subseção 5.2.2) avalia o impacto do tamanho da amostra de valores anônimos em função do nível de acurácia alcançado. Já o terceiro experimento (Subseção 5.2.3) analisa a quantidade de consultas especulativas e o tempo necessário para realizar a rotulagem de um atributo anônimo.

A fim de isolar o estudo do método de rotulagem de aspectos relacionados ao método de seleção de rótulos candidatos (passo 1 da abordagem), foram utilizados conjuntos de rótulos candidatos manualmente extraídos de páginas da Web. Dessa forma, para cada conjunto de dados, foram obtidos rótulos candidatos provenientes de 10 Web sites distintos do mesmo domínio de aplicação. Como pode ser visto na Tabela 5.1, em todos os testes

realizados foram usados mais rótulos candidatos que a quantidade de atributos anônimos no conjunto de dados. Além disso, foram usados somente rótulos candidatos que fossem suficientemente populares na Web; para cada rótulo candidato l_i, foi assegurado haver $P(l_i) > \varepsilon = 1$ milhão. A preferência *a priori* por estes rótulos, $P(l_i)$, foi definida para 1 como medida de simplificação.

5.2.1 Avaliação da Medida LAA

Inicialmente, são apresentados resultados detalhados para um dos conjuntos de dados anônimos, em seguida é oferecido um resumo dos resultados obtidos em todos os demais. A Figura 5.2 mostra um gráfico com valores de LAA calculados a partir de 50 execuções do algoritmo para cada um dos 5 atributos anônimos do conjunto de dados sobre relógios. Neste experimento, foram usados 18 (dezoito) rótulos candidatos distintos (observe a Tabela 5.1).

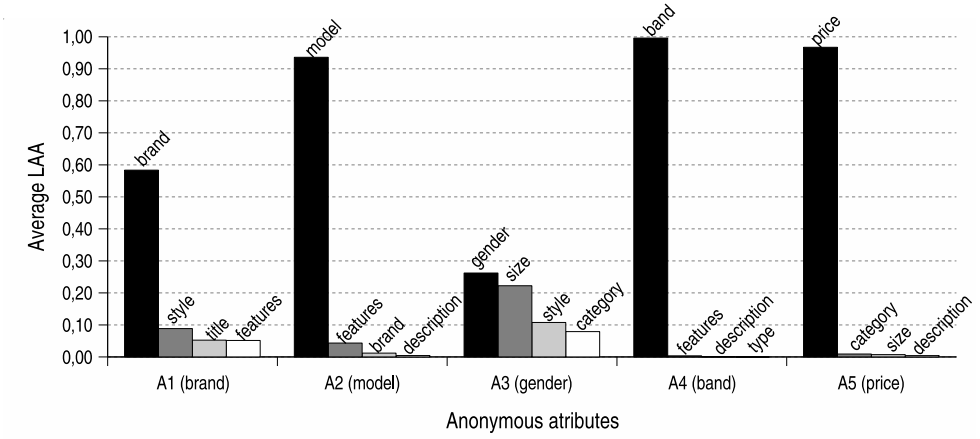

Figura 5.2: 4 primeiros valores de LAA para 18 rótulos candidatos com 5 atributos anônimos do conjunto de dados sobre relógios.

A Figura 5.2 mostra, para cada atributo anônimo Ai, uma barra vertical mostrando o valor médio alcançado pela medida LAA para um determinado rótulo, isso somente para os quatro rótulos com maior valor de LAA. O rótulo correto para cada atributo anônimo A_j está indicado entre parênteses no eixo X. Para o atributo $A1$, por exemplo, o rótulo

brand tem LAA igual a 0.58, enquanto que o rótulo style alcança 0.09. Em todos os casos o rótulo com maior valor de LAA corresponde a um rótulo correto.

Também é observado que na maioria dos casos o maior valor de LAA é, destacadamente, maior que o segundo valor de LAA obtido para o atributo. Uma excessão é observada no atributo A3, onde os três maiores valores de LAA são obtidos para os rótulos gender (0.26), size (0.22) e style (0.10). Este resultado não está incorreto já que, para o domínio de relógios, estes três rótulos representam conceitos similares. Por exemplo, valores típicos para o rótulo size neste domínio são[1]: "ladies","man", etc. Dessa forma, qualquer um destes rótulos poderiam ser corretos para rotular este atributo.

Além disso, este mesmo comportamento identificado no conjunto de dados de relógios também foi observado em atributos de todos os outros conjuntos, como pode ser visto na Tabela 5.2.

Attr	LAA	**livros**	**jogos**	**medicam.**	**filmes**	**música**	**posters**	**times**	**relógios**
A1	1^{st}	0,8944	0,7452	0,5976	0,5761	0,5500	0,4716	0,6382	0,5835
	2^{nd}	0,0528	0,0673	0,1651	0,2130	0,1317	0,1120	0,1491	0,0887
A2	1^{st}	0,9100	0,3413	0,4522	0,9208	0,6311	0,5994	0,4330	0,9360
	2^{nd}	0,0793	0,3279	0,1204	0,0384	0,2418	0,1764	0,1810	0,0435
A3	1^{st}	0,9172	0,3500	0,5438	0,7539	0,8915	0,6356	0,5946	0,2622
	2^{nd}	0,0366	0,2601	0,1894	0,1191	0,0343	0,1075	0,2741	0,2225
A4	1^{st}	0,9541	0,5893	0,7104	0,3248	0,6677	0,7697	0,6368	0,9959
	2^{nd}	0,0375	0,1319	0,0513	0,2929	0,1095	0,1218	0,1920	0,0035
A5	1^{st}	0,4444	0,2950	0,4365	0,4936	–	0,6807	0,4923	0,9672
	2^{nd}	0,2778	0,2161	0,3673	0,4358		0,1243	0,3793	0,0092
A6	1^{st}	0,7802	–	0,5198	–	–	0,9252	0,7517	–
	2^{nd}	0,1019		0,3189			0,0628	0,1038	
A7	1^{st}	0,7855	–	0,3990	–	–	0,7683	0,9128	–
	2^{nd}	0,1078		0,2904			0,1341	0,0190	
A8	1^{st}	0,3659	–	–	–	–	0,6673	0,7339	–
	2^{nd}	0,2855					0,1226	0,0552	
A9	1^{st}	–	–	–	–	–	–	0,5704	–
	2^{nd}							0,3354	
A10	1^{st}	–	–	–	–	–	–	0,4100	–
	2^{nd}							0,3375	

Tabela 5.2: Maior (1^{st}) e segundo maior (2^{nd}) valor de LAA para todos atributos anônimos. Em todos os casos, o rótulo com o maior valor de LAA é correto

Nesta tabela, é apresentado o maior (1^{st}) e o segundo maior (2^{nd}) valor de LAA obtidos para cada atributo. Novamente, em todos os casos, os rótulos com maior valor de LAA indicam atribuições de rótulos corretas. Além disso, foram identificados casos onde mais de um rótulo poderia ser apropriado para determinado atributo. Isto aconteceu,

[1] Veja `http://watchzone.com` como um exemplo.

por exemplo, com os atributos: A8 em livros com os rótulos list price (0.3659) e buy new (0.2855); A3 em jogos com os rótulos genre (0.3500) e category (0.2601); e A4 em filmes com os rótulos film (0.3248) e movie (0.2929). Em todos os casos, ambos os rótulos correspondem a conceitos similares.

5.2.2 O Impacto do Tamanho da Amostra

Este segundo experimento avalia o impacto do número k de amostras de valores distintos (vide Seção 4.4) em cada conjunto de dados anônimos e os níveis de acurácia obtidos nas rotulagens. Neste experimento, o processo de rotulagem é executado para cada atributo em todos os conjuntos de dados de acordo com a Tabela 5.1, usando $k = 1$, 3, 5, 7 e 9 valores distintos. Para cada k, o processo de rotulagem foi repetido 50 (cinqüenta) vezes, calculando o valor de LAA para cada um deles. Em seguida, a acurácia foi calculada como a porcentagem de vezes na qual o rótulo que atingiu o maior valor de LAA corresponde a um rótulo correto para o atributo. A Figura 5.3 apresenta os resultados deste experimento para todos os conjuntos de dados, onde a medida "accuracy" corresponde à média da acurácia de todos os atributo do conjunto de dados.

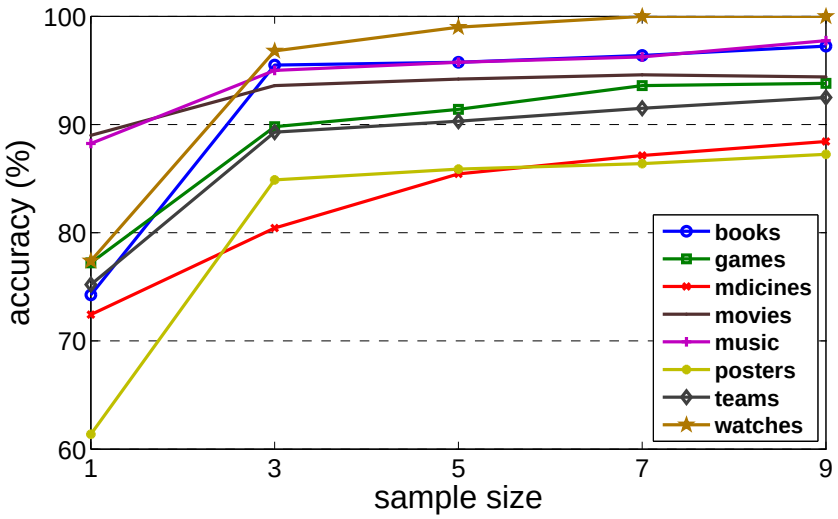

Figura 5.3: Acurácia versus o tamanho da amostra para todos os conjuntos de dados.

O gráfico na Figura 5.3 mostra no eixo X ("sample size") que apenas 3 amostras de

valores de dados foram suficientes para atingir uma acurácia média em torno de 90% para os conjuntos de dados nos domínios relógios, música, livros, filmes, jogos e times; onde todos estes possuem popularidade razoável na Web. Já para os conjuntos de dados nos domínios medicamentos e posters, que podem ser considerados menos populares na Web, a acurácia esteve acima de 80%, também usando três amostras de dados. Quando o número de amostras de valores cresce, a Figura 5.3 mostra que estas porcentagens são mantidas ou ligeiramente aumentadas . Com relação aos conjuntos de dados nos domínios medicamentos e posters, foi observado que, para muitos dos seus valores, a consulta especulativa correspondente não encontrou documento algum para alguns rótulos candidatos. Isso pode ser explicado pelos problemas que ocorrem com alguns valores da dados nos atributos anônimos (por exemplo, palavras com erros de grafia), ou mesmo pelo fato desses conjuntos de dados, como mencionado acima, virem de domínios não-populares na Web.

Em resumo, os resultados acima indicam que o método pode ser bastante efetivo para diversos domínios (principalmente aqueles que são mais populares na Web), mesmo quando usa poucas amostras dos conjuntos de dados anônimos.

5.2.3 O Número de Consultas Especulativas

Neste terceiro experimento são apresentados os resultados que mostram o comportamento geral do algoritmo de rotulagem com relação ao número de consultas especulativas necessárias para alcançar uma atribuição correta de rótulo, além dos tempos de execução.

Considerando os mesmos experimentos relatados na subseção 5.2.1, a Tabela 5.3 mostra, para cada conjunto de dados, o número médio de consultas especulativas (**q**) e o tempo médio de execução (**t**, em segundos) gasto para executar o processo de rotulagem para um dado atributo, para cada quantidade k de amostras de valores.

Lembrando que de acordo com o algoritmo de rotulagem especulativa (Seção 4.4), ambos **q** e **t** são proporcionais ao número de k amostras usadas e ao número de rótulos candidatos disponíveis. Além disso, o tempo médio gasto com cada consulta especulativa

	Amostras de valores de dados									
	1		**3**		**5**		**7**		**9**	
domínio	**q**	**t**	**q**	**t**	**q**	**t**	**q**	**t**	**q**	**t**
livros	12,75	22,71	34,50	55,21	56,13	85,96	79,00	111,89	111,13	142,00
jogos	20,27	21,77	57,00	66,86	95,00	98,50	129,25	169,88	169,55	245,85
medicamentos	20,41	29,60	61,24	94,30	99,54	142,34	139,36	199,28	179,17	238,18
filmes	18,00	25,05	54,00	62,73	90,00	114,05	122,60	135,83	144,20	166,54
música	12,50	26,45	37,50	69,31	64,75	117,82	84,50	153,49	101,25	177,55
posters	10,50	16,74	31,93	42,82	48,85	56,84	68,29	74,36	81,56	116,59
times	17,01	28,30	51,02	84,91	85,03	125,85	115,98	172,90	146,05	199,31
relógios	16,60	24,90	50,78	62,49	80,00	97,24	110,63	138,29	132,19	159,81
médias	**16,00**	**24,44**	**47,24**	**67,33**	**77,41**	**104,83**	**106,20**	**144,49**	**133,14**	**180,73**

Tabela 5.3: Resumo do comportamento geral do algoritmo de rotulagem para 8 conjuntos de dados.

foi de 1,4 segundos. [2] Como pode ser observado nas colunas 3 e 4 da Tabela 5.3, usando 3 amostras de dados o processo de rotulagem de cada atributo consome, em média, 47,24 consultas com duração média de cerca de 1 minuto. Observe que, como esperado, usando 9 amostras de valores foram necessárias em torno de 3 vezes mais consultas especulativas, com um aumento proporcional no tempo de execução. Entretanto, o ganho em corretude quando passou-se a usar de 3 para 9 amostras foi menor que 4% em média. A partir desta constatação, pode-se concluir que um pequeno número de amostras de valores é necessário para atingir bons níveis de corretude na rotulagem.

Como uma observação final, pode-se argumentar que o tempo gasto com o processo de rotulagem, que fica em torno de 1 minuto, em média, é maior que a quantidade típica de tempo gasto por métodos que tentam encontrar rótulos localmente (nos documentos da Web de onde os dados foram extraídos). Entretanto, deve-se destacar que os requisitos de tempo do método aqui apresentado estão na mesma escala dos requisitos de tempo de outras tarefas relacionadas aos métodos mencionados acima. Por exemplo, a coleta das páginas alvo da extração, a geração dos *wrappers* e a extração dos dados. Dessa forma, aplicar o método não representa uma sobrecarga inconveniente ao processo de extração de rotulagem. Além disso, enfatiza-se que o método, contrariamente às abordagens de rotulagem concorrentes, não exige que os rótulos estejam presentes nos documentos, páginas ou sites de onde os dados foram extraídos.

[2] Consultas submetidas através de link da RNP, PoP-Manaus.

5.3 Eficácia da Seleção de Rótulos Candidatos

Nesta seção são apresentados resultados do segundo grupo de experimentos que avaliam o método de seleção de rótulos candidatos.

Como discutido no Capítulo 3, este método tem o propósito de gerar uma lista de rótulos representativos que sirvam de entrada para o método de rotulagem. A fim de avaliar o método, duas métricas foram aplicadas ao problemas na forma de uma adaptação de métricas clássicas de recuperação de informação. Antes de apresentar os resultados, estas duas métricas são definidas.

Label MRR (LMRR) Esta métrica é uma adaptação da conhecida métrica *Mean Reciprocal Ranking* (MRR) [7]. A métrica MRR é apropriada para avaliar funções de ranking, tal qual o escore para rótulos candidatos definido na Subseção 3.3.2, num contexto onde o número de elementos relevantes no ranking é esperado que seja pequeno. Na maioria dos casos, MRR é usado em substituição à métrica *precisão*. Ao invés de medir a razão dos elementos relevantes no ranking, ela mede o quão distante o primeiro elemento relevante no ranking está da primeira posição. MRR é usado, por exemplo, para avaliar Sistemas de Perguntas e Respostas (*Q&A Systems*), onde o número de respostas corretas para uma dada questão é geralmente pequeno.

Para um determinado domínio representado por um conjunto de dados anônimos $R(A_1, \ldots, A_n)$, calcula-se a métrica LMRR de acordo com a equação 5.1, onde $L(i)$ é o ranking de rótulos candidatos com relação ao atributo A_i (veja a Subseção 3.3.2) e $f(L_i)$ é a posição do rótulo plausível para A_i mais próximo do topo neste ranking, de acordo com julgamento humano.

$$\mathrm{LMRR}(R) = \frac{\sum_{A_i \in R} \frac{1}{f(L_i)}}{n} \tag{5.1}$$

Observa-se que valores de LMRR próximos a 1 implicam que há um rótulo plausível numa posição próxima do topo do ranking e que os valores de LMRR são exponencialmente

reduzidos na medida em que a posição do primeiro rótulo plausível aumenta.

Label-Attribute Recall (LAR) Esta métrica é aplicada para o conjunto de rótulos L resultantes do método de seleção de rótulos candidatos. Ela avalia o quão bem os rótulos em L cobrem os atributos a serem rotulados. Para um dado domínio representado por um conjunto de dados anônimos $R(A_1, \ldots, A_n)$, calcula-se o valor LAR de acordo com a Equação 5.2, onde $\{A'_1, \ldots, A'_m\}$ é o conjunto de atributos de R que têm pelo menos um rótulo plausível em L.

$$\mathrm{LAR}(R) = \frac{m}{n} \tag{5.2}$$

Os resultados na Tabela 5.4 mostram valores médios de LMRR e LAR obtidos para 10 execuções do método sobre cada domínio.

Domínio	Méd. LMRR	Méd. LAR	Méd. rótulos candidatos
livros	0.75	97.50%	60.4
jogos	0.53	86.00%	41.5
medicamentos	0.52	87.14%	47.4
filmes	0.78	92.00%	37.9
música	0.64	97.22%	33.6
posters	0.37	88.75%	57.4
times	0.82	98.00%	79.0
relógios	0.72	96.00%	37.2

Tabela 5.4: Avaliação do Método de Seleção de Rótulos Candidatos.

Observe que para cada execução um conjunto diferente de rótulos candidatos foi usado (Veja o Algoritmo na Seção 3.5). Também é mostrado a quantidade média de rótulos candidatos obtidos nas execuções de cada domínio.

Como mostrado pelos valores da métrica LMRR na Tabela 5.4, em média, um rótulo plausível ocorreu antes da terceira posição no ranking de todos os domínios testados. Isso corrobora as premissas relacionando a co-ocorrência de valores e rótulos de um dado atributo e mostra que a função de escore funciona bem na prática.

Considerando valores de LAR, a Tabela 5.4 mostra que na grande maioria dos casos, um rótulo plausível foi encontrado para os atributos anônimos nos 8 domínios considerados. Por exemplo, considere o domínio jogos, para o qual o menor valor de LAR (86%) foi obtido. Isto significa que das 50 execuções, 10 para cada um dos 5 atributos, somente em 7 casos houve falhar na selecão de um rótulo plausível. Apesar destes casos, o método foi bastante efetivo em encontrar bons rótulos para os atributos anônimos.

5.4 Seleção e Atribuição de Rótulos

Neste terceiro e último grupo de experimentos buscou-se avaliar conjuntamente os métodos de seleção de rótulos candidatos e de atribuição de rótulos. Para isso, o processo de rotulagem foi complementarmente executado para todos os domínios usando $k = 3$ (limiar definido na subseção 5.2.3). Este processo foi repetido 10 vezes usando conjuntos distintos de rótulos candidatos obtidos a partir dos experimentos da Seção 5.3. A Tabela 5.5 apresenta, para estas 10 execuções, os valores médios de acurácia (**"acurácia"**), o número médio de consultas especulativas usadas (**"q"**) e o tempo de execução médio (**"t"**, em segundos) gasto ao executar o processo de rotulagem para um dado atributo.

domínio	**acurácia**	**q**	**t**
livros	91.25%	141.70	139.47
jogos	84.00%	119.10	110.07
medicamentos	72.86%	139.70	132.06
filmes	90.00%	108.30	105.12
música	92.50%	96.30	67.33
posters	63.75%	163.20	158.10
times	90.00%	213.78	235.09
relógios	82.00%	115.20	114.30
médias	83.30%	137.16	132.69

Tabela 5.5: Acurácia média atingida, consultas necessárias e tempo gasto para o processo de rotulagem usando rótulos selecionados automaticamente.

Como pode ser observado, os resultados de acurácia da Tabela 5.5 são consistentes com os apresentados na Figura 5.3. Nota-se, por exemplo, um comportamento similar para os

valores de acurácia encontrados para os domínios *posters* e *medicamentos*. Em adição, o número de consultas enviadas e o tempo gasto mostram um comportamento similar ao observado na Tabela 5.3. Deve ser considerado, entretanto, que o número de rótulos candidatos usados neste experimento é bem maior, como apresentado na Tabela 5.4. Como conseqüência, o processo de rotulagem necessita de mais consultas e consome mais tempo. Por exemplo, há uma média de de 79 rótulos selecionados para o domínio times. Conseqüentemente, 213.78 consultas foram necessárias em média para completar o processo de rotulagem, o que levou em média 235.09 segundos, ou menos que 4 minutos. Este é o pior caso encontrado nos experimentos. Tomando a média dos tempos de execução em todos os domínios, chega-se a uma medida em torno de 2 minutos.

Capítulo 6

Conclusão

Esta dissertação propôs uma nova abordagem automática e altamente efetiva para rotulagem de conjuntos de dados anônimos extraídos da Web. Esta abordagem propôs uma soluçado em dois passos que, a partir da Web, automaticamente seleciona um conjunto de rótulos candidatos e gera um conjunto de rotulagens semanticamente representativas. A atribuição de rótulos é baseada na noção probabilística de *afinidade* entre (os domínios de) atributos anônimos e rótulos candidatos. Diferentemente de trabalhos anteriores, o modelo apresentado permite ao usuário especificar sua preferência sobre o conjunto de rótulos candidatos utilizados. Para estimar as probabilidades no modelo é usada a quantidade de documentos relevantes retornados por *consultas especulativas* feitas a uma máquina de busca na Web. Tal como demonstrado por experimentos extensivos, a alta acurácia atingida indica que o método é bastante efetivo e a baixa quantidade de consultas especulativas submetidas para rotular um atributo anônimo garante sua eficiência.

Também de forma diferente dos métodos anteriores, esta abordagem de rotulagem não requer a presença dos rótulos nas páginas da Web que contêm os dados extraídos. Com isso, permite-se que rótulos definidos pelo usuário sejam considerados como rótulos candidatos, não restringindo o usuário a aceitar somente os rótulos escolhidos pelos autores das páginas extraídas. Além disso, a abordagem oferece um algoritmo bastante efetivo para encontrar rótulos candidatos a partir de termos que são hiperonímias dos valores de dados anônimos e que possuem co-ocorrência freqüente em determinadas páginas de Web.

Os resultados obtidos em experimentos com a abordagem em 8 diferentes domínios de aplicação apontam uma revocação média acima de 85% na seleção de rótulos candidatos e acurácia acima de 80% na rotulagem usando apenas 3 amostras de valores de dados anônimos. Isto confirma que o método de rotulagem é bastante efetivo ao longo de todos os domínios, mesmo quando uma pequena amostra de valores é usada.

Como trabalhos futuros, uma linha imediata identificada é como explorar uma fonte de rótulos candidatos em repositórios de metadados existentes na Web Semântica (por exemplo, Swoogle[1]), que indexa esquemas RDF [12] e OWL [14]. Estes repositórios oferecem interfaces de busca baseadas em palavras (semelhantes às máquinas de busca na Web), que permitem ao usuário, dada uma lista de palavras-chave, recuperar uma lista ordenada de documentos sobre metadados (essencialmente esquemas hierárquicos). No contexto deste trabalho, isto deixa em aberto a questão sobre que palavras-chave utilizar como busca. Com este propósito, uma investigação está sendo feita sobre o uso de técnicas para encontrar palavras-chaves relevantes fora de documentos da Web (por exemplo, veja [15]).

Uma observação interessante obtida a partir dos experimentos realizados é que consultas especulativas em domínios populares (sites de comércio eletrônico de livros e cd's, por exemplo) retornam uma quantidade de respostas bem maior comparados com outros menos populares (posters, por exemplo). Desta observação surgem questões interessantes relacionadas com a confiança da rotulagem produzida pelo algoritmo de Rotulagem Baseado em Consultas Especulativas. Primeiro, há interesse em caracterizar formalmente o que poderia ser aceitável como um limiar para interromper a submissão de consultas especulativas. Atualmente, este é um parâmetro que precisa ser especificado pelo usuário. Segundo, dado que o algoritmo trabalha com amostras de valores dos conjuntos de dados anônimos, é necessário estudar o quanto ele é resiliente diante de distorções nestas amostras.

Outro ítem que merece uma investigação posterior é o uso da técnica de rotulagem

[1] `http://swoogle.umbc.edu/`.

especulativa em outros idiomas (tal como o português), já que há interesse em avaliar a abordagem na Web brasileira. O principal problema com o uso desta abordagem aplicada a outros idiomas poderia ser a pouca cobertura oferecida pelas máquinas de busca em determinadas línguas, o que geraria um quantidade insuficiente de respostas às consultas especulativas.

Finalmente, observa-se que poderiam ser usadas outras formas para estimar a afinidade entre atributos anônimos e rótulos candidatos. Por exemplo, algumas das heurísticas usadas em outras abordagens de rotulagem poderiam ser usadas para refinar a atribuição de rótulos no método aqui proposto.

Referências Bibliográficas

[1] Arvind Arasu and Hector Garcia-Molina. Extracting structured data from web pages. pages 337–348, San Diego, CA, USA, 2003.

[2] Luigi Arlotta, Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. Automatic annotation of data extracted from large web sites. In *WebDB*, pages 7–12, 2003.

[3] David Buttler, Ling Liu, and Calton Pu. A Fully Automated Object Extraction System for the World Wide Web. In *International Conference on Distributed Computing Systems*, pages 361–370, 2001.

[4] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. Roadrunner: Towards automatic data extraction from large web sites. pages 109–118, Rome, Italy, 2001.

[5] Davi de Castro Reis, Paulo Braz Golgher, Altigran Soares da Silva, and Alberto H. F. Laender. Automatic web news extraction using tree edit distance. pages 502–511, New York, NY, USA, 2004.

[6] Sérgio Afonso L. F. de Sá Júnior, Altigran S. da Silva, and Daniel Oliveira. DESANA: Efficiently publishing relational databases on the web by using keyword-based query interfaces, 2006. Sessão de Demonstração.

[7] Koji Eguchi, Keizo Oyama, Emi Ishida, Noriko Kando, and Kazuco Kuriyam. Evaluation Methods for Web Retrieval Tasks. *Proc. of the NTCIR-3 Workshop*, pages 5–6, 2002.

[8] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised Named-Entity Extraction from the Web: An Experimental Study. *Artif. Intell.*, 165(1):91–134, 2005.

[9] Roy Goldman and Jennifer Widom. WSQ/DSQ: A practical approach for combined querying of databases and the web. pages 285–296, 2000.

[10] Marti A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. pages 539–545, 1992.

[11] Alberto H. F. Laender, Berthier A. Ribeiro-Neto, Altigran S. da Silva, and Juliana S. Teixeira. A brief survey of web data extraction tools. *SIGMOD Record*, 31(2):84–93, 2002.

[12] Ora Lassila and Ralph R. Swick. *Resource Description Framework (RDF) Model and Syntax Specification.* World Wide Web Consortium, February 22 1999. W3C Recommendation. `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/`.

[13] Bing Liu, Robert L. Grossman, and Yanhong Zhai. Mining data records in Web pages. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 601–606, Washington, DC, USA, 2003.

[14] Deborah L. McGuinness and Frank van Harmelen. *OWL Web Ontology Language Overview.* World Wide Web Consortium, February 10 2004. W3C Recommendation. `http://www.w3.org/TR/owl-features/`.

[15] Davood Rafiei and Alberto O. Mendelzon. What is this page known for? Computing Web page reputations. *Computer Networks*, 33(1-6):823–835, 2000.

[16] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Jorunal*, 10(4):334–350, 2001.

[17] Peter D. Turney. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning*, pages 491–502, Freiburg, Germany, 2001.

[18] Jiying Wang and Frederick H. Lochovsky. Data extraction and label assignment for web databases. pages 187–196, 2003.

[19] Jiying Wang, Ji-Rong Wen, Frederick H. Lochovsky, and Wei-Ying Ma. Instance-based schema matching for web databases by domain-specific query probing. pages 408–419, 2004.

[20] Douglas B. West. *Introduction to Graph Theory.* Prentice Hall, 1996.

[21] Wensheng Wu, AnHai Doan, and Clement T. Yu. WebIQ: Learning from the Web to Match Deep-Web Query Interfaces. page 44, 2006.

[22] Yanhong Zhai and Bing Liu. Web data extraction based on partial tree alignment. pages 76–85, Chiba, Japan, 2005.

[23] Hongkun Zhao, Weiyi Meng, Zonghuan Wu, Vijay Raghavan, and Clement Yu. Fully Automatic Wrapper Generation for Search Engines. In *Proceedings of the 14th international conference on World Wide Web*, pages 66–75, Chiba, Japan, 2005.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo