# Limpeza de Dados - Uma Visão Geral

Paulo Jorge Oliveira, Fátima Rodrigues

GECAD – Grupo de Engenharia do Conhecimento e Apoio à Decisão
Departamento de Engenharia Informática
Instituto Superior de Engenharia – Instituto Politécnico do Porto
Porto, Portugal
{pjo,fr}@isep.ipp.pt

Pedro Rangel Henriques

GEPL – Grupo de Especificação e Processamento de Linguagens
Departamento de Informática
Universidade do Minho
Braga, Portugal
prh@di.uminho.pt

**Resumo.** No contexto da actual necessidade de explorar bases de dados, para delas extrair informação/conhecimento para apoio à gestão, é fundamental a correcção/validade dos dados para a qualidade dos resultados extraídos. Sendo certo que são várias as soluções parciais para a resolução dos problemas nos dados, tornou-se necessário fazer uma sistematização de todos os erros que podem ocorrer no sentido de identificar aqueles ainda não resolvidos e preconizar uma abordagem global. Este artigo descreve precisamente o referido estudo, e as ilações que se extraem quando se comparam os erros com as abordagens de limpeza de dados actualmente existentes, perspectivando-se a concepção de uma nova aproximação global à limpeza de dados como trabalho futuro, em consequência das conclusões obtidas.

## 1. Introdução

As organizações públicas e privadas começam, finalmente, a perceber o valor dos dados que têm à sua disposição, e a considerá-los como um bem importante no aumento da produtividade, eficiência e competitividade. Como consequência, a exploração de enormes volumes de dados assume um papel cada vez mais importante na sociedade actual. Por exemplo, a consolidação de dados, a partir de fontes dispersas, num armazém de dados central permite às organizações executarem operações de análise de dados, e assim obterem informações que são de importância estratégica e táctica para as suas actividades [1, 2].

Além dos armazéns de dados, existem outros produtos de *software* que procuram tirar partido dos dados existentes como as ferramentas de: análise de dados; exploração de dados (em inglês: *data mining*); e gestão do relacionamento com o cliente (em inglês: *customer relationship management*). Todos estes produtos re-

querem um elevado grau de qualidade dos dados, uma vez que são utilizados na tomada de decisão. A correcção dos dados é vital para que se possam alcançar conclusões acertadas.

No entanto, constata-se que uma boa parte dos dados na generalidade das fontes apresenta erros ou anomalias ("sujidade"). Entre outras possibilidades, as anomalias nos dados correspondem a valores de atributos em falta, valores de atributos errados, ou representações diferentes dos mesmos dados. As anomalias nos dados criam problemas à sua efectiva utilização, influenciando negativamente a validade dos resultados e conclusões obtidas. Isto redunda num custo maior e num proveito menor para o utilizador. Assim sendo, antecedendo a aplicação de qualquer ferramenta orientada à análise, os dados devem ser "limpos" com o intuito de remover e reparar quaisquer anomalias que possam existir. Neste contexto, a limpeza de dados (em inglês: *data cleaning*, *data cleansing, data scrubbing* ou *data reconciliation*) tem sido objecto de um interesse crescente ao longo dos últimos anos. A limpeza de dados visa detectar e remover anomalias dos dados com o objectivo de aumentar/melhorar a sua qualidade [3].

Os problemas de qualidade podem surgir em conjuntos de dados isolados como ficheiros e bases de dados, sendo ainda mais críticos quando múltiplas fontes de dados necessitam de ser integradas. Isto acontece em virtude das diversas fontes frequentemente conterem dados redundantes sob diferentes representações. De modo a possibilitar-se um acesso preciso e consistente aos dados, é necessário proceder à consolidação das suas diferentes representações e eliminar todas as duplicações. Além da eliminação de duplicados, um processo mais abrangente de integração envolve a transformação de dados no formato desejado e a validação de restrições dependentes do domínio.

Ainda que a generalidade da investigação se centre sobre tradução e integração de dados, a limpeza de dados tem recebido pouca atenção entre a comunidade científica. Um número considerável de autores focalizou-se sobre o problema da identificação e eliminação de duplicados. Alguns grupos de investigação concentraram-se em problemas genéricos não limitados a esta área mas cuja aplicação é relevante na limpeza de dados, como determinadas aproximações de exploração de dados. Mais recentemente, fruto de diversos esforços de investigação foram propostas manipulações mais uniformes para a limpeza de dados, cobrindo diversas fases de transformação, operadores específicos e a sua implementação.

Neste artigo pretendemos sistematizar os erros que podem ocorrer nos dados (secção 2), cruzando-os (secção 4) com as abordagens de limpeza de dados existentes (secção 3), com vista à proposta de uma abordagem geral (secção 5).

## 2. Problemas de Qualidade dos Dados

Nesta secção serão sistematizados de uma forma tão exaustiva quanto possível os problemas nos dados – em termos absolutos do valor dos atributos numa tabela (sec. 2.1), ou em termos relativos, resultantes do relacionamento entre tabelas (sec. 2.2) – que interferem directamente na qualidade das análises que podem ser realizadas sobre

os dados. Problemas esses que, ou são corrigidos na fonte, ou inviabilizam potenciais estudos analíticos.

A fim de apresentar os resultados do estudo feito, os problemas nos dados são descritos detalhadamente sendo divididos nos dois grandes grupos acima identificados e estes ainda subdivididos no primeiro caso, ao nível do atributo, do tuplo e da tabela, seguindo o espírito da classificação apresentada em [3]. Optou-se por sistematizar a sua apresentação através de tabelas de quatro colunas em que a primeira contém um identificador do problema, para fácil referenciação, na segunda é feita uma caracterização do problema, na terceira coluna é apresentado um exemplo concreto, e na última é feita uma breve descrição do problema.

## 2.1 Numa só Tabela/Ficheiro

Neste grupo incluem-se todos os problemas de qualidade dos dados que é possível encontrar num conjunto de dados, materializado sob a forma de uma tabela ou ficheiro, quando analisado isoladamente.

### 2.1.1 Ao Nível do Atributo/Campo

| Id. | Problema | Exemplo | Comentário |
|---|---|---|---|
| P211a | valor em falta | | falta de preenchimento de atributos obrigatórios |
| p211b | valor ilegal | idade = 233 | fora do domínio de valores válidos (atributos numéricos e de enumeração) |
| p211c | valor incorrecto | idade = 25, mas idade é igual a 27 | o valor do atributo não corresponde à situação real |
| p211d | erro ortográfico | cidade = 'Brga' | esta anomalia encontra-se associada a atributos textuais |
| P211e | informação além do contexto do atributo | morada='Rua... 4200-123 Porto' | múltiplos valores introduzidos num único atributo textual |
| P211f | valor de significado indefinido | empresa = 'MS' | esta anomalia resulta da utilização de abreviaturas |
| P211g | utilização de sinónimos | profiss = 'professor' profiss = 'docente' | expressões sintacticamente diferentes, mas iguais ao nível semântico |
| P211h | inexistência de uma representação standard | data = 04/05/2004 data = 2004/05/04 | o valor do atributo surge sob variados formatos |

### 2.1.2 Ao Nível do Tuplo/Registo

| Id. | Problema | Exemplo | Comentário |
|---|---|---|---|
| p212a | violação da dependência entre atributos | idade = 30<br>dtnasc='15/03/1970' | existência de inconsistências entre os valores dos atributos |
| p212b | troca de valores entre atributos | nome = 'Rua ...'<br>morada = 'José ...' | introdução de valor no atributo errado |
| p212c | tuplo semi-vazio ou vazio | | a maioria dos atributos do tuplo não se encontra preenchida |

### 2.1.3 Ao Nível da Tabela/Ficheiro

| Id. | Problema | Exemplo | Comentário |
|---|---|---|---|
| P213a | valor não único | tuplo X: BI=121212<br>tuplo Y: BI=121212 | duas entidades diferentes possuem valores iguais num atributo de valor único |
| P213b | redundância sobre a mesma entidade | nome = 'José Alves'<br>nome = 'J. Alves' | a mesma entidade encontra-se representada sob formas iguais ou diferentes em vários tuplos |
| P213c | inconsistência sobre a mesma entidade | dtnasc='08/09/1973'<br>dtnasc='25/01/1973' | conflitos nos valores dos atributos de uma entidade representada em mais de 1 tuplo |

## 2.2 Em Múltiplas Tabelas, Base de Dados ou Ficheiros

Neste grupo incluem-se os problemas de qualidade dos dados que não resultam directamente de erros ou malformações absolutas num conjunto de dados individual, mas que advêm de vários conjuntos de dados (tabelas, ficheiros ou bases de dados), quando analisados como um todo.

| Id. | Problema | Exemplo | Comentário |
|---|---|---|---|
| p22a | referência inexistente | nomefunc = 'José Alves', coddep = 7 (inexistente na tabela relacionada) | trata-se de uma violação à integridade referencial |
| p22b | referência existente mas errada | nomefunc = 'José Alves', coddep = 7 (não pertence a este departamento) | existe um valor no atributo que não viola a integridade referencial, mas que não é o correcto (problema de actualização) |
| p22c | redundância sobre a mesma entidade | tabela funcionários: nome = 'José Alves'<br>tabela depcomercial: nome = 'J. Alves' | a mesma entidade encontra-se representada sob formas iguais ou diferentes em várias tabelas |

| Id. | Problema | Exemplo | Comentário |
|---|---|---|---|
| p22d | inconsistência sobre a mesma entidade | tabela funcionários: dtnasc='08/09/1973' tabela depcomercial: dtnasc='25/01/1973' | conflitos nos valores dos atributos de uma entidade representada em mais do que uma tabela |
| p22e | formatos de representação diferentes | tabela X: data = 04/05/2004 tabela Y: data = 2004/05/04 | o valor do atributo surge sob variados formatos de representação consoante a tabela |
| p22f | unidades de medida diferentes | tabela X: valor=123 (euros) tabela Y: valor=123 (dólares) | o valor do atributo surge sob variadas unidades de medida consoante a tabela |
| p22g | utilização de sinónimos | tabela X: profis = 'Professor' tabela Y: profis = 'Docente' | expressões sintacticamente diferentes, mas iguais ao nível semântico em tabelas diferentes |

## 3. Limpeza de Dados

A limpeza de dados visa detectar e remover anomalias dos dados com o objectivo de aumentar/melhorar a sua qualidade [3]. Tipicamente o processo de limpeza de dados não pode ser executado sem o envolvimento de um perito do domínio, uma vez que a detecção e correcção de anomalias requer conhecimento especializado. Como tal, este processo é por natureza semi-automático, devendo ser o mais automatizado possível em virtude dos grandes volumes de dados geralmente processados e do tempo necessário para que um perito proceda à sua limpeza manual. A limpeza de dados é então um processo semi-automático de operações realizadas nos dados que: (1) executa a adaptação de formatos nos tuplos e valores; (2) força as restrições de integridade; (3) deriva valores em falta a partir dos existentes; (4) remove conflitos nos tuplos ou entre os tuplos; (5) funde e elimina duplicados; e (6) detecta desvios, isto é, tuplos e valores com um grande potencial de serem inválidos.

Uma aproximação à limpeza de dados deve satisfazer diversos requisitos. O mais importante destes consiste na detecção e remoção de todas as principais anomalias quer ao nível das fontes individuais quer após a integração de múltiplas fontes. A aproximação deve ser suportada por ferramentas que limitem a análise manual e o esforço de programação, e ser extensível por forma a cobrir facilmente novas fontes.

Na sequência do estudo do estado da arte realizado, concluímos que as abordagens existentes para limpeza de dados podem ser classificadas em dois grandes grupos: especializadas e genéricas.

### 3.1 Abordagens Especializadas

As abordagens especializadas debruçam-se sobre uma determinada área da limpeza de dados, como é o caso da limpeza de nomes e endereços postais, ou centram-se sobre um problema concreto, como acontece com a detecção de duplicados.

#### 3.1.1 Correcção de Nomes e Endereços

A limpeza de nomes e endereços postais assume particular importância no domínio da gestão do relacionamento com o cliente. Um número considerável de ferramentas comerciais (por exemplo: *idcentric*, *pureintegrate*, *quickaddress*, *reunion, trillium* e *vality*) visa a limpeza deste tipo de dados. Estas ferramentas possibilitam a extracção e transformação dos elementos que compõem o nome e o endereço, procedendo à validação dos nomes próprios, apelidos, nomes das ruas, localidades e códigos postais e à sua representação num formato predefinido. A validação é efectuada com base em bibliotecas de regras previamente especificadas, que explicitam a forma de solucionar os problemas vulgarmente encontrados nestes tipos de dados. Além das ferramentas comerciais, também alguns trabalhos de investigação se tem debruçado sobre estes problemas específicos, como é o caso de [4] e [5].

#### 3.1.2 Detecção de Duplicados

Nesta área da limpeza de dados, a detecção de duplicados, também conhecido pelo problema de identificação do objecto, tem sido alvo de inúmeros trabalhos de investigação e originado diversas ferramentas comerciais (por exemplo: *datacleanser*, *matchit* e *mastermerge*). Cada método de detecção proposto envolve um algoritmo para determinar se dois ou mais tuplos são representações da mesma entidade. Para que a detecção de duplicados seja eficiente, cada tuplo necessita de ser comparado com todos os outros tuplos, ou seja, recorrendo ao produto cartesiano. Em [6] é apresentado um método eficiente que reduz o número de comparações necessárias, denominado de método da vizinhança ordenada (em inglês: *sorted neighbourhood method*). Os tuplos são ordenados em função de uma chave construída a partir dos atributos da tabela, na convicção de que os tuplos duplicados ficarão perto uns dos outros. De seguida, apenas os tuplos que se encontram no interior de uma janela de tamanho fixo é que são comparados entre si para detectar os duplicados. A janela vai-se deslocando ao longo dos tuplos que fazem parte da tabela. A classificação de um par de tuplos como duplicados é efectuada com base em regras, representando conhecimento específico do domínio. De modo a melhorar a precisão, os resultados das diversas passagens para detecção de duplicados podem ser combinados através da transitividade entre todos os pares de duplicados encontrados, sendo esta abordagem conhecida por método multi-passagem sobre a vizinhança ordenada [6]. Além deste trabalho de referência obrigatória na área, muitos outros trabalhos de investigação encontram-se conectados com este problema específico. Em [7] e [8] demonstra-se como os métodos de aprendizagem automática podem ser aplicados em situações de identificação de duplicados, aquando da existência de amostras de treino. Em [9] e [10] apresentam-se métodos para efectuar a identificação de duplicados em fontes de dados de elevada dimensão. Em [11] são apresentados métodos escaláveis e adapta-

tivos cujo objectivo é o de segmentar e de estabelecer a correspondência entre entidades.

### 3.2 Abordagens Genéricas

Designamos por abordagens genéricas aquelas que sistematizam a limpeza de dados, cobrindo um número mais vasto de problemas ou adaptando-se a diferentes domínios.

#### 3.2.1 AJAX

Na base do sistema AJAX [12, 13] encontra-se uma arquitectura flexível e extensível que procura separar os níveis lógico (especificação das operações de limpeza de dados a realizar) e físico (aspectos relacionados com a implementação) de um processo de limpeza de dados. O seu principal objectivo consiste em transformar dados de uma ou várias fontes num determinado esquema alvo, manipulando uma série de problemas típicos de qualidade dos dados (por exemplo: eliminação de duplicados) durante o processo.

A definição de uma arquitectura em que a lógica de um processo de limpeza de dados é modelada como um grafo dirigido de transformações sobre fluxos de dados é o primeiro aspecto relevante desta abordagem. O processo de limpeza de dados recebe um conjunto de fluxos de dados, possivelmente errados e/ou inconsistentes e origina um conjunto de fluxos de dados formatados, correctos e consistentes.

Uma linguagem declarativa e extensível, baseada em declarações SQL devidamente enriquecidas com um conjunto de primitivas de transformação, possibilita a especificação das transformações de dados (programas de limpeza de dados) de uma forma compacta e de manutenção simplificada. No AJAX encontram-se definidas cinco transformações: **vista SQL** (*SQL view*) – equivale a uma interrogação SQL típica, permitindo especificar uniões e junções SQL; **mapeamento** (*map*) – uniformiza o formato dos dados (por exemplo: datas) ou simplesmente funde ou divide atributos de modo a colocá-los num formato mais adequado; **correspondência** *(match)* – determina pares de tuplos que, com grande probabilidade, se referem à mesma entidade/objecto; **segmentação** (*cluster*) – com base nos resultados da transformação anterior, agrupa os pares de tuplos que possuem um elevado grau de semelhança, com base num determinado critério de agrupamento (por exemplo: transitividade); e **fusão** (*merge*) – aplicada sobre cada segmento de tuplos com o objectivo de eliminar duplicados, naquilo que se designa de *operação de consolidação*.

A semântica de cada uma destas transformações envolve a geração de excepções sobre situações anormais que possam ocorrer (erros ou inconsistências). Estas excepções são o alicerce de um ambiente interactivo com o utilizador, na manipulação dessas situações. O ambiente também permite a análise dos resultados intermédios, durante a execução de um processo de limpeza de dados.

O último aspecto relevante consiste na existência de um mecanismo genealógico dos dados, o que possibilita a obtenção de explicações. Para cada transformação de dados, o utilizador pode obter informação sobre quais os tuplos que estiveram na base da geração de um determinado tuplo.

Esta abordagem procura tirar o máximo partido das funcionalidades disponíveis nos Sistemas de Gestão de Bases de Dados, designadamente: linguagem, execução e optimização. No entanto, como seria de esperar, não abarca todos os aspectos envolvidos na limpeza de dados.

### 3.2.2 Arktos

O Arktos [14] é uma ferramenta que possibilita a modelação e execução de cenários de Extracção, Transformação e Carregamento (ETC) para a criação de armazéns de dados, com base num conjunto de primitivas que permitem a realização de tarefas usuais. Entre estas, além de operações de transformação, encontram-se também operações de limpeza de dados, consideradas como uma parte integrante do processo de ETC. Este processo consiste numa sequência de passos que extraem dados relevantes das fontes, efectuam a sua transformação para o formato pretendido, procedem à sua limpeza e, por último, executam o seu carregamento para o armazém de dados. As operações durante o processo de ETC são denominadas de actividades. Uma actividade constitui uma unidade atómica de trabalho, apresentando-se como um passo na sequência de operações do processo. Uma vez que a finalidade de uma actividade é efectuar processamento sob um fluxo de dados, cada uma destas encontra-se conectada a tabelas de entrada e de saída, de uma ou mais bases de dados. A cada actividade encontra-se também associada um tipo de erro particular e uma política. A lógica subjacente a uma actividade é descrita declarativamente através de uma instrução SQL. No entanto, não é obrigatório que a sua execução seja efectuada como tal.

O Arktos fornece uma variedade de primitivas (actividades) que correspondem às operações mais usuais de transformação e limpeza. As primitivas de limpeza disponíveis são: verificação da violação de chave primária; verificação de violação de referência; verificação da existência do valor nulo; verificação da violação de unicidade; verificação de violação do domínio.

### 3.2.3 IntelliClean

O *IntelliClean* [15, 16] assenta numa arquitectura genérica baseada em conhecimento para a limpeza inteligente dos dados (uniformização, detecção e remoção de anomalias), com especial ênfase na eliminação de duplicados. A arquitectura pode ser aplicada sobre qualquer base de dados, permitindo a implementação de qualquer estratégia de limpeza de dados actualmente existente. Estas estratégias, traduzindo conhecimento sobre o domínio, são representadas sob a forma de regras, sendo a sua aplicação efectuada através de um motor de inferência de um sistema pericial. A arquitectura especifica três fases distintas para o processo de limpeza de dados: pré-processamento, processamento e verificação e validação humana.

Na fase de pré-processamento, os tuplos são analisados e todas as anomalias sintácticas susceptíveis de serem detectadas nesta fase são corrigidas. Entre estas encontram-se verificações ao tipo de dados, uniformização de formatos e adopção de uma representação consistente para as abreviaturas.

A fase de processamento envolve a avaliação das regras de limpeza sobre os tuplos pré-processados, que alimentam o mecanismo de inferência do sistema pericial (os tuplos são os factos). As regras especificam a realização de determinadas acções me-

diante a ocorrência de determinadas situações nos tuplos, podendo conter predicados complexos e referências a funções externas tanto no antecedente como no seu consequente. As várias funções necessárias ao processo de limpeza de dados encontram-se organizadas por categorias de regras: **regras de identificação de duplicados** – especificam as condições que é necessário respeitar para que dois tuplos possam ser classificados como duplicados; **regras de fusão** – definem a forma de manipulação dos tuplos duplicados; **regras de actualização** – especificam a forma como os dados vão ser actualizados numa determinada situação; e **regras de alerta -** definem as condições sob as quais o utilizador é notificado, perante a ocorrência de um determinado evento.

Na fase de verificação e validação humana, como o nome o indica, é necessária a intervenção humana para analisar o ficheiro de registo. Este ficheiro permite verificar a consistência e precisão das acções efectuadas e, eventualmente, efectuar a sua correcção.

### 3.2.4 Potter's Wheel

O *Potter's Wheel* [17] é um protótipo de investigação baseado numa arquitectura interactiva simples mas poderosa para a transformação e limpeza de dados. Integrando intimamente transformação e detecção de discrepâncias nos dados, permite aos utilizadores gradualmente procederem à composição e análise do efeito das transformações num interface gráfico e intuitivo do tipo folha de cálculo. O seu principal objectivo passa por intercalar a descoberta de discrepâncias nos dados com a sua correcção.

O sistema permite a especificação gráfica de variadas transformações de dados. O utilizador define os resultados pretendidos sobre uma amostra de dados e, com base nestes, automaticamente são inferidas as expressões que os representam. Desta forma, não é necessária a sua prévia definição. O formato inferido é, então, usado na detecção de discrepâncias. A especificação do processo de limpeza de dados é efectuada interactivamente sob a forma de um conjunto de transformações simples aplicados sobre a amostra de dados. O efeito de uma transformação pode ser observado de imediato, pelo utilizador, nos tuplos visíveis no ecrã. Não é necessário aguardar pela transformação de todo o conjunto de dados para se analisar as suas consequências. Em simultâneo, algoritmos de exploração de dados e algoritmos específicos do domínio pesquisam incrementalmente, em segundo plano, a existência de discrepâncias na última versão transformada dos dados, assinalando-as à medida que estas são encontradas.

Os três tipos de discrepância considerados neste sistema são: **discrepância estrutural** – como consequência de diferenças ao nível do formato dos campos (por exemplo: 31/05/04 e 2004/05/31); **discrepância de esquema** – resultantes de uma deficiente estratégia de integração de dados provenientes de múltiplas fontes; e **violação de restrições do domínio** – a sua detecção é melhorada quando se utilizam algoritmos específicos do domínio. Este tipo de discrepância pode assumir duas formas distintas: **envolvendo um único tuplo** – quando o valor de um campo num tuplo viola directamente restrições referentes ao seu domínio; e **envolvendo vários tuplos** – quando o valor dos campos em dois ou mais tuplos violam uma restrição ainda que, individualmente, cada tuplo esteja correcto (por exemplo: violação de uma dependência funcional).

### 3.2.5 FraQL

O FraQL [18, 19] define uma arquitectura para as tarefas envolvidas na preparação de dados (integração, transformação, limpeza e redução de dados), tendo por base uma linguagem declarativa que permite o acesso e manipulação de tuplos armazenados em múltiplas fontes (por exemplo: bases de dados, documentos estruturados no formato relacional). Tendo por base um modelo de dados objecto-relacional, a linguagem é uma extensão ao SQL, com características que permitem cobrir as necessidades particulares inerentes à preparação de dados. A implementação das extensões como primitivas de base de dados permite tirar partido das potencialidades intrínsecas dos sistemas de gestão de base de dados, produzindo um efeito sinergético para ambas. A principal vantagem da utilização de uma linguagem deste tipo, que combina mecanismos de preparação dos dados e potencialidades poderosas de interrogação a várias fontes heterogéneas, consiste numa integração virtual, na qual é possível executar operações de transformação e limpeza, sem afectar o conjunto de dados original. Desta forma, é possível ensaiar e avaliar diferentes estratégias de integração e limpeza, sem ser necessário proceder ao carregamento e à materialização explícita dos dados, o que resulta num esforço computacional reduzido.

Uma atenção especial é concedida à limpeza de dados, materializada na possibilidade de realização de qualquer uma das seguintes operações: **identificação e reconciliação de duplicados** – o processo de eliminação de duplicados decompõe-se em duas fases: Na primeira fase, são identificadas as entidades que, provavelmente, se referem ao mesmo objecto do mundo real. Na segunda fase, as entidades duplicadas são reconciliadas (fundidas); **preenchimento de valores em falta** – algumas estratégias para o preenchimento de valores em falta são apresentadas; **manipulação de ruído nos dados –** o ruído nos dados é motivado por erros aleatórios. O FraQL possibilita a utilização de métodos baseados em histograma para identificar este tipo de anomalia nos dados; e **detecção e remoção de desvios –** Na detecção deste tipo de anomalia é, também, possível utilizar métodos baseados em histograma.

## 4. Problemas de Qualidade dos Dados versus Limpeza de Dados

Na tabela a seguir apresentada, relaciona-se os diversos tipos de problemas de qualidade dos dados (secção 2) com o suporte concedido (ao nível da detecção e correcção) pelas várias aproximações de limpeza de dados existentes (secção 3). O objectivo consiste em fornecer um retrato o mais rigoroso e exaustivo possível. Contudo, a tabela é elaborada tendo por base a informação disponível sobre cada uma das abordagens na literatura analisada.

A tabela está organizada em duas grandes classes: automática e manual. A detecção e correcção automática de cada um dos tipos de problemas (coluna 1) foi subdividida em: dependente de função extra a incorporar (respectivamente, colunas 2 e 5) e intrínseca ao próprio sistema de limpeza de dados (respectivamente, colunas 3 e 6).

Por dependente de função extra, entende-se a detecção ou correcção automática que implica a incorporação de funções adicionais para além das que se encontram disponíveis para utilização. Por intrínseca, entende-se a detecção ou correcção

automática que apenas necessita de parâmetros fornecidos pelo utilizador ou, eventualmente, nem isso, recorrendo a funções que já existem de base/raiz.

Por correcção manual (coluna 4), entende-se as que envolvem uma intervenção individualizada (caso a caso) por parte do utilizador.

Além das designações das diferentes aproximações, foram também utilizados na tabela os acrónimos *lne* e *dd* cujos significados são, respectivamente, limpeza de nomes e endereços e detecção de duplicados.

| **Id. probl.** | **Detecção Automática** | | **Correcção manual** | **Correcção Automática** | |
|---|---|---|---|---|---|
| | **função extra a incorporar** | **intrínseca (de base/raiz)** | | **função extra a incorporar** | **intrínseca (de base/raiz)** |
| P211a | | ajax; arktos; intelliclean; fraql | arktos | ajax; fraql | intelliclean |
| p211b | fraql | arktos; potter's wheel | arktos; potter's wheel | fraql | |
| p211c | | | | | |
| p211d | ajax | potter's wheel; lne | | ajax; potter's wheel | lne |
| p211e | ajax | lne | | ajax | lne |
| p211f | ajax | lne; intelliclean | | ajax | intelliclean; lne |
| p211g | ajax | intelliclean | | ajax | intelliclean |
| p211h | ajax | arktos; potter's wheel; intelliclean | | ajax | arktos; potter's wheel; intelliclean |
| p212a | ajax | | | ajax | |
| p212b | | | | | |
| p212c | | | | | |
| p213a | | arktos | arktos | | |
| p213b | ajax; fraql | dd; intelliclean | | ajax; fraql | intelliclean; dd |
| p213c | ajax; fraql | dd; intelliclean | | ajax; fraql | intelliclean; dd |
| p22a | | arktos | arktos | | |
| p22b | | | | | |
| p22c | ajax; fraql | dd | | ajax; fraql | dd |
| p22d | ajax; fraql | dd | | ajax; fraql | dd |
| p22e | ajax | arktos | | ajax | arktos |
| p22f | | | | | |
| p22g | ajax | | | ajax | |

Uma das conclusões que se extrai do estudo realizado é a da inexistência de aproximações que permitam efectuar a detecção e correcção automática de certos

problemas de qualidade dos dados (por exemplo: p22b – referência existente mas errada ou p22f – utilização de unidades de medida diferentes).

Também se constata que existem aproximações que possuem potencialidades intrínsecas de detectar certos tipos de problemas, mas que não possuem capacidades similares de efectuar o respectivo tratamento (por exemplo: p22a – referência inexistente). Nestes casos, o tratamento depende da incorporação de uma função extra ou, eventualmente, envolve mesmo a manipulação manual. Noutros casos, também a detecção é efectuada com base numa função extra, o que significa que é necessário a sua definição explicita (programação) para que o problema possa ser detectado e tratado (por exemplo: p212a – violação da dependência entre atributos).

Uma outra conclusão interessante é a inexistência de uma abordagem que cubra de forma intrínseca todos os problemas relacionados com a qualidade dos dados, a nível de detecção e tratamento. A título de exemplo, considere-se o caso do *IntelliClean* que apenas cobre seis tipos de problemas, o que é manifestamente pouco quando comparado com o universo de problemas de qualidade que podem ocorrer nos dados. A abordagem que cobre um maior leque de problemas é o *AJAX*. No entanto, esta abordagem assenta na filosofia de adição de funções extra para a detecção e tratamento dos problemas de qualidade dos dados. Isto significa que é necessário um esforço considerável na implementação/programação das ditas funções.

## 5. Conclusão

Este artigo reflecte um trabalho que visou a identificação e sistematização dos problemas de qualidade que podem ser encontrados num ou em vários conjuntos de dados relacionados. Igualmente, identifica e descreve as aproximações de limpeza de dados que actualmente maior relevância possuem, entre protótipos de investigação e ferramentas comerciais.

Deste cruzamento é possível extrair-se um conjunto de conclusões que essencialmente evidenciam uma cobertura deficiente, em amplitude ou qualidade de manipulação, das actuais soluções perante o vasto leque de problemas identificados.

Na sequência destas conclusões, perspectivamos trabalho futuro ao nível da limpeza de dados que visa explorar as lacunas identificadas. Assim, pretendemos desenvolver uma *framework* de limpeza de dados que suporte de uma forma global e integrada todos os problemas de qualidade dos dados identificados neste trabalho.

Para cada um dos tipos de problemas vamos efectuar um estudo exaustivo das técnicas que são utilizadas para a sua manipulação, em cada uma das abordagens de limpeza de dados. Deste modo, podemos seleccionar as técnicas mais adequadas, recorrer a uma combinação delas, ou investigar e desenvolver novas técnicas de manipulação sempre que se observem oportunidades de criar valor acrescentado para a limpeza de dados.

Uma especial atenção será concedida aos problemas que não encontram suporte em qualquer uma das aproximações actualmente existentes.

A automatização será um princípio orientador do nosso trabalho. Procurar-se-á eliminar ou, pelo menos, reduzir ao máximo a intervenção humana a um mero fornecimento de argumentos a funções já pré-implementadas numa biblioteca.

O recurso a soluções que envolvem a detecção e correcção baseadas em funções extra incorporadas, torna-as dispendiosas ao nível do tempo e custo de desenvolvimento. A este facto acresce ainda que a implementação de tais funções, frequentemente em linguagens do tipo *scripting*, não está ao alcance de qualquer utilizador, mas apenas dos programadores. A procura de abordagens que de base ou raiz ofereçam soluções o mais genéricas possível, permitirá que estas operações possam ser efectuadas por utilizadores menos especializados. Na procura da automatização, métodos de aprendizagem automática poderão vir a ser incluídos na *framework* para auxiliar o utilizador.

A solução que preconizamos para atingir estes vários objectivos basear-se-á na definição de uma linguagem formal para descrever cada tipo de erro e seu tratamento.

Por último, um aspecto importante que não vamos descurar é o da concepção de uma interface gráfica que seja amigável e intuitiva na interacção com o utilizador.

## Referências


[1] Ballou, D. e Tayi, G. K. Enhancing Data Quality in Data Warehouse Environments. Communications of the ACM, 42(1): 73-78. 1999.

[2] Inmon, W.H. Data Warehouse Performance. New York: John Willey.

[3] Rahm, E. e Do, H. H.. Data Cleaning: Problems and Current Approaches. IEEE Bulletin of the Technical Committee on Data Engineering, 24(4). 2000.

[4] Christen, P.; Churches, T. e Zhu, J.X. Probabilistic name and address cleaning and standardization. The Australian Data Mining Workshop. 2002.

[5] Churches, T.; Christen, P.; Lu, J. e Zhu, J.X. Preparation of name and address data for record linkage using hidden Markov models. BioMed. Central Med. Inform. Decision Making 2(9). 2002.

[6] Hernandez, M.A. e Stolfo, S.J. The merge/purge problem for large databases. Proceedings of the ACM SIGMOD Conference. 1995.

[7] Sarawagi, S e Bhamidipaty, A. Interactive deduplication using active learning. Very Large Data Bases ’02. 2002.

[8] Winkler W.E., Methods for record linkage and Bayesian networks. Proceedings of the Section on Survey Research Methods, American Statistical Association, 2002.

[9] Ananthakrishna, R.; Chaudhuri, S. e Ganti, V. Eliminating fuzzy duplicates in data warehouses, Very Large Data Bases ’02. 2002.

[10] Liang, J.; Li, C. e Mehrotra, S. Efficient record linkage in large data sets. 8th Annual International Conference on Database Systems for Advanced Applications. Japão. 2003.

[11] Cohen, W.W. e Richman, J. Learning to match and cluster large high-dimensional data sets for data integration, ACM SIGKDD ’02. 2002.

[12] Galhardas, H.; Florescu, D.; Shasha, D. e Simon E. AJAX: An Extensible Data Cleaning Tool. In Proceedings of the ACM SIGMOD on Management of Data. Dallas, EUA. 2000.

[13] Galhardas, H.; Florescu, D.; Shasha, D.; Simon E. e Saita, C.A. Declarative Data Cleaning: Language, Model and Algorithms. In Proceeding of the 27th Very Large Databases Conference. Roma. Itália. 2001.

[14] Vassiliadis, P.; Vagena, Z.; Skiadopoulos, S.; Karayannidis, N. e Sellis, T. ARKTOS: Towards the Modeling, Design, Control and Execution of ETL Processes. Information Systems, 26: 537-561. 2001

[15] Lee, M. L.; Ling, T. W. e Low, W. L. IntelliClean: A Knowledge-Based Intelligent Data Cleaner. In Proceedings of the ACM SIGKDD. Boston, EUA. 2000.

[16] Low, W. L.; Lee, M. L. e Ling, T. W. A Knowledge-Based Approach for Duplicate Elimination in Data Cleaning. Information Systems, 26: 585-606. 2001.
[17] Raman, V. e Hellerstein, J. M. Potter's Wheel: An Interactive Framework for Data Transformation and Cleaning. In Proceeding of the 27th Very Large Databases Conference. Roma. Itália. 2001.
[18] Sattler, K. U.; Conrad, S. e Saake, G. Adding Conflict Resolution Features to a Query Language for Database Federations. In Proceedings of the 3rd International Workshop on Engineering Federated Information Systems. Dublin, Irlanda. 2000.
[19] Sattler, K. U. e Schallehn E. A Data Preparation Framework Based on a Multidatabase Language. International Database Engineering Applications Symposium. Grenoble, France. 2001.