UNIFACS - UNIVERSIDADE SALVADOR
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO
MESTRADO EM SISTEMAS E COMPUTAÇÃO

MARTA MESQUITA MOTA DUNCE

AVALIAÇÃO DE SIMILARIDADE ENTRE CONCEITOS REPRESENTADOS PELA XBRL

Salvador
2013

MARTA MESQUITA MOTA DUNCE

AVALIAÇÃO DE SIMILARIDADE ENTRE CONCEITOS REPRESENTADOS PELA XBRL

Trabalho de conclusão de Dissertação apresentada ao Curso de Mestrado Acadêmico em Sistemas e Computação da Universidade Salvador - UNIFACS, como requisito parcial para obtenção do título de Mestre.

Orientador: Prof. Dr. Paulo Caetano da Silva.
Co-orientador: Prof. Dr. Sidney Viana.

Salvador
2013

Ficha Catalográfica elaborada pelo Sistema de Bibliotecas da Universidade Salvador – UNIFACS, Laureate Internacional Universities.

Dunce, Marta Mesquita Mota.

Avaliação de similaridade entre conceitos representados pela XBRL. Marta Mesquita Mota Dunce. – Salvador, 2013.

112 f. : il.

Dissertação apresentada ao Curso de Mestrado em Sistemas e Computação UNIFACS Universidade Salvador, Laureate Internacional Universities como requisito parcial para a obtenção do grau de Mestre.

Orientador: Prof. Dr. Paulo Caetano da Silva.
Co-orientador: Prof. Dr. Sidney Viana.

1. XBRL (Linguagem de marcação de negócio) 2. Gestão da Qualidade de Dados. 3. Avaliação de Similaridade. I. Silva, Paulo Caetano da, orient. II. Viana, Sidney, co-orient. III. Título.

CDD: 005.133

MARTA MESQUITA MOTA DUNCE

AVALIAÇÃO DE SIMILARIDADE ENTRE CONCEITOS REPRESENTADOS PELA XBRL

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Sistemas e Computação, UNIFACS Universidade Salvador, Laureate Internacional Universities, pela seguinte banca examinadora:

Paulo Caetano da Silva – Orientador _____
Ph.D. em Computer Science, Universidade Federal de Pernambuco (UFPE)
UNIFACS Universidade Salvador, Laureate Internacional Universities

Sidney da Silva Viana Co-Orientador _____
Doutorado em Engenharia Elétrica pela Universidade de São Paulo (USP)
UNIFACS Universidade Salvador, Laureate Internacional Universities

Jorge Alberto Prado de Campos _____
Pós-doutorado no National Center for Geographic Information and Analysis-NCGIA (EUA)
UNIFACS Universidade Salvador, Laureate Internacional Universities

Adicineia Aparecida de Oliveira _____
Doutorado em Engenharia Elétrica pela Universidade de São Paulo (USP)
Universidade Federal de Sergipe (UFS)

Salvador, 27 de abril de 2013.

Dedico este trabalho a meus filhos - Erik e Felipe – Amor infinito!

AGRADECIMENTOS

Acima de tudo, agradeço a Deus pela vida. Agradeço a meus pais, José Carlos e Nivea, e à minha irmã, Maria, por todo amor e por terem me ensinado através de exemplos e palavras a importância do conhecimento e do aprender. A meu marido e companheiro de toda a vida, Christian, gratidão eterna pelo apoio sempre constante, pelo carinho, consideração e por sempre ter acreditado em minha capacidade. Agradeço a meus colegas de trabalho, profissão e vida que, acima de tudo, são grandes amigos, pelo incentivo, apoio e interesse neste trabalho (Karina, Geisa, AB, Lívia, Manoel Neto, Kiko, Keila, Papo, Zé, Giovanna, Cabelo, Diego, Léo e Thiago). Agradeço a meus professores, pelo incentivo, disposição em compartilhar o seu conhecimento e amor pela profissão de ensinar, em especial aos meus orientadores Prof. Paulo Caetano e Prof. Sidney Viana, por acreditarem no êxito deste processo. O apoio de cada um foi a maior contribuição para a realização deste trabalho.

RESUMO

A XBRL é uma linguagem baseada em XML consolidada como o padrão para publicação e intercâmbio de informações financeiras na Web. A XBRL é extensível, característica que ampliou a sua área de atuação e incrementou a adesão da comunidade financeira. Os conceitos representados pela XBRL são a base para as informações financeiras. Devido à ampliação do uso da XBRL e aumento na criação de novos conceitos financeiros, torna-se relevante a aplicação da disciplina de Gestão da Qualidade de Dados (GQD) neste contexto. A avaliação de similaridade é um processo importante na GQD e serve como apoio para algumas de suas atividades essenciais. A classificação de conceitos representados pela XBRL não idênticos, segundo a semelhança entre eles é útil no estudo dos conceitos (agrupamento), na integração de conceitos (detecção de duplicados), no controle de versões (detecção de mudanças), na recuperação de informações (ordem dos resultados) e outras aplicações. O objetivo deste trabalho é propor um processo de avaliação de similaridade entre conceitos representados pela XBRL. Para tanto, foi feita uma revisão bibliográfica sobre técnicas de avaliação de similaridade, um levantamento sobre as características dos conceitos representados pela XBRL, a proposição de um processo para avaliação de similaridade entre conceitos representados pela XBRL com base em técnicas encontradas na literatura e um estudo de caso, no qual é aplicado o processo proposto para uma taxonomia em desenvolvimento.

Palavras-chave: Gestão da Qualidade de Dados. Avaliação de Similaridade. XBRL**.**

ABSTRACT

XBRL is an XML-based language consolidated as the standard for publishing and exchange of financial information on the Web. XBRL is extensible, feature that expanded its area of operation and increased the use by the financial community. XBRL concepts are the basis for the financial information. Due to the expanded use of XBRL and the increase in the creation of new financial concepts, it becomes relevant the application of the Data Quality Management (DQM) discipline in this context. Similarity evaluation is an important process in data management and serves as support for some of its core activities. The classification of XBRL concepts that are not identical according to the similarity between them is useful in the study of concepts (grouping), the integration of concepts (duplicate detection) in version control (change detection) in information retrieval (order results) and other applications. The objective of this work is to propose a process to evaluate similarity between concepts of XBRL. Therefore, a literature review on techniques for similarity evaluation and a survey on the characteristics of the concepts represented by XBRL were performed; a process for similarity evaluation between concepts represented by XBRL, based on techniques from the literature, was proposed and a case study was made, which applied the proposed process in a developing taxonomy.

Keywords: Data Quality Management. Similarity Evaluation. XBRL.

LISTA DE FIGURAS

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTO

A XML é uma linguagem de marcação, baseada em texto, para representação de informações estruturadas. Proposta pelo W3C[1] em 1996, é amplamente utilizada e tem evoluído desde então.

As linguagens de marcação cresceram em importância com a necessidade de se adicionar significado a informações sendo transferidas através da internet. O padrão HTML (*Hypertext Markup Language*) surgiu em 1989. Desenvolvido por Tim Berners-Lee, se tornou a linguagem mais utilizada na internet. Entretanto, os elementos de marcação do HTML se limitam a descrever como a informação deve ser apresentada e não descrevem seu significado. Além disso, o HTML é pouco flexível e a inclusão de novos elementos de marcação é um processo custoso.

Em 1986, a Organização para Padronização Internacional (ISO) [2] havia aprovado um padrão de metalinguagem (conjunto de regras para definição de novas linguagens) para etiquetar informações com conteúdo semântico, conhecido como *Standard Generalized Markup Language* (SGML) (ISO 8879). No entanto, a SGML se revelou uma linguagem muito genérica e complexa.

A XML foi especificada a partir da SGML, na tentativa de se resolver as limitações da HTML e da SGML. Um documento em XML pode ser publicado na Web, interpretado por pessoas ou processado por aplicações. É definido pelo W3C como um formato textual simples para representação de informações estruturadas (BRAY et al., 2008).

A XBRL (*eXstensible Business Reporting Language*) é uma linguagem, baseada em XML, *XML Schema* e *XLink*, para divulgação e intercâmbio de informações financeiras (SILVA et al., 2006). XBRL vem sendo adotada por diversas instituições e empresas em todo mundo com o suporte de um consórcio global com mais de 650 membros que incentivam a criação de

---

[1] W3C Consortium: http://www.w3.org/

[2] ISO – International Organization for Standardization: www.iso.org

jurisdições locais em cada país para estabelecimento de uma taxonomia local[3]. Atualmente o consórcio conta com 24 jurisdições, sendo que em diversos países, e.g. Estados Unidos, Grã-Bretanha e Austrália, XBRL já é a linguagem oficial para divulgação de relatórios obrigatórios aos órgãos de governo.

Os estudos para definição da XBRL iniciaram em 1998, quando Charles Hoffman, um contador público certificado nos Estados Unidos, começou a estudar o uso de XML para padronizar a divulgação eletrônica de informações financeiras, apoiado pelo *American Institute of Certified Public Accountants* (AICPA). Em 2000, foi lançada a XBRL 1.0. A partir do surgimento da *XML Schema* e Xlink. A versão 2.0 foi lançada em dezembro de 2001. Em dezembro de 2003, foi lançada a versão 2.1, corrigindo algumas deficiências detectadas com o uso da versão anterior (HOFFMAN, 2006). Até a data em que este trabalho foi elaborado, a versão 2.1 se manteve como a versão mais atual e estável da XBRL, apoiada pelo *XBRL Consortium*.

A linguagem XBRL define a estrutura básica dos documentos de instância, os que portam os dados, e especifica como taxonomias podem ser criadas para acomodar particularidades de cada organização por meio da introdução de novos elementos, denominados conceitos. A possibilidade de estender a linguagem é uma característica que possibilita a ampliação de seu uso.

Uma taxonomia XBRL é formada por um documento *XML Schema xsd* (FALLSIDE; WALMSLEY, 2004), no qual ocorrem as definições dos conceitos que irão informar os dados na instância, e documentos *linkbases*, que comportam um agrupamento de relações do tipo *XLink* (DEROSE et al., 2001) e estabelecem relacionamentos entre os conceitos. Uma taxonomia XBRL tem a propriedade de poder importar outra, que por sua vez pode importar uma terceira e assim sucessivamente. Um documento de instância XBRL pode fazer uso de várias taxonomias.

Devido ao aumento do uso da linguagem XBRL, da criação de novas taxonomias e da definição de novos conceitos, é cada vez mais relevante a necessidade de aplicação da Gestão de Qualidade de Dados (*Data Quality Management*) aos conceitos representados pela XBRL,

[3] XBRL Consortium: http://www.xbrl.org

por esta disciplina ter a propriedade de administrar os vários aspectos relativos à coleta, manutenção, aprimoramento e compartilhamento dos dados neste contexto. A má administração de conceitos financeiros criados em taxonomias XBRL pode levar a diversos problemas nas análises dos documentos de instância, tal qual a duplicidade de conceitos, levando a inconsistências nas declarações.

Na Gestão da Qualidade de Dados, o processo de avaliação de similaridade é um relevante apoio utilizado em várias atividades relacionadas a esta disciplina. O objetivo da avaliação de similaridade é classificar um conjunto de dados usando como critério a semelhança entre eles. Pode ser aplicado em atividades de limpeza ou integração de dados, cujo foco é a busca de dados duplicados. Em atividades de recuperação de dados, a avaliação de similaridade pode contribuir para a seleção e classificação dos resultados segundo critérios de relevância. No controle de versões, identifica a semelhanças e diferenças entre instâncias de um documento, indicando as alterações sofridas.

Devido à grande variedade de tipos de dados e domínios de aplicação, o estudo de avaliação de similaridade recebe diferentes nomes na literatura, tais quais, *fuzzy duplicates* (ANANTHAKRISHNA et al., 2002), linkagem de registros (FELLEGI; SUNTER, 1969), resolução de entidades (COHEN; RICHMAN, 2002), identificação de objetos (TEJADA et al., 2002) e identificação de duplicados (ELMAGARMID et al., 2007). Esta área de conhecimento é pesquisada desde a década de 70 e foi formalmente definida pela primeira vez em (FELLEGI; SUNTER, 1969).

1.2 JUSTIFICATIVA

Dentro do contexto apontado, a importância da linguagem XBRL e a existência de vasta literatura sobre técnicas de avaliação de similaridade foram as principais motivações para o desenvolvimento deste trabalho.

Projetos e taxonomias XBRL são criados com o objetivo de padronização e intercâmbio de informações financeiras, isto possibilita a ocorrência de problemas de qualidade nos dados sendo gerados e leva à necessidade de maior controle sobre os conceitos criados. A avaliação de similaridade entre conceitos representados pela XBRL auxilia na gestão das taxonomias

por ser um processo base para atividades de melhoria da qualidade de dados aplicadas a este contexto.

O processo de avaliação de similaridade tem recebido bastante atenção na comunidade acadêmica mundial, viabilizando a avaliação de técnicas propostas e conhecimento do estado da arte para aplicação no contexto dos conceitos representados pela XBRL.

1.3 PROBLEMA

A avaliação de similaridade entre conceitos representados pela linguagem XBRL é o foco deste trabalho. Estes conceitos podem estar presentes em uma mesma taxonomia ou em taxonomias diferentes. Existem várias aplicações para a aplicação do processo de avaliação de similaridade entre conceitos representados pela XBRL:

a) Detecção de conceitos duplicados: evitar representação de dados na instância por conceitos distintos, mas duplicados, gerando inconsistência nas análises dos documentos. A heterogeneidade de dados XML contribui para a probabilidade de criação de conceitos duplicados. Silva (2010) classifica em três tipos a heterogeneidade de dados XML: semântica, na qual as informações são similares, mas os nomes são diferentes; sintática de conteúdo, no qual mesmos conceitos podem ser representados de diferentes formas; e, estrutural, no qual as informações podem estar representadas em diferentes estruturas. A detecção de conceitos duplicados é útil também na integração de taxonomias, a fim de apoiar a junção dos conceitos, e.g. há a fusão de duas empresas e há necessidade de unir os conceitos de cada taxonomia em uma taxonomia única, eliminando os conceitos duplicados.

b) Agrupamento (*clustering*) de conceitos similares: o agrupamento pode ser utilizado para facilitar a compreensão dos conceitos, classificação de conceitos e aplicação de regras em conceitos similares.

c) Pesquisa de conceito por similaridade: quando se deseja pesquisar um conceito e não se tem certeza de todas as suas características, pode-se pesquisar a partir do uso de palavras-chave em atributos ou características de estrutura. A pesquisa retorna conceitos similares aos critérios de pesquisa.

1.4 OBJETIVO

O objetivo central deste trabalho é propor um processo de avaliação de similaridade entre conceitos representados pela XBRL.

Para alcançar esse objetivo algumas metas foram estabelecidas, por meio da realização de um levantamento das técnicas de avaliação de similaridade de dados e aplicação destas técnicas aos conceitos representados pela XBRL. São elas:

a) Definição de medidas de similaridade a serem aplicadas aos conceitos representados pela XBRL;
b) Especificação e implementação de um processo de avaliação de similaridade entre conceitos representados pela XBRL;
c) Realização de estudos de casos em domínios diferentes, a fim de avaliar a aplicabilidade do processo proposto e especificado.

1.5 METODOLOGIA

Com base na metodologia utilizada para coleta e análise dos dados, este trabalho se enquadra no grupo de pesquisa bibliográfica, devido à realização de um estudo sobre técnicas de avaliação de similaridade, identificando as que poderiam ser aplicadas ao contexto proposto: conceitos financeiros representados pela XBRL.

A seguir, foi construído o processo de avaliação de similaridade entre conceitos representados pela XBRL por meio da implementação das técnicas identificadas como mais apropriadas. A aplicação do processo proposto caracteriza a experimentação do trabalho como um estudo de caso em que o processo proposto é aplicado a dois contextos da vida real: uma taxonomia em construção e extensões de uma taxonomia. Após a aplicação das técnicas aos estudos de caso foi feito o registro e avaliação dos resultados obtidos.

1.6 APRESENTAÇÃO

O restante deste trabalho está organizado em cinco capítulos. O segundo capítulo aborda os fundamentos da linguagem XBRL necessários à compreensão do problema. No terceiro capítulo são tratados conceitos relativos à avaliação de similaridade e descrevem-se alguns

trabalhos desta disciplina. O quarto capítulo apresenta a justificativa para a escolha das técnicas de avaliação de similaridade aplicadas ao contexto dos conceitos representados pela XBRL e o processo proposto. O quinto capítulo apresenta os experimentos realizados e suas análises. O sexto capítulo apresenta as considerações finais sobre o trabalho realizado e propõe alguns trabalhos futuros. No apêndice são listados os roteiros utilizados para criação de tabelas funções e procedimentos do processo proposto.

2 XBRL

O objetivo deste capítulo é descrever as principais características da XBRL para entendimento do problema apresentado no trabalho. Visto que a XBRL é derivada da XML, são abordados inicialmente os conceitos da XML relevantes à XBRL. A seguir, são discutidos os assuntos da XBRL relevantes ao entendimento do problema.

2.1 XML

Um documento XML possui um conjunto de nós (raiz, elementos, atributos e valores) organizados em uma estrutura hierárquica de árvore. Todos os nós possuem uma identificação. O nó-raiz não tem antecessores e tem um conjunto de elementos filhos. Nós de elementos são delimitados por uma marcação de abertura e fechamento e podem ter conteúdo simples (somente nós de valores), complexo (somente nós de elementos) ou misto (nós de elementos e nós de valores). Atributos sempre possuem somente um nó de valor, são sempre de conteúdo simples. Nós de valor têm tipos pré-definidos e representam as folhas da árvore (ELMASRI; NAVATHE, 2005). A Figura 1 apresenta um exemplo de documento XML.

Figura 1 – Documento XML

```
<?xml version="1.0"?>
<relatoriocustos ano="2012">
        <departamento>
          <nome>Administrativo</nome>
          <custo>
                <previsto>22000</previsto>
                <realizado>25000</realizado>
          </custo>
        </departamento>
</relatoriocustos>
```

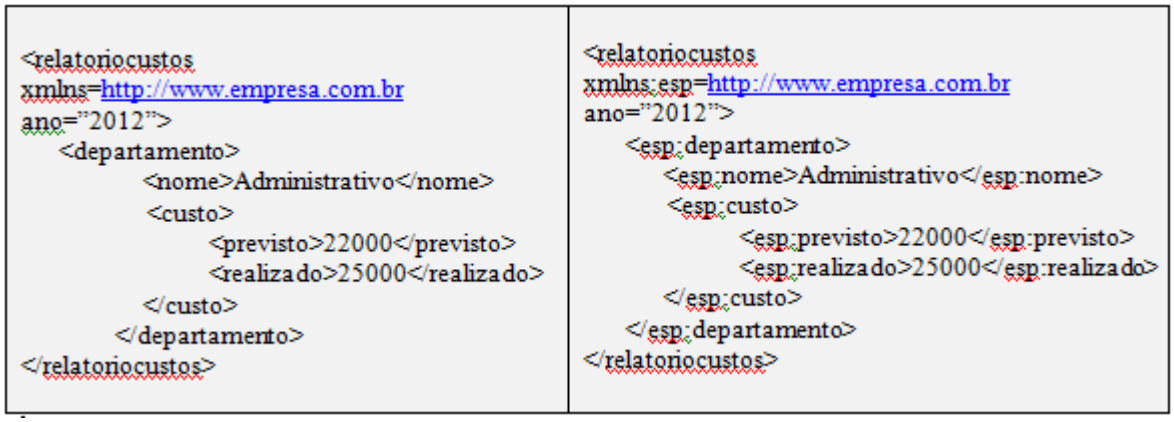
A seguir serão descritos alguns conceitos da XML relevantes ao trabalho proposto, a saber: *Namespaces, XML Schema e XLink.*

2.1.1 Namespaces

Em um documento XML os elementos e atributos são identificados através dos seus nomes. *Namespaces* é a especificação que determina a estrutura para evitar conflitos nos nomes de elementos, qualificando-os. A qualificação dos nomes é feita através de um *namespace XML* que é identificado por uma referência URI (BRAY et al., 2009).

Uma referência URI pode conter caracteres não permitidos em nomes de elementos e atributos, além de ser permitido que seu tamanho seja bastante grande (BERNERS-LEE et al., 2005) o que poderia tornar inconveniente o seu uso em documentos XML. Para contornar estes problema, a especificação *Namespaces* determina que a definição de um *namespace* seja feita através do atributo "**xmlns"**, permitindo a associação a um nome válido na XML para utilização em prefixos de nomes de elementos e atributos. Os nomes que utilizam *namespaces* como prefixos são conhecidos como nomes qualificados. Uma vez definido o atributo "**xmlns**" de um elemento, todos os seus elementos filhos serão associados ao *namespace*, sendo opcional o uso do nome qualificado. A Figura 2 apresenta dois exemplos de uso de *namespaces*. No primeiro exemplo o uso de *namespaces* é feito por omissão, ou seja, os nomes dos elementos filho não estão qualificados, enquanto que no segundo exemplo, todos os nomes estão qualificados.

Figura 2 – Uso de *namespaces*



2.1.2 XML Schema

A necessidade de restringir o conteúdo de um documento XML a fim de possibilitar o seu processamento automático fez surgir os esquemas XML. O esquema especifica quais

elementos e atributos são permitidos em um documento XML. O uso de um esquema permite a criação de subconjuntos da XML para domínios específicos, facilitando a troca de informações.

A especificação XML propôs, inicialmente, o padrão *Document Type Definition* (DTD) para a representação de esquemas (BRAY et al., 2008). O DTD utiliza uma sintaxe específica para definição de restrições sobre documentos XML que podem estar abrigadas dentro do próprio documento XML ou em um arquivo externo.

O padrão DTD, em razão de limitações, foi substituído pela *XML Schema Definition* (XSD), uma especificação recomendada pelo W3C, também chamada *XML Schema*. Assim como o DTD, a *XML Schema* permite especificar elementos e atributos que podem aparecer no documento, assim como ordem de apresentação dos elementos, cardinalidade, hierarquia, tipos de dados, valores padrão destes atributos e elementos (FALLSIDE; WALMSLEY, 2004). Algumas vantagens da *XML Schema* sobre a DTD é que eles são extensíveis, suportam tipos de dados e são escritos em XML. As *tags* de marcação da *XML Schema* utilizam o *namespace* "**xs**".

Abordaremos a seguir algumas características da *XML Schema* para facilitar a compreensão do trabalho aqui apresentado, ressaltando que o objetivo não é o detalhamento dessa especificação.

- *Tipos de dados*: Os tipos de dados da *XML Schema* podem ser simples ou complexos. A especificação XML Schema define um conjunto de tipos de dados primitivos, classificados como simples. Dentro do conjunto de tipos de dados primitivos, existem tipos de dados para representar texto ("string", "token", "ID", "Name" e outros), datas ("date", "dateTime", "duration" e outros) e números ("integer", "long", "short", "decimal", "byte" e outros). Estes tipos podem ser estendidos para se acrescentar novos tipos simples, que são refinamentos dos tipos primitivos feitos através de restrições (chamadas *Facets*). Os tipos de dados simples são utilizados na especificação de novos elementos e atributos. Os novos elementos e atributos declarados também podem ser considerados tipos de dados, classificados como complexos.

- *Declaração de elementos*: A declaração de novos elementos na *XML Schema* é feita através da *tag* "**element**". Elementos simples contêm apenas texto, elementos complexos contêm outros elementos ou atributos.

Elementos simples utilizam o atributo "**type**" indicando o tipo de dados do conteúdo e podem determinar um valor padrão, especificado na *tag* "**default**" ou fixar um valor que não poderá ser modificado, especificado na *tag* "**fixed**".

Elementos complexos devem utilizar a *tag* "complexType" para abrigar seus elementos e atributos. Dentro da *tag* "complexType", pode-se utilizar indicadores de ordem, de ocorrência ou de grupo para organizar os elementos filho.
Na XBRL, a definição de novos elementos é padronizada. Os atributos da tag "**xs:element**" mais utilizados na XBRL são:

-"**id**": identificador do elemento;
-"**type**": tipo do elemento;
-"**substitutionGroup**": indica que o elemento pode substituir outro elemento do grupo indicado em declarações;
-"**nillable**": indica se o elemento pode ter valor nulo;
-"**name**": nome do elemento;
-"**abstract**": opcional, indica se o elemento é abstrato ou pode receber valor.

- *Declaração de Atributos*: Os atributos da *XML Schema* são tipos simples, declarados através da *tag* "**attribute**" e somente podem ser utilizados em elementos complexos. Os principais atributos da *tag* "**attribute**" são "**name**" e "**type**". Por padrão, todos os atributos são opcionais, o atributo "**use**" com valor "**required**" deve ser utilizado para torná-lo obrigatório. A declaração de atributos de um elemento complexo sempre vem após as declarações dos seus elementos. A Figura 3 apresenta a definição de um novo atributo ("**balance**") criado na XML Schema da XBRL para ser aplicado a novos elementos.

Figura 3 – Definição do atributo “**balance**”

```
<schema targetNamespace="http://www.xbrl.org/2003/instance"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xbrli="http://www.xbrl.org/2003/instance"
  xmlns:link="http://www.xbrl.org/2003/linkbase"
  elementFormDefault="qualified">

  <attribute name="balance">
    <annotation>
      <documentation>
      The balance attribute (imposes calculation relationship restrictions)
      </documentation>
    </annotation>
    <simpleType>
      <restriction base="token">
        <enumeration value="debit" />
        <enumeration value="credit" />
      </restriction>
    </simpleType>
  </attribute>

</schema>
```

2.1.3 XLINK

A representação de relacionamentos é um aspecto fundamental em bases de dados, enriquecendo a semântica das informações. A própria estrutura da XML indica, de forma implícita, relacionamento hierárquico entre os elementos pai e filhos. Relacionamentos explícitos em XML podem ser feitos através do uso de **id** e **idref**.

A fim de aprimorar a representação de relacionamentos, complementando a XML, o W3C definiu a *XML Linking Language* (*XLink*) que especifica como deve ser feita a declaração de elementos que definem ligações (*links*) entre dois ou mais recursos (DEROSE et al., 2001). A XBRL faz uso extensivo da tecnologia *XLink*.

Os recursos participantes do *link* podem ser locais ou remotos. Recursos locais são definidos “por valor” e estão localizados no mesmo documento XML do *link*. Recursos remotos são definidos “por referência” e estão localizados externamente (arquivos, documentos, programas, URIs, etc.) ao documento XML em que o *link* se encontra. (DEROSE et al., 2001)

Os *links* podem ser do tipo simples ou estendidos. *Links* simples só podem envolver dois recursos: o de origem local e o de destino remoto. A Figura 4 apresenta um exemplo de link simples, que aponta para a foto (recurso remoto) de um médico (recurso local). *Links*

estendidos podem envolver um ou mais recursos e podem ter estrutura complexa envolvendo elementos do tipo: (DEROSE et al., 2001)

- *locator*: indicam a localização dos elementos remotos do *link*;
- *resource*: identificam os recursos locais participantes do *link*;
- *title:* estabelecem nomes legíveis para o *link*;
- *arc*: definem o comportamento do percurso entre um par de recursos do *link*, inclusive a direção em que eles devem ser atravessados, podendo ser unidirecionais ou multidirecionais. Segundo a localização dos recursos, são classificados como: *Inbound*: destino local e origem remoto; *Outbound*: destino remoto e origem local ou *Third-Party*: destino e origem remotos.

Figura 4 – *link* simples

```
<medicos>
   <medico title="Dra. Joana Caldas">
      <descricao
      xlink:type="simple"
      xlink:href=http://medicos.com/imagens/joanacaldas.gif>
      Dra. Joana Caldas – Pneumologista
      </descricao>
   </medico>
</médicos>
```

Conjuntos de *links* estendidos com arcos *inbound* e *third-party* são chamados de *linkbases* (*link databases*). Servem para isolar os relacionamentos dos documentos de dados, a fim de melhorar a organização e facilitar a manutenção dos *links*. Os *linkbases* são bastante utilizados na linguagem XBRL, como descrito nas seções a seguir.

2.2 XBRL

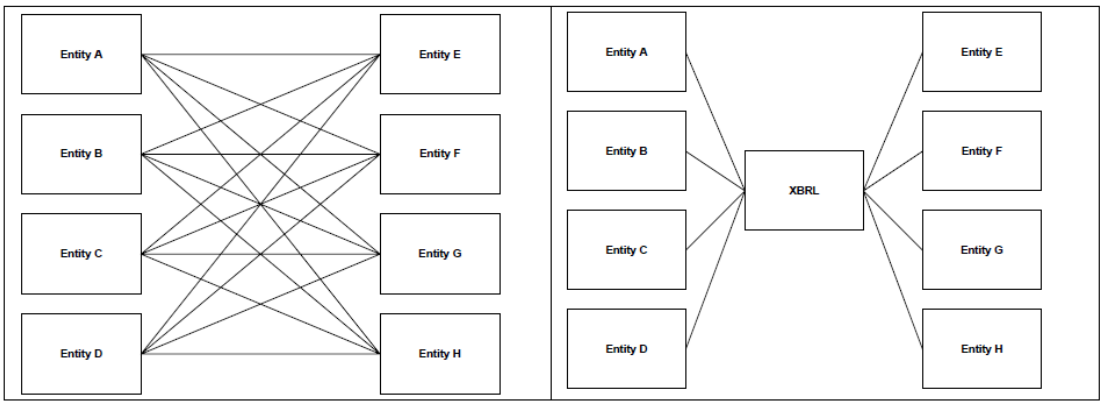
A XBRL está cada vez mais consolidada como a linguagem padrão para divulgação e intercâmbio de informações financeiras de maneira eletrônica. Os governos da Alemanha, Austrália, Estados Unidos e Grã-Bretanha já obrigam o uso da XBRL em relatórios compulsórios de auditorias, além de instituições e empresas em todo o mundo adotarem a XBRL como padrão. Em 2012, o consórcio global contava com mais de 650 membros e 24

jurisdições, incentivando a criação de jurisdições locais em cada país para estabelecimento de uma taxonomia com características regionais específicas[4].

2.2.1 Benefícios da padronização das informações financeiras

A padronização das informações financeiras traz muitos benefícios. A Figura 5 ilustra a vantagem do uso de uma linguagem única para padronizar a comunicação entre as diversas entidades consumidoras de informações financeiras em formato eletrônico (empresas, filiais, holding, órgãos reguladores, etc.), facilitando o intercâmbio de informações. Os custos de transformação da informação entre os diversos formatos, revalidação e correção de erros são eliminados com o uso de um padrão, além da economia de escala, pois entidades poderão compartilhar os custos de manutenção e documentação do formato padrão (HOFFMAN, 2006).

Figura 5 – Padronização de *interfaces*

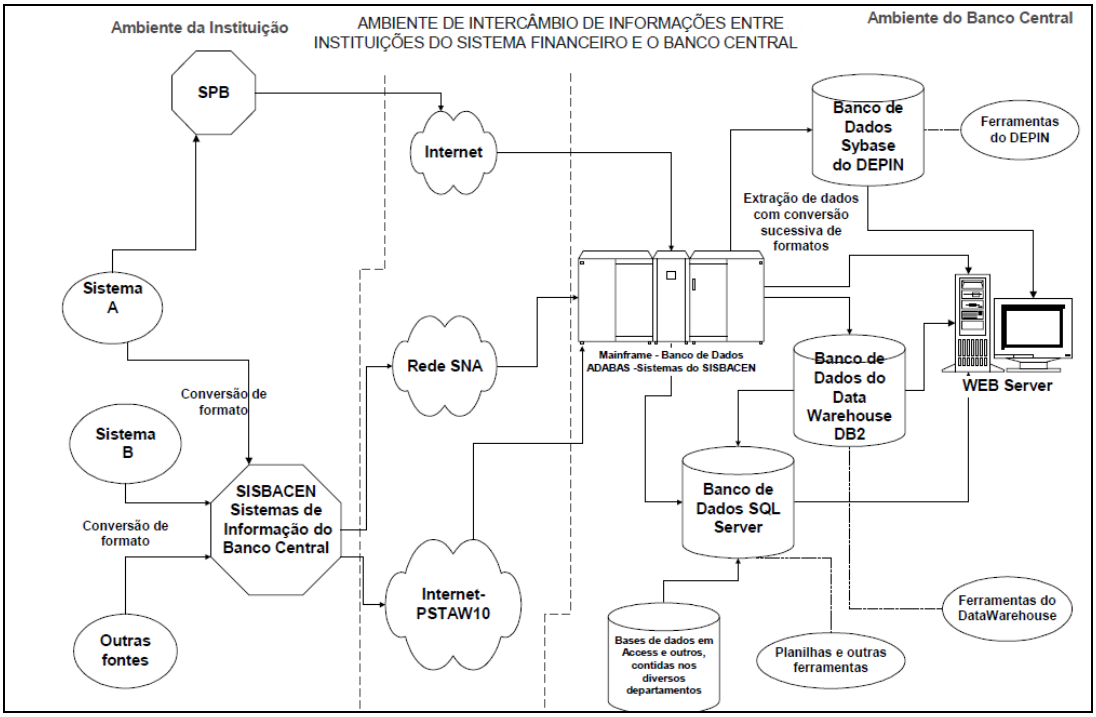


Fonte: Hoffman (2006).

Conforme Figura 6 (SILVA, 2003), representando o fluxo de informações financeiras do Banco Central do Brasil, mesmo dentro de uma mesma entidade os benefícios do uso de um único padrão são mantidos quando o ambiente de aplicações é heterogêneo e há intercâmbio de informações financeiras entre aplicações diferentes.

[4] XBRL Consortium: http://www.xbrl.org

Figura 6 – Ambiente Banco Central



Fonte: Silva (2003).

Ao longo de todas as etapas da cadeia de provisão de informações para divulgação em XBRL, muitos são os beneficiários da padronização (SILVA et al., 2006):

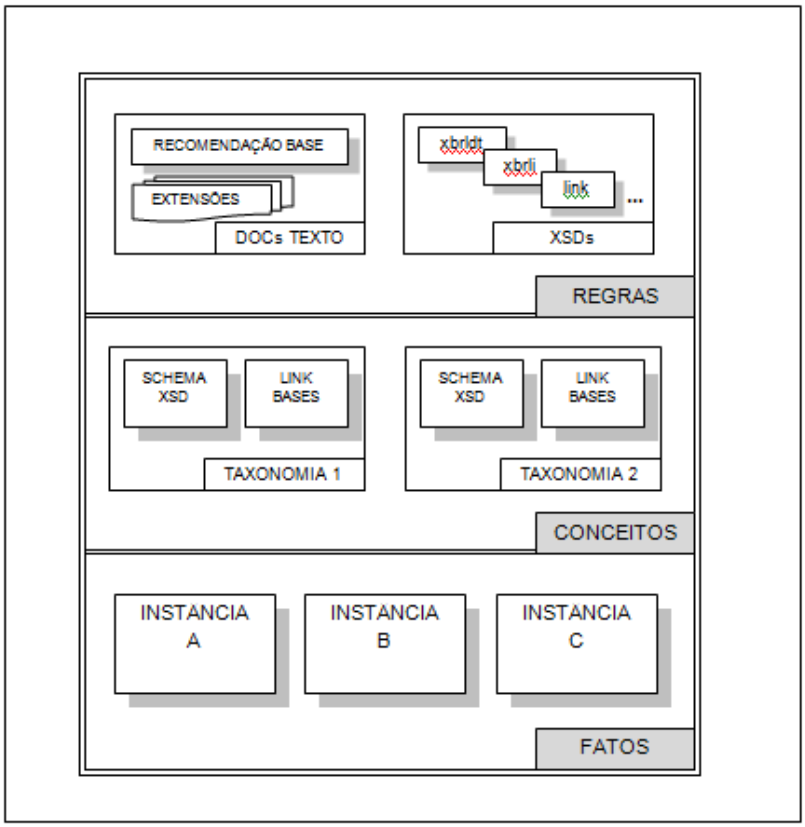
- *produtores de informação*: contadores, analistas financeiros, auditores, etc. têm o tempo de disponibilização da informação reduzido, aumento da acessibilidade e redução de erros de transformação de formatos;
- *consumidores de informação*: órgãos reguladores, investidores, credores, gerentes e executivos se beneficiam da atualização mais frequente das informações bem como da possibilidade de comparar relatórios de fontes diferentes, devido à interoperabilidade;
- *apoiadores do processo*: fornecedores de software, consultores de informática, companhias que preparam relatórios têm novas oportunidades de negócio em que a especialização na linguagem XBRL pode atingir grande número de usuários.

2.2.2 Estrutura da XBRL

A estrutura da XBRL é baseada em três componentes, segmentados em camadas distintas e com funcionalidades próprias e complementares: fatos, conceitos e regras. A 2.2.3 ilustra a estrutura da XBRL.

- *Fatos*: são os valores dos dados financeiros expressos nos relatórios financeiros. Os fatos são informados nos documentos XML chamados de instância;
- *Conceitos*: definem os termos que serão reportados através dos fatos. Os conceitos são declarados em um documento XSD de *Schema XML*. Informações semânticas sobre os conceitos indicando relacionamentos, restrições e detalhes são expressas através de linkbases, que podem estar dentro da *XML Schema* ou em documentos XML separados. A um conjunto de documentos (*XML Schema e Linkbases*) com declarações de conceitos é dado o nome de taxonomia XBRL;
- *Regras*: orientam a definição de conceitos e declaração dos fatos. A recomendação base é um documento textual que indica o que é ou não permitido na XBRL, garantindo a interoperabilidade dos relatórios (ENGEL et al., 2003). Estendendo a recomendação base, o *framework* XBRL inclui documentos com regras complementares (Ex. *Dimensions*, *Formula*, *Versioning* e *Rendering*). Além dos documentos texto, documentos *XML Schema* restringem os elementos e atributos XML que podem ser usados pela XBRL tanto para os documentos de fatos (instâncias), quanto os de conceitos (taxonomias). Todos estes documentos são criados e mantidos pelo XBRL Consortium.

Figura 7 – Estrutura da XBRL



2.2.3 Taxonomia XBRL

Na taxonomia XBRL são definidos os conceitos e as relações entre conceitos. Discute-se a seguir a forma como eles são definidos na XBRL.

2.2.3.1 Declaração de conceitos

Os conceitos representados pela XBRL são declarados em um documento *Schema XSD* através da tag **<element>.** A recomendação XBRL orienta explicitamente o uso dos seguintes atributos na declaração de conceitos:

- ***name***: nome do conceito. É obrigatório e deve ser único dentro do mesmo *Schema XSD*. Quando utilizado em conjunto com outros *Schema*s, a unicidade é garantida através da tecnologia de *namespaces*.
- ***substitutionGroup***: indica se o conceito é um item ou uma tupla. É obrigatório e seu valor só pode ser *xbrli:item* ou *xbrli:tuple*.
- ***type***: qual o tipo de dados do conceito. É um atributo obrigatório.

- ***id***: id do conceito. Não é obrigatório, mas é recomendado seu uso para facilitar o uso dos conceitos nos *linkbases*. Pela especificação *XML Schem*a, não deve haver id duplicado em um conjunto de *Schemas* XSDs relacionados. Sendo assim, a recomendação XBRL orienta usar o atributo ***name*** com um prefixo (por exemplo, o *namespace* da taxonomia).
- ***periodType***: atributo definido pela XBRL indicando se são conceitos medidos em um instante de tempo (valor ***instant***) ou ao longo de um período (valor ***duration***).
- ***balance***: atributo definido pela XBRL para ser utilizado em conceitos cujo tipo seja monetaryItemType ou derivados. Aceita os valores débito (***debit***) ou crédito (***credit***). Seu uso é opcional, apesar de indicado para conceitos contábeis.

Além dos atributos acima, a especificação indica que o conceito pode utilizar qualquer atributo *XML Schema* válido, sendo os mais utilizados ***abstract*** e ***nillable***. A Figura 8 demonstra um exemplo de criação de um conceito, extraído da taxonomia US-GAAP.

Figura 8 – Criação de um conceito

```
<xs:element
        id='us-gaap_Assets'
        name='Assets'
        nillable='true'
        substitutionGroup='xbrli:item'
        type='xbrli:monetaryItemType'
        xbrli:balance='debit'
        xbrli:periodType='instant'
/>
```

2.2.3.2 Relação entre conceitos

A fim de fornecer informações adicionais sobre os conceitos, a recomendação base da XBRL adota o uso de *links* simples e *links* estendidos, definidos na especificação *XLink.* Os *links* são utilizados para expressar relacionamentos entre conceitos, ou entre conceitos e sua documentação. Existem cinco padrões de *links* que podem ser usados no relacionamento entre conceitos, segundo a recomendação base da XBRL (ENGEL et al., 2003):

- *Calculation*: indicam o valor de um conceito como resultado da agregação dos valores de um conjunto de conceitos, cada um com seu multiplicador (“weight”);
- *Presentation*: sugerem hierarquia e ordem de apresentação dos conceitos. A recomendação básica da XBRL sugere a criação de conceitos abstratos para agrupar conceitos que não possuem outra relação entre si, mas que devem ser apresentados juntos. A Figura 9 exemplifica o uso de dois *links presentation* para indicar que os conceitos “caixa” e “banco” devem aparecer como filhos do conceito “atv_circ”, nessa ordem;

Figura 9 – Links presentation

```
<presentationArc
        xlink:type="arc"
        xlink:from="caixa"
        xlink:to="atv_cir"
        xlink:arcrole="http://www.xbrl.org/2003/arcrole/parent-child"
        order="1"/>

<presentationArc
        xlink:type="arc"
        xlink:from="banco"
        xlink:to="atv_cir"
        xlink:arcrole="http://www.xbrl.org/2003/arcrole/parent-child"
        order="2"/>
```

- *Definition:* proveem relacionamentos de especialização (*general-special*), de alias (*essence-alias*), de grupos de conceitos (*similar-tuples*) e de co-ocorrência (*requires-element*) entre conceitos. Atualmente esse tipo de link é também usado para estabelecer definições dimensionais, basedas na recomendação XBRL Dimensions (HERNÁNDEZ-ROS; WALLIS, 2012);
- *Label*: contêm a documentação dos conceitos e permitem a definição de rótulos, para fins de apresentação, em diversos idiomas para os conceitos. A Figura 10 exemplifica um *link label*. O *resource* “bc_atv_circ_lb” contém o rótulo, o *locator* “bc_atv_cir_loc” aponta para o conceito no *XSD Schema* e o *arc* faz a associação entre os dois;

Figura 10 – Criação de *link label*

```
<label
        xlink:type="resource"
        xlink:role="http://www.xbrl.org/2003/role/label"
        xlink:label="bc_atv_circ_lb"
        xml:lang="pt"> Ativo Circulante
/label>

<loc
        xlink:type="locator"
        xlink:href="bcxsdschema.xsd#atv_circ"
        xlink:label="bc_atv_circ_loc"/>

<labelArc
        xlink:type="arc"
        xlink:from="bc_atv_circ_loc"
        xlink:to="bc_atv_cir_lb"
        xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"/>
```

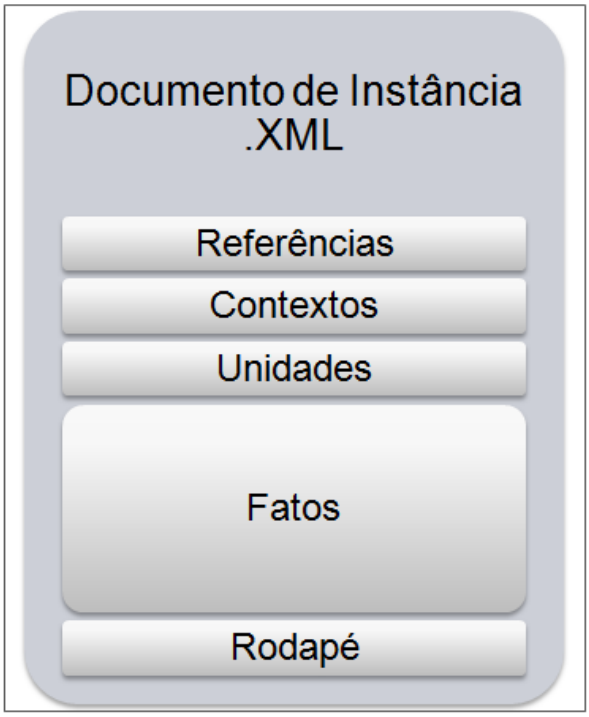
- *Reference*: estabelecem relacionamentos entre os conceitos e referências normativas e/ou legais.

Os links podem estar presentes no mesmo documento *XSD Schema* em que os conceitos são definidos, ou podem estar em documentos XML separados, chamados documentos *linkbases*. A *tag* **<linkbaseRef>** deve ser usada no *XSD Schema* da taxonomia para apontar para os documentos *linkbase*.

2.2.4 Instância XBRL

O documento de instância da XBRL contém os fatos. Nos fatos são declarados os valores reais para os conceitos em um determinado contexto. Uma instância pode reportar fatos de diferentes taxonomias e deve apontar para ao menos uma taxonomia. Uma taxonomia pode apontar para outras taxonomias. Ao conjunto de taxonomias que suportam um documento de instância é dado o nome de *DTS - Discoverable Taxonomy Set*. Uma instância XBRL só pode reportar fatos de conceitos que pertençam ao seu DTS, mas nem todos os conceitos do DTS precisam estar reportados em uma instância.

Figura 11 – Componentes da instância



São componentes da estrutura de um documento de instância:

- *Referências*: contém as referências a *schemas* e *linkbases* usados pelo documento de instância, indicando o seu DTS.

- *Contextos*: criados através do elemento ***context***, situam o fato reportado em um determinado contexto, que inclui informações sobre a entidade, período no tempo e cenário daquele valor. Deve conter o atributo **id** para uso nos fatos.

O elemento ***period*** é obrigatório e pode ser de três tipos: um intervalo de datas (***start date/end date***), uma data única (***instant***) ou sem limite (***forever***). O elemento que representa um fato (e.g. valor monetário) deve referenciar um contexto cujo período seja do tipo idêntico do atributo *periodType* da sua definição no XML *Schema*.

O contexto deve conter o elemento ***entity***, onde é obrigatória a definição do elemento ***identifier***. O valor do elemento identifier deve ser único para a entidade dentro de algum registro indicado no atributo obrigatório ***scheme*** que aponta para uma URI. O elemento ***entity*** também é obrigatório.

Dentro do elemento ***entity***, pode ser definido um segmento (através do elemento ***segment***). A recomendação base da XBRL não faz restrições sobre o elemento ***segment*** para que ele seja utilizado pelas taxonomias livremente com o objetivo de representar dimensões necessárias às informações sendo reportadas tais quais, por exemplo, departamento, projeto, centro de custo, função, unidade, etc..

O atributo **scenario** é opcional e indica que o contexto pressupõe um determinado cenário como, por exemplo, uma projeção em caso da taxa de juros elevar mais do que um x pontos percentuais.

No exemplo a seguir, a sintaxe de criação de um contexto identificado como “c1”.

Figura 12 – Cenário em um contexto

```
<context
        id="c1"
        <period><instant>2012-01-01</instant></period>
        <entity>
                <identifier scheme="http://www.sec.gov/CIK">789019
                </identifier>
                <segment>
                        <acme:project>ISO14000</acme:project>
                </segment>
        </entity>
</context>
```

- *Unidades*: relaciona as unidades de medida dos valores reportados utilizadas pelo documento. Deve conter o atributo **id** para uso nos fatos. Podem ser unidades simples ou compostas. As unidades compostas correspondem a uma relação de unidades como, por exemplo, lucros por ação (medida R$ / medida qtd ações).

- *Fatos*: expressam os valores dos conceitos para um contexto (atributo ***contextRef***), em uma unidade (atributo ***unitRef***), respeitando as definições e relações estabelecidas pela DTS do documento de instância. Os fatos são os principais componentes de um documento de instância, pois contêm as informações que devem ser representadas como objetivo principal de sua criação. A Figura 13 apresenta algumas declarações de fatos.

Figura 13 – Declaração de fatos

```
<dei:EntityCentralIndexKey
        contextRef="Context_FYE_30-Jun-2012">0001080627
</dei:EntityCentralIndexKey>

<us-gaap:DepositsAssetsNoncurrent
        contextRef="Context_As_Of_30-Jun-2012"
        unitRef="USD"
        decimals="0">2043750
</us-gaap:DepositsAssetsNoncurrent>

<cmdi:EffectiveBusinessTaxRateContinuingOperations
        contextRef="Context_FYE_30-Jun-2012"
        unitRef="pure"
        decimals="2">0.03
</cmdi:EffectiveBusinessTaxRateContinuingOperations>
```

- *Notas de Rodapé*: agregam informações não estruturadas ligadas a um fato. Possuem o atributo ***Lang*** para identificar o idioma.

2.2.5 Framework XBRL

O *framework* XBRL é um conjunto de especificações, cuja principal é a recomendação base (ENGEL et al., 2003). Além da recomendação base, o consórcio internacional da XBRL dá suporte a uma série de especificações adicionais, que estendem a recomendação base com funcionalidades complementares. As especificações complementares atualmente aprovadas pelo consórcio são:

- *XBRL Dimensions*: permitem às taxonomias XBRL definirem dimensões para serem utilizadas no elemento contexto de instâncias a fim de utilizar as informações em um modelo multidimensional (HERNÁNDEZ-ROS; WALLIS, 2012);
- *XBRL Formula*: permite o enriquecimento das informações sobre os conceitos, padronizando a declaração de regras de negócio. Também permite a geração de fatos novos baseado em fatos existentes na instância, a partir da criação de fórmulas sobre os conceitos (ENGEL et al., 2009);
- *XBRL Rendering*: padroniza a inclusão de tags XBRL em documentos HTML a fim de facilitar o consumo da informação dos documentos XBRL por humanos (ALLEN; STOKES-REES, 2011);
- *XBRL Versioning*: padroniza a comunicação de alterações em taxonomias XBRL, através de uma especificação de controle de versões (HOMMES; WARREN, 2013).

2.3 CONSIDERAÇÕES

Este capítulo descreveu as principais características da XML e da XBRL, necessárias para o entendimento do problema discutido nesta dissertação. No contexto da XML, foram destacadas as especificações *XML Schema* e *XLink*, por serem extensamente aplicadas à XBRL. Foi descrita a estrutura da XBRL e seus principais componentes: regras, conceitos e fatos. Discutiu-se com mais detalhes as especificações das taxonomias XBRL, por ser o local onde são criados os conceitos representados pela XBRL, alvo do processo proposto neste trabalho.

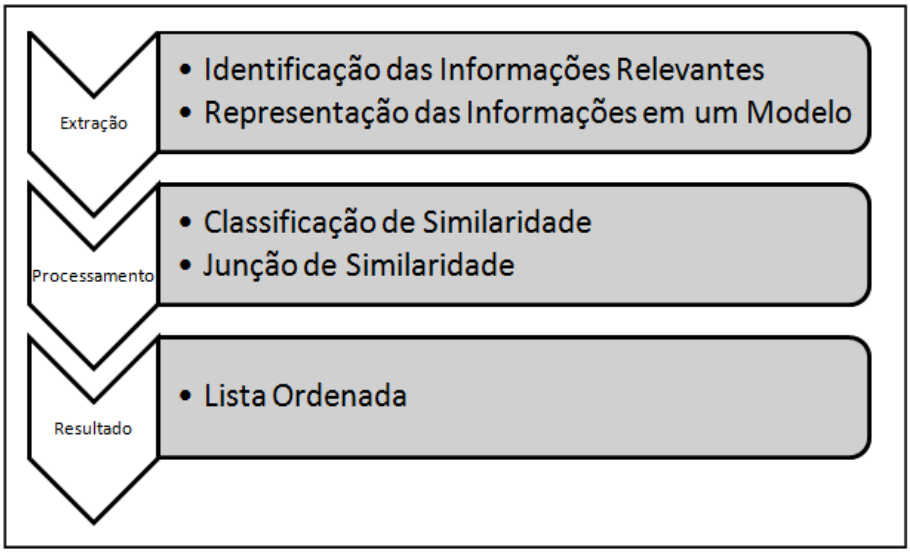
3 AVALIAÇÃO DE SIMILARIDADE

Neste capítulo discute-se o processo de avaliação de similaridade e suas atividades principais: classificação de similaridade e junção de similaridade. Na classificação de similaridade, são descritas técnicas para dados simples e complexos, com ênfase nas particularidades do processo de avaliação de similaridade aplicadas à XML. Na junção de similaridades, são descritas técnicas que objetivam aprimorar a eficiência do processo. Por fim, são descritas métricas para avaliação da eficácia do processo.

3.1 PROCESSO DE AVALIAÇÃO DE SIMILARIDADE

A avaliação de similaridade é o processo de comparar dados com o objetivo de classificar a semelhança entre eles. Recebe como entrada um conjunto de dados e o resultado é uma lista dos dados, ordenada pela semelhança entre eles. Existem diversas abordagens para o processo de avaliação de similaridade. Essencialmente, em todos os casos, inicia-se com a extração das informações relevantes ao processo. A seguir, são executadas as atividades principais: a classificação da similaridade, através de uma função de medida de similaridade e a junção por similaridade (*similarity join*), através de um algoritmo para comparação entre os dados. Por fim, o resultado é organizado com base nas atividades principais. A Figura 14 ilustra o processo e seus passos.

Figura 14 - Processo de avaliação de similaridade



São também relevantes ao processo de avaliação de similaridade questões referentes à preparação dos dados para análise (pré-processamento), métricas de qualidade e ferramentas de apoio.

O resultado do processo de avaliação de similaridade é, em geral, utilizado como entrada para atividades mais especializadas tais quais: limpeza de dados duplicados, registro de controle de versões e modificações, agrupamento de entidades (*clustering*), recuperação de informações, mineração de dados e outros.

3.2 CLASSIFICAÇÃO DE SIMILARIDADE

A classificação de similaridade deve determinar o grau de similaridade entre os dados. Isto é feito a partir da aplicação de uma ou mais funções de medidas de similaridade nos dados ou em subconjuntos dos dados, que irá determinar a semelhança entre eles.

A função de similaridade é comumente expressa como um número n variando entre 0 e 1. Quanto mais próximo do 1 reflete maior similaridade entre os campos, sendo o 1 utilizado para representar campos absolutamente iguais.

$$sim(e1, e2) = n, \quad 0 \leq n \leq 1$$

Geralmente, é determinado um limite θ do valor da medida para inclusão de duas entidades no resultado da avaliação de similaridade. Este limite varia a depender da aplicação, da medida utilizada, das características da base de dados e pode ser determinado pelo usuário ou automaticamente. O processo de avaliação de similaridade usa a medida de similaridade *sim* e o limite especificado para determinar se um par de elementos deve ou não ser incluído no resultado.

$$aval(e1, e2) \begin{cases} e1 \text{ e } e2 \text{ são similares} & \text{se } sim(e1, e2) > \theta \\ e1 \text{ e } e2 \text{ não são similares} & \text{caso contrário} \end{cases}$$

Os dados avaliados podem ser de vários tipos. Uma forma de identificar os tipos de dados é classificá-los como simples (dados atômicos sem relacionamentos) ou complexos (dados que se relacionam entre si) (ELMAGARMID et. al., 2007). Exemplos de dados simples são itens

de texto (nomes de pessoas, títulos de livros, descrições de produtos), blocos de texto, imagens e vídeos. Dados complexos podem estar armazenados de forma estruturada (modelos relacionais) ou semi-estruturadas (dados em formato XML).

Devido à grande variedade de tipos de dados que podem ser avaliados, existem várias técnicas para a função de medida de similaridade. A seguir, é descrito um breve resumo sobre as medidas mais relevantes para a compreensão do trabalho proposto. Maiores detalhes podem ser obtidos nas pesquisas de Elmagarmid e outros (2007) e Tekli e outros (2009), consultadas para este trabalho.

3.2.1 Medidas de similaridade para dados simples

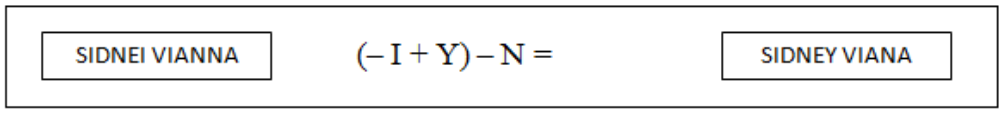
Ainda que o relacionamento entre os dados seja utilizado para a avaliação de similaridade, as medidas de similaridade para dados simples são utilizadas como base para qualquer processo de avaliação de similaridade. Na literatura, podemos encontrar medidas de similaridade para dados simples calculadas a partir do conceito de distância de edição, baseada em *tokens*, utilizando fonemas ou híbridas. Cada medida se adapta melhor a um determinado tipo de aplicação ou contexto.

3.2.1.1 Medidas de similaridade baseadas no conceito de distância de edição

O conceito de distância entre dois campos é caracterizado pela quantidade de operações de transformação que um campo deve sofrer para se igualar a outro. Quanto maior a distância entre dois campos, menor a sua similaridade. As medidas baseadas no conceito de distância utilizam a semelhança entre os caracteres dos campos que estão sendo comparados, levando em consideração a sua posição.

A medida proposta por Levenshtein (1966), calculada através de um algoritmo dinâmico, é a medida básica desta categoria. Ela computa o número mínimo de três operações: inserção, deleção e substituição de caracteres que deve ser aplicado aos campos comparados para que eles fiquem iguais. A Figura 15 ilustra a aplicação da medida de Levenshtein a dois nomes.

Figura 15 – Aplicação da distância de Levenshtein

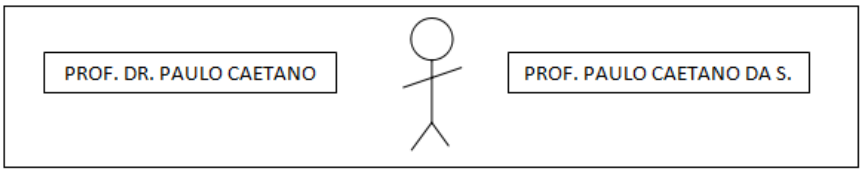


Pode-se transformar a medida de distância em medida de similaridade usando-se o conceito a seguir, gerando o resultado 0,7 para o exemplo da Figura 15.

$$\mathbf{Similaridade}(a, b) = 1 - \frac{\mathbf{DistanciaLevenshtein}(a, b)}{10}$$

Este tipo de medida funciona bem para erros tipográficos, ou de digitação, em que alguns caracteres são repetidos, esquecidos ou trocados. No entanto, esta medida falha quando grandes segmentos dos campos são diferentes como nos casos de abreviações e uso de títulos, conforme exemplo da Figura 16.

Figura 16 – Elementos duplicados contendo abreviação e título



A medida proposta por Smith e Waterman (1980) identifica o segmento comum mais longo entre os dois campos (*Paulo Caetano*, no exemplo) e determina pesos menores para diferenças no prefixo e sufixo dos campos. Para lidar com abreviações, denominadas *Gap*, a medida Affine-Gap estende as operações de edição da medida de Levenshtein acrescentando as operações de abrir *Gap* e estender *Gap* (WATERMAN et al., 1976). Estas novas operações possuem um custo menor do que se fossem consideradas as várias operações de deleção e inserção necessárias para considerá-las.

Jaro (1976) propôs uma medida que utiliza o conceito de transposição de caracteres em um algoritmo não dinâmico. Em primeiro lugar, ele computa o tamanho do campo, em seguida a quantidade de caracteres comuns aos dois campos e, por fim, o número de transposições que devem ser feitas para que os caracteres comuns fiquem na mesma posição. Para os caracteres serem considerados comuns, a diferença entre as suas posições deve ser menor que metade do

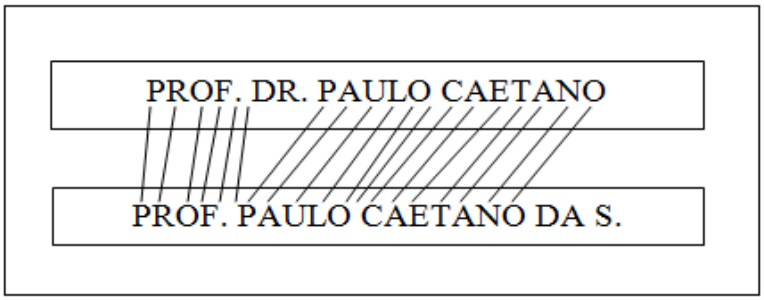
tamanho do campo mais curto. Formalmente, sendo α o conjunto de caracteres comuns e t a quantidade de transposições para que estes fiquem na mesma posição dentro do conjunto de caracteres comuns, a medida é calculada confirme fórmula a seguir:

$$SimJaro(a,b) = \frac{1}{3} \times \left(\frac{|\alpha|}{|a|} + \frac{|\alpha|}{|b|} + \frac{|\alpha| - 0{,}5t}{|\alpha|}\right)$$

Para o exemplo da Figura 16, a medida de similaridade seria calculada conforme equação a seguir. A identificação dos caracteres comuns é identificada na Figura 17.

$$\frac{1}{3} \times \left(\frac{17}{23} + \frac{17}{25} + \frac{17 - 0{,}5 \times 0}{17}\right) = 0{,}8$$

Figura 17 – Caracteres comuns



Uma extensão desta medida, que se revelou adequada para a comparação de sobrenomes, propõe a aplicação de pesos maiores nas operações no prefixo dos campos (WINKLER; THIBAUDEAU, 1991).

As medidas baseadas em distância funcionam de maneira eficaz para erros tipográficos. Entretanto, nos casos em que os campos possam sofrer rearranjo, como, por exemplo, posição do sobrenome e título em nomes de pessoas, estas medidas não funcionam muito bem por considerarem fortemente a posição dos caracteres para classificação de similaridade.

3.2.1.2 Medidas de similaridade baseadas em *tokens*

As medidas baseadas em *tokens* comparam os campos após dividi-los em partes menores que são subconjuntos do campo original. Os *tokens* podem ser obtidos usando um caractere delimitador (espaço, na maioria das vezes) ou a partir de um tamanho pré-definido. A ideia é

a de que se os campos são similares, então eles terão *tokens* em comum. Uma medida geral e básica para comparação entre os *tokens* é o Coeficiente de Jaccard.

$$Jaccard(s1, s2) = (\square|tokens(s1) \cap tokens(s2)|\square) \over (\square|tokens(s1) \cup tokens(s2)|\square)$$

Os *tokens* gerados para o exemplo da Figura 16 seriam: t1 = {‘PROF.’,’DR.’,’PAULO’, ‘CAETANO’} e t2 = {‘PROF.’, ‘PAULO’, ‘CAETANO’, ‘DA’, ‘S.’} e o resultado da medida de similaridade usando o coeficiente de Jaccard seria calculado conforme equação a seguir:

$$\frac{Jaccard(s1, s2) = (\square|\{'PROF.', \square'PAULO', \square'CAETANO'|\square)}{(\square|\{'PROF.', 'DR.', 'PAULO', 'CAETANO', \square'DA', \square'S.'\}|\square)} = \frac{3}{6} = 0,5$$

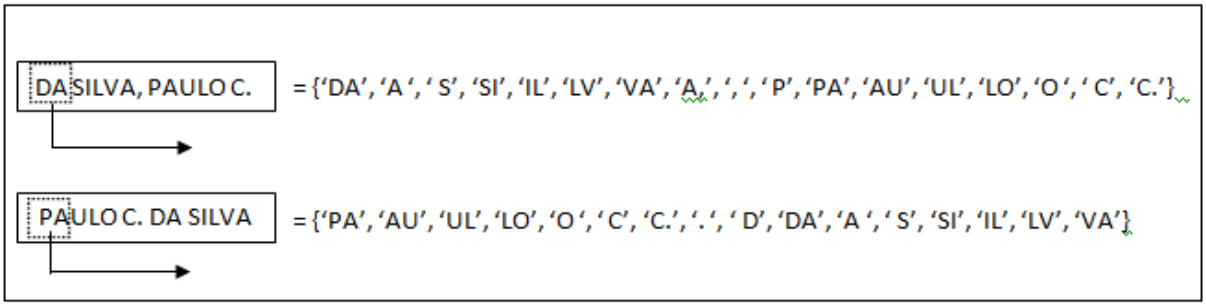
Outra forma tradicional de calcular a similaridade entre conjuntos de *tokens* é projetá-los em um espaço vetorial (*Space Vector Model – SVM*) e calcular o cosseno entre os dois vetores. Esta medida, conhecida como similaridade do cosseno (*Cosine Similarity*) é muito utilizada na área de recuperação de informações (*Information Retrieval* - IR) (BAEZA-YATES; RIBEIRO-NETO, 1999).

Baseados no coeficiente de Jaccard, Monge e Elkan (1996) apresentam o conceito de *atomic strings*, formadas a partis de tokens obtidos utilizando pontuações e espaços como delimitador de caractere. Duas *atomic strings* são consideradas compatíveis se são iguais ou se uma é o prefixo da outra. A similaridade entre os campos é obtida pela divisão de *atomic strings* compatíveis pela média de *atomic strings* obtidas.

Cohen (1998) propôs um sistema de cálculo de similaridade, chamado WHIRL (*Word-Based Heterogeneous Information Retrieval Logic*) baseado na similaridade do cosseno. Os *tokens* são obtidos através de palavras. A medida usa o conceito de *Term Frequency – Inverse Document Frequency* (TF-IDF) para determinar pesos aos *tokens* obtidos e projetá-los em um espaço vetorial representando as *strings*, para em seguida calcular o cosseno. A medida considera que os termos mais frequentes em todos os campos sendo avaliados são menos relevantes para se determinar a similaridade entre eles.

A medida de similaridade QGrams (UKKONEN, 1992) usa como *tokens* substring do campo de tamanho pré-definido. As substring são obtidas a partir do deslocamento de uma janela de tamanho k por todo campo a ser comparado. Um exemplo é apresentado na Figura 18.

Figura 18 – Tokens gerados pelo método QGrams



Sendo t1 o conjunto de *tokens* obtidos para o elemento e*1* (primeiro nome da Figura 18) e *t2* o conjunto de *tokens* obtidos para o elemento *e2* (segunda string da Figura 18), o coeficiente de similaridade é calculado a partir de quantos q-grams em comum os dois campos possuem, como a seguir:

$$t1 \cap t2 = \{\text{'DA', 'A ', ' S', 'SI', 'IL', 'LV', 'VA', 'PA', 'AU', 'UL', 'LO', 'O ', ' C', 'C.'}\}$$

$$t1 \cup t2 = \{\text{'DA', 'A ', ' S', 'SI', 'IL', 'LV', 'VA', 'A,', ', ', ' P', 'PA', 'AU', 'UL', 'LO', 'O ', ' C', 'C.', '. ', ' D'}\}$$

$$SimQGrams(e1, e2) = \left(|t1 \cap t2|\right) / \left(|t1 \cup t2|\right) = \frac{14}{20} = 0.7$$

A vantagem do uso de *tokens* é que eles são insensíveis a mudanças de posição, entretanto, erros tipográficos dentro dos *tokens* podem comprometer a medida.

3.2.1.3 Medidas de similaridade baseadas em fonemas

As medidas de similaridade baseadas no conceito de distância de edição e em *tokens* falham ao detectar campos que são foneticamente semelhantes, mas escritos de maneira diferente. As medidas de similaridade baseadas em fonemas levam em consideração o som dos caracteres. O princípio básico é o de substituir caracteres com sons iguais pela mesma representação antes de efetuar a comparação entre os campos.

Este tipo de substituição é, por natureza, sensível à língua sendo representada. Para o inglês, existem as técnicas mais conhecidas de Soundex[5] e *New York State Identification and Intelligence System* - NYSIIS (TAFT, 1970). Para o português, existe o BuscaBR (LUCENA, 2006). O BuscaBR segue as seguintes regras:

1. Converter todas as letras para Maiúsculo;
2. Eliminar acentos das vogais;
3. Substituir letras conforme Quadro 1;
4. Eliminar as letras M, R e S ao final das palavras;
5. Eliminar todas as vogais e a letra H.

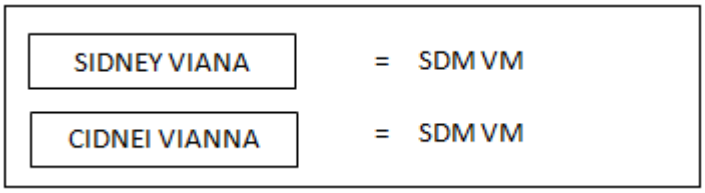
Quadro 1 – Quadro de substituições do BuscaBR

DE	PARA	DE	PARA	DE	PARA
BL, BR	B	L	R	RM	SM
CA	K	N, MD	M	RJ	J
CE, CI	S	MG	G	ST, TR, TL	T
CO, CU, CK	K	MJ	J	TS	S
Ç, CH	S	PH	F	W	V
CT	T	PR	P	X	S
GE, GI	J	Q	K	ST	T
GM	M	RG	G	Y	I
GL, GR	G	RS	S	Z	S
		RT	T		

A Figura 19 **Erro! Fonte de referência não encontrada.**demonstra a aplicação do algoritmo BuscaBR em um exemplo.

[5] Patente de RUSSEL, R. e ODELL, M. n.01 261 167, EUA, 1918.

Figura 19 – Aplicação do algoritmo BuscaBR



O exemplo da Figura 19 demonstra que dois elementos escritos de maneiras diferentes podem ser considerados iguais, após a aplicação da transformação fonética.

3.2.1.4 Medidas de similaridade híbridas

As medidas de similaridade híbridas propõem comparar os campos usando caracteres, *tokens* e fonemas. Desta forma, pode-se detectar erros fonéticos, erros tipográficos e também abreviaturas e deslocamento de *tokens* dentro da string. Uma forma de se aplicar técnicas distintas é utilizar medidas de similaridade diferentes em sequência e calcular a média entre elas para se obter o resultado final, por exemplo (BILENKO et al., 2003). Outra forma é incorporar em uma medida conceitos diferentes, como relacionadas a seguir.

Ananthakrishna e outros (2002), estendeu a medida de similaridade de Jaccard para considerar não somente os *tokens* iguais como também os *tokens* similares, segundo alguma medida básica baseada em caracteres.

Na medida Soft TF/IDF (BILENKO et al, 2003), é proposta a utilização de tokens similares (e não somente iguais) no cálculo do cosseno da medida WHIRL, enquanto que Gravano e outros (2003) propôs a utilização de QGrams ao invés de *tokens* para estender esta mesma medida.

3.2.2 Medidas de similaridade na presença de relacionamentos

Os estudos em avaliação de similaridade, inicialmente, levavam em consideração apenas dados simples. Em seguida sugiram estudos para avaliação de similaridade na presença de relacionamentos para modelos relacionais. Ananthakrishna e outros (2002) apresenta uma abordagem *top-down* para a detecção de elementos duplicados em data warehouses com

relações hierárquicas, baseada no conceito de que a similaridade entre dois elementos depende da similaridade entre seus elementos-filho.

A avaliação de similaridade em XML introduz alguns desafios em relação ao modelo relacional devido a suas características particulares. Em bases XML, além das diferenças de conteúdo existentes no modelo relacional, podem existir diferenças na estrutura de elementos que representam a mesma entidade. Mesmo documentos que sigam as definições de um único *schema* XML podem representar elementos de maneiras diferentes devido às propriedades de opcionalidade e cardinalidade de seus atributos.

Em XML, elementos podem estar relacionados entre si por meio da própria estrutura hierárquica do documento, formando uma estrutura em árvore, ou ainda fazer uso de ligações e referências (*keyrefs e linkbases*), formando uma estrutura em grafo. Estas estruturas também são encontradas no modelo relacional, entretanto, em XML, a identificação dos tipos dos elementos relacionados entre si é mais complexa do que a que é feita por meio da utilização de chaves estrangeiras no modelo relacional, pois estas informações estão embutidas nos documentos. Para se avaliar a similaridade em XML, deve-se levar em consideração não só o conteúdo dos elementos, como também a estrutura. Os estudos para avaliação de similaridade em XML procuram acomodar estas particularidades.

As técnicas de avaliação de similaridade em XML podem ser classificadas em três grupos, segundo (TEKLI et al., 2009):

- *Métodos de Distância de Edição*: utilizam a estrutura de dados de árvore ordenada para representar os documentos XML e aplicam técnicas de distância de edição de arvores (TED – *Tree Edit Distance*) para calcular a semelhança. São considerados mais apropriados para documentos bem estruturados, que sigam as definições de um *schema* (TEKLI et al., 2009).

- *Métodos de Recuperação de Informações (IR – Information Retrieval)*: utilizam vetores para representar documentos XML e aplicam técnicas de IR aos vetores para calcular a semelhança. Focados em conteúdo, aplicam pesos aos termos segundo a frequência em que eles aparecem no documento, conforme técnica de frequência de termos *Term Frequency-Inverse Document Frequency* (TF-IDF) utilizada na área de *Information Retrieval* (BAEZA-YATES; RIBEIRO-NETO, 1999).

- *Outros Métodos*: utilizam estruturas diversas para representar os documentos XML.

Carvalho e Silva (2003) representam o conteúdo de documentos XML em um modelo vetorial e apresenta quatro estratégias diferentes para definição da função de similaridade sobre este modelo, considerando que houve um mapeamento prévio entre elementos estruturalmente diferentes, mas semanticamente iguais.
O *framework* DogmatiX (WEISS; NAUMANN, 2005) define três passos para a avaliação de similaridade entre documentos XML baseado no primeiro modelo proposto na área (FELLEGI; SUNTER, 1969). O primeiro passo é a definição de elementos do documento XML que serão representados em tuplas. O segundo passo é a criação das tuplas com os dados dos elementos definidos no primeiro passo. O terceiro passo é a aplicação de uma medida de similaridade proposta sobre as tuplas, baseada em medidas de avaliação de textos. O framework DogmatiX considera que a avaliação de similaridade será aplicada a elementos de um único documento XML que segue as definições de um único *schema*.

Guha e outros (2006) e Ribeiro e Harder (2007) representam elementos XML em uma estrutura de árvore ordenada nomeada (*ordered labeled tree*). Guha e outros (2006) aplica o conceito de distância de edição em árvore (*tree-edit distance*) para avaliar a similaridade estrutural entre elementos XML. Esta medida computa a quantidade mínima de inserções, deleções e alterações que devem ser feitas em duas árvores para que elas fiquem iguais. Ribeiro e Harder (2007) aplicam o conceito de *pq-grams* para transformar a árvore ordenada em um conjunto de subconjuntos obtidos por meio do deslocamento de uma janela de tamanho pré-definido sobre os nós da árvore, tal qual definido em Augsten e outros (2005). Em seguida, aplica medidas conhecidas de avaliação de similaridade sobre conjuntos no resultado desta transformação.

Kade e Heuser (2008) propõem a decomposição de documentos XML em subárvores em um processo *top-down*. Cada subárvore gera uma tupla com o caminho da subárvore (representando a estrutura) e uma string concatenando todas as folhas da subárvore (representando o conteúdo). Em seguida, as tuplas são comparadas para calcular sua similaridade.

Os caminhos (*paths*) dos elementos são considerados informações relevantes em muitas propostas de avaliação de similaridade entre documentos XML (KADE; HEUSER, 2008);

(WEISS; NAUMANN, 2005). A avaliação de similaridade de caminhos é um processo importante na avaliação de similaridade de documentos XML. Vinson (2007) propõe um algoritmo (*PathSim*) para avaliação de similaridade de caminhos, baseado na medida de distância de Levenshtein (1966) e adaptado pelo autor a caminhos de XML e comprova a eficiência do algoritmo proposto.

3.2.3 Combinação de medidas de similaridade

Muitos processos de avaliação de similaridade obtêm valores de similaridade diferentes para o mesmo par de elementos sendo avaliados. Isto pode ocorrer quando são aplicadas técnicas diferentes ao mesmo par (por exemplo, avaliação de descendentes ou avaliação de parentes de mesma estrutura de caminhos), ou quando são avaliados atributos diferentes dos elementos (por exemplo, nome e endereço de pessoas). Nestes casos, deve-se combinar os resultados para obter um valor único de similaridade entre os elementos sendo avaliados.

Existem várias formas de combinação de resultados. A média aritmética entre elas é uma delas, utilizada quando os resultados individuais pertencem à mesma categoria e não podem ser diferenciados entre si, contribuindo de forma igual na avaliação de similaridade total (BILENKO et al., 2003); (COHEN, 1998).

Pode-se também obter os maiores valores (os mais similares) e fazer a média utilizando-se somente eles. Esta maneira se revela eficiente quando a variação entre os valores representa sua maior ou menor adequação ao domínio, indicando que somente os mais adequados devem ser considerados na combinação de resultados (COHEN, 1998).

Outra forma é utilizar o conhecimento do especialista a fim de calcular a média ponderada dos valores individuais, seja por meio da definição dos pesos a serem aplicados, ou por meio da definição de valores de similaridade para pares de elementos de um subconjunto de treinamento que será utilizado para determinar os pesos através de algoritmo próprio para este fim (ELMAGARMID et al., 2007).

A 3.3 a seguir apresenta um quadro resumindo os conceitos discutidos para a classificação de similaridade.

Figura 20 – Resumo classificação de similaridade

- ✓ **CLASSIFICAÇÃO DE SIMILARIDADE**
  - ✓ **Medidas de Similaridade para Dados Atômicos**
    - ✓ *Distância de Edição*: Levenshtein; Smith e Waterman; Affine-Gap; Jaro e Winkler
    - ✓ *Tokens*: Jaccard; Cosseno; Atomic Strings; WHIRL; QGrams
    - ✓ *Fonemas*: Soundex; NYSIIS; BuscaBR
  - ✓ **Medidas de Similaridade para Dados na Presença de Relacionamentos**
    - ✓ *Distância de Edição*: Tree Edit Distance; pq-Grams
    - ✓ *Vetoriais*: Carvalho e Silva
    - ✓ *Outros*: top-down; Dogmatix
  - ✓ **Combinação de Medidas de Similaridade**
    - ✓ *Média Aritmética:* visão geral
    - ✓ *Máxima Medida:* visão focada em similaridade
    - ✓ *Média Ponderada:* visão do especialista

3.3 JUNÇÃO DE SIMILARIDADE

Após a definição de como será feita a classificação de similaridade, o processo de avaliação de similaridade deve efetuar a junção entre todos os pares de dados para efetuar a classificação.

A abordagem mais elementar para a junção de similaridade em um conjunto de dados é a comparação de cada item de dado com todos os outros, avaliando o nível de similaridade de cada par. Considerando que o conjunto de dados tenha *n* elementos, cada elemento seria comparado a (*n*-1) elementos, conforme Figura 21, levando a uma complexidade de O(n2) (BILENKO et al., 2006). Esta complexidade pode inviabilizar a comparação em conjuntos com grande quantidade de elementos.

Figura 21 – Comparação de elementos

A fim de diminuir a quantidade de comparações, geralmente é adotada uma estratégia com dois passos: em primeiro lugar, conjuntos de dados são reunidos em grupos através do uso de uma operação mais simples para, em seguida, aplicar a função de medida de similaridade somente entre os grupos formados.

Uma estratégia tradicional de redução de comparações é a de blocagem (ANANTHAKRISHNA et al., 2002; BAXTER et al., 2003; BILENKO et al., 2006), na qual o conjunto de dados é dividido em subconjuntos a partir de uma regra, e a comparação é feita somente dentro de cada subconjunto. O principal problema desta proposta é a definição da regra da subdivisão. Caso seja muito genérica, o número de subconjuntos pode ser insuficiente para melhorar o desempenho do processo de avaliação de similaridade. Caso seja muito específica, pode custar a não detecção de verdadeiros similares, que estariam em subconjuntos diferentes.

A proposta de vizinhos ordenados (*sorted neighborhoods*) (HERNANDEZ; STOLFO, 1998) utiliza uma janela de tamanho fixo. Inicialmente, todos os dados são ordenados segundo uma chave, criada a partir de uma regra simples e que admite valores iguais. Em seguida, uma janela de tamanho fixo w é deslizada pelos registros e são feitas comparações entre cada registro da janela. O último registro comparado participa da próxima rodada de comparações da janela. Analogamente ao método de blocagem, neste método há o problema de definição da regra de ordenação dos registros para deslocamento da janela.

A técnica de *multipass* pode ser aplicada em métodos de ordenação ou blocagem a fim de aumentar a eficiência da avaliação de similaridade. Nesta técnica, podem ser escolhidas e aplicadas regras diferentes múltiplas vezes. A escolha das chaves ganha menos relevância,

visto que a fragilidade de uma chave pode ser compensada por outra, aumentando assim a probabilidade de detecção de similaridades.

3.4 MÉTRICAS DE QUALIDADE

O que se deseja de um processo de avaliação de similaridade é que seu resultado classifique corretamente a similaridade entre os dados e que todos os dados similares sejam encontrados, correspondendo às qualidades de correção e completude do processo, indicando sua eficácia. As métricas mais utilizadas para avaliação da eficácia do processo de avaliação de similaridade são precisão e evocação. Estas métricas são originalmente utilizadas para a área de IR (*Information Retrieval*) (BAEZA-YATES; RIBEIRO-NETO, 1999) e adaptadas ao contexto de avaliação de similaridade.

Para se chegar aos valores de precisão e evocação, é preciso definir quatro categorias de pares de elementos da base de dados: pares verdadeiramente similares existentes na base de dados (similares existentes), pares similares detectados no processo (similares declarados), pares similares detectados no processo que são verdadeiramente similares (similares verdadeiros) e pares similares detectados no processo que não são verdadeiramente similares (similares falsos).

$$\text{Precisão} = \frac{|\text{similares verdadeiros}|}{|\text{similares declarados}|}$$

$$\text{Evocação} = \frac{|\text{similares verdadeiros}|}{|\text{similares existentes}|}$$

O estabelecimento do limite (θ) para que dois campos sejam classificados como similares é um fator preponderante para a eficácia do processo. Um limite muito alto garante que os campos cuja similaridade foi detectada são realmente muito parecidos (ou mesmo iguais), aumentando a precisão, mas deixa sem detectar campos similares não tão parecidos, diminuindo a evocação. Um limite mais baixo pode capturar um maior número de pares, aumentando a evocação, mas também mais falsos similares, diminuindo a precisão. O ideal é um equilíbrio entre as duas métricas.

A fim de medir o equilíbrio entre as duas métricas em um valor, pode-se usar uma média harmônica, chamada de F-Measure, onde:

$$\mathbf{F-Measure} = \frac{2 \times \mathbf{Evocação} \times \mathbf{Precisão}}{\mathbf{Evocação} + \mathbf{Precisão}}$$

3.5 CONSIDERAÇÕES

Este capítulo discutiu o processo de avaliação de similaridade e suas principais atividades. Foram descritas técnicas para classificação de similaridade encontradas na literatura a fim de facilitar o entendimento das escolhas das técnicas que serão aplicadas ao processo proposto, como detalhado no capítulo a seguir.

4 AVALIAÇÃO DE SIMILARIDADE ENTRE CONCEITOS REPRESENTADOS PELA XBRL

Este capítulo tem por objetivo indicar quais são as informações utilizadas na avaliação de similaridade entre conceitos representados pela XBRL, tanto no que se refere ao conteúdo quanto na sua estrutura. Também são detalhadas as medidas de similaridade que podem ser utilizadas para cada informação, além de serem avaliadas as mais adequadas e indicadas as possibilidades de combinação dos resultados individuais de cada medida de similaridade obtida a fim de se chegar a uma avaliação combinada. Com base nas medidas adotadas, é apresentado um processo de avaliação de similaridade com as informações sobre a sua construção: suas atividades e a estrutura de dados montada. Por fim, são discutidos trabalhos de Gestão da Qualidade de Dados dentro do contexto da linguagem XBRL.

4.1 IDENTIFICAÇÃO DAS INFORMAÇÕES RELEVANTES PARA A AVALIAÇÃO DE SIMILARIDADE ENTRE CONCEITOS REPRESENTADOS PELA XBRL

O primeiro passo no processo de avaliação de similaridade é a definição de quais informações são relevantes ou não ao processo. Os conceitos representados pela XBRL são definidos nos arquivos das taxonomias. Possuem atributos que, sob o ponto de vista do processo de avaliação de similaridade, definem o seu conteúdo. As taxonomias também especificam relacionamentos entre os diversos conceitos que, neste trabalho, são consideradas a informação que reflete a estrutura dos conceitos, dentro do processo de avaliação de similaridade.

4.1.1 Informações de conteúdo

O conteúdo dos conceitos deve ser utilizado na classificação de semelhança entre eles. A avaliação de similaridade de conteúdo entre dois conceitos deve considerar as suas propriedades. Podemos classificar estas propriedades do conceito como conteúdo simples ou multivalorado, segundo suas características, as quais são discutidas a seguir.

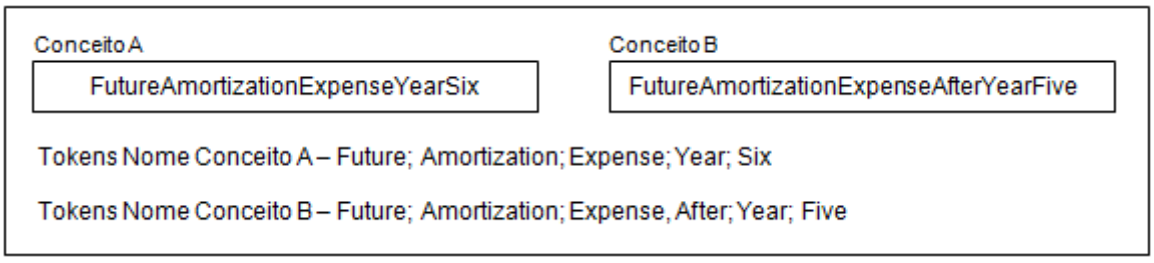
4.1.1.1 Conteúdo Simples

Os atributos obrigatórios (*name*, *type* e *substitutionGroup*) são, por relevância, as primeiras opções de conteúdo a serem consideradas. Demais atributos opcionais na definição dos conceitos podem também ser considerados como informações relevantes para a avaliação de similaridade. Estes atributos possuem apenas um valor na sua definição. A comparação individual entre eles depende de uma medida de similaridade para dados textuais simples.

A recomendação XBRL sugere que o valor do atributo obrigatório *name* e, consequentemente, do atributo *id* deve ser indicativo do conceito financeiro que ele irá representar. Normalmente, pelo que se pode observar nas taxonomias publicadas, o nome é composto por várias palavras, podendo haver transposição, abreviações ou mesmo erros tipográficos. Nestes casos, as medidas mais eficientes são as medidas baseadas em *tokens*, que não são influenciadas pelo posicionamento dos caracteres.

Uma maneira de se obter os *tokens* é através da técnica de *QGrams*. Outra forma é a utilização de letras maiúsculas como delimitadores dos *tokens*, visto que este é um padrão amplamente utilizado na indústria para separar palavras em atributos, como ilustrado no exemplo da Figura 22.

Figura 22 – Transformação de nomes de conceitos em Tokens



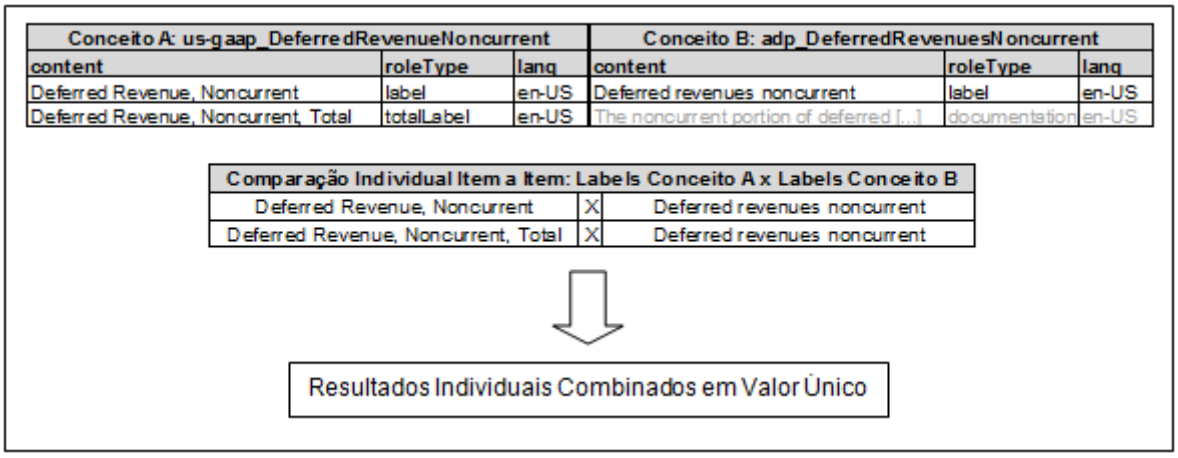
O processo de avaliação de similaridade proposto por este trabalho aplica medida de similaridade de *token* entre os nomes dos pares de conceitos sendo avaliados utilizando duas formas de obtenção de *tokens*: *QGrams* e palavras delimitadas por letras maiúsculas.

4.1.1.2 Conteúdo Multivalorado

Informações contidas nos *linkbases label* e *reference* podem ser consideradas na avaliação de similaridade, como atributos multivalorados. Um conceito pode estar relacionado a mais de um elemento *label* ou *reference* que, por sua vez, irão possuir seus atributos próprios. A comparação entre atributos multivalorados dos conceitos deve ser obtida através da combinação da comparação de todos os seus itens, individualmente. Isto significa dizer que se o processo de avaliação de similaridade entre conceitos for considerar seus atributos multivalorados, deve definir um processo para avaliação individual destes atributos também.
Os *resources* do tipo *label* possuem *type, role, lang* e *label* como atributos obrigatórios, além do seu próprio conteúdo. Para efeito de avaliação de similaridade, foram desconsiderados os *resources* do tipo *label* que servem para documentação (cujo *roleType* é *documentation*), visto serem descritos por meio de um texto livre dissertativo que poderia distorcer o processo por apresentar muitas falsas dessemelhanças a depender do estilo de escrita dos autores das taxonomias. O conteúdo dos recursos do tipo *label* representa a informação que o link precisa associar ao conceito, sendo assim, foi a informação utilizada na avaliação de similaridade.

A Figura 23 apresenta um exemplo de comparação entre *labels* de dois conceitos distintos, mas semelhantes. A primeira tabela relaciona os resources do tipo *label* de cada conceito. A segunda tabela demonstra a comparação individual que será feita pelo processo de avaliação de similaridade e que servirá de subsídio para a combinação dos resultados em um único valor de similaridade.

Figura 23 – Comparação entre *links label*



O elemento *reference* possui os mesmos atributos do elemento *label*, ambos do tipo *resource*, entretanto, em vez da informação ser expressa como texto embutido na *tag*, seu conteúdo é formado por meio de subelementos do tipo *part*. O elemento *part* é abstrato e pode ser substituído na taxonomia por outros elementos que detalhem a referência, por meio do atributo *substitutionGroup* na definição desses elementos. A informação mais relevante para a avaliação de similaridade do tipo *reference* é a junção dos valores de todos os seus elementos do tipo *part*, visto que eles são o conteúdo da referência.
A classificação de similaridade entre as informações relevantes dos *linkbases label* e *reference* pode ser feita através de medidas de *token*, de maneira análoga à utilizada nos atributos simples.

Os *links* do mesmo tipo devem ser comparados entre si, ou seja, todos os *links* do tipo *label* de um conceito devem ser comparados a todos os *links* do tipo *label* do outro conceito, assim como os *links* do tipo *reference*. No processo proposto neste trabalho, foi utilizada a média aritmética como forma de se combinar os resultados das comparações individuais para se obter um valor único, devido ao fato que os itens sendo combinados pertencem à mesma categoria (*links label* ou *reference*) e terem igual importância na avaliação de similaridade.

4.1.2 Informações de estrutura

A estrutura de um conjunto de conceitos é obtida através das informações do relacionamento entre conceitos, através dos *linkbases calculation*, *presentation* e *definition*. A partir dos linkbases *calculation* e *presentation*, pode-se obter uma rede de relacionamento hierárquica não cíclica, bem definida. Os *linkbases definition* são diversos, permitem todo tipo de ciclo e representam várias redes de relacionamento (e.g. de especialização, de domínio, de dependência e dimensional). A fim de aplicar as técnicas existentes para abordagens hierárquicas, foram utilizadas neste trabalho somente as informações presentes nos *linkbases calculation* e *presentation*.

Ao serem discutidas as técnicas de avaliação de similaridade de estrutura, é importante observar que elas podem ser aplicadas tanto aos *linkbases calculation* quanto aos *linkbases presentation*. Apesar dos *linkbases calculation* e *presentation* representarem informações diferentes, as suas definições são análogas e pode-se aplicar a mesma técnica de avaliação de similaridade em ambos, um de cada vez. A seguir, os resultados devem ser combinados para

obtenção de um valor único de similaridade. Foi utilizado neste trabalho o método de média aritmética para combinação dos valores

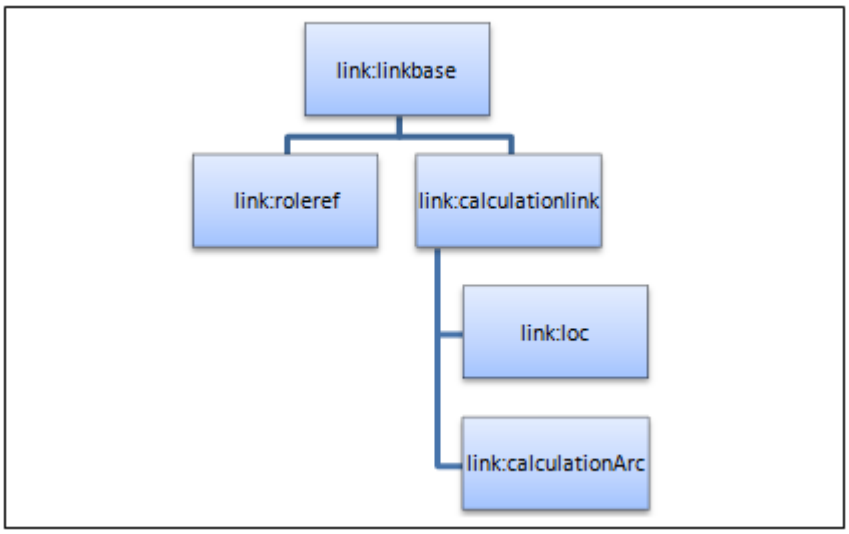
Enquanto a definição dos conceitos XBRL é estruturada e restrita a um *schema* bem definido, a transformação destes conceitos em uma estrutura hierárquica baseada na sua rede de relacionamentos gera estruturas variadas, que podem não ter qualquer semelhança entre si, conforme ilustrado no exemplo a seguir. A Figura 24 contém um trecho da definição de um *linkbase calculation* da taxonomia US-GAAP, em que são utilizados os elementos padrão das linguagens XML, XLink e XBRL.

Figura 24 – Definição de linkbase

```
<link:linkbase xmlns:link='http://www.xbrl.org/2003/linkbase' xmlns:xbrldt='http://xbrl.org/2005/xbrldt' xmlns:xlink='http
://www.w3.org/1999/xlink' xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xsi:schemaLocation='http://www.xbrl.org
/2003/linkbase http://www.xbrl.org/2003/xbrl-linkbase-2003-12-31.xsd'>
  <link:roleRef roleURI='http://fasb.org/us-gaap/role/statement/StatementOfFinancialPositionClassified' xlink:href='..
/elts/us-roles-2012-01-31.xsd#sfp-cls' xlink:type='simple' />
  <link:calculationLink xlink:role='http://fasb.org/us-gaap/role/statement/StatementOfFinancialPositionClassified' xlink
:type='extended'>

    <link:loc xlink:href='../elts/us-gaap-2012-01-31.xsd#us-gaap_Assets' xlink:label='loc_Assets' xlink:type='locator' />
    <link:loc xlink:href='../elts/us-gaap-2012-01-31.xsd#us-gaap_AssetsCurrent' xlink:label='loc_AssetsCurrent' xlink:type
='locator' />
    <link:loc xlink:href='../elts/us-gaap-2012-01-31.xsd#us-gaap_AssetsNoncurrent' xlink:label='loc_AssetsNoncurrent'
 xlink:type='locator' />

    <link:calculationArc order='10' use='optional' weight='1.0' xlink:arcrole='http://www.xbrl.org/2003/arcrole/summation
-item' xlink:from='loc_Assets' xlink:to='loc_AssetsCurrent' xlink:type='arc' />
    <link:calculationArc order='20' use='optional' weight='1.0' xlink:arcrole='http://www.xbrl.org/2003/arcrole/summation
-item' xlink:from='loc_Assets' xlink:to='loc_AssetsNoncurrent' xlink:type='arc' />

  </link:calculationLink>
</link:linkbase>
```

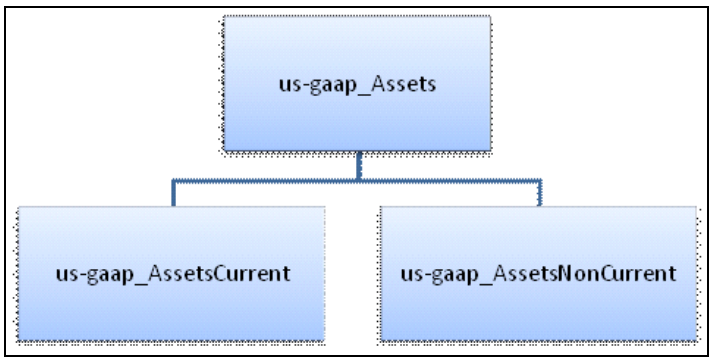
Os elementos da definição do *linkbase calculation* estão definidos e estruturados através das restrições da especificação base da XBRL e sempre podem ser representados através da estrutura ilustrada na Figura 25, em qualquer taxonomia.

Figura 25 – Representação da estrutura dos elementos da definição do linkbase



Entretanto, ao transformar as informações da definição do *linkbase* na estrutura de relacionamentos entre os conceitos que ele representa, obtém-se a estrutura ilustrada na Figura 26. As estruturas de relacionamento entre conceitos obtidas através das informações dos *linkbases* são variadas e não podem ser consideradas bem estruturadas, pois variam de uma taxonomia para a outra.

Figura 26 – Representação dos relacionamentos

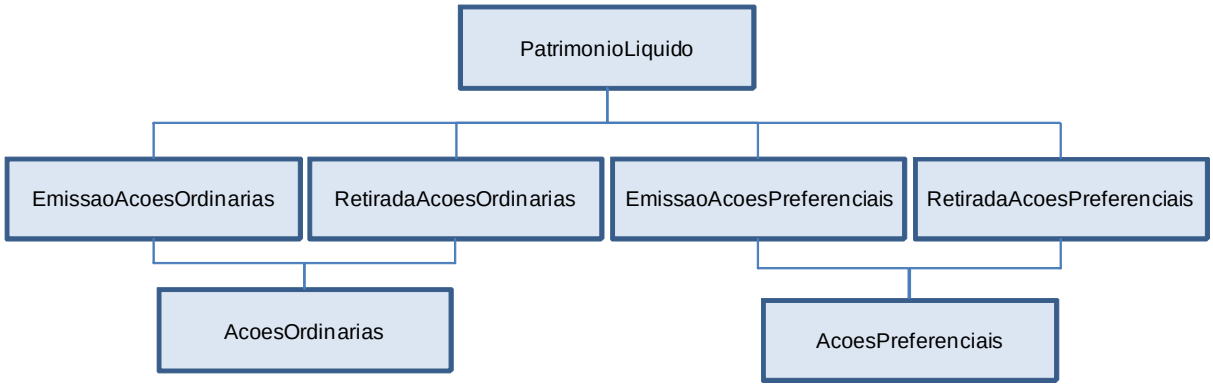


Relacionamentos não estruturados não se adéquam bem às medidas de distância (ALGERGAWY et al., 2011; TEKLI et al., 2009). Além da falta de estrutura comum, deve ser observado o fato de que os relacionamentos *calculation* e *presentation*, apesar de não permitirem ciclos, não necessariamente podem ser representados como uma árvore, dificultando a aplicação de técnicas de TED (*Tree Edit Distance*). São estruturas hierárquicas, criada através de relações pai-filho.

O exemplo da Figura 27 ilustra uma rede de relacionamentos do tipo *calculation* entre conceitos representando movimentos de capital que não formam uma árvore. Os conceitos

básicos são os de retirada e emissão de ações (ordinárias e preferenciais). Os conceitos calculados são os de patrimônio líquido e os valores líquidos das ações ordinárias e preferenciais.

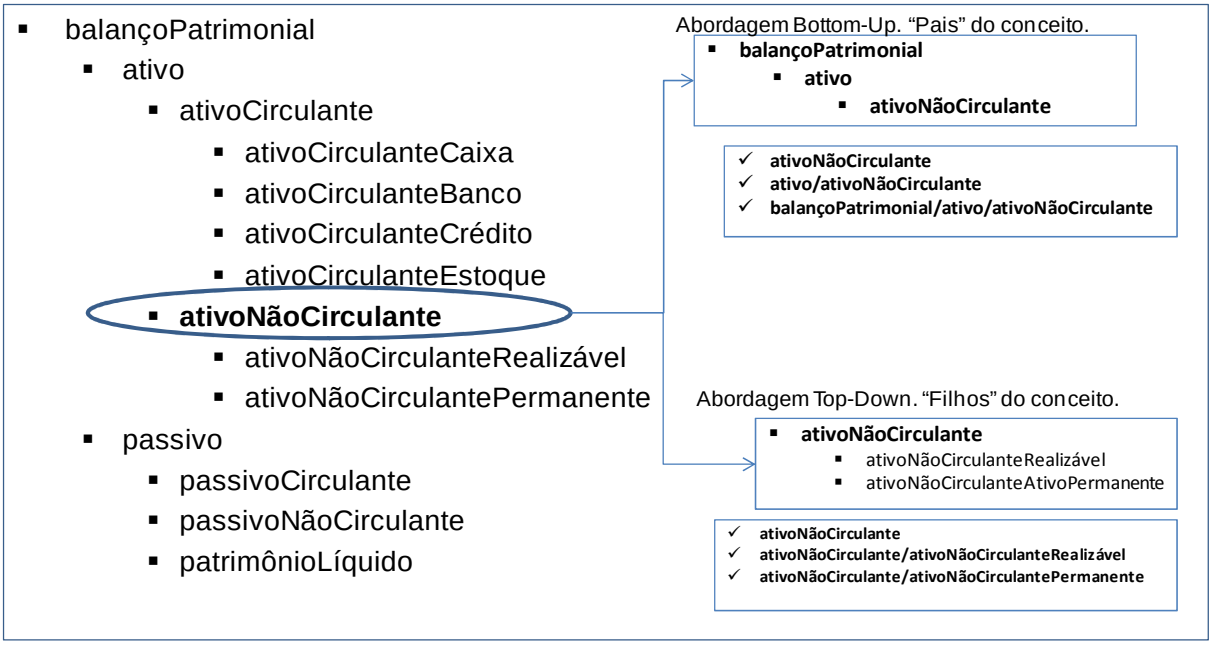
Figura 27 – Relacionamento *calculation* que não forma árvore



As técnicas ligadas à área de IR (*Information Retrieval*) que têm o foco em conteúdo e utilizam pesos baseados na frequência de utilização de palavras também não se revelam adequadas, pois o objetivo é avaliação de similaridade de estrutura sobre as redes de relacionamentos *calculation* e *presentation*.

Muitas técnicas com foco na semelhança estrutural se utilizam do caminho (*path*) dos elementos da estrutura no processo de avaliação de similaridade (ALGERGAWY et al., 2011). O caminho de cada elemento é obtido e este conjunto de caminhos é comparado entre si. Para efeito de comparação, podem ser obtidos os caminhos do elemento e de seus filhos (abordagem *top-down*) ou do elemento e seus parentes (abordagem *bottom-up*) ou uma combinação de ambos. A Figura 28 ilustra a extração de “Pais” (abordagem bottom-up) e “Filhos” (abordagem top-down) de um conceito a partir da representação hierárquica de uma estrutura de relacionamento de conceitos.

Figura 28 – Extração de “Pais” e “Filhos” de um conceito



A fim de determinar a semelhança entre dois caminhos, pode-se utilizar qualquer medida de similaridade de texto. Neste trabalho, optou-se por utilizar o *PathSim*, por ser um algoritmo comprovadamente eficiente em qualquer contexto, como demonstrado no trabalho de Vinson (2007).

4.1.3 Combinação de conteúdo e estrutura

As medidas obtidas no cálculo de similaridade de conteúdo e estrutura dos conceitos devem ser combinadas para se chegar a um valor único de semelhança entre os conceitos. Da mesma forma que na combinação dos resultados intermediários de cada informação (conteúdo e estrutura) pode-se utilizar a média aritmética entre eles, escolher o maior valor (mais representativo da semelhança) ou se determinar pesos para obtenção de uma média ponderada.

Neste trabalho, optou-se por incluir nos resultados dois tipos de combinação dos valores de similaridade de conteúdo e estrutura: média aritmética e máxima similaridade entre os dois, visto que são as combinações possíveis de serem feitas sem a necessidade de se atribuir pesos. A combinação feita por meio da aplicação de pesos ficou pendente para um trabalho futuro, pois necessita do envolvimento de especialistas para se determinar o peso de cada medida.

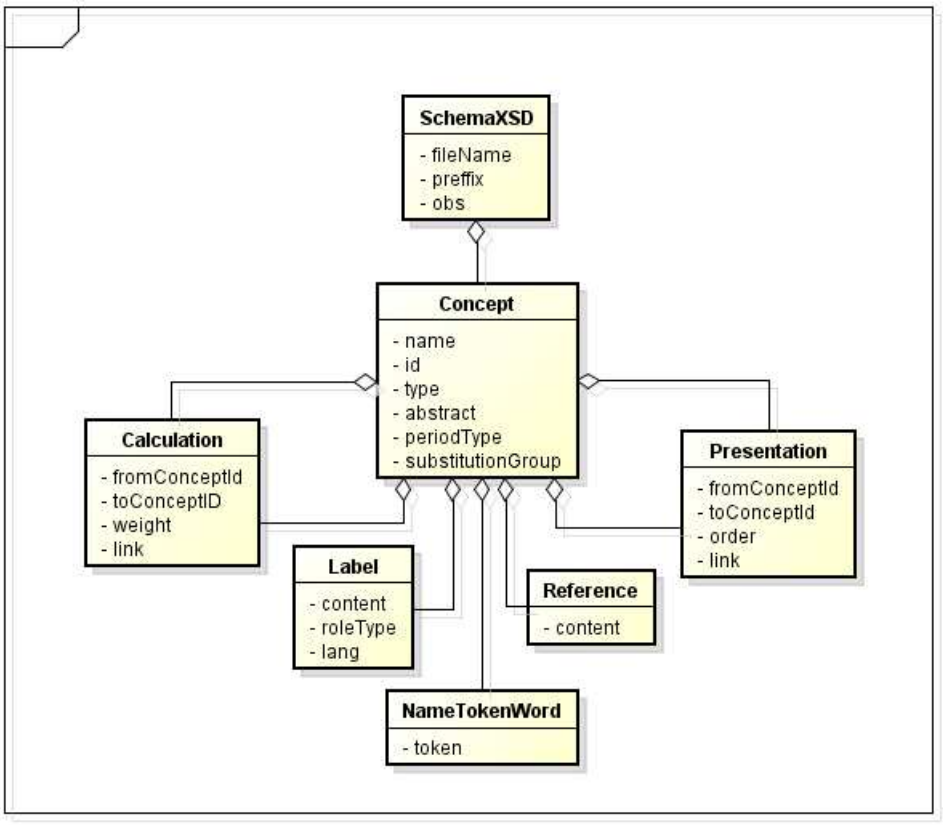
Além das medidas combinadas, foram incluídas no resultado final as medidas individuais de conteúdo e estrutura a fim de permitir a análise dos resultados utilizando cada uma como base para classificação da relevância da similaridade com o objetivo de se obter interpretações diferentes da informação.

4.2 ESTRUTURA DE DADOS PARA AVALIAÇÃO DE SIMILARIDADE ENTRE CONCEITOS REPRESENTADOS PELA XBRL

Uma estrutura de dados foi desenhada com o objetivo de dar suporte ao processo de avaliação de similaridade. Sendo assim, todas as informações consideradas relevantes ao processo, e discutidas neste capítulo, estão representadas nas tabelas utilizadas. Entretanto, informações pertinentes à XBRL que não foram utilizadas no processo de avaliação não foram representadas.

A Figura 29 representa o modelo da estrutura de dados.

Figura 29– Modelo de dados



A tabela **SchemaXSD** comporta as taxonomias carregadas na base. Há atributos para descrever a taxonomia, o prefixo (*namespace*) utilizado por ela, o nome do seu *entry point* e um campo identificador gerado automaticamente pelo SGBD.

A tabela **Concept** armazena os conceitos das taxonomias. Há atributos para armazenar as propriedades *name, id, type, abstract, periodType e substitutionGroup* do conceito. Cada taxonomia em **SchemaXSD** possui múltiplos conceitos em **Concept**, a ligação é feita através de uma chave estrangeira.

As tabelas **Label** e **NameTokenWord** estão vinculadas à tabela **Concept** como atributos multivalorados. Em **Label** são armazenados os *links label* do conceito. Em **NameTokenWord** são armazenados *tokens* para o atributo *name* do conceito obtidos através da utilização de letras maiúsculas como delimitadoras.

As tabelas **Calculation** e **Presentation** representam os relacionamentos entre os conceitos. São auto-relacionamentos da tabela **Concept**, indicando qual o conceito pai e qual o conceito filho da relação. Ambas possuem o atributo *linkbase*, para indicar em que rede este relacionamento está. Em **Calculation** há o atributo *weight* e em **Presentation** o atributo *order*.

4.3 PROCESSO DE AVALIAÇÃO DE SIMILARIDADE ENTRE CONCEITOS REPRESENTADOS PELA XBRL

A avaliação de similaridade proposta neste trabalho é feita através de processos que são executados em um SGBD relacional: carga dos dados e avaliação de similaridade. As seções a seguir descrevem esses processos e suas atividades.

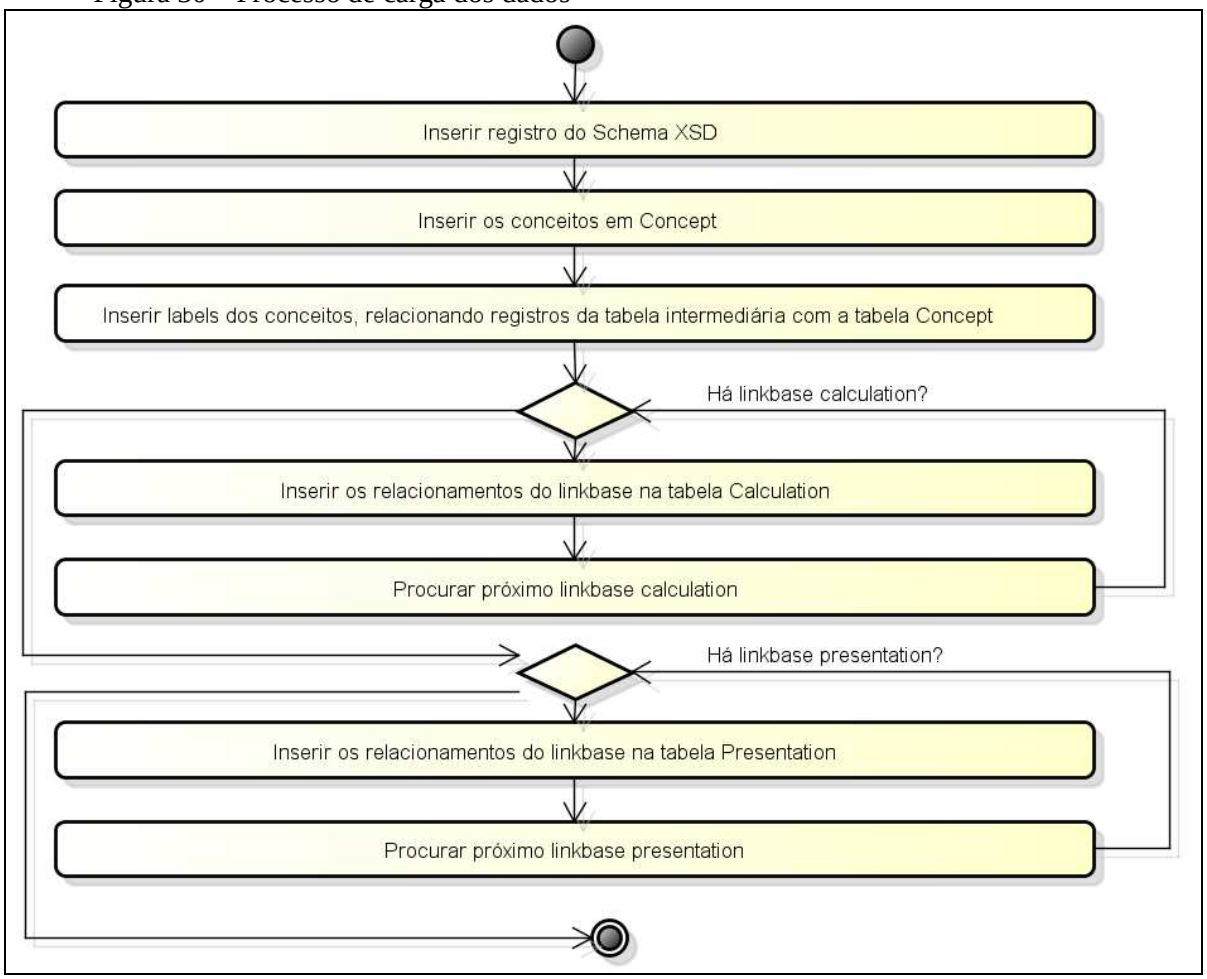
4.3.1 Processo de carga dos dados XBRL

O processo de carga recebe como parâmetro os atributos do *Schema* XSD sendo importado. O primeiro passo é inserir o registro do *Schema* XSD na tabela pertinente (**SchemaXSD**). A seguir, apenas os conceitos do *Schema XSD* e seus atributos são inseridos na tabela **Concept**, apontando para o registro incluído em SchemaXSD. O conteúdo do *linkbase* do tipo *reference* é inserido na respectiva tabela (**Label**), apontando para o respectivo conceito.

O processo de carga extrai dados de tabelas intermediárias e insere os dados nas tabelas que são usadas para a avaliação no processo proposto. As tabelas intermediárias são necessárias para que o processo de carga monte corretamente as referências (chaves estrangeiras) do modelo de dados proposto para representar a taxonomia XBRL no SGBD relacional.

A Figura 30 ilustra o processo de carga dos dados.

Figura 30 – Processo de carga dos dados



Para inclusão dos relacionamentos do tipo *Calculation* e *Presentation* são identificados, em primeiro lugar, os *linkbases* correspondentes. Os *linkbases* indicam redes de relacionamento distintas. A seguir são explorados os arcos dos relacionamentos de cada *linkbase*, e o registro para o relacionamento pai-filho é inserido na respectiva tabela (**Calculation** e **Presentation**).

Por fim, o processo carrega a tabela com os *tokens* do nome de cada conceito. Os *token* são palavras, obtidas utilizando letras maiúsculas como delimitador. A *tokenização* é feita na

carga, a fim de eliminar este passo do processo de avaliação de similaridade, melhorando o desempenho deste último.

4.3.2 Processo de avaliação de similaridade entre conceitos representados pela XBRL

A Figura 31 mostra o processo de avaliação da similaridade. Inicialmente, neste processo, são recebidas, como parâmetro, a chave de dois *Schemas XSD* (os quais podem ser o mesmo) e um coeficiente de similaridade β. Após o processamento, é retornada para o usuário uma tabela com os resultados da avaliação de similaridade entre os conceitos dos dois *Schemas XSDs* recebidos cuja similaridade entre os seus nomes seja maior que o coeficiente β. A tabela de retorno contém os pares de conceitos e as classificações de similaridade calculadas para o par.

Em primeiro lugar, o processo identifica o conjunto de pares de conceitos formados pelo produto cartesiano entre todos os conceitos do schemaA e todos os conceitos do schemaB. Caso a avaliação de similaridade esteja sendo feita sobre um mesmo *Schema XSD*, ou seja, schemaA = schemaB, os pares de conceitos duplicados são descartados do conjunto.

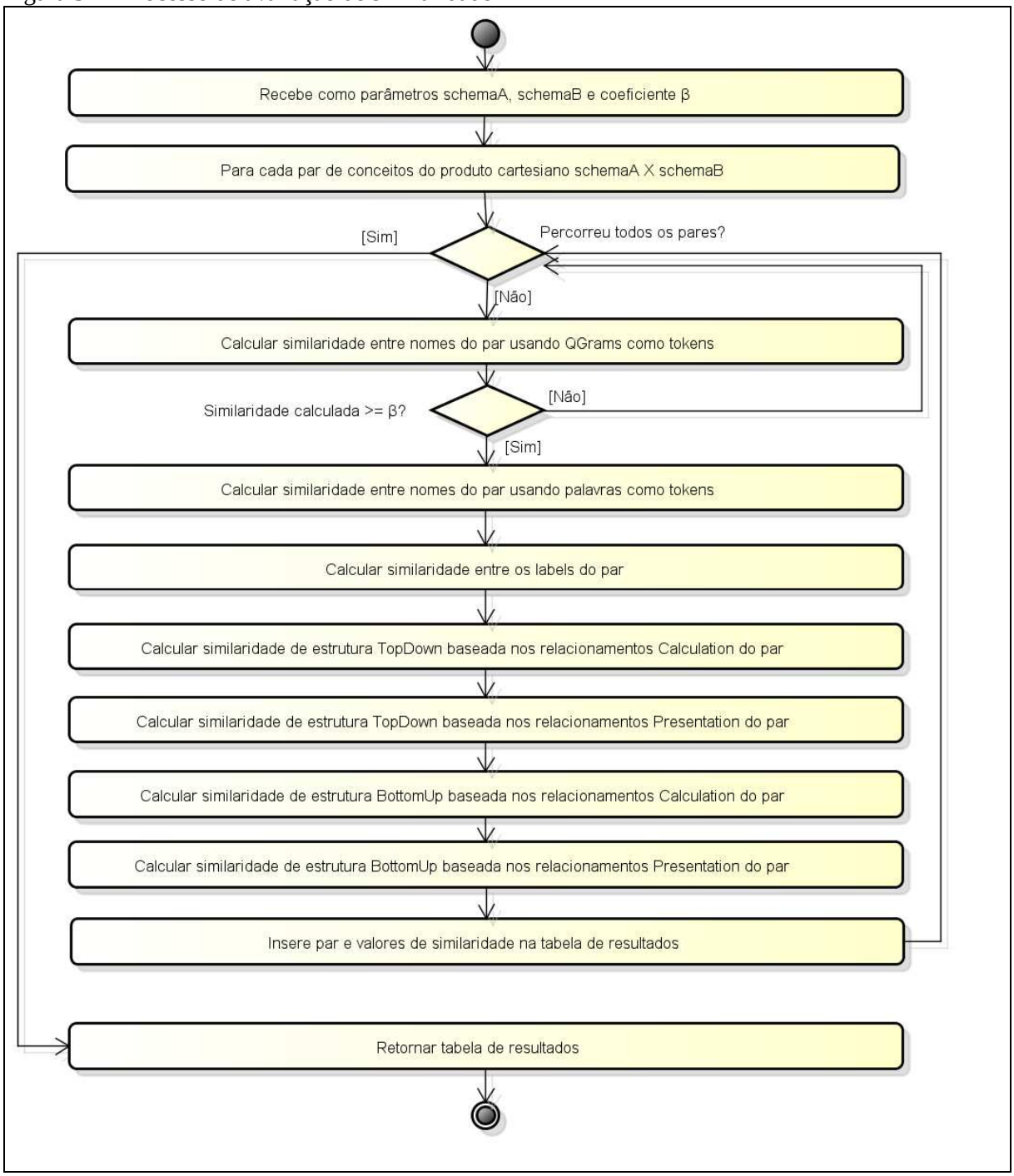
A seguir, um conjunto de atividades é realizado repetidamente para cada par de conceitos do conjunto identificado no primeiro passo. Cada atividade está descrita a seguir:

- Calcular Similaridade de Nomes com QGrams: esta é a primeira atividade realizada no par de conceitos sendo avaliado. Ela calcula a similaridade entre os nomes de cada conceito do par, utilizando a técnica de QGrams. Caso o valor calculado seja inferior ao coeficiente β, recebido como parâmetro, as demais atividades não são executadas para o par e o próximo par é avaliado.

- Calcular Similaridade de Nomes com Palavras: avalia a similaridade entre os nomes dos conceitos usando palavras como *tokens*.

- Calcular Similaridade de *label*: compara todos os labels dos conceitos do par entre si e computa o grau de similaridade entre eles através da função QGrams. A similaridade total é calculada a partir da média aritmética entre eles.

Figura 31 – Processo de avaliação de similaridade



- Calcular Similaridade *topdown*: calcula o grau de similaridade de estrutura dos relacionamentos *calculation* e *presentation* dos dois conceitos, considerando a abordagem *TopDown*. Navega em todos os conceitos descendentes dos conceitos recebidos como parâmetros, armazenando o caminho (*path*) de cada um. A seguir, todos os caminhos obtidos de cada conceito são comparados entre si, utilizando a rotina de classificação de similaridade

entre caminhos. A similaridade final é computada a partir da média aritmética das similaridades individuais entre os caminhos.

- Calcular Similaridade *bottomup*: análoga à atividade de Calcular Similaridade *topdown*, calcula o grau de similaridade de estrutura dos relacionamentos *calculation* e *presentation* dos dois conceitos, neste caso considerando a abordagem *BottomUp*. Navega em todo os conceitos pais dos conceitos recebidos como parâmetros, armazenando o caminho (*path*) de cada um. A seguir, o cálculo de similaridade é feito da mesma forma da rotina *Top Down*, em que a similaridade dos caminhos é calculada e combinada para o resultado final.

- Inserir Par na Tabela de Resultados: insere, na tabela de resultados do processo, o par de conceitos sendo avaliado (cuja Similaridade de Nomes com QGrams seja maior que o coeficiente β passado como parâmetro) e os valores de similaridade calculados: nomes com *QGrams*, nomes com palavras, l*abel, topdown calculation, topdown presentation, bottomupcalculation* e *bottomup presentation*.

4.4 TRABALHOS CORRELATOS

No contexto da linguagem XBRL, pode-se encontrar estudos que analisam a questão da Gestão da Qualidade de Dados (BARTLEY et. al, 2011); (BORITZ; NO, 2008); (ZHU ; WU, 2011).

Em Zhu e Wu (2011), é feito um estudo voltado à análise da qualidade dos padrões propostos pelas agências reguladoras, em especial o US-GAAP. Um dos pontos avaliados é se o padrão é completo o suficiente para só exigir a criação de novos conceitos no caso de particularidades pontuais. A fim de fazer essa análise, o estudo de (ZHU; WU, 2011) avaliou a similaridade entre conceitos da US-GAAP e contra um conjunto de taxonomias, comparando apenas seus nomes, utilizando o método do espaço vetorial. O estudo cita como trabalho futuro agregar mais informações para identificação de conceitos duplicados na avaliação de similaridade.

4.5 CONSIDERAÇÕES

Este capítulo apresentou o processo proposto para avaliação de similaridade entre conceitos representados pela XBRL, suas atividades e o modelo de dados para abrigar as informações. Foram identificadas as informações relevantes para o processo e as medidas de similaridade mais adequadas para cada informação.

O processo proposto apresentado neste capítulo busca agregar um conjunto de técnicas para a avaliação de similaridade dentro do domínio da linguagem XBRL, agregando informações sobre conteúdo e estrutura e incluindo informações complementares dos *linkbases*. A reunião de diversas técnicas para avaliação de similaridade possibilita mais diversidade e completude na análise dos resultados. O capítulo seguinte irá discutir a aplicação desse processo em dois estudos de caso.

5 ESTUDO DE CASO: AVALIAÇÃO DE UMA TAXONOMIA EM DESENVOLVIMENTO E AVALIAÇÃO DE TAXONOMIAS ESTENDIDAS DA US-SEC

Este capítulo apresenta o resultado de dois experimentos feitos com o processo de avaliação de similaridade entre conceitos representados pela XBRL. O primeiro experimento teve como objetivo avaliar o uso do processo proposto em uma taxonomia em construção, com o objetivo de fornecer ao especialista de negócio indicações de possíveis conceitos duplicados no desenvolvimento da taxonomia.

O segundo experimento realizou-se sobre 11 taxonomias estendidas da US-SEC (*Securities and Exchange Commission*)[6]. A US-SEC permite que as companhias que informam seus dados financeiros estendam a taxonomia original, fornecida pela US-SEC, para incorporar conceitos que são considerados relevantes pelas companhias e necessários de serem informados. Com isso, existe a possibilidade de que conceitos similares estejam presentes em taxonomias estendidas. A utilização do processo proposto teve como objetivo identificar a similaridade entre estes conceitos.

5.1 INFRAESTRUTURA PARA AVALIAÇÃO DE SIMILARIDADE ENTRE CONCEITOS REPRESENTADOS PELA XBRL

A fim de realizar os experimentos, foi montada uma base de testes em um SGBD (Sistema Gerenciador de Banco de Dados) relacional. Os dados das taxonomias foram representados em uma estrutura de dados (tabelas) dentro do SGBD. O modelo relacional foi utilizado para representar os dados por servir ao propósito da avaliação de similaridade e ser um modelo robusto e bem aceito, tanto na academia, quanto na indústria. O SGBD utilizado foi o Microsoft SQL Server 2012 Express, versão 11.0.2100.60[7].

Inicialmente, o processo de carga na base de dados foi feito a partir da exportação do conteúdo das taxonomias por meio de uma ferramenta de visualização e manipulação da

[6] http://www.sec.gov

[7] http://www.microsoft.com/pt-br/download/details.aspx?id=29062

XBRL, a Arelle[8], versão 1.0.0. Os dados das taxonomias foram exportados para documentos do tipo texto. A seguir foi utilizada uma ferramenta do SGBD (*Import and Export Data*) para carregar os dados em tabelas intermediárias. A partir das tabelas intermediárias, por meio de um código implementado no SGBD, são criadas as tabelas na estrutura de dados necessárias para a avaliação de similaridade. A estrutura de dados necessária para a avaliação da similaridade foi a descrita no capítulo anterior.

5.2 SIMILARIDADE ENTRE CONCEITOS DE UMA TAXONOMIA EM CONSTRUÇÃO

Este experimento foi feito sobre uma taxonomia em construção. Por estar em construção, ainda não estavam disponíveis informações sobre relacionamentos do tipo *calculation*, nem *presentation*. Sendo assim, o processo de avaliação de similaridade foi executado somente sobre as informações de conteúdo, sem incluir a análise sobre a estrutura da taxonomia. O objetivo dessa avaliação foi encontrar elementos duplicados na taxonomia em construção, de forma a melhorar a qualidade da taxonomia em construção, evitando a utilização de conceitos repetidos na construção dos *linkbases*.

A taxonomia possui um total de 6.503 conceitos. O processo de avaliação de similaridade avaliou 21.141.253 pares de conceitos ((6503*6502)/2). Destes pares, 241.277 obtiveram ‘similaridade de nome’ maior do que 0,5, ou seja, pouco mais de 1% do total.

O Quadro 2 apresenta os primeiros pares de conceitos avaliados no processo de avaliação de similaridade sobre a taxonomia em construção em uma tabela de resultados. As colunas *Nome* e *Label* apresentam as medidas de similaridade individuais dos respectivos atributos dos conceitos. A coluna *Med* apresenta a média aritmética dessas duas medidas. Os resultados foram classificados segundo a coluna *Med*, em ordem decrescente.

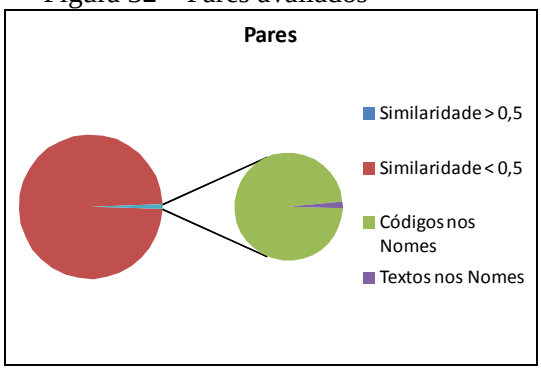
[8] http://www.arelle.org

Quadro 2 - Avaliação de similaridade de taxonomia em construção

SIMILARIDADES			NOMES DOS CONCEITOS	
NOME	LABEL	MED	A	B
1,00	1,00	1,00	P4.1.1.3.1.97.00	P4.1.3.1.1.97.00
0,97	0,97	0,97	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIE	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoII
0,97	0,97	0,97	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIE	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIR
0,97	0,97	0,97	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoII	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIR
0,92	1,00	0,96	ImpostosTaxasEContribuicoesDeMelhoria	ImpostosTaxasEContribuicoesDeMelhoriaVPD
0,92	1,00	0,96	DespesaComSaudeCusteadaComRecursosReferentesAASPSNaoAplicadaNoExercicioAnterior	DespesaComSaudeCusteadaComRecursosReferentesAASPSNaoAplicadaNoExercicioAnteriorConsorcio
0,92	1,00	0,96	P8.1.1.2.1.01.10	P8.1.2.2.1.01.10
0,92	1,00	0,96	P8.1.1.2.1.01.11	P8.1.2.2.1.01.11
0,92	1,00	0,96	P8.1.1.2.1.01.12	P8.1.2.2.1.01.12
0,91	1,00	0,96	P3.2.9.0.0.00.00	P3.2.9.9.0.00.00
0,98	0,93	0,96	DespesasComASPSCusteadasComDisponibilidadeVinculadaARestosAPagarCanceladosOuPrescritosEmExercicioDeReferenciaMenos1	DespesasComASPSCusteadasComDisponibilidadeVinculadaARestosAPagarCanceladosOuPrescritosEmExercicioDeReferenciaMenos2
0,91	1,00	0,96	EmprestimosEFinanciamentosConcedidosAtivoCirculante	EmprestimosEFinanciamentosConcedidosAtivoNaoCirculante
0,95	0,96	0,96	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoII	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIOF
0,95	0,96	0,96	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoII	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIPI
0,95	0,96	0,96	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIE	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIPI
0,95	0,96	0,96	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIE	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIOF
0,95	0,96	0,96	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIOF	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIOFOuro
0,95	0,96	0,96	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIOF	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIR

Do total de 6.503 conceitos, 4.729 (74%) eram conceitos cujo nome era composto por uma conta contábil como, por exemplo, “P8.9.2.9.0.00.00” e “P2.2.1.4.3.00.00”. O uso de código na composição do nome do conceito causa distorção na sua avaliação de similaridade, visto que a mudança de apenas um dígito modifica totalmente o código, mas não acarreta grande diferença em sua similaridade. Isto pode ser observado nos nomes dos conceitos do primeiro registro do Quadro 2: “P4.1.1.3.1.97.00” e “P4.1.3.1.1.97.00”. A maioria dos pares encontrados com similaridade maior do que 0,5 (98%) eram compostos por conceitos com códigos como nomes. A Figura 33 ilustra esse resultado. Para estes conceitos, a melhor análise seria desconsiderando a avaliação entre os nomes e considerar apenas a similaridade de *label*.

Figura 32 – Pares avaliados



Ao levar em consideração apenas os conceitos cujo nome é composto por um texto descritivo (1.774), o processo de avaliação de similaridade fez a avaliação de 1.527.651 pares desses conceitos ((1.774*1.773)/2). Desses pares, 4.562 obtiveram similaridade de nome maior do que 0,5, ou seja, menos de 1% do subtotal de pares de conceitos cujo nome não é um código. O Quadro 3 apresenta os primeiros pares da tabela de resultados do processo de avaliação de similaridade sobre a taxonomia em construção, considerando apenas conceitos cujo nome é composto por um texto descritivo e não códigos.

Quadro 3 – Taxonomia em construção: nomes sem códigos

SIMILARIDADES			NOMES DOS CONCEITOS	
NOME	LABEL	MED	A	B
0,97	0,97	0,97	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIE	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoII
0,97	0,97	0,97	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIE	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIR
0,97	0,97	0,97	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoII	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIR
0,92	1,00	0,96	ImpostosTaxasEContribuicoesDeMelhoria	ImpostosTaxasEContribuicoesDeMelhoriaVPD
0,92	1,00	0,96	DespesaComSaudeCusteadaComRecursosReferentesAASPSNaoAplicadaNoExercicioAnterior	DespesaComSaudeCusteadaComRecursosReferentesAASPSNaoAplicadaNoExercicioAnteriorConsorcio
0,98	0,93	0,96	DespesasComASPSCusteadasComDisponibilidadeVinculadaARestosAPagarCanceladosOuPrescritosEmExercicioDeReferenciaMenos1	DespesasComASPSCusteadasComDisponibilidadeVinculadaARestosAPagarCanceladosOuPrescritosEmExercicioDeReferenciaMenos2
0,91	1,00	0,96	EmprestimosEFinanciamentosConcedidosAtivoCirculante	EmprestimosEFinanciamentosConcedidosAtivoNaoCirculante
0,95	0,96	0,96	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoII	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIOF
0,95	0,96	0,96	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoII	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIPI
0,95	0,96	0,96	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIE	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIPI
0,95	0,96	0,96	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIE	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIOF
0,95	0,96	0,96	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIOF	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIOFOuro
0,95	0,96	0,96	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIOF	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIR
0,95	0,96	0,96	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIPI	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaDoIR
0,95	0,96	0,96	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaITBI	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaITR
0,95	0,96	0,96	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaITCD	MultaJurosAtualizacaoMonetariaEOutrosEncargosDaDividaAtivaITR
0,97	0,93	0,95	DespesasComASPSCusteadasComRecursosDeParcelaNaoAplicadaEmExercicioDeReferenciaMenos1	DespesasComASPSCusteadasComRecursosDeParcelaNaoAplicadaEmExercicioDeReferenciaMenos2
0,97	0,93	0,95	DespesasComASPSCusteadasComRecursosDeParcelaNaoAplicadaEmExercicioDeReferenciaMenos1	DespesasComASPSCusteadasComRecursosDeParcelaNaoAplicadaEmExercicioDeReferenciaMenos3

Com o objetivo de analisar os pares de conceito cujo nome é um código, eliminando a distorção da avaliação de similaridade entre seus nomes, foi construída uma análise baseada nos *labels*, apresentada no Quadro 4. A análise contém os pares de conceito cujo nome é um código, na ordem decrescente da similaridade entre seus *links label*. Nesta análise, foi acrescentada uma coluna com os *labels* dos conceitos, para ajudar na avaliação.

Quadro 4 - Taxonomia em construção: nomes com códigos

SIMILARIDADES			CONCEITO A		CONCEITO B	
NOME	LABEL	MED	NOME	LABEL	NOME	LABEL
0.70	1.00	0.850	P1.1.1.1.1.01.00	Caixa	P1.1.1.2.1.01.00	Caixa
0.62	1.00	0.810	P1.1.2.1.0.00.00	CrÃ©ditos TributÃ¡rios a Receber	P1.2.1.1.1.01.00	CrÃ©ditos TributÃ¡rios a Receber
0.57	1.00	0.785	P1.1.2.1.0.00.00	CrÃ©ditos TributÃ¡rios a Receber	P1.2.1.1.2.01.00	CrÃ©ditos TributÃ¡rios a Receber
0.50	1.00	0.750	P1.1.2.1.0.00.00	CrÃ©ditos TributÃ¡rios a Receber	P1.2.1.1.3.01.00	CrÃ©ditos TributÃ¡rios a Receber
0.50	1.00	0.750	P1.1.2.1.0.00.00	CrÃ©ditos TributÃ¡rios a Receber	P1.2.1.1.5.01.00	CrÃ©ditos TributÃ¡rios a Receber
0.50	1.00	0.750	P1.1.2.1.0.00.00	CrÃ©ditos TributÃ¡rios a Receber	P1.2.1.1.4.01.00	CrÃ©ditos TributÃ¡rios a Receber
0.91	1.00	0.955	P1.1.2.1.1.01.00	Impostos	P1.1.2.1.2.01.00	Impostos
0.77	1.00	0.885	P1.1.2.1.1.01.00	Impostos	P1.1.2.1.3.01.00	Impostos
0.77	1.00	0.885	P1.1.2.1.1.01.00	Impostos	P1.1.2.1.4.01.00	Impostos
0.77	1.00	0.885	P1.1.2.1.1.01.00	Impostos	P1.1.2.1.5.01.00	Impostos
0.90	1.00	0.950	P1.1.2.1.1.01.00	Impostos	P1.2.1.1.1.01.01	Impostos
0.90	1.00	0.950	P1.1.2.1.1.01.01	Imposto sobre a Renda e Proventos de Qualquer Natureza	P1.1.2.1.2.01.01	Imposto sobre a Renda e Proventos de Qualquer Natureza
0.75	1.00	0.875	P1.1.2.1.1.01.01	Imposto sobre a Renda e Proventos de Qualquer Natureza	P1.1.2.1.3.01.01	Imposto sobre a Renda e Proventos de Qualquer Natureza
0.75	1.00	0.875	P1.1.2.1.1.01.01	Imposto sobre a Renda e Proventos de Qualquer Natureza	P1.1.2.1.4.01.01	Imposto sobre a Renda e Proventos de Qualquer Natureza
0.75	1.00	0.875	P1.1.2.1.1.01.01	Imposto sobre a Renda e Proventos de Qualquer Natureza	P1.1.2.1.5.01.01	Imposto sobre a Renda e Proventos de Qualquer Natureza
0.91	1.00	0.955	P1.1.2.1.1.01.02	IPVA	P1.1.2.1.2.01.02	IPVA
0.77	1.00	0.885	P1.1.2.1.1.01.02	IPVA	P1.1.2.1.3.01.02	IPVA
0.77	1.00	0.885	P1.1.2.1.1.01.02	IPVA	P1.1.2.1.5.01.02	IPVA
0.77	1.00	0.885	P1.1.2.1.1.01.02	IPVA	P1.1.2.1.4.01.02	IPVA
0.91	1.00	0.955	P1.1.2.1.1.01.03	ITCMD	P1.1.2.1.2.01.03	ITCMD
0.77	1.00	0.885	P1.1.2.1.1.01.03	ITCMD	P1.1.2.1.3.01.03	ITCMD
0.77	1.00	0.885	P1.1.2.1.1.01.03	ITCMD	P1.1.2.1.5.01.03	ITCMD
0.77	1.00	0.885	P1.1.2.1.1.01.03	ITCMD	P1.1.2.1.4.01.03	ITCMD
0.91	1.00	0.955	P1.1.2.1.1.01.04	ICMS	P1.1.2.1.2.01.04	ICMS
0.77	1.00	0.885	P1.1.2.1.1.01.04	ICMS	P1.1.2.1.3.01.04	ICMS
0.77	1.00	0.885	P1.1.2.1.1.01.04	ICMS	P1.1.2.1.4.01.04	ICMS
0.77	1.00	0.885	P1.1.2.1.1.01.04	ICMS	P1.1.2.1.5.01.04	ICMS
0.91	1.00	0.955	P1.1.2.1.1.01.05	IPTU	P1.1.2.1.2.01.05	IPTU

Pode-se observar, na análise do Quadro 4, que os *labels* repetidos ocorrem dentro de um mesmo grupo de códigos contábeis, indicando que a repetição pode ter sido proposital. Baseado nesta observação, foi construída a análise do Quadro 5, em que são ilustrados os pares de conceitos com códigos sem relação de hierarquia, ou seja, cujos primeiros dígitos do códigos sejam distintos.

Quadro 5 - Taxonomia em construção: códigos sem hierarquia

SIMILARIDADES			CONCEITO A		CONCEITO B	
NOME	LABEL	MED	NOME	LABEL	NOME	LABEL
0.53	1.00	0.765	P1.1.2.1.5.01.04	ICMS	P3.5.2.1.5.01.00	ICMS
0.53	1.00	0.765	P1.1.2.1.1.02.02	Taxas pela PrestaÃ§Ã£o de ServiÃ§os	P3.7.1.2.1.02.00	Taxas pela PrestaÃ§Ã£o de ServiÃ§os
0.50	1.00	0.750	P3.7.1.0.0.00.00	Impostos, Taxas e ContribuiÃ§Ãµes de Melhoria	P4.1.0.0.0.00.00	Impostos, Taxas e ContribuiÃ§Ãµes de Melhoria
0.54	1.00	0.770	P3.7.1.1.0.00.00	Impostos	P4.1.1.0.0.00.00	Impostos
0.57	1.00	0.785	P3.7.1.2.0.00.00	Taxas	P4.1.2.0.0.00.00	Taxas
0.57	1.00	0.785	P3.7.1.3.0.00.00	ContribuiÃ§Ãµes de Melhoria	P4.1.3.0.0.00.00	ContribuiÃ§Ãµes de Melhoria
0.50	1.00	0.750	P3.7.2.0.0.00.00	ContribuiÃ§Ãµes	P4.2.0.0.0.00.00	ContribuiÃ§Ãµes
0.57	1.00	0.785	P3.7.2.1.0.00.00	ContribuiÃ§Ãµes Sociais	P4.2.1.0.0.00.00	ContribuiÃ§Ãµes Sociais
0.54	1.00	0.770	P3.7.2.2.0.00.00	ContribuiÃ§Ãµes de IntervenÃ§Ã£o no DomÃ-nio EconÃ´mico	P4.2.2.0.0.00.00	ContribuiÃ§Ãµes de IntervenÃ§Ã£o no DomÃ-nio EconÃ´mico
0.67	1.00	0.835	P3.4.2.0.0.00.00	Juros e Encargos de Mora	P4.4.2.0.0.00.00	Juros e Encargos de Mora
0.71	1.00	0.855	P3.4.2.9.0.00.00	Outros Juros e Encargos de Mora	P4.4.2.9.0.00.00	Outros Juros e Encargos de Mora
0.73	1.00	0.865	P3.4.2.9.1.00.00	Outros Juros e Encargos de Mora - ConsolidaÃ§Ã£o	P4.4.2.9.1.00.00	Outros Juros e Encargos de Mora - ConsolidaÃ§Ã£o
0.67	1.00	0.835	P3.4.3.0.0.00.00	VariaÃ§Ãµes MonetÃ¡rias e Cambiais	P4.4.3.0.0.00.00	VariaÃ§Ãµes MonetÃ¡rias e Cambiais
0.71	1.00	0.855	P3.4.3.9.0.00.00	Outras VariaÃ§Ãµes MonetÃ¡rias e Cambiais	P4.4.3.9.0.00.00	Outras VariaÃ§Ãµes MonetÃ¡rias e Cambiais
0.73	1.00	0.865	P3.4.3.9.1.00.00	Outras VariaÃ§Ãµes MonetÃ¡rias e Cambiais - ConsolidaÃ§Ã£o	P4.4.3.9.1.00.00	Outras VariaÃ§Ãµes MonetÃ¡rias e Cambiais - ConsolidaÃ§Ã£o
0.71	1.00	0.855	P3.4.3.9.3.00.00	Outras VariaÃ§Ãµes MonetÃ¡rias e Cambiais - Inter OFSS - UniÃ£o	P4.4.3.9.3.00.00	Outras VariaÃ§Ãµes MonetÃ¡rias e Cambiais - Inter OFSS - UniÃ£o
0.71	1.00	0.855	P3.4.3.9.4.00.00	Outras VariaÃ§Ãµes MonetÃ¡rias e Cambiais - Inter OFSS - Estado	P4.4.3.9.4.00.00	Outras VariaÃ§Ãµes MonetÃ¡rias e Cambiais - Inter OFSS - Estado
0.73	1.00	0.865	P3.4.3.9.5.00.00	Outras VariaÃ§Ãµes MonetÃ¡rias e Cambiais - Inter OFSS - MunicÃ-pio	P4.4.3.9.5.00.00	Outras VariaÃ§Ãµes MonetÃ¡rias e Cambiais - Inter OFSS - MunicÃ-pio
0.67	1.00	0.835	P3.5.1.0.0.00.00	TransferÃªncias Intragovernamentais	P4.5.1.0.0.00.00	TransferÃªncias Intragovernamentais
0.73	1.00	0.865	P3.5.1.1.2.05.00	Valores Diferidos - Baixa	P4.5.1.1.2.05.00	Valores Diferidos - Baixa
0.73	1.00	0.865	P3.5.1.1.2.08.00	Valores Diferidos - InscriÃ§Ã£o	P4.5.1.1.2.08.00	Valores Diferidos - InscriÃ§Ã£o
0.75	1.00	0.875	P3.5.1.3.2.01.00	Plano Financeiro	P4.5.1.3.2.01.00	Plano Financeiro
0.73	1.00	0.865	P3.5.1.3.2.01.01	Recursos para Cobertura de InsuficiÃªncias Financeiras	P4.5.1.3.2.01.01	Recursos para Cobertura de InsuficiÃªncias Financeiras

Embora o processo apresente informações consistentes sobre a similaridade entre os conceitos, a decisão final sobre sua duplicidade deve ser tomada por especialistas do negócio, uma vez que conceitos distintos podem ter, por exemplo, o mesmo *label*, usados em situações

de reporte diferentes. Um exemplo encontrado na taxonomia em avaliação é o dos conceitos “ImpostosTaxasEContribuicoesDeMelhoria” e “ImpostosTaxasEContribuicoesDeMelhoria VPD” que possuem o mesmo *label* “Impostos, Taxas e Contribuições de Melhoria”. O especialista pode chegar à conclusão de que há duplicidade de conceitos, que o *label* deve ser mais bem definido, ou que são aplicações de conceitos distintos, com o mesmo *label*, em relatórios financeiros diferentes.

A aplicação do processo de avaliação de similaridade proposto neste trabalho à taxonomia em construção trouxe informações que devem facilitar a análise dos especialistas para detecção de conceitos duplicados, ou muito semelhantes, que possam ser unidos ou diferenciados de outra forma.

5.3 EXPERIMENTO DE AVALIAÇÃO DE TAXONOMIAS DA SEC – MESMO SEGMENTO

Neste experimento, foram selecionadas 11 taxonomias de empresas do mesmo segmento de mercado aplicadas a relatórios entregues à Comissão de Títulos e Câmbios dos Estados Unidos – *Securities and Exchange Commission* (SEC–USA). O objetivo desta avaliação foi identificar semelhanças entre novos conceitos criados por estas taxonomias que estendem a taxonomia padrão da SEC-USA, a US-GAAP. A seleção de empresas de um mesmo segmento de mercado teve por objetivo restringir as taxonomias a um contexto que facilitasse a identificação de recorrência na criação de novos conceitos, visto que são empresas cuja operação é semelhante e provavelmente têm a mesma necessidade de uso de conceitos para representação de seus fatos financeiros.

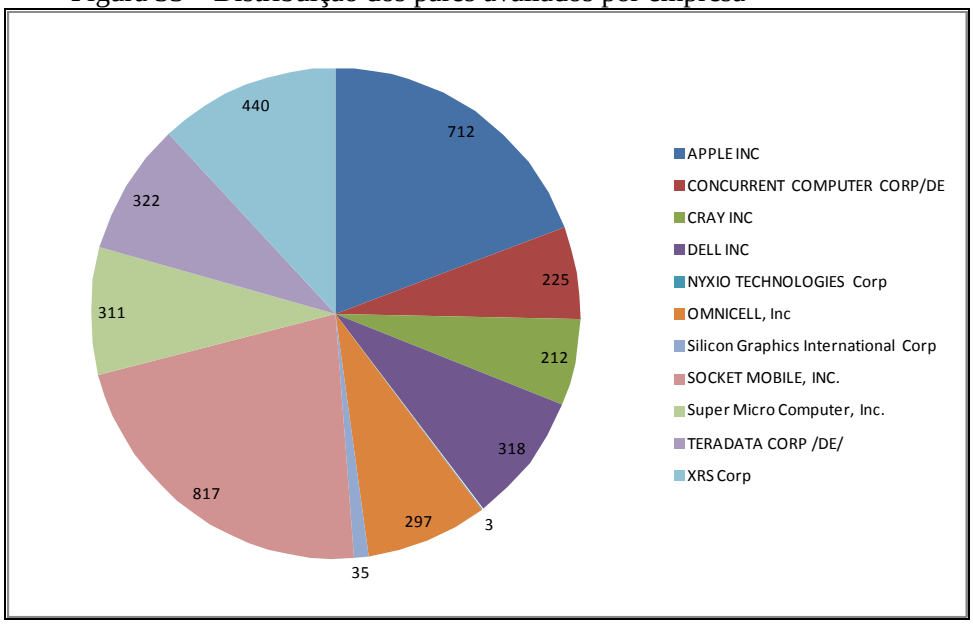
O segmento de mercado selecionado foi o de “Computadores Eletrônicos” (*Eletronic Computer*) cujo código de classificação padrão industrial – *Standard Industrial Classification* (SIC) é o 3571. Este segmento foi escolhido por ser ligado à área de tecnologia e pelo fato de possuir uma quantidade representativa de empresas que forneceram seus formulários anuais de reporte, categoria denominada 10-K, no padrão XBRL com extensão da taxonomia US-GAAP. Em 11 de março de 2013 foram descarregadas do site de relatórios públicos da SEC-USA as taxonomias de todas as empresas ativas do segmento SIC-3571 criadas para seus últimos relatórios anuais 10-k. A lista de empresas com as respectivas quantidades de novos conceitos de cada taxonomia é apresentada no Quadro 6.

Quadro 6 - Empresas avaliadas

CIK	EMPRESA	QTD. CONCEITOS	PREFIXO	DATA TAXONOMIA
320193	APPLE INC	76	aapl	29/09/2012
749038	CONCURRENT COMPUTER CORP/DE	95	ccur	30/06/2012
949158	CRAY INC	63	cray	31/12/2012
826083	DELL INC	73	dell	03/02/2012
1373761	NYXIO TECHNOLOGIES Corp	10	nyxo	31/12/2011
926326	OMNICELL, Inc	66	omcl	31/12/2012
1316625	Silicon Graphics International Corp	111	sgi	29/06/2012
944075	SOCKET MOBILE, INC.	24	sckt	31/12/2011
1375365	Super Micro Computer, Inc.	66	smci	30/06/2012
816761	TERADATA CORP /DE/	66	tdc	31/12/2012
854398	XRS Corp	89	xrsc	30/09/2012
	TOTAL DE CONCEITOS NOVOS	739		

As taxonomias de cada empresa foram comparadas com a taxonomia padrão US-GAAP para avaliar a similaridade entre seus conceitos. A taxonomia padrão US-GAAP possui um total de 12.050 conceitos, portanto, foram feitas 8.904.950 avaliações de pares de conceitos (12.050 * 739). Deste total, foram encontrados 3.692 pares com similaridade maior do que 0,5 (0,04%). A Figura 33 apresenta a distribuição dos pares similares por empresa.

Figura 33 – Distribuição dos pares avaliados por empresa



Pode-se observar, dentre esses pares, que alguns conceitos da taxonomia padrão US-GAAP tiveram mais resultados de similaridade ao compará-los com as taxonomias estendidas no processo de avaliação de similaridade, conforme Quadro 7.

Quadro 7 - Conceitos da US-GAAP com mais similares

CONCEITO	QTD. SIMILARES
ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsGrantsInPeriod	30
ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsVestedInPeriod	30
ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsGrantsInPeriod	30
ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsVestedInPeriodFairValue	29
ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsExercisableNumber	28
ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsExercisesInPeriod	28
ShareBasedCompensationArrangementByShareBasedPaymentAwardNonOptionEquityInstrumentsOther	28
ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsGrantsInPeriodIntrinsicValue	27
ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsVestedInPeriodWeightedAverageGrantDateFairValue	26
ShareBasedCompensationArrangementByShareBasedPaymentAwardAwardVestingPeriod	26
ShareBasedCompensationArrangementByShareBasedPaymentAwardNonOptionEquityInstrumentsGranted	26
ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsExpirationsInPeriod	26
ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsOutstandingNumber	25
ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsVestedInPeriodIntrinsicValue	25
ShareBasedCompensationArrangementByShareBasedPaymentAwardCompensationCost	25
ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsForfeitedInPeriod	25
ShareBasedCompensationArrangementByShareBasedPaymentAwardAwardVestingPeriod1	24
ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsNonvestedWeightedAverageGrantDateFairValue	24
ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsGrantsInPeriodWeightedAverageGrantDateFairValue	24

A análise feita sobre as informações ilustradas no Quadro 7 pode auxiliar a detectar verdadeiros duplicados neste conjunto de conceitos com mais similares apresentados no Quadro 8. Foram recuperados, por exemplo, todos os conceitos similares do conceito "ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstruments OtherThanOptionsGrantsInPeriod" (o primeiro da lista) ordenados pelos valores médios de

similaridade em ordem decrescente. Assim, o especialista pode avaliar se os conceitos são ou não duplicados, e recomendar que as empresas corrijam as suas taxonomias para utilizar o conceito padrão.

Quadro 8 - Conceitos das taxonomias estendidas similares ao da taxonomia padrão

TAXONOMIAS ESTENDIDAS		SIMILARIDADE
PREFIXO	CONCEITO	
xrst	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsSettledInPeriod	0,75
ccur	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsVestingPeriodMinimum	0,62
tdc	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsForfeitedInPeriodWeightedAverageGrantDateFairValues	0,59
smci	Sharebasedcompensationarrangementbysharebasedpaymentawardequityinstrumentsotherthanoptionsvestedinperiodnumber	0,59
xrst	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsSettledInPeriodWeightedAverageGrantDateFairValue	0,59
xrst	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsVestedAndUnsettledNumberOfShares	0,58
sgi	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsNonvestedOutstandingAggregateIntrinsicValue	0,58
xrst	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsExpectedToVestIntrinsicValue	0,58
ccur	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsVestingPeriodMaximum	0,57
sgi	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsVestedAndExpectedToVestOutstandingNumber	0,56
aapl	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsAssumedInPeriod	0,55
xrst	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsOutstandingNumber	0,55
omcl	ShareBasedCompensationArrangementByShareBasedPaymentAwardExpirationPeriod	0,55
sgi	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsCancellationsInPeriod	0,55
aapl	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsGrantsInPeriodUnroundedShares	0,55
sgi	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsNovestedWeightedAverageRemainingContractualTerm	0,52
sgi	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsVestedAndExpectedToVestOutstandingWeightedAverageGrantDateFairValue	0,52
xrst	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsVestedAndUnsettledIntrinsicValue	0,52

5.4 AVALIAÇÃO DOS RESULTADOS DOS EXPERIMENTOS

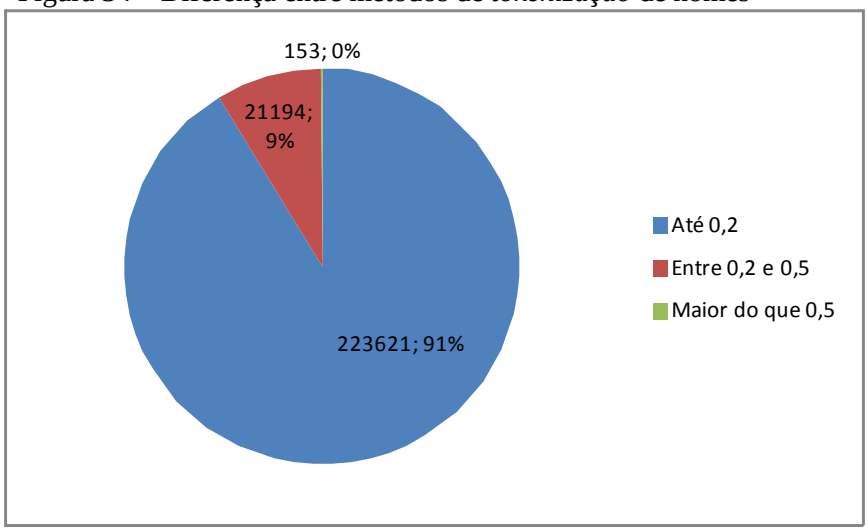
Na avaliação dos resultados dos experimentos, foram identificados alguns aspectos que aparecem em ambos os experimentos conduzidos. São questões relacionadas ao processo proposto e estão relacionadas a seguir:

5.4.1 Método para avaliação de similaridade do atributo *name* dos conceitos

No processo proposto, o atributo *name* pode ser considerado como o mais relevante. A similaridade de nomes entre os conceitos é o pilar da avaliação de similaridade de conteúdo (em conjunto com os valores dos *linkbases label* e *definition*) e de estrutura (por ser usado na avaliação de similaridade entre os caminhos hierárquicos).

O processo proposto utilizou dois métodos de *tokenização* dos nomes: *QGrams* e letra maiúscula como delimitador. De maneira geral, pôde-se observar que os dois métodos trouxeram resultados semelhantes, como ilustrado na Figura 34. Dos 244.968 pares avaliados como similares em ambos os estudos de caso, 91% apresentaram diferença nos valores de similaridade de nome entre os dois métodos menores do que 0,2.

Figura 34 – Diferença entre métodos de *tokenização* de nomes



Entretanto, no caso em que são utilizados códigos para os nomes dos conceitos, ambos os métodos utilizados no processo proposto falharam ao detectar sua similaridade. Ocorre que a mudança de apenas um dígito acarreta um valor de similaridade alto entre nomes que representam códigos totalmente diferentes, como ilustrado no Quadro 9. Para estes casos, o

processo deve considerar os *linkbases label* dos conceitos, por representarem informações mais relevantes e completas sobre os conceitos.

Quadro 9 - Similaridade de nomes com códigos

SIMILARIDADES		CONCEITOS	
QGrams	Palavra	A	B
1,00	1,00	P1.1.0.0.0.00.00	P1.1.1.0.0.00.00
1,00	1,00	P1.1.0.0.0.00.00	P1.1.1.1.0.00.00
1,00	1,00	P1.1.1.0.0.00.00	P1.1.1.1.0.00.00
1,00	1,00	P1.1.1.2.0.00.00	P1.1.2.0.0.00.00
1,00	1,00	P1.1.1.2.1.00.00	P1.1.2.1.1.00.00
1,00	1,00	P1.1.1.2.1.00.00	P1.2.1.1.1.00.00
1,00	1,00	P1.1.1.2.1.01.00	P1.1.2.1.1.01.00
1,00	1,00	P1.1.1.2.1.01.00	P1.2.1.1.1.01.00
1,00	1,00	P1.1.1.2.1.02.00	P1.1.2.1.1.02.00
1,00	1,00	P1.1.1.2.1.02.00	P1.2.1.1.1.02.00

Desde que os resultados de ambas as abordagens para *tokenização* do nome - *QGrams* e palavras – foram bem semelhantes, optou-se por considerar apenas o valor usando *QGrams* nas análises aqui apresentadas. Esta opção foi feita devido ao fato de não ser regra o uso em taxonomias XBRL de letras maiúsculas como delimitadores de palavras, sendo assim, não é garantido que este método de *tokenização* possa ser sempre utilizado.

5.4.2 Informações de conteúdo e estrutura

Na maioria dos 3.692 pares avaliados tanto para conteúdo quanto para estrutura (estudo de caso da SEC-USA), não houve diferenças significativas entre o valor da similaridade de conteúdo e o valor de similaridade de estrutura. Do total de 3.962 pares, 839 (22%) tinham diferença significativa entre os valores de similaridade de conteúdo e estrutura (diferenças maiores que 0,2). Em todos os pares desta categoria, foi possível observar que os valores de estrutura contribuíram para aumentar a média de similaridade, visto que o valor de similaridade de estrutura superou o valor de similaridade de conteúdo, conforme ilustrado no Quadro 10.

Quadro 10 - Diferença entre similaridades de conteúdo e estrutura

SIMILARIDADES			PREFIXO	CONCEITOS	
CONTEÚDO	ESTRUTURA	DIFERENÇA	TAXONOMIA	A	B
0,42	0,77	0,35	smci	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsVestedWeightedAverageExercisePrice	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsExercisableWeightedAverageExercisePrice
0,41	0,75	0,35	smci	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsVestedWeightedAverageExercisePrice	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsExercisesInPeriodWeightedAverageExercisePrice
0,42	0,76	0,34	smci	ShareBasedCompensationArrangementByShareBasedPaymentAwardNumberOfSharesAvailableBalance	ShareBasedCompensationArrangementByShareBasedPaymentAwardNumberOfSharesAvailableForGrant
0,42	0,76	0,34	smci	Sharebasedcompensationarrangementbyshareb asedpaymentawardequityinstrumentsotherthano ptionsvestedinperiodnumber	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsNonvestedNumber
0,41	0,74	0,34	smci	Sharebasedcompensationarrangementbyshareb asedpaymentawardequityinstrumentsotherthano ptionsvestedinperiodnumber	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsVestedInPeriodTotalFairValue
0,40	0,73	0,34	smci	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsVestedWeightedAverageExercisePrice	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsGrantsInPeriodWeightedAverageExercisePrice
0,56	0,89	0,33	ccur	DefinedBenefitPlanExpectedFutureBenefitPaymentsNextTwelveMonths	DefinedBenefitPlanExpectedFutureBenefitPaymentsNextTwelveMonths
0,39	0,72	0,33	smci	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsVestedWeightedAverageExercisePrice	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsExpirationsInPeriodWeightedAverageExercisePrice
0,40	0,73	0,33	smci	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsVestedWeightedAverageExercisePrice	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsVestedAndExpectedToVestExercisableWeightedAverageExercisePrice
0,40	0,72	0,33	smci	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsVestedWeightedAverageExercisePrice	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsOutstandingWeightedAverageExercisePrice
0,46	0,78	0,33	xrst	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsVestedAndUnsettledWeightedAverageGrantDateFairValue	ShareBasedCompensationArrangementByShareBasedPaymentAwardEquityInstrumentsOtherThanOptionsNonvestedWeightedAverageGrantDateFairValue
0,45	0,77	0,33	aapl	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsAssumedWeightedAverageExercisePrice	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsExercisableWeightedAverageExercisePrice
0,38	0,69	0,32	smci	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsVestedWeightedAverageExercisePrice	ShareBasedCompensationArrangementByShareBasedPaymentAwardOptionsForfeituresInPeriodWeightedAverageExercisePrice

Pode-se avaliar que este resultado era esperado, devido ao fato do resultado da similaridade de conteúdo ser utilizada na avaliação da similaridade de estrutura, por meio do uso do atributo *name* dos conceitos como base para a montagem e comparação entre os caminhos (*paths*) da estrutura.

Apesar de serem minoria os pares com muita diferença entre os valores de similaridade de conteúdo e de estrutura, pode-se observar a influência do valor de similaridade de estrutura no valor médio da similaridade, contribuindo para mais eficiência no processo de avaliação de similaridade, na medida em que refina os valores obtidos com o conteúdo considerando a estrutura.

5.4.3 Desempenho do processo de avaliação de similaridade

O processo de avaliação de similaridade proposto neste trabalho utilizou a abordagem mais elementar para a comparação entre os conceitos, ou seja, cada conceito foi comparado com todos os outros para se chegar ao valor de similaridade. Esta abordagem, conforme discutido

em seção anterior, pode se tornar inviável em conjuntos de dados com maior quantidade de registros.

Em média, a avaliação de similaridade entre cada par de conceitos dos estudos de caso apresentados levou 0,005s. O tempo para avaliação de toda a taxonomia em construção do primeiro estudo de caso, por exemplo, levou 24h.

A fim de tentar uma melhoria de desempenho, um processo de avaliação de similaridade com uso da técnica de vizinhos ordenados (*sorted neighborhoods*) (HERNANDEZ; STOLFO, 1998) foi criado. No processo de avaliação de similaridade com vizinhos ordenados foram utilizadas duas chaves de ordenação: o nome do conceito e o *token* de maior tamanho obtido através da *tokenização* em palavras usando letras maiúsculas como delimitador, conforme Quadro 11.

Quadro 11 - Chaves utilizadas para melhoria de desempenho

NOME	CHAVE 01	CHAVE 02
AdicionalICMSFundoDeCombateAPobreza	AdicionalICMSFundoDe	Adicional
AdjustmentToAcquisitionOfNFAs	AdjustmentToAcquisit	Acquisition
AdjustmentToCashSurplusDeficit	AdjustmentToCashSurp	Adjustment
AdjustmentToExpense	AdjustmentToExpense	Adjustment
AdjustmentToGrants	AdjustmentToGrants	Adjustment
AdjustmentToNetAcquisitionOfFinancialAssets	AdjustmentToNetAcqui	Acquisition
AdjustmentToNetAcquisitionOfNFAs	AdjustmentToNetAcqui	Acquisition
AdjustmentToNetIncurrenceOfDomesticLiabilities	AdjustmentToNetIncur	Liabilities
AdjustmentToNetIncurrenceOfForeignLiabilities	AdjustmentToNetIncur	Liabilities
AdjustmentToNetIncurrenceOfLiabilities	AdjustmentToNetIncur	Liabilities
AdjustmentToNetOperatingBalance	AdjustmentToNetOpera	Adjustment
AdjustmentToOtherRevenue	AdjustmentToOtherRev	Adjustment
AdjustmentToPurchasesOfNonfinancialAssets	AdjustmentToPurchase	Nonfinancial
AdjustmentToRevenue	AdjustmentToRevenue	Adjustment
AdjustmentToSalesOfNonfinancialAssets	AdjustmentToSalesOfN	Nonfinancial
AdjustmentToSocialContributions	AdjustmentToSocialCo	Contributions
AdjustmentToTaxRevenue	AdjustmentToTaxReven	Adjustment
AjusteDePerdasDeCreditosACurtoPrazo	AjusteDePerdasDeCred	Creditos
AjusteDePerdasDeCreditosALongoPrazo	AjusteDePerdasDeCred	Creditos
AjustesDeExerciciosAnteriores	AjustesDeExerciciosA	Exercicios

Para cada chave de ordenação (como na técnica de *multipass*), foi deslocada uma janela de tamanho 20 e a avaliação de similaridade foi executada nos conceitos de cada janela, reduzindo o número de comparações.

O ganho no desempenho foi de 70%. Entretanto, 20% dos pares com similaridade maior do que 0,5 deixaram de ser detectados com o uso de vizinhos ordenados.

A análise de desempenho do processo de avaliação de similaridade não é o foco deste trabalho e merece ser aprofundado em trabalhos futuros para resolver questões tais quais: que chaves são mais eficientes? Qual o melhor tamanho de janela para deslocamento? O método de vizinhos ordenados é o melhor neste contexto?

6 CONCLUSÃO

A linguagem XBRL vem se tornando cada vez mais relevante para a padronização da divulgação e intercâmbio de informações financeiras de maneira eletrônica. Por outro lado, o uso frequente da linguagem XBRL e a criação crescente de conceitos que representam as informações financeiras a serem reportadas introduz a necessidade de aplicação da disciplina de Gestão da Qualidade de Dados neste contexto.

Este trabalho identificou que a aplicação de um processo de avaliação de similaridade entre conceitos representados pela XBRL pode ser útil para a melhoria na qualidade das informações apresentadas por meio desta linguagem, por ser um processo relevante para a Gestão da Qualidade de Dados.

Ainda há carência de estudos para avaliação de similaridade neste contexto da linguagem XBRL. A contribuição dessa dissertação consiste na proposição de um processo de avaliação de similaridade entre conceitos representados pela linguagem XBRL e a comprovação de sua utilidade por meio de sua aplicação experimental em um conjunto de conceitos da XBRL. Incorporado ao processo, o trabalho gerou medidas de similaridade que uniram técnicas associadas ao modelo relacional e ao XML em medidas mais adaptadas ao contexto da XBRL, devido às suas características.

O estudo apresentado neste trabalho incorpora informações sobre os conceitos a fim de melhorar o processo de avaliação de similaridade: informações sobre os *links label* e informações sobre o relacionamento entre os conceitos nas estruturas *calculation* e *presentation*. Esta característica introduz uma contribuição inovadora, visto que a reunião de conteúdo e estrutura em uma avaliação de similaridade em dados XBRL é uma questão recente e pouco discutida pela comunidade acadêmica.

Dentre as limitações deste trabalho, a que mais se destaca é a impossibilidade de classificar os resultados da aplicação do processo proposto segundo o conhecimento do especialista. Sendo assim, a avaliação dos resultados ficou limitada a análises e projeções, dificultando a aplicação de métricas de qualidade.

A dificuldade de aplicar o processo quando os nomes dos conceitos são representados por códigos também pode ser considerado outro fator limitante encontrado no desenvolvimento deste trabalho.

6.1 PRINCIPAIS CONTRIBUIÇÕES

Entende-se que o desenvolvimento deste trabalho alcançou as seguintes contribuições:

- Proposição de um processo de avaliação de similaridade entre conceitos representados pela XBRL, consistindo de duas atividades principais: carga e avaliação de similaridade.

- Identificação e definição de medidas de similaridade a serem aplicadas no contexto dos conceitos representados pela XBRL;

- Especificação e implementação das rotinas para executar o processo de avaliação de similaridade entre conceitos representados pela XBRL proposto;

- Realização de estudos de casos em domínios diferentes, a fim de avaliar a aplicabilidade do processo proposto e especificado.

6.2 TRABALHOS FUTUROS

Algumas indicações de trabalhos futuros são:

- Explorar os resultados das avaliações de similaridade sobre taxonomias diversas, submetendo-os à análise de especialistas para avaliar a eficiência do processo.

- Estender o processo proposto para a avaliação de conceitos a partir de atributos-chave.

- Desenvolver uma interface para execução por usuários leigos, a fim de facilitar o envolvimento dos especialistas. A interface deve conter algoritmos de agrupamento (clustering) que facilitem a visão dos conceitos relacionados, permitindo uma análise mais visual e intuitiva dos resultados da avaliação de similaridade.

- Incorporar ao processo informações relacionadas às especificações complementares à especificação base da XBRL que podem enriquecer a avaliação de similaridade de estrutura entre os conceitos: XBRL *Dimension* e XBRL *Formula.*

- Enriquecer a avaliação de similaridade entre conceitos com o uso de thesaurus e dicionários que detectem semelhanças semânticas entre as informações utilizadas no processo;
- Avaliar o desempenho do processo. A definição das chaves de ordenação deve ser analisada para a especificação do método que propicie melhor custo-benefício no contexto dos conceitos representados pela XBRL.

REFERÊNCIAS

ALLEN, P. ; STOKES-REE, I **EInline XBRL Part 1:** Specification 1.0. 2010. Disponível em: <http://www.xbrl.org/Specification/inlineXBRL-part1/REC-2010-04-20/inlineXBRL-part1-REC-2010-04-20+corrected-errata-2011-08-17.html>. Acesso em: 13 jul. 2012.

ALGERGAWY, A. et al. XML Data Clustering: An Overview. **ACM Computing Surveys**, v.43, n.4, p. 1-41, 2011.

ANANTHAKRISHNA, R.; CHAUDHURI, S. ; GANTI, V. Eliminating fuzzy duplicates in data warehouses. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES, 2002. **Proceedings…** 2002. p.586-597.

AUGSTEN, N.; BOHLEN, M. ; GAMPER, J. E. Approximate Matching of Hierarquical Data using pq-Grams. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES, 2005. **Proceedings…** 2005. p. 918-929.

BAEZA-YATES, R. ; RIBEIRO-NETO, B. **Modern Information Retrieval**. 1. ed. Nova York: ACM Press ; Addison-Wesley, 1999.

BARTLEY, J.; CHEN, A. Y. S. ; TAYLOR, E. Z. A comparison of XBRL filings to corporate 10-ks-evidence from the voluntary filing program. **Accounting Horizons**, v. 25, n. 2, p. 227-245, 2011.

BAXTER, R.; CHRISTEN, P. ; CHURCHES, T. A comparison of fast blocking methods for record linkage. **ACM SIGKDD Workshop on Data Cleaning, Record Linkage and Object Consolidation**. p.25-27, 2003.

BERNERS-LEE, T.; FIELDING, R. ; MASINTER, L. **RFC3986:** Uniform Resource Identifier (URI): Generic Syntax. 2005. Disponível em: <http://www.rfc-editor.org/rfc/rfc3986.txt>. Acesso em: 28 ago. 2012

BILENKO, M. et al. Adaptive name matching in information integration. **IEEE Inteligent Systems,** v.18, n.5, pp.16-23, 2003.

BILENKO, M.; KAMATH, B. ; MOONEY, R.J. Adaptive Blocking: Learning to Scale Up Record Linkage. In: IEEE CONFERENCE ON DATA MINING, 2006. **Proceedings …** 2006. p. 9-96.

BORITZ, E. ; NO, W. G. Auditing XBRL-related documents: the case of United Technologies Corporation. 2008. Disponível em: <http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1288376> Acesso em: 28 ago. 2012

BRAY, T. et al. **Extensible Markup Language (XML) 1.0, W3C Recommendation**, 2008. Disponível em: <http://www.w3.org/TR/2008/REC-xml-20081126/>. Acesso em: 28 ago. 2012.

BRAY, T. et al. **Namespaces in XML 1.0 (Third Edition)**, 2009. Disponível em: <http://www.w3.org/TR/2009/REC-xml-names-20091208/>. Acesso em: 28 ago. 2012.

CARVALHO, J. P. ; SILVA, A. S. Finding Similar Identities among Objects from Multiple Web Sources. **CIKM-2003 Workshop on Web Information and Data Management**, p.90-93, 2003.

COHEN W. W. Integration of Heterogeneous Databases Without Common Domains Using Queries Based on Textual Similarity. In: ACM INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA (SIGMOD), 1998. **Proceedings…** 1998. p.201-212.

COHEN, W. W.; RICHMAN, J. Learning to match and cluster large high-dimensional data sets for data integration. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING (KDD), 2002. **Proceedings…** 2002. p.475-480.

DEROSE, S.; MALER, E.; ORCHARD, D. **XML Linking Language (XLINK) 1.0, W3C Recommendation**, 2001. Disponível em: <http://www.w3.org/TR/xlink>. Acesso em: 18 jul. 2012.

ELMAGARMID, A.K.; IPERIOTIS, P.G. ; VERYKIOS, V.S. Duplicate record detection: A survey. **IEEE Transactions on Knowledge and Data Engineering**, v.19, n.1, p. 2007.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. São Paulo: Pearson Education do Brasil, 2005.

ENGEL, P. et al. **Extensible Business Reporting Language (XBRL) 2.1.** 2003. Disponível em: < http://www.xbrl.org/Specification/XBRL-RECOMMENDATION-2003-12-31.pdf>. Acesso em: 3 jul. 2012.

ENGEL, P. et al. **Formula 1.0.** 2009. Disponível em: <http://www.xbrl.org/Specification/formula/REC-2009-06-22/formula-REC-2009-06-22.html>. Acesso em: 10 jul. 2012.

FALLSIDE, D. C. ; WALMSLEY, P. **XML SCHEMA Part 0:** Primer W3C Recommendation. 2004. Disponível em: < http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>. Acesso em: 3 set. 2012.

FELLEGI, I. P. ; SUNTER, A. B. A Theory for record linkage. **Journal of the American Statistical Association**. v.64, n.328, p.1183-1210, 1969.

GRAVANO, L. et al. Text Joins in an RDBMS for Web Data Integration. In: INTERNATIONAL WORLD WIDE WEB CONFERENCE (WWW12), 2003. **Proceedings…** 2003. p.267-270.

GUHA, S. et al.. Integrating XML Data Sources Using Approximate Joins. **ACM Transactions on Database Systems,** v.31, n.1, p.161-207, 2006.

HERNANDEZ, M. A. ; STOLFO, S. J. Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem. **Data Mining and Knowledge Discovery**, v.2, n.1, p.9-37, 1998.

HERNÁNDEZ-ROS, I. ; WALLIS, H. **XBRL Dimensions 1.0**, 2012. Disponível em: <http://www.xbrl.org/specification/dimensions/rec-2012-01-25/dimensions-rec-2006-09-18+corrected-errata-2012-01-25-clean.html>. Acesso em: 11 jul. 2012.

HOFFMAN, C. **Financial Reporting Using XBRL:** IFRS and US GAAP. 1. ed. Lulu: [s.n], 2006.

HOMMES, R. ; WARREN, P. **Versioning Base 1.0**, 2013. Disponível em: <http://www.xbrl.org/Specification/versioning-base/REC-2013-02-27/versioning-base-REC-2013-02-27.html>. Acesso em: 3 mar. 2013.

JARO, M. A. **UNIMATCH:** A Record Linkage System, User's Manual. 1. ed. Washington, DC: U.S.: Bureau of the Census, 1976.

KADE A. M. ; HEUSER, C. A. Matching XML Documents in Highly Dynamic Applications. In: ACM Symposium on Document Engineering,2008. **Proceedings…** 2008. p.191-198.

LEVENSHTEIN, V. I. Binary codes capable of correcting deletions, insertions, and reversals. **Soviet Physics Doklady,** v.10, n.8, p.707-710, 1966.

LUCENA, F. J. T. **Busca Fonética em Português do Brasil.** [S.l.]: [s.n.], 2006.

MONGE, A. E.; ELKAN, C. P. The Field Matching Problem: Algorithms and Application. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING (KDD), 1996. **Proceedings…** 1996. p.267-270.

RIBEIRO, L. ; HARDER T. Embedding Similarity Joins into Native XML Databases. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS, 22., 2007. **Anais...** 2007. p.285-299.

SILVA, P.C. **Explorando linguagens de marcação para representação de relatórios de informações financeiras**. 2003. Dissertação (Mestrado em Sistemas e Computação) - Universidade Salvador – UNIFACS, 2003.

SILVA, P.C. **Análise Multidimensional de Dados XML baseados em Links:** Modelos e Linguagens. 2010. Tese (Doutorado – Ciências da Computação) – Universidade Feredal de Pernambuco - UFPE, 2010.

SILVA, P. C; SILVA, L.G. ; AQUIONO JR., I. J. S. **XBRL Conceitos e Aplicações**. 1. ed. Rio de Janeiro: Ciência Moderna, 2006.

SMITH, T. F. ; WATERMAN, M. S. Identification of Common Molecular Subsequences, **Journal Molecular Biology**, v.147, p.195-197, 1981.

TEJADA, S.; KNOBLOCK, C. A. ; MINTON, S. Learning domain-independent string transformation weights for high accuracy object identification. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING (KDD), 2002. **Proceedings**… 2002. **…** p.350-359.

TEKLI, J.; CHBEIR, R.; YETONGNON, K. An overview on XML similarity: background, current trends and future directions. **Computer Science Review**, v.3, i.3, p.151-173, 2009.

TAFT, R. L. **Name Search Techniques**. 1. Albany: New York State Identification and Intelligence System. [S.l.]: [s.n.], 1970.

UKKONEN, E. Approximate String Matching with q-Grams and Maximal Matches. **Theoretical Computer Science**, v.92, n.1, p.191-211, 1992.

VINSON, A. R. **PathSim**: um algoritmo para calcular a similaridade entre caminhos XML. 2007. 71 f. Dissertação(Mestrado) – Ciência da Computação. Universidade Feredarl do Rio Grande do Sul, Porto Alegre, 2007.

WATERMAN, M. S.; SMITH, T. F.; BEYER, W. A. Some biological sequence metrics. **Advances in Math**, v.20, n.4, p.367-387, 1976.

WEISS, M. ; NAUMANN, F. DogmatiX Tracks down Duplicates in XML. In: ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA. 2005. **Proceedings…** 2005. p.96-110.

WINKLER, W. E. ; THIBAUDEAU, Y. An Application of the Fellegi-Sunter Model of Record Linkage to the 1990 US. **Decennial Census Technical Report Statistical Research Report Series US Bureau of the Census**, v.9, 1991.

ZHU, H.; WU, H. Quality of data standards: framework and illustration using XBRL taxonomy and instances. **Electronic Markets**, v. 21, n. 2, p. 129-139, 2011.

APÊNDICE A – Roteiros de criação de tabelas, funções e procedimentos no banco de dados

```
/*****************************************************/
/*                                                   */
/* Create TablesProcedure                            */
/* Script: createTables.sql                          */
/* Descrição: Comandos para criação das tabelas da base de*/
*/           dados                                   */
/*                                                   */
/*****************************************************/

/*** Table SchemaXSD ***/
create table [SchemaXSD] (
       [ID]                bigint       IDENTITY(1,1) NOT NULL,
       [fileName]          nvarchar(50)               NOT NULL,
       [preffix]           nvarchar(50)               NOT NULL,
    [obs]               nvarchar(MAX)                NULL,
    constraint PK_SchemaXSD primary key clustered (ID asc));

/*** Table Concept ***/
create table [Concept] (
    [ID]                bigint       IDENTITY(1,1) NOT NULL,
       [schemaXSDID]       bigint                     NOT NULL,
       [name]              nvarchar(max)              NOT NULL,
       [conceptId]         nvarchar(max)              NOT NULL,
       [type]              nvarchar(max)              NOT NULL,
       [abstract]          char(01)                   NOT NULL,
       [periodType]        char(10)                   NOT NULL,
       [substitutionGroup] nvarchar(50)               NOT NULL,
       [key1name20]        char(20)                   NOT NULL,
    constraint PK_Concept primary key clustered (ID asc));
go
/*** Concept X SchemaXSD ***/
alter table Concept add constraint FK_Concept_SchemaXSD FOREIGN KEY(schemaXSDID)
references SchemaXSD (ID);
go
/*** Índice da Key1 ***/
create nonclustered index [IX_Key1] ON [dbo].[Concept] ([key1name20] ASC);
go
/*** Table Label ***/
create table [Label] (
    [ID]                bigint       IDENTITY(1,1) NOT NULL,
       [conceptID]         bigint                     NOT NULL,
       [content]           nvarchar(max)              NOT NULL,
       [roleType]          nvarchar(max)              NOT NULL,
       [lang]              nvarchar(50)               NOT NULL,
    constraint PK_Label primary key clustered (ID asc));
go
/*** Label X Concept ***/
alter table Label add constraint FK_Label_Concept FOREIGN KEY(conceptID) references
Concept (ID);
go
/*** Table Calculation ***/
create table [Calculation] (
    [ID]                bigint       IDENTITY(1,1) NOT NULL,
       [schemaXSDID]       bigint                     NOT NULL,
       [fromConceptID]     bigint                     NOT NULL,
       [toConceptID]       bigint                     NOT NULL,
       [weight]            int                        NOT NULL,
```

```
      [link]             int                         NOT NULL,
    constraint PK_Calculation primary key clustered (ID asc));
go
/*** Calculation X SchemaXSD ***/
alter table Calculation add constraint FK_Calculation_SchemaXSD   FOREIGN
KEY(schemaXSDID)   references SchemaXSD (ID);
go
/*** Calculation X Concept (from) ***/
alter table Calculation add constraint FK_Calculation_ConceptFrom FOREIGN
KEY(fromConceptID) references Concept (ID);
go
/*** Calculation X Concept (to) ***/
alter table Calculation add constraint FK_Calculation_ConceptTo   FOREIGN
KEY(toConceptID)   references Concept (ID);
go
/*** Table Presentation ***/
create table [Presentation] (
    [ID]               bigint       IDENTITY(1,1) NOT NULL,
      [schemaXSDID]       bigint                     NOT NULL,
      [fromConceptID]     bigint                     NOT NULL,
      [toConceptID]       bigint                     NOT NULL,
      [order]             int                        NOT NULL,
      [link]              int                        NOT NULL,
    constraint PK_Presentation primary key clustered (ID asc));
go
/*** Presentation X SchemaXSD ***/
alter table Presentation add constraint FK_Presentation_SchemaXSD   FOREIGN
KEY(schemaXSDID)   references SchemaXSD (ID);
go
/*** Presentation X Concept (from) ***/
alter table Presentation add constraint FK_Presentation_ConceptFrom FOREIGN
KEY(fromConceptID) references Concept (ID);
go
/*** Presentation X Concept (to) ***/
alter table Presentation add constraint FK_Presentation_ConceptTo   FOREIGN
KEY(toConceptID)   references Concept (ID);
go
/*** Table Token ***/
create table [NameTokenWord] (
    [ID]               bigint       IDENTITY(1,1) NOT NULL,
      [conceptID]         bigint                     NOT NULL,
      [token]             nvarchar(max)              NOT NULL,
      constraint PK_ConceptToken primary key clustered (ID asc));
/*** NameTokenWord X Concept ***/
alter table NameTokenWord add constraint FK_NameTokenWord_Concept FOREIGN
KEY(conceptID) references Concept(ID);
go
/*** Table simEvalResult ***/
create table SimEvalResult (
   id                            bigint        identity(1,1) not null,
   ConceptAID                    bigint        NOT NULL,
   ConceptAName                  nvarchar(max) NOT NULL,
   ConceptBID                    bigint        NOT NULL,
   ConceptBName                  nvarchar(max) NOT NULL,
   NameQGramsSimilarity          decimal(8,2)  NOT NULL,
   NameTokenSimilarity           decimal(8,2)  NOT NULL,
   LabelSimilarity               decimal(8,2)  NOT NULL,
   CalcTopDownSimilarity         decimal(8,2)  NOT NULL,
   CalcBottomUpSimilarity        decimal(8,2)  NOT NULL,
   PresTopDownSimilarity         decimal(8,2)  NOT NULL,
   PresBottomUpSimilarity        decimal(8,2)  NOT NULL,
   constraint PK_SimEvalResult primary key clustered (ID asc));
```

```
CREATE procedure Upload (@filename nvarchar(50), @preffix nvarchar(50), @obs
nvarchar(max))
AS
BEGIN
/*****************************************************************************/
/*                                                                           */
/* Procedure Upload                                                          */
/* Script: Upload.sql                                                        */
/* Descrição: Procedimento de carga de uma taxonomia nas tabelas da base de  */
/*            dados                                                          */
/*                                                                           */
/*****************************************************************************/

       declare @id    bigint;
       declare @minid bigint;
       declare @maxid bigint;

       /* Insere o SchemaXSD */
       insert into SchemaXSD values (@filename, @preffix, @obs);

       /* Recupera ID do Schema inserido */
       select @id = id from SchemaXSD where fileName = @fileName and preffix = @preffix
and obs = @obs;

       /* Apaga conceitos que não sejam do schema sendo carregado */
       delete from [UploadConcepts] where label not like @preffix + ':%';

       /* Insere os conceitos na tabela Concept */
       insert into concept
       select @id,
                name, id, type, iif(abstract='true','T','F'), left([period type],10),
left([subs grp],50), left(name,20), ''
         from [UploadConcepts];

       /* Limpa os espaços dos labels */
       update [UploadLabels] set [conc] = ltrim([conc]);

       /* Popula tabela temporaria para calcular referencia dos labels */
       delete from [UploadLabels2];

       /* Insere os conceitos pai e filho(label) na tabela temporária */
       insert into [uploadLabels2]
       select pai.[conc] as concpai, filho.[conc] as concfilho, filho.[resourcerole] as
roleType, filho.lang
         from [UploadLabels] pai, [UploadLabels] filho
        where pai.ArcRole is null
          and filho.ArcRole = 'concept-label'
          and ((filho.seq  = pai.seq + 1) or (filho.seq = pai.seq + 2));

       /* Insere na tabela Label com as referências definidas */
       insert into Label
       select conc.ID as conceptID, lb.concFilho as content, lb.roletype as roletype,
lb.lang as lang
         from [UploadLabels2] lb, Concept conc
        where lb.concpai = @preffix + ':'  + conc.name
          and conc.schemaXSDID = @id;

       /* Limpa nomes da tabela calculation */
       update [UploadCalc] set conc = ltrim(replace(replace(conc,'(1)',''),'(-1)',''));

       /* Atualiza a referência dos conceitos filho */
```

```
        update [UploadCalc]
           set idconc = concept.id
          from schemaXSD, concept, [UploadCalc]
         where schemaXSD.id = concept.schemaXSDID
           and [UploadCalc].conc    = schemaXSD.preffix + ':' + concept.name;

        /* Atualiza a referência dos conceitos pai */
        update [UploadCalc]
           set idconcpai = pai.idconc
          from [UploadCalc] pai, [UploadCalc]
         where pai.seq = (select max(pai2.seq) from [UploadCalc] pai2 where pai2.seq <
[UploadCalc].seq and pai2.niv = [UploadCalc].niv-1);

        /* Insere os relacionamentos calculation na tabela definitiva */
        insert into calculation
        select @id,
                  idconcpai, idconc, [weight], [link]
          from [UploadCalc]
         where [weight] is not null and idconc is not null and idconcpai is not null;

        /* Limpa os nomes da tabela presentation */
        update [UploadPres] set conc = ltrim(conc);

        /* Atualiza a referencia dos conceitos filho */
        update [UploadPres]
           set idconc = concept.id
          from schemaXSD, concept, [UploadPres]
         where schemaXSD.id = concept.schemaXSDID
           and [xrscPres].conc    = schemaXSD.preffix + ':' + concept.name
           and Concept.schemaXSDID <> 1;

        /* Atualiza a referencia dos conceitos pai */
        update [UploadPres]
           set idconcPai = pai.idconc
          from [UploadPres] pai, [UploadPres]
         where pai.seq = (select max(pai2.seq) from [UploadPres] pai2 where pai2.seq <
[UploadPres].seq and pai2.niv = [UploadPres].niv-1);

        /* Insere os relacionamentos presentation na tabela definitiva */
        insert into presentation
        select @id,
                  idconcpai, idconc, seq, link
          from [UploadPres]
         where idconc is not null and idconcpai is not null;

        /* Transforma os nomes dos conceitos inseridos em tokens */
        select @minid = min(id), @maxid = max(id) from concept where schemaXSDID = @id;

        exec TokenizeConcepts @minid, @maxid

END;
```

```
CREATE procedure SimilarityEvaluation (@schemaXSDIDA bigint, @schemaXSDIDB bigint,
@threshold decimal(8,2))
AS
BEGIN
/*****************************************************************************/
/*                                                                           */
/* Procedure SimilarityEvaluation                                            */
/* Script: SimilarityEvaluation.sql                                          */
/* Descrição: Procedimento do processo de avaliação de similaridade entre    */
/*            conceitos representados pela XBRL                              */
/*                                                                           */
/*****************************************************************************/

  /* Declaracao de variaveis */
  declare @idA                          bigint;
  declare @idB                          bigint;
  declare @nameA                        nvarchar(max);
  declare @nameB                        nvarchar(max);
  declare @nameQGramsSim                decimal(8,2);
  declare @nameTokenSim                 decimal(8,2);
  declare @labelSim                     decimal(8,2);
  declare @calcTopDownSim               decimal(8,2);
  declare @calcBottomUpSim              decimal(8,2);
  declare @presTopDownSim               decimal(8,2);
  declare @presBottomUpSim              decimal(8,2);
  declare @conttotal                    bigint;
  declare @contbloco                    bigint;
  declare @contpar                      bigint;
  declare @dtinicio                     datetime;

  select @dtinicio = getdate();

  /* Inicio */
  PRINT concat('INICIO - ', convert(varchar(8), @dtinicio, 108));

  /* Limpa a tabela para incluir os resultados */
  delete from SimEvalResult;
  print concat('APAGOU TABELA - ', convert(varchar(8), getdate() - @dtinicio, 108));

  select @conttotal = 0, @contbloco = 0, @contpar = 0;

  /* Percorre os pares de conceitos entre os schemas */
  declare pairs cursor for
   select conca.id, conca.name, concb.id, concb.name
     from concept   conca  , concept   concb
    where conca.schemaXSDID = @schemaxsdidA
      and concb.schemaXSDID = @schemaxsdidB
      and conca.abstract    = 'F'
      and concb.abstract    = 'F'
         and concb.id          > iif(@schemaxsdidA = @schemaxsdidB, conca.id, 0);

  open pairs;

  fetch pairs into @idA, @nameA, @idB, @nameB;

  while @@FETCH_STATUS = 0
    begin
      /* Verifica similaridade de nome entre os conceitos */
         select @nameQGramsSim = dbo.QGramsSim(@nameA, @nameB), @conttotal = @conttotal
+ 1, @contbloco = @contbloco + 1;
```

```
         /* Divulga resultados */
         If @contbloco >= 1000
           begin
             PRINT concat('PERCORREU 1000 REGISTROS (', @conttotal,') - ',
convert(varchar(8), getdate() - @dtinicio, 108));
               select @contbloco = 0;
            end;

         /* Se a similaridade entre os nomes for maior do que o limite passado por
parametro,
            continua o processamento, caso contrário, passa para o próximo par */
      If @nameQGramsSim >= @threshold
           begin
               select @contpar = @contpar + 1;

            PRINT concat('ENCONTROU PAR ', @contpar, '(', @idA, ', ', @idB, ') - ',
convert(varchar(8), getdate() - @dtinicio, 108));

               select @nameTokenSim = dbo.NameTokenSim(@idA, @idB);

               /* Verifica similaridade entre os labels */
               select @labelSim = dbo.LabelSim(@idA, @idB);

               /* Verifica similaridades da estruturas calculation e presentation
TopDown e BottomUp*/
               select @calcTopDownSim  = dbo.TopDownSim (@schemaXSDIDA, @idA,
@schemaXSDIDB, @idB,'C', 0.3);--@threshold);
          select @calcBottomUpSim = dbo.BottomUpSim(@schemaXSDIDA, @idA,
@schemaXSDIDB, @idB,'C', 0.3);--@threshold);
               select @presTopDownSim  = dbo.TopDownSim (@schemaXSDIDA, @idA,
@schemaXSDIDB, @idB,'P', 0.3);--@threshold);
          select @presBottomUpSim = dbo.BottomUpSim(@schemaXSDIDA, @idA,
@schemaXSDIDB, @idB,'P', 0.3);--@threshold);

          /* Insere o par na tabela de resultados */
          insert into SimEvalResult
          select @idA,    @nameA, @idB, @nameB,
                 @nameQGramsSim , @nameTokenSim   , @labelSim,
                           @calcTopDownSim, @calcBottomUpSim,
                           @presTopDownSim, @presBottomUpSim

            --PRINT concat('PROCESSOU PAR ', @contpar, '(', @idA, ', ', @idB, ') - ',
convert(varchar(8), getdate() - @dtinicio, 108));
             end;

        fetch pairs into @idA, @nameA, @idB, @nameB;
      end;

  close pairs;

  deallocate pairs;

  PRINT concat('FIM - ', convert(varchar(8), getdate() - @dtinicio, 108));

END;
```

```
CREATE FUNCTION QGramsSim (@s1 nvarchar(max), @s2 nvarchar(max))

RETURNS decimal(8,2)

AS

BEGIN

/*****************************************************************************/
/*                                                                           */
/* Function QGramsSim                                                        */
/* Script: QGramsSim.sql                                                     */
/* Descrição: Calcula e retorna a similaridade entre duas strings através do  */
/*            método QGrams                                                  */
/*                                                                           */
/*****************************************************************************/

     /*** Declaracao de Variaveis ***/
        DECLARE @i          bigint;
        DECLARE @uniao      bigint;
        DECLARE @intersecao bigint;
        DECLARE @sim        decimal(8,2);

        DECLARE @ls1        nvarchar(max);
        DECLARE @ls2        nvarchar(max);

        DECLARE @ts1        table (qgrams char(03));
        DECLARE @ts2        table (qgrams char(03));

        /*** Pre-processamento das strings ***/
        -- Retira os espaços e transforma as strings em maiusculas
        set @ls1 = REPLACE(UPPER(@s1),' ','');
        set @ls2 = REPLACE(UPPER(@s2),' ','');

        /*** Tokenizacao em QGrams de Tamanho 3 ***/
        -- Carrega os qgrams da primeira string
        set @i = 1;
        IF len(@ls1) <= 3
          BEGIN
            INSERT INTO @ts1 values (@ls1);
          END
        ELSE
          BEGIN
           WHILE @i + 1 < len(@ls1)
                BEGIN
                  INSERT INTO @ts1 values (substring(@ls1,@i,3));
                  SET @i = @i + 1;
                END;
          END;

        -- Carrega os qgrams da segunda string
        set @i = 1;
        IF len(@ls2) <= 3
          BEGIN
            INSERT INTO @ts2 values (@ls2);
          END
        ELSE
          BEGIN
           WHILE @i + 1 < len(@ls2)
                BEGIN
                  INSERT INTO @ts2 values (substring(@ls2,@i,3));
                     SET @i = @i + 1;
```

```
            END;
          END;

        /*** Calculo da medida de similaridade ***/
        select @intersecao = count(distinct a.qgrams) from @ts1 as a, @ts2 as b where
a.qgrams = b.qgrams;
        select @uniao      = count(*) from (select * from @ts1 union select * from
@ts2) as uniao;
        If @uniao <> 0
          select @sim        = cast(@intersecao as decimal)/ cast(@uniao as decimal)
        Else
          select @sim        = 1;

     RETURN @sim;

END;
```

```
CREATE FUNCTION NameTokenSim (@idconcA bigint, @idconcB bigint)

RETURNS decimal(8,2)--@ret TABLE ( istring  INT,  QGrams   CHAR(03))

AS

BEGIN
/*****************************************************************************/
/*                                                                           */
/* Function NameTokenSim                                                     */
/* Script: NameTokenSim.sql                                                  */
/* Descrição: Retorna a similaridade entre os nomes dos conceitos usando os  */
/*            tokens da tabela NameTokenWord                                 */
/*                                                                           */
/*****************************************************************************/

     /*** Declaracao de Variaveis ***/
        DECLARE @uniao      bigint;
        DECLARE @intersecao bigint;
        DECLARE @sim        decimal(8,2);

        /*** Calculo da medida de similaridade usando os tokens armazenados na tabela
NameTokenWord ***/
        select @intersecao = count(distinct a.token) from NameTokenWord as a,
NameTokenWord as b where a.conceptID = @idconcA and b.conceptID = @idconcB and a.token
= b.token;
        select @uniao      = count(*) from (select token from NameTokenWord where
conceptid = @idconcA union select token from NameTokenWord where conceptid = @idconcB)
as uniao;
        If @uniao <> 0
          select @sim        = cast(@intersecao as decimal)/ cast(@uniao as decimal)
        Else
          select @sim        = 1;

     RETURN @sim;

END;
```

```
CREATE FUNCTION LabelSim (@conca bigint, @concb bigint)

RETURNS decimal(8,2)

AS

BEGIN

/*****************************************************************************/
/*                                                                           */
/* Função LabelSim                                                           */
/* Script: LabelSim.sql                                                      */
/* Descrição: Retorna a similaridade entre os labels de dois conceitos       */
/*            recebidos como parâmetro                                       */
/*                                                                           */
/*****************************************************************************/

        DECLARE @sim        decimal(8,2);

     select @sim = ISNULL( avg(dbo.QGramsSim(LabelConcA.content,LabelConcB.content) ),
0)
       from Label LabelConcA, Label LabelConcB
      where LabelConcA.conceptID = @conca
        and LabelConcB.conceptID = @concb
              and labelConcA.roleType  <> 'documentation'
              and LabelConcB.roleType  <> 'documentation';

     RETURN @sim;

END;
```

```
CREATE FUNCTION BottomUpSim (@schemaXSDIDconca bigint, @idconca bigint,
@schemaXSDIDconcb bigint, @idconcb bigint, @tpstruct char(01), @threshold
decimal(8,2))

RETURNS decimal(8,2)

AS

BEGIN

/*****************************************************************************/
/*                                                                           */
/* Função BottomUpSim                                                        */
/* Script: BottomUpSim.sql                                                   */
/* Descrição: Retorna a similaridade entre os relacionamentos calculation ou  */
/*            presentation de dois conceitos (a depender do parâmetro)       */
/*            utilizando o método bottom-up                                  */
/*                                                                           */
/*****************************************************************************/

     /*** Declaracao de Variaveis ***/
        DECLARE @sim        decimal(8,2);

        DECLARE @cont           int;
        DECLARE @count          int;
        DECLARE @conceito       int;
        DECLARE @idconcFolha    bigint;
        DECLARE @nmconcFolha    nvarchar(max);
        DECLARE @pathConcFolha  nvarchar(max);
        DECLARE @linkConcFolha  int;
        DECLARE @conceptPatha   nvarchar(max);
        DECLARE @conceptPathb   nvarchar(max);

        DECLARE @ttree          table (conceito char(01)   , cont    int         ,
idPai   bigint      , idFilho    bigint      , conceptPath nvarchar(max), link int);
        DECLARE @tsim           table (pathsim decimal(8,2));

     set @sim = 0;

        /* Monta as subárvores de cada conceito */

        set @conceito = 1;

        while @conceito < 3
          begin

               -- Inicializa o contador
               select @cont         = count(*)+1 from @ttree;
               select @count        = @cont;

               -- Insere o nó folha inicial
               insert into @ttree
                      select iif(@conceito=1,'A','B'), @cont, id, null, name, 0
                        from Concept
                       where id = iif(@conceito=1,@idconca,@idconcb);

               -- Percorre todos os pais do nó folha, montando a arvore
               while @cont <= @count
                 begin
                      -- Recupera as informacoes do pai do conceito sendo trabalhado
                      select @idconcFolha = idpai,@pathConcFolha = conceptPath,
@linkConcFolha = link
```

```
                    from @ttree where cont = @cont;

                  -- Insere todos os pais desse conceito
                  If (@tpstruct = 'C')
                    begin
                         insert into @ttree
                         select distinct iif(@conceito = 1, 'A', 'B'),
                                @count + ROW_NUMBER() OVER(ORDER BY calc.id),
                                      calc.fromConceptID, calc.toConceptID,
                                      conc.name + '/' + @pathConcFolha,
                                      link
                           from Calculation calc, Concept conc
                          where toConceptID      = @idconcFolha
                                and conc.ID          = calc.toConceptID
                                and calc.link        = iif(@linkConcFolha = 0,
                                                           (select min(calc2.link)
from calculation calc2 where calc2.toConceptID = @idconcfolha and calc2.schemaxsdid =
iif(@conceito = 1, @schemaXSDIDconca, @schemaXSDIDconcb)),--calc.link,

@linkConcFolha)
                                and calc.schemaXSDID = iif(@conceito = 1,
@schemaXSDIDconca, @schemaXSDIDconcb);
                      end
                    Else
                      begin
                         insert into @ttree
                         select distinct iif(@conceito = 1, 'A', 'B'),
                                @count + ROW_NUMBER() OVER(ORDER BY pres.id),
                                      fromConceptID     , toConceptID  ,
                                      conc.name + '/' + @pathConcFolha ,
                                      link
                           from Presentation pres, Concept conc
                          where toConceptID      = @idconcFolha
                                and conc.ID          = pres.toConceptID
                                and pres.link        = iif(@linkConcFolha = 0,
                                                           (select min(pres2.link)
from presentation pres2 where pres2.toConceptID = @idconcfolha and pres2.schemaxsdid =
iif(@conceito = 1, @schemaXSDIDconca, @schemaXSDIDconcb)),--pres.link,

@linkConcFolha)
                                and pres.schemaXSDID = iif(@conceito = 1,
@schemaXSDIDconca, @schemaXSDIDconcb);
                      end;

                  -- Incrementa o contador
                  set @cont = @cont + 1;

                  -- Verifica quantas linhas tem
                  select @count = count(*) from @ttree;

              end;

              set @conceito = @conceito + 1;
            end;

        /* Computa a similaridade dos caminhos de cada conceito */

        -- Percorre cada par de conceito a X conceito b
        declare cTrees cursor for
        select ta.conceptPath , tb.conceptPath
          from @ttree as ta, @ttree as tb
         where ta.conceito = 'A'
```

```
            and tb.conceito = 'B';

         open cTrees;

         fetch cTrees into @conceptPatha, @conceptPathb;

      set @cont = 0;

         while (@@FETCH_STATUS = 0)
           begin
             set @cont = @cont + 1;

             insert into @tsim
               select dbo.PathSim  (@conceptPatha , @conceptPathb);

             fetch cTrees into @conceptPatha, @conceptPathb;

           end;

         /* Calcula a similaridade final */
         select @sim = sum(pathsim) / count(*)
           from @tsim
          where pathsim >= @threshold;

      RETURN isnull(@sim,0);

END;
```

```
CREATE FUNCTION TopDownSim (@schemaXSDIDconca bigint, @idconca bigint,
@schemaXSDIDconcb bigint, @idconcb bigint, @tpstruct char(01), @threshold
decimal(8,2))

RETURNS decimal(8,2)

AS

BEGIN

/*****************************************************************************/
/*                                                                           */
/* Função TopDownSim                                                         */
/* Script: TopDownSim.sql                                                    */
/* Descrição: Retorna a similaridade entre os relacionamentos calculation ou  */
/*            presentation de dois conceitos (a depender do parâmetro)       */
/*            utilizando o método TopDown-up                                 */
/*                                                                           */
/*****************************************************************************/

     /*** Declaracao de Variaveis ***/
        DECLARE @sim        decimal(8,2);

        DECLARE @cont           int;
        DECLARE @count          int;
        DECLARE @conceito       int;
        DECLARE @idconcRaiz     bigint;
        DECLARE @pathConcRaiz   nvarchar(max);
        DECLARE @linkConcRaiz   int;
        DECLARE @conceptPatha   nvarchar(max);
        DECLARE @conceptPathb   nvarchar(max);

        DECLARE @ttree          table (conceito char(01)    , cont    int         ,
idPai   bigint      , idFilho    bigint      , conceptPath nvarchar(max), link int);
        DECLARE @tsim           table (pathsim decimal(8,2));

     set @sim = 0;

        /* Monta as subárvores de cada conceito */

        set @conceito = 1;

        while @conceito < 3
          begin

               -- Inicializa o contador
               select @cont         = count(*)+1 from @ttree;
               select @count        = @cont;

               -- Insere o nó raiz
               insert into @ttree
                      select iif(@conceito=1,'A','B'), @cont, null, id, name, 0
                        from Concept
                       where id = iif(@conceito=1,@idconca,@idconcb);

               -- Percorre todos os filhos do nó raiz, montando as subárvores
               while @cont <= @count
                 begin
                      -- Recupera as informacoes da subarvore sendo trabalhada
                      select @idconcRaiz = idfilho, @pathConcRaiz = conceptPath,
@linkConcRaiz = link
```

```
                    from @ttree where cont = @cont;

                  -- Insere todos os filhos dessa subarvore (para serem subarvores
tambem)
                  If (@tpstruct = 'C')
                    begin
                         insert into @ttree
                         select distinct iif(@conceito = 1, 'A', 'B'),
                                @count + ROW_NUMBER() OVER(ORDER BY calc.id),
                                      calc.fromConceptID, calc.toConceptID,
                                      @pathConcRaiz + '/' + conc.name,
                                      link
                           from Calculation calc, Concept conc
                          where fromConceptID    = @idconcRaiz
                                and conc.ID          = calc.toConceptID
                                and calc.link        = iif(@linkConcRaiz = 0,
                                                           (select min(calc2.link)
from calculation calc2 where calc2.fromConceptID = @idconcRaiz and calc2.schemaXSDID =
iif(@conceito = 1, @schemaXSDIDconca, @schemaXSDIDconcb)), --calc.link,

@linkConcRaiz)
                                and calc.schemaXSDID = iif(@conceito = 1,
@schemaXSDIDconca, @schemaXSDIDconcb);
                      end
                    Else
                      begin
                         insert into @ttree
                         select distinct iif(@conceito = 1, 'A', 'B'),
                                @count + ROW_NUMBER() OVER(ORDER BY pres.id),
                                      fromConceptID     , toConceptID  ,
                                      @pathConcRaiz + '/' + conc.name  ,
                                      link
                           from Presentation pres, Concept conc
                          where fromConceptID    = @idconcRaiz
                                and conc.ID          = toConceptID
                                and pres.link        = iif(@linkConcRaiz = 0,
                                                           (select min(pres2.link)
from presentation pres2 where pres2.fromConceptID = @idconcRaiz and pres2.schemaXSDID
= iif(@conceito = 1, @schemaXSDIDconca, @schemaXSDIDconcb)), --pres.link,

@linkConcRaiz)
                                and pres.schemaXSDID = iif(@conceito = 1,
@schemaXSDIDconca, @schemaXSDIDconcb);
                      end;

                  -- Incrementa o contador
                  set @cont = @cont + 1;

                  -- Verifica quantas linhas tem
                  select @count = count(*) from @ttree;

              end;

              set @conceito = @conceito + 1;
            end;

        /* Computa a similaridade das subárvores cada conceito */

        -- Percorre cada par de conceito a X conceito b
        declare cTrees cursor for
        select ta.conceptPath , tb.conceptPath
          from @ttree as ta, @ttree as tb
```

```
          where ta.conceito = 'A'
            and tb.conceito = 'B';

        open cTrees;

        fetch cTrees into @conceptPatha, @conceptPathb;

    set @cont = 0;

        while (@@FETCH_STATUS = 0)
          begin
            set @cont = @cont + 1;

            insert into @tsim
              select dbo.PathSim  (@conceptPatha , @conceptPathb);

            fetch cTrees into @conceptPatha, @conceptPathb;

          end;

        /* Calcula a similaridade final */
        select @sim = sum(pathsim) / count(*)
          from @tsim
         where pathsim >= @threshold;

    RETURN isnull(@sim,0);

END;
```

```
CREATE FUNCTION PathSim (@p1 nvarchar(max), @p2 nvarchar(max))

RETURNS decimal(8,2)

AS

BEGIN
/*****************************************************************************/
/*                                                                           */
/* Function PathSim                                                          */
/* Script: PathSim.sql                                                       */
/* Descrição: Retorna a similaridade entre dois caminhos(path)               */
/*                                                                           */
/*****************************************************************************/

     /*** Declaracao de Variaveis ***/
        DECLARE @i          bigint;
        DECLARE @j          bigint;
        DECLARE @lins       int;
        DECLARE @cols       int;
        DECLARE @sim        decimal(8,2);
        DECLARE @a          decimal(8,2);
        DECLARE @b          decimal(8,2);
        DECLARE @c          decimal(8,2);

        DECLARE @lp1        nvarchar(max);
        DECLARE @lp2        nvarchar(max);

        DECLARE @tp1        table (token nvarchar(max));
        DECLARE @tp2        table (token nvarchar(max));
        DECLARE @msim       table (lin int, col int, token nvarchar(max), val
decimal(8,2));

     set @sim = 0;

        /*** Pre-processamento das strings ***/
        -- Retira os espaços e transforma as strings em maiusculas
        set @lp1 = REPLACE(UPPER(@p1),' ','');
        set @lp2 = REPLACE(UPPER(@p2),' ','');

        /*** Inicializa a matriz com os tokens ***/
        INSERT INTO @msim values (0, 0, 'i', 0);

        -- Carrega os tokens da primeira string
     set @i    = 1
        set @lins = 1

     WHILE charindex('/',@lp1,@i) <> 0
          BEGIN
               INSERT INTO @msim values (@lins, -1,
substring(@lp1,@i,charindex('/',@lp1,@i)-@i), 0);
               INSERT INTO @msim values (@lins, 0, 'i', @lins);

             set @i    = charindex('/',@lp1,@i) + 1;
               set @lins = @lins + 1;
          END;

        INSERT INTO @msim values (@lins, -1, substring(@lp1,@i,len(@lp1) - @i + 1), 0);
        INSERT INTO @msim values (@lins, 0 , 'i', @lins);

        -- Carrega os tokens da segunda string
     set @i    = 1
```

```
        set @cols = 1

     WHILE charindex('/',@lp2,@i) <> 0
          BEGIN
               INSERT INTO @msim values (-1    , @cols,
substring(@lp2,@i,charindex('/',@lp2,@i)-@i), 0);
               INSERT INTO @msim values (0     , @cols, 'i', @cols);

            set @i    = charindex('/',@lp2,@i) + 1;
               set @cols = @cols + 1;
          END;

        INSERT INTO @msim values (-1    , @cols, substring(@lp2,@i,len(@lp2) - @i + 1),
0);
        INSERT INTO @msim values (0     , @cols, 'i', @cols);

        /** Calculo da Matriz de Similaridade **/
        set @i = 1;
        set @j = 1;

        WHILE @i <= @lins
          BEGIN
            WHILE @j <= @cols
                BEGIN
                  -- Calcula a similaridade
                     select @a = val + 1 from @msim where lin = @i-1 and col = @j;
                     select @b = val + 1 from @msim where lin = @i   and col = @j-1;
                     select @lp1 = token from @msim where lin = @i   and col = -1;
                     select @lp2 = token from @msim where lin = -1   and col = @j;
                     select @c = val + (1 - dbo.QGramsSim(@lp1,@lp2)) from @msim where
lin = @i-1 and col = @j-1;

                     select @sim = min(val)
                       from (select @a as val union select @b as val union select @c
as val) x;

                     insert into @msim values (@i, @j, 'l', @sim);

                  set @j = @j + 1;
                END;
              set @i = @i + 1;
              set @j = 1;
          END;

    select @sim = (iif(@lins>@cols,@lins,@cols) - val) / iif(@lins>@cols,@lins,@cols)
         from @msim
        where lin = @lins and col = @cols;

     RETURN @sim;

END;
```

```
CREATE procedure NameTokenizeWord (@id bigint)

AS

BEGIN
/*****************************************************************************/
/*                                                                           */
/* Function NameTokenizeWord                                                 */
/* Script: NameTokenizeWord.sql                                              */
/* Descrição: Cria e insere na tabela NameTokenWord os tokens de um conceito  */
/*            usando maiúsculas como delimitador                             */
/*                                                                           */
/*****************************************************************************/

     /*** Declaracao de Variaveis ***/
        DECLARE @i           bigint;
        DECLARE @name        nvarchar(max);
        DECLARE @token       nvarchar(max);

        /*** Apaga os tokens existentes para o conceito ***/
        delete from NameTokenWord
         where conceptID = @id;

       /*** Recupera o nome do conceito para tokenizar ***/
       select @name = name from concept where id = @id;


        /*** Tokenizacao em Palavras ***/
        set @i = 1;
        set @token = '';

        while @i <= len(@name)
          begin
               If (@i > 1) and
               (((    Ascii(substring(@name,@i   ,1)) between 65 and 90)  and
                       (not(Ascii(substring(@name,@i-1,1)) between 65 and 90)  or
not(Ascii(substring(@name,@i+1,1)) between 65 and 90)) )
                    or
                       (substring(@name,@i,1) = '.'))
                  begin
                    insert into NameTokenWord
                       values (@id, @token);

              set @token = '';
                 end;

            set @token = CONCAT(@token,
iif(substring(@name,@i,1)='.','',substring(@name,@i,1)));
                set @i = @i + 1;
          end;

       /*** Insere o ultimo token ***/
       If rtrim(ltrim(@token)) <> ''
              insert into NameTokenWord
              values (@id, @token)


END;
```

```
CREATE procedure TokenizeConcepts (@idini bigint, @idfim bigint)

AS

BEGIN
/*********************************************************************************/
/*                                                                               */
/* Procedure TokenizeConcepts                                                    */
/* Script: TokenizeConcepts.sql                                                  */
/* Descrição: Procedimento que atualiza a tabela NameTokenWord de um conjunto */
/*            de conceitos                                                       */
/*                                                                               */
/*********************************************************************************/

     /*** Declaracao de Variaveis ***/
        DECLARE @id          bigint;

        declare concepts cursor for
        select id
          from concept
         where id >= @idini and id <= @idfim
           and abstract = 'F';

        open concepts;

        fetch concepts into @id;

        while @@FETCH_STATUS = 0
          begin
            exec NameTokenizeWord @id;

                fetch concepts into @id;
          end;

     close concepts;
        deallocate concepts;

END;
```

```
CREATE procedure SortedSimilarityEvaluation (@schemaXSDIDA bigint, @schemaXSDIDB
bigint, @threshold decimal(8,2), @window int)

AS

BEGIN
/*****************************************************************************/
/*                                                                           */
/* Procedure SortedSimilarityEvaluation                                      */
/* Script: SortedSimilarityEvaluation.sql                                    */
/* Descrição: Procedimento do processo de avaliação de similaridade entre    */
/*            conceitos representados pela XBRL, utilizando o conceito de    */
/*            sorted neighborhoods com duas chaves                           */
/*                                                                           */
/*****************************************************************************/

  /* Declaracao de variaveis */
  declare @id                           bigint;
  declare @idA                          bigint;
  declare @idB                          bigint;
  declare @name                         nvarchar(max);
  declare @nameA                        nvarchar(max);
  declare @nameB                        nvarchar(max);
  declare @schemaxsdid                  bigint;
  declare @nameQGramsSim                decimal(8,2);
  declare @nameTokenSim                 decimal(8,2);
  declare @labelSim                     decimal(8,2);
  declare @calcTopDownSim               decimal(8,2);
  declare @calcBottomUpSim              decimal(8,2);
  declare @presTopDownSim               decimal(8,2);
  declare @presBottomUpSim              decimal(8,2);
  declare @conttotal                    bigint;
  declare @contbloco                    bigint;
  declare @contpar                      bigint;
  declare @contsorted                   int;
  declare @contkey                      int;
  declare @fim                          int;
  declare @tconca                       table (id bigint, name nvarchar(max));
  declare @tconcb                       table (id bigint, name nvarchar(max));
  declare @dtinicio                     datetime;

  /* Inicio */
  select @dtinicio = getdate();
  PRINT concat('INICIO - ', convert(varchar(8), @dtinicio, 108));

  /* Limpa a tabela para incluir os resultados */
  delete from SortedSimEvalResult;
  print concat('APAGOU TABELA - ', convert(varchar(8), getdate() - @dtinicio, 108));

  select @conttotal = 0, @contbloco = 0, @contpar = 0, @contkey = 1;

  while @contkey < 3
    begin
         /* Inicializa contadores e tabela temporária dos conceitos */
         select @contsorted = 0, @fim = 0;
         delete from @tconca;
         delete from @tconcb;

         /* Percorre a lista de conceitos ordenada pela chave */
         If @contkey = 1
           begin
                declare sorted cursor for
```

```
                select concept.schemaxsdid, concept.id, concept.name
                     from concept
                    where (schemaxsdid = @schemaXSDIDA or schemaXSDID = @schemaXSDIDB)
                      and abstract = 'F'
                    order by key1name20;

                print concat('PRIMEIRA CHAVE - ', convert(varchar(8), getdate() -
@dtinicio, 108));
              end
         Else
           begin
                declare sorted cursor for
                 select concept.schemaxsdid, concept.id, concept.name
                      from concept
                    where (schemaxsdid = @schemaXSDIDA or schemaXSDID = @schemaXSDIDB)
                      and abstract = 'F'
                    order by key2nametokenmax;

                print concat('SEGUNDA CHAVE - ', convert(varchar(8), getdate() -
@dtinicio, 108));
              end;

         open sorted;

         fetch sorted into @schemaxsdID, @id, @name;

         If (@@FETCH_STATUS <> 0)
              select @fim = 1

         while (@contsorted <> -1)
              begin

                /* Divulga feedback */
                If @contbloco >= 1000
                     begin
                       PRINT concat('PERCORREU 1000 REGISTROS (', @conttotal,') - ',
convert(varchar(8), getdate() - @dtinicio, 108));
                       select @contbloco = 0;
                     end;

                /* Verifica se é para processar os pares da janela (se preencheu o
total ou acabaram os registros) */
                If (@contsorted = @window) or (@fim = 1)
                     begin

                       declare pairs cursor for
                        select conca.id, conca.name, concb.id, concb.name
                             from @tconca conca, @tconcb concb
                            where concb.id          > iif(@schemaxsdidA =
@schemaxsdidB, conca.id, 0)
                              and not exists (select null from SortedSimEvalResult
where ConceptAID = conca.id and ConceptBID = concb.id);

                       open pairs;

                       fetch pairs into @idA, @nameA, @idB, @nameB

                       while @@FETCH_STATUS = 0
                            begin

                              /* Verifica similaridade de nome entre os conceitos */
```

```
                                select @nameQGramsSim = dbo.QGramsSim(@nameA, @nameB),
@conttotal = @conttotal + 1, @contbloco = @contbloco + 1;

                                /* Se a similaridade entre os nomes for maior do que o
limite passado por parametro,
                                      continua o processamento, caso contrário, passa
para o próximo par */
                                If @nameQGramsSim >= @threshold
                                     begin
                                       select @contpar = @contpar + 1;

                                       PRINT concat('ENCONTROU PAR ', @contpar, '(',
@idA, ', ', @idB, ') - ', convert(varchar(8), getdate() - @dtinicio, 108));

                                       select @nameTokenSim = dbo.NameTokenSim(@idA,
@idB);;

                                       /* Verifica similaridade entre os labels */
                                       select @labelSim = dbo.LabelSim(@idA, @idB);

                                       /* Verifica similaridades da estruturas
calculation e presentation TopDown e BottomUp*/
                                       select @calcTopDownSim  = dbo.TopDownSim
(@schemaXSDIDA, @idA, @schemaXSDIDB, @idB,'C', 0.3);--@threshold);
                                       select @calcBottomUpSim =
dbo.BottomUpSim(@schemaXSDIDA, @idA, @schemaXSDIDB, @idB,'C', 0.3);--@threshold);
                                       select @presTopDownSim  = dbo.TopDownSim
(@schemaXSDIDA, @idA, @schemaXSDIDB, @idB,'P', 0.3);--@threshold);
                                       select @presBottomUpSim =
dbo.BottomUpSim(@schemaXSDIDA, @idA, @schemaXSDIDB, @idB,'P', 0.3);--@threshold);

                                       /* Insere o par na tabela de resultados */
                                       insert into SortedSimEvalResult
                                       select @idA,    @nameA, @idB, @nameB,
                                                   @nameQGramsSim , @nameTokenSim   ,
@labelSim,
                                                   @calcTopDownSim, @calcBottomUpSim,
                                                   @presTopDownSim, @presBottomUpSim

--                                     PRINT concat('PROCESSOU PAR ', @contpar, '(',
@idA, ', ', @idB, ') - ', convert(varchar(8), getdate() - @dtinicio, 108));
                                    end;

                                  fetch pairs into @idA, @nameA, @idB, @nameB;
                               end;

                         close pairs;

                         deallocate pairs;

                         delete from @tconca;
                         delete from @tconcb;
                         select @contsorted = IIF(@fim = 1, -1, 0);

                      end;

                If @fim = 0
                     begin
                       /* Insere os conceitos nas tabelas temporarias */
                       If (@schemaxsdid = @schemaXSDIDA) insert into @tconca values
(@id, @name);
```

```
                        If (@schemaxsdid = @schemaXSDIDB) insert into @tconcb values
(@id, @name);
                    end;

                /* Incrementa o contador */
                select @contsorted = IIF(@fim = 1, -1, @contsorted + 1);

                fetch sorted into @schemaxsdID, @id, @name;

                If (@@FETCH_STATUS <> 0)
                    select @fim = 1

            end;

        close sorted;

        deallocate sorted;

        select @contkey = @contkey + 1;
  end;
  PRINT concat('FIM - ', convert(varchar(8), getdate() - @dtinicio, 108));

END;
```