5
Trabalhos relacionados

Diversas técnicas para alinhamento de esquemas têm sido propostas para automatizar a operação de alinhamento. Rahm & Bernstein (2001) apresentam um levantamento de diversas técnicas de alinhamento de esquemas conceituais e propõem uma taxonomia para classificá-las.

Em um levantamento mais recente, baseado no anterior, Euzenat & Shvaiko (2007) apresentam duas classificações para técnicas de alinhamento de esquemas e ontologias. A Figura 29 ilustra as duas classificações, apresentadas em forma de árvores compartilhando suas folhas na parte central da figura. As folhas representam as classes elementares de técnicas de alinhamento, incluindo exemplos concretos. As classificações são interpretadas segundo:

- Granularidade / Interpretação da entrada: o primeiro nível desta classificação baseia-se na granularidade das operações de alinhamento utilizada pelas técnicas (em nível de elemento ou de estrutura), ou seja, como as técnicas de alinhamento interpretam as informações de entrada. O segundo nível desta classificação baseia-se em como as técnicas interpretam as informações de entrada: sintaticamente, com auxílio de informações externas ou semanticamente.
- Tipo de entrada: classificação baseada nos tipos dos dados de entrada utilizados pelas técnicas de alinhamento: cadeias de caracteres (terminológico), estrutura (estrutural), modelos (semântico) ou instâncias.

Seguindo a classificação por tipo de entrada, as técnicas apresentadas nesta tese são classificadas como baseadas em instâncias (*Extensional*) que utilizam estatística e análise de dados (*Data analysis and statistics*) para gerar os mapeamentos. Os autores incluem nesta categoria as técnicas que utilizam uma porção representativa de dados de exemplo para encontrar regularidades e discrepâncias.

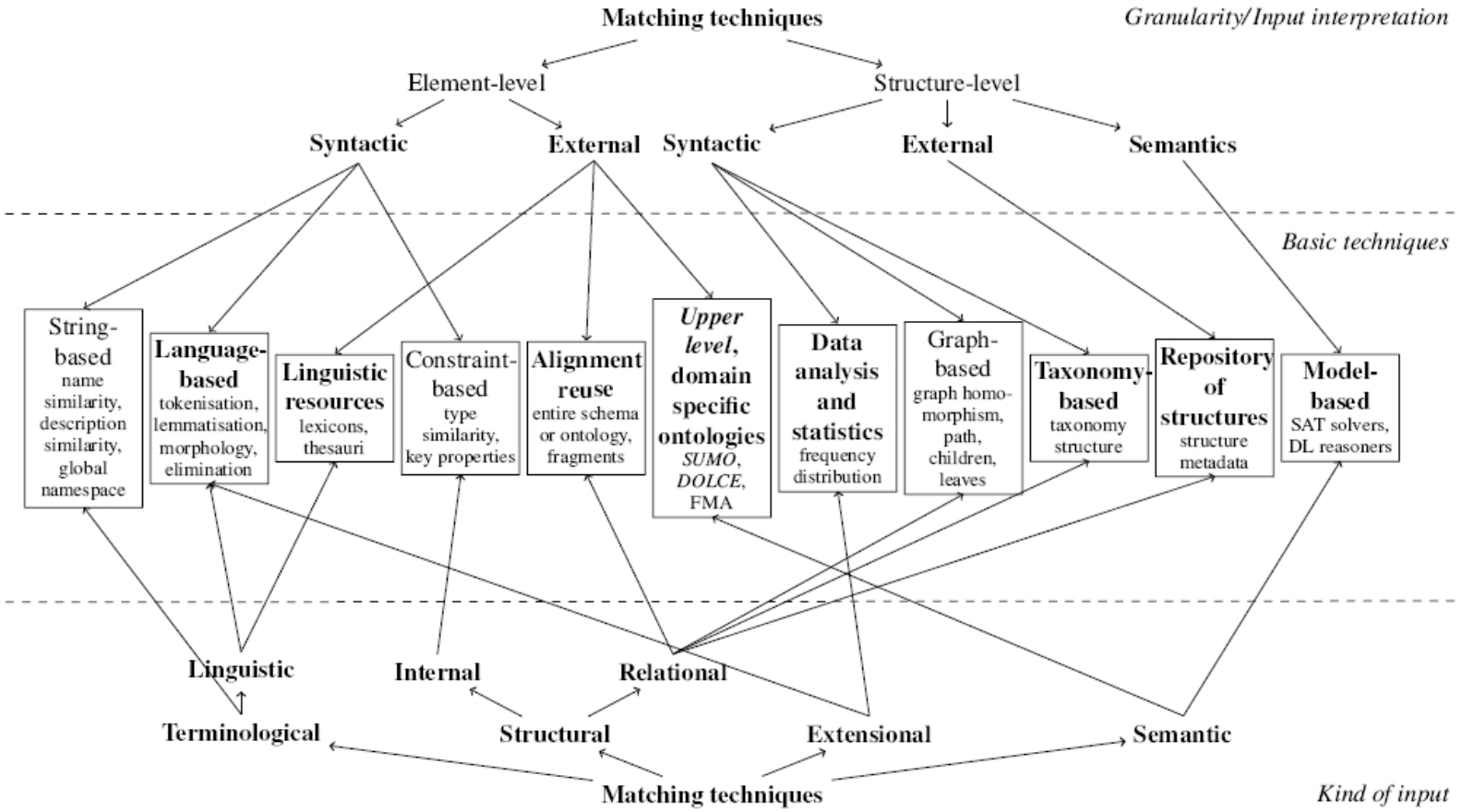

Figura 29 – Classificação das técnicas de alinhamento (Euzenat & Shvaiko, 2007).

Bernstein & Melnik (2007) apresentam os requisitos para um Sistema de Gerenciamento de Modelos (SGM). Um SGM é um componente reutilizável que oferece suporte a criação, compilação, reutilização, evolução e execução de mapeamentos entre esquemas. Um SGM pode ser embutido em ferramentas para carga de dados e mediação de consultas, por exemplo. Alguns dos requisitos de um SGM são:

- Oferecer suporte a esquemas representados em quaisquer modelos de dados populares, tais como: SQL, XML Schema, Entidade-Relacionamento (ER), RDF e OWL;
- Oferecer suporte a uma linguagem de mapeamentos rica, que possa ser aplicada aos diferentes tipos presentes nos modelos de dados suportados;
- Oferecer um módulo que execute os alinhamentos em tempo de execução do SGM e ofereça serviços sobre os mapeamentos descobertos, tais como: consultas, propagação de atualizações, notificações, exceções, sincronização e proveniência.

Bernstein & Melnik (2007) definem o processo de geração de mapeamentos de um SGM como um processo de três passos que produz os mapeamentos em três representações sucessivamente refinadas: *correspondências*, *restrições de mapeamentos* e *transformações*. O primeiro passo é gerar as correspondências. *Correspondências* são pares de elementos de dois esquemas que acredita-se estarem relacionados de alguma forma. Normalmente, as correspondências dão pistas sobre quais os elementos de dois esquemas precisam estar relacionados por um mapeamento. O problema de gerar correspondências é chamado pelos autores de *alinhamento de esquemas*. O segundo passo é traduzir as correspondências em restrições de mapeamentos. Cada *restrição de mapeamento* especifica uma pequena porção do mapeamento desejado de forma a auxiliar o entendimento do arquiteto de dados. O terceiro passo é traduzir estas restrições de mapeamentos em *transformações* executáveis. Consultas ou visões são exemplos destas transformações.

Wang et al. (2004) propõe uma técnica para alinhamento de esquemas aplicada a bancos de dados na Web baseada na técnica de sondagem de consultas. Um banco de dados na Web é um banco de dados disponível através da Web e acessível para consultas via formulários em um Website. Em particular, um banco de dados na Web possui dois esquemas diferentes: o

esquema de interface e o esquema de resultados. O *esquema de interface* consiste nos atributos sobre os quais os usuários podem criar suas consultas, ou seja, os campos dos formulários. O *esquema de resultados* consiste nos atributos que organizam os resultados da consulta que são apresentados ao usuário. Muitas vezes o esquema de resultado é implícito e só pode ser definido por suposição com base na forma de apresentação dos resultados.

A abordagem baseia-se em algumas observações:

(1) Consultas inadequadas freqüentemente geram erros de busca, isto é, não retornam nada. Para os autores, uma inadequação é caracterizada quando uma palavra-chave submetida como consulta para um elemento do esquema de interface em particular não corresponde a um valor aplicável ao atributo do banco de dados que o elemento está associado. Por exemplo, se uma cadeia de caracteres é submetida como consulta a um atributo originalmente definido como inteiro, será retornado um erro. Ou seja, se um *título* é submetido como consulta para o elemento do esquema de interface *ISBN.*

(2) As palavras-chave de uma consulta apropriada são candidatas a co-ocorrerem na página de resultados da consulta.

(3) Existência de um esquema global para bancos de dados na Web de um mesmo domínio. O esquema global contém atributos representativos dos objetos de um domínio específico.

(4) Existência de uma coleção de instâncias de exemplo (instâncias globais) classificadas segundo o esquema global.

A técnica de sondagem de consultas consiste em exaustivamente submeter consultas por palavra-chave para a interface de consulta dos bancos de dados na Web que desejam ser integrados, e coletar estes resultados para posterior análise. Estas consultas são submetidas utilizando os valores dos atributos das instâncias globais (4).

A análise dos resultados é baseada na observação (2). Dada uma consulta adequada, o conjunto de resultados provavelmente conterá uma co-ocorrência do valor submetido na consulta. Os resultados serão coletados das páginas HTML retornadas. Então, a co-ocorrência das *keywords* nos resultados podem ser utilizadas como um indicador de qual consulta submetida foi apropriada (i.e., para descobrir os elementos associados no esquema de interface). Além disso, a posição dos valores nas páginas de resultados podem ser utilizados para identificar os atributos do esquema de resultado.

Madhavan et al. (2005) propõem o uso de um conjunto de exemplos (*corpus*) de esquemas e mapeamentos para auxiliar os algoritmos de alinhamento de esquemas. Os autores utilizam um *corpus* como uma base de conhecimento de representações alternativas de conceitos de um domínio em particular. Por exemplo, dados os esquemas A, B, C e D de um mesmo domínio, e dado um corpus contendo os mapeamentos de A com B e de C com D, considere que se deseja alinhar A com C. Ao utilizar uma técnica de alinhamento baseada em esquemas, por exemplo, via comparação de nomes de atributos, a tarefa de alinhamento pode ser facilitada, pois os nomes dos atributos de B que mapeiam com os atributos de A podem ser usados como nomes alternativos para os elementos de A. Já utilizando uma técnica de alinhamento baseada em instâncias, podemos considerar as instâncias de B como instâncias de A, e as instâncias de D como instâncias de C como entrada do algoritmo de alinhamento para descobrir os alinhamentos entre A e B. Outro método para explorar o *corpus* é estimar estatísticas a respeito dos esquemas e seus elementos para aprender restrições de domínio a respeito dos esquemas.

Para encontrar similaridades entre um elemento do esquema e um elemento do corpus, os autores utilizam diversos *preditores*, algoritmos de aprendizado de máquina que geram predições quanto a similaridade dos elementos. O índice de similaridade final é computado a partir da combinação dos resultados dos preditores. Entre os preditores citados, estão os classificadores de texto para nomes e descrições dos elementos do esquema, para os valores dos atributos das instâncias e para informações de contexto. Para os autores, as informações de contexto de um dado elemento são os outros elementos que estão relacionados a ele. Por exemplo, para uma coluna de uma tabela, suas informações de contexto são: a tabela e as demais colunas da tabela.

Em (Madhavan et al., 2007) é apresentada uma arquitetura para integração de dados na Web: PayGo. A PayGo utiliza o conceito de *dataspaces* e o gerenciamento de dados seguindo o princípio *pay-as-you-go*. *Dataspaces* são coleções de dados heterogêneos, modelados como um conjunto de participantes e seus relacionamentos. Os participantes de um *dataspace* são as fontes de dados, que podem ser relacionais, XML, etc. As fontes de dados participantes podem ser estruturadas, semi-estruturadas ou sem estrutura. Em um *dataspace* deve ser possível modelar relacionamentos entre dois ou mais participantes, por exemplo quando um participante é uma visão ou réplica de outro, ou quando é preciso definir mapeamentos entre os esquemas de dois

participantes. O princípio *pay-as-you-go* define que um sistema de gerenciamento de *dataspaces* precisa ser capaz de evoluir seu *entendimento* sobre os dados que manipula de forma incremental e contínua, ou seja, o sistema precisa constantemente evoluir o seu conhecimento sobre a estrutura e semântica dos dados, e sobre os relacionamentos entre as fontes.

Diferente de sistemas de integração de dados tradicionais, o PayGo não usa um único esquema mediado e sim um repositório de esquemas. Os esquemas no repositório são agrupados em tópicos. Um esquema pode participar de mais de um tópico. Os alinhamentos são realizados entre quaisquer pares de esquemas utilizando a técnica descrita em (Madhavan et al., 2005).

Bilke & Naumann (2005) descrevem o algoritmo DUMAS, que usa duplicatas descobertas em coleções de dados de fontes distintas para alinhar os esquemas destas fontes. O DUMAS (*Duplicate-based Matching of Schemas*) percorre coleções de dados de fontes distintas em busca de tuplas similares, chamadas duplicatas. A partir das duplicatas, são derivadas correspondências entre os atributos dos esquemas com base nos valores similares dos atributos que ocorrem nas duplicatas, ou seja, valores de dados similares que ocorrem nas duplicatas indicam correspondência entre os atributos dos esquemas das fontes a serem alinhadas. Para detectar as duplicatas, o DUMAS considera cada tupla como uma única cadeia de caracteres, aplica um algoritmo de comparação para gerar um índice de similaridade para pares de tuplas. O alinhamento dos esquemas é feito a partir da média das matrizes de similaridades geradas para os pares de duplicatas descobertos.

Nottelmann & Straccia (2005, 2007) apresentam o sPLMap (*Probabilistic, Logic-based Mapping*), um framework para mapeamento de esquemas baseado em Datalog probabilístico (Fuhr, 2000). O sPLMap combina diferentes classificadores para aprender os mapeamentos entre esquemas heterogêneos, tais como: algoritmos de comparações sintáticas, Naive Bayes e kNN (k-Nearest Neighbor). O framework foi também aplicado para taxonomias de diretórios Web: oPLMap (Nottelmann & Straccia, 2006).

Udrea et al. (2007) apresentam o algoritmo ILIADS (*Integrated Learning In Alignment of Data and Schema*) para alinhamento de ontologias em OWL Lite. A abordagem combina um algoritmo de clusterização por similaridade com um algoritmo de inferência lógica incremental. Para calcular a similaridade são utilizadas informações léxicas, estruturais e de instâncias. Ao clusterizar as entidades da ontologia (classes, propriedades e instâncias), novos relacionamentos (*subsumption* e equivalência) são criados. Estes novos

relacionamentos podem gerar consequências lógicas quando analisados em conjunto com os axiomas originais da ontologia. Por exemplo, ao analisar as ontologias A e B, se o algoritmo adiciona o axioma (A:E-Coli-Poisoning, owl:sameAs, B:E-Coli) quando ele for avaliado em conjunto com os axiomas existentes (A:discoveredBy, owl:inverseOf, B:discoverer), (A:discoveredBy, owl:Type, owl:FunctionalProperty), (A:E-Coli-Poisoning, A:discoveredBy, A:TheodorEscherich) e (B:T.S.Esherich, B:discover, B:E-Coli) haverá a seguinte implicação lógica: (A:TheodorEscherich, owl:sameAs, B:T.S. Esherich).

A Tabela 21 mostra uma comparação entre as abordagens de alinhamento apresentadas nos trabalhos relacionados discutidos acima e as abordagens apresentadas nesta tese, evidenciando os pontos que cada uma das abordagens contempla. A comparação é feita utilizando três características das abordagens. A primeira é quanto ao objeto de alinhamento contemplado pela abordagem, isto é, se a abordagem de alinhamento pode ser aplicada a tesauros, ontologias ou esquemas conceituais. A comparação inclui também os recursos básicos utilizados, ou seja, os itens utilizados como entrada dos algoritmos de alinhamento: elementos dos esquemas ou instâncias. A terceira característica é quanto à forma de descoberta dos alinhamentos: *a priori* ou adaptativa.

Tabela 21 – Tabela comparativa dos trabalhos relacionados.

Técnicas	Objeto de alinhamento			Recursos básicos utilizados (input)		Forma de descoberta dos alinhamentos	
	Tesauros	Ontologias	Esquemas	Elementos	Instâncias	*A priori*	Adaptativa
Bilke & Naumann, 2005	✘	✘	✓	✘	✓	✓	✘
Madhavan et al., 2005	✘	✘	✓	✓	✓	✓	✘
Madhavan et al., 2007	✘	✘	✓	✓	✓	✘	✓
Nottelmann & Straccia, 2005 e 2007	✘	✘	✓	✓	✓	✓	✘
Nottelmann & Straccia, 2006	✘	✓	✘	✓	?	✓	✘
Udrea et al., 2007	✘	✓	✘	✓	✓	✓	✘
Wang et al., 2004	✘	✘	✓	✘	✓	✓	✘
Brauner 2008 (esta tese)	✓	✓	✓	✘	✓	✓	✓

✓ = Contempla; ✘ = Não contempla; ? = Em caráter duvidoso, não esclarecido na bibliografia disponível até a conclusão desta tese.