Toni Jardini

Ambiente Data Cleaning: Suporte extensível, semântico e automático para análise e transformação de dados

São José do Rio Preto - SP, Brasil

30 de Novembro de 2012

Toni Jardini

Ambiente Data Cleaning: Suporte extensível, semântico e automático para análise e transformação de dados

Dissertação apresentada para obtenção do título de Mestre em Ciência da Computação, área de concentração em Sistemas de Computação junto ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Campus de São José do Rio Preto.

Orientador:
Prof. Dr. Carlos Roberto Valêncio

Programa de Pós-Graduação em Ciência da Computação
Departamento de Ciências de Computação e Estatística
Universidade Estadual Paulista "Júlio de Mesquita Filho"

São José do Rio Preto - SP, Brasil

30 de Novembro de 2012

Toni Jardini

Ambiente Data Cleaning: Suporte extensível, semântico e automático para análise e transformação de dados

Dissertação apresentada para obtenção do título de Mestre em Ciência da Computação, área de concentração em Sistemas de Computação junto ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Campus de São José do Rio Preto.

BANCA EXAMINADORA

Prof. Dr. Carlos Roberto Valêncio
UNESP - São José do Rio Preto
Orientador

Prof. Dr. Nalvo Franco de Almeida Junior
Universidade Federal do Matro Grosso do Sul

Prof. Dr. José Márcio Machado
UNESP - São José do Rio Preto

S. J. do Rio Preto, 30 de Novembro de 2012

Felicidade nada mais é do que a independência de pessoas, apegos e expectativas.

Dedico este trabalho ao querido professor, Augusto Augustinho Neto, por sua amizade, sabedoria e carinho; foi um privilégio tê-lo encontrado nessa vida.

Agradecimentos

Agradeço aos meus avós e à minha mãe que foram essenciais em minha formação pessoal e profissional e incentivadores ferozes dos meus estudos acadêmicos.

Agradeço à minha irmã, ao meu companheiro de todas as horas, Marcelo Imperatriz, ao Paker e a todos meus amigos pessoais e do GBD - Grupo de Banco de Dados, Paulo, Ichiba, Bróxa, Matheus, Diogo, Camila, Thati, Clér, Andrielson e a todos os demais que estiveram presentes em minha vida acadêmica, profissional e pessoal durante os últimos anos.

Agradeço ao prof. Borges pelas espetaculares disciplinas de Física e Computação Quântica ministradas durante o programa de pós-graduação; aulas dignas de premiação!

Agradeço ao prof. Valêncio, orientador e amigo, pelos momentos de sucesso que passamos juntos ao longo dos últimos 12 anos, desde meus primeiros anos da graduação até a conclusão do Mestrado.

Agradeço à vida por me proporcionar somente alegrias e felicidade em todos os âmbitos.

RESUMO

Um dos grandes desafios e dificuldades para se obter conhecimento de fontes de dados é garantir consistência e a não duplicidade das informações armazenadas. Diversas técnicas e algoritmos têm sido propostos para minimizar o custoso trabalho de permitir que os dados sejam analisados e corrigidos. Porém, ainda há outras vertentes essenciais para se obter sucesso no processo de limpeza de dados, e envolvem diversas áreas tecnológicas: desempenho computacional, semântica e autonomia do processo. Diante desse cenário, foi desenvolvido um ambiente *data cleaning* que contempla uma coleção de ferramentas de suporte à análise e transformação de dados de forma automática, extensível, com suporte semântico e aprendizado, independente de idioma. O objetivo deste trabalho é propor um ambiente cujas contribuições cobrem problemas ainda pouco explorados pela comunidade científica na área de limpeza de dados como semântica e autonomia na execução da limpeza e possui, dentre seus objetivos, diminuir a interação do usuário no processo de análise e correção de inconsistências e duplicidades. Dentre as contribuições do ambiente desenvolvido, a eficácia se mostras significativa, cobrindo aproximadamente 90% do total de inconsistências presentes na base de dados, com percentual de casos de falsos-positivos 0% sem necessidade da interação do usuário.

Palavras-chave: *data cleaning*. limpeza de dados. KDD. ambiente de limpeza de dados.

ABSTRACT

One of the great challenges and difficulties to obtain knowledge from data sources is to ensure consistency and non-duplication of stored data. Many techniques and algorithms have been proposed to minimize the hard work to allow data to be analyzed and corrected. However, there are still other essential aspects for the data cleaning process success which involve many technological areas: performance, semantic and process autonomy. Against this backdrop, an data cleaning environment has been developed which includes a collection of tools for automatic data analysis and processing, extensible, with multi-language semantic and learning support. The objective of this work is to propose an environment whose contributions cover problems yet explored by data cleaning scientific community as semantic and autonomy in data cleaning process and it has, among its objectives, to reduce user interaction in the process of analyzing and correcting data inconsistencies and duplications. Among the contributions of the developed environment, efficiency is significant exhibitions, covering approximately 90% of database inconsistencies, with the 0% of false positives cases without the user interaction need.

Keywords: data cleaning. KDD. data cleaning environment.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Siglas

SGBD Sistema Gerenciador de Banco de Dados

SQL *Structured Query Language*

DDD Discagem Direta à Distância

CEP Código de Endereçamento Postal

AHP *Analytic Hierarchy Process*

tf.idf *Term Frequency - Inverse Document Frequency*

NYSSIS *New York State Identification and Intelligence System*

ONCA *Oxford Name Compression Algorithm*

PSI *Phonetic Similarity Identification*

ETL Extração, Transformação e Carga

RCDB *Rules Configuration Database*

PL/SQL *Programming Language / Structured Query Language*

DEC *Detect Explore Clean*

DCF Dependências Condicionais-Funcionais

UF Unidade da Federação

SIVAT Sistema de Vigilância de Acidentes de Trabalho

CEREST Centro Regional de Referência em Saúde do Trabalhador

1 Introdução

1.1 Considerações Iniciais

O armazenamento de informações tem se tornado cada vez maior e mais frequente, uma vez que esses dados podem conter informações valiosas. Além do significado evidente que um dado representa, é possível obter, pela correlação entre eles, um conjunto de novas informações e conhecimentos, até então, implícitos e desconhecidos (VALENCIO, 2010).

É essencial para o avanço da ciência moderna e da tecnologia a busca por novas técnicas de manipulação, extração, armazenamento, recuperação e correlação de informações uma vez que novos conhecimentos são obtidos de conhecimentos prévios.

Há vários tipos de dados armazenados, de diferentes áreas de conhecimento, de diferentes formas e formatos, em diferentes fontes e base de dados. Sua quantidade está muito além da capacidade humana de se conseguir analisá-los e interpretá-los e somente com o uso da prospecção de dados computacional é possível e, com os resultados alcançados de forma automatizada, obter conhecimento.

Porém, para que a prospecção seja realizada de forma a obter informações verossímeis de todas as fontes possíveis ou convenientes, é necessário garantir que os dados analisados sejam integrados de forma consistente e correspondam à realidade da informação. Essa garantia só é possível se houver um pré-processamento dos dados; etapa em que as informações armazenadas em diferentes fontes são integradas e posteriormente analisadas, a fim de se detectar inconsistências, duplicidades física ou semântica, normalizações e correções, etc.

Uma das etapas de pré-processamento dos dados, foco desse trabalho, é denominada Limpeza de Dados (mais conhecida pelo termo em inglês *Data Cleaning*) cujo objetivo desta área de pesquisa é desenvolver técnicas computacionais para tratamento de erros, inconsistências, duplicidades e diferenças entre as informações de uma mesma fonte ou de até mesmo fontes distribuídas. Somente após esse processo é que os dados estarão adequados para serem analisados e correlacionados (ANDRADE et al., 2011).

Apesar de haver na literatura diversas sugestões e técnicas para tentar resolver o problema de inconsistência e duplicação de dados, ainda não é possível encontrar ferramentas suficientes dos pontos de vista de eficiência e eficácia, uma vez que o custo do processamento de sua aplicação é bastante alto e os resultados obtidos nem sempre garantem que as informações estejam totalmente tratadas e corrigidas devidamente (ELMAGARMID; IPEIROTIS; VERYLIOS, 2007).

Diante desse cenário, vê-se importante investir no desenvolvimento de novas tecnologias que consigam tratar os problemas ainda pouco explorados, mas importantes para se conseguir consistência das informações armazenadas. Essas tecnologias precisam envolver semântica, melhoria de desempenho computacional e automatização do processo de análise e limpeza dos dados.

1.2 Motivação e objetivos

Apesar de haver várias contribuições e trabalhos desenvolvidos para tentar resolver o problema de detecção de inconsistências ou duplicidade de informações, ainda não há soluções que consigam abordar todos os problemas ou resolvê-los de forma satisfatória.

O processo de detecção e remoção de informações duplicadas por exemplo, é extremamente caro e exige um alto poder computacional; o tempo para que o processo seja realizado é diretamente proporcional ao tamanho da(s) base(s) de dados a serem analisadas e, isso vem aumentando exponencialmente a cada dia. Além disso, é necessária a interação humana na decisão de escolher quais dados são corretos ou treinar as ferramentas

com abordagem inteligente. Há também o grande problema semântico entre os dados que pouco foi explorado ainda e bastante comum nas fontes de informações, pois semântica depende de cultura, conhecimento, idioma, região, etc (ANDRADE et al., 2011).

Diante desse cenário, desenvolveu-se um ambiente com uma coleção de ferramentas de limpeza de dados estruturado como um *framework* devido à sua característica modular e principalmente sua extensibilidade. Também foi objetivo a busca por soluções que automatizem o processo de limpeza de base de dados, com base em políticas e regras configuráveis definidas pelo usuário. As técnicas e algoritmos que analisam e atacam os problemas de inconsistência e duplicidade dos dados também precisam contemplar os aspectos físicos e semânticos das informações, além de prever abordagens dinâmicas e inteligentes para que o ambiente seja treinado e, a cada iteração, conseguir automaticamente detectar e eliminar os problemas de duplicatas de forma mais eficaz possível.

1.3 Organização do Trabalho

O trabalho está organizado em cinco capítulos descritos a seguir:

- **Capítulo 2:**

  São apresentados os principais conceitos da área de limpeza de dados e discutido o estado da arte atual na área. As técnicas, algoritmos e ferramentas existentes, disponíveis no mercado e meio acadêmico são descritos brevemente. Também é realizado um levantamento da aplicabilidade dos ferramentais atuais e os desafios e problemas que ainda não foram solucionados ou precisam ser melhorados para se obter eficácia adequada.

- **Capítulo 3:**

  É apresentada a arquitetura proposta em detalhes, os módulos que compõem o ambiente *Data Cleaning* implementado e todas suas funcionalidades que suportam o processo de limpeza de dados manual, semi-automatizado ou automatizado, con-

templando técnicas que envolvem sinônimos multi-idiomas e abordagem inteligente com treinamentos iterativos.

- **Capítulo 4:**

  São apresentados os testes e experimentos realizados a fim de comprovar a eficácia dos recursos e funcionalidades do ambiente desenvolvido. Além dos testes que demonstram a aplicabilidade de todas as funcionalidades presentes nos módulos de análise e transformação, são discutidos diversos experimentos comparativos que apresentam a análise e diferenças das abordagens manual e automatizada na execução do processo de limpeza de dados.

- **Capítulo 5:**

  São apresentadas as conclusões do trabalho realizado bem como propostas para trabalhos futuros e a relação completa das referencias bibliográficas dos trabalhos utilizados para desenvolvimento desta dissertação.

2 Conceitos Fundamentais

2.1 Considerações iniciais

São apresentados neste capítulo os conceitos fundamentais para o entendimento do processo de limpeza de dados incluindo as principais definições, técnicas existentes e os problemas e desafios encontrados na literatura científica.

2.2 Limpeza de Dados

O processo de limpeza de dados tem por objetivo detectar e remover erros e inconsistências de uma ou mais fontes de informações a fim de melhorar a qualidade dos dados contidos e acessados. Este processo é necessário para se analisar e extrair conhecimento útil de um repositório, de um arquivo, de um conjunto de dados, de bases computacionais, etc., além de essencial para a integração consistente de fontes heterogêneas de dados e também uma das etapas de pré-processamento de dados para a prospecção de informações (*data mining*).

Os problemas e inconsistências geralmente encontrados surgem devido principalmente a erros de ortografia, dados inválidos ou falta de dados durante a entrada de informações em um arquivo ou repositório de dados. Na integração de fontes distintas, por exemplo, em data warehouses, sistemas de banco de dados federados ou sistema de informações em nuvens, as diferenças semânticas, isto é, dados redundantes em diferentes representações, contribuem para aumentar ainda mais a necessidade da consolidação das informações através da eliminação de informações duplicadas, padronização e da normalização das

diferentes representações dos dados (RAHM; DO, 2000).

Sistemas de fontes heterogêneas de dados como *data warehouses* precisam de um amplo tratamento das informações e, principalmente, oferecer suporte à limpeza dos dados. Nesse tipo de arquitetura os dados são carregados continuamente e enormes quantidades de informações são atualizadas a partir de uma variedade de repositórios cuja probabilidade de que alguma dessas fontes contenha dados sujos é alta. As informações contidas são utilizadas para tomadas de decisão, de modo que a correção de seus dados é essencial para evitar conclusões errôneas. Por exemplo, duplicação ou falta de informações produziriam estatísticas incorretas ou enganosas.

Devido à grande variedade de possíveis inconsistências e o grande volume de dados, a limpeza de dados é considerada uma das etapas mais complexas no processo de obtenção de conhecimento. Ainda nos dias de hoje, os sistemas computacionais oferecem apenas suporte limitado à limpeza de dados pois se concentram em transformações de dados para a tradução e integração de esquemas (ELMAGARMID; IPEIROTIS; VERYLIOS, 2007).

O processo de limpeza de dados deve satisfazer várias exigências. Primeiramente, deve detectar e remover todos os erros e inconsistências das informações armazenadas, tanto em fontes de dados únicas como em múltiplas fontes integradas. A abordagem deve ser apoiada por ferramentas que minimizem ao máximo o esforço manual e de programação e sejam genéricas e extensíveis para que facilmente cubram novas fontes de informações.

Além disso, a limpeza de dados não deve ser realizada isoladamente, mas juntamente com transformações relacionadas ao esquema de dados baseado em meta-dados abrangentes. Funções de mapeamento para a limpeza e outras transformações de dados devem ser especificadas de forma declarativa e serem reutilizáveis para outras fontes de dados, bem como para o processamento de consultas. Uma infraestrutura de fluxo de trabalho deve suportar todas as etapas de transformação de dados de múltiplas fontes e grandes conjuntos de dados de forma confiável e eficiente.

2.2.1 Principais Problemas de Erros e Inconsistência de Dados

Nesta seção e nas subsequentes serão discutidos os principais problemas de qualidade de informações a serem tratados pelo processo de limpeza de dados. Como mostrado na figura 2.1, os problemas são organizados basicamente em dois segmentos: problemas de dados de fonte única e problemas de dados de múltiplas fontes (RAHM; DO, 2000).

Figura 2.1: Problemas de Qualidade dos Dados. Adaptado de (RAHM; DO; 2000)

Para cada segmento, os tipos de problemas de dados são divididos em problemas de esquema e de instância, apesar de que um mesmo problema pode vir a ser resolvido e tratado via esquema ou via instância. Nos sub-tópicos seguintes serão abordados com maiores detalhes cada um dos tipos de problemas de dados e abordagens de como tratá-los.

2.2.2 Problemas de Fonte Única

A qualidade dos dados de uma fonte depende grande parte do grau de restrições de integridade e controle dos possíveis valores de entrada. Para as fontes sem esquema, como por exemplo arquivos, há poucas restrições sobre os dados que podem ser digitados e armazenados, acarretando em alta probabilidade de erros e inconsistências.

Já os Sistemas Gerenciadores de Banco de Dados (SGBD) contemplam alguns tratamentos e restrições de dados específicos de seus modelos via esquema (por exemplo,

a abordagem relacional requer valores de atributos simples, integridade referencial, etc.) além de restrições específicas de integridade controladas pela aplicação. Neste caso, os problemas que se referem à qualidade dos dados ocorrem devido à falta de restrições adequadas de integridade específicas do esquema do banco de dados, por exemplo, limitações ou um projeto de banco de dados mal elaborado, ou até mesmo porque apenas parte das possíveis restrições de integridade são definidas para limitar a sobrecarga do controle de integridade.

Mesmo havendo controles oferecidos pelos SGBD's, há diversos problemas específicos de instância, como erros e inconsistências que não podem ser tratados ou impedidos via esquema (por exemplo, erros de ortografia). Os escopos dos problemas de esquema e instância podem se diferenciar pelo atributo, registro, tipo de registro e fonte.

Uma vez que a limpeza de dados é um processo caro, impedir que dados sujos sejam inseridos é claramente uma ação importante para se reduzir o custo desta etapa. Isso requer um planejamento adequado para o esquema de banco de dados e restrições de integridade, bem como de validações de entrada de dados via aplicação. Além disso, a descoberta de regras de limpeza de dados pode sugerir melhorias para as limitações impostas por esquemas existentes, como unicidade e restrições de atributos (RAHM; DO, 2000).

2.2.3 Problemas de Múltiplas Fontes

Os problemas presentes em fontes únicas são agravados quando há necessidade de que fontes múltiplas de dados sejam integradas. Cada fonte pode conter informações inconsistentes e os dados em cada uma das fontes podem ser representados de formas diferentes, sobrepondo-se ou contradizendo-se. Isso ocorre porque as bases de dados fontes são normalmente desenvolvidas, implantadas e mantidas de forma independente para atender necessidades específicas, o que resulta em um elevado grau de heterogeneidade de sistemas de gerenciamento de dados, modelos de dados, desenhos de esquema e dados

reais.

Os dados e as diferenças de modelagem do modelo esquema devem ser abordados pelas etapas de tradução do esquema e do esquema de integração, respectivamente. Os principais problemas de modelagem do esquema são de nomeação e conflitos estruturais. Conflitos de nomes surgem quando o mesmo nome é usado para objetos diferentes (homônimos) ou nomes diferentes são usados para o mesmo objeto (sinônimos). Conflitos estruturais ocorrem em muitas variações e referem-se a diferentes representações do mesmo objeto em diferentes fontes, por exemplo, a representação atributo e tabela, diferentes estrutura de componentes, tipos de dados diferentes, restrições de integridade diferentes, etc.

Além dos conflitos de esquema, muitos conflitos aparecem somente na instância, ou melhor, nos dados. Todos os problemas pertinentes ao caso de uma fonte única podem ocorrer com diferentes representações em diferentes fontes, por exemplo, registros duplicados, contradição entre os registros, etc. Mesmo quando não são os mesmos atributos e mesmos tipos de dados, pode haver um mesmo dado com representações e valores diferentes, por exemplo, por estado civil, ou interpretação diferente de valores, por exemplo, unidades de medição dólar e real, entre as fontes. Além disso, informações nas fontes podem ser fornecidas em diferentes níveis de agregação, por exemplo, vendas por produto e vendas por grupo de produto ou se referirem a diferentes pontos no tempo, por exemplo, as vendas atuais de ontem para a fonte A e vendas da semana passada para a fonte B.

O principal problema para a limpeza de dados de fontes múltiplas é identificar dados sobrepostos, em particular, registros correspondentes que se referem à mesma entidade do mundo real. Este problema envolve aspectos da identidade de um objeto, eliminação, duplicação ou mescla. Frequentemente a informação é parcialmente redundante e as fontes podem se complementar umas as outras, fornecendo informações adicionais sobre uma entidade. Dessa forma, as informações duplicadas devem ser filtradas e as informações complementares devem ser consolidadas e juntadas a fim de alcançar uma visão consistente

das entidades do mundo real.

2.3 O Processo de Limpeza de Dados

Em geral, segundo o autor Rahm (RAHM; DO, 2000), o processo de limpeza de dados pode envolver as seguintes fases:

- **Análise dos dados** - A fim de detectar quais tipos de erros e inconsistências devem ser tratados, é necessária uma análise detalhada dos dados. Além de uma inspeção manual dos dados ou amostras de dados, programas de análise devem ser utilizados para se obter meta-dados sobre as propriedades de dados e detectar problemas de qualidade de dados;

- **Definição de fluxo de trabalho de transformação e regras de mapeamento** - Dependendo do número de fontes de dados, o grau de heterogeneidade e de dados sujos e inconsistentes exige um grande número de transformação de dados durante as etapas de limpeza a serem executadas. A tradução de esquemas distintos pode então ser utilizada para mapear fontes para um modelo de dados comuns. Os primeiros passos da limpeza de dados podem corrigir problemas de fonte de única instância. Etapas posteriores lidam com esquema e correção de problemas de fontes múltiplas como, por exemplo, duplicatas. Durante as etapas que realizam a limpeza, pode ser exigida a interação do usuário nos casos em que houver decisões semânticas;

- **Verificação** - A eficácia do fluxo e das definições de transformação se deve à realização de testes e avaliações de uma amostra ou cópia dos dados de origem para melhorar as definições, se necessário. Múltiplas iterações dos passos de projeto, análise e verificação podem ser necessárias, por exemplo, uma vez que alguns erros só se tornam aparentes após a aplicação de algumas transformações;

- **Transformação** - Execução de um conjunto de modificações das informações analisadas a fim de que seja efetuada a limpeza dos dados;

- **Refluxo de dados limpo** - Uma vez que os erros são removidos, os dados sujos nas fontes originais devem ser substituídos pelos dados limpos automaticamente, além de manter as regras aplicadas à limpeza a fim de se evitar retrabalho de limpeza de dados futuras. Já em ambientes que envolvem múltiplas fontes, deve-se também prever uma grande quantidade de meta-dados, como esquemas, características das instâncias dos dados, mapeamentos de transformação, definições de *workflow*, etc. Por coerência, flexibilidade e facilidade de reutilização, os meta-dados devem ser mantidos em um repositório baseado em SGBD's.

Nos tópicos seguintes serão discutidas com mais detalhes as abordagens possíveis para análise de dados, a definição de transformação e resolução de conflitos. Conflitos de nome são normalmente resolvidos pela renomeação e conflitos estruturais requerem uma reestruturação e fusão parcial dos esquemas de entrada.

2.3.1 Análise de Dados

Meta-dados refletidos em esquemas normalmente são insuficientes para avaliar a qualidade de dados de uma fonte, são utilizadas especialmente se apenas parte das restrições de integridade. Assim, é importante analisar as instâncias reais para se obter meta-dados reais sobre as características dos dados ou padrões de valor incomum por meio da reengenharia. Meta-dados podem contribuir para que problemas de qualidade de dados sejam encontrados e também na identificação de correspondências entre os atributos de esquemas de origem, ou seja, correspondência de esquema (RAHM; DO, 2000).

Há duas abordagens afins para a análise de dados: criação de perfis de dados e prospecção de dados. A abordagem do tipo perfis de dados concentra-se na análise de atributos individuais de uma instância e deriva informações como o tipo de dados, comprimento, faixa de valores, valores discretos e sua frequência, variância e singularidade; a ocorrência de valores nulos, máscara de dados, por exemplo, para números de telefone que possuim estilo próprio de representação do tipo (99) 9999-9999, etc., proporcionando uma visão

exata de várias qualidades dos aspectos do atributo.

A abordagem do tipo prospecção de dados é focada na descoberta de padrões específicos de informações em grandes conjuntos de dados como, por exemplo, as correlações entre vários atributos. Esse modelo inclui técnicas de agrupamento, sumarização, descoberta de associação e de descoberta de sequência. A integridade entre os atributos, tais como dependências funcionais ou regras específicas de uma aplicação, regras de negócio, podem ser derivadas e utilizadas para completar valores em falta, corrigir valores ilegais e identificar registros duplicados através das fontes de dados. Por exemplo, uma regra de associação com a confiança elevada pode se tornar uma dica para problemas de qualidade de dados que violam regras de instâncias. Por exemplo, uma confiança de 99% para a regra *total = quantidade * preço unitário* indica que 1% dos registros não são consistentes e podem exigir uma análise mais detalhada.

2.3.2 Transformação de Dados

O processo de transformação de dados é composto por várias etapas em que são executadas, em cada passo, alterações relacionadas a esquema e instância. Os sistemas de limpeza de dados devem oferecer ao usuário um processo de transformação a partir da geração de código de transformação o mais automatizado possível e, assim, reduzir a quantidade de programação. Para isso, deve-se especificar as transformações necessárias em uma linguagem apropriada, apoiada por uma interface gráfica. Uma abordagem mais geral e flexível é o uso do padrão da linguagem SQL para realizar as transformações de dados e utilizar a possibilidade de extensões de aplicativos específicos de linguagem, em particular, funções definidas pelo usuário. Além disso, sua execução pelo SGBD pode reduzir custos de acesso de dados e, assim, melhorar o desempenho (RAHM; DO, 2000).

Obviamente que implementar funcionalidades para tratamento de problemas de dados acarreta num esforço de implementação significativo e nem todas as transformações de esquema são suportadas. Em particular, funções simples e muitas vezes necessárias, como

divisão ou aglutinação de dois ou mais atributos, nem sempre são suportados genericamente e precisam muitas vezes ser reimplementadas para cobrir variações específicas em cada caso.

2.3.3 Tratamento de Conflitos

Um conjunto de etapas de transformação precisa ser especificado e executado para resolver problemas de qualidade de dados tanto em esquemas quanto em vários níveis de instância. Vários tipos de transformações devem ser executadas em fontes de dados individuais, a fim de lidar com uma única fonte de problemas e, somente depois integrar-se com outras fontes quando necessário. Além de uma possível tradução do esquema, essas etapas preparatórias tipicamente incluem:

- **Extração de valores livres da forma do atributo** - muitas vezes é necessário que valores individuais sejam extraídos de um atributo para obter uma representação mais precisa para posteriormente ser submetido às etapas de limpeza, como verificação de dados correspondentes e eliminação de duplicatas. As transformações exigidas nesta etapa incluem a reordenação de valores dentro de um campo para lidar com transposições de palavras e extração de valor para dividir um atributo em dois ou mais;

- **Validação e correção** - Nesta etapa é examinada cada instância e se tenta corrigi-la automaticamente quando possível. Por exemplo, a verificação ortográfica com base na consulta do dicionário pode ser útil para identificar e corrigir erros ortográficos. Dicionários de endereços e códigos postais, base de dados dos Correios por exemplo, podem ajudar na correção de dados de endereço. A relação entre os atributos e seus derivados como data de nascimento e idade, preço total e preço unitário por quantidade, cidade e código DDD de um número de telefone, pode permitir que os problemas com inconsistência ou falta de dados sejam corrigidos facilmente se aplicadas as funções de correlação entre elas;

- **Padronização** - Para facilitar a comparação ou integração de instâncias, os valores dos atributos devem ser convertidos para um formato consistente e uniforme. Por exemplo, entradas de data e hora devem ser trazidas para um formato específico; nomes e dados textuais devem ser convertidos para letras maiúsculas ou minúsculas, condensados e unificados através do processo denominado *stemming*, que remove prefixos ou sufixos das palavras; também podem ser aplicadas técnicas de remoção de palavras negativas, os stopwords. Além disso, abreviaturas e esquemas de codificação podem ser tratados através de consultas a dicionários de sinônimos especiais ou de aplicação de regras de conversão pré-definidas.

Lidar com múltiplas fontes exige uma reestruturação e reorganização de esquemas, como divisão ou fusão de atributos e tabelas, para se conseguir uma forma de integrá-las. No âmbito instância, as representações conflitantes e a sobreposição de dados, isto é, diferentes representações para uma mesma entidade do mundo real, devem ser tratadas. A tarefa de eliminação de dados duplicados é normalmente realizada após as outras etapas de transformação ou de limpeza, especialmente depois do processo de limpeza ter sido aplicado nas fontes únicas para tratamento dos erros e representações conflitantes. A eliminação de dados duplicados exige, primeiramente, identificar registros semelhantes em relação à mesma entidade do mundo real. Em uma segunda etapa, os registros similares são fundidos em um registro que contém todos os atributos relevantes sem redundância; em seguida os registros redundantes devem então ser removidos.

No caso mais simples, não é um atributo de identificação ou a combinação de atributos por registro que pode ser usado para registros de correspondência, por exemplo, se diferentes fontes compartilham a mesma chave primária ou se há outros atributos comuns únicos. Atributos diferentes em uma regra de correspondência podem contribuir com pesos diferentes para o grau total de similaridade. Para componentes tipo *string*, atributos como nome do cliente ou nome da empresa de correspondência exata, podem ser aplicadas diversas abordagens difusas.

Serão discutidos com mais detalhes as técnicas e algoritmos apresentados na literatura atual para detecção de similaridade ou duplicação de dados.

2.4 Algoritmos e Técnicas de Detecção de Similaridade ou Duplicação de Dados

Determinar casos de correspondência é tipicamente uma operação muito custosa para grandes conjuntos de dados. Há diversas técnicas e algoritmos que contribuem para a detecção de dados duplicados. Segundo Elmagarmid (ELMAGARMID; IPEIROTIS; VERYLIOS, 2007), a heterogeneidade dos dados pode ser classificada em dois principais tipos: estrutural e léxica.

A heterogeneidade estrutural ocorre quando os campos das tuplas de um banco de dados são estruturados de forma distinta em diferentes bancos de dados. Por exemplo, em uma base de dados, um endereço pode ser gravado em um campo chamado *Endereço*, enquanto, em outro banco, a mesma informação pode ser armazenada em vários campos, tais como rua, cidade, estado e CEP.

A heterogeneidade léxica ocorre quando as tuplas têm campos identicamente estruturados nos bancos de dados, mas os dados utilizam diferentes representações para se referirem à mesma entidade do mundo real mesmo, por exemplo, *R. Dom Pedro 1* e *Rua D. Pedro I.*

A seguir serão discutidos os principais algoritmos que abordam o problema organizados em três grupos: (1) baseados em similaridade de caracteres; (2) baseados em similaridade de *token* e (3) baseados em similaridade fonética (ELMAGARMID; IPEIROTIS; VERYLIOS, 2007).

2.4.1 Algoritmos e Técnicas de detecção de Similaridade Baseados em Caracteres

Os algoritmos para detecção de similaridade baseados em caracteres têm como objetivo tratar problemas principalmente de erros de ortografia ou de digitação dos dados. A seguir são apresentados os principais algoritmos que abordam esses casos encontrados na literatura.

Edit Distance

O algoritmo *edit distance* definido no trabalho de Zhan (ZHAN et. al., 2008) é o termo computacional utilizado para a técnica cuja implementação corresponde à distância de Levenshtein, que utiliza uma cadeia de caracteres para medir a quantidade de diferenças entre duas sequências. A distância Levenshtein entre duas palavras é definida como o número mínimo de edições necessárias para transformar uma palavra em outra, com as operações de inserção, remoção ou substituição de um único caractere.

Essa técnica é bastante popular dentre as ferramentas, pesquisas e trabalhos referentes à detecção de semelhança entre palavras. Os autores do trabalho (ZHAN et. al., 2008) propuseram uma ferramenta para detecção de plágio entre documentos - questões de preocupação crescente na comunidade acadêmica - para detecção de uma variedade de pequenas alterações que incluem inserção, deleção ou substituição de palavras. Tais mudanças simples, no entanto, requerem comparações de cadeias excessivas e o algoritmo *edit distance* mostrou-se interessante para essa varredura.

Smith Waterman Distance

O algoritmo denominado Smith Waterman Distance é uma melhoria da técnica do algoritmo *edit distance*. Basicamente a ideia proposta consiste em ignorar prefixos e sufixos a fim de que a distância entre as palavras seja menor e então, serem detectadas como semelhantes (ELMAGARMID; IPEIROTIS; VERYLIOS, 2007).

Affine Gap Distance

O algoritmo cujo nome é *Affice Gap Distance* (ELMAGARMID; IPEIROTIS; VERYLIOS, 2007) foi desenvolvido com o objetivo de melhorar a técnica utilizada pelo algoritmo *edit distance*, que apresenta resultados não satisfatórios para casos em que palavras correspondentes foram abreviadas ou truncadas, por exemplo, *Luis I. Lula da Silva* e *Luis Inácio Lula da Silva*. A fim de resolver esse problema, a técnica *Affine Gap Distance* introduz duas operações extras de edição para que esses casos sejam também detectados como semelhantes (ELMAGARMID; IPEIROTIS; VERYLIOS, 2007).

Jaro Distance Metric

Similar ao algoritmo *edit distance*, o algoritmo que recebe o nome de Jaro Distance calcula a semelhança geral entre duas palavras. No entanto, quando um subconjunto de caracteres não compartilha um prefixo comum com a da outra palavra, a distância é diminuída (INFORMATICA CORPORATIONS, 2008).

Q-Gram Distance

O algoritmo *Q-Gram Distance* apresentado no trabalho de Petrovic (PETROVIC; BAKKE, 2008) consiste na técnica que decompõe cada palavra em subcadeia de caracteres em que cada subcadeia corresponde ao conjunto de subcadeias da palavra decomposta e q é a quantidade de letras de cada conjunto formado. Por exemplo, a palavra MESTRADO com q=3 teria os seguintes q-grams: MES, EST, STR, TRA, RAD e ADO. Diversas técnicas têm sido desenvolvidas para comparar duas palavras com base em seus q-grams. Um exemplo simples seria contar o número de q-grams que duas palavras têm em comum, e uma quantidade alta de q-grams em comum significaria uma forte correspondência entre elas.

Tecnicamente, os algoritmos q-gram não são estritamente fonéticos, ou seja, não operam com base na comparação das características fonéticas das palavras. Em vez disso, as técnicas que envolvem q-grams são utilizadas para calcular a distância entre duas palavras. Como as palavras foneticamente semelhantes muitas vezes têm grafias semelhantes, esta técnica pode fornecer resultados favoráveis, principalmente na comparação de palavras

com erros ortográficos, mesmo que sejam foneticamente distintas.

Estudos recentes, como do autor Petrovic mostram que utilizar a técnica q-gram para detecção de semelhança entre palavras é muito mais eficaz do que outros algoritmos com propósitos semelhantes, como o *edit distance*. No experimento realizado, é constatado que o algoritmo detecta com maior precisão os casos de semelhança entre palavras.

Distância de Hamming

O algoritmo Distância de Hamming discutido no trabalho de Liu e demais autores (LIU; SHE; TORNG, 2011) corresponde ao número de posições nas quais dois conjuntos de mesmo tamanho diferem entre si. Vista de outra maneira, corresponde ao menor número de substituições necessárias para transformar um conjunto de caracteres em outro.

Os autores propuseram também uma melhora no algoritmo e criaram o algoritmo Distância de Hamming Dinâmica, denotado algoritmo HEngined, que apresentou a utilização de 5 vezes menos espaço para realização das consultas a serem processadas e melhoria de 16% no tempo de execução que o original.

Coeficiente de Jaccard

Também conhecido como o Coeficiente de Semelhança Jaccard, originalmente denominado de Coeficiente de Communauté por Paul Jaccard, o Coeficiente de Jaccard é um método estatístico utilizado para comparar a semelhança e diversidade de conjuntos de amostras e pode ser definido como o tamanho da intersecção dividido pelo tamanho da união dos conjuntos de uma amostra (JACCARD, 1901).

Os autores Wang e Ying-Hua (WANG; YING-HUA, 2009) propuseram um algoritmo baseado no Coeficiente de Jaccard em que pesos são definidos aos atributos do texto. A fórmula é proposta de tal maneira que cada atributo é multiplicado pelo peso, que é calculado por meio do processo AHP (Analytic Hierarchy Process). Com os atributos ponderados, é calculada a semelhança entre as entidades. Os resultados da experiência mostraram que a modificação proposta atinge maior precisão e eficácia na detecção de

semelhança entre conjuntos de palavras.

2.4.2 Algoritmos e Técnicas de detecção de Similaridade Baseados em *Token*

Os algoritmos para detecção de similaridade baseados em *token* têm como objetivo tratar problemas principalmente de erros de ortografia ou de digitação dos dados como os baseados em caracteres, porém com a diferença de tratar palavras compostas ou um conjunto de palavras. Casos frequentes como inversão de palavras como *Laranjão Supermercado* e *Supermercado Laranjão* são detectados com esses tipos de algoritmos, diferentemente dos anteriores que tratam palavras simples. Nas seções subsequentes são abordados os principais algoritmos que abordam esses casos encontrados na literatura.

Cosine Similarity

O método denominado *Cosine Similarity* discutido por Shiwei e demais autores (SHIWEI et al., 2010) mede a similaridade entre dois vetores baseado no valor do cosseno entre eles, contido no intervalo entre 0 e 1, em que o valor 1 indica que os vetores são exatamente idênticos e 0 indica que são completamente diferentes. Esse método é bastante utilizado para se medir a semelhança entre documentos, porém, para sua aplicação, é necessário especificar um limite de similaridade mínimo para execução do processo de análise de semelhança.

Shiwei e demais autores propuseram dois algoritmos baseados no *Cosine Similarity* que buscam automaticamente o melhor limite de similaridade por meio de uma estratégia de análise *diagonal-transversal*. Os resultados de ambos algoritmos apresentados, TOP-DATA e TOP-DATA-R, demonstraram maior eficiência e desempenho que o original.

Atomic Strings

Monge e Elkan (MONGE; ELKAN, 1996) propuseram um algoritmo para combinar campos de texto baseados em cadeias atômicas. Uma cadeia atômica é uma sequência de caracteres alfanuméricos delimitados por caracteres de pontuação. Duas cadeias atômicas

são correspondentes se forem iguais ou se uma for o prefixo da outra. Com base neste algoritmo, a semelhança de dois campos é o número de suas correspondentes cadeias atômicas divididos pelo seu número médio de cadeias atômicas.

WHIRL

Cohen (COHEN, 1998) descreveu um algoritmo denominado WHIRL que adota à recuperação de informação a similaridade do cosseno combinado com o esquema de ponderação tf.idf (Term Frequency - Inverse Document Frequency) para cálculo da similaridade de dois campos.

Q-Grams with tf.idf

Gravano (GRAVANO et al.; 2003) propõe uma melhora ao algoritmo WHIRL para incluir técnicas de tratamento de erros de entrada utilizando técnicas q-grams e, assim, utilizar *token*s ao invés de palavras. Essa técnica proposta apresenta bons resultados principalmente para casos em que há erros de grafia.

2.4.3 Algoritmos e Técnicas de Detecção de Similaridade Baseados em Fonética

As duas técnicas discutidas anteriormente tratam basicamente de semelhanças de representação dos caracteres que compõe uma ou um conjunto de palavras. Entretanto, dependendo de cada idioma, um conjunto de caracteres diferentes pode ser foneticamente semelhantes, mesmo que não sejam semelhantes fisicamente, isto é, semelhança de caracteres ou *token*s.

Soundex

Russell e O'Dell desenvolveram o algoritmo *Soundex* (HOLMES; MACCBE, 2002) que fornece uma capacidade de busca de termos inexatos a sistemas de recuperação de informação. A técnica do algoritmo consiste na comparação de textos de comprimento variável para códigos alfanuméricos de comprimento fixo. Essa técnica apresenta limitações pois o processo de conversão considera apenas letras isoladas, com exceção de consoantes

que se repetem, acarretando em perda de precisão no processo de detecção de semelhança entre palavras.

O trabalho dos autores Homes e Maccbe (HOLMES; MACCBE, 2002) propõe uma melhoria no algoritmo *soundex* integrando vários algoritmos fonéticos, apresentando uma melhora de 96% nos resultados detectados.

Metaphone e Double Metaphone

O algoritmo *metaphone* (MANDAL; HOSSAIN; NADIM, 2010) é uma adaptação do algoritmo *soundex*, otimizado para um idioma específico. O processo de conversão das letras considera regras fonéticas específicas de um idioma específico, contribuindo com uma melhora significativa na detecção de semelhança.

Posteriormente, foi proposto o algoritmo *double metaphone* que elimina vários tipos de ambiguidades que ocorriam no algoritmo *metaphone* original, melhorando sua eficácia e eficiência do processo de busca, além de tratar erros comuns de entrada de dados, como erros ortográficos.

Devido aos bons resultados que esse algoritmo apresenta, o trabalho de Mandal e demais autores (MANDAL; HOSSAIN; NADIM, 2010) utiliza o algoritmo *double metaphone* no desenvolvimento de uma ferramenta geradora de pesquisa eficiente que trata erros de escrita comuns em entradas manuais de dados. Os resultados dos experimentos realizados mostram que o algoritmo é eficiente aplicado também em grande bases de dados.

New York State Identification and Intelligence System (NYSSIS)

O sistema NYSIIS, proposto por Taft (TALF, 1970), é semelhante à técnica do algoritmo *soundex*, porém difere à medida em que mantém as informações das posições das vogais da palavra codificadas através da conversão da maioria das vogais para a letra A; e também não utiliza números para substituir letras e, ao invés disso, as consoantes são substituídas por outras letras foneticamente semelhantes, retornando assim puramente código alfa (sem caracteres numéricos). O sistema de codificação NYSIIS é ainda utilizado

pelos Serviços de Justiça Criminal de Estado de Nova Iorque.

Oxford Name Compression Algorithm (ONCA)

O algoritmo ONCA (GILL, 1997) utiliza uma técnica de dois estágios e foi desenvolvido com o intuito de superar a maioria das características insatisfatórias do algoritmo *soundex*. No primeiro passo, o algoritmo utiliza uma versão inglesa do método NYSIIS de compressão. No segundo passo, é aplicado o algoritmo *soundex* original na palavra anteriormente comprimida. Essa técnica tem demonstrado ser eficiente para detecção de palavras compostas grupos de palavras similares.

Phonetic Similarity Identification (PSI)

O algoritmo PSI aborda técnicas para identificação de registros duplicados independentemente do domínio do banco de dados e sua principal característica é abranger caracteres numéricos, além de caracteres alfabéticos. O algoritmo realiza a transcrição fonética das palavras conforme o idioma escolhido (disponível nos idiomas português e italiano) elevando significativamente o grau de possibilidades de resultados quando, posteriormente, aplicadas técnicas de detecção de semelhança entre palavras fisicamente diferentes ou contendo números (ANDRADE et al., 2011).

2.5 Técnicas de Detecção de Tuplas Duplicadas

Na seção anterior foram apresentados diversas técnicas e algoritmos propostos encontrados na literatura científica para tratamento e detecção de semelhança entre palavras. Contudo, as grandes bases de dados do mundo real contemplam dados complexos com múltiplos atributos e valores, tornando o processo de detecção de duplicatas uma tarefa bem mais complexa.

O trabalho de Elmagarmid e demais autores (ELMAGARMID; IPEIROTIS; VERYLIOS, 2007) realiza um levantamento das principais técnicas que consideram tuplas inteiras no processo de detecção de semelhança, organizadas em duas categorias:

1. Abordagens que são baseadas em treinamento de dados para aprendizado de como combinar e tratar os registros - esta categoria inclui abordagens probabilísticas e técnicas de aprendizado supervisionado de máquinas;

2. Abordagens que dependem de conhecimento de domínio ou em métricas de distância genéricos correspondentes aos registros - esta categoria inclui técnicas que utilizam linguagens declarativas para correspondência e abordagens que desenvolvem métricas de distância apropriadas utilizadas no processo de detecção de duplicadas.

Há propostas na literatura, sintetizadas no trabalho de Elmagarmid que contemplam abordagens probabilísticas para resolver o problema de detecção de duplicatas, que apresentam abordagens que utilizam técnicas de aprendizado supervisionado e variações baseadas em métodos de aprendizagem ativa, métodos baseados em distância, técnicas declarativas para a detecção de duplicatas e trabalhos que abordam técnicas de aprendizagem de máquina não supervisionada.

2.6 Ferramentas

Ao longo dos últimos anos, algumas ferramentas para limpeza de dados tem sido apresentadas no mercado e grupos de pesquisa têm contribuído apresentando técnicas e disponibilizando pacotes de *software*s livres que podem ser utilizados na detecção de registos duplicados.

WHIRL (WHIRL, 2012) é um sistema de detecção de registros duplicados disponível gratuitamente para uso acadêmico e de pesquisa. A ferramenta WHIRL utiliza a semelhança com base em métrica de *token* tf.idf para identificar sequências similares dentro de duas listas.

O Project5 Flamingo (FLAMINGO, 2012) é uma ferramenta semelhante à WHIRL, que fornece uma ferramenta simples de detecção de semelhança de *string*s, que recebe como entrada duas listas de *string*s e retorna os pares de *string*s que estão dentro de um

limiar pré-especificado de distância.

BigMatch (YANCEY, 2002) é um programa de detecção de duplicatas usado pelo Censo dos EUA o qual se baseia em estratégias para identificar possíveis correspondências entre os registros de duas relações para conjuntos de dados muito grandes. O único requisito é de que uma das duas relações devem caber na memória, isto é, máximo de 100 milhões de registos. O principal objetivo do BigMatch não é realizar a detecção de duplicatas de forma sofisticada, mas sim gerar um conjunto de pares de candidatos, que deverão numa segunda etapa ser processados por algoritmos mais sofisticados de detecção de tuplas duplicadas.

O Database Cleaner (VALENCIO et al.; 2010) possibilita a detecção de semelhança de dados utilizando alguns dos algoritmos escolhidos pelo usuário e a correção pela conversão da(s) tupla(s) duplicadas à tupla correta. Para melhorar o desempenho, a ferramenta também utiliza *multithreading* simultâneo para aproveitar ao máximo a arquitetura *multicore* dos computadores; as threads são executadas fisicamente em paralelo, alocadas em cada um dos núcleos do processador. A ferramenta possui também um painel SQL para execução de consultas diretamente no banco de dados e uma janela de visualização de referências.

WizSame (ELMAGARMID; IPEIROTIS; VERYLIOS, 2007), desenvolvido pela WizSoft, é um produto que permite a descoberta de registos duplicados em uma base de dados. Dois registros são detectados como semelhantes se contiverem uma fração significativa de palavras idênticas ou similares, cujo critério de semelhança corresponde às palavras dentro de uma distância pré-estabelecida.

O Febrl Syste3 (Freely Extensible Biomedical Record Linkage) (FEBRL, 2012) é um conjunto de ferramentas de limpeza de dados open-source e possui dois componentes principais: uma interface para padronização dos dados e uma para realização de detecção de dados duplicados. A padronização dos dados baseia-se principalmente em modelos ocultos de Markov (HMM) e, portanto, a ferramenta Febrl tipicamente requer treinamento

para analisar corretamente as entradas de banco de dados. Para a detecção de duplicatas, Febrl implementa uma variedade de métricas de similaridade de sequência de caracteres, como Jaro, *edit distance*, e *Q-Gram Distance*. A ferramenta também suporta codificação fonética (*soundex*, NYSIIS, e double metaphone) para detectar tuplas semelhantes. Uma vez que semelhança fonética é sensível a erros na de digitação, o Febrl também calcula semelhança fonética com a versão invertida da *string*, contornando o problema da primeira letra incorreta.

TAILOR (ELFEKY et al.; 2002) é uma caixa de ferramentas de padronização de registros que permite aos usuários aplicar diferentes métodos de detecção de duplicatas nos conjuntos de dados. A flexibilidade de usar vários modelos é útil quando os usuários não sabem qual o modelo de detecção de duplicatas é mais eficaz para ser utilizado. A ferramenta foi projetada em camadas, separando funções de comparação com a lógica de detecção de duplicatas. Além disso, as estratégias de execução que melhoram a eficiência são implementadas em uma camada separada, tornando o sistema mais extensível do que os sistemas que se baseiam em modelos monolíticos. A ferramenta também disponibiliza relatórios estatísticos, tais como precisão estimada e integralidade, o que pode ajudar os usuários a entender melhor a qualidade de uma execução de detecção dados duplicados ao longo de um novo conjunto de dados.

SmartClean (OLIVEIRA et al.; 2009) é uma ferramenta que detecta e corrige problemas de qualidade dos dados. Comparado com outras ferramentas existentes, o SmartClean oferece a vantagem de que o usuário não precisa especificar a sequência de execução das operações de limpeza de dados. Os problemas são tratados, isto é, detectados e corrigidos, seguindo esta sequência, que também suporta a execução incremental das operações.

É importante notar que, atualmente, os fornecedores de banco de dados diversos (Oracle, IBM e Microsoft) não oferecem ferramentas suficientes para a detecção de registros duplicados. A maioria dos esforços até agora tem se concentrado na criação de facilitar o uso de ferramentas ETL (Extração, Transformação e Carga) que podem padronizar os

registros do banco de dados e corrigir pequenos erros, principalmente no contexto dos dados de endereço. Outra função típica das ferramentas que são fornecidas hoje é a capacidade de usar tabelas de referência e padronizar a representação de entidades que são bem conhecidas por ter várias representações (ELMAGARMID; IPEIROTIS; VERYLIOS, 2007).

2.7 O Futuro das Pesquisas Relacionadas à Limpeza de Base de Dados

As pesquisas publicadas nos últimos anos na área de limpeza de dados têm coberto algumas frentes importantes como novos algoritmos, *frameworks* e ferramentas que contemplam uma combinação de diversas técnicas, envolvendo detecção de duplicadas e estatísticas, com objetivo de tentar conseguir melhorar a qualidade das informações com custo baixo e com o mínimo de interação do usuário no processo de limpeza de dados.

Nas seções subsequentes serão brevemente discutidos alguns trabalhos levantados na literatura atual da área de limpeza de dados.

2.7.1 *Frameworks* para Limpeza de Dados

Um *framework* pode ser denominado como um conjunto de conceitos com objetivo de oferecer e propor soluções a um problema de domínio específico. Essa abstração geralmente é composta de diversos projetos de *software* que oferece uma funcionalidade genérica e também define o fluxo de controle da aplicação.

Os autores Yan, Diao e Li (YAN; DIAO; LI; 2008) propõem um novo *framework* para limpeza de dados interativo e extensível com base na qualidade da informação. O *framework* contempla estratégias de análise de qualidade dos dados, estratégia de transformação de dados e estratégia de avaliação dos resultados provenientes do processo de limpeza. Também contempla o controle do processo de limpeza e oferece características significativas de extensibilidade e interatividade.

Já o *framework* proposto por Hung (HUNG et al.; 2009) visa detectar e eliminar os dados sujos e melhorar a qualidade das informações contidas na base de dados. Sua construção é baseada em modelo de usuário.

Yubin (YUBIN et al.; 2009) propõe algumas novas ideias e tecnologias específicas para limpeza de dados científicos, como a representação de domínio de conhecimento e uso, fluxo de limpeza personalizado e construção de forma dinâmica. O *framework* oferece modelagem de regras baseadas em conhecimento, modelagem baseada em fluxo de trabalho e algoritmos para aplicação da limpeza de dados. Uma de suas aplicações tem sido na limpeza de dados de oceanografia. Com esse framework, os autores esperam futuramente oferecer um sistema de limpeza de dados flexível e extensível.

Arasu e Kaushik (ARASU; KAUSHIK; 2009) propõem um *framework* formal que pode ser utilizado para manipular representações de dados. O *framework* utiliza linguagem declarativa e combina elementos de uma gramática gerativa com banco de dados de consulta. Também são contempladas funcionalidades de normalização e análise de dados para a preparação de dados para análise e pré-processamento dos dados para a execução da limpeza.

O trabalho de Huanzhuo (HUANZHUO et al.; 2010), denominado ODCF *(Open Data Cleaning Framework)*, consiste numa estrutura livre para limpeza de dados com escalabilidade, aplicada em diferentes áreas. O *framework* contempla a técnica denominada *auditoria contínua*, importante forma de auditoria assistida por computadores (CAATs), que é também uma área de investigação ativa na comunidade científica. Devido às rigorosas exigências de qualidade de dados para auditoria contínua, o *framework* contempla regras baseadas em semântica com funções de autoaprendizagem, visando melhorar a precisão e adaptabilidade do processo de limpeza de dados. As regras semânticas utilizadas são baseadas na hierarquia e dependência entre os campos.

No trabalho de Yanhua (YANHUA; SHUYU, 2010) é proposto um *framework* com modelagem de dados dinâmicos para o processo de limpeza. A modelagem melhora a

eficiência da limpeza e qualidade dos dados dinâmicos e contempla algumas tecnologias-chave de modelagem, como banco de dados dinâmico, regras dinâmicas e arquivos de classes dinâmicas de regras de limpeza; além de apresentar um método de adjunção de regra dinâmica, compilação e execução usando Java, devido às suas vantagens de robustez e portabilidade.

O modelo proposto no trabalho de Ali e Warraich (ALI; WARRAICH, 2010) implementa um *framework* de limpeza de dados robusto para garantir que dados limpos sejam despejados num data warehouse, baseado em RCDB *(Rules Configuration Database)*, que requer dois parâmetros de entrada de dados e de usuário. A versão inicial é implementada usando Oracle e linguagem PL/SQL e pode ser uma desvantagem, uma vez que não utiliza tecnologia gratuita. Berti-Equille e demais autores (BERTI-EQUILLE et. al., 2011) propõem um *framework* denominado DEC *(Detect Explore Clean)* para detecção e limpeza de dados complexos. O algoritmo desenvolvido é baseado em técnicas estatísticas para suportar as estratégias de seleção e limpeza dos dados, cobrindo diferentes tipos de sujeiras de informações e oferecendo estratégias mais eficazes que as estratégias tradicionais, além de suas características de efetividade e escalabilidade.

2.7.2 Novos Algoritmos e Técnicas para Limpeza de Dados

Ciszak (CISZAK; 2008) propõe um algoritmo baseado na metodologia de correlação de dados (*data mining*) para identificar e corrigir informações duplicadas físicas ou semânticas. O trabalho propõe dois algoritmos para detecção de sujeira e para limpeza da base de dados baseado em técnicas de prospecção.

Xinlin e demais autores (XINLIN et al.; 2009) introduzem a necessidade desenvolver um processo de limpeza específico para dados que contemplam informações provenientes de diversas fontes, geralmente por usuários da Internet, como o wikipedia (WIKIPEDIA, 2012). Uma vez que seu conteúdo não é controlado, pode ser grande o desafio de se propor soluções flexíveis e com alto desempenho para gerenciá-lo e mantê-lo consistente.

Wang (WANG; 2010) apresenta um algoritmo de limpeza de dados com uso de técnica de detecção de *outlier data*. Sua abordagem consiste em manter um histórico das limpezas efetuadas e, quando um dado inserido já foi limpo anteriormente, automaticamente é armazenada a informação correta. A ferramenta desenvolvida é mais voltada para o processo de integração de múltiplas fontes e, de forma automática, analisa e define o dado correto no processo de migração. A proposta é considerada versátil, mas ainda se encontra em fase de experimento.

Okita (OKITA; 2009) apresenta um algoritmo para limpeza de dados de tradutores automatizados. É uma abordagem muito específica e requer grande conhecimento em idiomas e estrutura de linguagem para ser implementado.

Bertossi e demais autores (BERTOSSI et al.; 2011) apresentam um exemplo do processo de limpeza de dados utilizando o conceito de dependências correspondentes como um procedimento de detecção de duplicatas. Essa nova abordagem contribui principalmente com a introdução de semântica às dependências correspondentes.

É proposto no trabalho de Chaturvedi (CHATURVEDI et al.; 2011) um método que seleciona um conjunto diversificado de registros de dados que, quando utilizados para criar a regra de dados baseados em modelo de limpeza, pode abranger o número máximo de registros. Esse método contempla uma métrica de similaridade entre dois registros que contribui para a escolha do conjunto diversificado de amostras de dados a serem limpos. Os resultados demonstram um aumento de 12% na eficiência do processo, comparando a ideia proposta com outro algoritmo.

No trabalho de Prasad (PRASAD et al.; 2011) é apresentada uma ferramenta de melhoria da qualidade de dados que identifica as variantes e sinônimos de uma determinada entidade presente nos dados, considerada uma tarefa importante para escrever regras de qualidade de dados para padronização das informações.

2.7.3 Novos Estudos Aplicados à Limpeza de Dados

É apresentada no trabalho Zhang (ZHANG et al.; 2010) um modelo de dados em 3 camadas baseado em sistema de limpeza multi-agente que contempla várias técnicas de limpeza de dados, além de ser um sistema inteligente que, ao ser treinado, reduz a participação de pessoas no processo.

No trabalho dos autores Eredics e Dobrowiecki (EREDICS; DOBROWIECKI; 2011) é apresentada a experiência de limpeza de uma base de dados de uma estufa e a importância para esse segmento da indústria. São analisados os problemas de falta de dados coletados em um sistema de estufa discutido como os problemas de falta de dados e inconsistências foram resolvidos. Os resultados demonstram que, após a correção das inconsistências, aumentou-se em 50% a quantidade de dados válidos a serem utilizados.

Bohannon (BOHANNON et al.; 2007) aborda um novo conceito para contribuir às pesquisas relacionadas à limpeza de dados: classe de restrições ou dependências condicionais-funcionais (DCF). Diferentemente das tradicionais dependências funcionais, que foram desenvolvidas principalmente para projetos de esquema, as DCFs visam capturar a consistência dos dados, incorporando ligações semanticamente relacionadas. Foi desenvolvido um sistema de inferência análoga a axiomas de Armstrong para dependências funcionais, bem como de análise de consistência. Uma vez que DCFs permitem estabelecer vínculos entre dados, um grande número de indivíduos pode manter restrições sobre uma tabela, evitando violações de restrição. Foram desenvolvidas técnicas para a detecção de violações de DCF em SQL, bem como novas técnicas para verificação de restrições múltiplas em uma única consulta. Essa nova abordagem é também um passo em direção a um método pratico baseado em restrições para melhorar a qualidade dos dados.

2.8 Considerações Finais

Este capítulo teve como objetivo apresentar um levantamento bibliográfico dos conceitos básicos e o estado da arte na área de limpeza de dados, também conhecida por *Data Cleaning*. Foram discutidas as principais técnicas utilizadas, algoritmos, ferramentas disponíveis e os trabalhos que têm sido propostos no meio acadêmico para contribuir para melhoria das técnicas de detecção de dados duplicados em grandes bases de dados.

O grande número de técnicas e algoritmos para tratar e comparar dados reflete a intensidade de erros ou transformações que podem ocorrer com os dados na vida real. Infelizmente ainda há poucos estudos que comparam a eficácia das diferentes técnicas de distância apresentados neste capítulo. Os poucos trabalhos focados em análise da eficácia das técnicas existentes demonstram que uma mesma técnica aplicada para alguns conjuntos de dados pode apresentar resultados ruins em outros.

O custo de se aplicar alguma das técnicas apresentadas é alto e têm sido propostas maneiras de minimizar esse custo, como reduzir o número de comparações entre os registros. Algumas técnicas abordadas por Elmagarmid e demais autores (ELMAGARMID et al.; 2007) incluem *blocking, sorted neighborhood, clustering and canopies* e *set joins*, e têm o objetivo de melhorar a eficiência da comparação entre os registros. Porém, as poucas propostas nessa área ainda estão em sua fase inicial e precisam ser melhor exploradas.

Além disso, todas as técnicas requerem uma interação muito grande com o usuário para decidir e analisar as detecções realizadas e, no que se refere a grandes bases de dados, é humanamente inviável a interação direta no processo de limpeza de dados que, do mundo real, facilmente contém milhões de registros. Essa interação deve-se principalmente ao poder de decisão semântico, pois as técnicas e trabalhos realizados até então focam especificamente em detecção de semelhança física entre os textos e entidades.

São resumidos na tabela 2.1 os principais algoritmos, técnicas e ferramentas apresentados neste capítulo:

Técnicas	Resumo
Edit distance *Smith Waterman Distance* *Affine Gap Distance* Jaro Discance Metric *Q-Gram Distance* Hamming Distance Coeficiente de Jaccard	Algoritmos e técnicas de detecção de similaridade baseados em caracteres
Cosine Similarity Atomic *string*s WHIRL *Q-Grams with tf.idf*	Algoritmos e técnicas de detecção de similaridade baseados em *tokens*
Soundex *Metaphone e Double Metaphone* NYSIIS ONCA PSI	Algoritmos e técnicas de detecção de similaridade baseados em fonética
WHIRL Project5 Flamingo BigMatch WizSAme Febrl Syste3 TAILOR SmartClean	Ferramentas para detecção de duplicatas disponíveis na literatura

Tabela 2.1: Algoritmos, Técnicas e Ferramentas para Detecção de Duplicatas

3 *Ambiente Data Cleaning*: Suporte Extensível, Semântico e Automático para Análise e Transformação de Dados

3.1 Considerações Iniciais

No capítulo anterior foram apresentados os conceitos fundamentais e o atual estado da arte referente à limpeza de dados. O processo de detecção e remoção de duplicadas mostra-se caro e exige um alto poder computacional; o tempo para que o processo seja realizado é diretamente proporcional ao tamanho da(s) base(s) de dados a serem analisadas e, isso vem aumentando significativamente a cada dia.

Além disso, é necessária uma forte atuação humana na decisão de escolher quais dados são corretos ou treinar as ferramentas com abordagem inteligente. Há também um espaço interessante para o tratamento semântico com o propósito de contribuir junto a este segmento que ainda é pouco explorado.

Com o propósito de contribuir junto a este segmento, foi desenvolvido neste trabalho um Ambiente *Data Cleaning* que automatiza o processo de limpeza de base de dados com políticas configuráveis suportadas por técnicas físico-semânticas e treinamento para detecção de duplicatas. Este ambiente contribui com o processo de limpeza de dados de forma a melhorar a qualidade das informações e diminuir o custo de sua execução.

3.2 Definição do Problema

O processo de limpeza de dados pode ser basicamente dividido em duas atividades principais: análise e transformação. Para ambos os casos, é necessário um grande esforço computacional para varrer toda a base de dados, encontrar duplicidade de informações e corrigi-las.

Na etapa de análise, é necessário que abordagens físicas e semânticas sejam contempladas, uma vez que em muitos casos, palavras com grafias semelhantes não correspondam a duplicações; ou no caso contrário, palavras totalmente diferentes fisicamente podem representar uma mesma entidade do mundo real, isto é, representaria duplicidade ou sujeira de informação na base de dados.

As técnicas apresentadas na literatura para detecção de duplicidade de informações são baseadas em suas semelhanças físicas (ortográfica) e fonética, mas pouco ainda foi explorado do ponto de vista semântico. O trabalho de se conseguir cobrir as variações semânticas que um mesmo termo possui é custoso, uma vez que a semântica varia dentro de um idioma não somente em sua construção formal, com sinônimos, mas os significados variam numa mesma língua em diferentes regiões, cultura, povoado e até mesmo segundo o contexto em que estão inseridas.

Um outro ponto importante a ser explorado corresponde à frequência com que as bases de dados precisam ser limpas constantemente. A maioria das bases de dados do mundo real possuem dados sendo inseridos constantemente, acarretando em sujeiras e duplicidades diariamente. Isso significa que, uma limpeza efetuada num primeiro momento precisará ser novamente realizada no dia seguinte, muitas vezes algumas horas posteriormente. Constata-se que uma parcela significativa dos problemas são recorrentes, ou seja, casos de sujeira que haviam anteriormente já sido detectados e tratados.

Os algoritmos e ferramentas disponíveis na literatura e no mercado oferecem soluções parciais, atacando parte dos problemas, com vantagens e desvantagens, mas não há

ainda trabalhos que propõem pretensões mais abrangentes envolvendo as fases de análise e transformação comprometidas com soluções eficazes.

Neste contexto, é desenvolvido um ambiente que oferece, numa única ferramenta, eficiência e eficácia no suporte a todo o processo de limpeza de dados. A arquitetura proposta contempla módulos específicos para o tratamento das etapas de análise e transformação, que são descritos com detalhes nas próximas seções deste capítulo.

3.3 Visão Geral do Sistema

O trabalho foi desenvolvido em linguagem C++ (CPLUSPLUS, 2012) suportado pelo framework de desenvolvimento Qt da Nokia (NOKIA, 2012). O sistema possui módulos para manipulação dos dados em linguagem SQL e é possível também inserir novos algoritmos ou funcionalidades em Java*Script* (JAVA*SCRIPT*, 2012) além de C++.

Uma das vantagens da ferramenta proposta consiste em sua portabilidade e expansibilidade, uma vez que foi construída de forma modularizada e pode ser portável para diferentes gerenciadores de banco de dados, desde que haja driver compilado para o framework de desenvolvimento Qt da Nokia.

Algumas de suas funcionalidades são diretamente dependentes do gerenciador de banco de dados utilizados, mas são oferecidas possibilidades de adaptação para novos gerenciadores que suportam mecanismos de controle de chaves estrangeiras e integridade referencial. A maioria dos gerenciadores de banco de dados conhecidos no mercado como PostgreSQL (PostgreSQL, 2012), Oracle (ORACLE, 2012) e DB2 (DB2, 2012) oferecem esse suporte. Caso um gerenciador mais simples seja utilizado, ainda sim será possível utilizar parte das funcionalidades do ambiente. São também utilizados repositórios auxiliares como arquivos e banco de dados embarcados (SQLITE, 2012).

Além de todas as técnicas oferecidas organizadas em módulos, o ambiente oferece diversas funcionalidades úteis voltadas à infraestrutura do processo de limpeza como co-

nexão com a base de dados por meio da interface própria, clonagem de banco de dados, restauração de *backup* de base de dados, dentre outras, as quais serão detalhadas nas próximas seções.

3.4 Arquitetura do Ambiente *Data Cleaning*

O Ambiente *Data Cleaning* proposto é organizado em módulos que suportam os processos de análise e transformação dos dados. Módulos globais auxiliares podem ser utilizados em cada uma das etapas junto aos módulos específicos, no suporte tanto da análise quanto da transformação. Sua arquitetura, ilustrada na figura 3.1, prevê módulos estruturados e independentes, possibilitando ao usuário liberdade para trabalhar da maneira flexível e realizar tarefas que vão desde a normalização, detecção e transformação de dados manualmente até utilização de todos os recursos oferecidos com o processo totalmente automatizado.

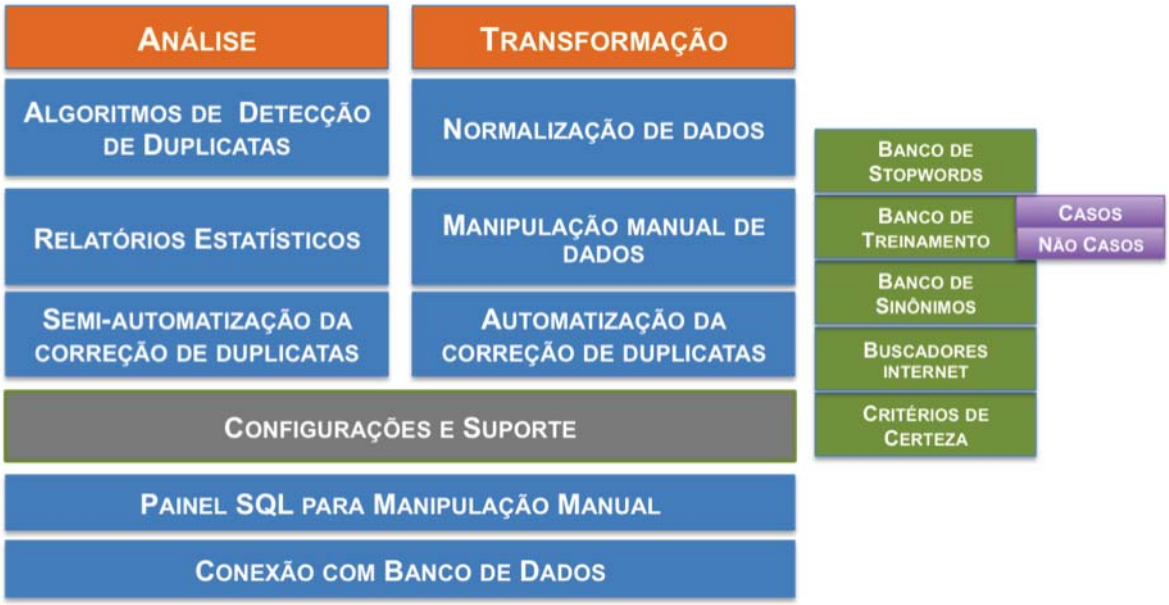

Figura 3.1: Arquitetura do Ambiente *Data Cleaning Desenvolvido*

Para apoio ao processo de análise são contemplados os módulos algoritmos de detecção de duplicatas; relatórios estatísticos e semi-automatização de correção de duplicatas. Para o processo de transformação são contemplados os módulos: normalização de dados,

manipulação manual de dados e automatização de correção de duplicatas.

A arquitetura proposta também contempla diversos módulos globais que são utilizados no suporte tanto da análise quanto da transformação: painel SQL para manipulação manual de dados; conexão com gerenciadores de banco de dados e configurações e Suporte, que envolve os seguintes sub-módulos:

- Banco de *stopwords*;
- Banco de treinamento (dividido em casos e não casos);
- Banco de sinônimos;
- Buscadores Internet;
- Configurações e regras para critério de certeza.

3.4.1 Arquitetura do Ambiente *Data Cleaning*: Análise de Dados

Há três módulos que fazem parte do processo de análise de dados cujos objetivos consistem em oferecer uma coleção de ferramentas; a análise e detecção de problemas, duplicatas e inconsistência dos dados.

O primeiro módulo oferece diversos algoritmos para detecção de duplicidade e inconsistência de dados. No segundo módulo são disponíveis relatórios estatísticos gerados por meio de análise automatizada realizada pela ferramenta, que permite ao usuário ter conhecimento do quão suja ou inconsistente a base de dados se encontra de acordo com critérios definidos e configurados. O terceiro módulo, denominado semi-automatização de correção de duplicatas, contém técnicas e algoritmos que sugerem ao usuário qual das tuplas detectadas como duplicidade é a correta, baseada em critérios referenciais e banco de históricos, caracterizando a ferramenta como inteligente e treinável e, dessa forma, auxiliando-o na tomada de decisão para a realização das correções e limpeza dos dados.

Algoritmos para detecção de duplicidade e inconsistência de dados

Existem vários algoritmos e suas variações para detecção de duplicatas nas bases de dados; cada um com objetivo de atacar um domínio específico. A fim de tornar o ambiente e os recursos utilizados mais completos e aplicáveis possível, foram implementados os principais algoritmos apresentados na literatura cujas técnicas envolvem similaridade de caracteres, similaridade fonética e *token*s. Sempre que o processo de análise e detecção de duplicidade e inconsistência de uma base de dados for iniciado, é necessário escolher um dentre os algoritmos implementados no ambiente *Data Cleaning*. é apresentado na figura 3.2 a interface da ferramenta em que o usuário escolhe um dos algoritmos disponíveis, estes são descritos a seguir.

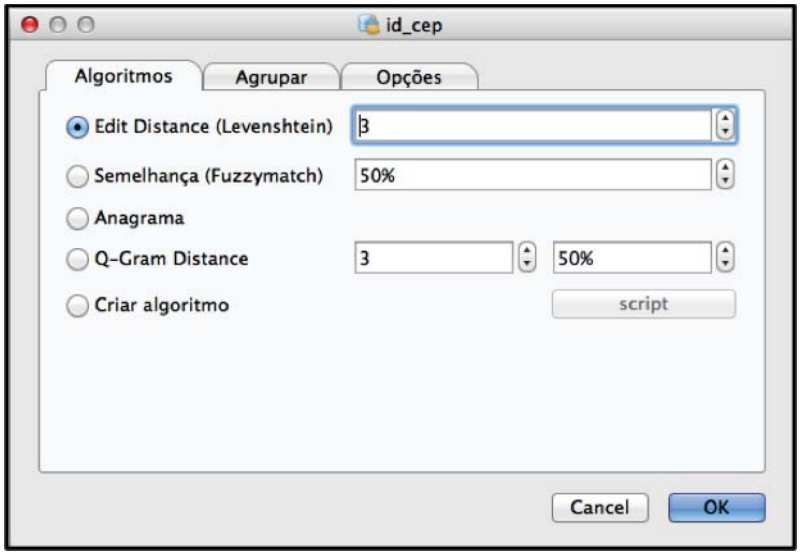

Figura 3.2: Algoritmos de Deteção de Duplicatas

1. *Edit distance* (Levenshtein): Este algoritmo verifica a semelhança entre as tuplas de uma tabela do banco de dados baseado na quantidade de edições necessárias para que um atributo seja transformado no outro, com as operações de edição admissíveis (inserção, remoção ou substituição de um único caractere). Ao escolher esse algoritmo, deve ser informada a distância desejada, em que quanto menor, mais semelhante fisicamente um atributo será de outro;

2. Semelhança (Fuzzy Match): é verificada neste algoritmo a semelhança entre tuplas de uma tabela baseada na semelhança parcial de uma com relação a outra, isto

é, o quanto um atributo contém o outro e assim detectar palavras incompletas ou abreviadas. Ao escolher esse algoritmo, é necessário informar o percentual de semelhança a ser detectada;

3. Anagrama: O algoritmo denominado anagrama foi desenvolvido com o intuito de detectar palavras que são escritas incorretamente ou com letras invertidas, ocorrências comuns e geralmente devido a erros de digitação. é verificada toda sua semelhança física, mesmo que fora de ordem para detectar a similaridade com as demais tuplas da base de dados e caso as quantidades e letras forem equivalente entre os atributos, são detectadas como possível duplicação;

4. *Q-Gram Distance*: Algoritmo baseado em *token* que quebra a palavra em vários grupos de letras e compara cada um com os grupos de outras tuplas, verificando sua semelhança pelo percentual de grupos idênticos. Na escolha desse algoritmo é necessário informar o tamanho do *token*, isto é, a quantidade de caracteres de um grupo, e o percentual de semelhança, para que sejam detectadas as duplicidades. Este percentual corresponde à quantidade de grupos de uma tupla que é idêntica as demais, por exemplo, se um atributo possui 10 grupos de 3 letras, se esses 10 grupos forem iguais a grupos de um outro atributo, sua semelhança será considerada 100%.

5. Novos algoritmos para detecção de semelhança: O ambiente *Data Cleaning* oferece a opção para o usuário criar novos algoritmos de detecção de semelhança de atributos por meio da interface do sistema, que oferece suporte à linguagem Java*Script*, em um painel que valida a sintaxe da linguagem, ilustrado na figura 3.3, e possibilita flexibilidade total para o usuário aplicar técnicas específicas, além das já oferecidas, de acordo com sua necessidade. Essa é uma das maneiras de expansibilidade que ao ambiente contempla, mas também é possível que outros algoritmos sejam incluídos em linguagem C++, como os nativos do ambiente. Neste módulo, o usuário poderá utilizar duas variáveis, str1 e str2, que correspondem a dois atributos da tabela alvo da aplicação da limpeza. Todas as tuplas são percorridas automaticamente em

que cada uma é comparada com todas as demais e a cada comparação, os valores são atribuídos às variáveis str1 e str2. Dessa forma, o algoritmo customizado pode ser desenvolvido de tal maneira que o módulo retornará TRUE ou FALSE como resposta de acordo com as condições implementadas pelo código utilizando as duas variáveis.

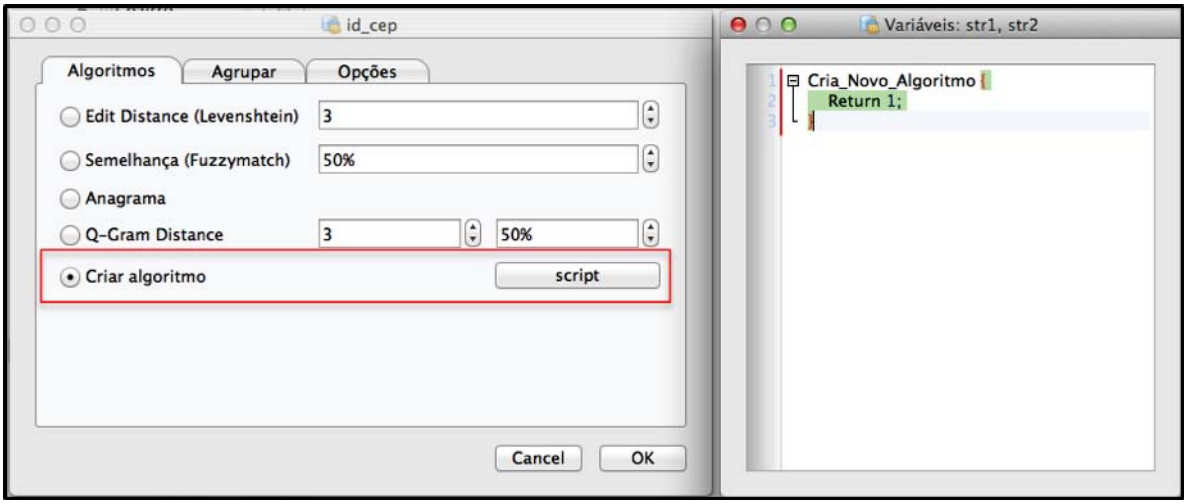

Figura 3.3: Painel para criação de novos algoritmos

Sinônimos

A funcionalidade sinônimos foi desenvolvida com o intuito de cobrir a falta de semântica no processo de detecção de inconsistências e duplicidades de informações nas base de dados. Com um banco de dados de sinônimos configurado pelo usuário, é realizada uma varredura na base de dados para que a detecção de semelhança seja feita não somente por meio das técnicas do algoritmo escolhido, mas incluídas também verificações dos termos semanticamente equivalentes, como por exemplo *ajudante, auxiliar* e *colaborador*.

Dependendo do contexto, as palavras podem ter mesmo significado, mas não seriam facilmente detectadas se não houvesse um dicionário de termos equivalentes. Um exemplo que auxilia na ilustração desses casos é o encontrado em uma base de dados de acidentes de trabalho em que são diagnosticadas pelo médico responsável pelo atendimento do acidentado como causador a mordedura de animais. Geralmente o acidentado relata que

foi mordido ou picado por um animal específico, como cachorro, cobra, etc, mas estatisticamente é importante que sejam apenas obtidos os acidentes em que foram ocasionados por animais. Neste caso, é possível incluir no banco de sinônimos uma lista de animais e sempre que algum deles for detectado, o sistema automaticamente os converterá para *animal*, e sua limpeza será realizada de forma automatizada.

O Ambiente *Data Cleaning* desenvolvido oferece a opção do banco de sinônimos para o usuário ir enriquecendo o dicionário à medida em que a limpeza é constantemente realizada.

Banco Multi-idioma

O banco multi-idioma possibilita ao usuário carregar dicionários de tradução de diversos idiomas a fim de transformar o ambiente num conjunto de ferramentas que consiga detectar duplicidades de informação independentemente do idioma em que esteja. Uma vez configurado, serão detectados atributos e dados semelhantes ou idênticos semanticamente mas que estejam descritos em diferentes idiomas.

Por exemplo, uma tabela que contém dados de animais, seriam detectadas como semelhantes tuplas com atributos cachorro, cão, dog, perro, vira-lata, cãozinho, etc. Com essa funcionalidade se é quebrada a barreira de idioma entre repositórios de dados, tornando-se factível a preparação de uma base integrada de idiomas distintos para extração de conhecimento por exemplo, após aplicação do processo de limpeza em que todas as informações seriam limpas e normalizadas para um idioma-chave.

Detecção Fonética Multi-idioma

Para cada uma das técnicas e algoritmos contemplados pelo ambiente desenvolvido é também oferecida a opção que considera a fonética das palavras no processo de análise dos dados. Dessa maneira, cada uma das técnicas considerará as diferenças fonéticas dentre as possibilidades de escrita de uma mesma palavra ao percorrer cada uma das tuplas da tabela em busca de semelhanças e duplicidades. Tais evidências ocorrem pela inserção incorreta de um dado, como erro ortográfico, bastante recorrente em sistemas com entrada

de dados por usuários comuns, erros de digitação, por exemplo *S* ao invés de *SS*, *CH* ao invés de *X*, etc.

O algoritmo implementado contempla as seguintes conversões fonéticas da língua portuguesa, representados na tabela 3.1:

Letra / Sílaba	Fonema	Condição
X	Z	Se não estiver no começo nem no final e for cercado por vogais, tem som de Z
Ç	S	
CH	X	
CE	SE	
CI	SI	
CA	KA	
CO	KO	
CU	KU	
GE	JE	
GI	JI	
PH	F	
QU	K	
RR	R	
SS	S	
S	Z	Se não estiver no começo nem no final e for cercado por vogais, tem som de Z
Z	S	Se for a última posição, retorna S
W	V	

Tabela 3.1: Transições Fonéticas Implementadas

Visando a expansibilidade do ambiente, as transcrições fonéticas são armazenadas na base de dados embarcada da ferramenta e podem ser enriquecidas com novas regras, permitindo também a inserção de regras fonéticas de quaisquer idiomas além da língua portuguesa.

Agrupamento de Atributos

Uma das funcionalidades importantes desenvolvidas consiste na possibilidade de aplicar o processo de detecção de duplicatas de uma tabela considerando apenas parte de seus atributos. A opção *agrupar*, ilustrada na figura 3.4 permite que o processo detecte a semelhança de acordo com alguma técnica ou algoritmo escolhido sobre atributos cujos

campos agrupados são idênticos. Dessa maneira, é possível restringir ou manipular parte das características de uma entidade de acordo com sua representação ou denotação do mundo real.

Uma aplicação bastante funcional dessa opção consiste em seu uso em uma base de dados de endereços, como por exemplo, a dos Correios. Ao efetuar a limpeza de duplicação de municípios por exemplo, a fim de evitar que duplicidades incorretas sejam encontradas, como os de mesmo nome em estados diferentes, é relevante configurar-se o agrupamento pelo campo UF e, assim, somente os municípios duplicados de um mesmo UF serão levados em consideração.

Figura 3.4: Opção para Agrupar Atributo(s)

Essa possibilidade faz com que os atributos que caracterizam uma entidade, mapeada do mundo real para esquemas de base de dados, sejam analisados semanticamente no processo de detecção de duplicidades, uma vez que, se utilizado de forma apropriada, evita que pseudo-duplicações sejam detectadas.

Relatórios Estatísticos

A fim de oferecer um panorama estatístico da base de dados com relação à sua inconsistência, o Ambiente *Data Cleaning* oferece alguns relatórios em forma de tabelas e gráficos de barra e pizza. é possível configurar a opção de pré-contagem, exibida na figura 3.5, em que sempre antes que uma limpeza é realizada, são contabilizadas as quantidades de tuplas da tabela e, após sua conclusão, são mostradas as diferenças, ou seja, o quanto e percentual da tabela foi corrigida ou limpa. Desde modo, é possível a cada tabela limpa, obter o percentual de limpeza.

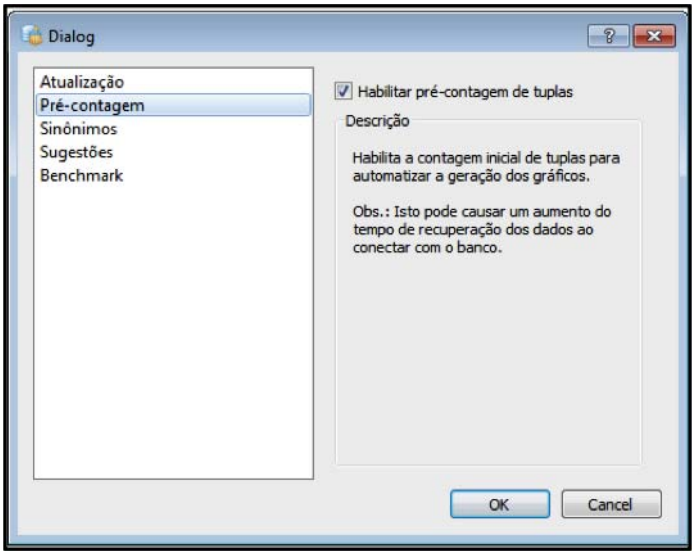

Figura 3.5: Configuração de Pré-Contagem

Os gráficos exibidos podem ser em forma de barras, linhas ou pizza, como ilustrado na figura 3.6.

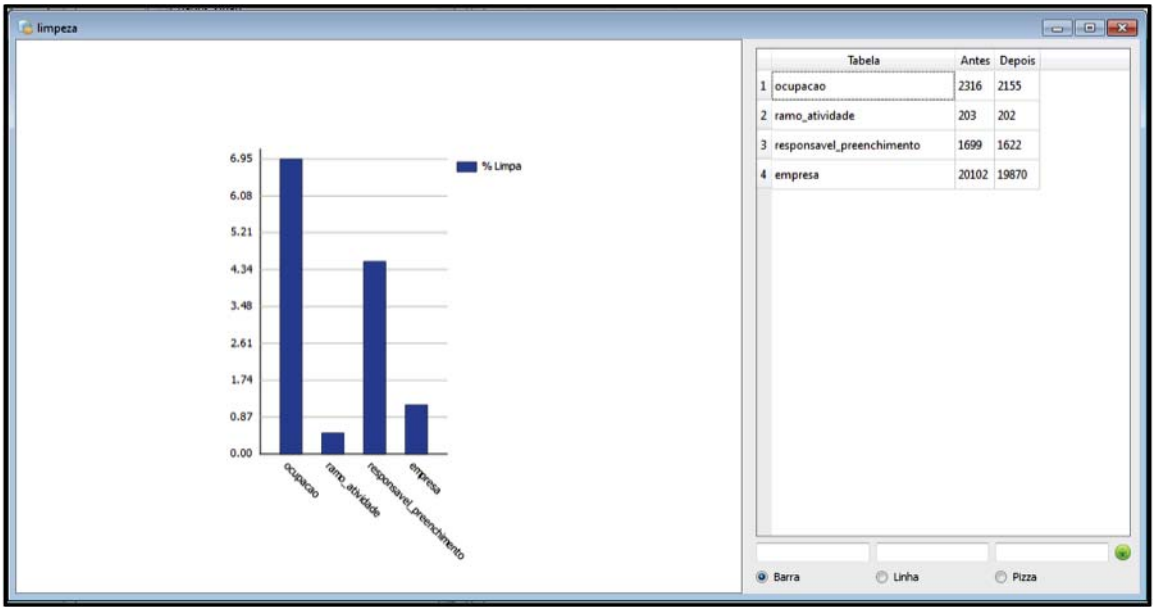

Figura 3.6: Gráfico de Barras do Percentual de Limpezas Efetuadas

Também são exibidos relatórios de todas limpezas realizadas, para que o usuário acompanhe e analise as execuções e correções. Neste tipo de relatório, é possível verificar quais semelhanças foram detectadas e para que tupla as demais duplicadas foram convertidas,

destacadas em cor azul, como mostrado na figura 3.7.

id	
2657	CHRISTIANE IAMACHITA LIMA DOS SANTOS
2606	CHRISTIANE SAMACHITA LIMA DOS SANTOS
^ Referências	
2663	WLADIMIR RICARDO MARTINELLI FILHO
1886	VLADIMIR RICARDO MARTINELLI FILHO
^ Referências	
662	MARINA DA SILVA
197	KARINA DA SILVA
^ Referências	
2706	AMANDA
1652	AMANDA

Figura 3.7: Relatório de Limpezas Realizadas

Limite de Caracteres

É possível configurar a quantidade mínima de caracteres de um atributo que deverá ser considerada durante o processo de limpeza, aplicado a casos em que um atributo possui uma, duas ou três letras, sendo facilmente detectado indevidamente como duplicidade de outro atributo.

Essa funcionalidade é útil em casos de siglas, abreviaturas e pronomes de tratamento que, muitas vezes, são formados por duas ou três letras. Uma vez definido o limite de caracteres, palavras somente com quantidade superior à definida serão consideradas no processo de limpeza.

Semi-automatização da correção de duplicatas

Uma das grandes dificuldades para o usuário no processo de limpeza de dados é, muitas vezes, definir se dois ou mais atributos semelhantes correspondem de fato a uma mesma entidade. Muitas vezes palavras parecidas não são correspondentes e a conversão de uma em outra pode gerar mais inconsistência na base de dados. Isso acarreta em alguns limitantes para a execução do processo de limpeza como, por exemplo, a necessidade de um especialista no contexto das informações para a tomada de decisões e/ou um grande

esforço em buscas e pesquisas manuais para verificar quais das informações detectadas é a correta e se de fato tratam-se da mesma entidade. é importante enfatizar que essa situação é constante e que muitas limpezas realizadas precisam ser repetidamente realizadas pois a entrada de dados inconsistentes é, na maioria dos casos, impossível de ser evitada por completo.

Com a constante realização desse processo, é possível minimizar seu custo com base no histórico de limpeza efetuadas. Foi desenvolvida uma análise automatizada baseada na quantidade em que cada tupla é referenciada pelas demais; se duas ou mais tuplas são detectadas como semelhantes de acordo com o algoritmo escolhido, o sistema verifica a quantidade em que cada uma delas é referenciada pelas demais tabelas da base de dados e, com isso, indica ao usuário a tupla com maior possibilidade de ser a correta. é válido pensar que, se uma tupla é mais referenciada que a outra, a chance dela conter a informação correta é maior, principalmente depois que várias limpezas já tenham sido realizadas e que outras tuplas também tenham sido convertidas a essa, fazendo com que sua quantidade referenciada seja cada vez maior.

O processo denominado semi-automatizado de correção de duplicatas pode ser descrito com o pseudo-algoritmo a seguir:

Result: Retornar sugestão de tupla correta dentre duplicatas detectadas

Verifica semelhança com base no algoritmo escolhido;

while *Há tuplas duplicatas detectadas* **do**

Verifica se as tuplas já foram detectadas anteriormente;

if *Sim* **then**

Verifica o histórico do que foi realizado, isto é, se foi ou não efetuada a limpeza, guarda-se a tupla correta;

else

Verifica para cada tupla o quanto é referenciada pelas demais tabelas do banco de dados;

Guarda a quantidade total;

Verifica entre as tuplas do grupo qual possui quantidade maior;

end

Indica ao usuário destacando qual é a mais referenciada ou historicamente a correta;

end

Algorithm 1: Semi-automatização de correção de duplicatas

Essa funcionalidade é bastante útil para o processo manual de limpeza, principalmente porque auxilia o usuário na tomada de decisões baseada no histórico de limpezas realizadas anteriormente, por meio da quantidade das tuplas referenciadas e no panorama atual da base de dados. Até mesmo para casos em que dois atributos são praticamente semelhantes, o usuário tem auxílio da ferramenta para decidir se, por exemplo, o nome de uma empresa é escrita de uma maneira ou de outra, de acordo com a quantidade em que são referenciadas, pois, espera-se numa base de dados constantemente limpa que o nome correto seja muito mais referenciado do que um novo dado inconsistente que tenha sido inserido há pouco tempo, ou numa duplicidade que já fora tratada e corrigida anteriormente.

3.4.2 Arquitetura do Ambiente *Data Cleaning*: Transformação de Dados

A etapa de transformação de dados consiste basicamente na aplicação das transformações e correções necessárias na base de dados. O ambiente desenvolvido disponibiliza um conjunto de ferramentais que oferece diversas possibilidades do usuário interagir e realizar as mudanças e correções necessárias. Essas ferramentas estão organizadas em três módulos: ferramentais de normalização de dados, ferramentais de correções manuais de dados e automatização da correção de duplicatas.

Ferramentais de Normalização de Dados

Um dos grandes problemas de inconsistências nas bases de dados referem-se a valores incorretos, inconsistentes ou mesmo a falta deles. A ausência de padronização dos dados pode ser considerado um tipo de inconsistência que precisa também ser corrigida e, para cobrir esse item, foi incorporado à ferramenta uma área de correções e padronizações dos dados.

Na figura 3.8 é mostrada a interface do Ambiente *Data Cleaning* no modo de normalização de dados. O usuário seleciona a tabela a ser corrigida na árvore à esquerda e, no painel ao lado direito encontram-se as opções para correções e padronizações dos valores de cada tipo de atributo. Os filtros de busca, as condições para normalização e correção dos dados de cada coluna podem ser elaboradas por meio de expressão regular, oferecendo total flexibilidade ao usuário para manipulação e correção das informações. Também podem-se definir regras matemáticas e estatísticas (médias, ponderações, etc.) e padronizações (codificações, caracteres especiais etc.).

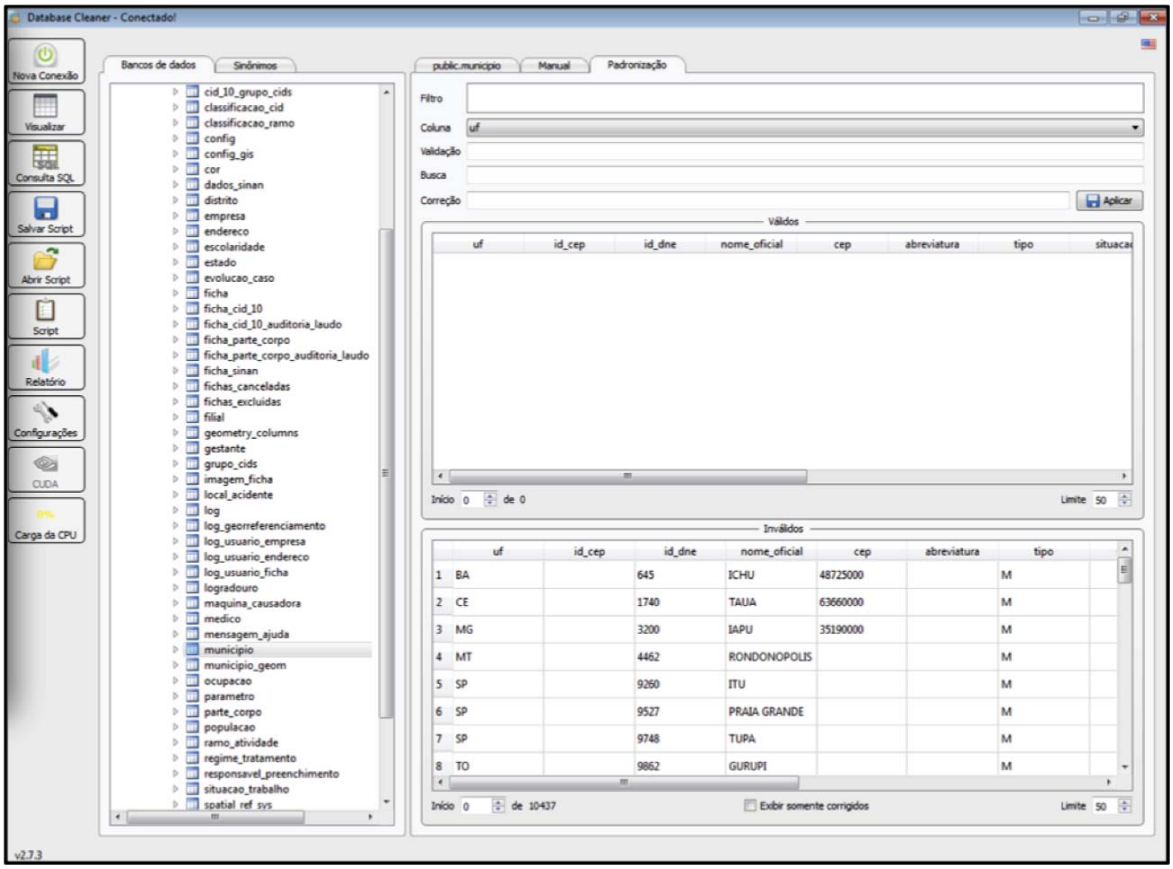

Figura 3.8: Módulo de Normalização de Dados

Com as regras devidamente definidas, clica-se no botão Aplicar para que o seja realiada em poucos segundos toda a limpeza e correção de acordo com as condições elaboradas pelo usuário utilizando expressão regular.

Ferramentais de Correção Manual de Dados

O ambiente desenvolvido contempla diversas funcionalidades e técnicas de detecção e transformação de dados e é também oferecido um painel SQL, apresentado na figura 3.9, para que o usuário tenha total liberdade de manipular as informações da base de dados através de consultas SQL ou linguagem PL-SQL.

Para usuários especializados, esse módulo representa mais uma possibilidade para realização de correções e verificações das informações contidas na base. As consultas digitadas são suportadas pela funcionalidade *syntax highligh*, isto é, os termos da linguagem são destacados com coloração diferente para facilitar o entendimento do código-fonte, e é

possível abrir consultas em formato SQL ou armazenar as desenvolvidas para utilizá-las futuramente.

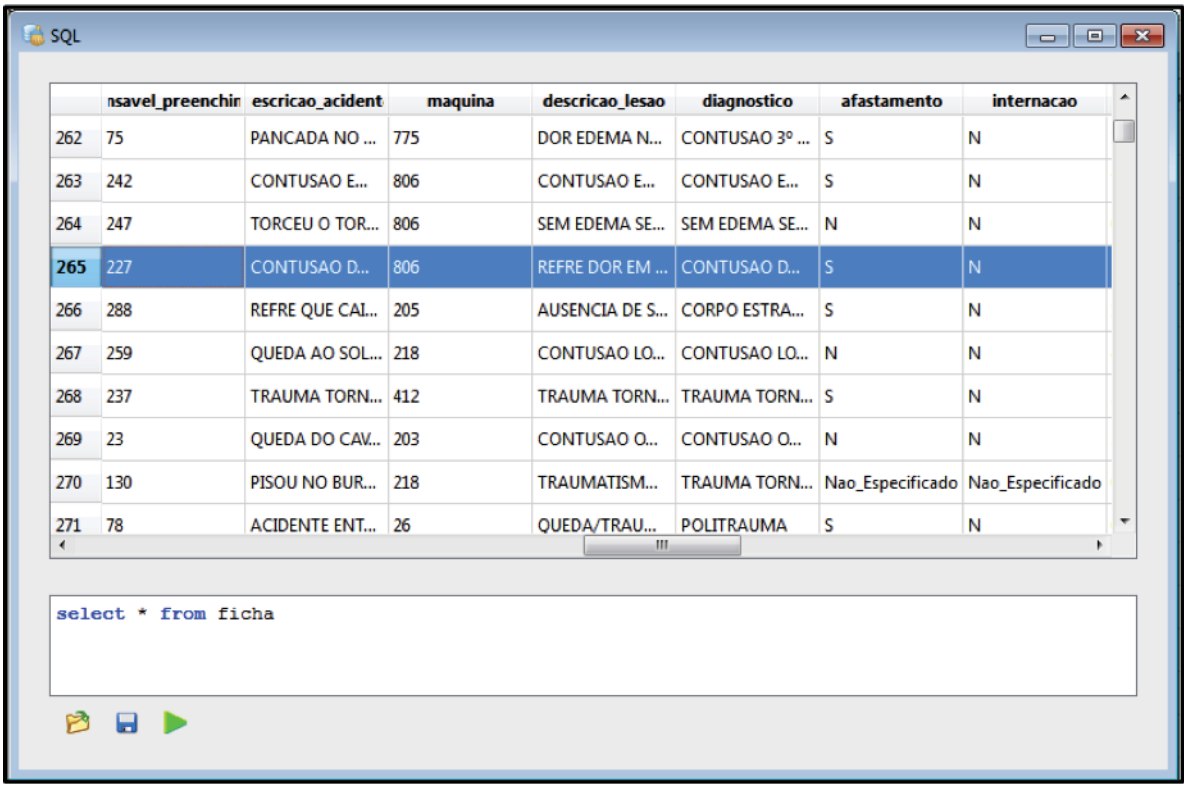

Figura 3.9: Painel SQL para manipulação manual dos dados

Automatização da Correção de Duplicatas

Um dos principais objetivos do desenvolvimento desse trabalho consiste na possibilidade de evitar ao máximo a necessidade de interação do usuário no processo de limpeza, visto que, para grandes bases de dados relacionais que contêm milhões de registros organizados em tabelas e colunas, uma análise humana para detecção de duplicidade e inconsistência é praticamente impossível sem o suporte computacional adequado.

Utilizando todas as técnicas de análise já descritas, é possível que parte, senão toda a base de dados seja limpa e tratada automaticamente. Para isso, foi desenvolvido o módulo denominado **Automatização da Correção de Duplicatas** que consiste na aplicação automática de análise e transformação da tabela de uma base de dados, sem a necessidade de interação do usuário durante o processo.

O usuário precisa apenas indicar qual tabela deverá ser limpa, escolher o tipo de atributo, o algoritmo que será utilizado para análise das detecções de duplicatas e marcar a opção Limpeza Automática, destacada em vermelho na figura 3.10. Uma vez acionado o mecanismo, o Ambiente *Data Cleaning* fará todo o trabalho de varredura, análise e correção dos dados inconsistentes e duplicados.

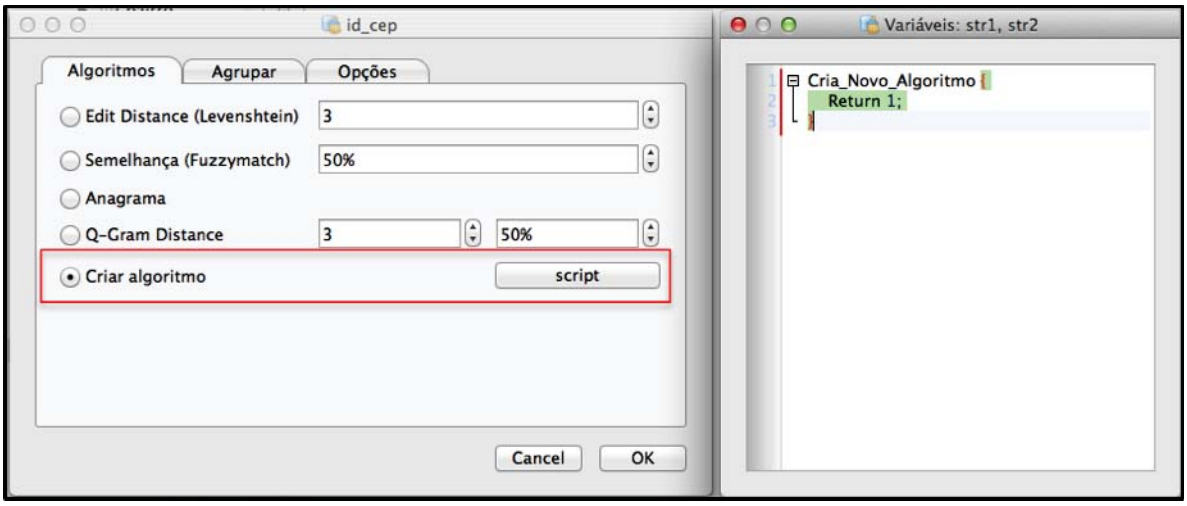

Figura 3.10: Opção para Correção Automática de Duplicatas

A fim de se evitar o máximo que a automatização da correção acabe convertendo tuplas que na verdade não são duplicidades, apesar da alta semelhança física ou semântica, foi construído um conjunto de regras e condições que garantem ao máximo uma correção automática correta, denominados **Critérios de Certeza**. Suas regras e condições são descritas pelo pseudo-algoritmo a seguir:

Result: Efetuar correção de duplicatas baseada no Critério de Certeza

Inicia-se o processo de detecção de duplicatas;

Detectam-se duas ou mais possíveis duplicadas;

if *atributos são idênticos* **then**

    Verifica qual das tuplas é a mais referenciada na base de dados;

    Efetua a limpeza;

else

    if *este caso já foi limpo anteriormente no histórico de limpeza* **then**

        Efetua a limpeza de acordo com o histórico;

    else

        Verifica quais das tuplas é a mais referenciada pela base de dados;

        if *Empate* **then**

            Verifica dentre as tuplas a que tem mais atributos com valores diferentes de vazio ou nulo;

            Efetua a limpeza;

        else

            Caso não haja desempate, a limpeza não é executada;

        end

    end

end

Algorithm 2: Critério de Certeza

As condições contempladas pelo Critério de Certeza contribuem para que as correções realizadas automaticamente não tenham falhas. Obviamente que o processo é altamente dependente do algoritmo e das configurações escolhidos para que as tuplas detectadas como possíveis duplicatas sejam de fato equivalentes.

O ambiente também possibilita uma verificação adicional antes de que todas as correções sejam de fato realizadas. Assim que a varredura por toda a tabela é finalizada, o sistema exibe ao usuário um relatório, exemplificado na figura 3.11, com todas as transformações que serão realizadas. Dessa maneira, é possível uma última análise, caso desejado

e alguma correção antes que a transformação seja consolidada.

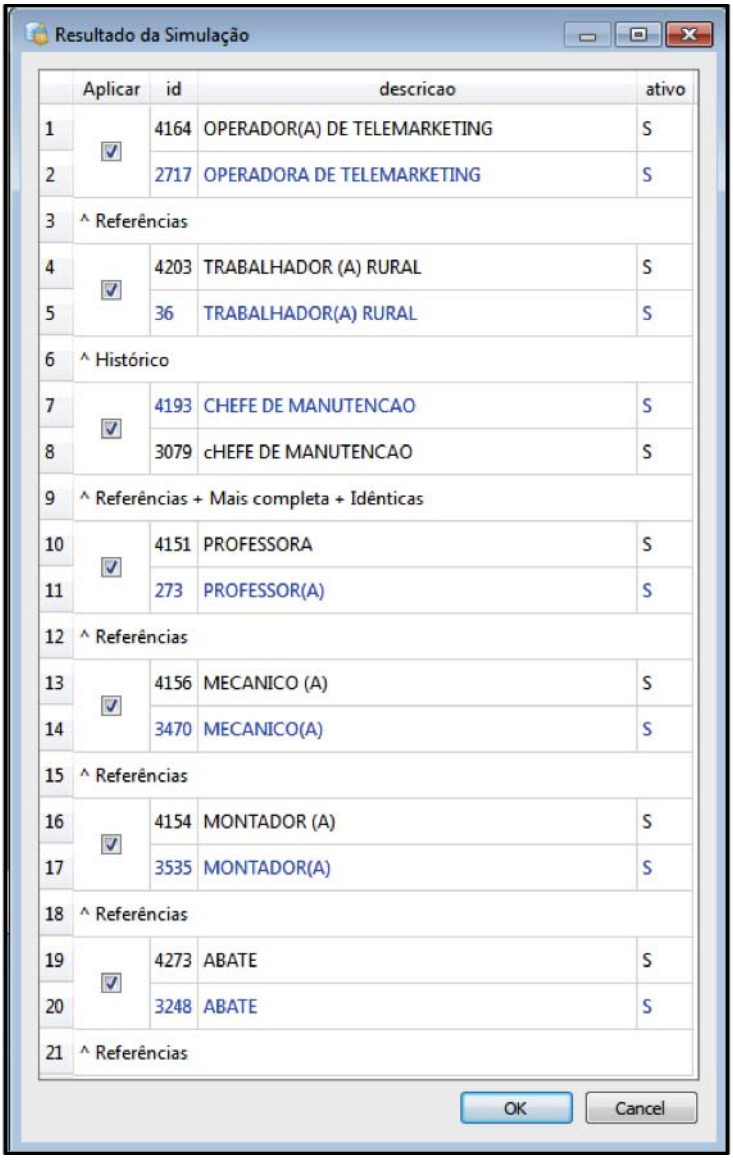

Figura 3.11: Relatório da Simulação de Correção Automática de Duplicatas

Nota-se que no relatório são exibidos todos os casos de possíveis duplicidades encontrados na tabela alvo e, de acordo com o critério de certeza, a tupla correta é destacada em cor azul. Pode-se então confirmar as correções sugeridas pelo sistema antes de aplicá-las de fato. A coluna do relatório *Aplicar* possibilita ao usuário desmarcar os casos que, após sua análise, não representam uma duplicidade real ou não devam ser alterados. Após a análise, ao confirmar, o Ambiente *Data Cleaning* se encarregará de automaticamente efetuar todas as correções detectadas em alguns segundos.

Treinamento do Ambiente *Data Cleaning*

A cada limpeza realizada, seja manual, semi-automatizada ou automatizada, o Ambiente *Data Cleaning* se encarrega de armazenar em um banco de treinamentos os casos em que as limpezas foram realizadas. Na opção manual, todos os casos de tuplas convertidas são armazenados como histórico positivo e, quando o mesmo caso for recorrente, o sistema automaticamente sugerirá a tupla correta, igualmente para a limpeza semi-automatizada. Para a opção de limpeza automatizada, todas as tuplas são já automaticamente convertidas, de acordo com o banco de histórico.

Da mesma maneira, o treinamento é realizado para os casos negativos, em que são armazenados no histórico negativo todos os casos nos quais o algoritmo, de acordo com os critérios e parâmetros configurados pelo usuário, indica possível duplicação, mas que, apesar de semelhantes, não correspondem a duplicatas. Nestes casos, é possível indicar que se trata de um falso-positivo e, uma vez indicado, os casos recorrentes serão automaticamente ignorados pela ferramenta.

No caso de falsos-positivos, é necessária a interação do usuário para treinar a ferramenta, indicando quais casos se tratam de não-duplicidades. Com o auxílio da limpeza automatizada, pode ser rápida essa análise e indicação, uma vez que será gerado um relatório de todos os possíveis casos encontrados para que o usuário possa marcá-los como falsos-positivos. Com todos os casos marcados, o histórico passa a ser registrado e, em limpezas futuras, as recorrências serão automaticamente descartados pelo Ambiente *Data Cleaning*.

É ilustrado na figura 3.12 o processo de treinamento de falsos-positivos. O ícone *lâmpada* indica que o caso identificado pela ferramenta é correto, ou seja, trata-se de uma duplicata. Nos casos de não-duplicidades, o usuário deverá clicar no ícone *lâmpada*, que ficará desmarcada, em cor cinza e, automaticamente, será armazenado no banco de histórico que o caso não se trata de duplicação.

À medida em que a ferramenta é treinada, a limpeza de uma base de dados específica

torna-se cada vez mais confiável e rápida, além de tornar o processo automatizado bastante conveniente, pois os casos que serão tratados de forma automática serão mais confiáveis.

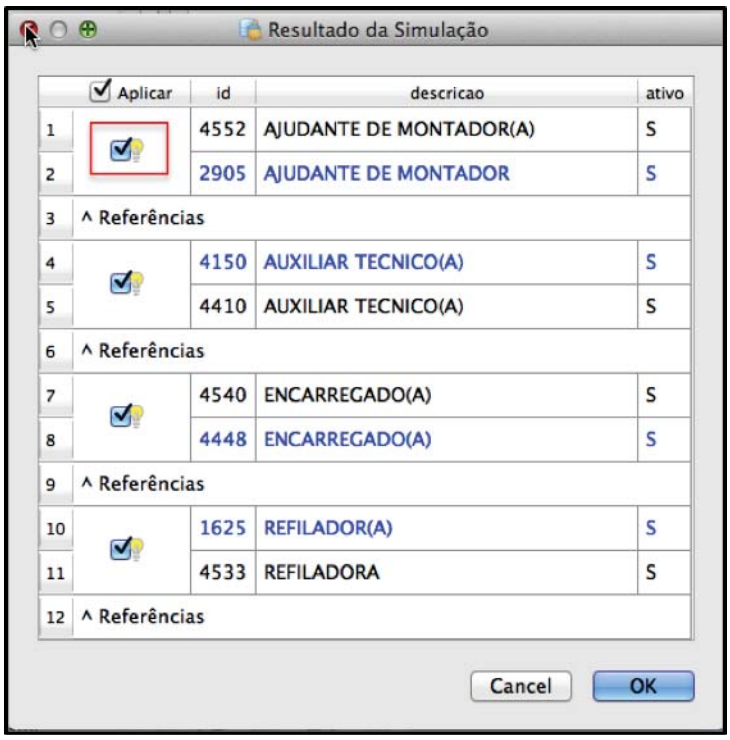

Figura 3.12: Treinamento para casos de falsos-positivos

Configurações

Os módulos do Ambiente *Data Cleaning* são independentes e expansíveis, mas todos utilizam ferramentais e funcionalidades presentes nos módulos auxiliares, denominados de configurações. Tanto na etapa de análise quanto de transformação, os módulos de configuração podem ser utilizados para melhorar ainda mais os critérios, condições e eficácia do processo de detecção e correção de duplicatas, seja ele manual, semi-automatizado ou automatizado.

Os módulos de configuração são organizados em quatro categorias: banco de *stopwords*, banco de treinamento (dividido em histórico positivo e histórico negativo), banco de sinônimos e buscadores Internet.

Banco de *Stopwords*

Muitos atributos possuem nomes compostos ou um grupo de palavras para caracterizá-lo e parte desses nomes podem ser compostos por artigos, preposições, conjunções e demais termos (adjuntos) de ligação que não influenciam no significado ou caracterização das entidades. Esses termos são denominados stopwords e podem ser ignorados durante a análise de semelhança entre os atributos.

Para isso, foi criado um banco de stopwords configurável e extensível que é consultado toda vez que qualquer algoritmo for aplicado numa tabela para detecção de duplicatas. Com isso, a eficácia da ferramenta aumenta consideravelmente, uma vez que são descartados todos os stopwords das tuplas antes da comparação.

Por ser totalmente configurável, a ferramenta permite que novos termos sejam criados gradativamente, inclusive suportando multi-idiomas, permitindo consolidar seu caráter extensível e flexível a qualquer língua. Mantendo um grupo de stopwords em inglês e em português, por exemplo, se a base de dados possuir dados em ambos os idiomas, os stopwords serão ignorados. é mostrado na figura 3.13 como o banco de stopwords pode ser organizado e expandido, utilizando o aplicativo SQLite.

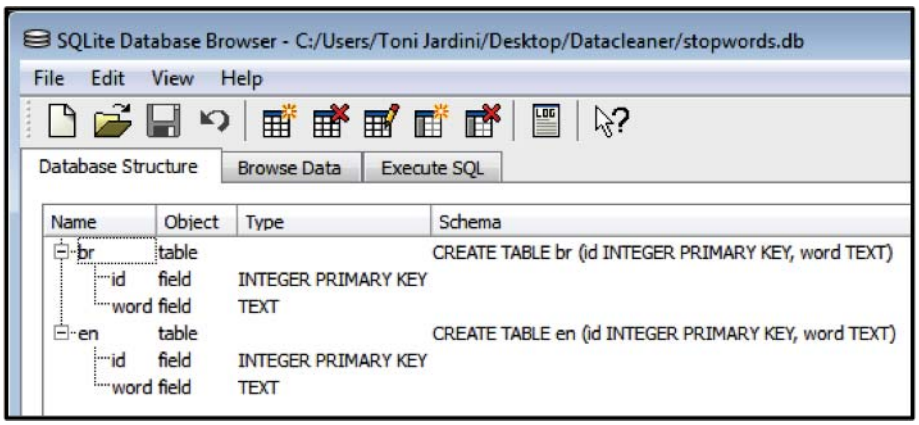

Figura 3.13: Banco de Stopwords em Idiomas Inglês (en) e Português (br)

Banco de Histórico

Uma das iterações mais custosas do processo de limpeza de dados é a análise necessária para identificar os problemas de inconsistência ou duplicações presentes em um banco de

dados que, muitas vezes, precisa ser realizada repetidas vezes devido ao fato de que os mesmos problemas de inconsistências são recorrentes. é coerente pensar que, na entrada de dados, os usuários que alimentam o sistema cometem os mesmos erros e que é esperado que os mesmos problemas se repitam ao longo do tempo.

As ferramentas e técnicas disponíveis no mercado ou apresentadas na literatura não oferecem suporte a esse tipo de caso, e, cabe ao usuário limpar e corrigir constantemente erros e inconsistências que são usualmente corrigidas.

A fim de solucionar o problema dessa iteração desgastante, foi criado um banco de histórico no ambiente *Data Cleaning* que guarda todas as ações e limpezas anteriormente realizadas, como descrito na seção Treinamento do Ambiente *Data Cleaning*.

É possível também manipular o treinamento armazenado, inserindo regras já pré-estabelecidas ou modificar qualquer histórico realizado, caso necessário ou conveniente. é mostrado na figura 3.14 o banco de histórico e sua interface de consulta e manipulação, também realizada pela interface do SQLite.

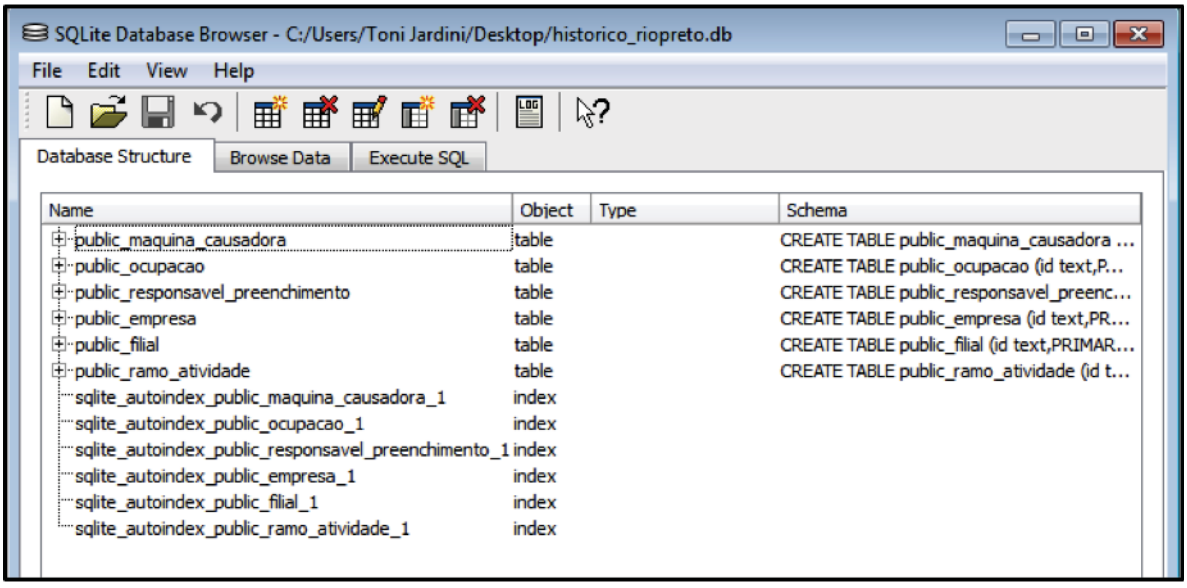

Figura 3.14: Banco de Histórico e Treinamento

Banco de Sinônimos Multi-Idioma

O módulo Banco de Sinônimos Multi-Idioma representa um dos grandes diferenciais do

ambiente *Data Cleaning* proposto neste trabalho com relação às ferramentas e trabalhos encontrados na literatura e no mercado, pois pouco ainda foi desenvolvido para suportar a semântica e idiomas distintos no processo de limpeza de dados.

Com o banco de sinônimos multi-idioma, ilustrado na figura 3.15, é possível definir um conjunto de palavras que representam um mesmo termo e, paulatinamente, enriquecer o banco de sinônimos para que a ferramenta cubra cada vez mais casos. é possível também importar um ou mais dicionários prontos e assim já trabalhar com um amplo banco de sinônimos que, uma vez configurado, será consultado e utilizado em todo o processo de análise e transformação dos dados, seja ele manual, semi-automatizado ou automatizado.

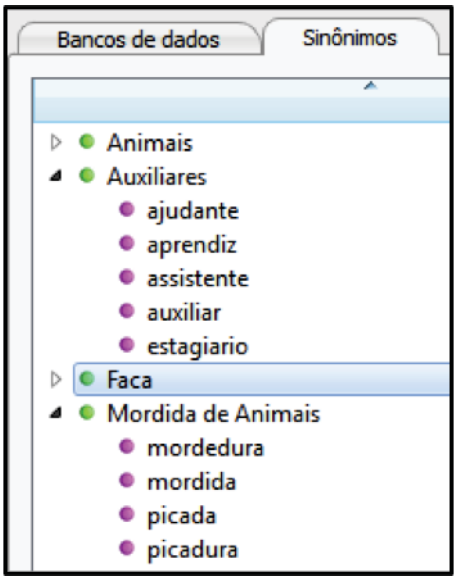

Figura 3.15: Exemplo de Banco de Sinônimos

Assim como o banco de *stopwords*, o banco de sinônimos multi-idioma foi desenvolvido de maneira a ser portável para qualquer idioma. Essa portabilidade permite que o usuário carregue não somente dicionários de sinônimos, mas também dicionários de tradução, podendo transformar o ambiente num conjunto de ferramentais poderosos que cubram por exemplo, a limpeza de banco de dados integrados com informações provenientes de múltiplas fontes de idiomas distintos e, assim, normalizar os termos num único idioma.

É mostrado na figura 3.16 um exemplo da possibilidade de limpeza de fontes multi-

idiomas com dicionários de tradução configurados no ambiente. Observe que para a palavra faca, são definidos alguns sinônimos em língua portuguesa (faca, facão) e língua inglesa (*knife*). Ao efetuar a análise numa tabela de utensílios de cozinha, por exemplo, serão detectadas como duplicatas as tuplas que tiverem os termos faca, facão e *knife*. Apesar da exemplificação simples, é possível agregar ao Ambiente *Data Cleaning* diversos dicionários de traduções em multi-idiomas, inclusive com suporte a caracteres especiais e idiomas que utilizam alfabetos não-latinos, como alfabeto grego, árabe, japonês, etc.

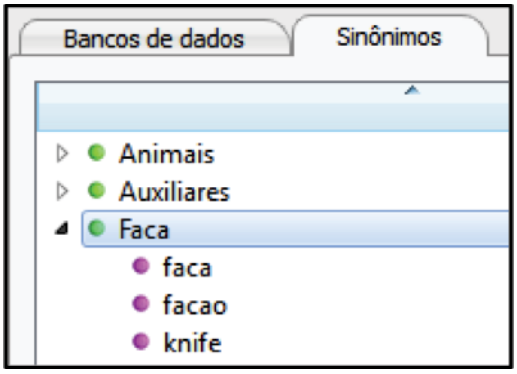

Figura 3.16: Exemplo junto ao Banco Multi-idioma de Sinônimos

Buscadores Internet

Com a opção da realização da limpeza dos dados de forma manual ou semi-automatizada, o ambiente disponibiliza ao usuário uma ferramenta denominada buscadores Internet em que a cada conjunto de possíveis duplicatas detectadas durante o processo de limpeza, são exibidos os resultados da pesquisa realizada em um dos buscadores disponíveis na ferramenta pelo termo analisado, oferecendo assim maior informação e suporte na tomada de decisão pelo usuário.

Os buscadores de Internet, como o Google, podem ser facilmente adicionados ao Ambiente *Data Cleaning* e o usuário tem a possibilidade de escolher, dentre o nicho de informações a serem analisadas, o buscador que mais se adequa ao contexto dos dados da tabela alvo. As informações exibidas no painel ao lado direito, como ilustrado na figura 3.17, oferecem grande ajuda para decidir dentre as tuplas, qual contempla informações

consistentes e deve ser considerada a correta. Vale lembrar que, para que essa funcionalidade esteja ativa, é necessário acesso à Internet no computador onde o usuário esteja utilizando o Ambiente *Data Cleaning*. Essa funcionalidade é bastante útil nas primeiras limpezas realizadas, fase em que o usuário ainda treina a ferramenta para as futuras limpezas.

São contemplados atualmente na ferramenta os seguintes buscadores internet: google, apontador, guia linha direta e telelistas.

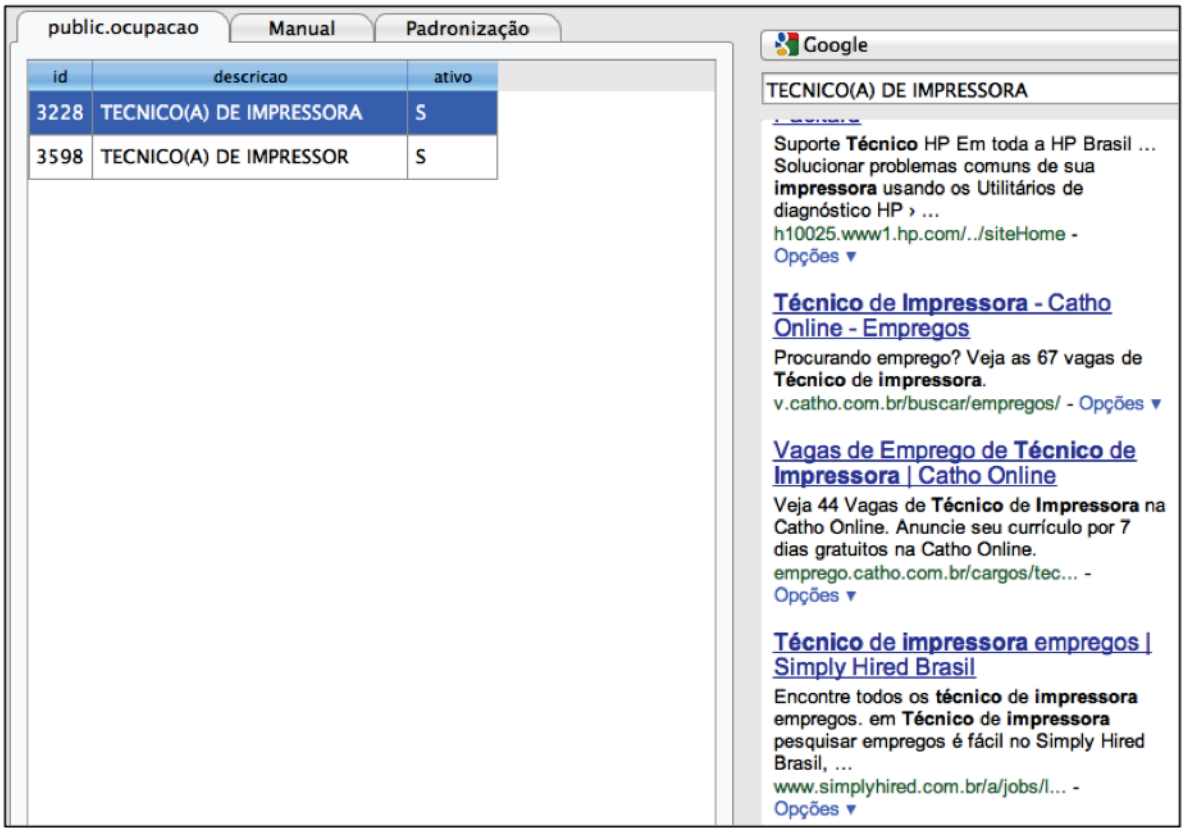

Figura 3.17: Exemplo de Uso de Buscadores Internet

3.4.3 Demais funcionalidades do Ambiente *Data Cleaning*

Além das funcionalidades para a análise e transformação dos dados no processo de limpeza, são oferecidas algumas ferramentas para ajudar o usuário durante sua interação com o sistema.

Cada base de dados que for ser analisada através do ambiente pode ser acessada

diretamente da interface. é possível também, como mostrado na figura 3.18, realizar *backup*s das bases ou até mesmo cloná-las, a fim de que experimentos iniciais e demais testes sejam realizados numa base cópia, evitando danificar a base original.

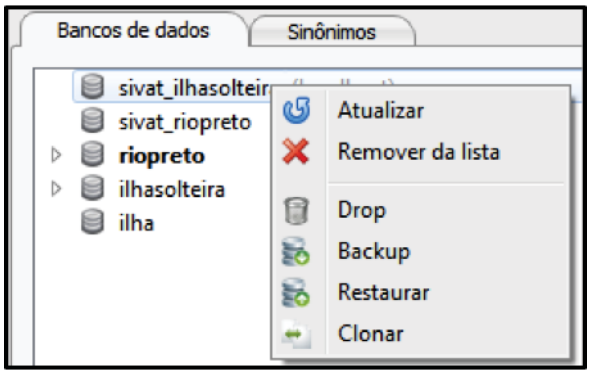

Figura 3.18: Funcionalidades de Acesso, *Backup* e Clonagem da Base de Dados

Toda a limpeza realizada durante o processo é aplicada automaticamente na base de dados utilizada. Além dos relatórios de limpeza, o sistema cria o *Script* SQL de toda a transformação realizada, sendo possível aplicar posteriormente na base de dados oficial, evitando que erros danifiquem os dados originais. A flexibilidade oferecida permite ainda que o usuário manipule o *Script* SQL final caso deseje e, somente após sua aprovação, aplique toda a correção das informações na base de dados oficial.

São ilustradas na figura 3.19 as funcionalidades de verificar o *script* gerado, salvá-lo em arquivo ou até mesmo abrir um já existente, para continuar o processo de limpeza iniciado num momento anterior.

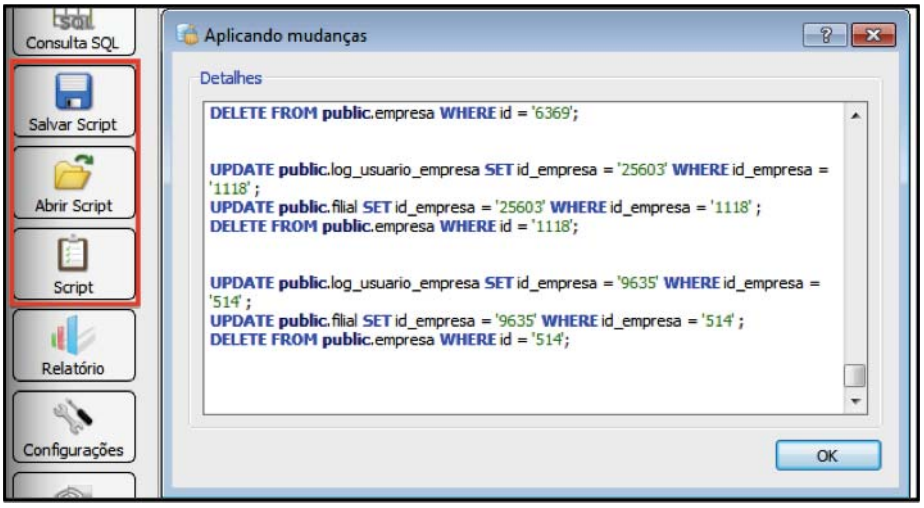

Figura 3.19: Funcionalidades para *Script* SQL de Limpeza Gerado

O Ambiente *Data Cleaning* também foi desenvolvido para ser uma plataforma multi-idioma e atualmente oferece opção para línguas portuguesa e inglesa. é possível, alterando o banco de dados disponível, acrescentar novos idiomas e assim compartilhar seu uso com pessoas de diferentes países. Na figura 3.20 é mostrada a interface do ambiente em idioma inglês. Para alternar dentre os idiomas disponíveis, basta clicar sobre o ícone da bandeira do Brasil ou dos Estados Unidos. Vale ressaltar que essa funcionalidade é referente ao idioma da interface da ferramenta ou seja, botões, ícones, mensagens. O suporte multi-idioma para o processo de limpeza de duplicidades trata-se de um módulo diferente, diretamente ligado com o processo de detecção, e não com a interface da ferramenta.

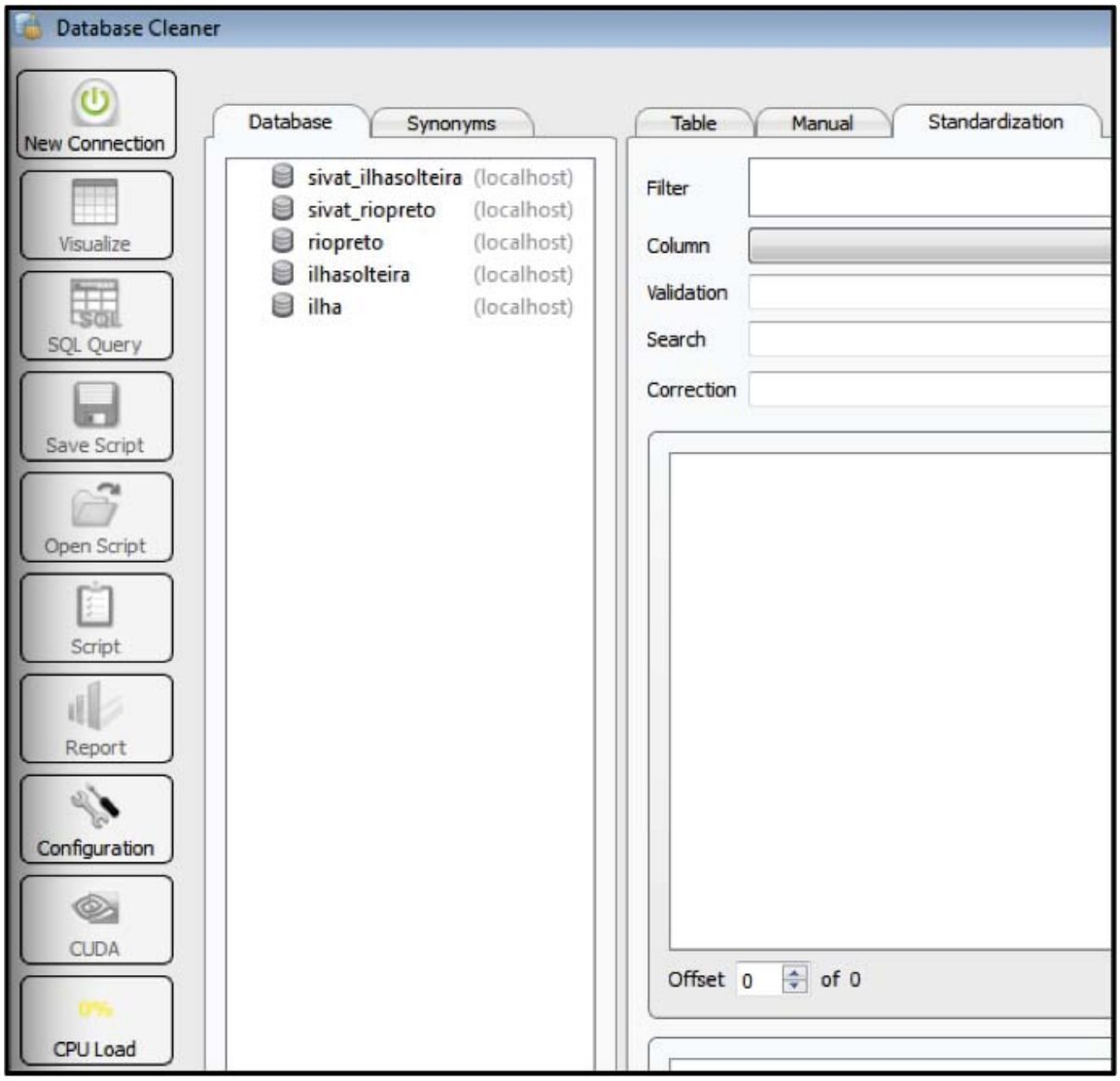

Figura 3.20: Ambiente *Data Cleaning* em Idioma Inglês

3.5 Considerações Finais

Neste capítulo foi apresentado o Ambiente *Data Cleaning* desenvolvido, que oferece recursos para a automatização configurável de limpeza de dados baseada em treinamentos e semelhança físico-semântica de dados, com objetivo de oferecer uma solução mais eficaz e extensível para realização de correção de informações.

Foram apresentados todos os módulos desenvolvidos da arquitetura do Ambiente *Data Cleaning* e discutidas suas funcionalidades e cobertura. é importante ressaltar que algumas lacunas pouco ou não exploradas na literatura como treinamento, suporte semântico,

suporte multi-idioma e automatização parcial ou total do processo de limpeza foram desenvolvidas e apresentadas com detalhes neste capítulo.

4 Experimentos e Resultados

4.1 Considerações Iniciais

Neste capítulo são demonstrados por meio de experimentos, os resultados pela aplicabilidade de cada um dos módulos do ambiente e suas funcionalidades principais. São também apresentados experimentos realizados que medem os resultados do processo automatizado de limpeza e o ganho que pode trazer aos usuários tanto em eficiência - tempo que se é economizado, quanto eficácia - problemas e duplicidades detectadas.

Os experimentos são organizados em duas etapas: análise e transformação, cujo objetivo principal é demonstrar que cada um dos módulos do Ambiente *Data Cleaning* atendem de forma satisfatória seu propósito e oferecem recursos inovadores com características de *framework* devido à sua extensibilidade.

4.2 Dados e Configurações Utilizadas nos Experimentos

Foi utilizada em todos os experimentos realizados neste trabalho a base de dados do sistema SIVAT - Sistema de Vigilância de Acidentes de Trabalho que gerencia informações sobre acidentes de trabalho dos municípios de São José do Rio Preto, Ilha Solteira e região, totalizando mais de cem cidades. Atualmente, possui cerca de 90 mil notificações cadastradas de acidentes coletados pelo CEREST - Centro Regional de Referência em Saúde do Trabalhador, órgão vinculado à prefeitura dos dois municípios e responsável pela gestão e ações que envolvem a prevenção e investigação dos acidentes de trabalho.

O sistema SIVAT foi desenvolvido e é mantido pelo GBD - Grupo de Banco de Dados do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista - UNESP, câmpus São José do Rio Preto / SP.

O sistema gerenciador de banco de dados do sistema é o PostgreSQL com extensão PostGIS. Destaca-se que o ambiente dá suporte a todos os gerenciadores de banco de dados que tenham driver compilado para a plataforma de desenvolvimento Qt, da Nokia.

4.3 Resultados de Testes Realizados das Funcionalidade do Ambiente *Data Cleaning*

Nesta seção são apresentados e discutidos os resultados obtidos da aplicação de cada um dos módulos e funcionalidades do Ambiente *Data Cleaning*. Todos os testes realizados apresentam resultados de sua aplicabilidade numa base de dados do mundo real.

Normalização e Padronização de Tuplas

Para exemplificar as funcionalidades de normalização e padronização de tuplas do Ambiente *Data Cleaning*, foi utilizada a ferramenta de normalização e padronização de tuplas sobre a base de dados o sistema SIVAT. A notificação de acidentes de trabalho contém informações do número de telefone da empresa onde o acidentado trabalha, que é inserida no sistema através de um campo tipo texto (livre de validações).

Na figura 4.1 são mostrados alguns números de telefones de empresas inseridos pelo usuário no sistema SIVAT. Observa-se que não há padronização entre os dados e cada telefone é cadastro de uma maneira distinta. A fim de padronizar esses dados automaticamente, foi utilizado o módulo de padronização do ambiente. Utilizando expressão regular, pôde-se transformar todos os números com o aspecto XX-XXXXXXXX para o formato (0XX) XXXXXXXX.

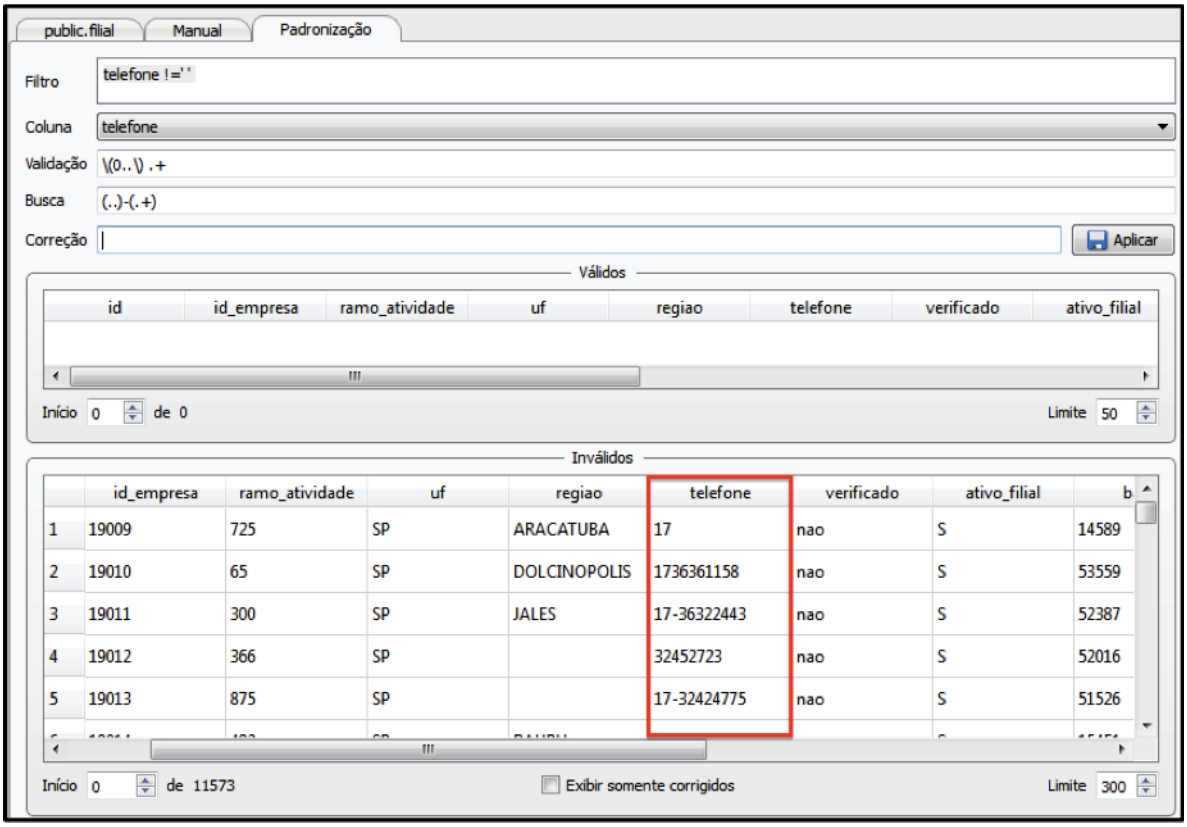

Figura 4.1: Números de Telefone de Empresas não Padronizados

Na figura 4.2 são mostradas as transformações realizadas em tempo-real do formato XX-XXXXXXXX para (0XX) XXXXXXXX, assinaladas em verde pelo sistema. Todas as tuplas da base foram automaticamente convertidas pelo novo padrão definido.

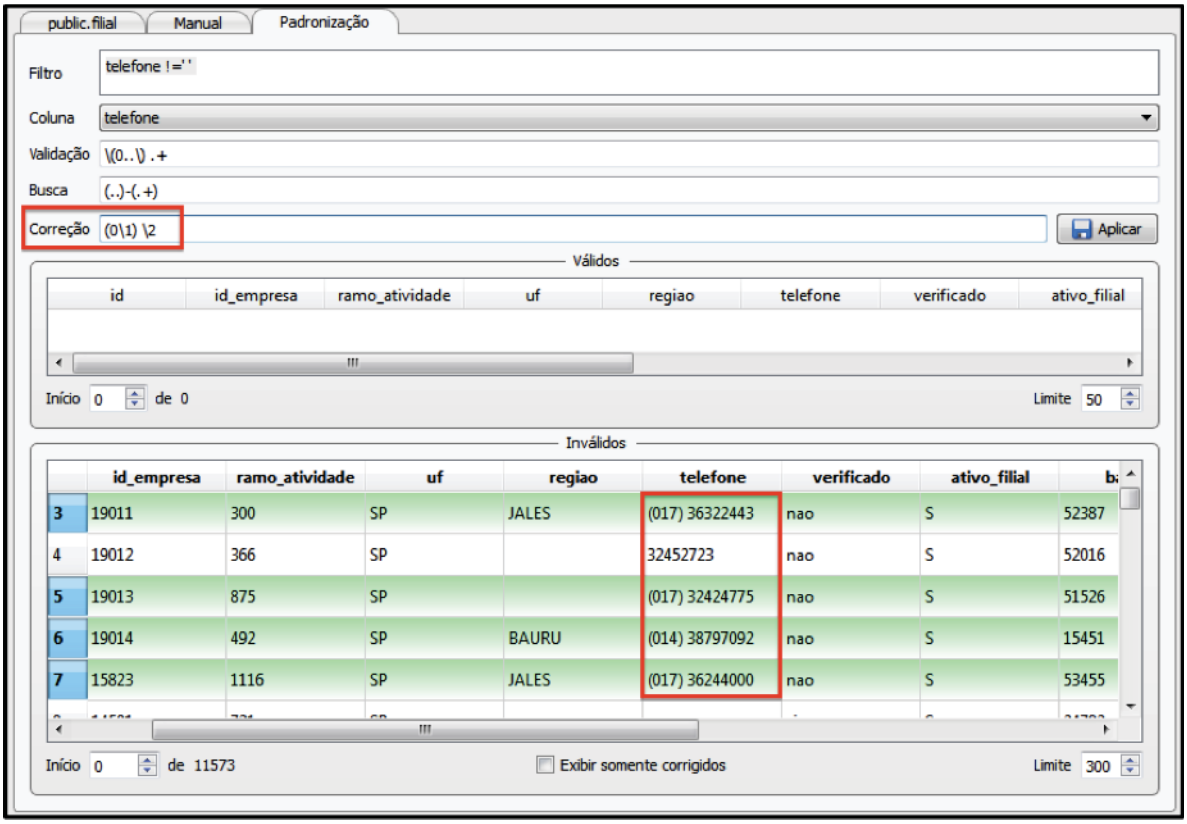

Figura 4.2: Números de Telefones Padronizados

Algoritmos de Detecção de Duplicatas Implementados

Foram realizados alguns experimentos com cada um dos algoritmos e técnicas de detecção de duplicatas implementados e incorporados ao ambiente (*edit distance*, Semelhança, Anagrama, e Q-gram Distance) a fim de demonstrar sua aplicabilidade.

Edit Distance

Ao utilizar o algoritmo *edit distance* para aplicação da limpeza dos dados na tabela "Ocupação do Acidentado", com distancia 1, foram detectados alguns casos de possíveis duplicações cujos termos são idênticos ou diferem em no máximo um caractere. Na figura 4.3 são mostrados alguns exemplos de duplicatas detectadas com a aplicação do algoritmo. Note que caracteres não alfanuméricos, como parênteses foram desconsiderados automaticamente no processo.

id	descricao	ativo
2516	AGENTE DE CONTRLOLE DE VETORES	S
4236	AGENTE DE CONTROLE DE VETORES	S

id	descricao	ativo
4164	OPERADOR(A) DE TELEMARKETING	S
2717	OPERADORA DE TELEMARKETING	S

id	descricao	ativo
4216	AUXILIIAR DE PRODUCAO	S
2849	AUXILIAR DE PRODUCAO	S

Figura 4.3: Alguns exemplos de duplicatas detectadas pelo algoritmo *edit distance*

Semelhança

Ao utilizar o algoritmo Semelhança para aplicação da limpeza dos dados na tabela *Ocupação do Acidentado*, com semelhança de 90%, foram detectados diversos casos de possíveis duplicações, alguns exemplificados na figura 4.4:

id	descricao	ativo
2717	OPERADORA DE TELEMARKETING	S
660	OPERADOR(A) TELEATENDENTE	S

id	descricao	ativo
3968	AUXILIAR DE MECANICO REFRIGERACAO	S
3262	AUXILIAR DE TECNICO(A) DE REFRIGERACAO	S

id	descricao	ativo
4160	ENCARREGADO(A) DE ACOUGUE	S
4542	ENCARREGADO DE ACOUGUE	S

Figura 4.4: Duplicatas detectadas pelo algoritmo Fuzzymatch com 90% de semelhança

Anagrama

Ao utilizar o algoritmo Anagrama para aplicação da limpeza dos dados na tabela *Ocupação do Acidentado* foram detectados alguns casos de possíveis duplicações que são exemplificados na figura 4.5.

id	descricao	ativo
2420	OPERADOR(A) DE DOBRADEIRA	S
1057	OPERADOR(A) DE BORDADEIRA	S

id	descricao	ativo
4150	AUXILIAR TECNICO(A)	S
4410	AUXILIAR TECNICO(A)	S

id	descricao	ativo
4532	DESOSSADOR	S
4003	DESSOSADOR	S

Figura 4.5: Exemplos de duplicatas detectadas por meio do algoritmo Anagrama

Q-Gram Distance

Ao utilizar o algoritmo *Q-Gram Distance* para aplicação da limpeza dos dados na tabela *Ocupação do Acidentado*, com q=3 e semelhança de 90%, foram identificados alguns dos casos de possíveis duplicações mostrados na figura 4.6.

id	descricao	ativo
2905	AJUDANTE DE MONTADOR	S
4552	AJUDANTE DE MONTADOR(A)	S

id	descricao	ativo
4540	ENCARREGADO(A)	S
4448	ENCARREGADO(A)	S

id	descricao	ativo
1625	REFILADOR(A)	S
4533	REFILADORA	S

Figura 4.6: Exemplos de duplicações detectadas por meio do algoritmo Q-Grams

Semi-automatização de Correção de Duplicatas

Ao aplicar o processo de semi-automatização da correção de duplicatas, a todo conjunto das possíveis duplicações detectadas na base de dados foi indicada ao usuário a

tupla que provavelmente era a correta e quais deviam ser eliminadas. São exemplificados na figura 4.7 alguns exemplos dessa funcionalidade. Note que a tupla com cor azul foi indicada como a possível correta dentre o conjunto detectado e essa indicação foi calculada baseada nas regras de semi-automatização de correção de duplicadas, descritas no capítulo 3.

id	descricao ▲	ativo
2905	AJUDANTE DE MONTADOR	S
4552	AJUDANTE DE MONTADOR(A)	S

Figura 4.7: Semi-automatização do processo de limpeza de dados

Semi-automatização de Correção de Duplicatas utilizando Banco de *Stopwords*

Ao utilizar o banco de *stopwords* no processo de semi-automatização de correções de duplicatas, independentemente do algoritmo escolhido, o sistema analisou todas as tuplas e removeu os termos que são considerados *stopwords*, contidos no banco de dados. Com isso, a eficácia da detecção aumentou significativamente, uma vez que termos que não caracterizam uma entidade não foram considerados durante as comparações entre os termos.

São exibidos na figura 4.8 alguns exemplos de duplicações que foram detectadas porque tiveram stopwords removidas, consolidando a eficácia do processo. O algoritmo escolhido foi o *Q-Gram Distance* com q=3, com 100% de semelhança, ou seja, somente tuplas idênticas deviam ser detectadas. Se o banco de stopwords não tivesse sido utilizado, o caso de duplicação não teria sido detectado.

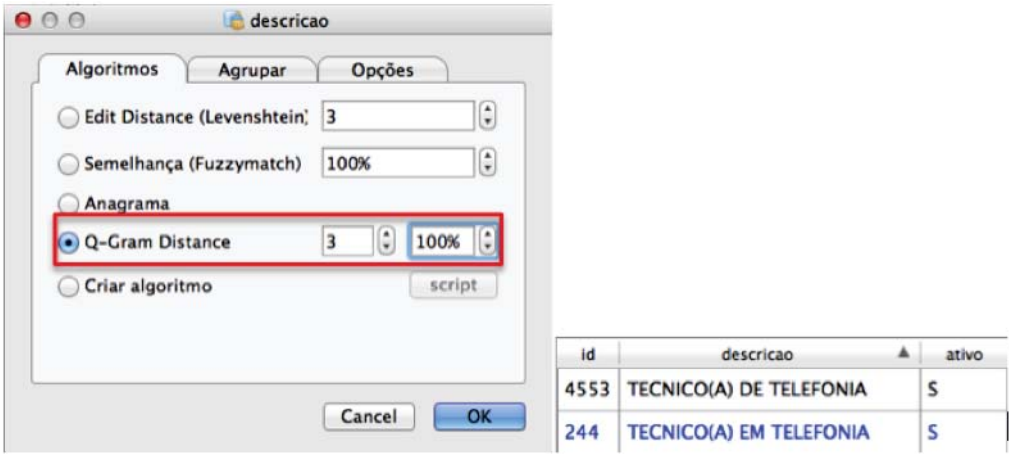

Figura 4.8: Semi-automatização de duplicatas utilizando banco de *stopwords*

Semi-automatização de Correção de Duplicatas utilizando Banco de Histórico

Ao utilizar o banco de treinamento no processo de semi-automatização de correções de duplicatas, independentemente do algoritmo escolhido, com base no histórico de todas as limpezas já realizadas, foi indicada ao usuário qual tupla dentre as sugeridas como possíveis duplicatas é a correta. é importante destacar que, um sistema treinado que evita a necessidade de se ter de analisar possíveis duplicidades já tratadas anteriormente pode gerar um ganho significativo de tempo no processo.

Na figura 4.9 são exibidos alguns resultados de testes realizados para tuplas detectadas baseadas no histórico de limpeza da base de dados.

id	nome	ativo
1141	MAQUINA DE COSTURA	S
1154	MAQUINA COSTURA	S

Figura 4.9: Sugestão de Tupla Correta baseada em Histórico

Semi-automatização de Correção de Duplicatas utilizando Banco de Sinônimos

Ao utilizar o banco de sinônimos no processo de detecção de duplicatas, o sistema

cobriu não somente as semelhanças físicas na análise, mas também o significado semântico dos termos que possibilita dessa maneira uma varredura mais profunda e, consequentemente, encontrar muito mais duplicações que não foram detectadas com quaisquer algoritmos e técnicas baseadas somente na correlação física entre os termos. Alguns resultados detectados são ilustrados na figura 4.10 e é mostrada a importância em se utilizar a semântica no processo de detecção. Com o banco de sinônimos carregado, todos os sinônimos definidos para cada termo também foi comparado durante o processo de detecção de semelhança e cobre um número significativamente maior de possibilidades que não seriam possíveis de serem detectadas levando em conta apenas sua semelhança física.

id	nome	ativo
203	ANIMAL	S
1150	ANIMAL PECONHENTO	S
1148	CAO	S
1147	ESCORPIAO	S
1149	GATO	S

Figura 4.10: Detecção de Duplicatas com base Semântica

Semi-automatização de Correção de Duplicatas por meio de Banco de Sinônimos como Banco Multi-idioma

São apresentados alguns resultados que comprovaram a eficácia na utilização do Banco de Sinônimos como Banco Multi-idiomas. O ambiente foi carregado com as bases de sinônimos e stopwords contendo palavras nos idiomas inglês e português. Os resultados demonstraram que as tuplas, mesmo em idiomas distintos, são detectadas como duplicatas. Esse processo possibilita também que bancos de dados integrados de fontes com idiomas distintos sejam limpos de forma a serem normalizados para um idioma padrão, ou mesmo ter suas informações traduzidas para um terceiro idioma.

Na figura 4.11 é exemplificada a detecção de palavras totalmente distintas fisicamente, mas semanticamente representavam um mesmo objeto, ainda escrita em idiomas diferentes. A palavra em azul é a sugerida pela ferramenta para que as demais sejam convertidas.

id	nome	ativo
1152	GATO	S
1155	DOG	S
1156	CAT	S
1151	CAO	S
1158	CACHORRO	S
1153	ANIMAL PECONHENTO	S
203	ANIMAL	S

Figura 4.11: Detecção de Duplicatas Multi-idioma

Semi-automatização de Correção de Duplicatas com Suporte de Buscadores Internet

Alguns experimentos também foram realizados para demonstrar a utilidade dos buscadores Internet contemplados pelo ambiente. é muito útil quando se está analisando as informações e ter que decidir pela tupla mais adequada ou correta se o contexto é desconhecido ou pouco dominado. Com as buscas simultâneas nos buscadores, foi possível ter uma base de informação adicional que suporta a escolha para realização da limpeza.

Na figura 4.12 é exemplificado um teste em que se foi útil ter suporte de buscadores Internet atrelado ao ambiente. Em tempo de execução, foi possível verificar os resultados das buscas dos termos detectados como possíveis duplicados e assim, tomar a decisão de qual tupla era a correta. Aplicando o processo de limpeza numa tabela que contém dados de empresas da região de São José do Rio Preto, foi possível decidir qual tupla é a correta pelos resultados da busca na Internet que foram realizados. Sem essa funcionalidade, é complicado decidir no momento da limpeza, qual dos resultados trazidos é o correto, comprometendo a eficiência do processo, pois a busca manual e desatrelada da ferramenta poderia se tornar demorada.

No exemplo, foram sugeridos como duplicatas os dados *Supermercado do Ponto* e *Supermercado Ponto Certo*. é difícil para um usuário que não tenha conhecimento dos nomes

de redes de supermercados saber se ambos estão corretos ou se um deles é proveniente de uma entrada de dados incorreta. O buscador Internet mostrou em tempo-real se o termo detectado existe, e ainda trouxe informações adicionais que auxiliaram na tomada de decisão.

Figura 4.12: Resultados da utilização do suporte de buscadores Internet no processo de limpeza

Semi-automatização de Correção de Duplicatas com Suporte de Agrupamento de Atributos

Além de todas as funcionalidades que puderam ser utilizadas de forma combinada para potencializar os resultados do processo de limpeza, são demonstrados alguns testes utilizando o agrupamento de atributos. Essa ferramenta foi bastante útil pois possibilitou encontrar duplicatas dentro de um contexto caracterizado por um ou mais atributos da mesma tupla.

No exemplo a seguir é demonstrada a aplicação de limpeza de bairros e municípios. Note que há municípios de estados diferentes com o mesmo nome, assim como bairros com nomes idênticos ou parecidos em uma mesma cidade. Para isso, pôde ser utilizado o agrupamento de atributos.

Para limpeza de municípios, fez-se necessário agrupar a limpeza por UF (estado), fazendo com que os resultados de possíveis duplicações detectados sejam efetivamente de um mesmo estado. A eficácia da funcionalidade é demonstrada por meio da figura 4.13 uma vez que são detectados municípios correspondentes de um mesmo estado.

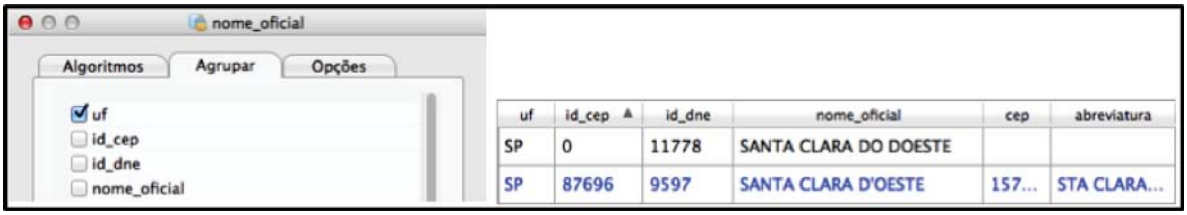

Figura 4.13: Detecção de Municípios Duplicados de um Mesmo Estado (UF)

Para limpeza de bairros, os tipos de atributo UF e Município foram agrupados, para ter certeza de que os possíveis bairros duplicados detectados são parte da mesma cidade. Os resultados podem ser conferidos na figura 4.14.

nome_oficial_municipio	id_cep	id_dne	nome_oficial
JABOTICABAL	108	20001	CONJUNTO HABITACIONAL HUGO LACORTE VITALLE II
JABOTICABAL	94	20000	CONJUNTO HABITACIONAL HUGO LACORTE VITALLE I

Figura 4.14: Detecção de Bairros Duplicados em um Mesmo Município

Automatização de Correção de Duplicatas

Com os testes realizados, o processo de limpeza de dados automatizado se comportou de forma análoga ao semi-automático, com a diferença de que, ao invés do usuário escolher a tupla correta e realizar a transformação dos dados baseado na sugestão indicada pelo Ambiente *Data Cleaning*, no processo automático não houve essa interação e toda a correção foi realizada de acordo com as regras do Critério de Certeza definido no capítulo anterior. Assim que toda base de dados foi percorrida, todos os casos detectados foram automaticamente corrigidos e, posteriormente, um relatório foi exibido ao usuário com todas as correções realizadas.

No teste realizado, exibido na figura 4.15, as correções foram realizadas e exibidas no

relatório final. A tupla considerada correta pelo Ambiente *Data Cleaning* é destacada na cor azul.

id	descricao	ativo
2905	AJUDANTE DE MONTADOR	S
4552	AJUDANTE DE MONTADOR(A)	S
^ ReferÃªncias		
4553	TECNICO(A) DE TELEFONIA	S
244	TECNICO(A) EM TELEFONIA	S
^ ReferÃªncias		
4150	AUXILIAR TECNICO(A)	S
4410	AUXILIAR TECNICO(A)	S
^ ReferÃªncias		
4540	ENCARREGADO(A)	S
4448	ENCARREGADO(A)	S
^ ReferÃªncias		
1625	REFILADOR(A)	S
4533	REFILADORA	S
^ ReferÃªncias		

Figura 4.15: Resultado da Limpeza Automatizada

4.4 Experimentos Comparativos

Nesta seção são apresentados e discutidos os resultados obtidos por meio de experimentos realizados do processo de limpeza de dados, com objetivo de comprovar a eficácia da limpeza com a utilização do ambiente desenvolvido neste trabalho. A tabela utilizada foi a de *Máquina e Ferramenta Causadora*, com um total de 639 registros. Uma análise manual foi realizada em todos os dados da tabela e constatou-se que havia 473 registros não duplicados, ou seja, aproximadamente 27% da base apresentava inconsistência ou dados duplicados.

É importante destacar que, diferentemente dos resultados apresentados pela maioria dos trabalhos encontrados na literatura, cujo propósito principal é destacar a eficácia das técnicas ao detectar casos de duplicatas válidas, é também apresentado neste trabalho o percentual de falsos-positivos, isto é, a taxa de duplicatas incorretas que foram detec-

tadas. O processo de limpeza de dados é custoso e se a quantidade de falsos-positivos for alta, a dificuldade para o usuário analisar e efetuar as correções necessárias aumenta significativamente.

Também vale ressaltar que a proposta de automatizar o processo de limpeza precisa prever o tratamento de falsos-positivos encontrados, uma vez que transformações incorretas poderiam gerar mais inconsistências na base de dados ao invés de oferecer um tratamento adequado de correções e limpeza.

Será utilizado o termo *eficácia* para denotar o percentual de tuplas detectadas como duplicadas em relação ao total de tuplas contida na tabela alvo do experimento.

Foram realizados experimentos aplicando a limpeza de dados manual / semi-automatizada e limpeza automatizada, utilizando o algoritmo de detecção de duplicatas Q-Gram, com parâmetro q=3, com os percentuais de semelhança 90%, 80%, 75%, 70% e 65%. Foi observado por meio de alguns testes preliminares que, acima de 90% de semelhança, não são detectados falsos-positivos e, abaixo de 65%, os resultados apresentados pelo Ambiente *Data Cleaning* já não são tão relevantes.

A aplicação do processo de limpeza para os experimentos realizados foram organizados em 6 etapas. A cada etapa fora adicionada uma funcionalidade ou técnica a fim de verificar a melhoria na detecção de inconsistências na base de dados.

Na primeira etapa, foi aplicado somente o algoritmo de detecção de duplicatas Q-Gram a fim de verificar o percentual de detecções que foram obtidas. Nas etapas seguintes, foram sendo adicionados (2) tratamento de caracteres especiais, normalização e *stemming*, (3) tratamento da fonética das palavras, (4) tratamento e remoção de *stopwords*, (5) sinônimos e banco multi-idiomas e na última etapa (6) foi aplicada a limpeza com a ferramenta treinada previamente. Em cada uma das etapas, também foi medido o percentual de falsos-positivos detectados.

4.4.1 Aplicação da Limpeza de Dados Manual / Semi-automatizada

Como pode ser observado pelos resultados exibidos na Tabela 4.1, com taxa de semelhança em 90%, 37 inconsistências foram detectadas com a aplicação do algoritmo Q-Gram, correspondendo à 22% do total de inconsistências e duplicatas presentes na base de dados. Ao serem adicionadas técnicas de refinamento, a quantidade de inconsistências apresentadas aumenta significativamente. é interessante notar que, com a utilização de sinônimos no processo de limpeza, a eficácia aumenta 71%, e 75% do total das inconsistências presentes são detectadas. Com o Ambiente *Data Cleaning* treinado, os percentuais subiram para 72% e 78%.

Q-Gram, q=3, 90%	Inconsistências	Falso Positivo	+ Eficácia	Eficácia
Algoritmo Ad-hoc	37	0	0%	22%
Caracteres Especiais	37	0	0%	22%
Fonética	42	0	12%	25%
Stopwords	58	0	36%	35%
Sinônimos	127	2	71%	75%
Treinamento	130	0	72%	78%

Tabela 4.1: Resultados obtidos com a limpeza utilizando o algoritmo Q-Grams, com q=3 e semelhança 90%

Com semelhança de 90%, praticamente não são detectados falsos-positivos, porém nem todas as inconsistências presentes na base de dados são encontrados pela ferramenta.

Já com fator de semelhança de 80%, como mostrado na Tabela 4.2, 53 inconsistências foram detectadas com a aplicação do algoritmo Q-Gram, porém 6 delas falsos-positivos. A diferença, ou seja, 47 casos correspondem à 28% do total de inconsistências e duplicatas presentes na base de dados. Ao serem adicionadas técnicas de refinamento, a quantidade de inconsistências apresentadas aumenta significativamente, representando um total de 139 casos detectados e a quantidade de falsos-positivos diminui. Com a utilização de sinônimos no processo de limpeza, a eficácia aumenta 62%, e 80% do total das inconsistências presentes são detectadas. Com o Ambiente *Data Cleaning* treinado, a eficácia atinge 83%.

Q-Gram, q=3, 80%	Inconsistências	Falso Positivo	+ Eficácia	Eficácia
Algoritmo Ad-hoc	53	6	0%	28%
Caracteres Especiais	54	5	2%	29%
Fonética	58	4	9%	32%
Stopwords	73	3	27%	42%
Sinônimos	139	5	62%	80%
Treinamento	139	0	62%	83%

Tabela 4.2: Resultados obtidos com a limpeza utilizando o algoritmo Q-Grams, com q=3 e semelhança 80%

Ressalta-se também que, com a ferramenta treinada, não foram apresentados falsos positivos.

Com fator de semelhança de 75%, como exibido na Tabela 4.3, 65 inconsistências foram detectadas com a aplicação do algoritmo Q-Gram, porém 8 delas falsos-positivos. A diferença, ou seja, 57 casos correspondem à 34% do total de inconsistências e duplicatas presentes na base de dados. Ao serem adicionadas técnicas de refinamento, a quantidade de inconsistências apresentadas aumenta ainda mais, representando um total de 143 casos detectados e a quantidade de falsos-positivos também diminui. Com a utilização de sinônimos no processo de limpeza, a eficácia aumenta 56%, e 83% do total das inconsistências presentes são detectadas. Com o Ambiente *Data Cleaning* treinado, a eficácia atinge 86%.

Q-Gram, q=3, 75%	Inconsistências	Falso Positivo	+ Eficácia	Eficácia
Algoritmo Ad-hoc	65	8	0%	34%
Caracteres Especiais	67	9	3%	35%
Fonética	70	9	7%	37%
Stopwords	83	8	22%	45%
Sinônimos	147	9	56%	83%
Treinamento	143	0	55%	86%

Tabela 4.3: Resultados obtidos com a limpeza utilizando o algoritmo Q-Grams, com q=3 e semelhança 75%

Assim como o experimento com fator de semelhança 80%, com a ferramenta treinada, não foram apresentados falsos positivos. Com percentual de semelhança de 70%, como

ilustrado na Tabela 4.4, 74 inconsistências foram detectadas com a aplicação do algoritmo Q-Gram, porém 11 delas falsos-positivos. A diferença, ou seja, 63 casos correspondem à 38% do total de inconsistências e duplicatas presentes na base de dados. Ao serem adicionadas técnicas de refinamento, a quantidade de inconsistências apresentadas se eleva, representando um total de 143 casos detectados e a quantidade de falsos-positivos também diminui. Com a utilização de sinônimos no processo de limpeza, a eficácia aumenta 50%, e 83% do total das inconsistências presentes são detectadas. Com o Ambiente *Data Cleaning* treinado, a eficácia atinge 86%.

Q-Gram, q=3, 70%	Inconsistências	Falso Positivo	+ Eficácia	Eficácia
Algoritmo Ad-hoc	74	11	0%	38%
Caracteres Especiais	75	11	1%	38%
Fonética	78	10	5%	41%
Stopwords	85	11	13%	44%
Sinônimos	148	10	50%	83%
Treinamento	143	0	48%	86%

Tabela 4.4: Resultados obtidos com a limpeza utilizando o algoritmo Q-Grams, com q=3 e semelhança 70%

Assim como os experimentos com fatores de semelhança 90%, 80% e 75%, com a ferramenta treinada, não foram apresentados falsos positivos.

Por fim, com percentual de semelhança de 65%, verificado na Tabela 4.5, 94 inconsistências foram detectadas com a aplicação do algoritmo Q-Gram, porém 14 delas falsos-positivos. A diferença, ou seja, 80 casos correspondem à 48% do total de inconsistências e duplicatas presentes na base de dados. Ao serem adicionadas técnicas de refinamento, a quantidade de inconsistências apresentadas se eleva, representando um total de 153 casos detectados, mas a quantidade de falsos-positivos acaba aumentando até que a ferramenta estivesse treinada. Com a utilização de sinônimos no processo de limpeza, a eficácia aumenta 42%, e 84% do total das inconsistências presentes são detectadas. Com o Ambiente *Data Cleaning* treinado, a eficácia atinge 92%.

Q-Gram, q=3, 65%	Inconsistências	Falso Positivo	+ Eficácia	Eficácia
Algoritmo Ad-hoc	94	14	0%	48%
Caracteres Especiais	97	21	3%	46%
Fonética	96	17	2%	47%
Stopwords	103	19	9%	50%
Sinônimos	161	20	42%	84%
Treinamento	153	0	39%	92%

Tabela 4.5: Resultados obtidos com a limpeza utilizando o algoritmo Q-Grams, com q=3 e semelhança 65%

Destacam-se nesses primeiros experimentos o alto grau de eficácia que o Ambiente *Data Cleaning* desenvolvido apresenta no processo de limpeza manual / semi-automatizada e o tratamento adequado de falsos-positivos que não apresentados ao usuário para sua análise.

4.4.2 Aplicação da Limpeza de Dados Automatizada

O próximo conjunto de experimentos aborda o processo de limpeza automatizada de dados, ou seja, não houve interação do usuário tanto na etapa de análise quanto de transformação dos dados.

Como pode ser observado pelos resultados exibidos na Tabela 4.6, com taxa de semelhança em 90%, 34 inconsistências foram detectadas com a aplicação do algoritmo Q-Gram, correspondendo à 20% do total de inconsistências e duplicatas presentes na base de dados. Ao serem adicionadas técnicas de refinamento, a quantidade de inconsistências apresentadas aumenta significativamente. é interessante notar que, da mesma maneira com o processo manual, com a utilização de sinônimos no processo de limpeza automatizada, a eficácia aumenta 58%, e 54% do total das inconsistências presentes são detectadas. Com o Ambiente *Data Cleaning* treinado, os percentuais subiram para 69% e 74%.

Q-Gram, q=3, 90%	Inconsistências	Falso Positivo	+ Eficácia	Eficácia
Algoritmo Ad-hoc	34	0	0%	20%
Caracteres Especiais	39	0	13%	23%
Fonética	39	0	13%	23%
Stopwords	55	0	38%	33%
Sinônimos	92	1	58%	54%
Treinamento	124	0	69%	74%

Tabela 4.6: Resultados obtidos com a limpeza automatizada utilizando o algoritmo Q-Grams, com q=3 e semelhança 90%

Com semelhança de 90%, praticamente não são detectados falsos-positivos, porém nem todas inconsistências presentes na base de dados são encontradas pela ferramenta.

Já com fator de semelhança de 80%, como mostrado na Tabela 4.7, 51 inconsistências foram detectadas com a aplicação do algoritmo Q-Gram, porém 5 delas falsos-positivos. A diferença, ou seja, 46 casos correspondem à 28% do total de inconsistências e duplicatas presentes na base de dados. Ao serem adicionadas técnicas de refinamento, a quantidade de inconsistências apresentadas aumenta significativamente, representando um total de 133 casos detectados e a quantidade de falsos-positivos diminui. Com a utilização de sinônimos no processo de limpeza, a eficácia aumenta 47%, e 60% do total das inconsistências presentes são detectadas. Com o Ambiente *Data Cleaning* treinado, a eficácia atinge 80%.

Q-Gram, q=3, 80%	Inconsistências	Falso Positivo	+ Eficácia	Eficácia
Algoritmo Ad-hoc	51	5	0%	28%
Caracteres Especiais	55	4	7%	31%
Fonética	55	4	7%	31%
Stopwords	70	3	27%	40%
Sinônimos	104	3	47%	60%
Treinamento	133	0	59%	80%

Tabela 4.7: Resultados obtidos com a limpeza automatizada utilizando o algoritmo Q-Grams, com q=3 e semelhança 80%

Ressalta-se também que, com a ferramenta treinada, não foram apresentados falsos positivos.

Com fator de semelhança de 75%, como exibido na Tabela 4.8, 61 inconsistências foram detectadas com a aplicação do algoritmo Q-Gram, porém 9 delas falsos-positivos. A diferença, ou seja, 52 casos correspondem à 31% do total de inconsistências e duplicatas presentes na base de dados. Ao serem adicionadas técnicas de refinamento, a quantidade de inconsistências apresentadas aumenta ainda mais, representando um total de 136 casos detectados e a quantidade de falsos-positivos também diminui. Com a utilização de sinônimos no processo de limpeza, a eficácia aumenta 42%, e 62% do total das inconsistências presentes são detectadas. Com o Ambiente *Data Cleaning* treinado, a eficácia atinge 81%.

Q-Gram, q=3, 75%	Inconsistências	Falso Positivo	+ Eficácia	Eficácia
Algoritmo Ad-hoc	61	9	0%	31%
Caracteres Especiais	64	9	5%	33%
Fonética	64	9	5%	33%
Stopwords	77	8	21%	41%
Sinônimos	111	7	42%	62%
Treinamento	136	0	53%	81%

Tabela 4.8: Resultados obtidos com a limpeza automatizada utilizando o algoritmo Q-Grams, com q=3 e semelhança 75%

Assim como o experimento com fator de semelhança 80%, com a ferramenta treinada, não foram apresentados falsos positivos.

Com percentual de semelhança de 70%, como ilustrado na Tabela 4.9, 66 inconsistências foram detectadas com a aplicação do algoritmo Q-Gram, porém 9 delas falsos-positivos. A diferença, ou seja, 57 casos correspondem à 34% do total de inconsistências e duplicatas presentes na base de dados. Ao serem adicionadas técnicas de refinamento, a quantidade de inconsistências apresentadas se eleva, representando um total de 136 casos detectados e a quantidade de falsos-positivos também diminui. Com a utilização de sinônimos no processo de limpeza, a eficácia aumenta 39%, e 60% do total das inconsistências presentes são detectadas. Com o Ambiente *Data Cleaning* treinado, a eficácia atinge 81%.

Q-Gram, q=3, 70%	Inconsistências	Falso Positivo	+ Eficácia	Eficácia
Algoritmo Ad-hoc	66	9	0%	34%
Caracteres Especiais	68	10	3%	35%
Fonética	68	10	3%	35%
Stopwords	77	9	14%	41%
Sinônimos	111	10	39%	60%
Treinamento	136	0	50%	81%

Tabela 4.9: Resultados obtidos com a limpeza automatizada utilizando o algoritmo Q-Grams, com q=3 e semelhança 70%

Assim como os experimentos com fatores de semelhança 90%, 80% e 75%, com a ferramenta treinada, não foram apresentados falsos positivos.

Por fim, com percentual de semelhança de 65%, verificado na Tabela 4.10, 79 inconsistências foram detectadas com a aplicação do algoritmo Q-Gram, porém 16 delas falsos-positivos. A diferença, ou seja, 63 casos correspondem à 38% do total de inconsistências e duplicatas presentes na base de dados. Ao serem adicionadas técnicas de refinamento, a quantidade de inconsistências apresentadas se eleva, representando um total de 147 casos detectados, mas a quantidade de falsos-positivos acaba aumentando até que a ferramenta estivesse treinada. Com a utilização de sinônimos no processo de limpeza, a eficácia aumenta 33%, e 62% do total das inconsistências presentes são detectadas. Com o Ambiente *Data Cleaning* treinado, a eficácia atinge 88%.

Q-Gram, q=3, 65%	Inconsistências	Falso Positivo	+ Eficácia	Eficácia
Algoritmo Ad-hoc	79	16	0%	38%
Caracteres Especiais	80	14	1%	40%
Fonética	80	14	1%	40%
Stopwords	87	14	9%	44%
Sinônimos	119	15	33%	62%
Treinamento	147	0	46%	88%

Tabela 4.10: Resultados obtidos com a limpeza automatizada utilizando o algoritmo Q-Grams, com q=3 e semelhança 65%

Mesmo com o processo de limpeza executado automaticamente sem interação do usuá-

rio na análise dos dados, destacam-se o alto grau de eficácia que o Ambiente *Data Cleaning* desenvolvido apresenta, cobrindo aproximadamente 90% do total de inconsistências da base de dados, e o tratamento adequado de falsos-positivos, pois nenhuma limpeza foi realizada incorretamente, com o Ambiente *Data Cleaning* treinado.

4.4.3 Análise e Comparação dos Resultados Obtidos

São exibidos nos próximos gráficos comparativos dos resultados obtidos pela aplicação das limpezas de dados com abordagens manuais e automatizadas. Devido às regras estabelecidas pelo Critério de Certeza utilizado na abordagem automática, alguns casos em que não houve desempate não foram limpos automaticamente e, por isso, a eficácia da cobertura total de casos inconsistentes tratados é um pouco menor.

Pode ser observado na figura 4.16 os percentuais de eficácia desde a aplicação somente do algoritmo, à esquerda, até a execução do processo com todos os recursos disponíveis, como tratamento de caracteres especiais, fonética, stopwords, sinônimos multi-idioma e treinamento. A limpeza automatizada teve uma eficácia somente 5% menor que a automatizada quando a semelhança do algoritmo aplicado era 90%.

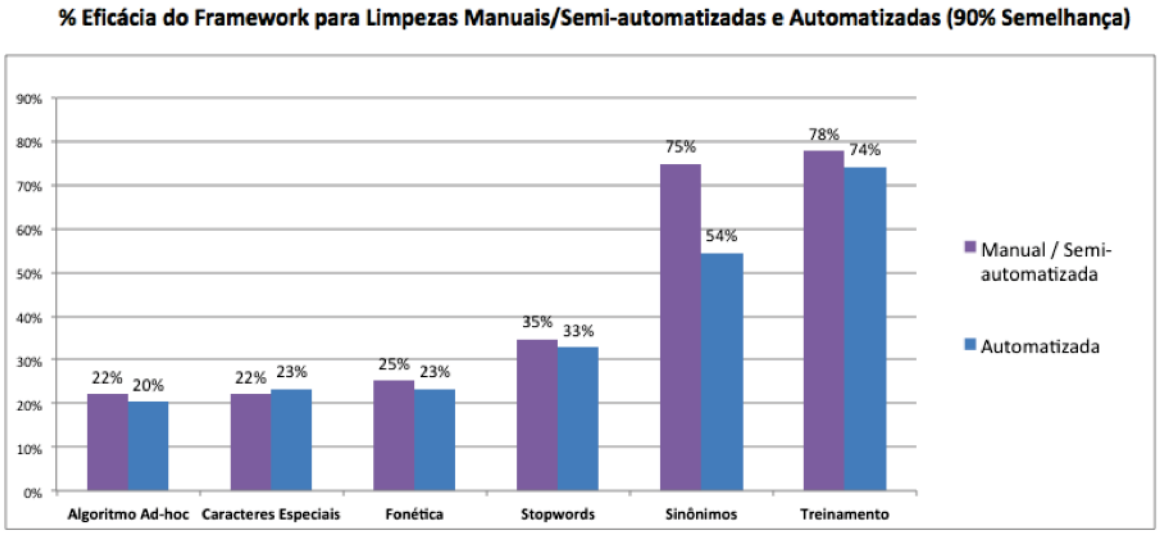

Figura 4.16: Percentuais de Eficácia do Ambiente *Data Cleaning* para Limpezas Manuais e Automatizadas com 90% de Semelhança

Como ilustrado na figura 4.17, com semelhança do algoritmo em 80%, a limpeza automatizada com a ferramenta treinada teve uma eficácia somente 3,5% menor que a manual.

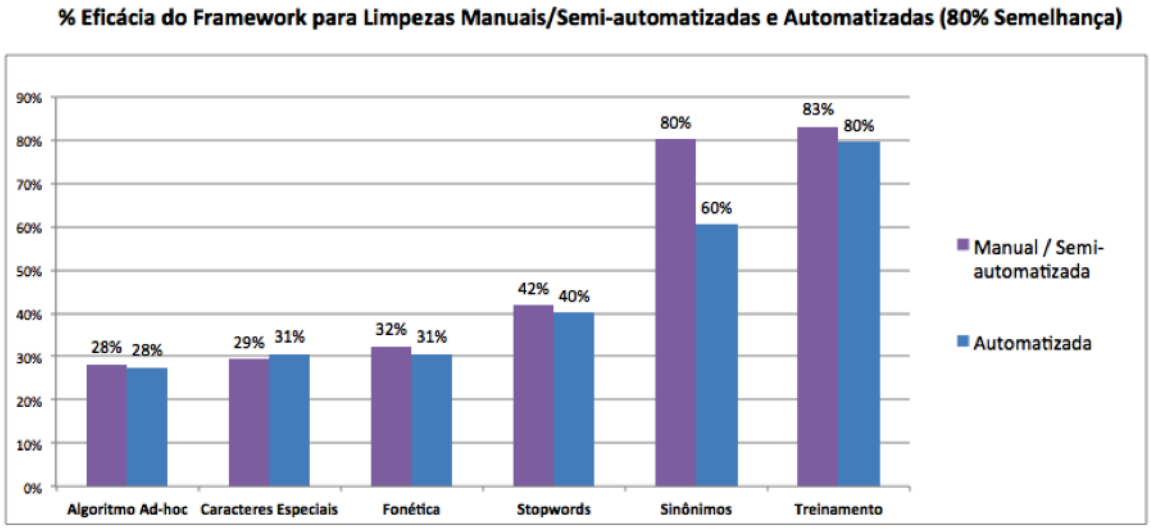

Figura 4.17: Percentuais de Eficácia do Ambiente *Data Cleaning* para Limpezas Manuais e Automatizadas com 80% de Semelhança

Observa-se nas figuras 4.18 e 4.19, com semelhanças do algoritmo em 70% e 65%, que a limpeza automatizada com a ferramenta treinada teve uma eficácia de aproximadamente 6% menor que a manual.

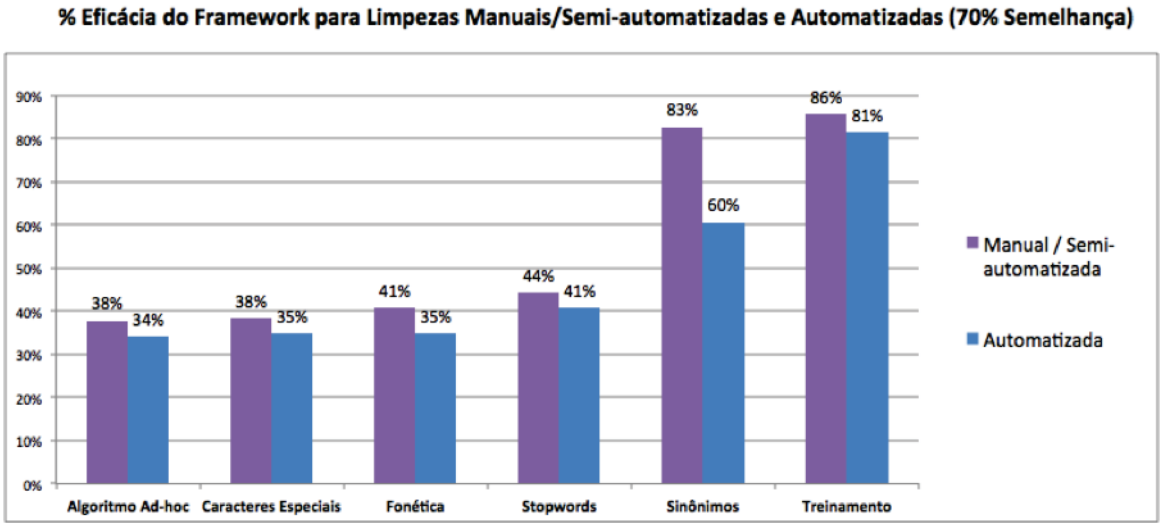

Figura 4.18: Percentuais de Eficácia do Ambiente *Data Cleaning* para Limpezas Manuais e Automatizadas com 70% de Semelhança

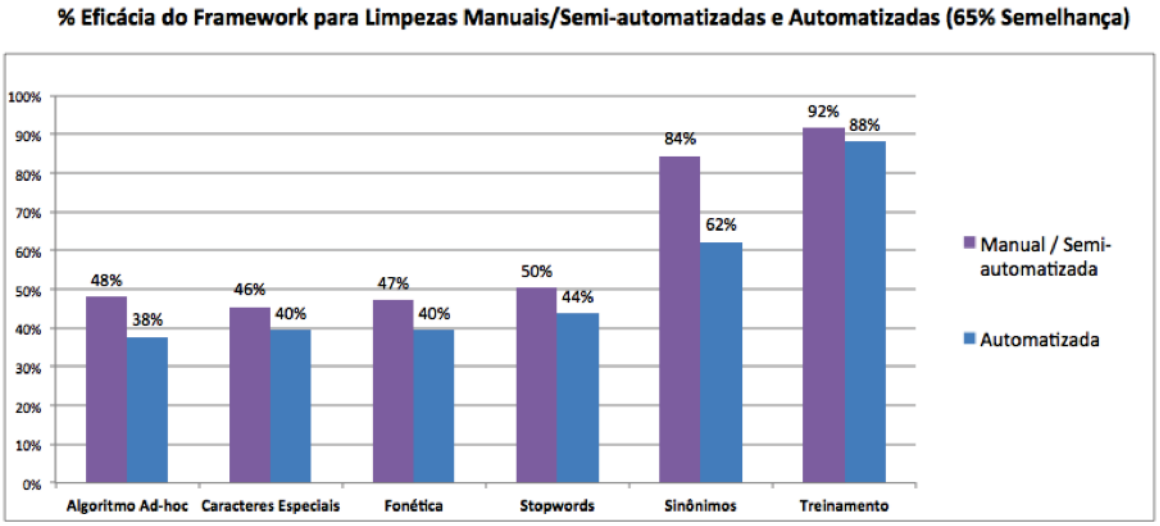

Figura 4.19: Percentuais de Eficácia do Ambiente *Data Cleaning* para Limpezas Manuais e Automatizadas com 65% de Semelhança

O processo automatizado de limpeza mostra-se eficiente se comparado com o processo manual de análise, detecção e tratamento de problemas de inconsistências e duplicatas em bases de dados, porém é importante observar que só é possível automatizar o processo quando há ferramentais que suportam o tratamento de casos de falsos-positivos. Os percentuais de falsos-positivos exibidos nos gráficos das figuras 4.20 e 4.21 demonstram

que, somente com uma abordagem inteligente, é possível tratar esses casos e, assim, evitar que inconsistências sejam apresentadas ao usuário, no caso de uma abordagem manual, ou sejam indevidamente transformadas, no caso de abordagem automatizada.

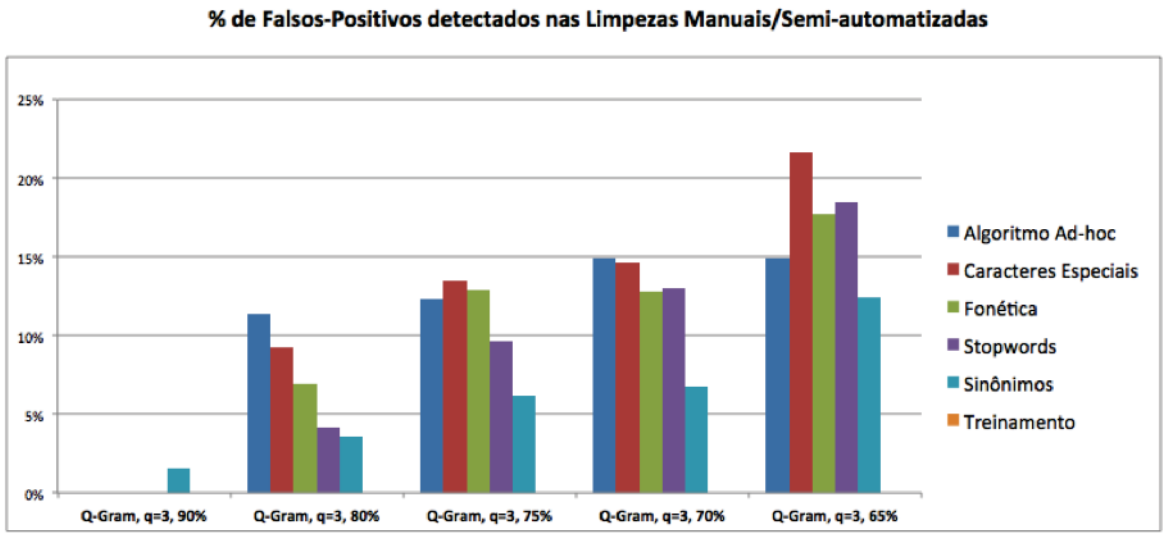

Figura 4.20: Percentual de Falsos-Positivos Detectados no Processo de Limpeza Manual

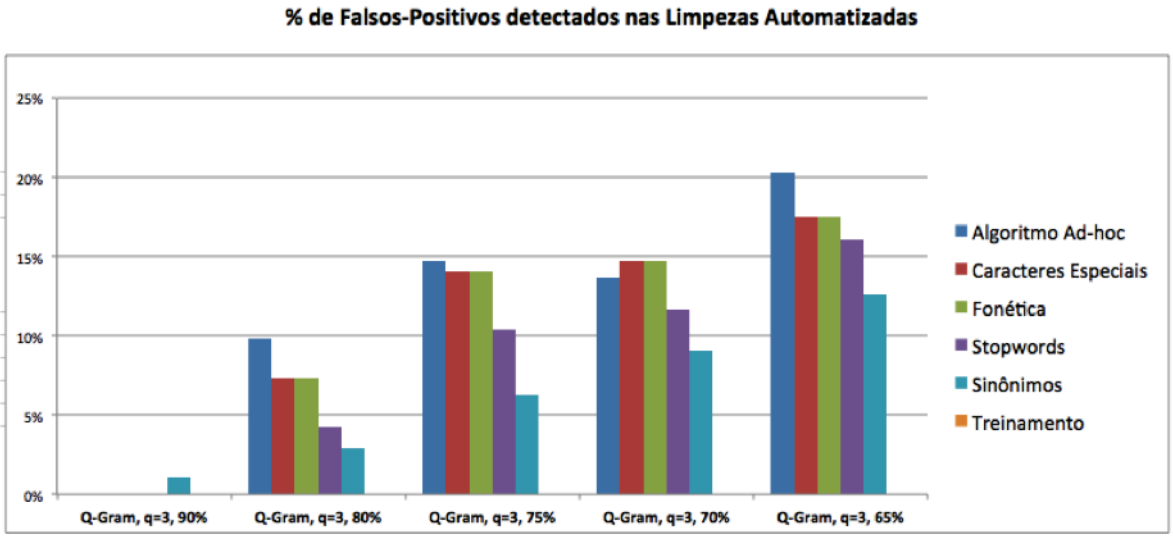

Figura 4.21: Percentual de Falsos-Positivos Detectados no Processo de Limpeza Automatizado

Dependendo do fator de semelhança utilizado na aplicação de um dos algoritmos para detecção de duplicatas, o percentual de falsos-positivos pode ultrapassar 20%. Porém, em todos os experimentos realizados em que todas as técnicas presentes nos módulos

do Ambiente *Data Cleaning* desenvolvido foram aplicadas simultaneamente, incluindo a abordagem inteligente para que a ferramenta estivesse treinada, a taxa de falsos-positivos apresentadas foi 0%, garantindo segurança ao usuário quanto a transformações incorretas das informações presentes nas bases de dados em que o Ambiente *Data Cleaning* foi aplicado.

4.5 Considerações Finais

Neste capítulo foram apresentados testes e experimentos realizados com o Ambiente *Data Cleaning* desenvolvido. Inicialmente foram mostrados os resultados dos testes realizados a fim de comprovar a eficácia de todas as funcionalidades desenvolvidas, pertinentes a cada módulo da arquitetura proposta.

Posteriormente foram apresentados e discutidos experimentos realizados em diversos cenários a fim de verificar a eficácia da ferramenta. Com um algoritmo devidamente calibrado e os módulos propostos e desenvolvidos utilizados em conjunto, a eficácia da ferramenta é significativa e cobre aproximadamente 90% do total de inconsistências presentes na base de dados, com percentual de casos de falsos-positivos a 0%. Os trabalhos da área levantados na literatura apresentam somente parte das funcionalidades e tratamentos em comparação com o ambiente desenvolvido que, de acordo com os resultados demonstrados, a eficácia não passa de 50%.

Destaca-se também que o objetivo dos experimentos foi demonstrar ferramentais e abordagens não somente para detectar e tratar casos positivos de inconsistências e duplicidades de informações, mas também abordar casos de falsos-positivos detectados e analisar os impactos negativos que representam no processo de limpeza de dados, seja ele manual ou automatizado.

É importante enfatizar que essa perspectiva de análise ainda não é fortemente discutida na literatura mesmo sendo essencial ser considerada com o intuito de que resultados cada vez mais eficazes sejam obtidos.

5 Conclusões

5.1 Considerações Finais

O armazenamento de informações tem se tornado cada vez maior e mais frequente, uma vez que dados podem conter informações valiosas, tornando-se essencial para o avanço da ciência moderna e da tecnologia a busca por novas técnicas de manipulação, extração, armazenamento, recuperação e correlação de informações, uma vez que novos conhecimentos são obtidos de conhecimentos prévios.

Para que seja possível obter informações verossímeis de todas as fontes possíveis, é necessário garantir que os dados analisados sejam integrados de forma consistente e correspondam à realidade da informação. Essa garantia só é possível se houver um pré-processamento dos dados; etapa em que as informações armazenadas em diferentes fontes são integradas e posteriormente analisadas, a fim de se detectar inconsistências, duplicidades física ou semântica, normalizações e correções, etc.

A Limpeza de Dados mostra-se como essencial e a mais importante etapa para se obter conhecimento consistente e em seu universo estão presentes técnicas, algoritmos e ferramentas que ainda não atendem às necessidades do mundo real e não oferecem soluções eficazes e eficientes.

Ainda há poucos estudos que comparam a eficácia das diferentes técnicas de limpeza de dados e os poucos trabalhos focados nessas análises são insuficientes, uma vez que a maioria das técnicas e soluções propostas até o momento requerem uma interação muito grande com o usuário para decidir e analisar as detecções realizadas e, no que se refere

a grandes bases de dados, é humanamente inviável a interação direta no processo de limpeza que facilmente contém milhões de registros. Essa interação deve-se principalmente ao poder de decisão semântico, pois as técnicas e trabalhos realizados até então focam especificamente em detecção de semelhança física entre os textos e entidades.

O trabalho desenvolvido possui, dentre seus objetivos, diminuir a interação do usuário no processo de análise e correção de inconsistências e duplicidades. Foi construído um Ambiente *Data Cleaning* que oferece uma coleção de ferramentais configuráveis de limpeza de dados automatizada baseada em treinamentos e semelhança físico-semântica de dados que oferece uma solução mais eficaz e extensível, abordando algumas lacunas pouco ou não exploradas na literatura como treinamento, suporte semântico, suporte multi-idioma e automatização parcial ou total do processo de limpeza.

Os resultados dos testes realizados comprovaram a eficácia de todas as funcionalidades desenvolvidas, pertinentes a cada módulo da arquitetura proposta e os experimentos realizados demonstraram, em diversos cenários, a eficácia da ferramenta. Com um algoritmo devidamente calibrado e os módulos propostos e desenvolvidos utilizados em conjunto, a eficácia é significativa e cobre aproximadamente 90% do total de inconsistências presentes na base de dados, com percentual de casos de falsos-positivos 0%. Os trabalhos da área apresentam somente parte das funcionalidades e tratamentos em comparação com o Ambiente *Data Cleaning* desenvolvido que, de acordo com os resultados demonstrados, sua eficácia não passa de 50%.

Também foram demonstradas abordagens que, além de detectar e tratar casos positivos de inconsistências e duplicidades de informações, abordam casos de falsos-positivos detectados e consideram os impactos negativos que representam no processo de limpeza de dados, seja ele manual ou automatizado, ainda não discutida fortemente na literatura.

As contribuições mais significativas desse trabalho referem-se à ferramenta desenvolvida que, automaticamente sem necessidade da interação do usuário, é capaz de analisar e eliminar 90% das inconsistências e duplicidades de informações presentes numa base de

dados, com não ocorrência de casos de falsos-positivos. Além disso, o Ambiente *Data Cleaning* é extensível e portável, permitindo que facilmente seja melhorado e incrementado com novos algoritmos, técnicas, dicionários, idiomas, etc.

5.1.1 Análise de Cobertura do Ambiente *Data Cleaning* Desenvolvido em Comparação com Demais Trabalhos Publicados

É sintetizada na figura 5.1 a cobertura de funcionalidades, técnicas e abordagens para o processo de limpeza de dados de diversas ferramentas e *frameworks* disponíveis na literatura e no mercado. É válido ressaltar que o Ambiente *Data Cleaning* desenvolvido aborda diversos aspectos ainda pouco explorados pelos trabalhos presentes no estado da arte, confirmando assim sua contribuição junto as pesquisas relacionadas.

		2012			2011					2010					2009			2008		2007	2002	
	***Framework* Desenvolvido**	WHIRL (WHIRL, 2012)	Project5 Flamingo (FLAMINGO, 2012)	Febrl Syste3 (FEBRL, 2012)	PSI (ANDRADE et al., 2011	(BERTI-EQUILLE et. al., 2011)	(PRASAD et al.; 2011)	(CHATURVEDI et al.; 2011)	(BERTOSSI et al.; 2011)	Database Cleaner (VALENCIO et al; 2010)	(HUANZHUO et al.; 2010)	(YANHUA; SHUYU, 2010)	(ALI; WARRAICH, 2010)	(WANG; 2010)	SmartClean (OLIVEIRA et al.; 2009)	(HUNG et al.; 2009)	(YUBIN et al.; 2009)	(YAN; DIAO; LI; 2008)	(CISZAK; 2008)	WizSame (ELMAGARMID et al.; 2007)	TAILOR (ELFEKY et al.; 2002)	BigMatch (YANCEY, 2002)
Detecção de Duplicatas	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Detecção e Transformação	✓	✓	✓	✗	✗	✓	✗	✓	✗	✓	✓	✓	✓	✗	✓	✗	✓	✓	✗	✗	✓	✗
3 ou mais Algoritmos Ad-Hoc	✓	✗	✗	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗	✗	✓	✓
Tratamento de Caracteres Especiais	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✗	✓	✓	✗	✓	✓	✓
Fonética	✓	✗	✗	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗
Fonética Multi-Idioma	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Stopwords	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Stopwords Multi-Idioma	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Semântica	✓	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
Semântica Multi-Idioma	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Abordagem Inteligente Treinamento	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Limpeza Manual	✓	✓	✓	✗	✗	✓	✗	✓	✗	✓	✓	✓	✓	✗	✓	✗	✓	✓	✗	✗	✓	✗
Limpeza Semi-Automatizada	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Limpeza Automatizada	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Extensibilidade	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗
Portabilidade	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗

Figura 5.1: Comparativo de Funcionalidades Contempladas por Diversas Ferramentas de Limpeza de Dados

5.1.2 Sugestões de Trabalhos Futuros

A fim de incentivar a continuidade no desenvolvimento de soluções para o processo de limpeza de dados e melhorar a eficácia e eficiência da ferramenta desenvolvida, são sugeridas algumas frentes a serem exploradas em trabalhos futuros:

1. **Eficiência**: Tão importante quanto a eficácia, a eficiência das ferramentas de limpeza de dados é um fator crucial para sua real aplicabilidade, uma vez que a quantidade de dados a serem analisados e tratados é incalculável. Ainda há poucos trabalhos que apresentem soluções de melhoria de desempenho dos ferramentais de limpeza suportadas por novas arquiteturas como *multi-threading*, computação em nuvem e placas de vídeo;

2. **Múltiplas Fontes de Dados**: Informações armazenadas na Internet, banco de dados distribuídos, banco de dados em nuvem e novas arquiteturas de armazenamento de dados já estão presentes nos dias atuais, alimentando a necessidade de que estudos e propostas para limpeza de dados armazenados nesses tipos de repositórios sejam mais explorados;

3. **Limpeza de Dados não Convencionais**: Pouco tem sido desenvolvido e proposto no âmbito limpeza de dados não convencionais e essa frente foi pouco atacada;

4. **Abordagens Inteligentes**: é essencial que sejam desenvolvidas e propostas novas abordagens inteligentes de análise e transformação de dados com objetivo de possibilitar maior autonomia às ferramentas de limpeza de dados e evitar ao máximo a interação com usuários.

Referências

[1] ALI, K.; WARRAICH, M. A. **A framework to implement Data Cleaning in Enterprise Data Warehouse for Robust Data Quality.** In IEEE, 2010, p. 1-6.

[2] ANDRADE, T. L.; SOUZA, R. C. G.; BABINI, M.; VALENCIO, C. R. **Optimization of Algorithm to Identification of Duplicate Tuples Through Similarity Phonetic Based on Multithreading.** Parallel and Distributed Computing, Applications and Technologies (PDCAT) 2011. 12th International Conference on, 2011, p. 299-304.

[3] ARASU, A.; KAUSHIK, R. **A Grammar-based Entity Representation Framework for Data Cleaning.** SIGMOD'09 ACM. 2009, p. 233-244.

[4] BERTI-EQUILLE, L.; DASU, T.; SRIVASTAVA, D. **Discovery of Complex Glitch Patterns: A Novel Approach to Quantitative Data Cleaning.** In: ICDE Conference 2011, IEEE, p. 733-744.

[5] BERTOSSI, L.; KOLAHI, S.; LAKSHMANAN, L. V. S. **Data Cleaning and Query Answering with Matching Dependencies and Matching Functions.** In: ICDT 2011, ACM 2011, p. 268-279.

[6] BOHANNON, P.; FAN, W.; GEERTS, F.; JIA, X.; KEMENTSIETSIDIS, A. **Conditional Functional Dependencies for Data Cleaning.** Journal ACM Transactions on Data Systems (TODS), 2008. Vol. 33, p. 746-755.

[7] CHATURVEDI, S.; FARUQUIE, T. A.; VENKATACHALIAH, G.; PADMANABHAN, S. **Optimal Training Data Selection for Rule-based Data cleansing Models.** In: 2011 Annual SRII Global Conference. IEEE Computer Society, 2010, p.126-134.

[8] CISZAK, L. **Application of Clustering and Association Methods in Data Cleaning.** In: Proceedings of the International Multiconference on Computer Science and Information Technology. IEEE, 2008, p. 97-103.

[9] COHEN, W. W. **Integration of Heterogeneous Databases without Common Domains Using Queries Based on Textual Similarity.** Proc. 1998 ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '98), 1998. P. 201-212.

[10] CPLUSPLUS C++ Language Tutorial, 2012. Disponível em <http://www.cplusplus.com/doc/tutorial/>. Acesso em 13 Agosto 2012.

[11] DB2 Universal Database. 2012. Disponível em <http://www-01.ibm.com/software/br/db2/data/db2/>. Acesso em 13 Agosto 2012.

[12] ELFEKY, M. G; ELMAGARMID, A. K.; VERYKIOS, V. S. **TAILOR: A Record Linkage Tool Box.** Proc. 18th IEEE Int'l Conf. Data Eng. (ICDE '02), 2002, p. 17-28.

[13] ELMAGARMID, A. K.; IPEIROTIS, P. G.; VERYKIOS, V. S. **Duplicate Record Detection: A Survey.** In: IEEE Transactions on Knowledge and Data Engineering. 2007, vol. 19, p. 1-16.

[14] EREDICS, P.; DOBROWIECKI, T. P. **Data Cleaning for an Intelligent Greenhouse.** In: 6th IEEE International Symposium on Applied Computational Intelligence and Informatics. 2011, p. 293-297.

[15] FLAMINGO Project on Data Cleaning, 2012. Disponível em <http://www.ics.uci.edu/ flamingo/>. Acesso em 10 Abril 2012.

[16] FREELY Extensible Biomedical Record Linkage, 2012. Disponível em <http://sourceforge.net/projects/febrl>. Acesso em: 10 Abril 2012.

[17] GILL, L. E. **OX-LINK: The Oxford Medical Record Linkage System.** Proc. Int'l Record Linkage Workshop and Exposition. 1997, p. 15-33.

[18] GRAVANO, L. IPEIROTIS, P. G.; KOUDAS, N.; SRIVASTAVA D. **Text Joins in an RDBMS for Web Data Integration.** Proc. 12th Int'l World Wide Web Conf. (WWW12), 2003, p. 90-101.

[19] HOLMES, D.; MCCABE, M. C. **Improving Precision and Recall for Soundex Retrieval.** Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'02), IEEE Computer Society, 2002.

[20] HUANG, Y.; ZHANG, X.; YUAN, Z.; JIANG, G. **A Universal Data Cleaning Framework Based on User Model.** In: 2009 ISECS International Colloquium on Computing, Communication, Control and Management. IEEE CCCM 2009, p. 200-202.

[21] HUANZHUO, Y.; DI, W.; SHUAI, C. **An Open Data Cleaning Framework Based on Semantic Rules for Continuous Auditing.** In: 2010 2nd International Conference on Computer Engineering and Technology. 2010, vol. 2, p. 158-162.

[22] INFORMATICA COMPORATION. Informatica Data Quality User Guide. Versão 8.6.1. Set. 2008, p. 89-90.

[23] JACCARD, P. **Etude comparative de la distribution florale dans une portion des Alpes et des Jura.** Bulletin de la Société Vaudoise des Sciences Naturelles, 1901. Vol. 37, p. 547-579.

[24] JAVASCRIPT Ecmascript Programming Language, 2012. Disponível em <http://www.ecmascript.org>. Acesso em 13 Agosto 2012.

[25] LIU, A. X.; SHEN, K.; TORNG, E. **Large Scale Hamming Distance Query Processing.** In: ICDE Conference 2011. IEEE, p. 553-564.

[26] MANDAL, A. K.; HOSSAIN, M. D., NADIM, M. **Developing an Efficient Search Suggestion Generator, Ognoring Spelling Error for High Speed Data Retrieval Using Double Metaphone Algorithm.** In: Proceedings of 13th International Conference on Computer and Information Technology (ICCIT 2010), Dec. 2010, p.317-320.

[27] MONGE, A. E.; ELKAN, C. P. **The Field Matching Problem: Algorithms and Applications.** Proc. Second Int'l Conf. Knowledge Discovery and Data Mining (KDD '96), 1996. p. 267-270.

[28] OKITA, T. **Data cleaning for word alignment.** In: ACL-IJCNLP 2009 - Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore, p. 72-80.

[29] OLIVEIRA, P.; RODRIGUES, F.; HENRIQUES, P. **SmartClean: An Incremental Data Cleaning Tool.** In: 2009 Ninth International Conference on Quality Software, IEEE Computer Society. 2009, p. 452-457.

[30] ORACLE Oracle database. 2012. Disponível em <http://www.oracle.com>. Acesso em 13 Agosto 2012.

[31] PETROVIC, S.; BAKKE, S. **Improving the Efficiency of Misuse Detection by Means of the g-gram Distance.** In: The Fourth International Conference on Information Assurance and Security. IEEE, 2008. p. 205-208, 2008.

[32] POSTGRESQL The world's most advanced open source database. 2012. Disponível em <http://www.postgresql.org>. Acesso em 13 Agosto 2012.

[33] PRASAD, K. H.; FARUQUIE, T. A.; JOSHI, S.; CHATURVEDI, S.; SUBRAMANIAM, V.; MOHANIA, M. **Data Cleansing Techniques for Large Enterprise Datasets.** In: In: 2011 Annual SRII Global Conference. IEEE Computer Society, 2010, p.135-144.

[34] NOKIA Qt Cross-platform application and UI framework, 2012. Disponível em <http://qt.nokia.com>. Acesso em 13 Agosto 2012.

[35] RAHM, Erhard; DO, Hong Hai. **Data Cleaning: Problems and Current Approaches.** IEEE Data Engineering Bulletin, Dez. 2000. p. 3-13.

[36] SQLITE SQLite. 2012. Disponível em <http://www.sqlite.org>. Acesso em 13 Agosto 2012.

[37] SHIWEI, Z.; JUNJIE, W.; GUOPING, X. **TOP-K Cosine Similarity Interesting Pairs Search.** In: 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010). IEEE Circuits and Systems Society, 2010, p.1479-1483.

[38] TALF, R. L. **Name Search Techniques.** Technical Report Special Report n. 1, New York State Identification and Intelligence System, Albany, N.Y. Feb. 1970.

[39] VALêNCIO, C. R. ; ENUMO, C. C. N. ; JARDINI, T. ; LAURENTI, C. H. **Otimização de técnicas e algoritmos aplicados à limpeza de Banco de Dados utilizando Multithreading.** In: WWW/Internet 2010 - Conferência IADIS Ibero-Americana, 2010, Algarve, Portugal. Actas da Conferência IADIS IBERO-AMERICANA WWW/INTERNET 2010, 2010. p. 414-418.

[40] WANG, H. **The research of outlier data cleaning through revelance comparison.** In: e-Business and Informtion System Security (EBISS), 2010 2nd International Conference on, May 2010. p. 1-3.

[41] WANG, Y.; YING-HUA, L. **Deep Web Entity Identification Method Based on Improved Jaccard Coefficients.** In: 2009 International Conference on Research Challenges in Computer Science. IEEE Computer Society, 2009. p. 112-115.

[42] WIKIPEDIA, 2012. Disponível em <www.wikipedia.com>. Acesso em 10 Abril 2012.

[43] WORD-BASED Information Representation Language, 2012. Disponível em <http://www.cs.cmu.edu/ wcohen/whirl/>. Acesso em 10 Abril 2012.

[44] XINLIN, Q.; LIPING, D.; DEREN, L.; PINGXIANG, L.; LITE, S.; LIEFEI, C. **Data Cleaning Approaches in Web2.0 VGI Application.** In Geoinformatics, 2009 17th International Conference on. 2009, p. 1-4.

[45] YAN, H.; DIAO, X.; LI, K. **Research on Information Quality Driven Data Cleaning Framework.** 2008 International Seminar on Future Information Technology and Management Engineering. IEEE Computer Society, 2008, p. 537-539.

[46] YANCEY, W. E. **Bigmatch: A Program for Extracting Probable Matches from a Large File for Record Linkage.** Technical Report Statistical Research Report Series RRC2002/01, US Bureau of the Census, Washington, D.C., mar. 2002.

[47] YANHUA, S.; SHUYU, L. **The Design and Implementation of Dynamic Data Cleaning Modeling.** In: 2010 International Conference on Computer Application and System Modeling (ICCASM 2010). Vol. 5, p. 257-260, 2010.

[48] YUBIN, B.; JIE, S.; JINGANG, S.; GE, Y. **Case Study on Modeling Approaches and Framework of Scientific Data Cleaning.** In: IEEE Ninth International Conference on Computer and Information Technology. IEEE Computer Society, 2009, p. 266-271.

[49] ZHAN, S.; BYUNG-RYUL, A.; KI-YOL, E.; MIN-KOO, K.; JIN-PYUNG, K.; MOON-KYUN, K. **Plagiarism Detection Using Levenshtein Distance and Smith-Waterman Algorithm.** In: The 3rd International Conference on Innovative Computing Information and Control (ICICIC'08), IEEE, 2008.

[50] ZHANG, P.; DANG, X.; CHEN, H. **Based on Multi-Agend 3-layer Data Cleaning System Model Design.** In: Intelligent Computing and Integrated Systems (ICISS), 2010 International Conference on. IEEE, 2010, p. 544-547.