2
Meta-alinhamento de ontologias

Embora várias técnicas para alinhar ontologias venham sendo propostas na literatura, não há uma técnica que seja completamente eficaz em todos os casos. Isso se dá porque, para encontrar boas técnicas, é necessário ter conhecimento do contexto em que as técnicas serão aplicadas, dos dados disponíveis e das diferenças presentes nos modelos a serem analisados. Assim, é necessário reconsiderar as técnicas escolhidas a cada caso. Contudo, a utilização conjunta e coordenada de técnicas distintas, preferencialmente complementares, pode ajudar na obtenção de melhores alinhamentos entre ontologias. Neste contexto, são utilizados meta-alinhadores de ontologias.

Este capítulo discute sobre o problema de meta-alinhamento de ontologias. As definições que utilizamos são apresentadas na seção 2.1.

2.1
Definições

Linguagens para descrição de ontologias permitem a especificação de diferentes tipos de entidades[1], como classes, indivíduos, relações, tipos de dados e valores. Classes (ou conceitos) são as entidades principais de uma ontologia, as quais são interpretadas como um conjunto de indivíduos do domínio. Indivíduos (ou objetos, ou instâncias) são interpretados como objetos específicos de um domínio, instâncias de uma classe. Indivíduos, bem como classes, podem se relacionar com outros indivíduos ou classes através de diversos tipos de relações.

Tipos de dados especificam conjuntos de valores, ao contrário de indivíduos, contudo, valores não possuem identidades. Valores são instâncias de tipos primitivos. Por exemplo, "Pet Sounds" é um valor que pode ser o título de um *álbum musical*.

Classes não necessitam ser nomeadas e podem participar de construções lógicas como especialização, exclusão, instanciação e atribuição.

[1] O termo entidade, neste contexto, possui um significado mais amplo do que quando utilizado na área de Banco de Dados em modelos como Entidade-Relacionamento.

A especialização entre duas classes ou duas propriedades é interpretada como a inclusão das interpretações dessas entidades. Por exemplo, a classe *Carro* é uma especialização da classe *Veículo*, o que significa dizer que a noção de *Carro* está incluída na noção de *Veículo*, isto é, $Carro \subset Veículo$. Vale ressaltar que a relação de inclusão traduz, na linguagem da teoria dos conjuntos, a operação lógica de implicação de proposições. De fato, dizer que $A \subset B$ é o mesmo que dizer que a proposição "$x \in A$" implica a proposição "$x \in B$". Tem-se então $A \subset B$ *sse* $\forall x \in \mathbb{U}, x \in A \Rightarrow x \in B$.

A exclusão entre duas classes ou propriedades é interpretada como a exclusão das interpretações dessas entidades, isto é, quando sua interseção é vazia. Por exemplo, a classe *Livro* pode ser declarada como exclusiva para a classe *Pessoa*. Assim, a relação de exclusão entre A e B implica que A é exclusivo para B *sse* $\forall x, y \in \mathbb{U}, (x \in A \Rightarrow x \notin B) \wedge (y \in B \Rightarrow y \notin A)$.

A instanciação ou tipagem entre indivíduos e classes, instâncias de propriedades e propriedades, valores e tipos de dados é interpretada como associação. Por exemplo, o indivíduo *Rio de Janeiro* é uma instância da classe *Cidade*.

Por fim, a atribuição é uma relação entre instâncias de propriedades, indivíduos e valores, a qual associa um valor a uma propriedade de um dado indivíduo. Por exemplo, é atribuído o valor 1565 à propriedade *Data de Fundação* do indivíduo *Rio de Janeiro*. Em resumo, podemos considerar uma ontologia como na definição 2.1 (Euzenat & Shvaiko, 2007):

Definição 2.1 (Ontologia) *Seja O o conjunto de ontologias, onde $o \in O$ é um tupla $o = \langle C, I, R, T, V, \leq, \perp, \in, = \rangle$, tal que:*

C é o conjunto de classes;
I é o conjunto de indivíduos;
R é o conjunto de relações;
T é o conjunto de tipos de dados;
V é o conjunto de valores;
$\leq$ é uma relação em $(C \times C) \cup (R \times R) \cup (T \times T)$ chamada especialização;
$\perp$ é uma relação em $(C \times C) \cup (R \times R) \cup (T \times T)$ chamada exclusão;
$\in$ é uma relação sobre $(I \times C) \cup (V \times T)$ chamada instanciação;
$=$ é uma relação sobre $I \times R \times (I \cup V)$ chamada atribuição.

O objetivo das técnicas de alinhamento de ontologias é encontrar correspondências entre entidades definidas em diferentes ontologias. Quase sempre, estas correspondências são relações de equilavência, que são descobertas através de medidas de similaridade entre as entidades das ontologias. O modo

mais comum de avaliar a similaridade entre duas entidades é definindo uma medida de similaridade, conforme a definição 2.2.

Definição 2.2 (Similaridade) *Seja κ o conjunto de todos os pares (o_i, o_j) definidos em $O \times O$, uma similaridade $\sigma : \kappa \to \mathbb{R}$ é uma função que avalia um par de entidades $x \in o_i$ e $y \in o_j$ e mapeia para um número real que expressa o grau de similaridade entre dois objetos tal que:*

$$\begin{aligned} \forall x, y\ \sigma(x, y) \geq 0 \quad & \textit{(positividade)} \\ \forall z \in o_j, \sigma(x, x) \geq \sigma(y, z) \quad & \textit{(maximalidade)} \\ \forall x, y\ \sigma(x, y) = \sigma(y, x) \quad & \textit{(simetria)} \end{aligned}$$

Uma vez que uma função de similiaridade é aplicada sobre entidades de ontologias, podem-se definir funções de similaridade específicas para combinação de tipos de entidades, por exemplo, classes e classes, relações e relações, classes e relações etc. Como cada função de similaridade é aplicada sobre algumas entidades e, ainda, cada função de similaridade utiliza técnicas diferentes de análise, a escolha da função de similaridade mais adequada para avaliar duas ontologias torna-se dependente da estrutura das ontologias dadas. Contudo, a análise de duas ontologias pode ser feita, de forma composta, por várias funções de similaridade, gerando uma função composta, conforme a definição 2.3.

Definição 2.3 (Similaridade composta) *Uma similaridade composta $\delta : o \times o \to \mathbb{R}$ é uma função de similaridade dada pela soma de similaridades $\sigma_1 + \sigma_2 + ... + \sigma_n$, onde σ_i indica uma função de similaridade sobre entidades de o e $i \in [1..n]$.*

Definição 2.4 (Similaridade normalizada) *Uma similaridade composta é dita normalizada se seu valor varia no intervalo de números reais [0,1], onde o valor 1 representa uma similaridade perfeita, o valor 0 representa nenhuma similaridade e valores intermediários representam a probabilidade das entidades serem similares. Uma versão normalizada de uma similaridade composta δ é denotada como $\bar{\delta}$.*

Novas funções de similaridade compostas podem ser definidas utilizando outras funções de similaridade, compostas ou não. Assim, podemos definir uma função de similaridade composta como, por exemplo, $\delta''(a, b) = \sigma(a, b) + \delta'(a, b)$.

Para normalizar funções de similaridade, pode-se utilizar pesos associados a cada função membro da composição. Cada peso representa a relevância da função de similaridade para o cálculo da similaridade composta.

Definição 2.5 (Composição com pesos) *Sejam o_1 e o_2 duas ontologias e $F = g_1 \circ g_2 \circ ... \circ g_n$ uma função de similaridade composta por n funções membros. Dadas duas entidades a e b onde $a \in o_1, b \in o_2$, seja $g_i(a,b)$ a i-ésima função de similaridade membro de F(a,b), k um valor inteiro não negativo e os pesos p_i, a função normalizada $\bar{F}(a,b)$ pode ser definida como*

$$\bar{F}(a,b) = \sum_{i=1}^{n} g_i(a,b)p_i, onde \sum_{i=1}^{n} p_i \leq k, \tag{2-1}$$

sendo $p_i \in \mathbb{R}$.

A partir do retorno de uma ou mais funções membros de similaridade, alinhadores podem decidir qual a correspondência devida entre pares de entidades e gerar o alinhamento entre as ontologias. Pode-se definir alinhamentos conforme a definição 2.6 (adaptada de Shvaiko & Euzenat, 2005).

Definição 2.6 (Alinhamento) *Dadas duas ontologias o e o', alinhamento é um conjunto de correspondências entre pares de entidades $\langle e, e' \rangle$ pertencentes a o e o', respectivamente. Uma correspondência é descrita como uma quádrupla $\langle e, e', r, n \rangle$ onde:*

- *e e e' são as entidades (por exemplo, termos, classes, indivíduos) sobre as quais uma relação é afirmada pela correspondência.*
- *r é a relação, entre e e e', afirmada pela correspondência, podendo ser uma relação da teoria dos conjuntos, uma relação fuzzy, uma medida de similaridade, etc.*
- *n é o grau de confiança na correspondência, onde $n \in [0,1]$. Deve-se observar que este grau não se refere à relação r, mas à medida de confiança no fato de que a correspondência é verdadeira.*

O processo de alinhamento de ontologias realizado por um alinhador pode ser descrito como uma função F_a onde, dado um par de ontologias o e o', um alinhamento pré-existente A, um conjunto de parâmetros p e um conjunto de recursos r, um alinhamento A' é retornado (Bouquet et al, 2004), conforme a equação 2-2.

$$A' = F_a(o, o', A, p, r) \tag{2-2}$$

Ressalta-se que os conjuntos A, p e r podem ser vazios. Dessa forma, um alinhador pode utilizar um processo que gera alinhamentos sem a necessidade de alinhamentos pré-existentes, ou sem a possibilidade de configuração do alinhador ou, ainda, sem a utilização de outros recursos, como base de dados

externas, outras ontologias, documentos, feedback do usuário (Souza et al, 2010a), etc.

Um alinhador pode ser avaliado quanto à qualidade dos alinhamentos gerados. Verificar a qualidade de alinhamentos significa verificar se o alinhamento possui correspondências corretas. Esse processo de avaliação pode ser realizado por um usuário especialista no domínio. É possível definir medidas de avaliação de alinhadores, caso exista um alinhamento de referência conhecido. Essas medidas podem avaliar alinhamentos através da comparação entre as correspondências obtidas pelo alinhador e as correspondências esperadas.

Definição 2.7 (Avaliação de alinhamentos) *Uma avaliação de alinhamentos ε é uma função $\varepsilon : A \rightarrow [0, 1]$ que associa um alinhamento A a um número real que representa a qualidade do alinhamento A em relação ao alinhamento correto entre o e o'.*

Segundo Euzenat & Shvaiko (2007), meta-alinhadores são sistemas nos quais a originalidade está no modo em que estes sistemas utilizam e combinam outros alinhadores. Meta-alinhadores utilizam variadas técnicas para encontrar um bom alinhamento entre ontologias à partir do resultado de outros alinhadores. Para o contexto desse trabalho, define-se uma função de meta-alinhamento conforme a definição 2.8 e um meta-alinhador conforme a definição 2.9.

Definição 2.8 (Função de meta-alinhamento) *Seja $S \subset \Re$ um conjunto de valores que indicam todos os graus de similaridade calculados obtidos a partir da aplicação de uma função de similaridade, uma função de meta-alinhamento h é uma função $h : S \mapsto \Re$ que define a relevância de cada grau de similaridade previamente calculado $s \in S$. O resultado obtido é um grau de similaridade otimizado $s_o \in \Re$. Chamamos de grau de similaridade otimizado o melhor valor possível encontrado, ou seja, o grau de similaridade mais próximo de 1.*

Definição 2.9 (Meta-alinhador) *Um meta-alinhador é uma função $\Psi : o \times o' \stackrel{h}{\rightarrow} A$ que associa duas ontologias de entrada o e o' a um alinhamento A fazendo uso de uma função de meta-alinhamento h, de tal forma que $\varepsilon(\Psi(o, o'))$ seja o mais próximo possível de 1.*

Na prática, meta-alinhamento é a técnica de selecionar os algoritmos, correspondências e pesos mais apropriados em diferentes cenários de alinhamento com o objetivo de obter um alinhamento satisfatório entre ontologias (Martinez-Gil & Aldana-Montes, 2012).

2.2 Algoritmos genéticos

Algoritmos genéticos são métodos de solução de problemas computacionais inspirados na teoria da evolução das espécies de Darwin e na genética (Goldberg, 1989). Em algoritmos genéticos, uma população de possíveis soluções para o problema em questão evolui de acordo com operadores probabilísticos concebidos a partir de metáforas biológicas, de modo que há uma tendência de que, na média, os indivíduos representem soluções cada vez melhores à medida que o processo evolutivo continua (Tanomaru, 1995).

Algoritmos genéticos são particularmente aplicados em problemas complexos de otimização (Langdon & Poli, 2002): problemas com diversos parâmetros ou características que precisam ser combinadas em busca da melhor solução; problemas com muitas restrições ou condições que não podem ser representadas matematicamente; e problemas com grandes espaços de busca (Pacheco, 1999). Algoritmos genéticos têm sido aplicados a diversos problemas de otimização (Goldberg & Sastry, 2011), tais como otimização de funções matemáticas, otimização combinatoria, otimização de planejamento, otimização de distribuição etc.

Um algoritmo genético é composto de alguns passos básicos, apresentados no algoritmo 1.

Algoritmo 1: Passos básicos de um algoritmo genético (adaptado de Talbi, 2009, pp. 200)

Saída: Melhor indivíduo ou melhor população encontrada
início
  Gerar($P(0)$); /* População inicial */
  $t = 0$;
  enquanto *não Critério_Parada($P(t)$)* **faça**
    Avaliar($P(t)$);
    $P'(t) \leftarrow$ Selecao($P(t)$);
    $P'(t) \leftarrow$ Reproducao($P'(t)$); Avaliar($P'(t)$);
    $P(t+1) \leftarrow$ Substituir($P(t)$, $P'(t)$);
    $t \leftarrow t + 1$;
  fim enquanto
fim

Uma população em algoritmo genético pode ser representada por um conjunto S de cromossomos onde um dado cromossomo C dessa população contém necessariamente a informação da solução que ele representa (Haupt et al, 2004). Um cromossomo é composto por um conjunto de genes e cada gene G representa individualmente uma parte da solução do problema. Sendo assim,

tem-se um cromossomo $C = \{G_1, G_2, G_3, \ldots, G_n\}$ contendo n genes e o conjunto $S = \{C_1, C_2, C_3, \ldots, C_m\}$ de uma população contendo m cromossomos, onde $|S|$ é o tamanho da população (Bonnans, 2006).

A tabela 2.1 apresenta um paralelo entre a metáfora do processo de evolução e a resolução de um problema de otimização.

Tabela 2.1: Processo de evolução versus resolução em um problema de otimização, adaptado de (Talbi, 2009)

Metáfora	**Otimização**
Evolução	Resolução do problema
Indivíduo ou Cromossomo	Solução
População	Conjunto de soluções candidatas
Aptidão	Função objetivo
Ambiente	Problema de otimização
Gene	Elemento da solução
Alelo	Valor do elemento (gene)

Uma geração de cromossomos é uma população que pode ter sido gerada inicialmente de forma aleatória ou a partir de construtores da solução, ou pode ter sido originada através da seleção e cruzamento entre indíviduos. A quantidade de gerações pode ser definida como um critério de parada para a execução do algoritmo.

Em algoritmos genéticos, tem-se três operadores que guiam o algoritmo no processo de obtenção da solução do problema para um comportamento que simula a evolução das espécies: operadores de seleção, reprodução e mutação.

A seleção dos cromossomos é feita por um operador apropriado, o qual leva em conta a aptidão do indivíduo (cromossomo). A aptidão está diretamente relacionada com a qualidade da solução que este indivíduo codifica. A aptidão do indivíduo é avaliada por uma *função de aptidão* ou *função objetivo*. A função objetivo é a função que se pretende maximizar ou minimizar. Assim, para uma maximização, o objetivo do algoritmo genético é encontrar uma solução que retornará o maior valor possível para a função objetivo.

O processo de reprodução ou cruzamento consiste em combinar cromossomos na tentativa de gerar uma nova população em que os filhos dos cromossomos cruzados possuam os melhores genes de seus pais, fazendo que a cada cruzamento efetuado a solução encontrada se aproxime da desejada. Alguns métodos de cruzamento (*crossover*) utilizados são o ponto de cruzamento único, dois pontos de cruzamento, cruzamento uniforme, entre outros (Haupt et al, 2004). A figura 2.1 exemplifica os três métodos de cruzamento citados.

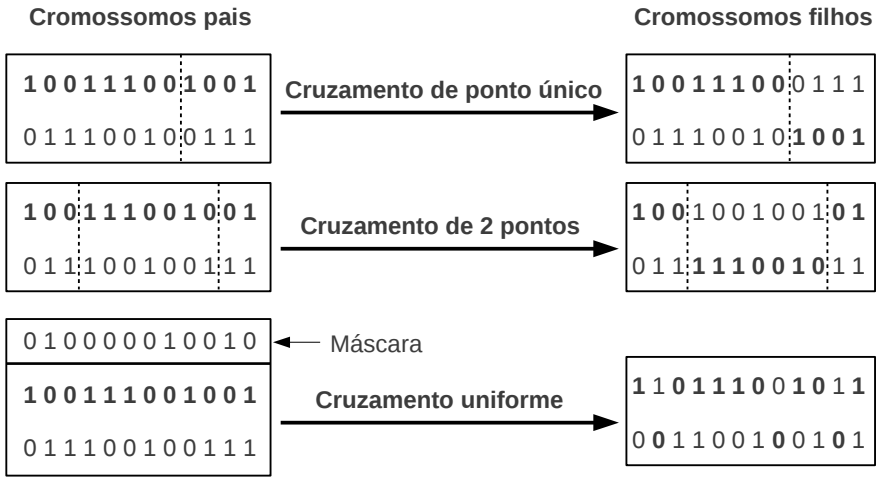

Figura 2.1: Método de ponto de cruzamento único, cruzamento de dois pontos e cruzamento uniforme

No método de cruzamento de ponto único, é determinado um ponto p de cruzamento e o indivíduo $I_{A,B}$ gerado a partir dos cromossomos A e B, de tamanho n, herda os genes de A posicionados entre 1 e p e os genes de B posicionados entre $p+1$ e n. No método de cruzamento de dois pontos, dois pontos p e q são determinados. Assim, o indivíduo $I_{A,B}$ herda os genes de A posicionados entre os pontos 1 e p e entre q e n, além dos genes de B posicionados entre $p+1$ e $q-1$. Por fim, o cruzamento uniforme se utiliza de uma máscara binária que denota quais genes serão herdados de um cromossomo A (posição marcada por 1 na máscara) e quais serão herdados de um cromossomo B (posições marcadas por 0 na máscara).

A mutação em algoritmos genéticos tem como finalidade evitar a convergência prematura para um mínimo (ou máximo) local, fazendo com que outras regiões do espaço de solução possam ser exploradas. A mutação pode ser realizada através de uma modificação em um gene de um cromossomo de acordo com uma probabilidade p. Um exemplo de mutação é a alteração arbitrária de um valor de um gene como, por exemplo, anular uma característica do cromossomo (Fletcher, 2004). Assim, dado um cromossomo $C_1 = \{0.5, 0.6, 0.7\}$, pode-se efetuar a mutação substituindo o valor do primeiro gene por zero, gerando um novo cromossomo $C_1 = \{0, 0.6, 0.7\}$.

Para alguns problemas, pode ser interessante aplicar uma abordagem para mortalidade dos cromossomos (Sumida et al, 1990), evitando-se soluções sempre superiores (chamadas de super-indivíduos). Também pode ser utilizada para fazer com que indivíduos considerados ruins sejam constantemente atualizados, ou seja, a cada geração cromossomos ruins morrem e dão espaço para

que novos cromossomos ocupem seus lugares na população.

Em algoritmos evolucionários, o genótipo representa a codificação enquanto o fenótipo representa a solução. O genótipo deve ser decodificado para gerar o fenótipo. Os operadores de reprodução e mutação agem no genótipo enquando a função de aptidão utiliza o fenótipo do indivíduo.

2.3 Meta-alinhadores

A expressão Meta-Alinhamento de Ontologias foi introduzida por Euzenat & Shvaiko (2007) para nomear sistemas que tentam configurar automaticamente funções de alinhamento de ontologias. Em seguida, diversos trabalhos foram publicados para tratar esse problema. Em geral, existem algumas características em comum para as estratégias de meta-alinhamento (Martinez-Gil & Aldana-Montes, 2012):

1. Não é necessário que o processo de meta-alinhamento seja realizado em tempo de execução. As funções de alinhamento podem ser computadas em *background* e serem aplicadas em tempo de execução;

2. Este deve ser um processo automático. Assim, deve ser possível que o processo seja implementado por alguma ferramenta de alinhamento.

3. O processo deve retornar a melhor função de alinhamento possível. Caso esta função não seja conhecida, o processo deve retornar uma função mais próxima possível da melhor função de alinhamento, se comportando como uma especialista que experimenta diversas combinações e pesos de funções de alinhamento.

4. Uma estratégia de meta-alinhamento é avaliada com a função de alinhamento retornada.

Na literatura, as expressões combinação de alinhadores (*matcher combination*), auto-adaptação de alinhadores (*matcher self-tuning*) e meta-alinhamento (*meta-matching*) podem causar alguma confusão. Combinação de alinhadores diz respeito à ordem de combinação de certos alinhadores pertencentes a uma biblioteca de alinhadores. Esta tarefa aumenta a complexidade do problema de alinhamento, uma vez que é necessário identificar quais alinhadores devem ser combinados e de que forma essa combinação deve ser realizada. Segundo Martinez-Gil & Aldana-Montes (2012), este processo, atualmente, só pode ser realizado em tempo de projeto por ferramentas especializadas. A auto-adaptação de alinhadores é a tentativa de calibrar e adaptar automaticamente

soluções de alinhamento para as configurações de estabilidade e desempenho que a aplicação opera. Esse processo geralmente ocorre em tempo de execução, onde, por exemplo, os sistemas podem escolher alinhadores mais rápidos ao receber ontologias muito grandes. O meta-alinhamento de ontologia diz respeito à combinação de um conjunto de alinhadores heterogêneos. O principal objetivo desse processo é encontrar valores apropriados para seus pesos, pontos de cortes (*thresholds*) e quaisquer outros parâmetros que possam afetar os resultados do alinhamento. Ao contrário da auto-adaptação de alinhadores, o objetivo principal de um meta-alinhador não é manter um sistema funcionando de forma efetiva, mas gerar uma boa função de alinhamento de ontologias.

Meta-alinhadores podem realizar tarefas pré-alinhamento ou pós-alinhamento. As tarefas pré-alinhamento são realizadas automaticamente e podem não ser realizadas em tempo de execução. Estas tarefas consistem na seleção e treinamento dos alinhadores, além da configuração de parâmetros. Tarefas pós-alinhamento consistem na identificação de falsos positivos e falsos negativos. Uma vez que tenta-se evitar a intervenção do usuário, algumas técnicas utilizam estratégias de consulta a bases externas para melhorar os resultados gerados e checar consistências, por exemplo, como as medidas web propostas por Gracia & Mena (2008).

Segundo Martinez-Gil & Aldana-Montes (2009), quanto às técnicas utilizadas para resolver o problema, pode-se classificar as abordagens de meta-alinhamento em dois grandes grupos: o meta-alinhamento baseado em aprendizado de máquina e o meta-alinhamento heurístico.

Abordagens baseadas em aprendizado de máquina utilizam avaliação de relevância dos alinhamentos iniciais através da interação com o usuário (Ehrig et al, 2005), aprendizado bayesiano para capturar interdependências entre os alinhadores e, assim, tentar melhorar o modo como esses alinhadores são combinados (Duchateau et al, 2009), árvores de decisão para decidir quais alinhadores utilizar com base em critérios definidos pelo usuário, como a redução do custo de processamento (Duchateau et al, 2008) e, por fim, treinamento de redes neurais utilizando diferentes bases de testes e, então, utilizar o conhecimento para predizer novas funções de similaridade (Huang et al, 2007; Mao et al, 2008; Spohr et al, 2011). Conforme apontado por Martinez-Gil & Aldana-Montes (2012), as abordagens baseadas em aprendizado de máquina geralmente são utilizadas em tempo de projeto, uma vez que demandam considerável processamento na fase de treinamento da abordagem.

Abordagens heurísticas para meta-alinhamento de ontologias utilizam diversas técnicas como algoritmos evolutivos, algoritmos baseados em regras e

algoritmos gulosos. Uma vez que este trabalho se encaixa nesta abordagem, as abordagens heurísticas propostas na literatura serão apresentadas e discutidas com detalhes na próxima seção.

2.4 Abordagens heurísticas para meta-alinhamento

As ferramentas mais conhecidas para meta-alinhamento de ontologias que utilizam técnicas heurísticas são as ferramentas eTuner (Lee et al, 2007), GAOM (Wang et al, 2008), GOAL (Martinez-Gil & Aldana-Montes, 2009), MaSiMe (Martinez-Gil & Aldana-Montes, 2009) e as abordagens de Gînsca & Iftene (2010) e Acampora et al (2012).

Outras abordagens heurísticas não foram incluídas nesta análise por não serem consideradas abordagens de meta-alinhamento. É o caso das ferramentas MapPSO (Bock & Hettenhausen, 2010) e MapEVO (Bock et al, 2012), as quais utilizam, respectivamente, uma abordagem de otimização por enxame de partículas[2] e programação evolucionária[3]. Estas propostas buscam o melhor alinhamento através de funções objetivos que utilizam suas próprias funções de alinhamento, tratando, assim, o problema de alinhamento, não de meta-alinhamento. Além disso, os resultados alcançados por estas propostas não foram significativos em comparação com os resultados alcançados pelas demais propostas heurísticas, como pode ser visto em (Bock et al, 2012).

2.4.1 Ferramenta eTuner

O eTuner é uma ferramenta de meta-alinhamento que automaticamente calibra um sistema de alinhamento de ontologias que produz alinhamentos do tipo 1:1. Para tal, o eTuner utiliza um algoritmo de regras de perturbação que escolhe os melhores parâmetros e os alinhadores mais efetivos.

A partir de um esquema S e um parâmetro n, o eTuner aplica um conjunto de regras de transformação no esquema S e nos dados de S, randomicamente perturbando o esquema S, para gerar uma coleção de esquemas sintéticos $S_1, S_2, \ldots, S_n$. As regras de perturbação são definidas pela ferramenta e incluem regras para perturbação de dados dos esquemas (instâncias), nomes e estruturas. Uma vez gerado os esquemas $S_1, S_2, \ldots, S_n$ a partir de S, o eTuner

[2]Otimização por enxame de partículas simula o comportamento social de organismos naturais, como a reunião de pássaros ou peixes para encontrar um local com alimento suficiente (Shi & Eberhart, 1998).

[3]Computação evolucionária simula espécies que competem entre si no espaço do problema. Ao contrário de algoritmos genéticos, a atualização da população não envolve recombinação genética (Whitley, 2001).

infere os alinhadores corretos para gerar os alinhamentos entre o melhor esquema sintético S_i e S.

Para encontrar o melhor esquema sintético, a ferramenta utiliza uma busca gulosa em um conjunto finito de esquemas sintéticos gerados randomicamente. A escolha dos parâmetros para definir o tamanho do conjunto e a seleção dos dados que serão perturbados fazem parte da estratégia de busca adotada pela ferramenta.

Segundo Lee et al (2007), o desempenho do sistema é muito bom, comparados com os demais sistemas de calibragem, uma vez que não é necessário treinamento para a escolha das regras de pertubação que serão utilizadas. Por outro lado, ainda segundo os autores, a estratégia de busca adotada pelo eTuner é adequada para a calibragem de esquemas de tamanho médio a moderado e em tempo de projeto somente, uma vez que a quantidade de esquemas sintéticos que necessitam ser criados em esquemas de tamanho grande pode inviabilizar o processo de busca.

2.4.2 Ferramenta MaSiMe

O MaSiMe é uma ferramenta que implementa uma medida de similaridade customizável que tenta estabelecer os pesos mais apropriados para uma composição de alinhadores. A ferramenta MaSiMe utiliza uma estratégia gulosa que consiste na utilização de múltiplos de um valor (chamado de granularidade) para reduzir o espaço de soluções. A medida de similaridade entre os conceitos c_1 e c_2 utilizada pelo MaSiMe é definida conforme a equação 2-3.

$$MaSiMe(c_1, c_2) = x \in [0,1] \in \mathbb{R} \rightarrow \exists \langle \overrightarrow{A}, \overrightarrow{w}, g \rangle, x = max(\sum_{i=1}^{i=n} A_i w_i), \quad (2\text{-}3)$$

com as seguintes restrições: $\sum_{i=1}^{i=n} w_i \leq 1 \wedge \forall w_i \in \overrightarrow{w}, w_i \in \{\dot{g}\}$, onde $\overrightarrow{A}$ é um vetor de alinhadores, $\overrightarrow{w}$ é um vetor numérico de pesos, g é uma granularidade e $\dot{g}$ é o conjunto formado pelos múltiplos de g.

A estratégia gulosa do MaSiMe consiste na geração de todos os resultados possíveis no espaço de soluções reduzido e na escolha da melhor solução encontrada. Assim, seja S o conjunto de todos os alinhadores, seja A um subconjunto de S, seja g a granularidade, seja Q o conjunto dos números racionais positivos, seja $i, j, k, \ldots, t$ os índices pertencentes ao conjunto $\dot{g}$ de múltiplos da granularidade, então um conjunto de vetores racionais r existe onde cada elemento r_i é o resultado do produto escalar entre A e o padrão de índices $(i, j-i, k-j, \ldots, 1-t)$, sendo todos os índices sujeitos a

$j \geq i \wedge k \geq j \wedge 1 \geq k$. Além disso, o resultado final, denotado por R, é o máximo dos elementos r_i e sempre menor ou igual a 1.

Na forma matemática, tem-se: $\exists A \subset S, \exists g \in [0,1] \in Q+, \forall i, j, \ldots, t \in \{\dot{g}\} \rightarrow \exists \overrightarrow{r}, r_i = \overrightarrow{A} \cdot (i, j-i, k-j, \ldots, 1-t)$, com a restrição $j \geq i \wedge k \geq j \wedge 1 \geq k$. $R = max(r_i) \leq 1$.

A discretização do espaço de soluções garante que o algoritmo encontre uma solução em tempo finito. Por outro lado, o MaSiMe necessita avaliar cada solução r_i através do produto escalar dos alinhadores em $\overrightarrow{A}$ e dos pesos $\overrightarrow{w}$ para um par de conceitos c_1 e c_2. Uma vez que o MaSiMe utiliza somente um par de entrada, a solução R encontrada é aplicada para alinhar toda a ontologia. Assim, a escolha do par de entrada pode influenciar substancialmente a qualidade da solução.

2.4.3 Ferramenta GAOM

O GAOM (*Genetic Algorithm based Ontology Matching*) foi o primeiro meta-alinhador de ontologias baseado em algoritmo genético. O meta-alinhamento é modelado como uma otimização de um mapeamento entre duas ontologias. O problema é tratado como um processo de combinação de características. Cada ontologia é caracterizada por um conjunto de características extensionais e intensionais e, em seguida, é realizada uma busca para encontrar a melhor combinação. Características intensionais de um conceito são definidas como uma tupla (n, p, I) que descreve a essência do conceito, onde n é o nome do conceito, p é o conjunto de propriedades relacionadas com o conceito e I é o conjunto de instâncias associadas ao conceito. Características extensionais de um conceito é definido como o conjunto R de relações do conceito com outros conceitos da ontologia.

Na sua abordagem genética, o GAOM representa cada indivíduo como um vetor numérico de tamanho n_1 para armazenar valores entre 1 e n_2, onde n_1 é a quantidade de entidades de uma ontologia o_1 e n_2 é a quantidade de entidades de uma ontologia o_2. Assim, cada indivíduo é denotado como $N_1 N_2 N_3 \ldots N_{n_1}$, onde $M(i) \in \{1, 2, \ldots, n_2\}, i \in \{1, 2, \ldots, n_1\}$ e a representação significa que o i-ésimo conceito em o_1 é mapeado para N_i-ésimo conceito em o_2.

Uma vez que uma ontologia é representada como uma coleção de características, o GAOM calcula a similaridade entre duas ontologias como um processo de combinação de características. Assim, a função objetivo utilizada mede a similaridade global entre duas ontologias o_1 e o_2 conforme a equação

2-4.

$$S_{o_1,o_2}(M) = \frac{f(C)}{f(C) + \alpha f((F_{o_1} - F_{o_2})|M) + \beta f((F_{o_2} - F_{o_1})|M)} \quad (2\text{-}4)$$

Na equação 2-4, M é um indivíduo da população que corresponde a um mapeamento; F_{o_1} e F_{o_2} são conjuntos de características de uma ontologia o_1 e o_2, respectivamente; $C = (F_{o_1} \cap F_{o_2})|M$, o qual denota o conjunto de conceitos mapeados de F_{o_1} e F_{o_2} em relação ao mapeamento M; $(F_{o_1} - F_{o_2})|M$ e $(F_{o_2} - F_{o_1})|M$ são dois conjuntos de conceitos não mapeados em relação ao mapeamento M; α e β são dois parâmetros entre 0 e 1 que determinam a relevância dos dois conjuntos de conceitos não mapeados; f é uma função definida como a cardinalidade do conjunto.

Dadas duas ontologias e um mapeamento M, ao computar a aptidão do indivíduo com a equação 2-4, o GAOM verifica o quanto o mapeamento M contribui para o mapeamento de todas as entidades das duas ontologias. A melhor solução para o GAOM é aquela que representa o mapeamento de todas as entidades das duas ontologias.

2.4.4 Ferramenta GOAL

O GOAL (*Genetics for Ontology Alignments*) é um algoritmo genético para calibrar alinhadores automaticamente utilizando alinhamentos fornecidos por especialistas. Cada indivíduo é representado como um vetor numérico onde cada gene determina o peso associado a um alinhador. O GOAL trata o problema de calibragem como uma otimização de uma medida de qualidade utilizada na área de Recuperação de Informação.

O GOAL permite a utilização de quatro funções objetivos, embora somente uma possa ser utilizada por vez. Cada função objetivo avalia o indivíduo em relação ao alinhamento esperado para verificar a precisão, cobertura, medida-F ou *fall-out* (ver seção 5.1 para detalhes sobre essas medidas de qualidade). Assim, para que o GOAL possa funcionar corretamente, é necessário que os especialistas de domínio forneçam casos de teste suficientes para que possa ser realizada, por exemplo, a maximização da precisão ou a minimização dos falsos positivos no alinhamento gerado.

Nesta abordagem, a avaliação de um indivíduo I da população consiste na geração de um alinhamento A' utilizando um conjunto de alinhadores pré-determinados e aplicando a estes alinhadores os pesos representados em I. Com o alinhamento gerado, a função objetivo escolhida avalia cada correspondência

em A' em relação às correspondências ao alinhamento de referência A. Diferente de algumas abordagens que utilizam algoritmos genéticos, o GOAL é capaz de encontrar soluções muito boas para o problema de calibragem, uma vez que o algoritmo tem conhecimento dos alinhamentos de referência. Por outro lado, o custo computacional para avaliar cada indivíduo da população é alto, uma vez que cada avaliação pressupõe a execução de todos os alinhadores para gerar um novo alinhamento e uma comparação entre as ocorrências contidas no alinhamento gerado e no alinhamento de referência.

2.4.5 Abordagem de Gînsca & Iftene (2010)

Gînsca & Iftene (2010) apresentam uma abordagem genética para o problema de meta-alinhamento que, segundo os autores, guarda muitas similaridades com a abordagem desenvolvida pelo sistema GOAL (seção 2.4.4). A principal diferença entre as abordagens está no fato de que, enquanto o GOAL tenta otimizar uma agregação de alinhadores, esta abordagem tenta otimizar outros parâmetros do sistema, como o ponto de corte (*threshold*).

Pode-se, contudo, apontar outras características importantes desta abordagem. Em primeiro lugar, a abordagem tenta diminuir o tempo de processamento do meta-alinhamento ao inserir uma fase de pré-processamento. Nesta fase de pré-processamento, os alinhadores que serão utilizados para alinhar as ontologias são executados e os resultados gerados por cada alinhador para o par de ontologias de entrada é armazenado em um arquivo XML. Durante a execução do algoritmo genético, sendo necessário recuperar o valor de alguma similaridade, o valor é recuperado do arquivo XML ao invés de ser recalculado.

Um cromossomo é definido como uma sequência de bits que representa uma coleção de correspondências entre os conceitos das duas ontologias e um valor de ponto de corte. Assim, cada cromossomo representa um alinhamento e seu ponto de corte. Assim como no sistema GOAL, Gînsca & Iftene (2010) permitem a utilização de uma das quatro medidas de qualidade do alinhamento utilizadas pelo GOAL para avaliar a aptidão do cromossomo. Assim, é necessário que o algoritmo genético conheça o alinhamento de referência para que a função objetivo possa guiar o comportamento do algoritmo.

2.4.6 Abordagem de Acampora et al (2012)

Acampora et al (2012) apresentam uma abordagem heurística para resolver o problema de meta-alinhamento de ontologias como um problema

de otimização. Os autores utilizam algoritmos meméticos (Moscato & Cotta, 2003). Algoritmos meméticos são métodos evolucionários híbridos baseados em metaheurísticas populacionais e na aplicação de métodos de melhoramentos locais da solução. Neste método, cada indivíduo passa por um refinamento local dentro do espaço de busca, de modo que o indivíduo pode ter seu nível de aptidão aumentado após passar pela etapa de refinamento.

Os autores utilizam uma função objetivo $F(A) = \sum_{i=1}^{|A|} f(c_i)$, onde A é o conjunto de correspondências c_i entre as duas ontologias alinhadas e $c_i \in A$. A função f é denominada função de adequação, a qual associa um valor no intervalo $[0, 1]$ para cada ocorrência c_i no alinhamento A. A função f é utilizada para avaliar a qualidade de uma correspondência para alcançar o alinhamento ótimo. Esta função é calculada como uma agregação utilizando pesos e uma coleção de medidas de similaridade, como mostra a equação 2-5.

$$f(c_i) = \phi(\overrightarrow{s}(c_i), \overrightarrow{w}) \tag{2-5}$$

onde a função ϕ representa uma estratégia de agregação, a qual combina o vetor de medidas de similaridades $\overrightarrow{s}$ considerando o vetor de pesos $\overrightarrow{w}$.

Como restrição do trabalho, Acampora et al (2012) somente consideram alinhadores que implementam medidas de distância. Dessa forma, o problema de meta-alinhamento é modelado como uma minimização da função objetivo, onde, consequentemente, um valor de $F(A)$ próximo de zero corresponde a um alinhamento A próximo do alinhamento ótimo.

Para representação dos indivíduos, cada cromossomo S representa um alinhamento A entre duas ontologias o_1 e o_2. Assim $S = \{(e_0, e_{j_0}), (e_1, e_{j_1}), \ldots, (e_h, e_{j_h})\}$, onde $h = |o_1| - 1$ e $j_l \in \{0, 1, 2, \ldots, |o_2| - 1\}$ com $l = 0, 1, 2, \ldots, h$. Com isso, para avaliar a aptidão de cada indivíduo, é necessário recalcular a função de adequação f (equação 2-5) para cada indivíduo, o que, por consequência, exige o cálculo de cada medida de similaridade para cada gene do indivíduo a cada avaliação.

Como critério de parada, os autores adotam as mesmas medidas de qualidade adotadas pelo sistema GOAL (seção 2.4.4), sendo necessário que o algoritmo memético tenha conhecimento do alinhamento de referência para retornar bons alinhamentos.