Augusto Ferreira de Souza

DEEPEC: UMA ABORDAGEM PARA EXTRAÇÃO E CATALOGAÇÃO DE CONTEÚDO PRESENTE NA *DEEP WEB*

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Ciência da Computação.
Orientador: Prof. Dr. Ronaldo dos Santos Mello

Florianópolis
2013

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Souza, Augusto Ferreira de
DEEPEC: Uma abordagem para extração e catalogação de conteúdo presente na Deep Web / Augusto Ferreira de Souza ; orientador, Ronaldo dos Santos Mello - Florianópolis, SC, 2013.
63 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação.

Inclui referências

1. Ciência da Computação. 2. Extração de dados na web. 3. Bancos de dados escondidos. 4. Catalogação de dados na web. 5. Deep Web. I. Mello, Ronaldo dos Santos. II. Universidade Federal de Santa Catarina. Programa de PósGraduação em Ciência da Computação. III. Título

Augusto Ferreira de Souza

DEEPEC: UMA ABORDAGEM PARA EXTRAÇÃO E CATALOGAÇÃO DE CONTEÚDO PRESENTE NA D_EEP WEB_

Esta Dissertação foi julgada adequada para obtenção do Título de "Mestre", e aprovada em sua forma final pelo Programa Pós-Graduação em Ciência da Computação.

Florianópolis, 26 de Agosto de 2013.

_____
Prof. Ronaldo dos Santos Mello, Dr.
Coordenador do Curso

Banca Examinadora:

_____
Prof. Ronaldo dos Santos Mello, Dr.
Orientador
Universidade Federal de Santa Catarina

_____
Prof.ª Carina Friedrich Dorneles, Dr.ª
Universidade Federal de Santa Catarina

_____
Prof. Roberto Willrich, Dr.
Universidade Federal de Santa Catarina

_____
Prof.ª Viviane Pereira Moreira, Dr.ª
Universidade Federal do Rio Grande do Sul

Este trabalho é dedicado aos meus
pais, minha irmã e minha esposa.

AGRADECIMENTOS

Agradeço, à minha esposa Vanessa, pelo apoio e companheirismo durante todo o desenvolvimento deste trabalho.

À minha família pelo esforço e dedicação a fim de me darem uma educação de qualidade e por todo o suporte dado durante minha vida.

Ao grande professor Ronaldo pela orientação, paciência, incentivo e principalmente por ter acreditado em mim.

Aos professores Raimundo e Alessandro, por terem escrito a carta de recomendação para o ingresso no mestrado.

A professora Marcia Mantelli, por ter me dado apoio e me liberado sempre que foi necessário.

Aos colegas do GBD (Grupo de Banco de Dados), pelas trocas de experiências e discussões realizadas para a construção deste trabalho.

Em especial aos colegas Carlos, Guilherme, Renato, Moretto, Elaine, Tiago e Lucas que contribuíram para a realização do trabalho.

À CAPES e à UFSC que proporcionaram o desenvolvimento deste trabalho.

Aos professores Carina Dornelles, Roberto Willrich e Viviane Pereira Moreira por disponibilizarem sua atenção para o presente trabalho.

Enfim, a todas as pessoas que de alguma forma contribuíram para a conclusão desta etapa.

Se você quer ser bem sucedido,
precisa ter dedicação total,
buscar seu último limite e
dar o melhor de si.
Ayrton Senna.

RESUMO

Esta dissertação apresenta uma solução chamada *DeepEC* (*DeepWeb Extraction and Cataloguing Process*) para realizar a extração e catalogação de dados relevantes em bancos de dados presentes na *Deep Web*, também denominados de bancos de dados escondidos. Essas informações são extraídas a partir de um conjunto de páginas HTML geradas a partir de consultas definidas sobre formulários *Web*. A intenção é adquirir conhecimento sobre esses bancos de dados e, consequentemente, permitir buscas estruturadas sobre esse conteúdo escondido. Experimentos comprovaram a eficácia da abordagem proposta. Comparado com trabalhos relacionados, as contribuições desta dissertação são a realização conjunta e sequencial de um processo de extração e catalogação dos dados de bancos de dados escondidos, um processo de extração automático com suporte de uma base de conhecimento e um processo de catalogação que gera registros estruturados e é capaz de realizar a detecção de atributos cujos valores não estão presentes nos dados extraídos.

Palavras-chave: Extração de Dados na *Web*, Catalogação de Dados na *Web*, *Deep Web*, Bancos de Dados Escondidos.

ABSTRACT

This work presents an approach called DeepEC (Deep Web Extraction and Cataloguing Process) that performs the extraction and cataloging of relevant data presented in Deep Web databases, also called hidden databases. This information is extracted from a set of HTML pages generated by queries posed on web forms. The intention is to obtain knowledge about these databases and thus enable structured queries over this hidden content. Experiments have shown the effectiveness of the proposed approach. Compared to related work, the contributions of this paper are the simultaneous process of data extraction and cataloging of hidden databases, an automatic extraction process with a knowledge base support, and a cataloging process that generates structured records and it is able to detect attribute values that are missing in the extracted data.

Keywords: Web Data Extraction, Web Data Cataloging, Deep Web, Hidden Databases.

LISTA DE FIGURAS

Figura 1 – *Deep Web* versus *Web* visível. 17
Figura 2 – Exemplos de formulários de acesso ao conteúdo da *Deep Web*. 18
Figura 3 – (a) dado de entrada; (b) dado catalogado. 21
Figura 4 – Segmento de uma tabela de referência. 21
Figura 5 - (a) Exemplo de dados semiestruturados. 24
Figura 6 - Exemplo do processo de extração do *Road Runner*. 25
Figura 7 – Exemplo de árvore DOM para uma página HTML. 26
Figura 8 - Árvore DOM com a região de dados. 27
Figura 9 – Passos do JUDIE para catalogação dos dados. 28
Figura 10 – Ciclo de vida definido para a criação da BC. 31
Figura 11 – Esquema do modelo de dados na BC. 32
Figura 12 – (a) Exemplo de conteúdo da Base de Conhecimento no domínio de Automóveis. 33
Figura 13 – Exemplo parcial da BC representada no formato XML. 35
Figura 14 – Arquitetura do *DeepEC*. 36
Figura 15 – Página HTML com resultados de uma consulta a um banco de dados escondido visualizada em um browser *Web*. 37
Figura 16 – Exemplo de comparação com atributo mandatário. 40
Figura 17 – Exemplo de arquivo com os registros extraídos. 41
Figura 18 – Exemplo de detecção de valores de atributos de registros. 45
Figura 19 – Esquema do BD relacional para o armazenamento de registros extraídos da *Deep Web*. 47
Figura 20 – Exemplo de registros catalogados. 48
Figura 21– Exemplo de uma visão com todas as informações catalogadas de um registro exemplo. 48
Figura 22 – Diagrama de classe da *DeepEC*. 49
Figura 23 – Fórmulas para os cálculos de revocação, precisão e medida F. 51

LISTA DE TABELAS

Tabela 1- Comparativo das técnicas de extração de dados na *Web*. 23
Tabela 2- Comparativo das técnicas de catalogação de dados na *Web*. 29
Tabela 3- Resultados da etapa de extração dos dados. 52
Tabela 4- Resultados da extração e catalogação do *DeepEC*. 53
Tabela 5- Ganho com a detecção de informação durante a catalogação. 54
Tabela 6- Comparativo das técnicas de extração de dados na *Web*. 56

LISTA DE ABREVIATURAS E SIGLAS

BC - Base de Conhecimento
BD - Banco de Dados
CRF - *Conditional Random Fields*
DeepEC - Deep Web Extraction and Cataloguing Process
DOM - *Document Object Model*
HTML - *Hyper Text Markup Language*
MDR - *Mining Data Records*
STM - *Simple Tree Matching*
URL - *Uniform Resource Locator*
UML - *Unified Modeling Language*
XML - *Extensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO **14**
2 FUNDAMENTAÇÃO TEÓRICA **17**
2.1 *DEEP WEB* 17
2.2 EXTRAÇÃO DE DADOS 20
2.3 CATALOGAÇÃO DE DADOS 20
3 TRABALHOS RELACIONADOS **22**
3.1 EXTRAÇÃO DE DADOS 22
3.2 CATALOGAÇÃO DE DADOS 27
4 DEEPEC **30**
4.1 BASE DE CONHECIMENTO 30
4.2 ARQUITETURA DA DEEPEC 36
4.3 PROCESSO DE EXTRAÇÃO 38
4.4 PROCESSO DE CATALOGAÇÃO 42
4.5 ARMAZENAMENTO DOS REGISTROS PROCESSADOS PELO COMPONENTE DE CATALOGAÇÃO 46
4.6 IMPLEMENTAÇÃO DA *DEEPEC* 48
5 EXPERIMENTOS **50**
5.1 ORIGEM DOS DADOS 50
5.2 MÉTRICAS UTILIZADAS 50
5.3 RESULTADOS 51
6 CONCLUSÃO **55**
REFERÊNCIAS BIBLIOGRÁFICAS **58**

1 INTRODUÇÃO

O aumento do volume de dados disponíveis na *Deep Web* (Halevy et al. 2009) faz com que também aumente o interesse no acesso a essas informações por parte dos usuários. A *Deep Web* representa, dentre outras fontes de dados, bancos de dados disponíveis na *Web* cuja estrutura e conteúdo tornam-se visíveis (ou parcialmente visível) apenas quando mostrados em páginas dinâmicas criadas a partir do resultado de uma pesquisa geralmente definida sobre um formulário *Web* (Bergman, 2001). O formulário *Web* é a principal interface de pesquisa para estes bancos de dados. Pelo fato destes bancos de dados estarem "ocultos" na *Web*, eles são denominados bancos de dados escondidos.

Para se ter acesso às informações de bancos de dados escondidos e seus formulários de acesso na *Web*, são necessários sistemas para a sua descoberta, como por exemplo, *focused crawlers*, *meta searchers* e sistemas de integração de dados na *Web*. Porém, após a descoberta, se faz necessária a utilização de técnicas de extração dos dados exibidos nas páginas e sua catalogação, visando facilitar o posterior acesso dos usuários a essas fontes de dados. Esta atividade de extração é complexa devido à existência de uma grande variedade de *Websites* com padrões diferenciados para a exibição do conteúdo desses bancos de dados, bem como a existência de muitas informações irrelevantes (menus, anúncios, etc.) que dificultam o reconhecimento do que realmente é relevante dentro do universo de informações que é apresentado (Meng et al. 2010).

Os trabalhos sobre extração de dados não apresentam um consenso sobre qual a melhor técnica, acarretando em diversas soluções para o problema, como por exemplo, a utilização de árvores, *wrappers*, técnicas de *machine learning*, ontologia, entre outras. Contudo, nenhuma delas trata da catalogação dos dados extraídos ((Kim et al. 2007) (Liu et al. 2000) (Hsu et al. 1998)).

Com relação às abordagens de catalogação de dados costumam utilizar um dicionário para ajudar na comparação dos dados para a catalogação, porém ficam limitadas às informações que estão disponíveis para a catalogação não realizando detecção de informações disponíveis nos dicionários ((Zhao et al. 2008) (Cortez et al. 2010) (Silva et al. 2011)).

Assim sendo, o objetivo desta dissertação é desenvolver e avaliar uma abordagem que realiza de forma sequencial e automática a extração e catalogação de dados de bancos de dados escondidos exibidos em páginas da *Web* obtidas a partir de consultas submetidas a estes bancos

de dados. O processo de catalogação possui ainda a capacidade de realizar a detecção de informações que não estão disponíveis nos dados extraídos. Esta detecção é possível com o suporte de uma base de conhecimento construída para os principais domínios presentes na *Deep Web*.

A principal justificativa para este trabalho é a criação de um banco de dados da *Deep Web* que possa servir de base para diversos serviços, como por exemplo:

- Criação de sistemas de busca na *Deep Web* a partir de seus dados/metadados;
- Criação de catálogos de bancos de dados escondidos por domínio.

Neste contexto, aplicações imediatas dos resultados desta dissertação incluem a utilização desses dados extraídos na extensão do banco de dados da máquina de busca *DeepPeep*[1](Barbosa et. al 2010) para dados da *Deep Web*, bem como a extensão do banco de dados do projeto WF-Sim (Gonçalves et. al. 2011), que permite buscas por similaridade a dados de formulários *Web*.

As principais contribuições desta dissertação são:

- Um processo de extração do conteúdo relevante existente em bancos de dados escondidos na *Deep Web* e que não estão necessariamente presentes em seus formulários;
- Um processo de catalogação e detecção de dados presentes em bancos de dados escondidos;
- Modelagem de uma base de conhecimento utilizada como suporte no processo de extração, catalogação e detecção de dados;
- Melhoria da qualidade de sistemas de busca estruturada sobre bancos de dados escondidos, uma vez que o conhecimento sobre os mesmos fica enriquecido, após a extração de novos atributos e novos valores.

Esta dissertação está organizada conforme segue. O capítulo 2 apresenta uma fundamentação teórica dos principais assuntos abordados neste trabalho. No capítulo 3, uma revisão bibliográfica é apresentada, sumarizando os principais trabalhos relacionados, suas características e limitações. O capítulo 4 descreve a base de conhecimento que fornece apoio aos processos de extração e catalogação. O capítulo 5 descreve a solução proposta nesta dissertação, intitulada *DeepEC*. No capítulo 6

[1]http://www.deeppeep.org/

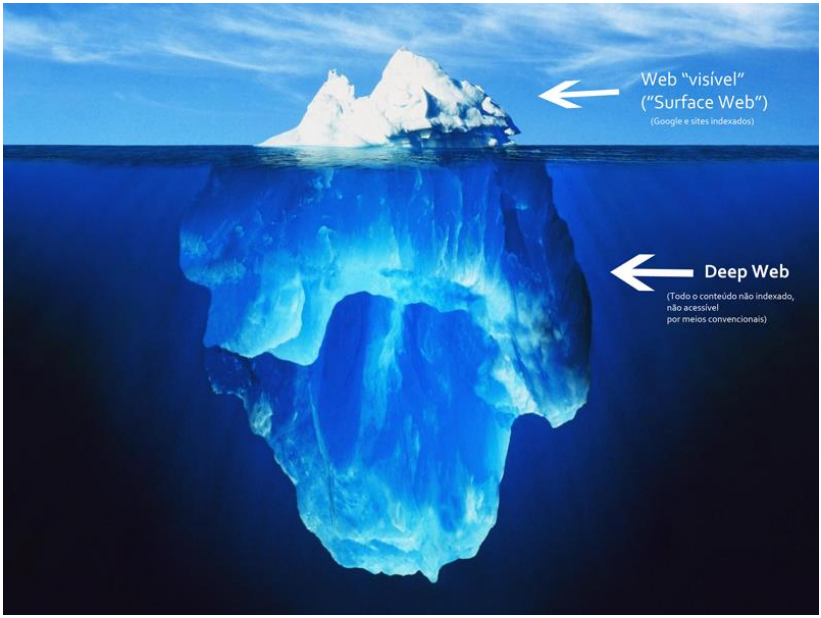
são apresentados e discutidos os resultados obtidos através de experimentos preliminares realizados. Por fim, o capítulo 7 apresenta as conclusões, bem como propostas para futuros trabalhos neste tema.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 *DEEP WEB*

A *Web* pode ser comparada a um *iceberg* (Figura 1), onde alguns dados estáticos presentes em páginas HTML são facilmente visíveis e indexáveis pelas tradicionais máquinas de busca (ponta visível do *iceberg*) (Albuquerque, 2013). Por outro lado, existe a *DeepWeb* (parte submersa e bem mais volumosa do *iceberg*), que corresponde aos dados que não estão visíveis/indexáveis e, consequentemente, são difíceis de serem encontrados. Estimativas indicam que a *Deep Web* é 500 vezes maior que a *Web* visível e que seu volume de dados está em torno de 91 mil *terabytes*.

Figura 1 – *Deep Web* versus *Web* visível.

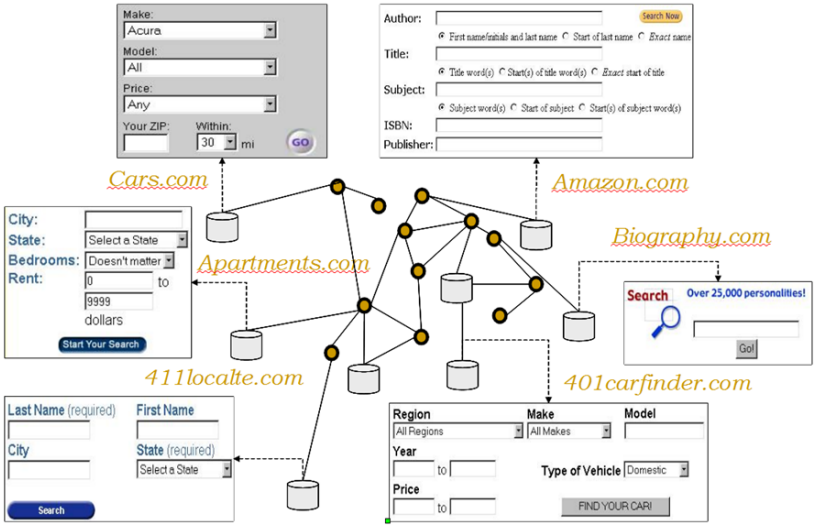


Fonte: Bergman, 2001.

Os dados na *Deep Web*, como comentado no capítulo anterior, tornam-se visíveis (ou parcialmente visíveis) apenas quando exibidos em páginas dinâmicas criadas a partir do resultado de uma pesquisa geralmente definida sobre um formulário *Web*. A Figura 2 apresenta exemplos de formulários que podem apresentar conteúdos da *Deep Web* nos mais diferentes domínios de aplicação.

Os principais modelos de formulário existentes são os de autenticação, como por exemplo, usuário e senha para acesso à conta de *e-mail* e os formulários apresentados na Figura 2, que exibem alguns campos (atributos) sobre os quais o usuário especifica filtros e então submete consultas aos bancos de dados escondidos. Estes são compostos principalmente de rótulos, valores e dependências. Os rótulos indicam aos usuários a intenção dos atributos. Os valores são as informações disponíveis para seleção em cada atributo e as dependências geralmente ocorrem nos atributos que possuem relacionamentos entre si, como por exemplo, marca e modelo no domínio de Automóveis e autor e título no domínio de Livros.

Figura 2 – Exemplos de formulários de acesso ao conteúdo da *Deep Web*.



Fonte: He et al. 2007.

O interesse pela *Deep Web* existe devido ao fato de seu conteúdo ser a principal fonte de dados estruturados na *Web* a disposição. Não aproveitar esses dados é um desperdício, pois existe a possibilidade de utilizá-los em aplicações de busca de dados na *Web* com base nos seus dados/metadados, sistemas integrados de busca/prestação de serviços e mesmo busca por formulários *Web* similares a um dado formulário, dentre outros.

Os principais tópicos de pesquisa na *Deep Web* são os seguintes:

- *Crawling*: visa descobrir bancos de dados escondidos através da procura por páginas que possuam formulários para estes bancos de dados. Um exemplo de abordagem para esse tópico é encontrar páginas HTML com a *tag* <*form*>. Resultados de trabalhos nessa área podem ser encontrados em ((Akilandeswari e Gopalan, 2007) (Raghavan e Molina, 2001) (Álvarez et al. 2007));
- Preenchimento: corresponde à seleção, geralmente automática, de valores para o preenchimento dos campos do formulário para que se obtenha acesso ao banco de dados alcançável através do formulário (Toda et al. 2010);
- Extração: corresponde à recuperação de informações, esse processo é descrito com mais detalhes na próxima seção;
- *Matching*: visa à integração dos dados previamente extraídos. Enfoques tradicionais de *matching* de bancos de dados são usados para o casamento de esquemas de formulários *Web* e a definição de afinidades entre os atributos. Esse tópico de pesquisa está presente nos trabalhos de ((Cheng e Chang, 2007) (Hong et al. 2010) (Nguyen et al. 2010));
- Consulta: corresponde ao acesso aos dados previamente extraídos e integrados. Exemplos são a máquina de busca baseadas em *keywords DeepPeep* (Barbosa et al. 2010), que indexa domínios, rótulos e valores de atributos extraídos dos formulários *Web*, bem como o sistema WF-Sim (Gonçalves et al. 2011), que realiza busca por similaridade em dados de formulários *Web*.

2.2 EXTRAÇÃO DE DADOS

A extração de dados na *Web* consiste em resgatar conteúdo de interesse apresentado de forma não-estruturada, semiestruturada ou estruturada. A dificuldade de acesso a esse conteúdo se justifica principalmente pela dificuldade de descobrir o conteúdo relevante em páginas HTML que os exibem, bem como pela forma heterogênea como são apresentados. Assim sendo, para acessar de forma facilitada esses dados, é necessário executar a sua extração e posterior catalogação.

Para (Ferrara et al. 2012), a extração de dados da *Web* é um problema importante que tem sido estudado por meio de diferentes instrumentos científicos e em uma ampla gama de domínios de aplicação. Ele é um processo considerado em diferentes problemas de gerenciamento de dados, como por exemplo, descoberta de estrutura, de dados, de domínios e buscas por similaridade (Meng et al. 2010).

Em particular, a extração de dados na *Deep Web* corresponde à aquisição de informações relevantes referentes a bancos de dados escondidos, envolvendo a recuperação de metadados (atributos) de formulários *Web* e do conteúdo presente em resultados de pesquisas geradas a partir do preenchimento de formulários *Web* e submissão de buscas a estes bancos de dados. O foco desta dissertação se enquadra nesta problemática, ou seja, na extração de registros estruturados que representam conteúdo obtido destes bancos de dados, tendo em vista que boa parte da literatura de extração de dados na *Deep Web* preocupa-se com a extração de informação presente em formulários *Web* ((Nguyen et al. 2008) (Hong et al. 2009) (Barbosa et al. 2010)) e não em páginas de resultado.

Trabalhos desenvolvidos em várias áreas de pesquisa como banco de dados, inteligência artificial, mineração de dados e recuperação de informação definem abordagens para extrair dados presentes na *Web*. Algumas dessas abordagens estão descritas no Capítulo 3.

2.3 CATALOGAÇÃO DE DADOS

Catalogar dados, no contexto deste trabalho, consiste em identificar informações de interesse disponíveis em fontes de dados, filtrar informação não relevante e classificar/armazenar informação relevante. Trata-se de um processo de indexação e descrição de recursos (dados, documentos, etc.), de modo que eles possam ser localizados e consultados (Boisson et al. 2006). Para tanto, técnicas de análise da informação são necessárias. A Figura 3 apresenta um exemplo de

catalogação, onde (a) representa uma entrada de dados e (b) o dado catalogado.

Figura 3 – (a) dado de entrada; (b) dado catalogado.

Fonte. Silva, 2011 (adaptada).

Os processos de catalogação dos dados existentes na literatura geralmente contam com o suporte de uma tabela de referência ((Agichtein e Ganti, 2004) (Zhao et al. 2008)) ou de uma base de conhecimento ((Chiang et al. 2012) (Serra et al. 2011)) nos domínios referenciados. Um exemplo da tabela de referência utilizada em (Zhao et al. 2008) para o domínio de endereços pode ser visto na Figura 4. O conhecimento presente nestas fontes é utilizado para comparação com os termos da entrada, gerando uma contextualização dos seus respectivos termos. Após a catalogação, os dados podem ser consultados pelos usuários com uma semântica mais precisa, com possibilidade de buscas mais refinadas.

Figura 4 – Segmento de uma tabela de referência.

BUSINESS	STREET	CITY	STATE	PHONE
1 Hour Auto Glass Inc	403 West St	New York	NY	(212)691-3344
1 Hundred 60 4th St Auto Svc	8412 164th St	Jamaica	NY	(718)523-9018
10 Minute Oil Change	1156 Hempstead Tpke	Uniondale	NY	(516)486-0060
A Salerno Realty Crop	11 Mill	Rhinebeck	NY	(914)876-5551
Circuit City	111 E El Camino Real	Sunnyvale	CA	(408)720-1043

Fonte: Zhao, 2008.

Esta dissertação adota uma abordagem de extração e catalogação com o suporte de uma base de conhecimento que mantém informação preliminar sobre os principais domínios de aplicação da *Deep Web*. Maiores detalhes sobre a base de conhecimento definida encontram-se no Capítulo 4.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta uma revisão bibliográfica de alguns trabalhos relacionados à problemática de extração de dados estruturados na *Web* e de catalogação de dados.

3.1 EXTRAÇÃO DE DADOS

A extração de dados na *Web* lida principalmente com a definição de algoritmos que sejam capazes de identificar e recuperar informações presentes em páginas *Web* (Kaiser and Miksch 2005). As abordagens mais comuns de extração de dados estruturados na *Web* são baseadas nas seguintes técnicas:

- Árvore: nesta técnica ocorre à transformação do documento HTML em uma estrutura de dados em árvore, geralmente utilizando o modelo DOM (*Document Object Model*). Alguns trabalhos utilizam esta técnica (Kim et al. 2007) (Liu et al. 2003) (Zhai e Liu, 2005) e, após a construção da árvore, fazem uma busca de padrões na hierárquica para o reconhecimento e extração dos dados relevantes;
- *Web Wrappers*: esta técnica aplica um procedimento específico para consulta e/ou atualização de fontes de dados na *Web*. Este procedimento indica quais dados são extraídos, como extraí-los e como transformá-los e apresentá-los em um formato estruturado ((Liu et al. 2000) (Muslea et al. 2001));
- *Machine Learning*: esta técnica utiliza algoritmos de aprendizado de máquina que extraem informações presentes em fontes de dados na *Web* de um domínio específico. Ela se baseia em regras de extração que consideram delimitadores e sessões de treinamento, durante o qual o algoritmo de extração adquire experiência no domínio. Os dados de treinamento requerem a análise e definição manual de rótulos para os dados presentes em páginas *Web*, exigindo um alto nível de envolvimento humano (Phan et al.2005);
- Ontologia: uma ontologia é utilizada para representar o conhecimento de um determinado domínio, proporcionando o compartilhamento de informações por

diversas aplicações. A sua construção é uma cuidadosa tarefa a ser conduzida manualmente por um especialista no domínio em questão. Para uma aplicação de domínio específico, uma ontologia é usada para a localização de constantes na página e a construção de objetos com ela. Quanto maior a sua representatividade, mais automatizada é a extração. Um exemplo desta técnica pode ser visto em (Embley et al. 1998).

A Tabela 1 apresenta um comparativo entre as abordagens de extração de dados. Ela apresenta diferenças e semelhanças entre as abordagens apresentadas e informa algumas soluções utilizadas por cada uma delas.

Tabela 1- Comparativo das técnicas de extração de dados na *Web*.

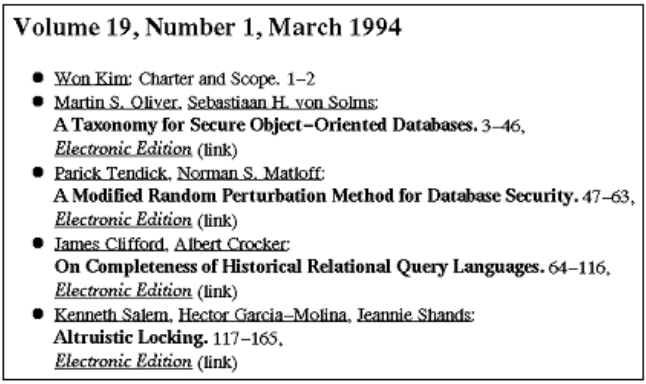
Abordagem	**Principais Soluções**	**Grau de automatização**	**Tipo de conteúdo**	**Formatos de Saída**
Árvore	MDR	Automática	SD	HTML
Wrappers	Road Runner	Automática	SD	XML, HTML
	XWrap (Liu at el. 2000)	Automática	SD	XML
Machine Learning	Phan et al. 2005	Semiautomática	SD	HTML
Ontologia	Embley et al. 1998	Manual	ST/SD	HTML

Quanto ao grau de automatização da extração, ela pode ser realizada de maneira manual, semiautomática ou automática. A extração manual possui um alto índice de qualidade, porém é indicada para uma pequena quantidade de dados. No modelo automático, não há intervenção humana, contudo pode-se perder precisão durante a extração. Este método é indicado para grandes quantidades de dados. Já o modo semiautomático utiliza geralmente um sistema de apoio onde o usuário descreve quais campos de dados possui interesse.

Com relação ao conteúdo da página, tem-se uma divisão em duas categorias: texto semiestruturado (ST) e dados semiestruturados (SD). Para ilustrar, considere as páginas apresentadas nas Figuras 5 (a) e (b), que são exemplos de páginas contendo dados semiestruturados e texto

semiestruturado, respectivamente. Enquanto as páginas da primeira categoria possuem itens de dados (por exemplo, nomes de autores, títulos de artigos, etc.) implicitamente formatados para serem reconhecidos individualmente, as páginas da segunda categoria trazem texto livre a partir do qual os itens podem apenas ser inferidos pelo processo de extração.

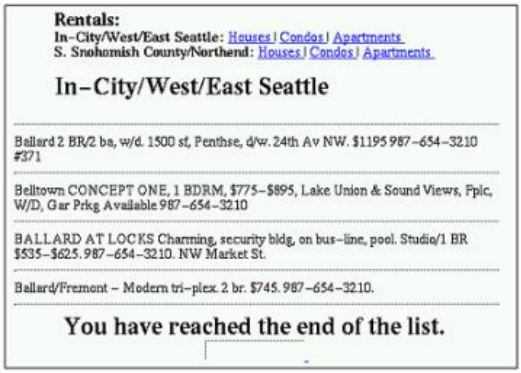
Figura 5 - (a) Exemplo de dados semiestruturados.



Volume 19, Number 1, March 1994

- Won Kim: Charter and Scope. 1–2
- Martin S. Olivier, Sebastiaan H. von Solms: **A Taxonomy for Secure Object–Oriented Databases.** 3–46, *Electronic Edition* (link)
- Parick Tendick, Norman S. Matloff: **A Modified Random Perturbation Method for Database Security.** 47–63, *Electronic Edition* (link)
- James Clifford, Albert Crocker: **On Completeness of Historical Relational Query Languages.** 64–116, *Electronic Edition* (link)
- Kenneth Salem, Hector Garcia–Molina, Jeannie Shands: **Altruistic Locking.** 117–165, *Electronic Edition* (link)

(b) Exemplo de texto semiestruturado.



Rentals:
In–City/West/East Seattle: Houses | Condos | Apartments
S. Snohomish County/Northend: Houses | Condos | Apartments

In–City/West/East Seattle

Ballard 2 BR/2 ba, w/d. 1500 sf, Penthse, d/w. 24th Av NW. $1195 987–654–3210 #371

Belltown CONCEPT ONE, 1 BDRM, $775–$895, Lake Union & Sound Views, Fplc, W/D, Gar Prkg Available 987–654–3210

BALLARD AT LOCKS Charming, security bldg, on bus–line, pool. Studio/1 BR $535–$625. 987–654–3210. NW Market St.

Ballard/Fremont – Modern tri–plex. 2 br. $745. 987–654–3210.

You have reached the end of the list.

Fonte: Laender, 2002.

O formato de saída gerado após a extração varia entre HTML e XML. Entende-se que o formato XML, por ser amplamente utilizado

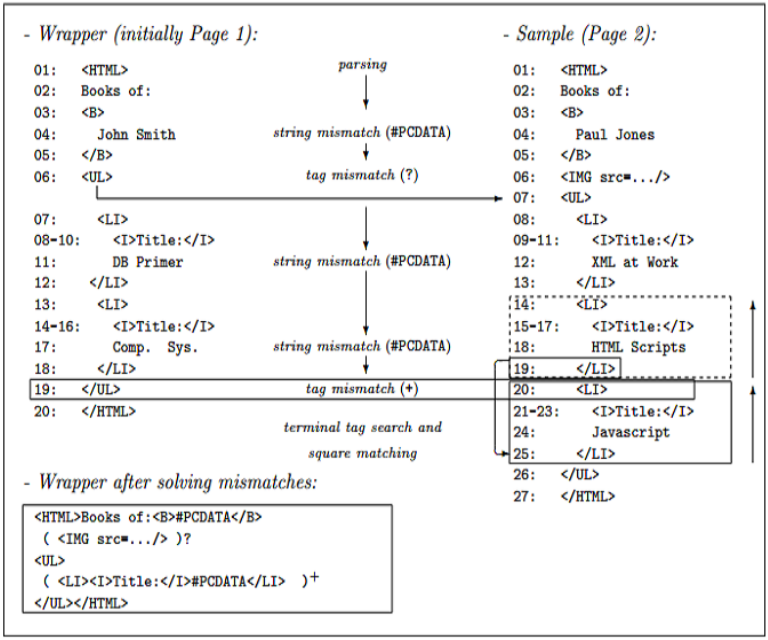
para a representação e troca de dados na *Web* vem como uma tendência promissora para manter dados extraídos.

A análise dos trabalhos relacionados ressalta que nenhuma das abordagens inclui um processo de catalogação dos dados extraídos, visando facilitar futuras consultas. Essa limitação motivou o desenvolvimento desta dissertação.

Dentre as soluções existentes e citadas na Tabela 1, este trabalho considerou dois métodos de extração durante os experimentos, para fins de comparação de resultados com a abordagem proposta: *Road Runner* e MDR. Estes métodos são amplamente referenciados na literatura ((Hong, 2010) (Oro et al. 2011) (Zhai et al. 2005)).

O *Road Runner* (Muslea et al. 2001) é um método totalmente automático para a extração de dados. A Figura 6 mostra um exemplo do processo de extração do *Road Runner*. Dada uma amostra com páginas *Web* semelhantes como entrada, contendo um ou mais registros de dados, o método atua em duas páginas ao mesmo tempo: uma chamada de *Sample*, e outra chamada de *Wrapper*. Uma delas é considerada a versão inicial do *wrapper*.

Figura 6 - Exemplo do processo de extração do *Road Runner*.



Fonte: Muslea et al. 2001.

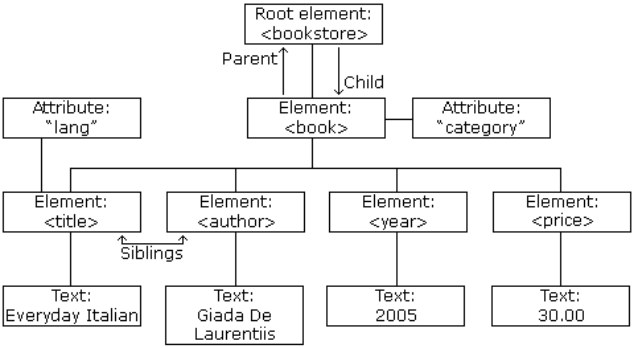
O *Road Runner* compara os conteúdos HTML para detectar semelhanças e diferenças, a fim de definir uma expressão regular comum para as duas páginas. A cada nova comparação, esta expressão é refinada/generalizada, resolvendo diferenças entre *strings* que dizem respeito aos conteúdos dos dados. Isto é feito através da resolução de incompatibilidades entre o *wrapper* e a amostra. Uma incompatibilidade acontece quando alguma parte da amostra não coincide com a gramática especificada pelo *wrapper*. Sempre que uma incompatibilidade é encontrada, *Road Runner* tenta resolvê-la para generalizar o *wrapper*.

Diferenças entre *strings* são detectadas quando duas páginas possuem valores diferentes no campo de dados. Quando isso ocorre é colocado "#PCDATA" no lugar da *string* e isso é interpretado como informação adicional, pois ela não será usada para gerar a expressão. Diferenças entre *tags* dizem respeito a itens opcionais da página HTML. Depois de resolver todas as incompatibilidades, a análise é concluída, gerando uma expressão comum para as páginas HTML que é utilizada para extrair registros de dados de outras páginas da *Web*.

Cabe salientar que nem sempre se consegue construir uma expressão regular genérica o suficiente para um conjunto de páginas em um mesmo domínio, o que prejudica a qualidade da extração.

O método MDR (*Mining Data Records*) (Liu et al. 2003) inicia criando uma árvore DOM da página HTML alvo. A Figura 7 apresenta um exemplo de árvore DOM onde as *tags* são os nodos internos e os textos são os nodos folha.

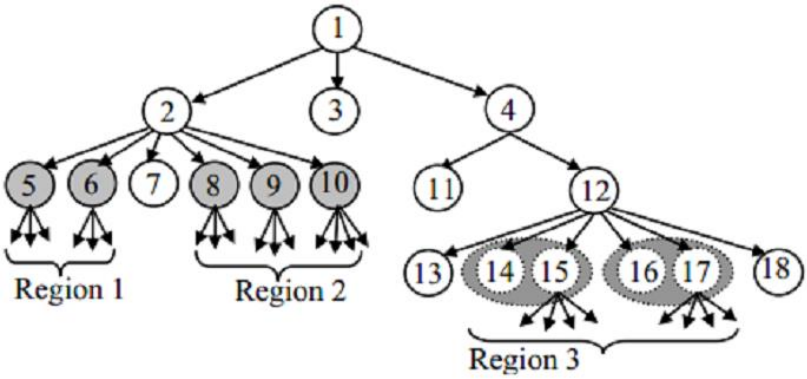
Figura 7 – Exemplo de árvore DOM para uma página HTML.



Fonte: http://www.w3schools.com/dom/

O método MDR percorre essa estrutura em árvore de maneira *bottom-up* verificando a similaridade dos nodos adjacentes. Para isso, ele usa o algoritmo STM (*Simple Tree Matching*) e a métrica de distância de edição para comparar os *strings*. Após isso, ele insere nodos similares em um conjunto. Esses conjuntos são denominados regiões de dados, conforme ilustra a Figura 8. Os nodos dessa região devem ter o mesmo nodo pai, a mesma quantidade de filhos, devem ser todos adjacentes e ter o resultado da função de distância de edição menor que um *threshold* definido pelo usuário.

Figura 8 - Árvore DOM com a região de dados.



Fonte: Liu et al. 2003.

Por fim, o MDR procura campos de dados em cada um desses conjuntos para extrair só o conteúdo que interessa. Cabe salientar, como limitação, que, o método pode detectar diversos padrões, sendo difícil saber qual padrão é o correto, além de não conseguir extrair os dados aninhados.

3.2 CATALOGAÇÃO DE DADOS

Esta seção apresenta algumas técnicas presentes na literatura para a solução do problema de catalogação de dados.

O trabalho de (Zhao et al. 2008) desenvolve uma técnica para catalogação de texto baseada no conceito de CRFs (*Conditional Random Fields*). CRF utiliza tabelas com dados estruturados como referência, de acordo com cada domínio cujo dado de entrada pertença. A Figura 4

apresenta uma tabela de referência utilizada pelo CRF. Em outras palavras, os rótulos dos dados de treinamento são definidos a partir de tabelas de referência. Assumindo que as sequências de texto para ser segmentado vêm em lotes elas estarão em conformidade com a mesma ordem dos atributos definidos no treinamento. Após esse treinamento, onde é definida a ordem dos atributos no esquema gerado, ele é aplicado no restante da entrada.

ONDUX (*On-Demand Unsupervised Learning for Information Extraction*) (Cortez et al. 2010) utiliza a indução automática para a catalogação, ou seja, características aprendidas com os dados obtidos a partir de uma fonte de entrada são utilizadas para induzir o papel de cada item de dado na estrutura de outros textos relacionados ao assunto. O ONDUX requer a intervenção direta do usuário para delimitar cada registro de entrada no texto durante o aprendizado, bem como para definir a estrutura dos registros a serem extraídos. A estrutura do registro desempenha um papel muito importante neste contexto, uma vez que depende de características estruturais (posicionamento e sequenciamento) para executar a tarefa de catalogação.

Outra solução neste contexto é a JUDIE (*Joint Unsupervised Structure Discovery and Information Extraction*) (Silva et al. 2011). JUDIE recebe como entrada um texto (Figura 9(a)) contendo um conjunto de registros de dados. Após o recebimento da entrada, ocorre a segmentação dos dados e uma rotulação de valores potenciais, (Figura 9(b) e (c)), realizados através da comparação com um conjunto de dados pré-existente relacionado ao domínio mantido em uma base de conhecimento. Após a segmentação e a primeira rotulagem, um algoritmo baseado em um modelo de posicionamento e sequenciamento dos dados que são frequentemente repetidos ao longo do texto de entrada rotula-os novamente (Figura 9 (d)), confirmando ou efetuando possíveis correções nos rótulos inicialmente catalogados.

Figura 9 – Passos do JUDIE para catalogação dos dados.

Fonte: Silva, 2011.

A Tabela 2 apresenta um comparativo entre as abordagens de catalogação de dados. Ela apresenta a relação do que é utilizado como apoio para a comparação dos dados extraídos, utilizando para isso geralmente uma tabela de referência ou uma base de conhecimento e as características pelas quais a abordagem realiza a comparação da fonte de entrada com o que é utilizado como apoio.

Tabela 2- Comparativo das técnicas de catalogação de dados na *Web*.

Abordagem	**Utiliza como Apoio**	**Características**
Zhao et al. 2008	Tabela de Referência	Os valores dos atributos da amostra de dados devem apresentar uma ordem fixa
ONDUX	Base de Conhecimento	Indução automática
JUDIE	Base de Conhecimento	Indução automática e identificação de sequências

Algumas considerações a respeito destas abordagens, que foram importantes para justificar o desenvolvimento desta dissertação, são:

- Na técnica baseada em CRF, o esquema assume uma ordem fixa conforme a amostra, prejudicando os dados fora do esquema previamente definido;
- Os trabalhos apresentados não realizam a detecção de informações. No contexto deste trabalho, detecção de informações corresponde à complementação de informações que fazem parte do domínio, porém não estão presentes na fonte de dados de entrada para a catalogação.

O capítulo seguinte discorre sobre a abordagem *DeepEC*, que trata algumas das limitações das abordagens presentes nos trabalhos relacionados.

4 DEEPEC

Este capítulo apresenta a abordagem proposta nesta dissertação, intitulada *DeepEC* (*DeepWeb Extraction and Cataloguing Process*). Ela é responsável pela execução simultânea da extração e da catalogação de informações relevantes presentes em bancos de dados escondidos apresentadas em páginas HTML. Supõe-se que estas páginas são obtidas com o preenchimento e submissão de consultas através de formulários *Web*, estando esses processos fora do escopo desta dissertação.

A abordagem utiliza uma Base de Conhecimento (BC) como auxílio nos processos de extração, catalogação e detecção de dados. A seção seguinte detalha a base de conhecimento projetada para este trabalho.

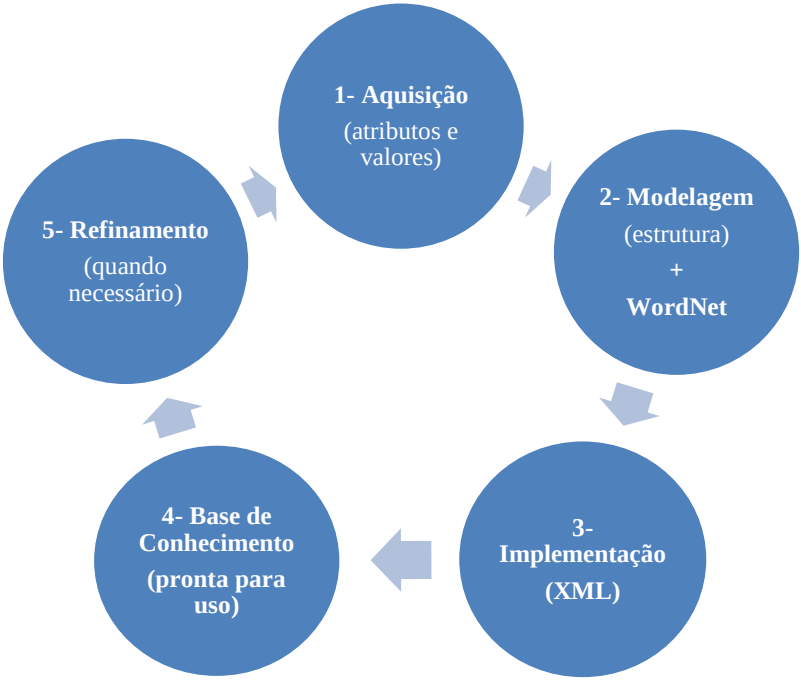
4.1 BASE DE CONHECIMENTO

Uma BC tem como proposta manter dados e/ou conhecimento acumulado sobre um determinado assunto. Esse conhecimento pode ser utilizado na solução de problemas, geralmente por meio de ferramentas de Inteligência Artificial ou Sistemas Especialistas (Hu et al. 2009). Entre os benefícios da utilização de uma BC pode-se citar a maior organização de toda informação relativa a um domínio referenciado e a facilidade para a consulta dos registros de dados.

Este trabalho utiliza uma BC como auxílio nos processos de extração, catalogação e detecção de dados. O ciclo de vida da metodologia utilizada na construção da BC para este trabalho é mostrado na Figura10. Esta metodologia foi adaptada do modelo de cinco fases do processo de criação do conhecimento apresentado por (Nonaka e Takeuchi, 1997) para a abordagem proposta nesta dissertação. Primeiramente, ocorre a aquisição de conhecimento. Em seguida, a modelagem do conhecimento adquirido na etapa anterior, integrando conhecimento com base no *corpus* presente no *WordNet*. Após, a implementação constrói a BC no formato XML e, por fim, ocorre o refinamento.

A seguir é descrita a aplicação desta metodologia para este trabalho.

Figura 10 – Ciclo de vida definido para a criação da BC.



A fase de *Aquisição* (1) da informação foi concretizada com o acesso a alguns *Websites* relativos aos domínios previamente escolhidos para os experimentos preliminares da abordagem proposta neste trabalho (Automóveis e Livros) e extração manual de amostras de rótulos de atributos em formulários e seus valores associados. Vale ressaltar que um formulário Web é composto basicamente por atributos, sendo que cada atributo possui um rótulo e um conjunto de valores relacionados ao rótulo. Um formulário pode ainda conter restrições implícitas ou explicitas sobre um determinado conjunto de valores válidos para cada atributo, pertencentes ao domínio do banco de dados escondido (Mello et al. 2010).

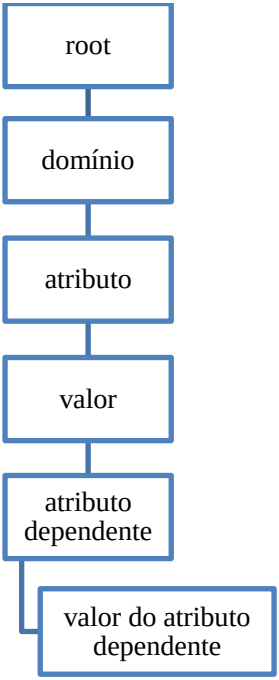
A fase de *Modelagem* (2) do conhecimento foi realizada com o auxílio do que é disponibilizado pela *Freebase*[2]. A *Freebase* é um repositório colaborativo de dados estruturados que está disponível para pesquisa. Ela é referenciada em alguns trabalhos ((Zheng et al. 2012)

[2]http://www.freebase.com/

(Serra et al. 2011)) e conta atualmente com aproximadamente 23 milhões de entidades. A estrutura da BC foi definida e construída hierarquicamente como um diretório de pastas e arquivos, onde as pastas representam os rótulos e os arquivos valores. Na raiz desta hierarquia se encontra a divisão em domínios e, para o domínio de Automóveis, considerou-se os atributos mais frequentes encontrados em formulários, que são: "*make*", "*model*", "*year*", "*color*", "*door*", "*mileage*" e "*price*". Para o domínio de Livros, foram considerados os seguintes atributos: "*title*", "*author*", "*price*", "*isbn*", "*year*" e "*publisher*". Além disso, o esquema da BC considera ainda dependências de valor presentes entre atributos e que são muito comuns em formulários *Web*. Um exemplo é o valor de atributo "*make*", que determina um ou mais valores do atributo "*model*".

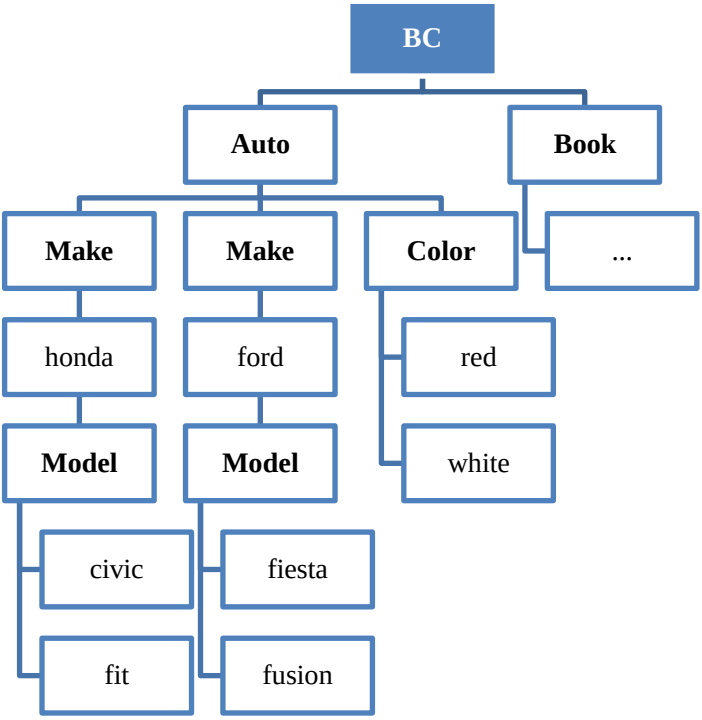
A Figura 11 apresenta o esquema de dados da BC. Esse esquema hierarquizado permite a realização da detecção de valores de atributos durante o processo de catalogação de dados extraídos. O processo de detecção é detalhado no Capítulo5.

Figura 11 – Esquema do modelo de dados na BC.

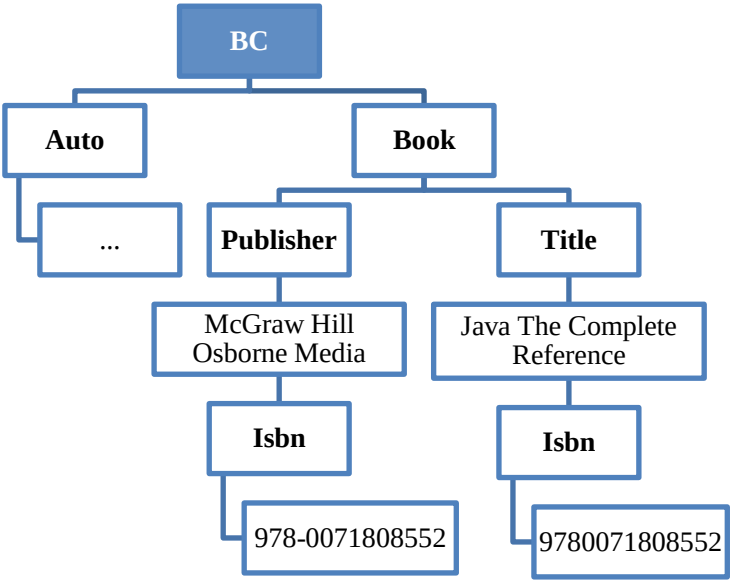


A Figura 12 (a) e (b) apresentam um exemplo de modelagem parcial da BC para os domínios de Automóveis e Livros com base no esquema do modelo de dados da Figura 11.

Figura 12 – (a) Exemplo de conteúdo da Base de Conhecimento no domínio de Automóveis.



(b) Exemplo de conteúdo da Base de Conhecimento no domínio de Livros.



A etapa de Modelagem considera ainda o enriquecimento semântico da BC através da integração dos dados recuperados durante a aquisição de conhecimento (etapa 1) com o *WordNet*[3] (Miller, 1995). *WordNet* é um grande banco de dados lexical da língua inglesa. Substantivos, verbos, adjetivos e advérbios são agrupados em conjuntos de sinônimos cognitivos (*synsets*), cada um expressando um conceito distinto. *Synsets* estão interligados por meio de relações conceituais-semânticas e lexicais. Por definição, cada *synset* que compõe a rede representa o conceito lexicalizado pelas unidades lexicais sinônimas que o compõem.

A integração foi realizada de forma automática, utilizando a API do *WordNet* para a linguagem Java chamada *Java Wordnet Interface*[4] (JWI) através da comparação dos termos existentes na BC com a base de dados do *WordNet.* Os novos termos reconhecidos pelo *WordNet* foram incrementados na BC.Com esse conhecimento adquirido foi possível catalogar casos como, por exemplo, *"make"* e

[3]http://wordnet.princeton.edu/
[4]http://projects.csail.mit.edu/jwi/api/

"brand", que são classificados com baixo grau de equivalência pelos algoritmos de similaridade de texto, porém, possuem o mesmo significado.

A *Implementação* (3) propriamente dita da *Base de Conhecimento* (4) foi especificada no formato XML. XML foi adotado considerando toda a tecnologia disponível para a manipulação de dados neste formato, bem como para o intercâmbio de dados entre humanos e aplicações. A Figura 13 apresenta uma amostra da BC populada e representada no formato XML, com exemplos de conteúdos dos rótulos, inclusive com sinônimos para alguns rótulos enriquecidos a partir do *WordNet*.

Figura 13 – Exemplo parcial da BC representada no formato XML.

```
<?xml version="1.0"?>
- <KB>
  - <domain>
    - <auto>
        <mandatoryCar value="make,model"/>
      - <make value="HONDA">
          <model value="CIVIC,FIT,CRV,ACCORD"/>
        </make>
      - <make value="FORD">
          <model value="ESCORT,FIESTA,FUSION"/>
        </make>
        <color value="RED,WHITE,BLUE,BLACK,SILVER"/>
        ...
      </auto>
      <book>...</book>
    </domain>
</KB>
```

O elemento *mandatory* indica os chamados atributos mandatórios de cada domínio, ou seja, os atributos mais relevantes neste domínio. A definição de atributo mandatório é apresentada no Capítulo 5.

O *Refinamento* (5) corresponde à inclusão/atualização de conhecimento, é realizado quando necessário durante a utilização da BC. Ela refaz todas as etapas para a atualização/incremento do conhecimento.

Nesta versão da abordagem proposta, a BC é totalmente carregada em memória e manipulada através de métodos DOM. Esta estratégia simples foi adotada uma vez que apenas dois domínios foram

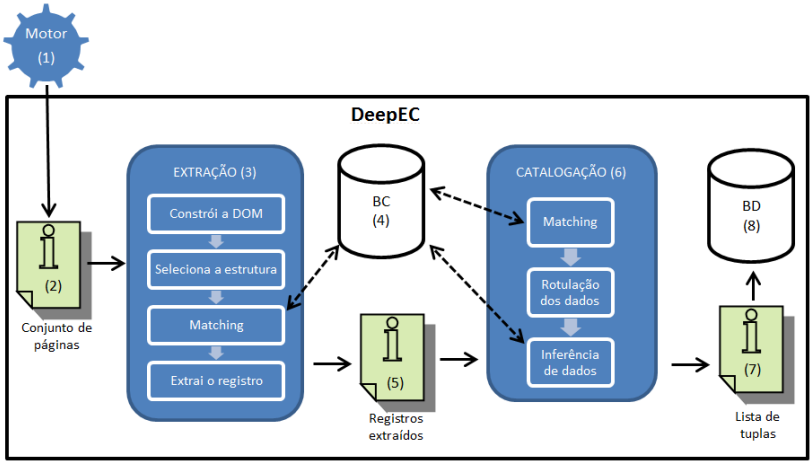
considerados e o volume da BC (relativamente pequeno) não prejudicou o desempenho. Considerando a possível extensão da BC, para incorporar outros domínios, e prováveis problemas de desempenho, pretende-se adotar, para novas versões, o seu particionamento em vários arquivos XML por domínio e a indexação destes domínios e de seus elementos, utilizando, por exemplo, a ferramenta *Lucene*[5].

A BC projetada e apresentada neste capítulo é o componente chave da solução proposta nesta dissertação, intitulada *DeepEC*.

4.2 ARQUITETURA DA DEEPEC

A *DeepEC* está centrada em dois principais processos que realizam a extração e a catalogação de dados da *Deep Web*. A Figura 14 apresenta a arquitetura da abordagem com os dois componentes correspondentes a estes processos. Estes componentes são explicados a seguir.

Figura 14 – Arquitetura do *DeepEC*.

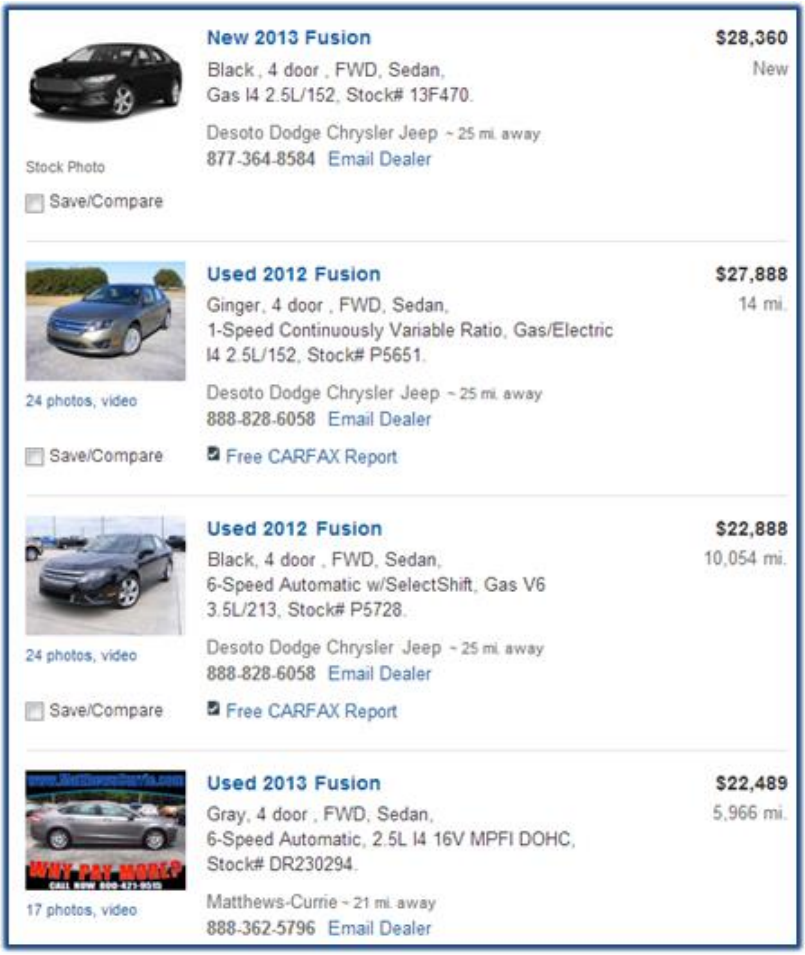


O componente externo Motor (1) encapsula o processo de descoberta de bancos de dados escondidos na *Web* e, na sequência, o preenchimento de formulários *Web*, a submissão de consultas e a geração de páginas HTML com os resultados obtidos. O trabalho da *DeepEC* inicia a partir deste ponto, ou seja, ela recebe como entrada este

[5]http://lucene.apache.org/core/

conjunto de páginas (2). Cada página desse conjunto serve de entrada para o componente de Extração (3). A Figura 15 mostra um exemplo de página de entrada para o *DeepEC* no domínio de Automóveis, estando os dados relevantes presentes no centro da página.

Figura 15 – Página HTML com resultados de uma consulta a um banco de dados escondido visualizada em um browser *Web*.



Com as páginas de entrada apresentando os dados da *Deep Web*, são detalhados a seguir os processos de Extração e Catalogação dos dados.

4.3 PROCESSO DE EXTRAÇÃO

O processo de extração adotado pela *DeepEC* é baseado em duas heurísticas que visam melhorar a qualidade da recuperação dos registros de dados presentes nas páginas HTML. Estas heurísticas são definidas a seguir.

Definição 1 (Heurística da Estrutura Relevante - HER). A estrutura de representação de dados que mais se repetir na página HTML é onde provavelmente estão localizados os registros relevantes advindos do banco de dados.

Definição 2 (Heurística da Estrutura Irrelevante - HEI). As estruturas HTML definidas pelas *tags* "*script*", "*select*" e "*option*" são descartadas, pois provavelmente são funções ou campos de formulários sem dados relevantes.

Essas heurísticas foram definidas com base na análise de páginas HTML utilizadas em experimentos preliminares durante o desenvolvimento do extrator proposto. A primeira heurística é a mais importante para o bom funcionamento do extrator, pois é ela que decide o que é considerado relevante. Ela foi criada com base no trabalho de (Hong, 2010), que afirma que a estrutura com maior repetição tem a maior probabilidade de ser a estrutura buscada, com o diferencial de utilizar como auxílio a BC para encontrar os valores relevantes, confirmando que a estrutura mais frequente contém os dados desejados (ou não). Com a segunda heurística, o extrator consegue, além do foco no conjunto de dados com maior probabilidade de ser relevante, analisar de forma mais minuciosa a estrutura HTML, deixando de considerar *tags* que não possuem dados úteis.

O Algoritmo 1 detalha o processo de extração dos dados das páginas HTML. Inicialmente, o algoritmo recebe como entrada a página HTML e os termos da BC. O componente de extração utiliza a BC (4) para ajudar na localização dos registros relevantes a serem extraídos. Na linha 6, a página é instanciada no modelo DOM e na linha 7 é aplicada a heurística HEI que realiza a remoção da estrutura irrelevante. Nas linhas 8 e 9 é realizada uma busca por atributos mandatórios de um

determinado domínio disponíveis na BC. A definição de atributo mandatário é dada a seguir.

Definição 3 (Heurística de Atributo Mandatário - HAM). Um atributo mandatário A_m é um atributo significativo, e geralmente uma propriedade específica, de um determinado domínio, servindo para caracterizar, de forma mais evidente, o domínio ao qual o registro que A_m pertence.

Exemplo. O atributo *make* é um atributo mandatário no domínio de Automóveis, pois é uma propriedade inerente a qualquer automóvel.

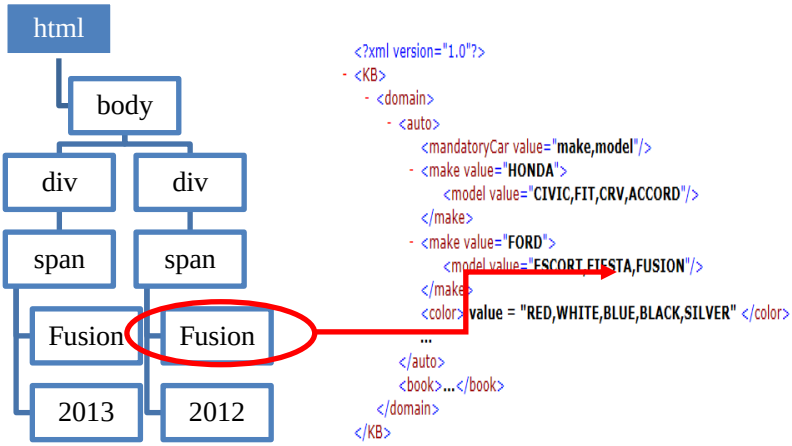
A definição do atributo mandatório é realizada manualmente e permite uma otimização no Algoritmo 1, uma vez que somente atributos significativos do domínio em questão são comparados com os termos do registro a ser extraído.

Algoritmo 1: Método de Extração.

```
1:  Entrada: Base de Conhecimento BC;
2:  Entrada: páginaHTML;
3:  Início
4:    CaminhosCandidatos←{}
5:    CaminhosRelevantes←{}
6:    PDOM ← DOM Parser(pagina)
7:    SDOM ←HEI(PDOM )
8:    para cada valor de HAM v ∈ BC faça
9:      para cada termo t de um nodo folha n ∈ SDOM faça
10:       se matching(v, t) então
11:         CaminhosCandidatos←CaminhosCandidatos + Caminho (n)
12:       fim se
13:     fim para
14:   fim para
15:   CaminhosRelevantes← HER (CaminhosCandidatos)
16:   para ∀ termo t de um nodo folha n ∈ CaminhosRelevantes faça
17:     Extrai(t)
18:   fim para
19: Fim
```

A Figura 16 exemplifica o processo de *matching* entre o valor de um atributo mandatório com um termo presente nas folhas de um nodo DOM da página HTML (linha 10). Mesmo a BC sendo extensa, este processo é realizado somente com as informações que pertencem a *tag mandatoryCar*.

Figura 16 – Exemplo de comparação com atributo mandatário.

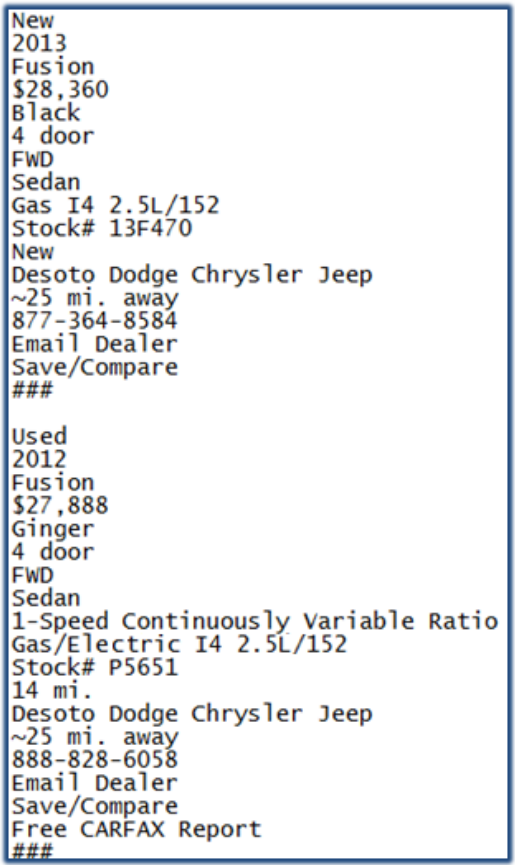


Após a identificação de tais registros (linha 10), o caminho da *tag* raiz que representa o registro de dado até os nodos folha, para os nodos que atendem a heurística descrita anteriormente, é salvo na variável "CaminhosCandidados" (linha 11). Na linha 15 é aplicada HER sobre os caminhos candidatos, selecionando apenas os caminhos relevantes. Por fim, na linha 17, ocorre a extração dos termos de dados referentes ao registro que se encontram nos caminhos relevantes.

O método de *matching* compara um valor de atributo mandatário de cada domínio com um conteúdo do registro candidato a ser extraído, com o objetivo de reforçar a hipótese de que o registro a ser extraído pertence a um domínio de interesse. Ele executa a função de similaridade de *strings Jaro-Winkler* (Winkler, 1990). A utilização dessa função foi baseada na sua recomendação de utilização para nomes próprios, dentre alguns outros algoritmos de similaridade existentes (Dorneles et al. 2009) e, além disso, obteve bom desempenho para os outros tipos de valores presentes em domínios da *Deep Web*.

Cada página analisada pelo componente de extração gera um arquivo de saída, como mostra a Figura 17. Este arquivo contém os registros extraídos (5). Um registro significa uma tupla do BD cujo rótulo e valores estão presentes no BD escondido. Cada linha corresponde ao valor de um campo presente em cada nodo que foi extraído. Ainda, cada registro extraído é delimitado (“###”) para fins de identificação. A identificação do registro é gerada quando ocorre a leitura/extração do nodo pai subsequente ao nodo que teve seus itens extraídos. Este arquivo serve de entrada para o componente de Catalogação.

Figura 17 – Exemplo de arquivo com os registros extraídos.



4.4 PROCESSO DE CATALOGAÇÃO

O componente de Catalogação (6) adota uma abordagem de análise e caracterização de segmentos de texto, utilizando como apoio a BC (4). O Algoritmo 2 apresenta o princípio do funcionamento do método de catalogação. O algoritmo possui como entrada os termos da BC e o arquivo com os registros extraídos na etapa anterior. Ele utiliza as delimitações (de campos de registros (cada linha) e de registros em si ("###")) definidas durante a extração para realizar a catalogação dos valores presentes em cada linha. A *DeepEC* lê cada termo/conjunto de termos presente na linha e compara com os valores da BC.

Na linha 4 é chamado o método para detecção do domínio a qual pertence os itens de dados a serem catalogados. Esta detecção é baseada na heurística de detecção de domínio, definida a seguir.

Definição 4 (Heurística de Detecção de Domínio - HDD). Uma detecção de um domínio ocorre quando um item de dado de um registro extraído casa, por similaridade, com valores de atributos mandatários da BC pertencentes a um mesmo domínio.

A definição dessa heurística foi determinada a partir da análise dos principais atributos dos domínios presentes na *Deep Web*. Verificou-se que os atributos mandatários são específicos para cada domínio, possibilitando a detecção a qual domínio pertence à página analisada. Essa heurística apresentou resultados satisfatórios nos experimentos realizados com os dois domínios considerados.

Algoritmo 2: Método de Catalogação.

```
1:  Entrada: Base de Conhecimento BC;
2:  Entrada: conjunto de registros extraídos RegEx;
3:  Início
4:    d←HDD(RegEx)
5:    se d≠ nulo então
6:      para cada atributo com tag t e valor v ∈ BC(d) faça
7:        para cada registro r ∈ RegEx faça
8:          para cada item i ∈ r faça
9:            se matching(v, i) então
10:             Cataloga(i,t,d)
11:           senão
12:             se detectaPadrao(i) então
13:               Cataloga (i,t,d)
14:             fim se
15:           fim se
16:         fim para
17:         para cada atributo t da tupla catalogada tp |valor (t,tp) = nulo faça
18:           itemdetectado←HDC (t)
19:           se itemdetectado≠nulo então
20:             Cataloga (itemdetectado,t,d)
21:           fim se
22:         fim para
23:       fim para
24:     fim para
25:   fim se
26: Fim
```

Se o domínio é detectado (linha 5), nas linhas6 a 8 é lido cada registro extraído e comparado com os termos da BC pertencentes ao domínio detectado. As igualdades entre os termos da BC e os itens dos registros são identificadas (linha 9) utilizando igualmente a função de similaridade *Jaro-Winkler*.

Quando uma correspondência é identificada, o termo e o seu significado (*tag*) são armazenados (linha 10). O método de armazenamento corresponde a uma tripla, que contém o item (campo)*(i*) do registro, a *tag* (*t*) da BC que corresponde ao rótulo do campo correspondente ao item do registro e ao domínio (*d*). Maiores detalhes sobre o armazenamento dos termos de um registro no BD de saída são mostrados na Seção 4.5.

Para a otimização do processo de catalogação, implementou-se uma função genérica para reconhecimento de valores que possuem o

mesmo padrão de escrita (linha12). Esse reconhecimento é realizado através do uso de expressões regulares (Jargas, 2012) e funções que testam se essas expressões casam com o valor a ser catalogado. Os padrões que são reconhecidos através de funções na *DeepEC* são: ano, preço e quilometragem. Estes padrões estão associados aos domínios considerados pela BC da *DeepEC*. A intenção é que este número de padrões seja extensível, conforme novos domínios sejam incorporados. Maiores detalhes são apresentados no Algoritmo 3.

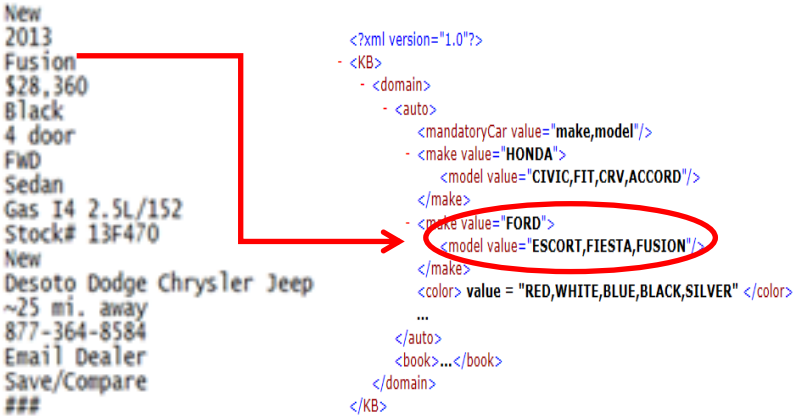
No contexto do processo de Catalogação, uma das contribuições desta dissertação é a detecção de informação, ou seja, após a identificação das correspondências entre termos pertencentes a registros extraídos com termos de elementos de amostras da BC, busca-se a indução e consequente complementação de informações não disponíveis nos registros extraídos. Desta forma, enriquece-se o conteúdo deste registro no ato da catalogação. O enriquecimento ocorre através da complementação de valores na tupla. Este procedimento de detecção é definido a seguir.

Definição 5 (Heurística de Detecção de Conteúdo - HDC). Uma detecção de conteúdo de atributo ocorre quando existe uma dependência de valor entre atributos *X* e *Y*, sendo *X* o atributo pai e *Y* o atributo dependente na hierarquia da BC. Neste caso, se o valor de *Y* é catalogado, o valor de *X* também o será.

Na linha 17, ocorre a análise da existência de algum valor nulo na tupla que foi formada para a inserção no BD. Caso exista algum campo nulo, é verificada a possibilidade de detecção de conteúdo através da heurística HDC (linha 18). Caso ocorra a detecção de conteúdo, o mesmo é catalogado, conforme indicado na linha 20.

A Figura 18 demonstra como ocorre a detecção de informações durante a catalogação. A parte esquerda da figura mostra um exemplo de registro extraído e a parte direita mostra um exemplo de fragmento da BC no formato XML.

Figura 18 – Exemplo de detecção de valores de atributos de registros.



Devido à hierarquia da BC, o sistema é capaz de detectar alguns valores de atributos (*tags*) durante a categorização de um registro. Neste exemplo, para o domínio de Automóveis, é possível detecção o valor do atributo *make* (*"Ford"*) a partir do casamento do valor do conteúdo do atributo *model* (*"Fusion"*) presente no registro extraído. Analogamente, a BC possibilita, para o domínio de Livros, realizar a detecção de valores do título, autor e editora quando se possui o ISBN.

O Algoritmo 3 apresenta um exemplo de função interna à função *detectaPadrao()* para o reconhecimento de um conteúdo que corresponde a um ano. Ela possui como entrada o item extraído (linha 1). A expressão para reconhecimento do tipo ano é definida na linha 4 e este padrão é comparado com o item de entrada na linha 5. Dependendo se houver ou não correspondência (linhas 6 e 7), a função retorna TRUE ou FALSE.

Algoritmo 3: Função REGEX para reconhecimento de valor do tipo ano.

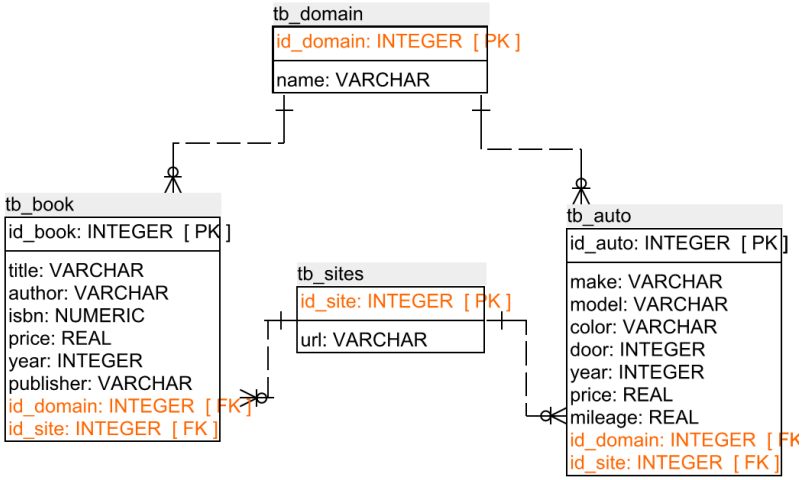
```
1:  entrada: valor numérico item (i)
2:  Início
3:    item← i
4:    padrão←compila("[1|2][0-9]{3}$")
5:    se item = padrão então
6:       retorna TRUE
7:       senão retorna FALSE
8:  Fim
```

Uma vez processado adequadamente um registro para fins de catalogação, ele é armazenado no BD de saída (8), que é detalhado na próxima Seção.

4.5 ARMAZENAMENTO DOS REGISTROS PROCESSADOS PELO COMPONENTE DE CATALOGAÇÃO

Os dados catalogados pelo Componente de Catalogação são armazenados em um BD relacional. A Figura 19 apresenta o esquema deste BD. Ele contém uma tabela com a relação dos domínios e ainda uma tabela por domínio, com os principais atributos destes domínios. Estas últimas tabelas armazenam efetivamente os registros de dados catalogados. O atributo URL é persistido na tabela específica (*sites*) e corresponde à URL do site do serviço onde está o formulário. Ela é passada como entrada juntamente com cada página para o *DeepEC*.

Figura 19 – Esquema do BD relacional para o armazenamento de registros extraídos da *Deep Web*.

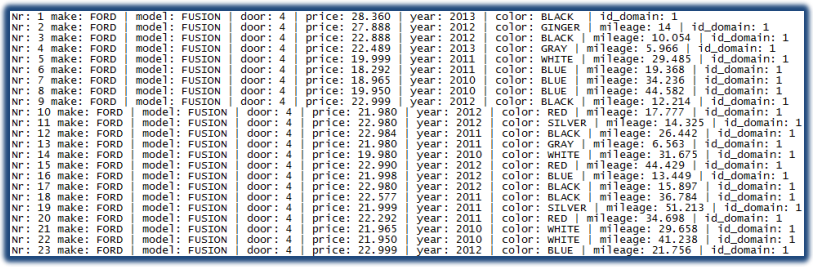


As colunas das tabelas de domínio do BD possuem os mesmos nomes dos atributos principais (*tags*) presentes na BC, como por exemplo, *make* e *year* (ver Figura 21). Desta forma, fica facilitada a descoberta da coluna correta onde o item de dado do registro deve ser armazenado.

Cabe salientar que, quando existe algum item de dado de um registro a ser catalogado que não corresponde a nenhum atributo da tabela no BD, esse valor é descartado. Ainda, quando não é encontrado algum valor de atributo para a tabela na qual a tupla deve ser catalogada, este atributo fica sem valor (nulo). Para casos de atributos multivalorados, como por exemplo, *author* no domínio de Livros, todos os valores são concatenados e armazenados como um único valor no atributo correspondente na tabela do BD.

A Figura 20 apresenta o resultado obtido com o processo de catalogação dos dados que foram extraídos anteriormente. De acordo com a arquitetura do *DeepEC* mostrada na Figura 14, as tuplas listadas (7) no arquivo estão prontas para serem inseridas no BD.

Figura 20 – Exemplo de registros catalogados.



Para exemplificar melhor o processo de catalogação e a detecção de novas informações, a Figura 21 mostra parte da tabela *tb_auto* do BD com tuplas cujos valores de atributos foram diretamente armazenados a partir dos registros extraídos ou detectados a partir da BC.

Figura 21– Exemplo de uma visão com todas as informações catalogadas de um registro exemplo.

make	model	color	door	year	price	mileage	id_domain	id_site
FORD	FUSION	BLACK	4	2013	28.360		1	4
FORD	FUSION	GINGER	4	2012	27.888	14	1	4
FORD	FUSION	BLACK	4	2012	22.888	10.054	1	4
FORD	FUSION	GRAY	4	2013	22.489	5.966	1	4

Cabe salientar que as tabelas de domínios do BD para os dados catalogados possuem muita redundância de dados por não terem sofrido um processo de normalização nesta primeira versão da *DeepEC*, como por exemplo, o fato de que *Fusion* é um modelo da fabricante *Ford*, conforme mostra a Figura 21.Esta otimização do projeto do BD é alvo de trabalhos futuros.

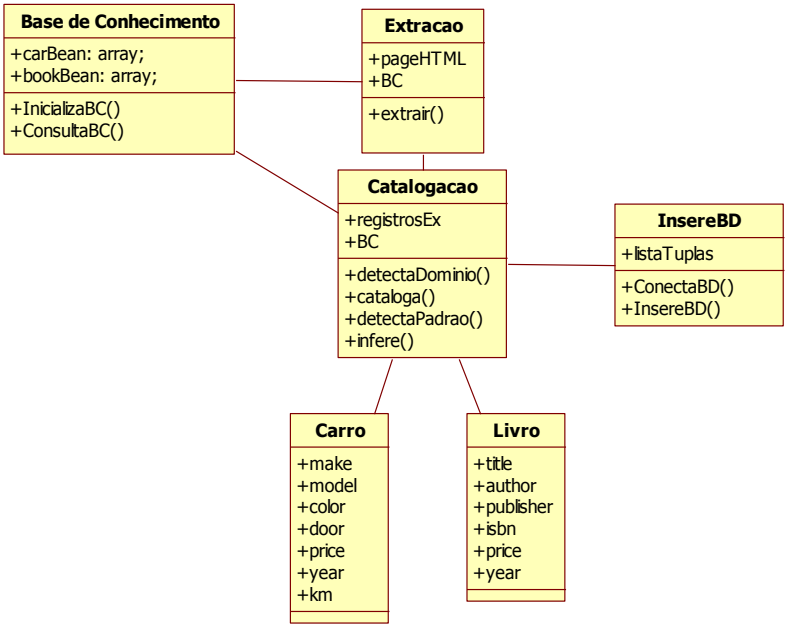
4.6 IMPLEMENTAÇÃO DA *DEEPEC*

DeepEC foi prototipada na linguagem Java. A Figura 22 apresenta o diagrama de classe na linguagem UML com as principais classes implementadas e seus principais métodos. Inicialmente, definiu-se uma classe *Extração* responsável pela extração dos dados da página HTML, utilizando como apoio a BC. Esta classe está associada à classe *Base de Conhecimento (BC)* e a classe *Catalogação*. A classe BC é responsável pela criação da base com os valores dos domínios suportados pela *DeepEC* no momento, sendo utilizada nos processos de

Extração e Catalogação. Já a classe *Catalogação* realiza a identificação do domínio, a catalogação propriamente dita, o reconhecimento de padrões dos termos e a detecção de dados.

As classes *Carro* e *Livro* estão em composição com a classe *Catalogação*, pois cada uma define os seus atributos específicos. Por fim, a classe *InsereBD* realiza a conexão com o BD e a inserção das tuplas catalogadas no mesmo.

Figura 22 – Diagrama de classe da *DeepEC*.



Por fim, cabe ressaltar que, nesta versão da *DeepEC*, um registro considerado relevante está sendo comparado duas vezes com a BC, ou seja, primeiramente durante a etapa de Extração e em seguida na etapa de Catalogação. Isso ocorre porque, durante a extração, procura-se apenas por um provável registro válido como dado de um BD escondido na *Deep Web* em um determinado domínio. Já na catalogação, contextualiza-se este registro para inseri-lo no domínio correto. Neste segundo caso, comparam-se vários itens de dados do registro para ter certeza de qual domínio ele pertence, conforme indicado na Definição 4.

5 EXPERIMENTOS

Este capítulo apresenta os experimentos realizados com a finalidade de avaliar a abordagem *DeepEC*, ou seja, a execução simultânea dos processos de extração e catalogação de dados presentes em BDs na *Deep Web*.

5.1 ORIGEM DOS DADOS

Conforme citado no Capítulo 4, as amostras de dados utilizadas nos experimentos pertencem aos domínios de *Automóveis* e *Livros*. A escolha por estes domínios se justifica pela parceria do Grupo de Banco de Dados[6] da UFSC (GBD/UFSC) com os desenvolvedores da máquina de busca *DeepPeep* (Barbosa et. al 2010). *DeepPeep* é uma máquina de busca para formulários *Web* pertencentes a alguns domínios. As URLs de páginas *Web* pertencentes aos dois domínios escolhidos para os experimentos foram cordialmente cedidos pelos mantenedores do *DeepPeep*. Estes são os domínios com maior volume de dados indexados por esta máquina de busca. Parte destes dados foi utilizada também na população da BC.

Inicialmente, realizou-se o acesso e a submissão de consultas aos formulários existentes em algumas dessas páginas. Com o retorno da submissão, as páginas contendo os resultados da busca foram salvas para serem utilizadas como entrada para a *DeepEC*. Essas páginas geraram 1376 registros no domínio de Automóveis e 352 registros no domínio de Livros.

5.2 MÉTRICAS UTILIZADAS

As métricas utilizadas para a avaliação da qualidade do *DeepEC* são a precisão, a revocação e a medida F, que são métricas consagradas na área de recuperação de informação (Yates e Neto, 1999). A Figura 23 apresenta as fórmulas de cada métrica.

Precisão é a razão entre a quantidade de registros recuperados que são relevantes (registros de BDs escondidos) e o total de registros recuperados. Revocação é a razão entre a quantidade de registros relevantes recuperados e o total de registros relevantes existentes nos BDs. Já a medida F é a média harmônica das duas métricas anteriores.

[6]http://www.gbd.inf.ufsc.br/

Figura 23 – Fórmulas para os cálculos de revocação, precisão e medida F.

$$R = \frac{A}{A+B}, P = \frac{A}{A+C} \text{ and } F = \frac{2(R*P)}{(R+P)}$$

Nas fórmulas acima, “A” corresponde ao número de registros relevantes recuperados, “B” corresponde ao número de registros relevantes não recuperados e “C” ao número de registros irrelevantes recuperados.

5.3 RESULTADOS

Os experimentos avaliaram a qualidade da extração e da catalogação de páginas nos dois domínios escolhidos. Inicialmente, realizaram-se experimentos somente para a etapa de extração. Nesta etapa, os resultados obtidos foram comparados com os resultados gerados pelo MDR e *Road Runner*, dois conhecidos algoritmos para extração de dados estruturados presentes em páginas *Web*, que estão disponíveis para *download*. Posteriormente, verificou-se a eficácia da *DeepEC* com experimentos para a etapa de catalogação utilizando os registros recuperados durante a etapa de extração.

A Tabela 3 apresenta os resultados da etapa de extração da *DeepEC* em comparação com os algoritmos *Road Runner* e MDR. A tabela apresenta os resultados de *precisão*, *revocação* e *medida F*, para os *Domínios* de automóveis e livros e o *Tempo* corresponde ao tempo de processamento de cada algoritmo para a extração dos registros presentes nas páginas de entrada. Observa-se aqui que esses experimentos relacionados com a etapa de extração obtiveram melhores resultados utilizando *make* e *model* como atributos mandatórios para o domínio de Automóveis, bem como *publisher* e *author* para o domínio de Livros.

Tabela 3- Resultados da etapa de extração dos dados.

Algoritmo	Domínio	P	R	F	Tempo (s)
Road Runner	Auto	0,94	0,95	0,95	18,10
MDR		0,92	0,95	0,93	13,56
DeepEC Extração		0,94	0,96	0,95	10,09
Road Runner	Livros	0,84	0,91	0,88	4,95
MDR		0,81	0,90	0,85	3,62
DeepEC Extração		0,87	0,90	0,89	2,88

Nota-se, através dos melhores resultados, que a etapa de extração da *DeepEC* apresenta resultados compatíveis com a precisão e revocação dos trabalhos relacionados. Obteve-se a melhor revocação para o domínio de Automóveis, ou seja, o melhor percentual de acertos em termos de recuperação de registros de BDs da *Deep Web* presentes nas páginas de resultado. Outro ponto positivo da *DeepEC* foi a melhor medida F para o domínio de Livros. Esse desempenho superior era esperado pelo fato do *DeepEC* possuir uma BC para direcionar os registros a serem extraídos. Além disso, a *DeepEC* obteve o melhor tempo de processamento dentre os algoritmos comparados, ou seja, ele realizou a extração de forma mais eficiente que os demais.

Analisando os registros incorretos, a quantidade extraída ocorreu devido à existência de diversos menus/propagandas que possuíam os próprios valores do domínio no seu interior, fazendo com que o extrator recuperasse esses dados. Esse é um ponto negativo que deve ser melhor avaliado e melhorado nos trabalhos futuros da *DeepEC*.

Com relação à justificativa da comparação da *DeepEC* com as abordagens MDR e *Road Runner*, salienta-se que, apesar destas abordagens serem classificadas como extratores sintáticos, por não considerarem a semântica em seus métodos, elasse enquadram dentre as poucas que realizam o processo de extração de forma automática, assim como a *DeepEC*, e possuem implementações disponíveis para uso. Ao mesmo tempo, a *DeepEC*, ao adotar uma abordagem baseada em semântica, com o suporte de uma BC, esperava-se, com os experimentos, alcançar um desempenho no mínimo idêntico ao MDR e ao *Road Runner*. Tal desempenho mínimo ocorreu, conforme apresentado na Tabela 4.

A seguir, apresenta-se, na Tabela 4, o desempenho total da abordagem *DeepEC*, ou seja, os valores de precisão, revocação e medida F, considerando o que é catalogado em relação a quantidade de registros extraídos disponíveis nos arquivos.

Tabela 4- Resultados da extração e catalogação do *DeepEC*.

Domínio	**Precisão**	**Revocação**	**Medida F**
Auto	0,94	0,96	0,95
Livros	0,90	0,93	0,91

Os experimentos comprovam que a *DeepEC* apresentou bons resultados de Precisão e Revocação para os registros que foram extraídos (com medida F superior à 90%), levando em consideração a existência de ruídos gerados pela etapa de extração. Além disso, este bom desempenho se justificou pelo fato de grande parte dos registros catalogados possuírem algum tipo de informação presente na BC, como por exemplo, ocorrências dos principais modelos de carros para o caso do domínio de Automóveis. Cabe ressaltar que, mesmo com a BC contendo uma pequena amostra de valores de alguns atributos mandatórios dos domínios, isso já torna possível a catalogação correta de um conjunto de dados muito mais volumoso, pois basta haver um casamento parcial de conteúdo dos dados da amostra com o registro extraído para ocorrer a catalogação.

Nos experimentos para a catalogação dos dados, os principais casamentos ocorreram com os valores do atributo *model* para o domínio de Automóveis e *publisher* no de Livros, por serem os conteúdos mais presentes nesses domínios. Alguns registros extraídos não foram catalogados devido à inexistência de conteúdo similar na BC. A extensão automática de conteúdo da BC a partir de registros relevantes extraídos é um trabalho futuro iminente.

Como mencionado anteriormente, outra contribuição da *DeepEC* é o enriquecimento das informações extraídas através da detecção de informação ausente nas tuplas catalogadas no BD relacional. Para avaliar esta contribuição, a Tabela 5 apresenta a quantidade de valores de atributos que foram complementados com sucesso para o conjunto de dados considerado nos experimentos em ambos os domínios. Leva-se em conta a quantidade de registros utilizada como entrada na *DeepEC*.

Tabela 5- Ganho com a detecção de informação durante a catalogação.

Domínio	**Atual**	**Valores de Atributos detectados**	**Ganho de registros modificados (%)**
Auto	1376	150	10,90
Livros	352	25	7,10

Conforme apresentado, observa-se que, aplicando o mecanismo de detecção de conteúdo foi possível um ganho de qualidade na catalogação de registros em torno de 10% com a abordagem *DeepEC*. Este percentual de ganho não foi tão elevado para a amostra de dados testada, pois grande parte dos registros a ser catalogada possuía valores para todos os atributos previstos nas tabelas de domínio.

O domínio de Automóveis permitiu mais casamentos com conteúdos da BC e, consequentemente, mais detecções, pois diversos *Websites* omitem a informação do fabricante após a seleção do mesmo no formulário de busca, possibilitando a detecção dessa informação a partir do modelo do veículo, uma vez que a variedade de fabricantes e modelos não é tão extensa na prática. Já o domínio de Livros, por ser mais amplo em termos de variedade de conteúdo e por apresentar páginas com informações mais detalhadas, permitiu menos detecções. Mesmo assim, houve alguns casos de descoberta de valores de atributos.

6 CONCLUSÃO

Esta dissertação apresenta a abordagem *DeepEC*, uma contribuição para a problemática de tornar visível, estruturado e contextualizado o conteúdo escondido presente nos BDs da *Deep Web*.

Diferentemente do estado da arte sobre extração e catalogação de dados da *Deep Web*, que não trata essas atividades como um processo único e possui limitações em termos da extração distinta de metadados e valores, além da falta de complementação de informações, este trabalho apresenta um novo método para extrair registros de dados estruturados de páginas de resultados de consulta sobre *Web sites* da *Deep Web* e também executar a catalogação/complementação dessas informações.

A abordagem proposta se baseia na existência de uma BC projetada para manter metadados e amostras de valores mais relevantes dos principais domínios da *Deep Web*. Ela é utilizada para fins de comparação e inferência de registros extraídos de páginas *Web* com resultados de consultas definidas sobre formulários *Web*.

Por meio de uma avaliação experimental, verificou-se em termos de qualidade de resultado, das atividades de extração e catalogação de dados da *DeepEC*. Os resultados indicaram que a nossa abordagem executa no mínimo com qualidade similar ao dos principais métodos de extração automáticos, porém com um melhor desempenho, e com uma ótima atuação para a atividade de catalogação nos domínios considerados. Além disso, *DeepEC* proporcionou um ganho de até 10% no enriquecimento do BD para os dados catalogados. Estima-se que este ganho seja maior para um volume maior de registros em domínios cujo conteúdo seja mais homogêneo, como é o caso do domínio de Automóveis, um dos mais extensos da *Deep Web*. Deseja-se realizar novos experimentos com maior volume de dados extraídos de páginas *Web* para validar esta hipótese.

A Tabela 6 apresenta uma atualização do comparativo dos trabalhos relacionados à extração de dados na *Web*, classificando a abordagem do *DeepEC* com os critérios definidos anteriormente. Conforme ressaltado nesta Tabela, o diferencial do componente de extração da *DeepEC* é a utilização de suporte semântico para a realização da extração de forma automática.

Tabela 6- Comparativo das técnicas de extração de dados na *Web*.

Abordagem	**Principais Soluções**	**Grau de automatização**	**Tipo de conteúdo**	**Formatos de Saída**
Árvore	MDR	Automática	SD	HTML
Wrappers	Road Runner	Automática	SD	XML, HTML
	XWrap	Automática	SD	XML
Machine Learning	Liu et al. 2000	Semiautomática	SD	HTML
Ontologia/ BC	Phan et al. 2005	Manual	ST/SD	HTML
	DeepEC	**Automática**	**SD**	**Lista de registros**

Com relação aos trabalhos que realizam a catalogação de dados, a *DeepEC* possui o diferencial de realizar a detecção de valores que não estão presentes nos registros extraídos.

Cabe ressaltar que a contribuição efetivada *DeepEC* é a sua aplicabilidade como um mecanismo gerador de dados estruturados da *Deep Web* para aplicações que realizam buscas e/ou integração de dados desta natureza, assim como um mecanismo que facilite a geração de catálogos de BDs escondidos por domínio. Além disso, a *DeepEC* pode ser estendida para a extração de outras fontes de dados na *Web*, como listas, considerando a BC enriquecida com novos domínios.

Diversos trabalhos futuros estão em planejamento.

- Aplicar e avaliar o *DeepEC* em outros domínios populares da *Deep Web*, como hotéis, vendas de passagens e ofertas de emprego;
- Verificar, através de experimentos com novos domínios, a acurácia das heurísticas propostas, em especial, a heurística de inferência de domínio;
- Realizar a extração e identificação de rótulos de atributos eventualmente presentes nos registros extraídos, visando enriquecer os metadados da BC e facilitar a catalogação do conteúdo;
- Permitir a extensão da BC a partir de novos registros extraídos, ampliando, assim, o seu conhecimento;
- Detectar e catalogar registros relevantes de certo domínio mesmo sem haver informações (conhecimento) parciais sobre ele na BC. Esta detecção poderia ser feita por

comparação de posicionamento do conteúdo do registro. Por exemplo, se um registro possui um valor "Chevrolet" e este valor está na mesma posição do valor "Ford" num outro registro, poder-se-ia detectar que "Chevrolet" também é uma marca;

- Definir um índice por similaridade para o acesso aos registros da BC, utilizando como base outro trabalho desenvolvido no GBD/UFSC (Koerich e Mello, 2013);
- Implementar uma interface para consulta ao BD populado pela *DeepEC*. Este suporte é importante para sistemas que desejam realizar buscas sobre dados da *Deep Web*, como é o caso do projeto WF-Sim, em desenvolvimento no GBD/UFSC e que visa a construção de um sistema de busca por similaridade a dados de formulários *Web* (Gonçalves et al. 2011);
- Avaliar o desempenho, em termos de tempo de processamento durante o processo de catalogação da *DeepEC*;
- Comparar o desempenho, em termos qualitativos, da *DeepEC* com os outros métodos de catalogação apresentados no Capítulo 3, caso se obtenha acesso às suas implementações.

Esta dissertação de mestrado resultou em uma publicação premiada como melhor trabalho de pesquisa na VIII Escola Regional de Banco de Dados (ERBD 2012) (Souza e Mello, 2012) e uma publicação no 21st *European Conference on Information Systems* (conferência classificada como B1 no Qualis da CAPES) (Souza e Mello, 2013) que utilizou dados extraídos dos algoritmos MDR e *Road Runner*, pois o componente de extração do *DeepEC* não estava finalizado. Além disso, essa dissertação serviu de motivação para alguns trabalhos de conclusão de curso de Graduação em Ciências da Computação da UFSC. Um dos trabalhos desenvolveu o componente de extração de dados da *DeepEC*, trabalho este ainda em aprimoramento. Outro trabalho está projetando e populando a BC de forma automática a partir de bases de dados de conhecimento universais disponíveis, no caso, a *Freebase* e a *Wikipédia*.

REFERÊNCIAS BIBLIOGRÁFICAS

AGICHTEIN, E. and GANTI, V. (2004). **Mining Reference Tables for Automatic Text Segmentation**. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Seattle, USA, p. 20-29.

ALBUQUERQUE, Ícaro. ***Deep Web*: o lado oculto da Internet**. Disponível em http://all-your-tech.blogspot.com.br/2013/01/deep-web-o-lado-oculto-da-internet.html Acesso em: 27 Jun. 2013.

AKILANDESWARI, J. and GOPALAN, N.P. (2007). **A Novel Design of Hidden *Web* Crawler Using Reinforcement Learning Based Agents**. In Proceedings of the 7th international conference on Advanced parallel processing technologies, p. 433-440.

ÁLVAREZ, M. RAPOSO, J., PAN, A., CACHEDA, F., BELLAS, F. and CARNEIRO, Víctor (2007). **DeepBot: A Focused Crawler for Accessing Hidden *Web* Content**. In Proceedings of the 3rd International Workshop on Data Enginering Issues in E-Commerce and Services (DEECS), p. 18 - 25.

BARBOSA, L., NGUYEN, H., NGUYEN, T., PINNAMANENI , R., and FREIRE, J. (2010). **Creating and Exploring *Web* Form Repositories**. In Proceedings of the ACM SIGMOD International Conference on Management of data, p. 1175-1178.

BERGMAN, M. K. (2001). **The *Deep Web*: Surfacing Hidden Value**. In The Journal of Electronic Publishing, Vol. 7, No. 1.

BOISSON, P., CLERC, S., DESCONNETS, J.-C. and LIBOUREL, L. (2006). **Using a Semantic Approach for a Cataloguing Service**. On the Move to Meaningful Internet Systems, Lecture Notes in Computer Science (LNCS) Vol. 4278 p. 1712-1722.

CHENG, Tao and CHANG, Kevin ChenChuan (2007). **Entity Search Engine: Towards Agile Best Effort Information Integration over the *Web***. In Conference on Innovative Data Systems Research (CIDR), p. 108-113.

CHIANG, F., ANDRITSOS, P., ZHU, E. and Miller, R. J. (2012). **AutoDict: Automated Dictionary Discovery**. In International Conference on Data Engineering (IEEE), p. 1277-1280.

CORTEZ, E., Silva, A. S., GONÇALVES, M. A. and Moura, E. S. (2010). **ONDUX: On-Demand Unsupervised Learning for Information Extraction**. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, p. 807-818.

DORNELES, C. F., HEUSER, C. A., ORENGO, V. M., SILVA, A. S., MOURA, E. S. (2009). **A strategy for allowing meaningful and comparable scores in approximate matching**. In Information System, vol 34, n 08, p. 673-689.

EMBLEY, D. W., CAMPBELL, D. M. and SMITH, R. D. (1998). **Ontology-Based Extraction and Structuring of Information from Data-Rich Unstructured Documents**. In CIKM Proceedings of the seventh International Conference on Information and Knowledge Management, p. 52-59.

FERRARA, E., FIUMARA, G., MEO, P. and BAUMGARTNER, R. (2012). ***Web* Data Extraction, Applications and Techniques: A Survey**. In ACM Computing Surveys, Vol. V, July 2012, p. 1-48.

GONÇALVES, R., D'AGOSTINI, C. S., Silva, F. R., DORNELES, C. F. and MELLO, R. S. (2011). **A Similarity Search Method for *Web* Forms**. In IADIS International Conference WWW/Internet, p. 381-387.

HALEVY, A., MADHAVAN, J., AFANASIEV, L. and ANTOVA, L. (2009). **Harnessing the *Deep Web*: Present and Future**. In Conference on Innovative Data Systems Research (CIDR).

HE, B., PATEL, M., ZHANG, Z. and CHANG, K. C-C. (2007). **Accessing the *Deep Web***. In Communications of the ACM. Vol. 50, No. 5, p. 95-101.

HONG, Jun; HE, Zhongtian and BELL, David A. (2009). **Extracting *Web* Query Interfaces Based on Form Structures and Semantic Similarity**. In Proceedings of the IEEE International Conference on Data Engineering (ICDE), p. 1259-1262.

HONG, Jun; HE, Zhongtian and BELL, David A. (2010). **An Evidential Approach to Query Interface Matching on The *Deep Web***. In Journal Information Systems, p. 140-148.

HU, X., ZHANG, X., LU, C., PARK, E. K., and ZHOU, X. (2009). **Exploiting Wikipedia as external knowledge for document clustering**. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris, France, p. 389-396.

JARGAS, A.M. (2012). **Expressões Regulares - Uma abordagem divertida**. Novatec, 4ª Edição.

KAISER, K. and MIKSCH, S. (2005). **Information extraction. A survey**. Tech.rep., E188 – Institute of Software Technology & Interactive Systems. Vienna University of Technology.

KIM, Y., PARK, J., KIM, T. and CHOI, J. (2007). ***Web* information extraction by html tree edit distance matching**. In International Conference on Convergence Information Technology. IEEE, p. 2455-2460.

KOERICH, W. V. e MELLO R. S. (2013). **Um Método para Indexação de Formulários *Web* visando Consultas por Similaridade**. In Escola Regional de Banco de Dados (ERBD).

LAENDER, A. H. F., RIBEIRO-NETO, B. A., SILVA, A. S. e TEIXEIRA, J. S. (2002). **A Brief Survey of *Web* Data Extraction Tools**. In SIGMOD Record, Vol.31, N° 2, p. 84-93.

LIU, L., PU, C. and HAN, W. (2000). **XWRAP: An XML-enable Wrapper Construction System for *Web* Information Sources**. In 16th International Conference on Data Engineering (ICDE'00), p. 611-621.

LIU, B., GROSSMAN, R. and ZHAI, Y. (2003). **Mining Data Records in *Web* Pages**. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, p. 601-606.

MELLO, R. S., Pinnamaneni, R., Freire, J. (2010). **Indexing Web Form Constraints**. In Journal of Information and Data Management, Vol. 1, Nº 3, p. 343-358.

MENG, X., LIU,W. and MENG, W. (2010). **ViDE: A Vision-Based Approach for *Deep Web* Data Extraction**. In IEEE Transactions on Knowledge and Data Engineering, p. 447-460.

MILLER, George A. (1995). **WordNet: A Lexical Database for English**. In Communications of the ACM Vol. 38, N°. 11, p. 39-41.

MUSLEA, I., MINTON, S. and KNOBLOCK, C. (2001). **Hierarchical Wrapper Induction for Semistructured Information Sources**. In Journal Autonomous Agents and Multi-Agent Systems archive Vol. 4 Issue 1-2, p. 93-114.

NGUYEN, Hoa; NGUYEN, Thanh and FREIRE, Juliana (2008). **Learning to Extract Form Labels**. In Proceedings of the VLDB Endowment, p. 684-694.

NGUYEN, Hoa; NGUYEN, Thanh and FREIRE, Juliana (2010). **PruSM: A Prudent Schema Matching Approach for *Web* Forms**. In Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM), p. 1385-1388.

NONAKA, I., TAKEUCHI, H. (1997). **Criação de Conhecimento na Empresa: como as empresas japonesas geram a dinâmica da inovação**. Rio de Janeiro: Campus, p. 97.

ORO, E. and RUFFOLO, M. (2011). **SILA: a Spatial Instance Learning Approach for *Deep Web* Pages**. In Conference on Information and Knowledge Management (CIKM), p. 2329-2332.

PHAN, X., HORIGUCHI, S. and HO, T. (2005). **Automated Data Extraction from the *Web* with Conditional Models**. International Journal of Business Intelligence and Data Mining, Vol. 1, p. 194-209.

RAGHAVAN, S. and MOLINA, H. G. (2001). **Crawling the Hidden *Web***. In Proceedings of the 27th International Conference on Very Large Data Bases (VLDB), p. 129-138.

SERRA, E., CORTEZ, E. SILVA, A. S. and MOURA, E. S. (2011). **On Using Wikipedia to Build Knowledge Bases for Information Extraction by Text Segmentation**. In Journal of Information and Data Management, Vol. 2, No. 3, p. 259–272.

SILVA, A. S., CORTEZ, E., OLIVEIRA, D., MOURA, E. S. and LAENDER, A. H. F. (2011). **Joint Unsupervised Structure Discovery and Information Extraction**. In Special Interest Group on Management of Data (SIGMOD), p. 12-16.

SOUZA, A. F., and MELLO, R. S. (2012). **Análise de Abordagens para Matching de Formulários na *Deep Web***. In Escola Regional de Banco de Dados (ERBD).

SOUZA, A. F. e MELLO, R. S. (2013). ***DeepEC*: An Approach for *Deep Web* Content Extraction and Cataloguing**. In 21st European Conference on Information Systems (ECIS), Utrecht.

TODA, G., CORTEZ,E., SILVA, A.S., MOURA, E.S. (2010). **A Probabilistic Approach for Automatically Filling Form-Based Web Interfaces**. Proceedings of the Very Large data Bases PVLDB, p. 151-160.

YATES, R. B. and NETO, B. R. (1999). **Modern Information Retrieval**. First Edition. Addison Wesley Longman Limited, England.

WINKLER, W. E. (1990). **String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage**. In Proceedings of the Section on Survey Research Methods, American Statistical Association, p. 354-359.

ZHAI, Y. and LIU, B. (2005). **Web Data Extraction Based on Partial Tree Alignment**. In Proceedings of the 14th international conference on World Wide *Web* (WWW), p. 76-85.

ZHAO, C., MAHMUD, J. and RAMAKRISHNAN, I. V. (2008). **Exploiting Structured Reference Data for Unsupervised Text Segmentation with Conditional Random Fields**. In International Conference on Data Mining (SIAM), p. 420-431.

ZHENG, Z., SI, X., LI, F., CHANG, E. Y. and ZHU, X. (2012). **Entity Disambiguation with Freebase**. In International Conferences on *Web* Intelligence and Intelligent Agent Technology (IEEE/WIC/ACM), p. 82-89.