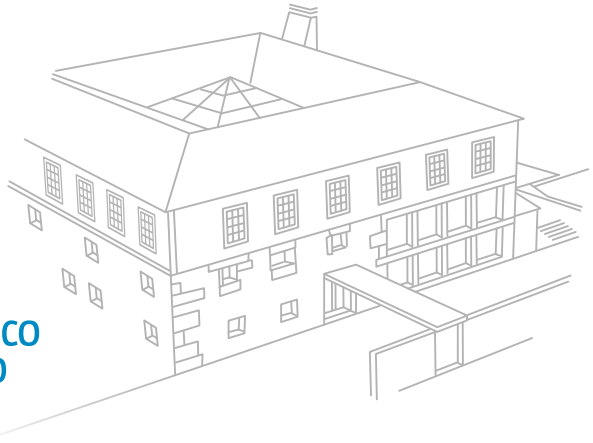


ESTGF | POLITÉCNICO DO PORTO

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Integração Semântica de Modelos Conceptuais

DESIGNAÇÃO DO MESTRADO Engenharia Informática

AUTOR Luís Miguel Mota e Costa

ORIENTADOR(ES) Professora Doutora Carla Sofia Pereira

Professor Doutor Cristovao Dinis Sousa

ANO 2016

AGRADECIMENTOS

Concluída mais uma etapa do meu percurso académico, gostaria de agradecer àqueles que direta e indiretamente contribuíram para a concretização deste objetivo que, por muitas vezes, acabou por se tornar num grande desafio a ser conquistado.

Agradeço à minha família e, em especial, aos meus pais, pelo carinho, compreensão e apoio incondicional.

À minha orientadora, a Professora Carla Pereira, e ao meu coorientador, o Professor Cristóvão Sousa, pelos ensinamentos, pela atenção e pelo auxílio, desde a realização do curso de Licenciatura em Engenharia Informática, e, mais efetivamente, durante todo o desenvolvimento desta investigação.

À minha equipe de trabalho do INESC TEC, pela colaboração nos momentos decisivos para a finalização deste estudo.

E aos meus amigos, pelo incentivo e por sempre acreditarem em mim!

Muito obrigado!

RESUMO

A criação de modelos conceptuais permite a representação de um domínio, sob a forma de conceitos e relações. Quando inserida num contexto colaborativo, a conceptualização apresenta melhores resultados, no entanto, o aparecimento de diversas soluções para um domínio é um problema muito comum. Desta forma, a partir dos modelos elaborados, os especialistas devem chegar a um consenso sobre quais conceitos irão compor o modelo final. Nesta perspetiva, este trabalho pretende fornecer uma ferramenta para a integração de modelos conceptuais e, assim, ajudar os especialistas na difícil tarefa de negociação, de forma a desenvolverem um modelo final partilhado. A partir da utilização de medidas de similaridade semântica e sintática, foi desenvolvida uma abordagem semântica, que analisa conceitos de dois modelos e indica aos especialistas as possibilidades de conceitos repetidos. Como resultado prático, foram implementados dois artefactos, um serviço *web* e uma interface cliente. O serviço *web* calcula a similaridade semântica e é decomposto por três fases: normalização, análise sintática e análise semântica. A interface cliente auxilia os especialistas, ao longo do processo de integração dos modelos. Para avaliar a abordagem proposta, foram calculados os valores das medidas *precision* e *recall* em dois cenários de aplicação prática. Os resultados obtidos revelaram-se melhores em comparação com as ferramentas existentes, aplicadas a modelos semi-formais (mapas de conceitos), e muito próximos das melhores ferramentas orientadas para modelos formais (ontologias).

Palavras-chave: Similaridade semântica, Modelação conceptual, Conceptualização partilhada.

ABSTRACT

The creation of conceptual models allows for the representation of a domain by using concepts and relations between them. In a collaborative context, the conceptualization provides better results. Despite this, the existence of several solutions for a given domain is a very common problem. Given this, specialists must reach a consensus on the concepts that will encompass the final solution. Therefore, this work aims to provide a tool for the integration of conceptual models for helping specialists during the negotiation phase of developing the final shared model. A semantic approach was developed using measures of semantic and syntax similarity. This approach analyses the concepts of two models and shows the repeated concepts to the specialists. The final solution encompasses two artefacts: a web service and a client interface. The web service calculates the semantic similarity and includes three stages, namely: normalization, syntax analysis and semantic analysis. The user interface helps the specialist during the integration process of the models. To evaluate the proposed approach, the values of precision and recall measures were calculated in two practical application scenarios. The obtained results proved to be better when compared to the existing tools, applied to semi-formal models (concept maps), and very close to the best tools oriented to formal models (ontologies).

Keywords: Semantic similarity, Conceptual modelling, Shared conceptualization.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

API: *application programming interface*

CCP: *collaborative conceptualisation process*

CBT: *conceptual blending theory*

ChEBI: *chemical entities of biological interest*

CCM: *collaborative concept mapping*

CCP: *collaborative conceptualisation process*

CRC: conceito-relação-conceito

CRRM: *conceptual relations reference model*

DL: *description logic*

FCA: *formal concept analysis*

FP: *false positive*

FN: *false negative*

GO: *gene ontology*

GUI: *graphical user interface*

HCONE: *Human Centered Ontology Environment*

IA: Inteligência artificial

IC: *information content*

IDF: *inverse document frequency*

LCS: *least common subsumer*

LSI: *latent semantic index*

MAFRA: *Mapping Framework for distributed ontologies*

MeSH: *Medical Subject Headings*

MVC: *model – view – controller*

NLP: *natural language process*

OAEI: *Ontology Alignment Evaluation Initiative*

OSM: *Open Street Map*

OWL: *Web Ontology Language*

RDF: *resource description framework*

SBO: *semantic bridging ontology*

SI: sistemas de informação

SMW: *semantic media wiki*

SPSM: *structure preserving semantic matching*

TF: *term frequency*

TF-IDF: *term frequency – inverse document frequency*

TN: *true negative*

TP: *true positive*

1 INTRODUÇÃO

1.1 Contextualização

A informação e o conhecimento têm um papel vital nas organizações da atualidade, pois estão presentes em todos os processos de desenvolvimento das suas atividades, sendo que muitas delas têm sido realizadas de forma colaborativa. Assim, é imprescindível a utilização de sistemas para a gestão da informação e do conhecimento.

A modelação conceptual surge como uma forma de representação do conhecimento, pois estabelece um conjunto de conceitos e relações para um determinado domínio, ou seja, para uma área de aplicação. Num processo de conceptualização colaborativo, várias pessoas ou grupos discutem entre si para chegar a um consenso na criação de conceitos e de relações sobre um domínio. Por vezes, para um determinado problema, surgem várias soluções, e é necessário acordar sobre quais conceitos e relações serão utilizados para se alcançar um modelo final partilhado [1].

No decorrer do processo de conceptualização, detalhado por Sousa [2], um domínio pode ser representado sob a forma de um mapa conceptual. São criadas estruturas do tipo conceito - relação - conceito (CRC), em que cada conceito e cada relação podem conter propriedades semânticas. Em Costa et al. [3] é apresentada uma plataforma que tem como objetivo apoiar a criação de modelos conceptuais, introduzindo mecanismos que auxiliam os especialistas ao longo do processo de conceptualização colaborativa.

Um processo de conceptualização termina quando, para um problema, existe uma solução com o consenso dos especialistas. A escolha dos conceitos que irão compor a solução final é uma tarefa complexa e requer grande esforço. Pereira [1] demonstra em detalhe as várias etapas existentes numa negociação conceptual, ao longo de um processo de conceptualização colaborativa.

Como forma de facilitar a tarefa de negociação, e auxiliar os especialistas na integração de modelos conceptuais, podem ser exploradas medidas de similaridade semântica. Atualmente, a similaridade semântica é vastamente utilizada e possui diferentes tipos de aplicação. De uma forma geral, as medidas de similaridade semântica podem ser usadas para [4]:

a) Melhorar a precisão das técnicas de recuperação de informação;
b) Realizar correspondências semânticas;
c) Integrar ontologias;
d) Anotar documentos de forma automática; e
e) Resolver problemas na reutilização de conhecimento.

De acordo com o contexto exposto, pretende-se apresentar uma proposta para calcular similaridades entre conceitos presentes em diferentes modelos conceptuais. A abordagem desenvolvida utiliza um conjunto de medidas para avaliar tanto a componente sintática como a componente semântica

entre os conceitos. No campo sintático, as medidas avaliam unicamente o grau de semelhança entre conceitos, pela sua composição lexical. Por sua vez, no campo semântico, as medidas para os cálculos de similaridade podem:

a) Recorrer a ontologias e taxonomias externas;
b) Avaliar a estrutura conceptual; e
c) Utilizar o *corpus* do conceito (conjunto de frases extraídas de documentos do domínio ao qual o conceito pertence).

1.2 Descrição do problema

Num processo de conceptualização colaborativa, a obtenção de múltiplas soluções, para um determinado domínio, implica uma atividade de negociação para que se obtenha um consenso sobre quais os conceitos irão compor o resultado final. A integração de modelos conceptuais pode tornar-se uma tarefa complexa, e, quanto maior o nível de discrepância entre os conceitos utilizados nos modelos, maior será a dificuldade para que se chegue a um acordo que dê origem ao modelo final. Resumidamente, a atividade de obtenção do consenso é longa e árdua para os especialistas de determinado domínio devido às seguintes razões:

a) Os especialistas têm dificuldade em explicitar o seu conhecimento;
b) Em grandes repositórios de conhecimento, torna-se difícil acompanhar as modificações propostas, as razões que sustentam essas modificações e as suas repercussões no modelo conceptual final;
c) Há a necessidade de estabelecer consenso entre várias pessoas; e
d) A informação e o seu significado são dependentes do tempo e do contexto.

A principal problemática, quando se elaboram modelos conceptuais de forma colaborativa, está na dificuldade de se chegar a um modelo único. Atualmente, existem algumas ferramentas para integração de ontologias e dados estruturados (serão apresentadas no capítulo 3). No entanto, nenhuma das soluções existentes consegue responder às especificidades da integração de modelos conceptuais construídos de forma livre, sem domínio específico e em múltiplas línguas.

Deste modo, o foco principal está na complexidade existente na fase de negociação conceptual, e é neste ponto que este trabalho está focado. Na Figura 1 é possível verificar precisamente a aplicação desta investigação.

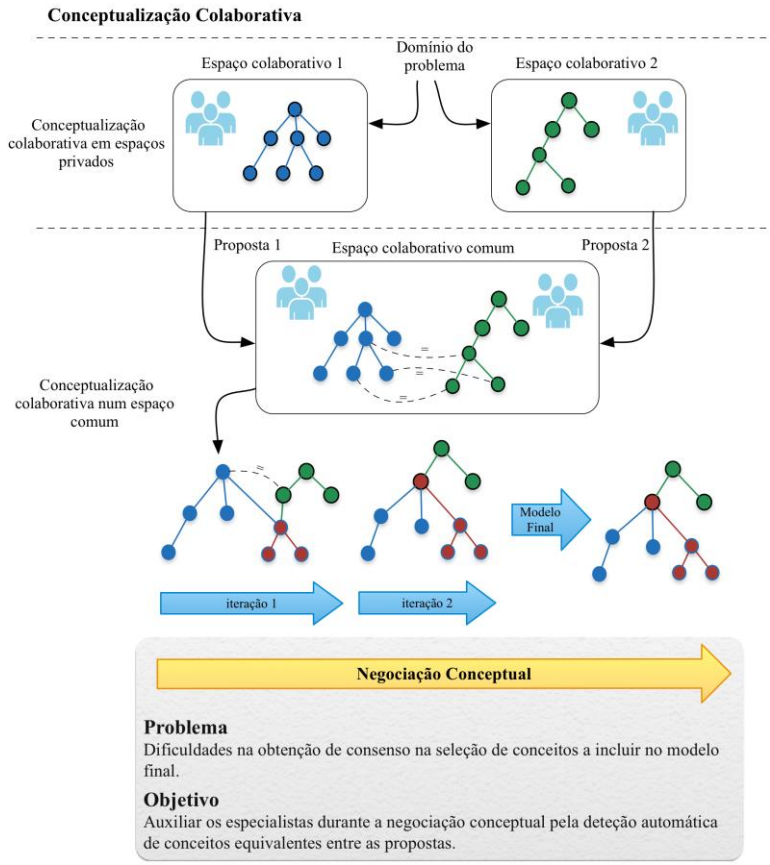


Figura 1: Processo de conceptualização colaborativo, problema e objetivo deste estudo

1.3 Questão de investigação e objetivos

Apresentados o contexto e a problemática onde está inserida esta dissertação, pretende-se responder à seguinte questão de investigação: como fornecer, aos especialistas de domínio, o apoio adequado nas atividades de integração de modelos conceptuais inerentes a um processo de conceptualização colaborativa, garantido o entendimento comum sobre o conhecimento gerado?

Para responder a essa questão, o objetivo é apoiar a integração de modelos conceptuais, através do alinhamento semântico entre os modelos, de modo a promover o consenso sobre um modelo unificado final. Para tal, será necessário encontrar mecanismos de medição sistemática da similaridade semântica e sintática dos modelos gerados pelos especialistas e das transformações que estes sofrem até o modelo comum final.

De modo a alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

a) Identificar e categorizar as medidas de avaliação de similaridade semântica entre modelos;
b) Selecionar as medidas mais apropriadas para suportar as atividades de integração de modelos conceptuais semi-formais (exemplo: mapas de conceitos); e
c) Desenhar uma abordagem híbrida que permita, de uma forma interativa, apoiar a integração de modelos conceptuais, através do sucessivo cálculo da similaridade semântica entre os diferentes elementos que compõem os modelos conceptuais, ou seja, conceitos e relações conceptuais.

1.4 Contribuições e resultados

Como contribuição deste estudo, procurou-se melhorar a eficiência e a eficácia da fase de negociação conceptual, durante o processo de conceptualização colaborativa, a partir da introdução de medidas de similaridade semântica. Elaborou-se uma solução independente, que pode ser utilizada por qualquer ambiente de desenvolvimento de modelos conceptuais.

Em relação aos resultados, foi criada uma ferramenta para integrar os modelos conceptuais, para múltiplos domínios, composta por:

a) Um serviço para o cálculo de similaridade de modelos conceptuais, disponível na *web*, e que possa ser invocado por qualquer aplicação; e
b) Uma interface cliente para a interação entre os especialistas do domínio e o serviço desenvolvido.

1.5 Estrutura da dissertação

Este documento está estruturado em seis capítulos, dos quais o primeiro é composto pela introdução, que faz a contextualização da investigação, a descrição do problema, a questão de investigação, os objetivos pretendidos, as contribuições e os resultados esperados.

A fundamentação teórica é realizada no segundo capítulo e apresenta os principais temas que compõem a dissertação, que são: modelação conceptual, colaboração, modelação conceptual colaborativa, negociação conceptual, ontologia e taxonomia. Neste segundo capítulo, o tema similaridade semântica, foco desta investigação, é descrito com mais detalhe. São apresentadas as suas áreas de aplicação, os mecanismos de interoperabilidade de ontologias e as abordagens para cálculo de similaridade semântica e sintática.

No terceiro capítulo é realizado um estudo sobre o estado da arte das medidas de similaridade e das aplicações existentes para integração de modelos. As medidas encontram-se organizadas em dois grandes grupos: semânticas e sintáticas. As medidas semânticas, por sua vez, estão divididas em: ontologias, conteúdo de informação, caraterísticas e híbridas. Também são apresentadas as *frameworks*, as *application programming interfaces* (API's) e as ferramentas existentes para o cálculo da similaridade e para a integração de modelos.

O quarto capítulo descreve o processo de similaridade semântica proposto neste trabalho. É constituído pelos tópicos: levantamento de requisitos; descrição das fases do serviço; interação com o cliente; aspetos técnicos do serviço; e as tecnologias e a arquitetura que compõem a proposta.

No quinto capítulo foram analisados dois cenários práticos de utilização para avaliar os resultados obtidos pelo serviço desenvolvido neste trabalho.

A investigação é concluída no sexto capítulo, que também indica possibilidades de estudos futuros.

2 FUNDAMENTOS TEÓRICOS

Neste capítulo são abordados os principais conceitos que compõem esta investigação, entre eles estão: modelação conceptual, colaboração, negociação conceptual, ontologia, taxonomia e similaridade semântica.

2.1 Modelação conceptual

A modelação conceptual é a atividade onde, formalmente, são descritos alguns aspetos do ambiente físico e social que nos envolve, com o intuito de alcançar um entendimento comum e facilitar a comunicação [5]. Tipicamente, da atividade de modelação conceptual resultam modelos artificiais, isto é, elaborados segundo processos desenhados por e para pessoas, e genéricos, cujo intuito é representar a conceptualização de um determinado domínio [6].

Durante um processo de conceptualização é realizada uma normalização da linguagem natural, de modo que um modelo conceptual seja representado por estruturas conceptuais do tipo conceito - relação - conceito (CRC). Desta normalização, podem resultar mapeamentos consensuais de diversos domínios, ou seja, de diversas áreas de conhecimento [1].

O uso de uma representação visual torna-se fundamental para uma melhor perceção dos resultados de uma conceptualização. Os mapas conceptuais, criados por Novak em 1972, constituem uma estrutura basilar na construção e representação do conhecimento de um indivíduo, num determinado domínio [7], [8]. Os autores Novak et al. [9] e Yao et al. [10] destacam que os mapas de conceitos são considerados como uma notação gráfica flexível para representação de conhecimento.

Num mapa conceptual, a ligação entre conceitos, levada a cabo por preposições, é um pré-requisito obrigatório para a compreensão conceptual individual [9], [11]. Os mapas conceptuais representam o conhecimento de uma forma gráfica, a partir de nós que ilustram os conceitos e de arcos que demonstram as relações entre conceitos. A estrutura dos mapas conceptuais é definida, de um modo geral, pela forma como os conceitos e arcos estão dispostos.

Existem várias formas de organizar e dispor os conceitos num mapa conceptual. Novak defende que, de modo a facilitar a aprendizagem e a interpretação, os conceitos mais generalistas e inclusivos deverão estar situados no topo do mapa, da mesma forma que os conceitos mais específicos e menos inclusivos deverão estar dispostos abaixo dos primeiros [9], [8]. Contudo, Canãs [11] afirma que os mapas conceptuais não são estritamente hierárquicos na sua organização, pois seguem uma lógica semi-hierárquica.

O nível de informação fornecida pelos mapas conceptuais está dependente de uma caraterística denominada na literatura como direcionalidade (*directedness*). Enquanto um mapa conceptual, com alta direccionalidade, fornece aos seus utilizadores os conceitos, as relações que os ligam, as preposições nas ligações e uma estrutura do mapa, de modo a melhor ilustrar um dado domínio, um mapa com baixa

direccionalidade atribui aos utilizadores a tarefa de decidir que conceitos deverão constar, bem como as ligações que existirão entre estes nos respetivos mapas [12].

Na construção de um modelo conceptual podem ser utilizadas ferramentas concebidas para esse intuito. A ferramenta CMapTools[1], por exemplo, permite a criação da estrutura relacional mencionada previamente, resultando em mapas conceptuais. A Figura 2 ilustra um exemplo de um mapa conceptual elaborado com a ferramenta CMapTools.

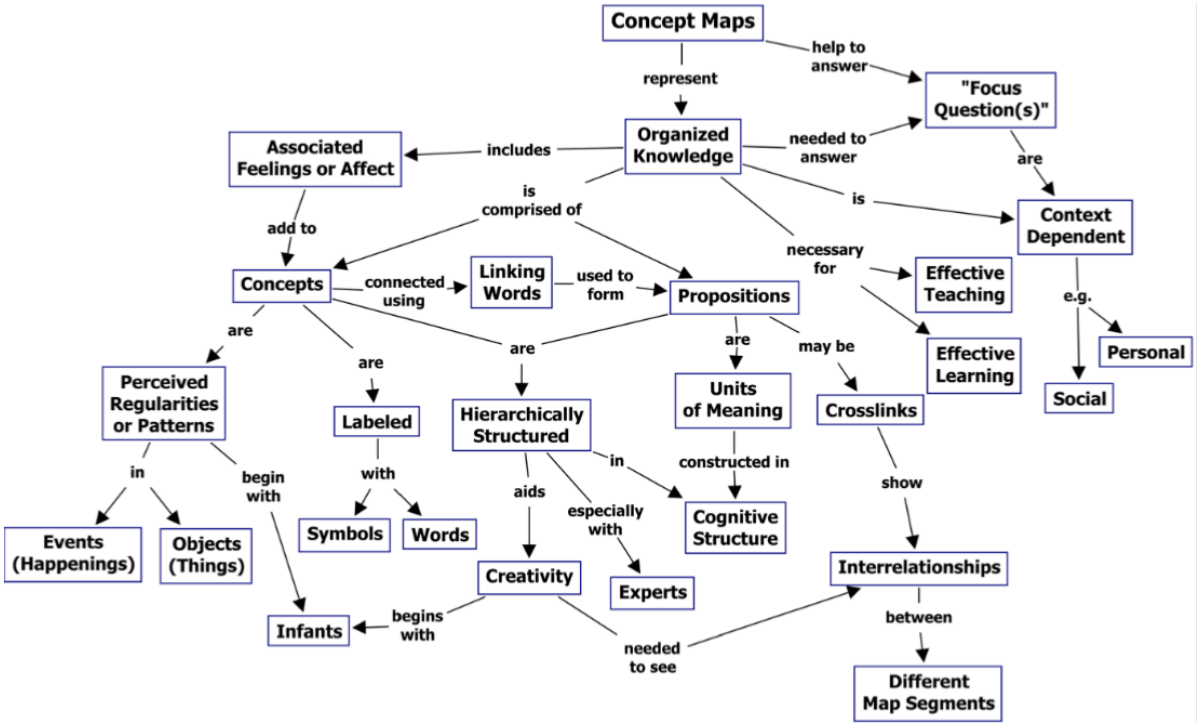


Figura 2: Exemplo de mapa conceptual [9]

Como mencionado anteriormente, é possível verificar a existência dos nós rotulados (retângulos), representando os conceitos e as ligações (arcos), contendo proposições que resultam no estabelecimento de relações entre os conceitos. No exemplo fornecido é também possível verificar a estrutura semi-hierárquica com a qual os conceitos estão dispostos.

A seção seguinte irá abordar o tema da colaboração, no sentido de aclarar a sua relevância, não apenas no processo de modelação conceptual, mas também de negociação conceptual.

2.2 Colaboração

A colaboração entre pessoas ou grupos de pessoas ganhou relevância em atividades de criação de conhecimento. Colaboração significa trabalhar em conjunto para cumprir um objetivo final comum e pode ser vista como um processo de criação partilhado. Colaborar implica a existência de um acordo mútuo entre os participantes para resolverem os problemas em conjunto [13], [14]. O conceito de colaboração surge muitas vezes confundido com o de cooperação, no entanto, existem diferenças entre

[1] **CMapTools**: http://ftp.ihmc.us/

eles. A colaboração implica a participação conjunta de diferentes pessoas numa atividade com vista a completar um determinado objetivo. Por sua vez, a cooperação envolve também um grupo de pessoas, mas existe uma divisão do trabalho entre cada participante. É da responsabilidade de cada participante desenvolver o seu trabalho de uma forma quase independente, mas coordenada com outros participantes [13].

Atualmente, as atividades colaborativas têm vindo a adquirir destaque na criação de conhecimento e na aprendizagem. Numa aprendizagem colaborativa, Maria et al. [15] consideram as seguintes vantagens:

- Facilita, em tempo real ou de forma assíncrona, a troca de informação via texto, voz e vídeo;
- Auxilia nas atividades básicas de gestão de projetos, como por exemplo, gestão de tarefas, gestão do fluxo de trabalho e gestão do tempo;
- Apoia a criação de conhecimento em grupo, permitindo a modificação da informação gerada, de uma forma assíncrona ou em tempo real;
- Facilita na obtenção de um consenso através de discussões em grupo;
- Melhora a forma de partilhar e aceder à informação criada.

Num contexto empresarial, empresas ou organizações também podem ser beneficiadas se adotarem atividades colaborativas. Alguns dos benefícios de uma colaboração empresarial, identificados por Bititci et al. [16], são:

- Partilhas de dados, informação, tecnologias, sistemas, riscos e benefícios [17];
- Aumentar a utilização dos ativos;
- Aumentar a quota de mercado;
- Melhorar a competência e conhecimento;
- Diminuir o risco de falha do desenvolvimento do produto.

Apesar das vantagens apresentadas, existem barreiras que precisam ser ultrapassadas quando se fala em colaboração. Os autores Pirkkalainen e Pawlowski [18] definem barreira como desafio, risco, dificuldade, obstáculo, restrição ou impedimento que podem restringir uma pessoa, um grupo ou uma organização no alcance dos seus objetivos.

Na literatura, estão identificadas quatro dimensões ou áreas que caraterizam as barreiras à colaboração [18]:

a) **Dimensão organizacional e contextual**: inclui desafios relacionados com uma determinada organização, contexto ou tarefas;
b) **Dimensão social:** inclui desafios que se relacionam ao comportamento individual ou de um grupo;

c) **Dimensão tecnológica:** indica os desafios relacionados com as tecnologias e suas especificidades; e
d) **Dimensão cultural:** engloba desafios relacionados com as caraterísticas culturais únicas dos intervenientes nas ações de colaboração.

Para além destas dimensões, a literatura identifica a linguagem como um fator que pode representar uma barreira considerável à colaboração. As diferenças de linguagem dificultam a colaboração, quer num contexto internacional, devido à necessidade de traduções, quer na forma como os membros exprimem as suas ideias [19], [20]. A incorreta comunicação e interpretação de ideias traduz-se numa dificuldade a ter em consideração num ambiente colaborativo.

Na seção seguinte será abordada a modelação conceptual acrescida da componente colaborativa.

2.2.1 Modelação conceptual colaborativa

A colaboração é um dos requisitos para uma eficaz modelação conceptual e é considerada, na literatura, como uma forma importante para a difusão do conhecimento. Tal difusão chega a ser instantânea durante o processo de colaboração, em que os autores ou especialistas de vários domínios partilham o seu conhecimento e experiência [21].

Um processo de conceptualização colaborativo (*collaborative conceptualisation process* - CCP) consiste numa conceptualização partilhada, envolvendo um grupo de especialistas na criação de representações conceptuais. Comparado a um processo de conceptualização individual, o CCP acrescenta um conjunto de atividades sociais, como por exemplo, a negociação conceptual e a gestão prática das atividades no processo colaborativo [22].

O objetivo da negociação conceptual (apresentada com mais detalhe na seção seguinte) é obter a concordância, entre os especialistas, sobre os conceitos que farão parte do resultado final. A gestão prática das atividades consiste na introdução de mecanismos ou ferramentas capazes de apoiar os especialistas ao longo do processo colaborativo [22].

Os mapas conceptuais colaborativos (*collaborative concept mapping* - CCM) são o resultado de uma conceptualização colaborativa. O princípio para a criação dos CCM é comum ao apresentado por Novak [9] e [11], com a diferença de existirem vários especialistas a participar na seleção de conceitos e relações de um domínio.

Atualmente, o CCM é visto como uma ferramenta para apoiar a aprendizagem colaborativa. Maria et al. [15] apresentam um caso de estudo em que reforçam a importância da utilização de mapas de conceitos num ambiente colaborativo para criação de conhecimento. Para realçar a utilidade do CCM, os autores Basque e Lavoie [23] analisaram 39 estudos, desde 1980, e concluíram que existe uma evidência que os mapas conceptuais colaborativos, quando comparados com mapas conceptuais construídos individualmente, apresentam uma melhor qualidade de construção, beneficiando a criação, a partilha de conhecimento e a aprendizagem.

As barreiras apresentadas na seção anterior, inerentes às atividades colaborativas (seção 2.2, página 6), devem também requerer uma especial atenção quando se está a desenvolver modelos conceptuais de forma colaborativa. O envolvimento de vários especialistas na construção de modelos conceptuais deve ser auxiliado por uma gestão cautelosa das atividades.

As diferenças, tanto de conhecimento como da forma com que descrevem um domínio, dão origem a discrepâncias entre os especialistas na escolha dos conceitos a incluir num modelo conceptual. Desta forma, se por um lado existem vantagens ao inserir a componente colaborativa num processo de conceptualização, por outro lado, o esforço necessário para conseguir a concordância sobre os conceitos a utilizar pode aumentar.

Gao et al. [24] apresentam um caso de estudo sobre a criação de conhecimento num contexto de modelação conceptual colaborativa. O objetivo deste caso de estudo era validar a importância de uma abordagem colaborativa para representação de um domínio. Para tal, um grupo de alunos construiu um modelo conceptual de domínio, de forma colaborativa, ao longo de várias etapas.

Deste caso de estudo, os autores concluíram que a utilização de uma modelação colaborativa revelou-se produtiva, reforçando o aumento de qualidade na criação de conhecimento pela troca de ideias e experiências entre os alunos [24]. Ainda neste caso de estudo, foram detetadas várias atividades que envolveram um grande número de interações entre os alunos. A Tabela 1 apresenta o resultado da análise das interações entre os alunos na criação do modelo conceptual colaborativo [24].

Tabela 1: Análise das interações num problema real de conceptualização colaborativa [24]

Categoria	**Descrição**	**Número de interações**	**Percentagem**
Propor	Propostas de novos elementos (conceitos, ligações, rótulos e proposições) para o mapa de conceitos de grupo	28	11%
Refinar	Propostas de alterações ou revisões para qualquer elemento no mapa conceptual	8	3%
Colaborar	Negociar / discutir sobre os elementos adicionados ao mapa de conceitos	48	18%
Eliciar	Fazer perguntas para obter respostas relacionadas com o conteúdo	13	5%
Acordar	Expressar acordo sobre os elementos do mapa conceptual	37	14%
Elaborar/Esclarecer	Detalhar o que foi feito para os outros membros do grupo	56	22%
Responder sem elaborar	Resposta simples para uma pergunta ou uma eliciação sem fornecer explicação ou justificação	9	3%
Pedir confirmação	Garantir a compreensão exata do que foi proposto	26	10%
Discordar	Expressar desacordo com o que foi proposto	10	4%
Avaliar	Avaliar a qualidade das propostas ou revisões sugeridas	17	6%
Socializar	Outro tipo de atividades sociais; interações não relacionadas com qualquer conteúdo ou procedimento	10	4%

Da tabela anterior é possível observar um maior número de interações nas categorias colaborar, acordar e esclarecer. Face aos desafios apresentados na seção sobre colaboração (página 6), as discussões sobre a utilização de determinados conceitos são muito comuns. Por vezes, os especialistas têm de discutir sobre conceitos existentes no modelo conceptual com significados iguais.

A existência de conceitos iguais, descritos por termos diferentes, aumenta o número de interações e, consequentemente, aumenta a complexidade de um processo de conceptualização colaborativo. Para solucionar este problema, o desafio proposto nesta dissertação é desenvolver uma abordagem que permita detetar e integrar os conceitos similares existentes em modelos conceptuais e, assim, diminuir o esforço dos especialistas nas tarefas de negociação conceptual.

Na seção seguinte é apresentada a negociação conceptual como uma das atividades importantes numa conceptualização colaborativa. Serão estudadas formas para suportar e aumentar a qualidade de construção de modelos conceptuais num ambiente colaborativo.

2.3 Negociação conceptual

A criação colaborativa de artefactos semânticos (por exemplo: ontologias e taxonomias) é vista como um mecanismo sociotécnico, em que o significado é construído socialmente através da colaboração e negociação [1]. A ocorrência de dificuldades na comunicação entre os participantes aumenta o risco de surgirem problemas de inconsistências e duplicação de conceitos [25]. Tal como demonstrado, anteriormente, na Tabela 1, existe um elevado número de interações na atividade de negociar e acordar sobre os elementos que devem fazer parte da solução final.

Durante a negociação, os especialistas devem identificar os conceitos e as relações, e acordar quanto ao uso e representação desses elementos. Este processo é longo e árduo, devido aos seguintes fatores: a) os especialistas têm dificuldade em explicitar o seu conhecimento; b) em grandes repositórios de conhecimento, torna-se difícil acompanhar as modificações propostas, as razões que sustentam essas modificações e as suas repercussões no modelo conceptual final; c) há a necessidade de estabelecer consenso entre várias pessoas; e d) a informação e o seu significado são dependentes do tempo e do contexto [26].

A negociação conceptual, definida por Druckman [27], é uma das fases do processo de conceptualização colaborativa. Esta tarefa visa a obtenção de consenso, em cenários que denunciem situações de conflito e/ou ambiguidade no uso dos termos para representação de domínios específicos [28]. Como referido na seção da colaboração (seção 2.2, página 6), existem algumas barreiras que fazem aumentar a complexidade do processo de conceptualização colaborativa, *collaborative conceptualisation process* (CCP). Desta forma, torna-se necessária a existência de métodos para apoiar uma conceptualização colaborativa [2], [22] , [28], [29], [30] e [31].

Para o sucesso da aplicação de práticas colaborativas, deve ser desenvolvida uma base conceptual comum, onde as ontologias são fundamentais porque fornecem especificações formais da semântica partilhada. A formalização semântica é uma base sólida para o desenvolvimento de serviços e sistemas colaborativos, no entanto, em ambientes interorganizacionais, a existência de ontologias pré-existentes ou mal definidas aumentam a dificuldade no alinhamento de várias ontologias. Assim, é necessário a existência de um processo sociotécnico para o alinhamento de ontologias e para a negociação do significado [32].

Na seção seguinte, será abordado o método ColBlend, desenvolvido para apoiar a construção colaborativa de modelos conceptuais.

2.3.1 Método ColBlend

Baseado na teoria da integração de modelos conceptuais (*conceptual blending theory* – CBT) [33], Pereira [1] propôs o método ColBlend com o objetivo de apoiar o processo de conceptualização colaborativa (CCP). Em termos práticos, o ColBlend visa apoiar a construção colaborativa de um conjunto de modelos conceptuais acordados pelos especialistas, podendo os modelos desenvolvidos serem traduzidos em taxonomias, glossários ou ontologias. O funcionamento do método ColBlend é feito através de um processo que envolve explicação, discussão e negociação [2].

Assim como definido na CBT, o método ColBlend define três tipos de espaços de colaboração. Na Figura 3 estão representados os três espaços existentes na CBT e que também fazem parte do ColBlend.

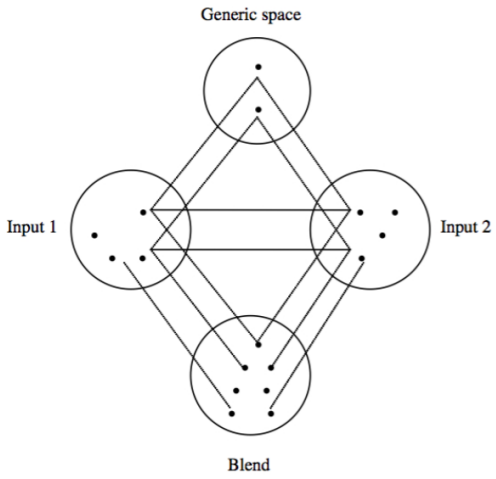


Figura 3: *Conceptualization blend theory* [33]

Os três espaços ilustrados na Figura 3 são:

a) ***input 1 e 2***: onde cada grupo ou organização elabora a sua proposta de conceptualização de um domínio;
b) ***blend***: espaço composto por todos os conceitos publicados e suscetíveis de negociação. Desta atividade podem surgir novos conceitos que serão publicados no *generic space*; e
c) ***generic space***: espaço onde é publicado o resultado da conceptualização partilhada.

O método ColBlend é composto pelos seguintes passos [28]:

Passo 0: de acordo com o contexto e objetivos definidos, um subgrupo de participantes elabora uma primeira proposta que será partilhada no espaço genérico e será usada como entrada no passo seguinte. Nesta primeira proposta é apresentado como resultado um mapa de conceitos com os conceitos chave do domínio.

Passo 1: é atribuído um ou mais espaços de entrada para cada grupo ou organização. Em cada espaço de entrada será proposto o mapa resultante da conceptualização privada.

Passo 2: neste passo, são realizadas operações automáticas com o objetivo de identificar elementos homólogos entre as propostas de cada espaço de entrada. Após a identificação de elementos homólogos, os resultados dos mapeamentos obtidos são apresentados para discussão no espaço *blend*.

Passo 3: todos os membros analisam a estrutura conceptual proposta para discussão e aceitam partes da proposta em discussão, com o mínimo de negociação. As partes aceites por todos são copiadas para o espaço genérico.

Passo 4: considerando os espaços de entrada, o contexto, os objetivos, o espaço genérico e a documentação disponibilizada nos espaços de entrada, é realizada uma "combinação" (*blend*) para obter novas propostas de conceptualização.

Passo 5: após a negociação das novas estruturas conceptuais propostas no espaço *blend*, os conceitos que obtiveram o consenso são copiados para o espaço genérico. Por outro lado, os conceitos que não foram validados no espaço *blend* serão eliminados. Neste passo, as estruturas de entrada podem sofrer alterações resultantes da projeção de elementos do espaço *blend*.

Passo 6: se ocorrem alterações nos espaços de entrada, o método deve voltar ao passo 2, não implicando a criação de uma nova combinação (*blend*).

Passo 7: após todos os participantes manifestarem o seu acordo com a conceptualização representada no espaço genérico, o processo é concluído.

A Figura 4 apresenta de uma forma genérica o método ColBlend.

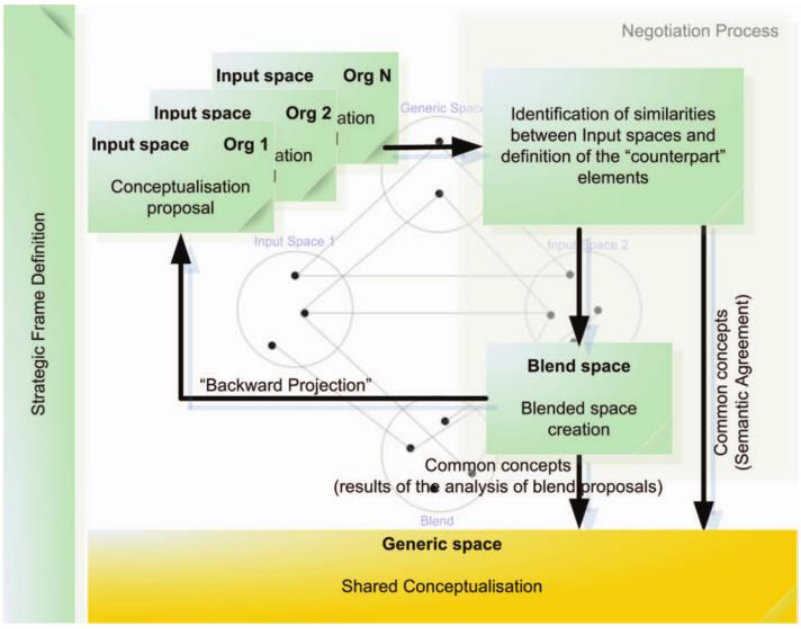


Figura 4: Método ColBlend para apoiar o processo de conceptualização colaborativo [28]

Como referido pelos autores Pereira et al. [28], a viabilidade do método ColBlend está dependente do apoio de uma ferramenta que facilite e controle todo o processo. O trabalho efetuado nesta dissertação pretende contribuir para a deteção automática de elementos homólogos, referida no passo 2 do método ColBlend, e assim facilitar a negociação conceptual num processo de conceptualização colaborativo.

Nas seções 2.4 e 2.5 seguintes, serão apresentadas, de forma geral, as ontologias e taxonomias como ferramentas utilizadas na organização da informação e que servem como apoio à similaridade semântica descrita na seção 2.6.

2.4 Ontologia

Nas últimas duas décadas, as ontologias têm sido alvo de um forte estudo por parte da comunidade científica. Tal estudo tem por base a transposição deste conceito do contexto filosófico para o contexto da Inteligência Artificial (IA) e dos Sistemas de Informação (SI) [6].

Enquanto no primeiro contexto, a ontologia refere-se a um sistema particular de categorias para uma determinada visão do mundo e, como tal, o sistema não depende de uma linguagem em particular, no contexto da IA e SI, uma ontologia refere-se a um artefacto de engenharia. Este artefacto é constituído por uma espécie de vocabulário, utilizado para representar uma determinada realidade, juntamente com um conjunto explícito de premissas, que descrevem o significado das palavras do vocabulário [6], [34].

De um modo geral, uma ontologia constitui um conjunto de termos representacionais, cujas definições associam os nomes das entidades (por exemplo, classes, relações, funções, ou outros objetos) com texto legível para os humanos, que descreve o que os nomes pretendem denotar, e com os axiomas formais, que restringem a interpretação e a utilização apropriada de tais termos [6], [34].

A necessidade de construção de uma ontologia pode ter vários fundamentos. Seguidamente, apresentam-se algumas das razões que levam à construção de uma ontologia [35]:

a) Partilhar uma compreensão comum de uma estrutura de informação entre um grupo de pessoas ou agentes de *software*;
b) Permitir a reutilização de um domínio de conhecimento;
c) Tornar explícitas determinadas premissas de um domínio;
d) Separar o domínio do conhecimento do conhecimento operacional; e
e) Analisar um domínio de conhecimento.

Os benefícios a retirar do uso das ontologias são vastos, e com base nos pontos supramencionados verifica-se que [35]:

a) a partilha de uma ontologia permite aos agentes computacionais a extração e a agregação da informação, a partir de variadas fontes de informação, de modo a responder a diferentes *queries* colocadas pelos utilizadores ou a servir de dados de entrada para outras aplicações;
b) a reutilização de um domínio de conhecimento possui vantagens na medida que a ontologia inclua um conjunto de noções de tempo, de pontos de interesse (para o domínio), entre

outros aspetos, de modo a que tal granularidade no conhecimento utilizado sirva de base para a sua reutilização ou integração em outros domínios[2];

c) a clarificação das premissas de um domínio possibilita a sua interpretação por parte de novos utilizadores, que têm de aprender o significado dos termos existentes no domínio;
d) a separação do domínio do conhecimento do domínio operacional permite que, por exemplo, uma ontologia que retrate o domínio da eletrónica possa ser utilizada para definir/caraterizar um computador como para definir a melhor configuração para elevadores. Ou seja, possuímos o mesmo domínio do conhecimento para dois domínios operacionais distintos; e
e) a análise de um domínio do conhecimento é possível após a existência de uma especificação declarativa dos termos. Uma análise formal dos termos é valiosa em casos de reutilização ou extensão de ontologias existentes.

2.5 Taxonomia

Enquanto as ontologias, como verificado anteriormente, se prendem com o estabelecimento de relações entre conceitos, de modo a retratar um determinado domínio, as taxonomias são mencionadas na literatura como ferramentas essenciais na organização da informação.

Uma taxonomia é um conjunto estruturado de nomes e descrições utilizados na organização da informação de uma forma consistente [36]. Emprega um vocabulário controlado em que cada termo, normalmente, possui relações hierárquicas, fazendo com que a taxonomia imponha uma estrutura tópica (no sentido do conceito principal para os sub-conceitos) à informação. Utiliza uma organização lógica e não tem em conta as necessidades específicas dos utilizadores para a tomada de decisão e ação [37].

Atualmente, entre os motivos para a utilização das taxonomias, Fouché [38] destaca:

a) **sobrecarga de informação**: a convergência das tecnologias de informação e comunicação resultaram no aumento da acessibilidade à informação e, consequentemente, na sua sobrecarga. Técnicas de organização da informação deverão ser aplicadas para melhorar o acesso e facilitar o controlo sobre os fluxos de informação;
b) **ascensão da *web***: a *web* oferece oportunidades para a organização da informação, apresentando desafios em relação à sua classificação; e
c) **aumento da utilização de tecnologias de gestão de informação não estruturada**: como parte da evolução da *web*, tem ocorrido uma maior aposta em tecnologias para a gestão da informação. O foco está em aumentar os benefícios destas tecnologias e fornecer uma infraestrutura de informação consistente que pode ser partilhada por diferentes aplicações.

[2] Alguns exemplos são: a biblioteca de ontologias *Ontolingua* (http://www.ksl.stanford.edu/software/ontolingua/) ou a DAML (http://www.daml.org/ontologies/).

Com o advento da Internet, houve um aumento na necessidade de utilização das taxonomias, devido ao crescimento exponencial da informação que circula na rede [39]. Neste âmbito, um dos principais benefícios que as taxonomias fornecem, está no melhoramento significativo em operações de pesquisa e recuperação de informação.

Num contexto mais restrito, como um ambiente organizacional, quando a informação está bem organizada e consistente, o tempo despendido em operações de pesquisa e navegação irá diminuir, contribuindo com o acesso rápido à informação por parte dos colaboradores [40]. Apesar dos benefícios em que as taxonomias se revelam mais-valias para as empresas, há que abordar alguns aspetos que não abonam tanto na sua utilização.

Quando uma taxonomia não apresenta no seu *design* aspetos de armazenamento e gestão, ou se não possibilita uma pesquisa acessível, todos os sistemas de gestão na organização se tornam praticamente inúteis. Aliado a este aspeto, surge a agravante das organizações não estarem dispostas a alocar os recursos necessários ao *design* de uma taxonomia. E quando o inverso acontece, os investimentos efetuados revelam-se insuficientes, inadequados ou inapropriados para realizar a categorização da informação [41].

Outro fator que constitui uma barreira à adoção das taxonomias diz respeito à sua (falta de) flexibilidade para se adaptar às alterações dentro das organizações. Daqui emerge a necessidade da decisão sobre a adoção de uma taxonomia rígida ou sobre o desenvolvimento de um conjunto de taxonomias que consigam abarcar várias áreas do conhecimento. Estas opções visam a prevenção de um excesso de rigidez da taxonomia, que irá prejudicar as ações de partilha ou cooperação [36], [42].

2.6 Similaridade semântica

A necessidade de determinar o nível de similaridade semântica é um problema que, atualmente, se encontra presente em grande parte da área da linguística computacional. Medidas de similaridade semântica são usadas em aplicações como desambiguação de sentidos, determinação de estrutura de discurso, resumo e anotação de texto, indexação automática e correção automática em processadores de texto [43].

A formalização e quantificação do conceito intuitivo de similaridade é um problema com uma longa história na filosofia, psicologia e até na área da inteligência artificial [43]. Similaridade é um conceito facilmente identificável no mundo real, em que uma pessoa consegue reconhecer sinónimos, mesmo que estes não tenham uma escrita semelhante, como por exemplo, roupa e vestuário, andar e caminhar, filme e película, ou automóvel e carro [44].

Uma definição formal, porém pouco precisa, de similaridade, geralmente atribuída a Leibniz, consiste em classificar duas expressões como sinónimas quando a substituição de uma pela outra, numa frase, não altere o sentido da mesma [45]. A similaridade léxica dos termos, isoladamente, não tem capacidade para identificar todas as possibilidades de aproximação dos termos, sendo por isso necessário encontrar abordagens que tenham em conta o aspeto semântico [46].

A semântica permite relacionar dois termos de diversas formas. De acordo com Mangold [47], para dois termos *a* e *b* diferentes, as mais importantes relações semânticas são:

a) *a* e *b* são sinónimos quando se referem ao mesmo conceito. Ex.: Automóvel e carro;
b) *a* e *b* são homónimos quando se referem a conceitos diferentes, tendo a mesma pronúncia e grafia. Ex.: Banco (banco de madeira) e Banco (instituição bancária);
c) *a* é hiperónimo de *b* se o conceito associado a *a* é mais geral que o conceito associado a *b*. Neste caso, *b* diz-se hipónimo de *a*. Ex.: Calçado (hiperónimo) e sapatos (hipónimo); e
d) *a* é merónimo de *b* se o conceito associado a *a* faz parte do conceito associado a *b*. Neste caso, *b* é holónimo de *a*. Ex: Dedo (merónimo) e mão (holónimo).

A similaridade semântica pode então ser definida como uma métrica associada a um conjunto de documentos ou termos, dentro de uma lista de termos, em que a noção de distância entre os diferentes termos se baseia na proximidade do seu significado ou do seu conteúdo semântico, e não na sua representação sintática. Esta métrica pode ser obtida definindo a similaridade topológica e usando, por exemplo, ontologias para determinar a distância entre palavras [45], [46]. Diferentes medidas de similaridade semântica serão discutidas no capítulo 3.1.

Seguidamente, apresentam-se algumas áreas de aplicação das medidas de similaridade semântica.

2.6.1 Áreas de aplicação

A similaridade semântica pode ser utilizada em diferentes áreas de aplicação, como por exemplo nas áreas de: Informática biomédica, Geoinformática, Linguística computacional e *Web* semântica.

2.6.1.1 Informática biomédica

Medidas de similaridade semântica podem ser aplicadas em ontologias biomédicas [48], [49], nomeadamente a Ontologia Genética, no inglês *Gene Ontology* (GO) [50], [51]. Estas medidas são usadas, principalmente, para a comparação de genes e proteínas, através da similaridade das suas funções e não da similaridade das suas sequências. Esta metodologia pode também ser aplicada a outras entidades, tais como: compostos químicos [52] e doenças [53].

Neste domínio, a similaridade pode ser obtida recorrendo a ferramentas disponíveis gratuitamente na internet:

a) **ProteInOn**: calcula valores de similaridade semântica baseados na GO para proteínas [54];
b) **CMPSim**: determina uma medida de similaridade funcional entre compostos químicos e caminhos metabólicos, usando medidas de similaridade semântica baseadas na ontologia de entidades moleculares e base de dados *Chemical Entities of Biological Interest* (ChEBI) [52]; e
c) **CESSM**: permite efetuar a avaliação automática de medidas de similaridade semântica baseadas na GO [55].

2.6.1.2 Geoinformática

Medidas de similaridade semântica também podem ser usadas para encontrar caraterísticas geográficas semelhantes [56]. Neste âmbito, existem diferentes ferramentas que podem ser utilizadas. Por exemplo, o servidor SIM-DL pode ser usado para calcular semelhanças entre conceitos guardados em ontologias de caraterísticas geográficas, por sua vez, a rede *Open Street Map* (OSM) pode ser usada para calcular a similaridade semântica de conceitos.

A Figura 5 demonstra um exemplo de uma ontologia que pode servir como recurso para o cálculo de similaridade semântica no domínio geográfico. Entre os conceitos representados estão, por exemplo, "edifício", "universidade" e "escola".

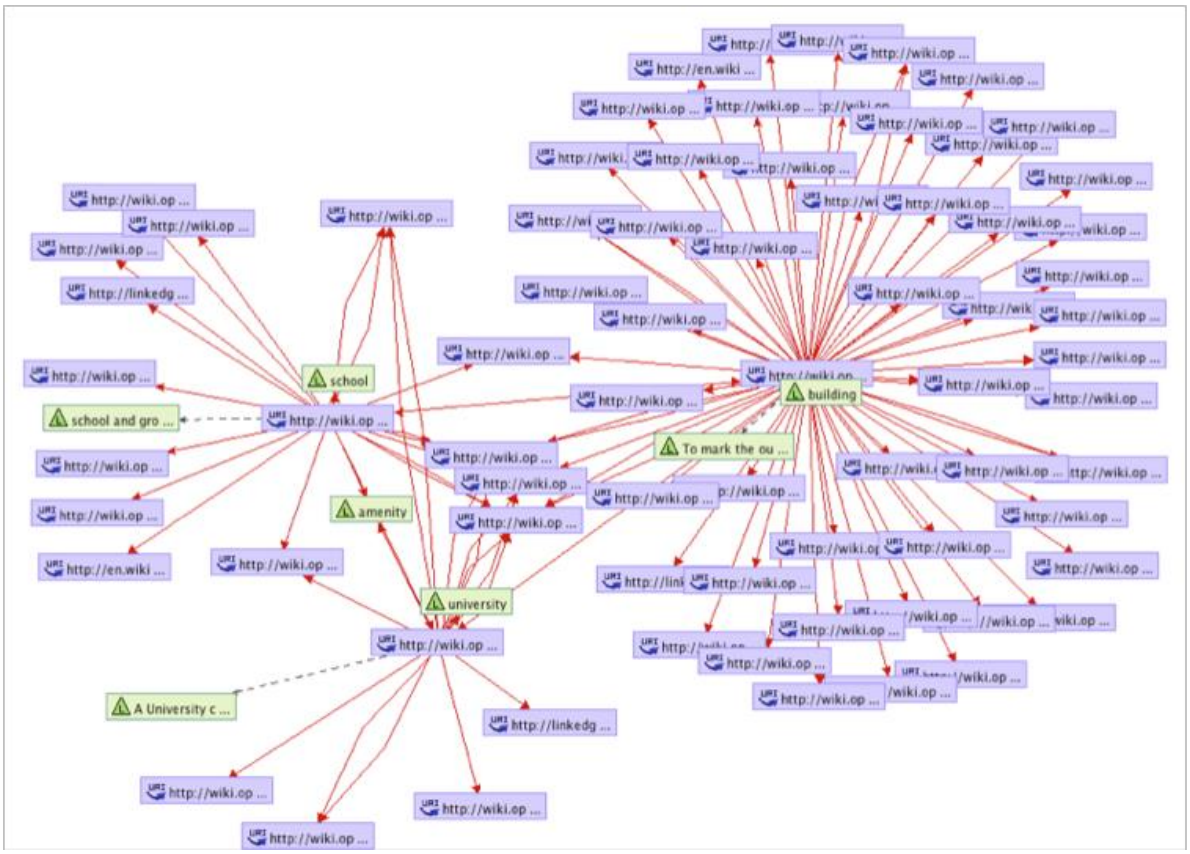


Figura 5: Subconjunto da rede semântica OSM[3] com a representação de conceitos

2.6.1.3 Linguística computacional

Como já foi referido anteriormente, a noção de similaridade semântica desempenha um papel de relevo na área da linguística computacional. Nesta área, o aparecimento de redes como a *Medical Subject Headings*[4] (MeSH) e o WordNet[5] foram de grande importância, pois tratam-se de bases de dados com definições de relações entre diversos conceitos. A Figura 6 demonstra um exemplo de conjunto de hiperónimos da palavra "gato".

[3] http://wiki.openstreetmap.org/wiki/File:Osn_snapshot.png
[4] https://www.nlm.nih.gov/mesh/meshhome.html
[5] ***WordNet*:** Dicionário léxico *online* desenvolvido pelo Laboratório de Ciências cognitivas de Princeton.

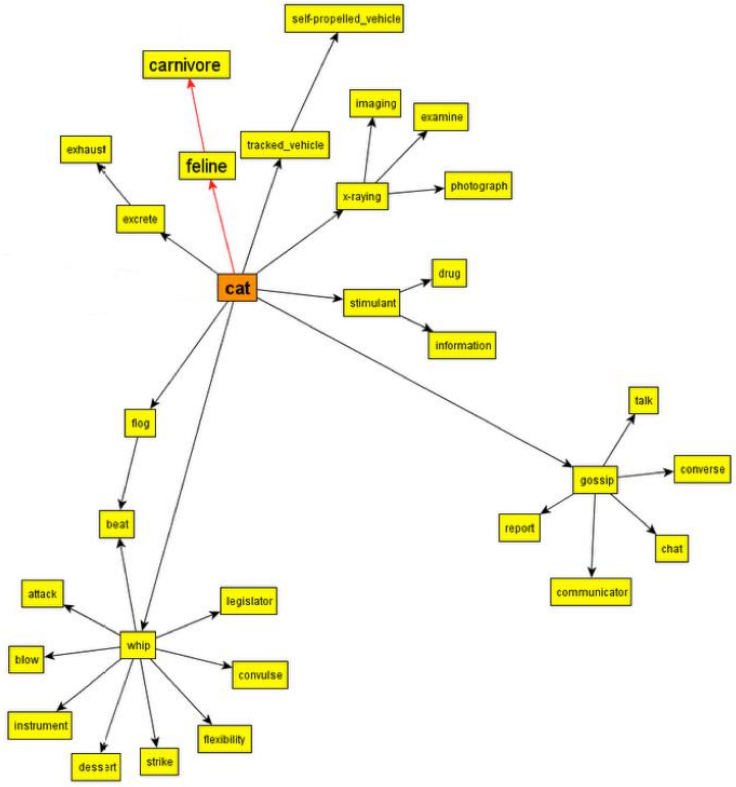


Figura 6: Exemplo de conjunto de hiperónimos da palavra "gato" obtido através do WordNet

2.6.1.4 Web Semântica

Tradicionalmente, a realização de pesquisas *online* procura por determinados termos nos documentos disponíveis. Contudo, o facto de um certo documento não conter a palavra ou a expressão exata que se procura, não significa que não seja de interesse para a pesquisa, sendo por isso importante que as pesquisas e a indexação sejam feitas recorrendo a medidas de similaridade semântica.

A *web* semântica, *semantic web,* é uma nova visão da *web,* cujo principal objetivo é tornar os conteúdos disponíveis *online* compreensíveis e processáveis, tanto por humanos como por máquinas. Este conceito foi introduzido por Tim Berners-Lee [57], que defendia que a maioria do conteúdo *web* foi concebido para ser lido por humanos, não para ser manipulado, significativamente, por máquinas. Os computadores são perfeitamente capazes de analisar informação em páginas *web* por disposição (*layout*) e estrutura (cabeçalho, links entre as páginas, etc.), mas, não têm nenhuma forma confiável de processar a semântica dessa informação.

A *web semântica* não é uma *web* separada, mas antes uma extensão da atual, na qual a informação tem um significado bem definido. Dispõe de extensões semânticas que permitem encontrar resultados semelhantes, a partir do seu conteúdo e não apenas recorrendo a descritores arbitrários, permitindo assim, melhorar a cooperação entre pessoas e computadores [58].

Seguidamente, serão apresentados os diferentes mecanismos para a interoperabilidade de ontologias.

2.6.2 Mecanismos de interoperabilidade de ontologias

Para que as diferentes fontes de informação, acessíveis na *web*, possam ser utilizadas por sistemas, é necessário que sejam compreendidas e processadas por eles. Dada a elevada heterogeneidade de conteúdos presentes nas fontes de informação, tanto a nível estrutural como semântico, deve-se proceder a uma compatibilização entre esses conteúdos [59].

No caso particular da *web* semântica, o uso de ontologias fornece uma teoria do domínio, bem como, uma representação comum para a troca de informação. Desta forma, diferentes agentes de *software*, que operem neste ambiente, podem obter suporte no processamento automático de informações. Contudo, é necessário implementar mecanismos que garantam a interoperabilidade de ontologias, de forma a garantir a troca automática de informação [60].

Os mecanismos utilizados para o processo de compatibilidade de ontologias são:

a) Combinação (*merge*);
b) Alinhamento (*matching*);
c) Mapeamento; e
d) Integração.

2.6.2.1 Combinação de ontologias (*Merge*)

Através da utilização da combinação de ontologias, o resultado é uma versão das ontologias originais combinadas numa ontologia única, com a união dos termos e sem haver uma definição clara de qual a sua ontologia original. Tipicamente, as ontologias usadas nesta situação aplicam-se a domínios similares, ou seja, tratam de assuntos semelhantes, ou de sobreposição[6] [60].

Na Figura 7, encontra-se um exemplo do mecanismo de combinação de ontologias, onde dois conceitos compatíveis – carro da ontologia *O1* e veículo da ontologia *O2* – são combinados dando origem a uma ontologia única.

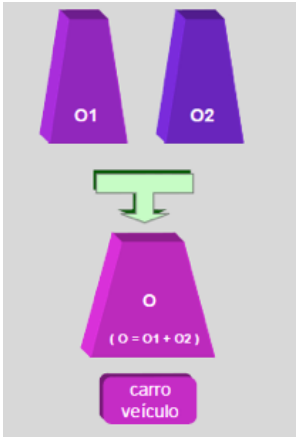


Figura 7: Combinação de ontologias [60]

[6] Define-se a sobreposição de um domínio A sobre um domínio B sempre que A possua, além de todas as informações contidas em B, também informações adicionais.

2.6.2.2 Alinhamento de ontologias (*Matching*)

Quando a duas ontologias, *O1* e *O2*, se aplica um alinhamento, o resultado da operação são as duas ontologias originais, nas quais são adicionadas ligações entre os seus termos equivalentes. Estas ligações permitem que as ontologias alinhadas reutilizem as informações entre si. Este processo é normalmente realizado quando as duas ontologias são de domínios complementares, ou seja, quando um dos domínios adiciona informações em relação ao outro [60].

Na Figura 8, é apresentado um exemplo de alinhamento de ontologias. Neste caso, o conceito carro da ontologia *O1* é alinhado ao conceito veículo da ontologia *O2*.

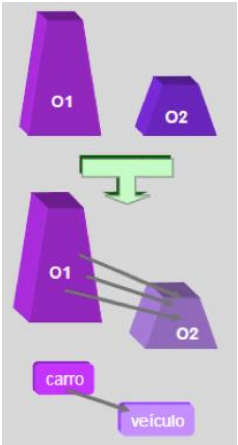


Figura 8: Alinhamento de ontologias [60]

2.6.2.3 Mapeamento de ontologias

A partir do mapeamento de ontologias obtém-se uma estrutura formal, com expressões que ligam os termos entre ontologias. Com este mecanismo, é possível transferir instâncias de dados, esquemas de integração e de combinação, bem como outras tarefas similares [59].

Na Figura 9, é demonstrado um exemplo do processo de mapeamento de ontologias, onde os conceitos carro e veículo, pertencentes às ontologias *O1* e *O2*, respetivamente, são mapeados por expressões formais.

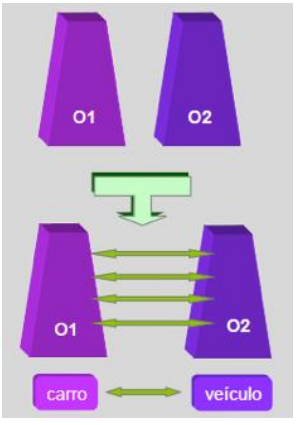


Figura 9: Mapeamento de ontologias [60]

2.6.2.4 Integração de ontologias

Na integração de ontologias obtém-se como resultado final uma ontologia única, criada a partir de montagem, extensão, especialização ou adaptação de outras ontologias de assuntos diferentes. Neste processo, é possível identificar as ontologias originais das diferentes partes constituintes da ontologia final.

A Figura 10 ilustra um exemplo do processo de integração de ontologias, onde os conceitos carro, veículo, automóvel e meio de transporte terrestre, pertencentes às ontologias *O1*, *O2*, *O3* e *O4*, respetivamente, são integrados, ou seja, unidos, formando uma ontologia única. Tanto neste mecanismo, como no mecanismo de combinação, o resultado final é uma única ontologia, obtida a partir da união dos termos das ontologias iniciais. Contudo, no caso da combinação, as ontologias originais tratam de assuntos similares, o que não acontece necessariamente no processo de integração [60].

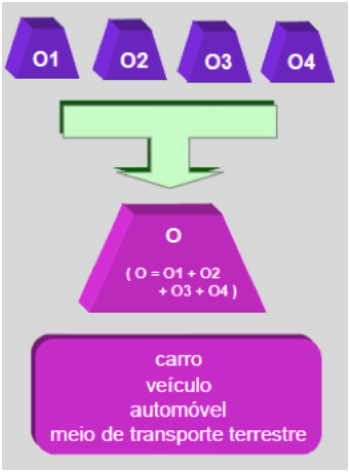


Figura 10: Integração de ontologias [60]

Nas seções 2.6.3 e 2.6.4 seguintes, serão apresentadas as diferentes abordagens, semânticas e sintáticas, para o cálculo de similaridade entre conceitos.

2.6.3 Abordagens semânticas para cálculo de similaridade

Existem diferentes abordagens para determinar a similaridade entre entidades com significado semântico, tais como conceitos (ex.: classes) que são representados em ontologias, ou elementos (ex.: recursos) que são representados em esquemas. As entidades a comparar apresentam propriedades associadas além do seu nome, como por exemplo, sob a forma de atributos, que permitem a implementação de métodos de comparação de similaridade semântica.

Os métodos semânticos têm em consideração o nível de generalidade (ou especificidade) de cada entidade, dentro da ontologia a que pertence, bem como a sua relação com outros conceitos [61]. É importante salientar que medidas de similaridade que se baseiem apenas em comparação de palavras-chave, *keywords*, não têm capacidade para usar a informação descrita acima (propriedades das entidades, nível de generalidade e relação com outros conceitos).

2.6.3.1 Baseadas em ontologias

Ontologias são ferramentas que podem ser usadas de forma bastante versátil na representação do conhecimento. Existem, por isso, diferentes abordagens propostas para a sua utilização na comparação de conceitos, numa ontologia, ou entre diferentes ontologias.

Ontologia Única

Neste caso, todas as fontes de informação estão ligadas a uma única ontologia, que fornece um vocabulário comum para especificação da semântica de cada entidade. Neste âmbito, uma abordagem proeminente é a chamada de SIMS [62], que inclui uma base de conhecimento hierárquica e terminológica, cujos os nós representam cada classe de objeto e as ligações entre os nós definem as relações entre objetos. Cada fonte de informação independente é descrita a partir do relacionamento dos seus objetos com o modelo de domínio global (*global domain model*).

A Figura 11 apresenta os passos para o cálculo de similaridade, utilizando apenas uma única ontologia.

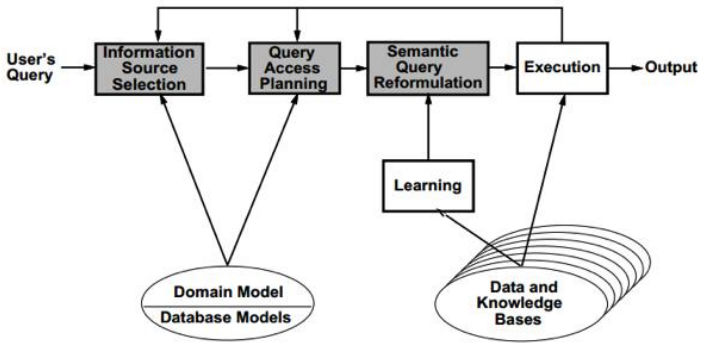


Figura 11: Arquitetura SIMS [62]

A abordagem de ontologia única é adequada em casos em que todas as fontes de informação partilhem, na sua maioria, conceitos de um domínio comum (por exemplo, em casos de conhecimento de senso comum, a base de conhecimento *WordNet* pode ser utilizada porque possui informação de conceitos de vários domínios). A comparação de um conceito com a ontologia traduz-se então na procura de conceitos semelhantes dentro da mesma [61].

Híbrida

Nesta segunda abordagem, a semântica de cada fonte é descrita pela sua própria ontologia, mas todas as ontologias são construídas tendo por base um vocabulário comum. Este vocabulário, contém os termos básicos, chamados de primitivas de um domínio, sobre os quais as ontologias são construídas para criar termos complexos, a partir da combinação de primitivas e operadores, tais como os operadores de união e interceção [61].

Dado que cada conceito de uma ontologia fonte pode ser descrito a partir do uso de primitivas, a comparação de conceitos pode ser feita recorrendo a métodos associados à abordagem de uma ontologia

única. Neste caso, a desvantagem é que não é possível recorrer de forma simples ao uso de ontologias existentes, tendo estas de serem redefinidas, visto que todas as ontologias fonte têm de estar associadas a um vocabulário partilhado [61].

Ontologias Múltiplas

Quando diferentes fontes de informação são descritas por diferentes ontologias é necessário recorrer a uma abordagem de ontologias múltiplas. O conhecimento dentro de cada fonte pode ser representado sem nenhuma referência a outras fontes de informação e às suas ontologias respetivas. Visto não necessitar de nenhum compromisso com uma ontologia comum, esta abordagem simplifica a introdução ou a modificação de informação nas fontes. Contudo, a falta de um vocabulário comum dificulta a comparação entre diferentes ontologias fonte [61].

A abordagem de ontologias múltiplas é a mais geral, mas é também a mais difícil de implementar, requerendo algoritmos de alta complexidade. A abordagem mais direta é a conversão de todas as fontes de informação numa ontologia comum, que pode ser armazenada. Esta alternativa requer grandes esforços humanos e não é facilmente extensível a alterações nas fontes de informação [61].

Os autores Mena et al. [63] apresentam uma tentativa de fornecer semântica intuitiva para o mapeamento entre conceitos de diferentes ontologias, recorrendo a relações retiradas da linguística para relacionar termos em várias ontologias. De um modo geral, o mapeamento da ontologia identifica termos semanticamente correspondentes, de diferentes ontologias fonte, e tem ainda de considerar diferentes visões do domínio, como por exemplo, diferente agregação dos conceitos da ontologia.

Uma boa alternativa para comparar conceitos de diferentes ontologias é fazer o mapeamento de esquemas, encontrando correspondências semânticas de atributos ou instâncias, em fontes heterógenas (abordagem comum para a comunidade de bases de dados). A abordagem fundamental para este caso é o chamado *matching*, que recebe dois esquemas de entrada (*input*) e produz um mapeamento entre elementos que correspondem semanticamente entre si.

A técnica de correspondência de esquemas pode ser dividida em duas categorias: *label-based* e *instance-based*, dependendo da diferente informação em que se baseiam. No caso de *label-based*, o mapeamento é feito recorrendo apenas à similaridade entre definições ou rótulos (*labels*) dos atributos de duas fontes de informação. Os métodos *instance-based* usam a sobreposição de conteúdos e as propriedades estatísticas para determinar a similaridade de dois conceitos [61].

Um método de unificação, baseado em afinidade para a construção de um mapeamento global para diferentes ontologias, foi proposto pelos autores Castano et al. [64], em que o conceito de afinidade é usado para identificar termos com relações semânticas em diferentes ontologias. Os termos presentes em diferentes ontologias são analisados e identificados, caso apresentem relações semânticas usando um processo de agrupamento hierárquico. A integração de todas as ontologias é conseguida recorrendo a esses agrupamentos.

2.6.3.2 Baseadas no conteúdo de informação

Nesta categoria, as medidas baseiam-se no conteúdo de informação, *information content* (IC), de cada conceito. A noção de conteúdo de informação está intimamente ligada à frequência de um termo, numa dada coleção de documentos. Neste caso, a chave para similaridade entre dois conceitos é a extensão da informação partilhada entre ambos, indicada por um conceito altamente específico que os agrupe [61].

Resnik [65] demonstra como podem ser definidas as probabilidades associadas a conceitos de uma taxonomia. Seja uma taxonomia aumentada a partir da função $p : C \rightarrow [0,1]$, de tal forma que $c \in C$, então $p(c)$ é a probabilidade de encontrar uma ocorrência do conceito *c*. A probabilidade $p(c)$ é então definida como $p(c) = freq(c)/N$, onde *N* é o número total de termos da taxonomia e $freq(c) = \sum_{n \in words(c)} n$ e *words(c)* é o conjunto de termos agrupados por *c*.

A partir desta definição, observa-se que se c_1 *is-a* c_2 então $p(c_1) \leq p(c_2)$, o que intuitivamente significa que quanto mais geral for um conceito maior será a sua probabilidade. Assim, o conteúdo de informação de um conceito *c* pode ser quantificado utilizando o logaritmo da probabilidade $\ln p(c)$, ou seja, o aumento da probabilidade de um conceito implica o decréscimo da sua informação (por exemplo, quanto mais abstrato um conceito, menos informação o mesmo contém).

Dada a probabilidade definida acima, várias medidas de similaridade semântica foram definidas. A fórmula (1) representa as medidas que se baseiam no IC das raízes comuns dos termos c_1 e c_2, onde $S(c_1, c_2)$ é o conjunto de conceitos que agrupam c_1 e c_2. Visto poderem existir várias raízes para cada conceito, dois conceitos podem partilhar raízes com caminhos diferentes. Nestes casos, define-se a probabilidade mínima $p(c)$ quando existe mais do que uma raiz partilhada, e *c* passa a ser denominado o agrupador mais informativo (*most informative subsumer*) [61].

$$p_{mis}(c_1, c_2) = min_{c \in S(c_1, c_2)}\{p(c)\} \quad (1)$$

A Figura 12 representa os conceitos introduzidos acima. No caso apresentado, os termos *coin*, *cash*, etc. são todos os membros de *S(nickel, dime)*, mas o termo que estruturalmente é o limite mínimo superior é o *coin*, e será também o agrupador mais informativo. O IC do agrupador mais informativo será usado na quantificação da similaridade de dois conceitos.

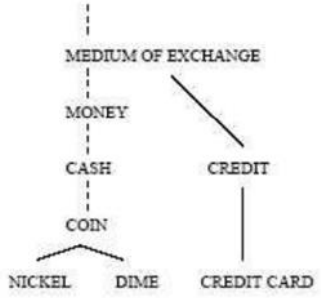


Figura 12: Excerto da taxonomia *WordNet* [61]

2.6.3.3 Baseadas em caraterísticas

Nesta abordagem não se considera a posição dos termos numa taxonomia, sendo apenas considerado o conjunto de caraterísticas que se referem à palavra desejada. O grau de similaridade é tanto maior quanto mais caraterísticas os termos tiverem em comum, ou seja, com este método, o cálculo da similaridade semântica depende da combinação das várias caraterísticas associadas a cada termo [44].

Algumas das medidas que utilizam esta abordagem introduzem o conceito de relacionamento semântico, baseado num vetor de contextos extraídos do *corpus*[7], onde um conceito *c1* pode ser representado por um vetor de contextos. Com esta técnica, é possível construir uma matriz de coocorrência, ou seja, a ocorrência de duas ou mais palavras, simultaneamente, num texto ou base de dados [44]. A cada célula da matriz está associada uma pontuação, que é calculada a partir da similaridade entre o termo encontrado na descrição do conceito, e cada palavra que coocorre no *corpus*.

Numa matriz deste tipo, as linhas correspondem aos termos usados na descrição do conceito. Após a criação de todos os vetores de contexto, os conceitos podem ser representados por palavras descritivas, que são compostas por uma média de todos os vetores associados às palavras [44].

2.6.3.4 Híbridas

Além das abordagens descritas anteriormente, é ainda possível proceder-se à determinação de uma medida de similaridade semântica, combinando os diferentes tipos de abordagem. A maioria das medidas que se enquadram nesta abordagem, procedem à comparação de dois conceitos *c1* e *c2*, combinando a distância em que os termos se encontram numa taxonomia e as caraterísticas associadas a cada termo [61].

[7] ***Corpus*:** é um grande e estruturado conjunto de textos, usado para análises estatísticas, verificação de ocorrências e validação de regras linguísticas num determinado domínio [112].

2.6.3.5 Resumo comparativo

Seguidamente, na Tabela 2, apresenta-se, resumidamente, os princípios, as vantagens e as desvantagens das abordagens descritas anteriormente.

Tabela 2: Comparativo dos diferentes tipos de abordagens apresentadas

Medidas	Princípio	Vantagens	Desvantagens
Baseadas em ontologias	Comprimento do caminho que separa os conceitos e a posição dos mesmos na taxonomia	Aplicação simples	- Apenas aplicável a relações *is-a* - Necessidade de taxonomia - Similaridade igual para conceitos com distâncias e profundidades iguais na taxonomia
Baseadas em IC	Quanto mais informação dois conceitos partilharem maior a similaridade entre eles	Utiliza mais informação além do próprio conceito	Pares de conceitos com similaridade igual pela soma do IC de outros pares
Baseadas em caraterísticas	Quanto mais caraterísticas comuns e menos caraterísticas não comuns, mais semelhantes são os termos	Considera caraterísticas adicionais aos conceitos	- Elevada complexidade computacional - Necessita de um conjunto completo de caraterísticas
Híbridas	Combinam propriedades e princípios das categorias anteriores	Melhor qualidade de distinção entre os conceitos	Necessário introduzir pesos ou parâmetros para definir as medidas mais importantes, caso contrário origina desvios

2.6.4 Abordagens sintáticas para cálculo de similaridade

Ao contrário das abordagens semânticas, em que a comparação entre conceitos se faz recorrendo à sua definição, atributos e caraterísticas associadas, na abordagem sintática esses fatores não têm influência. Sintaticamente, o cálculo da similaridade entre conceitos faz-se por comparação dos elementos constituintes do conceito em questão, nomeadamente, os seus carateres ou conjuntos de carateres.

Na comparação de duas palavras (*strings*) tem-se em consideração que operações (inserção, substituição e remoção de carateres) têm de ser aplicadas para se obter a *string* alvo, a partir da *string* inicial. Tipicamente, cada operação tem um peso associado e a distância entre conceitos é obtida pela soma dos pesos das operações a efetuar. Quanto maior for o número de operações efetuadas mais distantes as palavras são [66].

O capítulo seguinte abordará a revisão do estado da arte, composta pelas medidas de similaridade, pelas *frameworks* e pelas ferramentas para o cálculo de similaridade.

3 REVISÃO DO ESTADO-DA-ARTE

Neste capítulo serão apresentadas as diferentes alternativas para determinar a similaridade entre conceitos. A similaridade entre conceitos e/ou modelos pode ser obtida através de:

a) **Medidas de similaridade:** A utilização de medidas de similaridade permite obter o grau de similaridade entre dois conceitos;
b) ***Frameworks*:** Implementam medidas de similaridade e disponibilizam uma API para utilização em ferramentas; e
c) **Ferramentas:** Programas especificamente criados para permitirem a integração de dois modelos, após análises de similaridade entre conceitos. De acordo com as abordagens de cada ferramenta, podem ser utilizadas diferentes medidas e/ou *frameworks*.

3.1 Medidas de similaridade

As medidas de similaridade encontram-se divididas em dois grandes grupos: medidas semânticas e medidas sintáticas. O grau de similaridade obtido por cada medida representa uma indicação de quão similares são dois conceitos, sendo que, quanto maior for o grau de similaridade, maior a probabilidade de dois conceitos terem o mesmo significado.

3.1.1 Medidas semânticas

As medidas semânticas focam-se no significado de cada conceito para calcular o grau de similaridade. Neste tipo de medidas são utilizados recursos externos (por exemplo: ontologias, taxonomias ou *corpus*) como complemento aos termos que designam os conceitos em análise. Os autores Abdelrahman e Kayed [67] dividem as medidas semânticas em quatro categorias: as que se baseiam em ontologias; as que se baseiam na quantidade de informação que dois conceitos partilham; as que se baseiam nas caraterísticas de cada conceito; e as que se baseiam em combinações das opções anteriores, ou seja, híbridas.

3.1.1.1 Medidas baseadas em ontologias

Esta categoria abrange todos os métodos que usem recursos ou fontes de conhecimento (tais como: ontologias, dicionários ou vocabulário) no cálculo do grau de similaridade semântica entre conceitos. Estes métodos têm na sua base estruturas em redes ou grafos, utilizando as variáveis caminho (*path*) ou distância (*length*). Este tipo de abordagem recorre a taxonomias do tipo *is-a* para definir relações de subclasses e superclasses entre os diferentes conceitos. A Figura 13 demonstra um tipo de taxonomia simples.

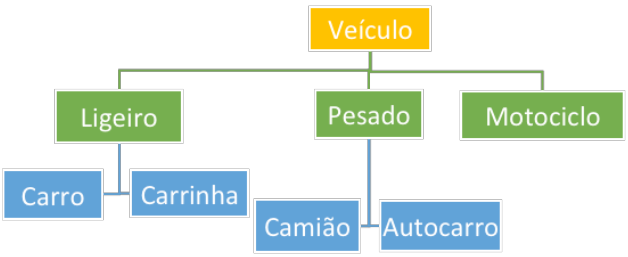


Figura 13: Exemplo de uma taxonomia simples

Se considerarmos os conceitos *c1* e *c2*, presentes numa taxonomia, as medidas vão verificar quais as posições de cada conceito nessa taxonomia. As medidas que serão apresentadas a seguir têm por base uma versão simplificada da teoria de ativação por propagação, *spreading activation theory*. Esta teoria assume que a hierarquia de conceitos se encontra organizada a partir de linhas de similaridade semântica, ou seja, quanto mais similares são dois conceitos, mais próximos estes devem encontrar-se e, quanto mais ligações existirem entre os conceitos, mais estes se encontram relacionados [61].

3.1.1.1.1 Caminho mais curto (*Shortest Path*)

A primeira medida a ser apresentada está relacionada com quão próximos dois conceitos se encontram numa taxonomia [68]. A medida *shortest path* é definida pela fórmula (2).

$$sim_{sp}(c_1, c_2) = 2MAX - L \tag{2}$$

Na fórmula (2), *MAX* é a distância máxima entre dois conceitos na taxonomia e *L* é o número mínimo de ligações entre os conceitos *c1* e *c2*. Esta medida foi desenhada para funcionar com hierarquias. A sua motivação reside em duas observações: o comportamento da distância conceptual assemelha-se a uma métrica e a distância conceptual, entre dois conceitos, é frequentemente proporcional ao número de ligações que os separam na hierarquia.

Uma medida deste género pode ser implementada num sistema de recuperação de informação que se baseie na indexação de documentos e em ações de termos de uma hierarquia semântica.

3.1.1.1.2 Ligações pesadas ou ponderadas

Esta medida surge como uma extensão da medida apresentada anteriormente [69]. O peso de uma ligação pode ser afetado por:

a) Densidade da taxonomia no ponto considerado;
b) Profundidade na hierarquia; e
c) Força da conotação[8] entre os nós (conceitos) pais e filhos.

[8] A conotação de um termo é uma listagem de condições para a denotação, enquanto a denotação é o conjunto de coisas a que o termo pode ser corretamente aplicado. Por exemplo, a conotação da palavra "quadrado" é "retangular e equilátero", enquanto a sua denotação são todos os símbolos geométricos com essas caraterísticas [43].

Desta forma, calcular a distância entre dois conceitos consiste na soma dos pesos de cada ligação entre eles, em vez da sua simples contagem.

3.1.1.1.3 Hirst e St-Onge

Esta medida baseia-se na ideia que dois conceitos, *c1* e *c2*, são semanticamente semelhantes se estiverem ligados por um caminho que não seja demasiado longo, e que não mude de direção com muita frequência [70]. A medida Hirst e St-Onge é definida pela fórmula (3).

$$sim_{H\&S}(c_1, c_2) = C - L - kd \tag{3}$$

Na fórmula (3), *d* corresponde ao número de mudanças de direção, *L* representa o número de ligações entre os conceitos, e *C* e *k* são constantes. Apesar desta medida introduzir uma nova perspetiva acerca da similaridade entre dois conceitos, a sua performance parece ser fraca, principalmente devido ao facto de se basear na tendência para dispersar e não nas relações entre os conceitos.

3.1.1.1.4 Wu e Palmer

Esta medida de similaridade é representada pela fórmula (4) e considera as posições dos conceitos *c1* e *c2*, na taxonomia, em relação à posição do conceito específico mais comum *c* [71]. Visto poderem existir múltiplas raízes para cada conceito, dois conceitos podem partilhar raízes a partir de diversos caminhos. O conceito específico mais comum *c* é a raiz comum que se relaciona com os conceitos *c1* e *c2* com o mínimo de ligações *is-a*.

$$sim_{W\&P}(c_1, c_2) = \frac{2H}{N_1 + N_2 + 2H} \tag{4}$$

Na fórmula (4), *N1* e *N2* são os números de ligações *is-a*, a partir dos conceitos *c1* e *c2*, respetivamente, até ao conceito *c*, e *H* é o número de ligações *is-a*, desde *c* até à raiz da taxonomia. Esta medida apresenta valores entre 1 (para conceitos similares) e 0 (para conceitos sem similaridade).

3.1.1.1.5 Li et al.

A medida Li et al. foi derivada, intuitiva e empiricamente, da medida anterior. Combina o *L* (percurso mais curto entre os conceitos *c1* e c2), e *H* (profundidade do conceito específico mais comum *c*), a partir de uma função não linear [72]. A medida Li et al. é definida pela fórmula (5).

$$sim_{Li}(c_1, c_2) = e^{-\alpha L} \frac{\left(e^{\beta H} - e^{-\beta H}\right)}{e^{\beta H} + e^{-\beta H}} \tag{5}$$

Na fórmula (5), os parâmetros $\alpha \geq 0$ e $\beta > 0$ representam a escala do comprimento do percurso mais curto e a escala da profundidade, respetivamente. A motivação desta medida reside no facto de as fontes de informação serem praticamente infinitas, enquanto os humanos comparam a semelhança entre

os conceitos, dentro de um intervalo finito ("completamente semelhante" ou "nada em comum"). Intuitivamente, a transformação de um intervalo infinito para um intervalo finito não é linear. Esta medida, tal como a anterior, tem valores entre 1 e 0, sendo 1 associado a conceitos semelhantes e 0 a conceitos não semelhantes.

3.1.1.1.6 Leacock e Chodorow

A medida proposta por Leacock e Chodorow [73] baseia-se no comprimento do caminho entre dois conceitos, c_1 e c_2, numa taxonomia do tipo *is-a*. Considera-se o caminho mais curto como aquele que inclui o menor número de conceitos intermédios na transição entre os dois conceitos a comparar. Nesta medida, a profundidade da taxonomia *D* é também levada em consideração, e é definida como a maior distância a percorrer desde um nó externo até à raiz da taxonomia [65]. A medida proposta é definida na fórmula (6).

$$sim_{LCH}(c_1, c_2) = \max\left[-\log\frac{dist(c_1, c_2)}{2D}\right] \quad (6)$$

Na fórmula (6), $dist(c_1, c_2)$ é a menor distância entre os termos c_1 e c_2, ou seja, a distância associada ao caminho que compreende o menor número de nós, e *D* a profundidade da taxonomia.

3.1.1.1.7 Lesk

O algoritmo de Lesk, proposto originalmente em 1985, tem como objetivo esclarecer o sentido de palavras em frases relativamente curtas [74]. De acordo com este método, duas palavras estão tão mais relacionadas quanto maior for a correspondência entre as suas definições. Em 2002, este método foi expandido para poder usar o *WordNet* como dicionário para obter as definições das palavras em questão [75].

Dada uma determinada palavra, cujo significado necessita ser clarificado, o algoritmo compara a sua definição com as definições de cada uma das restantes palavras da frase, e determina o significado correto, como sendo aquele em que há maior correlação com os significados das restantes palavras. O algoritmo repete o processo para as restantes palavras, não utilizando as definições já determinadas para as palavras anteriores [75].

3.1.1.2 Medidas baseadas em IC

Seguidamente, serão apresentadas algumas medidas de determinação de similaridade semântica, baseadas no conteúdo de informação partilhado (IC).

3.1.1.2.1 Lord et al.

Nesta medida, a forma de comparar dois conceitos consiste em usar a probabilidade da raiz partilhada mais específica [76]. A medida Lord et al. é definida pela fórmula (7).

$$sim_{Lord}(c_1, c_2) = 1 - p_{min} \tag{7}$$

A fórmula (7) resulta em valores entre 1 (para conceitos muito similares) e 0 (para conceitos não similares). É usada para verificar o quanto a determinação de similaridade é sensível apenas à frequência de informação partilhada.

3.1.1.2.2 Resnik

A medida Resnik, definida na fórmula (8), usa o IC de raízes partilhadas [65].

$$sim_{Resnik}(c_1, c_2) = -\ln p_{min} \tag{8}$$

Esta medida implica que quanto mais informação dois termos tiverem em comum, mais semelhantes serão, e a informação partilhada entre ambos é indicada pelo IC do termo que os agrupa na taxonomia. Visto p_{min} poder variar entre 0 e 1, esta medida de similaridade resulta em valores entre infinito (para termos semelhantes) e 0 (para termos não semelhantes).

Na prática, se N for o número de termos numa taxonomia, o maior valor que esta medida pode resultar é igual a $-\ln 1/N = \ln N$, ou seja, esta medida indica o tamanho do *corpus*.

Nesta medida, a comparação de um conceito com ele próprio, depende da sua localização na taxonomia. Apresenta valores mais elevados para os conceitos menos frequentes e, portanto, revela informação sobre a utilização de um dado conceito no *corpus* em questão.

3.1.1.2.3 Lin

Nesta medida, são levadas em consideração, tanto a quantidade de informação utilizada para definir o que dois termos têm em comum, como a quantidade de informação necessária para os descrever [77]. A medida Lin é definida na fórmula (9).

$$sim_{Lin}(c_1, c_2) = \frac{2 \ln p_{min}(c_1, c_2)}{\ln p(c_1) + \ln p(c_2)} \tag{9}$$

Na fórmula (9), visto que $p_{min} \geq p(c_1)$ e $p_{min} \geq p(c_2)$, os valores obtidos podem variar entre 1 (para conceitos semelhantes) e 0 (para conceitos não semelhantes). Neste caso, a comparação de um conceito com ele próprio implica um resultado 1, por definição.

3.1.1.2.4 Jiang et al.

Esta medida recorre à distância semântica [78] e é definida pela fórmula (10).

$$dist_{Jiang}(c_1, c_2) = -2 \ln p_{min}(c_1, c_2) - (\ln p(c_1) + \ln p(c_2)) \tag{10}$$

A partir da fórmula (10), a similaridade entre dois conceitos, $sim_{Jiang}(c_1, c_2)$, é calculada pela expressão $sim_{Jiang}(c_1, c_2) = 1 - dist_{Jiang}(c_1, c_2)$. Esta medida pode resultar em altos valores, tal

como acontece na medida de Resnik, mas na prática tem o limite de $2 \ln N$, sendo N o tamanho do *corpus*.

Esta medida, tal como na medida de Lin, também combina o IC de raízes comuns e dos conceitos comparados. Conclui-se, então, que esta medida combina propriedades de algumas medidas apresentadas anteriormente.

3.1.1.3 Medidas baseadas em caraterísticas

Nas medidas apresentadas anteriormente, as caraterísticas dos conceitos não foram levadas em consideração. Contudo, essas caraterísticas contêm informação valiosa, relativa ao conhecimento de um conceito.

A medida apresentada de seguida considera as caraterísticas dos conceitos, de modo a calcular a similaridade entre eles, ignorando a sua posição na taxonomia e os seus diferentes conteúdos de informação.

3.1.1.3.1 Tversky

Esta medida baseia-se nos conjuntos descritivos dos conceitos. Cada conceito é descrito por um conjunto de palavras que indicam as suas propriedades ou caraterísticas. Então, quanto mais caraterísticas em comum e menos caraterísticas não comuns dois conceitos tiverem, mais semelhantes estes serão [79]. A fórmula (11) define a medida de Tversky.

$$sim_{Tversky}(c_1, c_2) = \frac{|C_1 \cap C_2|}{|C_1 \cap C_2| + k|C_1/C_2| + (k-1)|C_2/C_1|} \tag{11}$$

Na fórmula (11), C_1 e C_2 correspondem aos conjuntos descritivos dos conceitos c_1 e c_2, respetivamente, e $k \in [0,1]$ define a importância relativa das caraterísticas não comuns. Esta medida varia entre 1 (para conceitos semelhantes) e 0 (para conceitos não semelhantes).

A determinação do valor de k baseia-se na observação de que a similaridade não é necessariamente uma relação simétrica. As caraterísticas comuns entre uma subclasse e a sua superclasse têm uma contribuição maior para a avaliação da similaridade do que as caraterísticas comuns na direção inversa. Esta hipótese permite uma abordagem sistemática para a determinação da assimetria de uma medida de similaridade.

3.1.1.4 Medidas com abordagens híbridas

Este tipo de medidas de comparação de dois conceitos, c_1 e c_2, combina algumas das medidas apresentadas anteriormente, considerando o caminho que liga os conceitos numa taxonomia, as ligações *is-a* dos conceitos às suas raízes no grafo e as caraterísticas dos conceitos.

3.1.1.4.1 Rodriguez et al.

Esta medida de similaridade sugere um método de identificação das caraterísticas distintas dos conceitos, além da classificação usual das caraterísticas como atributos, sendo classificadas como funções, partes e atributos. As funções representam o que é feito com a ocorrência de uma classe, as partes são elementos estruturais e os atributos são caraterísticas adicionais. Por exemplo, se considerarmos o conceito "universidade", uma função é "educação", as partes podem ser "telhado" ou "chão" e os atributos podem ser "propriedades arquitetónicas".

Considerando todas as caraterísticas distintas de um conceito, a função de similaridade global é a soma ponderada (ou pesada) dos valores de similaridade para funções, partes e atributos, onde ω_f, ω_p e ω_a são os pesos correspondentes [80]. A medida Rodriguez et al. é definida pela fórmula (12).

$$sim_{Rodriguez}(c_1, c_2) = \omega_p . S_p(c_1, c_2) + \omega_f . S_f(c_1, c_2) + \omega_a . S_a(c_1, c_2) \tag{12}$$

Na fórmula (12), todos os pesos são maiores ou iguais a zero e $\omega_p + \omega_f + \omega_a = 1$. Para cada tipo de caraterística distinta, $S_p(c_1, c_2)$, $S_f(c_1, c_2)$ e $S_a(c_1, c_2)$, usa-se uma função de similaridade baseada no modelo de correspondência de caraterísticas de Tversky, apresentada anteriormente.

Tal como foi apresentado na medida proposta por Tversky, esta medida também recorre ao uso do número de caraterísticas comuns e não comuns, entre dois conceitos, para determinar a sua semelhança. Contudo, esta medida difere da abordagem de Tversky no seguinte:

a) considera uma classificação atípica das caraterísticas como funções, partes e atributos; e
b) define *k* a partir da distância entre conceitos na hierarquia, ou seja, a função *k* relaciona a distância entre os conceitos c_1 e c_2 e o agrupador mais informativo c_{min}, fórmula (13).

$$k(c_1, c_2) = \begin{cases} \dfrac{d(c_1, c_{min})}{d(c_1, c_2)} & , d(c_1, c_{min}) \leq d(c_2, c_{min}) \\ 1 - \dfrac{d(c_1, c_{min})}{d(c_1, c_2)} & , d(c_1, c_{min}) > d(c_2, c_{min}) \end{cases} \tag{13}$$

Sendo $d(c_1, c_2) = d(c_1, c_{min}) + d(c_2, c_{min})$.

3.1.1.4.2 Knappe

Esta medida baseia-se no facto de que podem existir múltiplos caminhos a ligar dois conceitos. A consideração de todos os caminhos possíveis implica num aumento considerável da complexidade da abordagem. Portanto, a ideia geral dá ênfase aos conceitos partilhados e pode ser definida uma medida de similaridade, que represente a parte da ontologia que contém os conceitos comparados [81]. Nesta abordagem, também é considerado o facto de conceitos complexos poderem ser definidos como sendo constituídos por um ou mais termos.

Inicialmente, a decomposição $\tau(c)$ do conceito c, num conjunto *C*, é definida e é feita a expansão ascendente $\varpi(c)$ de *C*. A decomposição de c é definida como o conjunto de todos os termos e atributos

que compõem o conceito c (no caso de um conceito conter apenas um termo, inclui-se apenas esse termo, ou seja, o termo c).

Considerando $u(c)$ como sendo todos os nós acessíveis acima de c, ou seja, $u(c) = \varpi(\tau(c))$, podemos então definir as direções ascendente e descendente no grafo como generalização e especialização, respetivamente. Três propriedades deverão ser consideradas no momento da definição da função de similaridade [81]:

a) O custo de generalização deverá ser maior do que o custo de especialização, ou seja, a função de similaridade não deverá ser simétrica;
b) O custo de atravessar ligações deverá ser menor quando os nós (conceitos) forem mais específicos; e
c) A maior especialização deverá estar associada a menor similaridade.

A medida proposta por Knappe traduz-se pela fórmula (14).

$$sim_{Knappe}(c_1, c_2) = \rho \frac{|u(c_1) \cap u(c_2)|}{|u(c_1)|} + (1 - \rho) \frac{|u(c_1) \cap u(c_2)|}{|u(c_2)|} \quad (14)$$

Na fórmula (14), sendo $\rho \in [0,1]$ e associado ao grau de influência de generalizações. Esta medida varia entre 0 (para conceitos diferentes) e 1 (para conceitos similares).

3.1.1.5 Tabela comparativa

Nesta seção apresenta-se um breve resumo das medidas semânticas discutidas anteriormente:

a) **Caminho mais curto:** medida de simples aplicação, ligada diretamente ao caminho mais curto entre os dois conceitos a comparar numa taxonomia;
b) **Hirst & St-Onge**: variante da medida de caminho mais curto, que leva em consideração não só o caminho mais curto, mas também o número de desvios no caminho. Apresenta fraca performance;
c) **Wu & Palmer**: medida associada à distância dos conceitos até à sua raiz comum e a posição dessa raiz na taxonomia. Esta medida varia entre 0 e 1;
d) **Li**: medida não linear, que varia entre 0 e 1. Depende do caminho mais curto entre os conceitos e da sua profundidade na taxonomia. Introduz os parâmetros α e β, que podem ser ajustados para otimizar a performance;
e) **Leacock & Chodorow**: medida logarítmica que considera tanto a distância entre conceitos como a sua profundidade na taxonomia. Não apresenta limite superior;
f) **Lesk:** utiliza a sobreposição de atributos entre conceitos no mesmo contexto para determinar os significados corretos para cada conceito;
g) **Lord**: medida simples que permite analisar a sensibilidade da similaridade em relação à frequência e não ao IC;

h) **Resnik**: medida calculada a partir do IC do conceito que agrupa os conceitos em análise na taxonomia. Permite ainda determinar o tamanho do *corpus*;
i) **Lin**: leva em consideração a quantidade de informação em comum e também a quantidade de informação necessária para descrever cada conceito. Varia entre 0 e 1;
j) **Jiang**: define a distância semântica entre dois conceitos e pode resultar em altos valores, dependentes do tamanho do *corpus*;
k) **Tversky**: determinada a partir das caraterísticas comuns de ambos os conceitos. Não é simétrica, assumindo distinções nas relações entre superclasses e subclasses;
l) **Rodriguez**: considera uma classificação atípica das caraterísticas como funções, partes e atributos; e
m) **Knappe:** baseia-se na existência de múltiplos caminhos entre os dois conceitos e na possibilidade de conceitos complexos poderem ser constituídos por um ou mais termos.

Os aspetos fundamentais, das medidas apresentadas, encontram-se resumidos na Tabela 3 seguinte.

Tabela 3: Comparativo das medidas semânticas apresentadas

Medida	Aumento c/ semelhança	Diminuição c/ diferença	IC	Posição na hierarquia	Tamanho do caminho	Valor máximo = 1	Simétrico
Caminho mais curto	×	✓	×	✓	✓	✓	✓
Hirst & St-Onge	×	✓	×	✓	✓	×	×
Wu & Palmer	✓	✓	×	✓	✓	✓	✓
Li	✓	✓	×	✓	✓	✓	✓
Leacock	✓	✓	×	✓	✓	×	✓
Lesk			×	×	×	×	
Lord	✓	×	✓	✓	×	✓	✓
Resnik	✓	×	✓	✓	×	×	✓
Lin	✓	✓	✓	✓	×	✓	✓
Jiang	✓	✓	✓	✓	×	×	✓
Tversky	✓	✓	×	×	×	✓	×
Rodriguez	✓	✓	×	×	✓	✓	×
Knappe	✓	×	×	✓	×	✓	×

Notas: para cada critério de análise, foram estabelecidas três situações:
✓**:** apresenta;
×**:** não apresenta; e
Vazio: indeterminado.

Entre os critérios de análise definidos, é possível verificar:

a) As medidas em que o valor de similaridade cresce, com o aumento das caraterísticas em comum;

b) As medidas em que o valor de similaridade diminui, com a diferença nas caraterísticas em comum;
c) As medidas que têm em consideração o IC;
d) As medidas em que a posição de um conceito na taxonomia tem influência no cálculo da similaridade;
e) As medidas que apresentem resultados limitados ou passíveis de obterem valores infinitos; e
f) As medidas simétricas, ou seja, $sim(a, b) = sim(b, a)$.

Apresentadas as medidas semânticas, na próxima seção serão abordadas as medidas sintáticas.

3.1.2 Medidas sintáticas

Desde a década de 60 até aos dias de hoje, diversas medidas de comparação entre conjunto de carateres, ou seja, palavras (*strings*), foram propostas por diversos autores. Dentre elas, algumas destacam-se e serão apresentadas a seguir.

3.1.2.1 Q-Gram

Um *q-gram* é o conjunto de todas as partes (*substrings*) que podem ser definidas, a partir de uma *string* inicial, em que *q* representa o tamanho de cada uma destas *substrings*. Por exemplo, partindo da *string* "Teclado" e definindo $q = 3$, obtêm-se os seguintes *q-grams*:

$$\{\#\#T; \#Te; Tec; ecl; cla; lad; ado; do\$; o\$\$\}$$

Inicialmente, o algoritmo *q-gram* foi utilizado como método de filtragem, cujo objetivo era eliminar áreas onde não fosse possível existir correspondência entre palavras. No entanto, serve também para identificar sequências de texto que possuam palavras em comum. A comparação de duas *strings*, *a* e *b*, pode ser feita gerando os respetivos *q-grams* e contabilizando quantos estas têm em comum. A distância *q-gram* é definida pela fórmula (15).

$$dist_{q-gram} = \big(\max(|a|, |b|) + (q-1)\big) - (q - grams\ em\ comum) \tag{15}$$

O tempo associado à aplicação deste método é relativamente reduzido, estando relacionado ao tamanho da menor *string* a ser comparada. No entanto, a qualidade dos resultados é fraca, obtendo-se valores de similaridade baixos para palavras semelhantes [82].

3.1.2.2 Hamming

A medida de distância de *Hamming*, também conhecida como coeficiente de *Hamming*, é o método mais simples para determinar a distância entre duas cadeias de carateres. Esta medida é definida pela fórmula (16).

$$d_{Hamm}(a,b) = \sum_{i=0}^{n} |a_i - b_i| \tag{16}$$

Na fórmula (16), *a* e *b* são duas cadeias de carateres e *n* é o tamanho das mesmas. É importante referir que neste método o tamanho das cadeias deverá ser idêntico para ambas. Esta técnica é tipicamente utilizada na análise de sequências binárias, visto a largura dos bytes ser constante e, portanto, todas as palavras têm o mesmo tamanho [83].

Na Figura 14, as duas cadeias de *bits* têm uma distância de *Hamming* igual a 4, já que 4 dos seus *bits* são diferentes.

1 0 0 1 1 0 1 0
1 0 0 0 1 1 0 1

Figura 14: Comparação de duas cadeias de bits

3.1.2.3 Jaro

Uma medida de similaridade, amplamente utilizada, é a medida de *Jaro*. Este método baseia-se no número e na ordem dos carateres em comum entre duas palavras [74]. Assim, dadas duas cadeias de carateres *a* e *b*, definidas como $a = a_i \dots a_m$ e $b = b_1 \dots b_n$, diz-se que um caratere a_i da cadeia *a* é comum a *b* se existir um $b_j = a_i$, onde *j* não pode diferir de *i* mais do que $h = \min(|a|, |b|)/2$, sendo $|a|$ e $|b|$ o tamanho de cada cadeia, ou seja, *j* está limitado por $i - h \leq h \leq i + h$.

Sendo $a' = a'_1 \dots a'_k$ os carateres de *a* que são comuns a *b* (na mesma ordem em que se encontram na cadeia original) e, analogamente, $b' = b'_1 \dots b'_k$, os carateres de *b* que são comuns a *a*, é possível definir uma transposição como uma posição em que $a'_i \neq b'_i$. Define-se então $T_{a',b'}$ como sendo metade do número de transposições de a' para b' [75].

A métrica de *Jaro*, aplicada às *strings a* e *b*, é definida pela fórmula (17) [84].

$$Jaro(a,b) = \frac{1}{3}\left(\frac{|a'|}{|a|} + \frac{|b'|}{|b|} + \frac{|a'| - T_{a',b'}}{|a'|}\right) \tag{17}$$

3.1.2.4 Jaro-Winkler

Uma variante da medida de *Jaro,* proposta em 1990 por Winkler [85], usa também uma variável *P*, designada como o tamanho do maior prefixo comum a *a* e *b*. Definindo $P' = \max(P, 4)$, a medida de *Jaro-Winkler* é apresentada na fórmula (18) [75].

$$Jaro - Winkler(a,b) = Jaro(a,b) + \frac{P'}{10}(1 - Jaro(a,b)) \tag{18}$$

Tanto a medida de *Jaro*, como a variante de *Jaro-Winkler*, foram desenvolvidas para *strings* relativamente pequenas, sendo nesses casos que se obtêm os melhores resultados.

3.1.2.5 Levenshtein

A medida de *Levenshtein* é muito utilizada na área da medicina, nomeadamente na análise de cadeias de ADN [86]. Este algoritmo, desenvolvido em 1965 por Vladimir Levenshtein [87], tem como objetivo o cálculo da distância entre duas *strings*, *a* e *b*, cujos tamanhos são *m* e *n*, respetivamente, sendo esta distância também conhecida como *edit distance* (distância de edição).

A métrica de *Levenshtein* é uma métrica de distância entre palavras, que são referidas como palavra fonte e palavra destino. A distância total entre as duas *strings* é a soma do custo total de operações básicas (inserção, exclusão e substituição) necessárias para obter a palavra destino, a partir da palavra fonte [84].

A função de *Levenshtein* cria uma matriz que pode ser definida pela fórmula (19).

$$M(i,j) = \max \begin{cases} M(i-1,j) - 1 & \rightarrow \text{inserção} \\ M(i-1,j-1) + p(i,j) & \rightarrow \text{correspondência ou substituição} \\ M(i,j-1) - 1 & \rightarrow \text{remoção} \end{cases} \quad (19)$$

Na fórmula (19), $M(0,0) = 0$ e a função $p(i,j)$ serve para determinar se há ou não correspondência entre dois carateres, e é definida por $p(i.j) = \begin{cases} 2 \; se \; a_i = b_i & \rightarrow \text{correspondência} \\ -1 \; se \; a_i \neq b_i & \rightarrow \text{substituição} \end{cases}$. Este algoritmo, apesar de não ser muito eficiente, é dos que apresenta maior flexibilidade, visto os pesos associados a cada operação poderem ser adaptados em função da aplicação desejada [83].

3.1.2.6 Monge-Elkan

A medida de *Monge-Elkan* é uma medida recursiva, usada para comparar duas *strings* de grandes dimensões, *a* e *b*. O algoritmo começa por dividir cada uma das *strings* em *substrings* ou *tokens*, ou seja, $a = a_i \dots a_K$ e $b = b_i \dots b_L$, compara cada uma das *substrings*, usando uma medida secundária de distância, $sim'(a_i, b_j)$, e associa a_i a b_j, de modo a obter o valor de similaridade máximo. Os resultados obtidos para cada *substring* de *a* são somados e normalizados pelo número de *tokens* de *a* [88].

A medida é *Monge-Elkan* definida pela fórmula (20).

$$sim_{ME}(a,b) = \frac{1}{K} \sum_{i=1}^{K} \max_{j=1}^{L} sim'(a_i, b_j) \quad (20)$$

3.1.2.7 Cosine e TF-IDF

Dados dois vetores n-dimensionais, V e W, a medida de similaridade *Cosine* determina o cosseno do ângulo α entre eles através da fórmula (21).

$$sim_{cosine}(V,W) = \cos \alpha = \frac{V.W}{\|V\|\|W\|} \quad (21)$$

Na fórmula (21), sendo $V.W$ o produto interno entre os dois vetores e $\|V\|$ o módulo do vetor $V = [a, b, c, \dots]$, calculado a partir da fórmula $\|V\| = \sqrt{a^2 + b^2 + c^2 + \cdots}$. Quando este método é aplicado à deteção de duplicados, os vetores V e W podem representar, tanto *tokens* de uma *string*, como descrições de um candidato [88]. Assumindo que se pretende comparar duas *strings*, *a* e *b*, divididas em *tokens*, levanta-se a questão de como construir os vetores V e W, a partir dos *tokens* obtidos. Para responder a esta questão, é necessário primeiro analisar a dimensionalidade *d* dos vetores, antes de verificar os valores que serão inseridos na criação dos vetores.

A dimensionalidade *d* dos vetores em questão irá corresponder a todos os *d tokens* distintos que apareçam em qualquer *string*, num dado domínio finito *D*. Neste caso, assumindo que tanto *a* como *b* são originários do mesmo atributo relacional, a_r, *D* corresponde a todos os *tokens* distintos em todos os vetores de a_r. Para grandes bases de dados, o número de *tokens* pode ser elevado e, consequentemente, V e W têm na prática uma alta dimensionalidade *d*.

Para a composição dos vetores, considera-se um vetor *T* de *D*, que tem também dimensionalidade *d*. Este vetor contém os pesos associados a cada um dos *d tokens* distintos de *D*, estando cada peso relacionado à relevância de cada *token* em *D*, quando comparado com os restantes. Para a determinação dos pesos referidos, o método *Term Frequency – Inverse Document Frequency* (TF-IDF) é usualmente utlizado [87].

De um modo geral, TF mede a frequência de ocorrência de um *token*, numa dada *string*, estando por isso, diretamente associado à relevância desse *token* na *string*, isto é, quanto mais vezes ocorre, mais relevante é o *token* no contexto da *string*. A frequência TF pode também ser associada à ocorrência de um dado termo num documento.

Existem diversas formas para se calcular esta frequência, sendo a mais simples aquela que associa *tf* à frequência absoluta $f(t, d)$, de um determinado *token* no domínio. Outras alternativas são, por exemplo, a frequência booleana, que define $tf_{t,d} = 1$, se *t* ocorre em *d* e $tf_{t,d} = 0$, caso contrário, a frequência de escala logarítmica, descrita por $tf_{t,d} = \log(f(t, d) + 1)$, ou ainda, uma última definição que evita parcialidade em domínios muito extensos, ficando $tf_{t,c}$ apresentado pela fórmula (22).

$$tf_{t,d} = 0.5 + \frac{0.5 \times f(t, d)}{\max\{f(w, d) : w \in d\}} \tag{22}$$

Por outro lado, em IDF são atribuídos pesos mais elevados a *tokens*, que ocorrem menos vezes no conjunto das descrições de candidatos. Isto é útil, pois define pesos baixos para *tokens* que apareçam frequentemente num domínio. Formalmente, o IDF de um determinado *token* é determinado pela fórmula (23).

$$idf_{t,D} = \log \frac{|D|}{|\{d \in D : t \in d\}|} \tag{23}$$

Na fórmula (23), sendo *D* o número total de documentos do *corpus*, e $|\{d \in D: t \in d\}|$, o número de documentos que contém *t*. TF-IDF é calculado combinando as duas frequências, apresentadas anteriormente, numa única operação, utilizando a seguinte fórmula (24).

$$tfidf_{t,d,D} = tf_{t,d} \times idf_{t,D} \tag{24}$$

Esta operação é efetuada para cada *token* $t_i \in D$, e os valores obtidos são associados ao vetor *T*. Por último, para criar os vetores V e W, associam-se os termos de ordem *i* aos valores da mesma ordem do vetor *T*, se o *token* associado t_i fizer parte das *strings* associadas a V e W. Caso contrário, ao termo de ordem *i* é atribuído o valor zero.

3.1.2.8 Tabela comparativa

Seguidamente, descreve-se, resumidamente, as principais caraterísticas de cada uma das medidas sintáticas, apresentadas anteriormente:

a) **Q-Gram**: é uma medida que permite a comparação de *strings* de tamanhos diferentes e cujo tempo de implementação depende apenas do tamanho da *string* menor. Contudo, os resultados que apresenta são fracos;
b) **Hamming**: é uma medida eficaz e de simples implementação, contudo, apenas pressupõe substituições de carateres, e implica a comparação de *strings* com o mesmo tamanho;
c) **Jaro/Jaro-Winkler**: medidas que apresentam bons resultados e permitem a comparação de *strings* de tamanhos diferentes. Particularmente, são adaptadas a *strings* pequenas;
d) **Levenshtein**: apresenta bons resultados e permite variações nos tamanhos das *strings*. Tem, no entanto, um tempo de implementação da ordem de |a|.|b|, tornando-se, portanto, bastante moroso para *strings* grandes;
e) **Monge-Elkan**: medida recursiva que permite a comparação de *strings* de tamanhos diferentes; e
f) **Cosine/TF-IDF**: medida flexível, que permite a utilização do *corpus*.

A Tabela 4 demonstra o comparativo das medidas sintáticas apresentadas.

Tabela 4: Comparativo das medidas sintáticas

	Tempo de execução ✓: satisfatório ×: insatisfatório	**Eficiência** ✓: eficiente ×: ineficiente	**Complexidade** ✓: baixa ×: alta	**Flexibilidade** ✓: flexível ×: limitado
Q-Gram	✓	×	✓	✓
Hamming	✓	×	✓	×
Jaro / Jaro-Winkler	✓	✓	×	✓
Levenshtein	×	×	✓	✓
Monge-Elkan	✓	✓	✓	✓
Cosine / TF-IDF	×	✓	×	✓

Abordadas as medidas de similaridade, semânticas e sintáticas, a seção seguinte irá introduzir as *frameworks* que implementam algumas dessas medidas.

3.2 Frameworks

Uma *application programming interface* (API) define e disponibiliza um conjunto de métodos usados por programadores, para interagir com o *software* ou o sistema, não impondo, idealmente, nenhuma estrutura específica ao programa. Por outro lado, uma *framework* é uma ferramenta, ou um conjunto de ferramentas, que implementam a API e ditam a forma como determinadas aplicações têm de ser desenvolvidas [89].

De seguida, será apresentado um conjunto de *frameworks* para o cálculo da similaridade semântica entre conceitos.

3.2.1 SimPack

SimPack é uma *framework*, cujo principal objetivo é o estudo da similaridade entre conceitos numa ontologia ou em ontologias como um todo. Pode também ser aplicada na análise de similaridades entre código-fonte para, por exemplo, detetar alterações entre classes de *software* de diferentes versões, e para determinar similaridades em dados hierarquicamente estruturados (ex.: XML) para comparar, procurar ou integrar dados de diferentes fontes [90].

Esta *framework* disponibiliza um grande número de medidas de similaridade, tais como as apresentadas anteriormente: *Jaro, Levenshtein, Jiang* e *Conrath, Lin, Resnik, Cosine* e *TF-IDF*. Tem, ainda, a vantagem de ser uma *framework* genérica, ou seja, permite a sua utilização em diferentes estruturas de dados, desde que os pré-requisitos de acesso sejam cumpridos. SimPack está implementada na linguagem *Java*, garantindo, assim, a possibilidade de ser utilizada em diversos sistemas operativos [90].

3.2.2 WS4J

WS4J ou *WordNet Similarity for Java* fornece uma API para diversas medidas de similaridade semântica. As medidas disponíveis nesta *framework* utilizam como recurso o *WordNet* para obter a similaridade entre conceitos. Entre as medidas disponibilizadas estão: *Hirst* e *St-Onge; Leacock* e *Chodorow; Lesk; Wu* e *Palmer*; *Resnik*; *Jiang* e *Conrath;* e *Lin*, também mencionadas anteriormente.

A *framework* WS4J é a solução implementada em *Java*, do projeto WordNet::Similarity, que foi originalmente desenvolvido em Perl. Esta derivação foi conseguida pelo grupo do professor Ted Pedersen, na Universidade do Minesota [91].

3.2.3 DKPro

DKPro é uma *framework open source*, desenvolvida em *Java* para o tratamento de linguagem natural. Esta *framework* está dividida em vários módulos para uma separação das suas funcionalidades [92]. Os principais módulos são:

- **DKPro Core**: Incorpora funcionalidades para o tratamento de linguagem natural (NLP), baseado em Apache UIMA[9]; e
- **DKPro Similarity**[10]: É um módulo que complementa o anterior e disponibiliza uma variedade de medidas para cálculos de similaridade entre textos.

As medidas implementadas no módulo DKPro Similarity estão agrupadas em dois grupos [92]:

1) **Medidas Lexicais**: Medidas que apenas comparam textos: GreedyStringTiling, Jaro, Levenshtein, LongestCommonSubsequence, MongeElkan e NGramBased; e
2) **Medidas semântico-lexicais**: Medidas que utilizam recursos como o *WordNet* para comparação de textos, tais como: GlossOverlap, JiangConrath, LeacockChodorow, Lin, Resnik e WuPalmerComparator.

3.2.4 Tabela comparativa

Nesta seção serão comparadas as *frameworks* descritas anteriormente. A Tabela 5 apresenta o resultado da comparação, com base nos seguintes critérios:

a) Tecnologia: Linguagem de programação utilizada para o desenvolvimento da *framework*;
b) Versão: Data e versão da última distribuição da *framework*;
c) Medidas implementadas: Medidas de similaridade implementadas e disponibilizadas na API da *framework*; e
d) Vantagens e desvantagens: Principais vantagens e desvantagens de cada *framework*.

Tabela 5: Resumo comparativo das *frameworks* (API)

	Frameworks		
	SimPack	**WS4J**	**DKPro**
Tecnologia	*Java*	*Java*	*Java*
Versão	0.91 (2008)	0.1.0 (2015)	*Core*: 1.8.0 (2016) *Similarity*: 2.2.0 (2016)
Medidas sintáticas	Jaccard Cosine Dice Averaged String Matching Jaro TFIDF	×	GreedyStringTiling Jaro Levenshtein LongestCommonSubsequence MongeElkan NGramBased
Medidas semânticas	Jiang & Conrath Lin Resnik	Wu & Palmer Jiang & Conrath Leacock & Chodorow Lin Resnik Banerjee & Pedersen Hirst & St-Onge	GlossOverlap JiangConrath LeacockChodorow Lin Resnik WuPalmerComparator
Recursos adicionais	Processamento de linguagem natural (NLP) Similaridade de documentos, grafos e árvores	×	Processamento de linguagem natural (NLP) Classificação de texto

[9] **Apache UIMA**: https://uima.apache.org
[10] **DKPro Similarity**: https://dkpro.github.io/dkpro-similarity

	Frameworks		
	SimPack	**WS4J**	**DKPro**
Vantagens	Possui medidas sintáticas e semânticas; Disponibiliza exemplos de teste	Implementação das principais medidas baseadas em *WordNet*	Framework recente; Open source; Comunidade ativa na evolução da *framework*; Disponibiliza exemplos de teste
Desvantagens	Documentação limitada	Apenas disponibiliza medidas baseadas em *WordNet*; Pouca documentação	×

Apresentadas as *frameworks*, serão descritas, na seção seguinte, algumas das ferramentas existentes para a integração de modelos.

3.3 Ferramentas

Atualmente, existem diversas ferramentas que permitem operar com ontologias. Seguidamente, serão apresentadas algumas das mais proeminentes ferramentas disponíveis.

3.3.1 HCONE

Dadas duas ontologias, O_1 e O_2, a ferramenta HCONE (*Human Centered Ontology Environment*) procura encontrar uma semelhança entre as duas ontologias e uma ontologia intermédia "escondida". Na Figura 15, verifica-se o *WordNet* no papel de ontologia intermédia, sendo os conceitos das ontologias originais mapeados, utilizando morfismo semântico determinado a partir do método LSI (*Latent Semantic Index*), que associa os conceitos da ontologia a significados do *WordNet.*

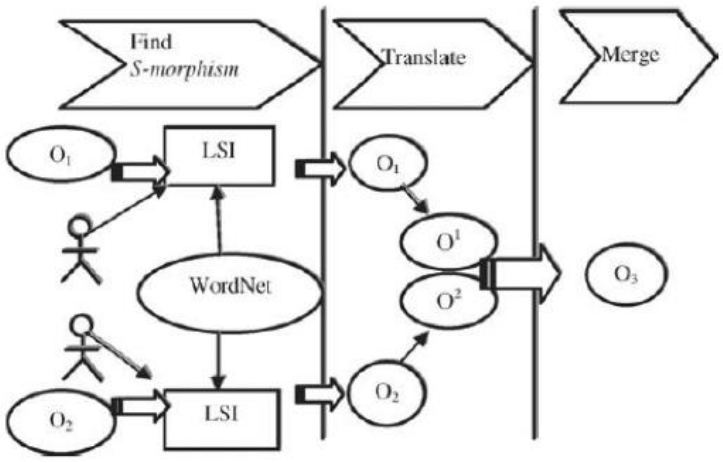


Figura 15: Representação esquemática do método HCONE [93]

Nesta ferramenta não se considera que o *WordNet* inclua quaisquer ontologias intermédias, visto que a inclusão dessa hipótese impõe grandes restrições na especificação das ontologias originais, ou seja, o método apenas funcionaria para ontologias que preservassem as relações de inclusão entre significados do *WordNet*. Na realidade, HCONE assume que a ontologia intermédia existe e é construída ao mesmo tempo que associa os conceitos à *WordNet* [93].

WordNet pode ser substituída por qualquer outro dicionário, ou mesmo por coleções de documentos que descrevam os conceitos presentes nas ontologias. Contudo, as vantagens associadas à *WordNet*, como por exemplo, o facto de ser bem estruturada e bastante vasta, contendo um grande número de entradas e relações semânticas, a tornam uma candidata ideal para o papel de ontologia intermédia.

Tal como se vê na Figura 15, apresentada anteriormente, após o mapeamento associado à ontologia intermédia, é feita a tradução das ontologias originais, que passam a designar-se O^1 e O^2. No final, as ontologias são combinadas criando uma nova ontologia O_3.

3.3.2 FCA-Merge

A ferramenta FCA-merge é utilizada para a comparação de ontologias [94] que tenham um conjunto de instâncias partilhadas, ou um conjunto de documentos partilhados anotados com conceitos em ontologias fonte. Baseada nesta informação, a ferramenta FCA-merge usa técnicas matemáticas de Análise Formal de Conceitos (FCA - *Formal Concept Analysis*) [95] para produzir uma rede de conceitos que relacionam os conceitos das ontologias originais. O algoritmo sugere correspondências e relações de superclasse-subclasse. Nesta etapa, cabe ao utilizador analisar o resultado e usá-lo como guia na criação de uma única ontologia [94], [96]. Na Figura 16 está descrita a representação esquemática da ferramenta FCA-merge.

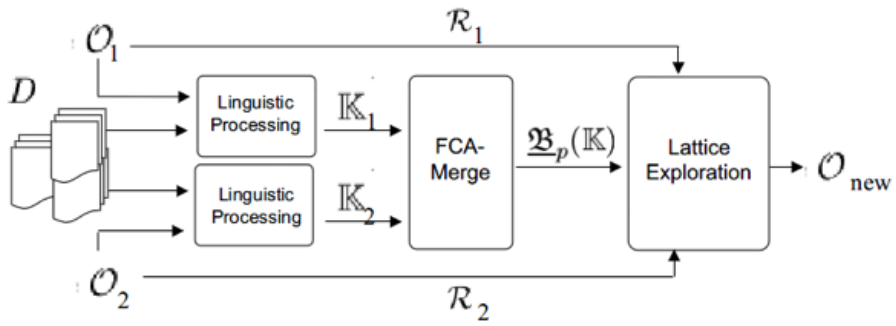


Figura 16: Representação esquemática do método FCA-merge [94]

Contudo, a hipótese referida inicialmente de que as duas ontologias devem ter instâncias partilhadas é considerada forte e, na prática, esta situação ocorre pouco frequentemente. Uma alteração proposta, de modo a contornar este problema, é a utilização de técnicas de processamento de linguagem natural (NLP), para anotação de um conjunto de documentos com conceitos de ambas as ontologias [97].

3.3.3 MAFRA

A ferramenta *MApping FRAmework for distributed ontologies* (MAFRA) é utilizada *para o* mapeamento de ontologias distribuídas na *Web Semântica.* De acordo com os autores Maedche et al. [98], é mais simples mapear ontologias existentes do que criar uma ontologia comum, visto o processo em causa envolver uma menor comunidade. Esta ferramenta introduz o conceito de ponte semântica

(*semantic bridge*), sendo este definido como uma representação declarativa de uma relação semântica entre ontologias fonte e alvo, e fornece os mecanismos necessários para transformar uma instância da ontologia fonte numa instância da ontologia alvo [99].

Na Figura 17 encontra-se uma representação conceptual da arquitetura do sistema MAFRA. Nesta arquitetura, um conjunto de módulos são organizados em dimensões horizontais e verticais.

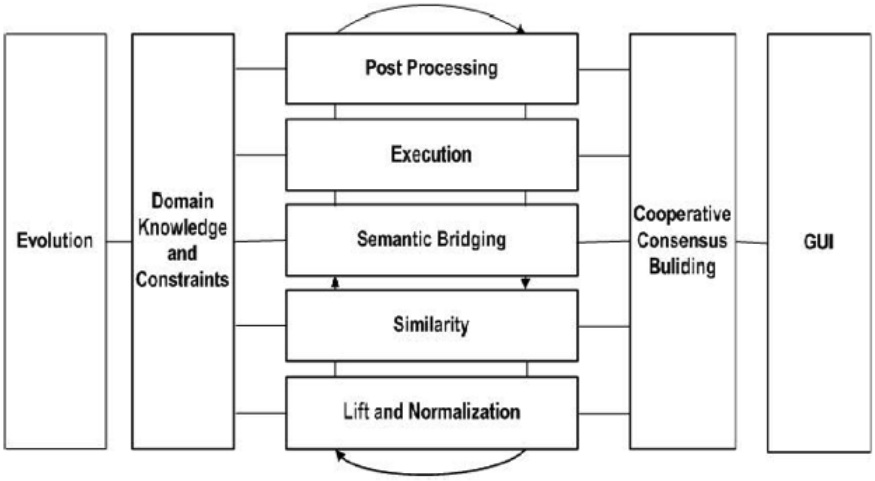


Figura 17: Arquitetura MAFRA [98]

Os módulos horizontais correspondem a cinco fases:

a) levantamento e normalização;
b) similaridade;
c) ligação semântica (*semantic bridging*);
d) execução; e
e) pós-processamento.

Enquanto os módulos verticais correspondem a quatro fases:

a) evolução;
b) conhecimento de domínio e restrições;
c) construção cooperativa consensual; e
d) GUI (*Graphical User Interface*).

Os módulos das duas dimensões interagem entre si, durante todo o processo de mapeamento de ontologias.

3.3.4 PROMPT

A ferramenta PROMPT engloba um conjunto de módulos que têm vindo a ter um grande impacto na área de combinação e alinhamento de ontologias. Os componentes principais da PROMPT foram

desenvolvidos como extensões (*plug-ins*) do ambiente de desenvolvimento de ontologias *Protégé-2000* [97]. Na Figura 18 é apresentada a ferramenta PROMPT.

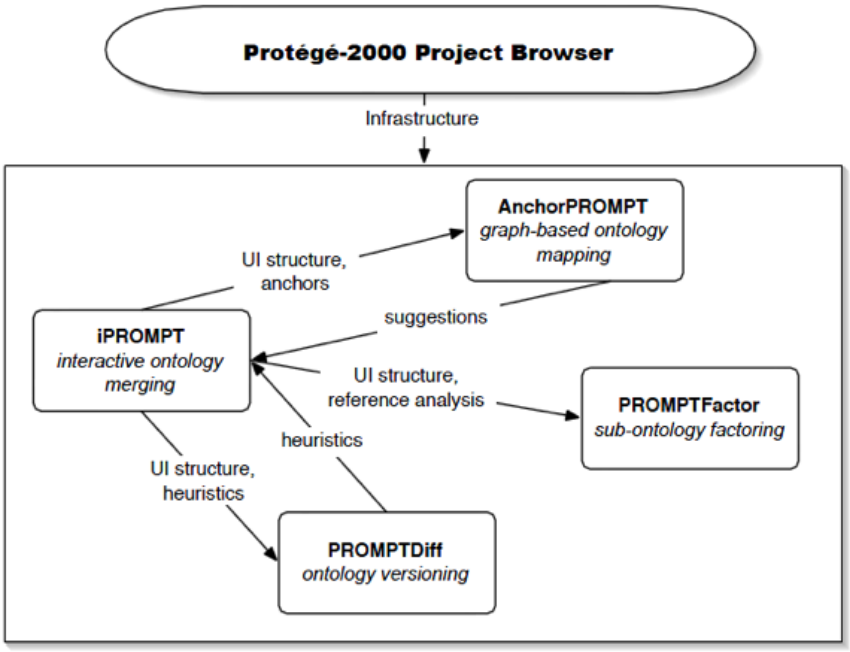


Figura 18: Infraestrutura da ferramenta PROMPT e interações entre os seus módulos [97]

Do conjunto apresentado, anteriormente, na Figura 18, o módulo iPROMPT (anteriormente designado por PROMPT) é utilizado para a combinação de ontologias. Trata-se de um módulo interativo, que ajuda o utilizador na tarefa de combinar ontologias, fornecendo sugestões acerca de quais elementos podem ser combinados. Identifica inconsistências e potenciais problemas e, ao mesmo tempo, sugere possíveis estratégias de resolução dos mesmos [99].

O algoritmo usado na combinação de ontologias encontra-se esquematizado na Figura 19, estando as caixas cinzentas associadas a ações da PROMPT, enquanto as caixas brancas correspondem a intervenção humana.

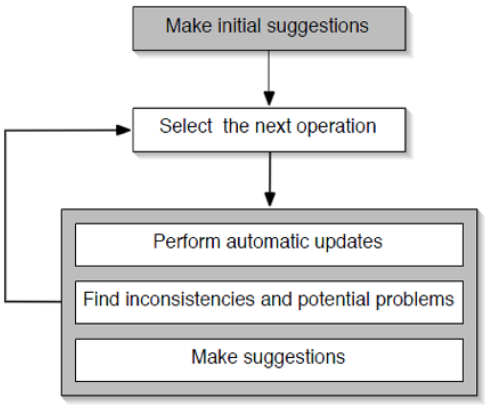


Figura 19: Representação esquemática do algoritmo iPROMPT [97]

A ferramenta PROMPT cria uma lista inicial de correspondências, baseada em nomes de classes, iniciando o ciclo iterativo. Neste ciclo, o utilizador escolhe uma das sugestões da PROMPT, ou define, diretamente, a operação desejada, recorrendo ao ambiente de edição de ontologias. O algoritmo executa, automaticamente, alterações adicionais, tendo por base o tipo de operação. Uma nova lista de sugestões é apresentada ao utilizador, com os conflitos originados na última operação e com as possíveis soluções para os mesmos [99].

3.3.5 S-Match

A ferramenta *open source* S-Match[11] possui implementações de diversos algoritmos para efetuar correspondências semânticas entre *lightweight ontologies*. As *lightweight ontologies* resultam da transformação dos termos numa classificação, para uma designação formal. São usadas técnicas de processamento de linguagem natural (NLP), para a transformação efetuada, resultando numa expressão formal inequívoca, ou seja, uma expressão proposicional *Description Logic* (DL) [100].

Na sua implementação, a ferramenta S-Match apresenta três diferentes formas de análise de correspondências: *basic semantic matching*, *minimal semantic matching* e *structure preserving semantic matching* (SPSM) [100].

O algoritmo para *basic semantic matching* determina o grau de similaridade, considerando os termos que nomeiam os conceitos, a posição do conceito na árvore (*lightweight ontology*) e as relações entre termos e conceitos na árvore. No final deste algoritmo, para cada par de conceitos, é criado um relacionamento semântico, de acordo com o tipo de relação obtida. Os tipos de relações podem ser: equivalência (=), menos geral ($\subseteq$), mais geral ($\supseteq$) e disjunção ($\perp$).

A análise *minimal semantic matching* considera a hierarquia de conceitos das *lightweight ontologies* e efetua a subsunção de conceitos, de acordo com a interpretação semântica das relações. A subsunção tem como objetivo minimizar a estrutura hierárquica, associando conceitos específicos aos conceitos mais genéricos, diretamente ligados. Esta análise tem como resultado uma lista menor de correspondências, em comparação com a análise *basic semantic matching,* para facilitar a interpretação dos especialistas de domínio.

A análise SPSM é uma variante do algoritmo *basic semantic matching*, mas com a diferença de preservar a estrutura dos conceitos em análise. Considera apenas uma correspondência por conceito e só efetua correspondências com elementos no mesmo nível estrutural (conceitos genéricos com conceitos genéricos e conceitos específicos com conceitos específicos).

A ferramenta S-Match divide a sua análise em quatro passos. Os dois primeiros passos são responsáveis por efetuar as transformações de termos e conceitos dos inputs recebidos, caraterizando a componente o*ffline* da ferramenta. Os outros dois passos, compõem a componente *online* e é onde se

[11] http://semanticmatching.org/s-match.html

efetua uma análise de similaridade entre termos, que nomeiam os conceitos, e uma análise de similaridade com base na estrutura. Na Figura 20 é apresentada a arquitetura da ferramenta S-Match.

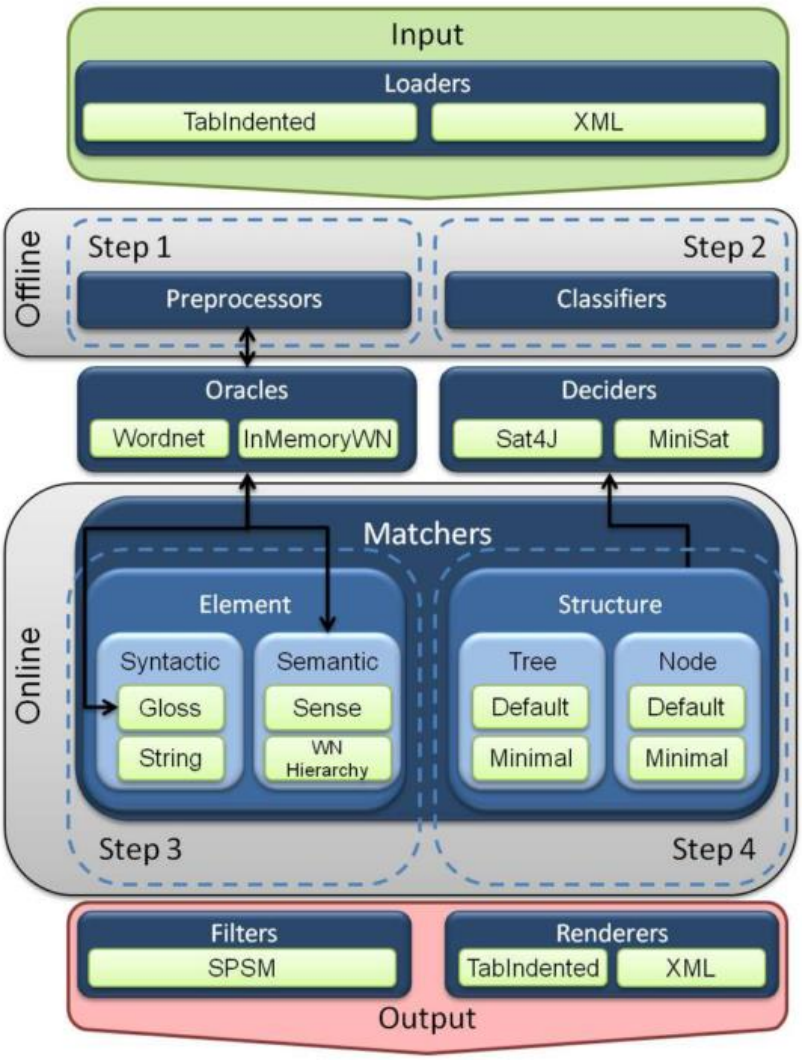


Figura 20: Arquitetura S-Match [100]

Na componente *offline*, os *inputs* (classificações, árvores de conceitos, etc.) são processados e transformados em *lightweight ontologies*. Este processamento envolve a aplicação de técnicas de linguagem natural e de classificadores, para a transformação dos termos que nomeiam os conceitos, e para a criação das expressões DL que definem a *lightweight ontology*. Recursos externos, como por exemplo, o WordNet, são utilizados para completar a informação dos conceitos.

A componente *online* agrupa os dois últimos passos, sendo responsável pela análise do grau de similaridade entre os conceitos. Ao nível do elemento (termos que nomeiam nomes e relações) são usadas medidas sintáticas (Prefix, Suffix, EditDistance, NGram) e semânticas (WordnetMatcher e WNHierarchy), e informação obtida de recursos externos (WNGloss e WNExtendedGLoss) [100]. Ao nível da estrutura, são feitas análises à arvore de conceitos resultante dos passos 1 e 2 (componente *offline*).

A ferramenta S-Match pode ser utilizada a partir das seguintes interfaces:

- **Java API:** Útil para as aplicações implementarem novas funcionalidades ou alterarem o modo de funcionamento da ferramenta, de acordo com as necessidades do problema;
- **Linha de comandos:** Útil para criar correspondências em bloco, de forma automatizada, com a possibilidade de definir as opções desejadas, a partir de ficheiros de configuração; e
- **Aplicação cliente:** Útil para os especialistas testarem e visualizarem os resultados, ou seja, as correspondências obtidas. Na Figura 21 é possível verificar o resultado de um alinhamento efetuado com a aplicação cliente.

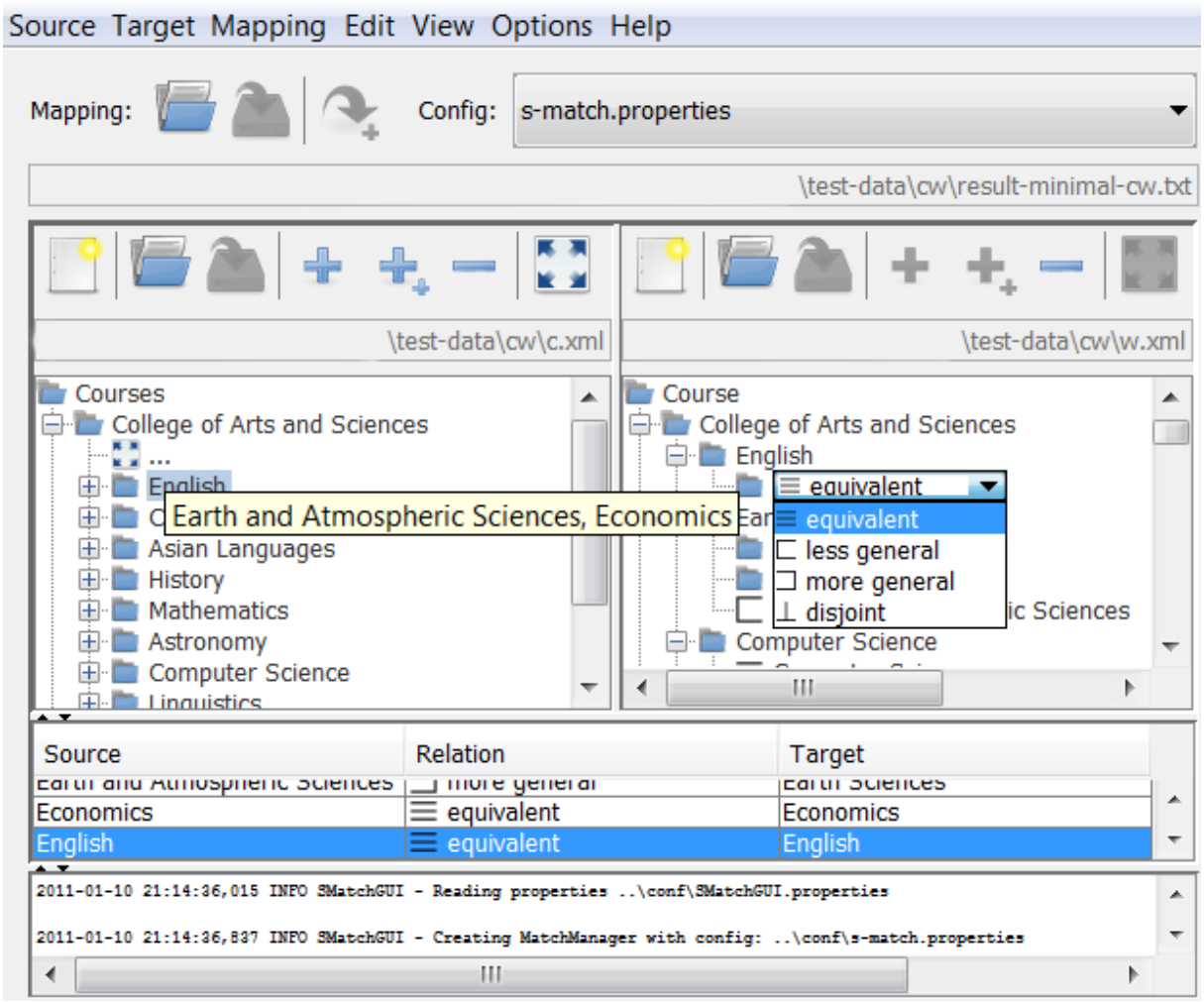


Figura 21: Resultado de um alinhamento utilizando a ferramenta S-Match

3.3.6 Comparação das ferramentas

A diversidade das ferramentas existentes torna a comparação entre elas uma tarefa difícil. Na verdade, a escolha da ferramenta mais apropriada vai depender da tarefa pretendida e do tipo de recurso que os especialistas elaboram.

No sentido de facilitar a escolha da ferramenta, existem alguns critérios que podem ser tomados em consideração:

a) **Tipo de mapeamento efetuado**: Ponto a ponto, mapeamento mediado, mapeamento *top-bottom* ou *bottom-top;*
b) **Resultado obtido**: Mapeamento e/ou combinação; e
c) **Nível de envolvimento do utilizador**: Para verificar o nível de automatização.

Na Tabela 6 é apresentada uma comparação das ferramentas descritas anteriormente, de acordo com os critérios definidos.

Tabela 6: Comparação entre as ferramentas

	Ferramentas				
	HCONE	**FCA**	**MAFRA**	**PROMPT**	**S-Match**
Input	Duas ontologias	Duas ontologias e um conjunto de documentos	Duas ontologias	Duas ontologias	Duas *lightweight ontologies*
Output	Uma ontologia combinada	Uma ontologia combinada	Mapeamento das duas ontologias (RDF[12])	Uma ontologia combinada	Lista com os relacionamentos entre pares de conceitos (=, ⊆, ⊇ e ⊥)
Tipo de resultado	Combinação	Combinação	Mapeamento	Combinação	Mapeamento
Princípio de funcionamento	Ontologia intermédia	Análise linguística	Pontes semânticas (SBO[13])	Análise heurística	- Análise sintática e semântica, utilizando recursos externos - Análise estrutural
Automatização	Automático ou semiautomático	Semiautomático	Semiautomático	Semiautomático	Automático

No capítulo seguinte será descrito todo o processo de integração de modelos conceptuais proposto neste trabalho.

[12] **RDF:** Resource Description Framework.
[13] **SBO:** Semantic Bridging Ontology.

4 PROCESSO DE INTEGRAÇÃO DE MODELOS CONCEPTUAIS

Neste capítulo será apresentada, de forma aprofundada, a abordagem desenvolvida para o cálculo da similaridade semântica entre modelos conceptuais. Em primeiro lugar, é realizada uma descrição geral da abordagem para integração de modelos conceptuais. Posteriormente, serão detalhados os requisitos funcionais e não funcionais que a proposta deve satisfazer. Seguidamente, será descrito o processo proposto para o cálculo de similaridade, composto por três fases.

A primeira fase tem como objetivo normalizar os nomes dos conceitos e das relações existentes nos modelos conceptuais. Na segunda fase serão aplicadas medidas sintáticas, entre pares de conceitos, para o cálculo de similaridade. A terceira e última fase corresponde à análise semântica entre os modelos conceptuais, a partir da aplicação de medidas específicas para o cálculo de similaridade.

Na terceira fase, a análise de similaridade está orientada, não aos termos que nomeiam os conceitos, mas aos elementos semânticos existentes nos modelos conceptuais. A similaridade entre os modelos, nesta última fase, considera a análise de acordo com: a estrutura conceptual; a categorização de relações; a taxonomia; as relações do mesmo tipo, partilhadas por conceitos vizinhos comuns; e o *corpus* associado a cada conceito, ou seja, conjuntos de documentos em que ocorre o termo associado a cada conceito.

Por último, será exposta uma descrição técnica sobre o serviço, que implementa a proposta, e o cliente, que auxilia os especialistas ao longo do processo de integração dos modelos. Serão mencionadas as tecnologias utilizadas, tanto no serviço como cliente, e a forma como o serviço e o cliente comunicam entre si.

4.1 Visão geral da abordagem

A construção colaborativa de modelos conceptuais, de um determinado domínio, requer que os especialistas trabalhem em conjunto e discutam quais os conceitos e as relações devem ser adicionados ao modelo. Cada especialista possui a sua própria visão do domínio e que, muitas vezes, não coincide com as ideias dos restantes membros do grupo de trabalho. Desta forma, numa construção colaborativa de modelos conceptuais, podem existir discrepâncias entre modelos e é necessário haver uma fase de negociação para alcançar o modelo final, com a concordância dos especialistas.

Para auxiliar os especialistas na fase de negociação, este trabalho tem como objetivo desenvolver uma abordagem para o cálculo de similaridade semântica, entre modelos conceptuais, e indicar quais os conceitos semelhantes, não só numa perspetiva sintática, mas também semântica.

A solução para o problema deve ser independente de qualquer ferramenta de desenvolvimento de modelos conceptuais. Para isso, é necessário que esteja acessível a partir de qualquer cliente e que consiga utilizar os modelos conceptuais. Como forma de apoio, deve existir uma ferramenta que acompanhe os especialistas na integração de modelos.

4.1.1 Requisitos funcionais

Considerando os objetivos apresentados anteriormente, a solução proposta, para o cálculo de similaridade e para a integração de modelos conceptuais, foi desenvolvida considerando os seguintes requisitos funcionais (genéricos e não estruturados):

a) Enquanto especialista de domínio, pretendo, em colaboração com o meu grupo, obter informações sobre o grau de similaridade, entre conceitos de dois modelos conceptuais, de modo a facilitar a decisão do grupo, no processo de negociação e integração de modelos conceptuais;
b) Como especialista de domínio, quero submeter para análise, modelos conceptuais com as seguintes caraterísticas:
    - Não existe um domínio e língua pré-definidos, deve ser possível analisar modelos de qualquer domínio e em qualquer língua;
    - Nomes dos conceitos e relações definidos de forma livre; e
    - Os conceitos podem conter, como informação adicional, uma lista de variantes, uma definição e um conjunto de frases para contextualização.
c) Como especialista de domínio, pretendo obter uma lista com o valor de similaridade entre os conceitos dos dois modelos (*matches*);
d) Como especialista de domínio, pretendo parametrizar o processo que conduz ao cálculo de similaridade efetuado:
    - Deve ser possível definir um valor mínimo de similaridade para limitar a lista de *matches* obtida (requisito c). *Matches* com o grau de semelhança inferior ao valor mínimo definido, não devem estar incluídos na lista de *matches* resultante; e
    - Deve ser possível variar a medida de similaridade a aplicar.
e) Como especialista de domínio, quero escolher que pares de conceitos considero válidos para efetuar o *merge* ou integração.

Na Figura 22 é apresentado o diagrama de casos de uso para o processo de integração de modelos conceptuais.

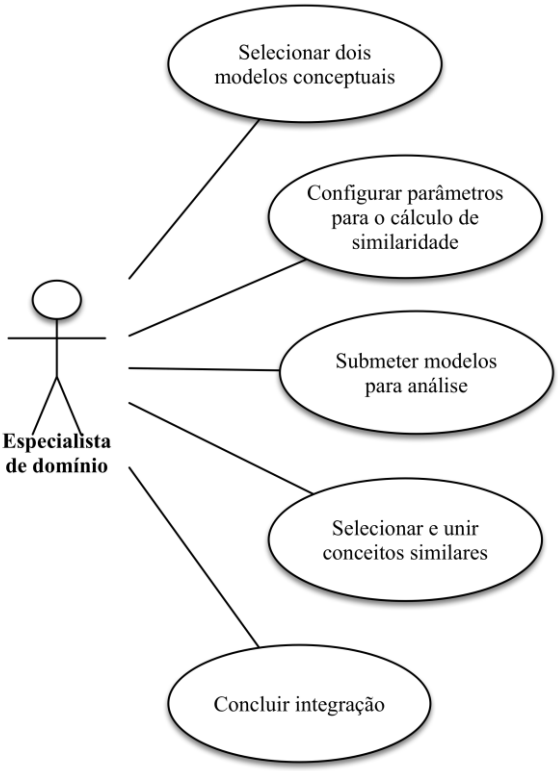


Figura 22: Caso de uso do processo de integração de dois modelos conceptuais

4.1.2 Requisitos não funcionais

Em complemento aos requisitos funcionais, a solução deverá corresponder aos seguintes requisitos não funcionais:

a) Estar disponível a partir de qualquer ponto e desacoplada de qualquer aplicação interessada em calcular similaridade entre modelos conceptuais; e
b) Estar disponível a partir de *web service*, utilizando o protocolo *HTTP REST* para comunicação, com trocas de mensagens, entre os clientes e o serviço, realizadas no formato *JSON*.

4.2 Proposta de abordagem para cálculo de similaridade semântica

A abordagem proposta neste trabalho assenta na aplicação de diferentes tipos de medidas para o cálculo de similaridade semântica. É uma abordagem interativa e iterativa, por considerar o envolvimento dos especialistas ao longo de *n* iterações. No final de cada iteração, será apresentado um conjunto de propostas de similaridade entre conceitos, e cabe ao especialista decidir se aceita ou não determinadas propostas.

As propostas sugeridas, em cada iteração, são o resultado da aplicação de diferentes medidas ao longo de três fases. Na primeira fase, serão aplicadas técnicas para a transformação e a normalização dos termos usados para nomear conceitos e relações. Na segunda fase, é efetuada uma análise sintática, recorrendo a medidas sintáticas para comparação de conceitos e relações. Por último, serão utilizadas medidas semânticas para uma análise à estrutura conceptual (conceito - relação - conceito), considerando

os tipos de relações conceptuais, a partilha de conceitos vizinhos comuns e o conteúdo de informação existente em cada conceito.

O processo termina após um número de iterações que os especialistas considerarem necessário, com vista à melhoria constante das propostas de integração de conceitos, resultantes do cálculo da similaridade semântica. Na Figura 23 é apresentada, conceptualmente, as diferentes fases que compõem a proposta.

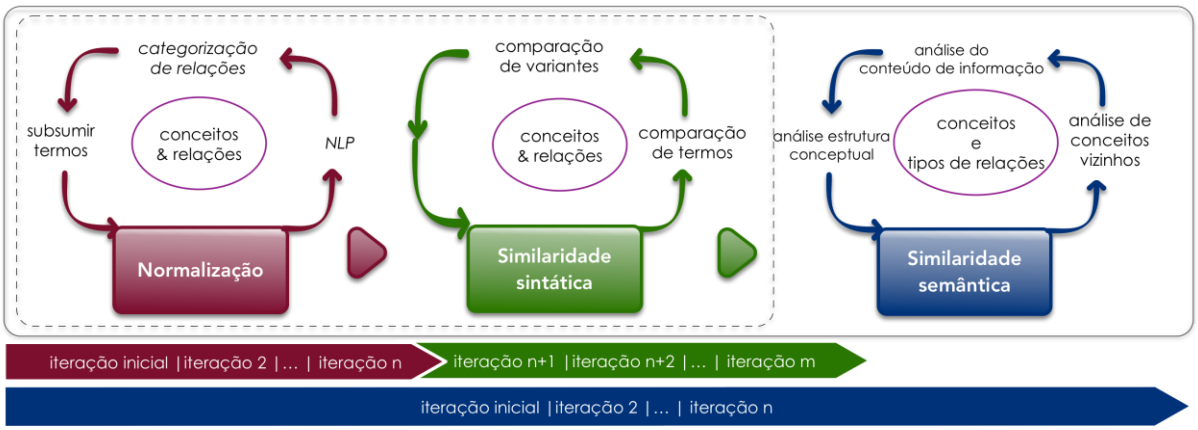
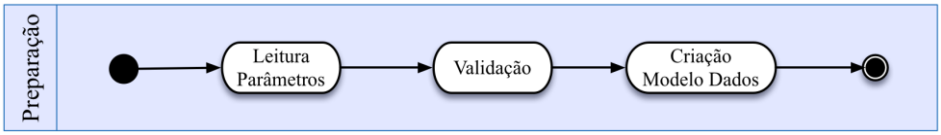


Figura 23: Processo de similaridade semântica

4.2.1 Preparação do processo

Antes de aplicar cada uma das três fases existentes no processo, é realizada uma primeira etapa para a inicialização e a validação dos modelos. Esta preparação inicial tem como objetivo garantir a validade de cada modelo conceptual. Um modelo é considerado válido se todos os conceitos e todas as relações tiverem um nome definido e, no caso das relações, estas devem conter referência para um conceito de origem (*source*) e para um conceito de destino (*target*). O resultado desta preparação estará na base das três fases do cálculo da similaridade semântica. Na Tabela 7 é descrito o fluxo da preparação do processo.

Tabela 7: *Workflow* da preparação do processo

Preparação	
Propósito	Validação e inicialização do modelo de dados.
Pressupostos	Nome dos conceitos e das relações devem estar definidos. Uma relação é considerada válida se possuir um conceito *source* e um conceito *target*.
Resultados	Modelo de dados a utilizar nas três fases do processo.
Recursos utilizados	*Parser* JSON to *Object*.
Fluxo	
	

4.2.2 Fase 1 – Normalização

A abordagem para o cálculo de similaridade semântica inicia-se com a normalização dos termos que nomeiam os conceitos e as relações presentes no modelo conceptual. Os termos definidos nos conceitos e nas relações serão transformados na sua versão linguística base[14]. Esta transformação, independente do contexto, é efetuada utilizando mecanismos de processamento de linguagem natural (*Natural language processing* - NLP). Com a aplicação de NLP, pretende-se eliminar variações linguísticas dos conceitos, que podem influenciar os resultados. Em particular, serão utilizados algoritmos de *Stemming*[15] e remoção de *stop words*[16], embora estejam sempre dependentes de um *dataset* para o idioma utilizado.

Exemplos de transformações aplicando os algoritmos de *stemming*:

- Carros ⇒ Carro
- Carrinho ⇒ Carro

Exemplo de transformações aplicando uma lista de *stop words*:

- Automóvel de passageiros ⇒Automóvel passageiro

Ainda, na fase 1, as relações serão categorizadas segundo uma ontologia de relações existente. Esta atividade de categorização de relações vai permitir uma uniformização da interpretação das estruturas conceptuais[17], que compõem os modelos conceptuais [101], facilitando o cálculo da similaridade semântica na fase 3. Nesta categorização, recorreu-se à ontologia de relações *Conceptual Relations Reference Model* (CRRM), proposta por Sousa [2].

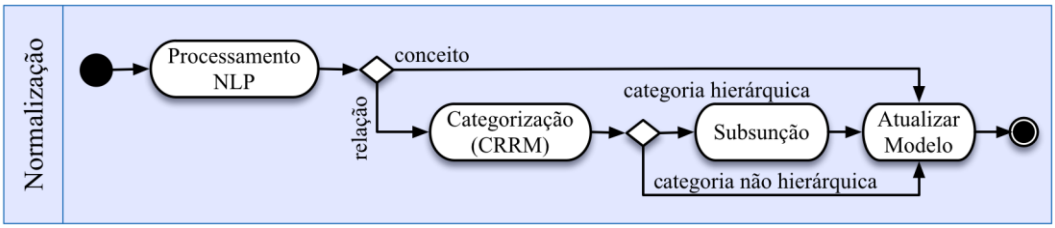
Após categorização das relações, se existirem relações taxonómicas (categoria hierárquica), os conceitos que estão ligados pela relação *is-a* serão subsumidos ao conceito mais genérico. Desta forma, as métricas de similaridade, aplicadas nas fases seguintes, irão considerar a informação complementar atribuída ao conceito genérico. Na Tabela 8 é descrita a Fase 1 – Normalização.

Tabela 8: *Workflow* da Fase 1 – Normalização

Fase 1 - Normalização	
Propósito	Uniformização das palavras. Categorização das relações. Subsunção de conceitos.
Pressupostos	Existe uma ontologia de relações para categorizar as relações existentes nos modelos.
Limitações	Língua em que o modelo está escrito. Ontologia de relações deve estar o mais completa possível.
Resultados	Modelos com termos que nomeiam os conceitos e relações normalizados.

[14] A versão linguística base representa o termo na sua forma original, ou seja, sem derivações [111].
[15] Stemming: Processo de reduzir os termos para a sua versão base [29].
[16] Stop Words: Lista de palavras que não serão consideradas [29].
[17] Uma estrutura conceptual é composta por: conceito – relação – conceito (CRC) [2].

Fase 1 - Normalização	
Recursos utilizados	*Stemming*, *stop words* e ontologia CRRM [2].
Fluxo	



Atividade de Processamento NLP

Nesta atividade serão realizadas as seguintes tarefas:

a) **Aplicar *stemming***: LucenePorterStemmer[18] implementado na API SimPack[19];

b) **Remover pontuação**: Substituição de texto a partir de expressões regulares (*regex replace*); e

c) **Remover *stop words***: *Regex replace* em conjunto com uma lista de palavras consideradas irrelevantes.

Atividade de categorização de relações

A categorização de relações é feita com recurso à ontologia de relações CRRM [2]. Nesta atividade, cada relação irá ser categorizada com base nos tipos de relações definidos na CRRM. Caso uma relação não se enquadre em nenhum dos tipos de relações definidas na ontologia, será atribuída uma categoria *default,* ou seja, relação sem categoria definida. Para evitar que existam muitas relações sem categoria, a ontologia CRRM deve ser alvo de constantes evoluções, acrescentando-se novas relações às categorias existentes na ontologia.

Atividade de subsunção de conceitos

A subsunção de conceitos é apenas aplicada a relações com categoria hierárquica [102]. Para efeitos de cálculos de similaridade, assume-se que nas estruturas conceptuais, com relações do tipo hierárquicas, o conceito genérico representa o conceito específico. Na Figura 24 é visível o resultado de uma subsunção.

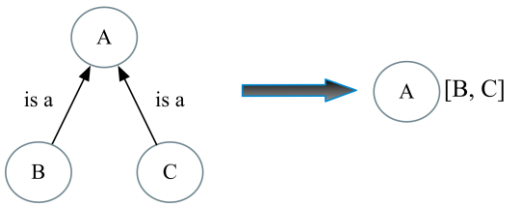


Figura 24: Subsunção de conceitos

[18] https://lucene.apache.org
[19] https://files.ifi.uzh.ch/ddis/oldweb/ddis/research/simpack/index.html

Como está demonstrado na Figura 24, apresentada anteriormente, os conceitos *B* e *C* podem ser vistos como o conceito *A*, ou seja, *B* = *A* e *C* = *A*. Esta subsunção de conceitos é utilizada para suportar a análise da estrutura conceptual, com base nos tipos de relações e conceitos vizinhos comuns, descrita na fase 3.

Atividade de atualização de modelos

Nesta última atividade da primeira fase, o modelo de dados é atualizado com o resultado das atividades anteriores. Após esta fase 1, pretende-se reduzir os problemas linguísticos que podem ocorrer no momento da criação de conceitos e de relações, num processo de conceptualização e, assim, aumentar a qualidade dos resultados, principalmente, quando for aplicada a fase 2, que visa essencialmente a comparação sintática entre os nomes dos conceitos.

4.2.3 Fase 2 – Análise sintática

Utilizando medidas de similaridade sintáticas, é realizada uma primeira análise, recorrendo unicamente aos nomes e variantes dos conceitos. Nesta etapa, os resultados indicarão se dois conceitos estão próximos (idênticos) ou distantes sintaticamente. Não é possível garantir que dois conceitos próximos sintaticamente sejam na realidade o mesmo conceito, pode-se, no entanto, informar o especialista sobre a probabilidade de existência de similaridade entre dois conceitos.

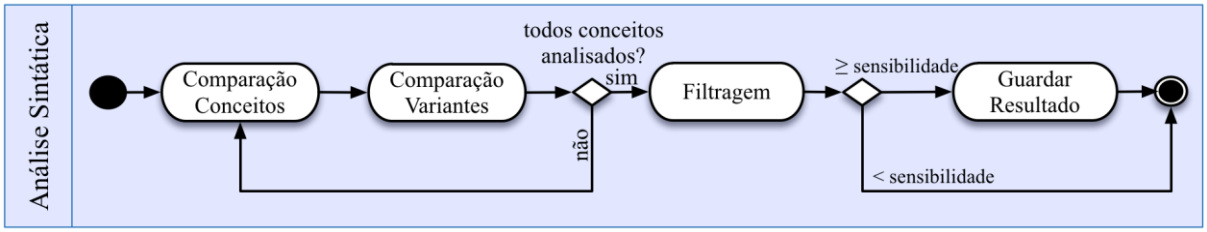
Nesta fase, o valor de similaridade entre dois conceitos é obtido a partir da aplicação das medidas *Levenshtein*, *Jaro* ou *MongeElkan* (descritas na seção 3.1.2, página 36). Em cada iteração do processo de similaridade, a seleção da medida a utilizar é uma decisão do especialista. Assim, os especialistas conseguem perceber a variação do valor de similaridade entre conceitos. Na Tabela 9 é descrita a Fase 2 – Análise Sintática.

Tabela 9: *Workflow* da Fase 2 – Análise Sintática

Fase 2 - Análise Sintática	
Propósito	Analisar sintaticamente o grau de semelhança entre conceitos.
Pressupostos	Deve ser definido um termo para nomear um conceito e uma relação.
Limitações	Apenas análise sintática (textual). Maus resultados para palavras homógrafas[20] e homónimas[21].
Resultados	Grau de similaridade sintático entre pares de conceitos.
Recursos utilizados	*SimPack* e *DKPro*[22].

[20] Palavras homógrafas: palavras com a mesma grafia, mas com pronúncia e significado diferentes [47].
[21] Palavras homônimas: palavras com grafia e pronúncia iguais, mas com significado diferente [47].
[22] https://dkpro.github.io

Fluxo



Atividades de comparação de conceitos e variantes

Estas atividades são realizadas em ciclo até que todos os conceitos e variantes sejam analisados. Neste processo cíclico, o valor de similaridade entre os conceitos é dado pela medida *Levenshtein*, disponível na *API SimPack* e pelas medidas *Jaro* e *MongeElkan*, disponíveis na *API DKPro*. Estas medidas irão analisar, conceito a conceito, as semelhanças sintáticas existentes. Sintaticamente, quanto mais carateres comuns dois conceitos tiverem, mais semelhantes eles serão. A possibilidade de escolher uma medida, entre as três introduzidas nesta fase, é vista como uma forma de perceber se o valor de similaridade entre os conceitos é constante nas três medidas, e, assim, concluir com mais precisão se os conceitos são ou não similares.

Atividades de filtragem e armazenamento dos resultados

Após o cálculo de similaridade efetuado entre todos os pares de conceitos dos dois modelos, a lista de resultados vai ser atualizada. Contudo, nem todos os resultados das comparações irão ser expostas ao especialista. O especialista pode definir um grau mínimo de similaridade para se considerar como correspondência (*match*). O parâmetro *sensibility,* definido pelos especialistas, permite excluir resultados com grau de similaridade abaixo do valor *sensibility* definido. É um parâmetro essencial para que a lista de resultados não seja muito extensa, a ponto de, no limite, existir um *match* por cada par de conceitos existentes nos dois modelos.

Na fase 2, devido ao elevado número de comparações, verifica-se um grande esforço computacional, mas, em compensação, é muito simplista em termos de aplicabilidade. É uma fase caraterizada pela capacidade de detetar conceitos com nomes ou variantes próximas sintaticamente. Como limitações, a fase 2 apresenta resultados de similaridade falsos, quando compara palavras homógrafas, ou seja, escritas da mesma forma, mas com significados diferentes. Por outro lado, conceitos distantes, ou seja, escritos de forma diferente, podem, no entanto, ser o mesmo conceito. Estes tipos de limitações devem-se ao fato das medidas sintáticas não terem em consideração a semântica dos conceitos e relações. Por este motivo, a abordagem proposta neste trabalho inclui, na sua fase 3, medidas semânticas para ultrapassar limitações sintáticas e encontrar novos conceitos similares.

4.2.4 Fase 3 – Análise semântica

A segunda vertente de análise de similaridade introduz abordagens semânticas para detetar novos *matchs* e superar as limitações existentes na fase 2. Estas abordagens semânticas consideram o significado das estruturas conceptuais, que compõem um modelo conceptual, não se limitando apenas à construção sintática do nome e das variantes que um conceito possui.

Nesta terceira fase, as medidas de similaridade semântica baseiam-se nas informações provenientes dos modelos conceptuais, tais como:

a) **Tipos de relações**: Análise do número de relações do mesmo tipo, associadas a dois conceitos. O tipo de relação é obtido da ontologia de relações CRRM;
b) **Tipos de relações e conceitos vizinhos comuns**: Análise realizada com base no número de conceitos vizinhos comuns e nas relações do mesmo tipo entre os conceitos;
c) ***Corpus****:* Similaridade obtida pelo cálculo de coocorrência de palavras no *corpus* dos conceitos em análise; e
d) **Taxonomia:** Similaridade calculada a partir da distância entre dois conceitos na taxonomia.

A qualidade dos resultados de similaridade está diretamente ligada à informação que é possível obter dos modelos conceptuais. Sendo assim, não é suficiente utilizar uma medida baseada no corpus, se os conceitos em análise não tiverem qualquer *corpus* associado, por exemplo. A utilização de medidas que recorrem a variados tipos de informação, tem como objetivo suportar diferentes caraterísticas que podem existir nos modelos conceptuais (exemplo: modelos conceptuais com relações taxonómicas e sem corpus; modelos conceptuais com utilização de relações existentes na ontologia CRRM, permitindo melhor análise da estrutura conceptual; etc.).

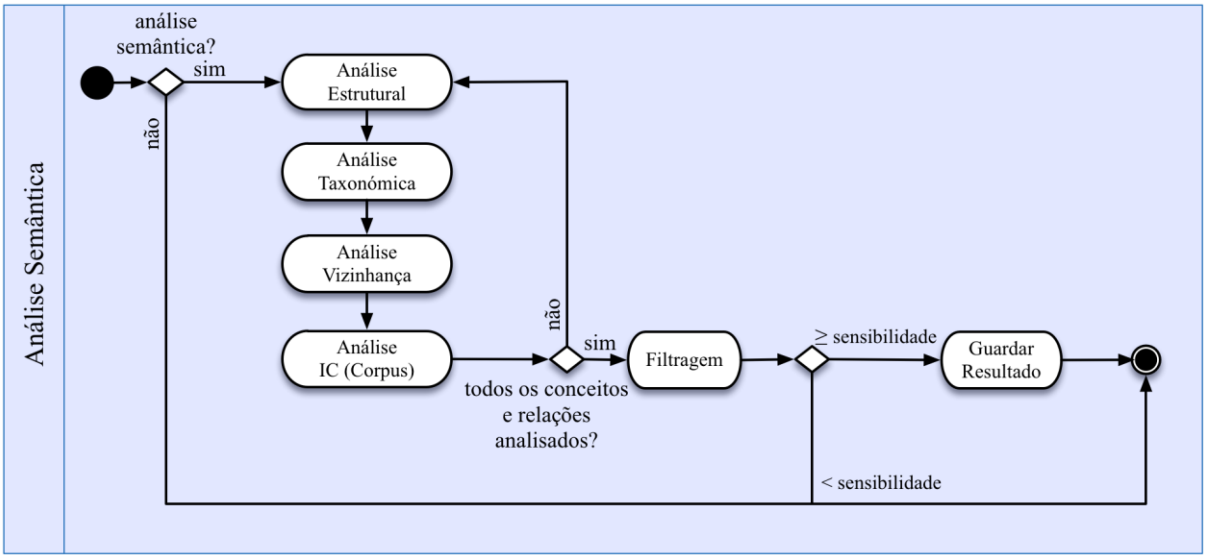
Nesta terceira fase da abordagem para o cálculo de similaridade, as quatro medidas são executadas, sequencialmente, para todos os pares de conceitos. A aplicação sequencial das medidas implica um maior esforço computacional, resultado do aumento do número de comparações necessárias. Em compensação, a fase 3 utiliza diferentes informações provenientes dos modelos conceptuais, alargando a compatibilidade com esses modelos. Na Tabela 10 é descrita a Fase 3 – Análise Semântica / Estrutural.

Tabela 10: *Workflow* da Fase 3 – Análise Semântica / Estrutural

Fase 3 - Análise Semântica / Estrutural	
Propósito	Calcular o grau de similaridade entre conceitos, com base em medidas semânticas e estruturais.
Pressupostos	Multi-domínio. Multi-língua.
Limitações	Modelos com um grau de complexidade elevada originam um grande número de comparações. (De todo o modo, e segundo as boas práticas de criação de mapas conceptuais, cada modelo deverá conter entre 15 a 25 conceitos [9]).

Fase 3 - Análise Semântica / Estrutural	
Resultados	Valor da similaridade semântico entre pares de conceitos.
Recursos utilizados	Medidas *Dice* - *CRRM*, Taxonómica, *Corpus* e Caraterísticas.

Fluxo



Atividade de análise estrutural

A atividade de análise estrutural tem como princípio calcular o valor de similaridade, considerando a categoria das relações. Para este cálculo, será utilizada a medida *Dice,* alterada para suportar relações conceptuais, doravante denominada de Dice – CRRM [2].

Segundo Sousa [2], o grau de similaridade entre conceitos dependerá do:

a) Número de relações comuns, com base no tipo de relação; e
b) Grau de relações dos conceitos em análise (relações com entrada e saída do conceito).

O valor de similaridade *Dice* - CRRM é obtido pela seguinte fórmula (25):

$$sim(c1, c2) = \frac{2 \times nN(G_{c1}, H_{c2})}{deg(G_{c1}) + deg(H_{c2})} \quad (25)$$

Em que, $nN(G_{c1}, H_{c2})$ é o número de relações entre os conceitos $c1$ e $c2$ com a mesma categoria, e $deg(G_{c1})$, $\deg(H_{c2})$ correspondem ao número de relações que ligam os conceitos $c1$ e $c2$, respetivamente. A Figura 25 ilustra um cenário de aplicação da medida *Dice – CRRM*, calculando a similaridade do conceito *a* com o conceito *a'*.

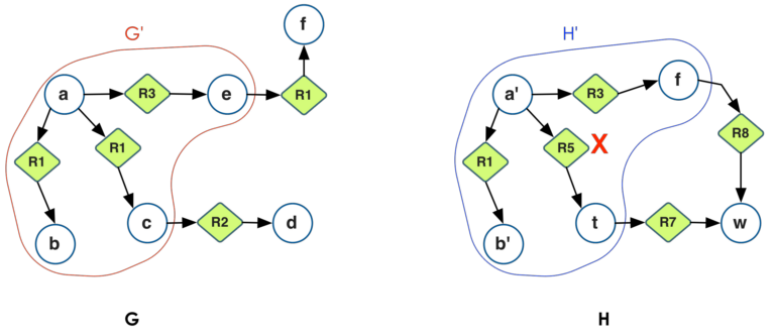


Figura 25: Medida Dice adaptada [2]

No exemplo apresentado, existem duas relações comuns entre *a* e *a'* (*R1* e *R3*). O grau (deg) de relações existentes em cada conceito (*a* e *a'*) é 3 (ambos os conceitos possuem 3 relações diretamente ligadas). Aplicando a fórmula (25), mencionada anteriormente, o valor de similaridade entre o conceito *a* e *a'* é:

$$sim(a, a') = \frac{2 \times 2}{3 + 3} \cong 0{,}67$$

Atividade de análise taxonómica

Esta atividade é caraterizada por considerar a posição dos conceitos numa taxonomia quando é calculado o valor de similaridade entre pares de conceitos. O grau de similaridade é obtido, utilizando a medida taxonómica proposta por Wu e Palmer (descrita na seção 3.1.1.1.4, página 29). Essa medida considera as distâncias taxonómicas entre:

- Os conceitos analisados e o conceito comum mais próximo - *least common subsumer* (*LCS*); e
- *LCS* e a raiz da taxonomia, em que a raiz é um conceito virtual adicionado no topo da taxonomia para que a hierarquia se inicie apenas num conceito.

A taxonomia utilizada para obter as distâncias entre conceitos é gerada automaticamente, a partir das modelos *source* e *target* em análise. Poder-se-ia utilizar taxonomias externas, mas a análise estaria dependente da existência dessas taxonomias para o domínio abordado nos modelos conceptuais submetidos.

A geração automática da taxonomia é feita utilizando as relações taxonómicas existentes nos modelos conceptuais. A integração de todos os triplos (considerado a estrutura conceptual conceito - relação - conceito), com relações taxonómicas, num só modelo, dá origem a uma taxonomia. Além dos triplos taxonómicos, são também consideradas as integrações de conceitos feitas pelo especialista, ao longo das iterações do processo de similaridade semântica. Desta forma, são criados elos de ligação entre conceitos na taxonomia, permitindo a existência de um conceito comum LCS entre conceitos do modelo *source* e os conceitos do modelo *target*. Na Figura 26 é demonstrado como é aplicada a medida taxonómica Wu e Palmer.

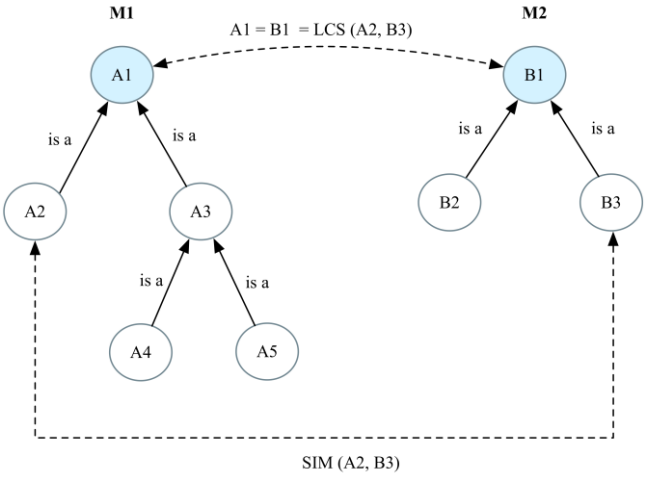


Figura 26: Aplicação da medida taxonómica

No exemplo da Figura 26 anterior, os conceitos *A1* e *B1* foram integrados pelo especialista numa iteração anterior do processo de integração dos modelos. Considerando que se pretende calcular a similaridade entre os conceitos *A2* e *B3*, a integração de *A1* com *B1* permitiu identificar o conceito LCS (*A1/B1*) comum a *A2* e *B3*. Os valores das distâncias taxonómicas neste exemplo são:

- *A2* até *A1/B1* = 1;
- *B3* até *A1/B1* = 1; e
- *A1/B1* até à raiz = 1 (uma ligação até ao topo da hierarquia).

Aplicando a fórmula (4), de Wu e Palmer (descrita na seção 3.1.1.1.4, página 29), o valor de similaridade entre os conceitos A2 e B3 é:

$$sim_{W\&P}(A_2, B_3) = \frac{2\times1}{1+1+2\times1} = 0{,}5$$

Enquanto o especialista não integrar nenhum conceito no processo de similaridade semântica, a análise taxonómica, baseada no Wu e Palmer, não terá qualquer efeito no cálculo da similaridade. Só depois da integração de conceitos é que será possível a criação da taxonomia e, assim, obter as distâncias dos conceitos na mesma. Ao longo das iterações e à medida que os conceitos forem integrados, a taxonomia vai crescendo, com novos elos de ligação.

A análise estrutural permite contornar o problema da falta de informação existente, quer da má construção dos modelos ou quer da falta de recursos existentes para a língua em que o modelo foi desenvolvido.

Atividade de análise da vizinhança (*Feature*)

Nesta atividade, o grau da similaridade entre dois conceitos vai ser obtido considerando as caraterísticas (*Feature*) de vizinhança. Serão analisados o número de conceitos vizinhos comuns e o número de relações com a mesma categoria. Consideram-se conceitos comuns quando, a partir de dois

conceitos *C1* (modelo 1) e *C2* (modelo 2), existem conceitos iguais, ligados diretamente a *C1* e *C2*. Assume-se que conceitos são iguais quando respeitam as seguintes regras:

a) Junções de conceitos em iterações anteriores; e
b) Junções com conceitos específicos, resultantes da subsunção de conceitos, descrita na fase 1 da abordagem para o cálculo de similaridade.

O valor da similaridade entre dois conceitos, utilizando a medida baseada em caraterísticas é dado pela fórmula (26).

$$Sim(c1, c2) = \left(\frac{Tn}{C} \times cw\right) + \left(\frac{Tr}{R} \times rw\right) \quad (26)$$

Sendo, Tn o número total de conceitos vizinhos comuns, C o número total de conceitos vizinhos, cw o peso atribuído ao número de conceitos comuns, Tr o número de relações comuns, R o número total de relações e rw o peso atribuído ao número de relações comuns. Os valores dos pesos utilizados no serviço de similaridade semântica foram definidos de forma equitativa, sendo $cw = rw = 0.5$. Na Figura 27 é apresentada a aplicação da medida baseada em tipos de relações e conceitos vizinhos comuns.

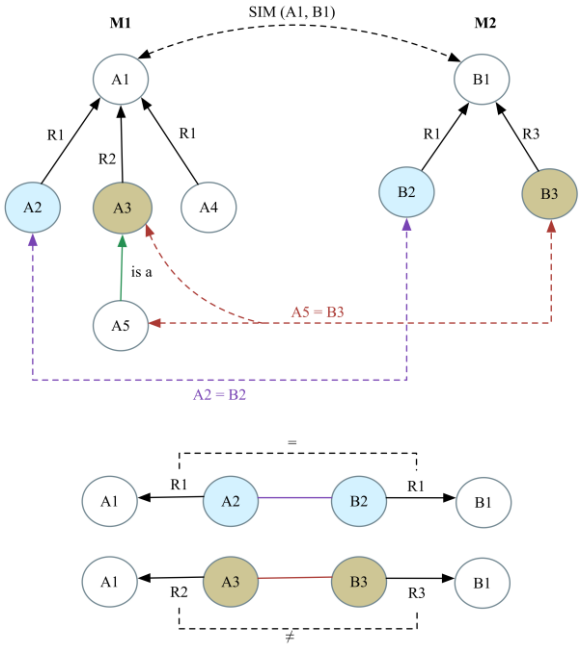


Figura 27: Exemplo da aplicação da medida baseada em tipos de relações e conceitos da vizinhança

No exemplo da Figura 27, os conceitos *A2*, *A3*, *B2* e *B3* são os vizinhos comuns entre *A1* e *B1*. *A2 – B2* são comuns porque foram integrados pelo especialista numa iteração anterior e *A3 – B3* são comuns porque *A5* foi integrado com *B3* e *A3* é o conceito genérico de *A5* (resultado da subsunção de conceitos). Das quatro relações que envolvem os conceitos vizinhos comuns, apenas as duas relações

R1 são da mesma categoria. Aplicando a medida baseada na caraterística de conceitos vizinhos comuns com relações da mesma categoria, o valor de similaridade para os conceitos *A1* e *B1* é:

$$Sim(A1, B1) = \left(\frac{4}{5} \times 0{,}5\right) + \left(\frac{2}{4} \times 0{,}5\right) = 0{,}65$$

Atividade de análise *corpus*

Por último, realiza-se uma análise baseada no conteúdo de informação partilhada IC. Utilizando a medida *Cosine* (descrita na seção 3.1.2.7, página 38) e considerando o *corpus* de cada conceito como recurso de informação, o grau de similaridade é obtido pelo conjunto de informação partilhada no *corpus* entre os conceitos. O *corpus* associado a cada conceito nos modelos conceptuais é composto por:

a) Definição;
b) Frases de documentos em que ocorre o mesmo termo que nomeia determinado conceito; e
c) Variantes.

A medida *Cosine* vai considerar as ocorrências de palavras comuns no *corpus* associado a cada conceito em análise.

Exemplo de corpus:
Carro: (...) um carro possui rodas (...)
Automóvel: (...) o meio de transporte automóvel possui rodas (...)

Neste simples exemplo é possível identificar um padrão comum, no *corpus*, "possui rodas". Face a esta partilha de informação, carro e automóvel irão possuir um valor de similaridade mais elevado, o que poderá ser um indício de que se trata do mesmo conceito. Quanto mais elementos comuns existirem, maior será o valor de similaridade. Esta medida tem a desvantagem de só apresentar resultados viáveis se os conceitos estiverem bem definidos.

Atividades de filtragem e armazenamento dos resultados

Tal como na fase 2, nesta terceira fase também existem as atividades de filtragem e armazenamento dos resultados. A atividade de filtragem tem como objetivo descartar os pares de conceitos (*matches*), de acordo com as opções definidas pelos especialistas (descritas com detalhe na seção 4.3.2, página 65).

4.3 Exemplo ilustrativo da abordagem

Para o auxílio dos especialistas, no processo de integração de modelos conceptuais, foi desenvolvida uma interface cliente, denominada de *SimSemanticaClient*, para simular a interação entre os especialistas e o serviço (descrito em detalhe na seção 4.4, página 70), que implementa a abordagem proposta neste trabalho.

Apesar do serviço ser capaz de calcular a similaridade entre conceitos de forma independente, sem qualquer cliente específico, respeitando apenas as regras definidas, a integração (*merge*) tem melhores resultados se for considerado o envolvimento dos especialistas [103].

4.3.1 Importação de modelos conceptuais

Finalizada a primeira fase da conceptualização colaborativa [1], da qual resultam várias soluções, os modelos precisam ser integrados num único modelo. Para utilização do cliente *SimSemanticaClient*, como ferramenta para a integração dos modelos, é necessário importar os modelos, resultantes da conceptualização, para dar início ao processo de integração. O cliente, desenvolvido neste trabalho, permite a importação de modelos conceptuais, desenvolvidos em CMapTools [104] ou em plataformas *Wiki*[23], baseadas no *semantic media wiki* (SMW[24]), como por exemplo o ConceptME [3].

O modelo importado é armazenado no cliente, de forma a estar disponível para operação de *merge*. A Figura 28, apresentada a seguir, mostra as funcionalidades de listagem e importação de modelos.

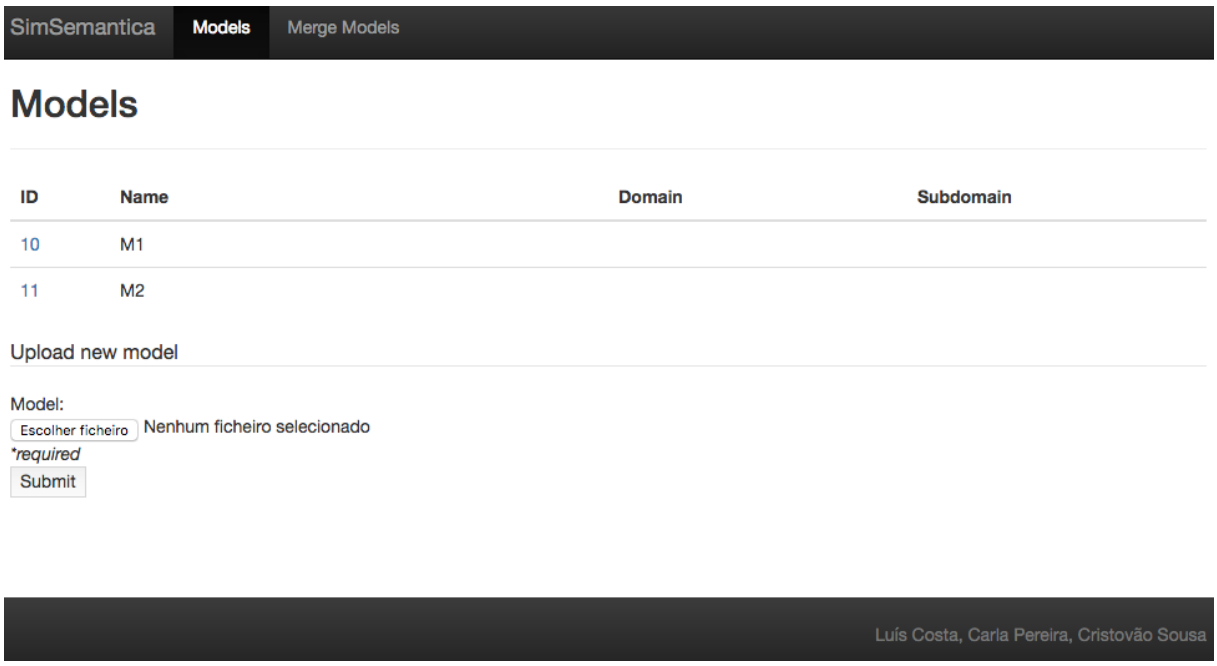


Figura 28: Listagem e *upload* de modelos no cliente

4.3.2 Integração de modelos conceptuais

A etapa de integração de modelos conceptuais é efetuada iterativamente e com o envolvimento dos especialistas. Este processo iterativo consiste na contínua análise de similaridade entre conceitos, após as ações tomadas em cada iteração. No início do processo, iteração 0, o especialista seleciona quais os modelos que pretende integrar (*merge*). A Figura 29 apresenta o ecrã de integração de modelos.

[23] *Wiki*: Software com funcionalidades de criar, editar e partilhar conhecimento de forma colaborativa (https://www.mediawiki.org/wiki/MediaWiki).
[24] *SMW*: Extensão que introduz a componente semântica na gestão do conhecimento, em sistemas *wikis* (https://www.semantic-mediawiki.org/wiki/Semantic_MediaWiki).

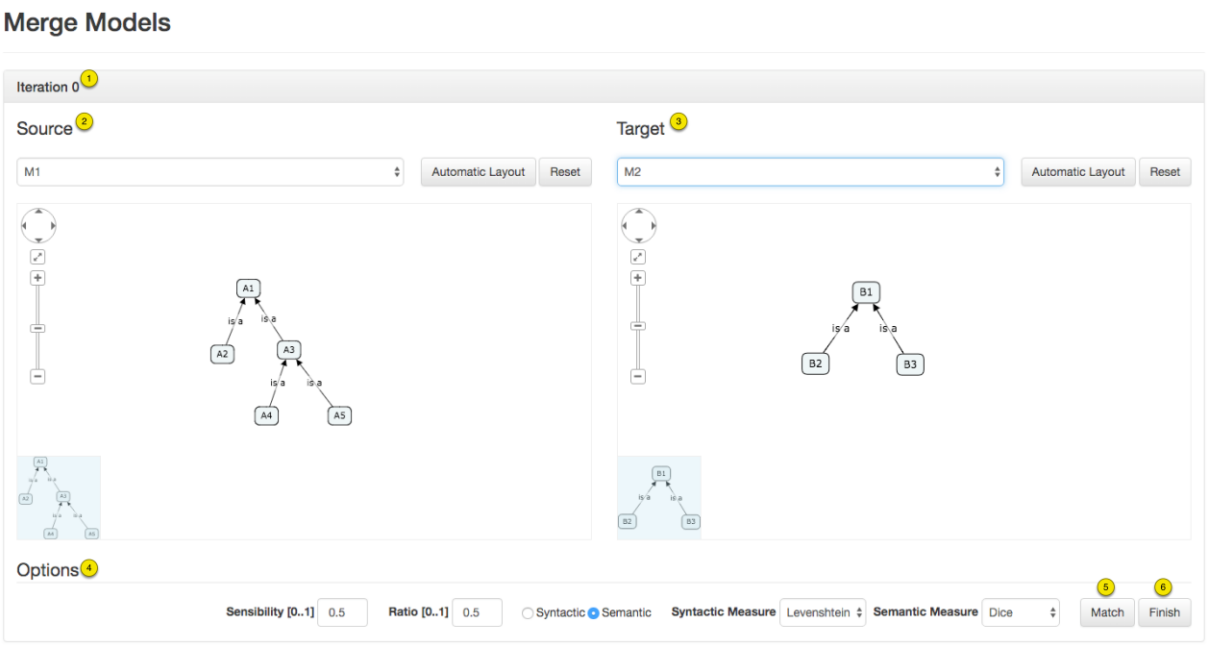


Figura 29: Início do processo de integração de dois modelos

Pontos importantes na vista de integração de modelos (*merge models*):

1. Indicação da iteração atual;
2. Seleção do modelo *source* (modelo de origem);
3. Seleção do modelo *target* (modelo de destino);
4. Parametrização das opções disponíveis para o cálculo de similaridade:
    a. **Sensibilidade (*sensibility*):** entre 0 e 1 para filtrar resultados dos *matches*. O valor mínimo 0 significa que não vai ser feita nenhuma filtragem, ou seja, todos os resultados do serviço serão apresentados ao especialista como *match*. O valor máximo 1 significa que apenas se pretende obter *matches* com o grau mais elevado de similaridade;
    b. **Rácio (*ratio*)**: valor entre 0 e 1 que serve de complemento à sensibilidade quando usadas as medidas sintáticas e semânticas para o cálculo de similaridade. O rácio define a maior diferença entre o valor de similaridade, obtido com a medida sintática, e o valor de similaridade, obtido aplicando a medida semântica. Por exemplo, para um $ratio = 0{,}5$, $similaridade\ sintática = 0{,}7$ e $similaridade\ semântica = 0{,}1$ o *match* era ignorado porque a diferença entre os dois valores é superior ao *ratio* $(0{,}7 - 0{,}1 = 0{,}6 \geq 0{,}5)$;
    c. ***Syntactic* ou *Semantic***: opção para indicar se o especialista pretende efetuar apenas uma análise sintática ou a conjugação das análises sintática e semântica;
    d. **Medida sintática *(syntactic measure)***: Seleção da medida sintática a utilizar na análise de similaridade; e

e. **Medida semântica *(semantic measure)***: Seleção da medida semântica a utilizar na análise de similaridade. Esta opção apenas estará disponível quando na opção *c* for definido que se pretende utilizar uma análise semântica.

5. Solicitação do serviço */api/match* (descrito na seção 4.4.1.1, página 71) para calcular o valor de similaridade entre os conceitos dos modelos e com as opções especificadas; e
6. Finalização do processo iterativo.

A introdução das opções anteriormente descritas (alíneas *a, b, c, d* e *e*, do ponto 4) tem como objetivo tornar o processo de integração adaptável às diferentes caraterísticas que podem existir nos modelos conceptuais. Selecionado diferentes opções, o resultado do processo varia, aumentando ou diminuindo o número de *matches* devolvidos pelo serviço. No cenário implementado no capítulo 5.2.1 (página 87), foi comprovado que os melhores resultados (aumento dos *matches* corretos e diminuição dos *matches* incorretos) são obtidos utilizando diferentes opções.

Após a invocação do serviço, uma lista de correspondências (*matches*) será obtida. Na Figura 30 é visível o resultado do serviço, que implementa o processo de similaridade, com as opções da Figura 29 anterior.

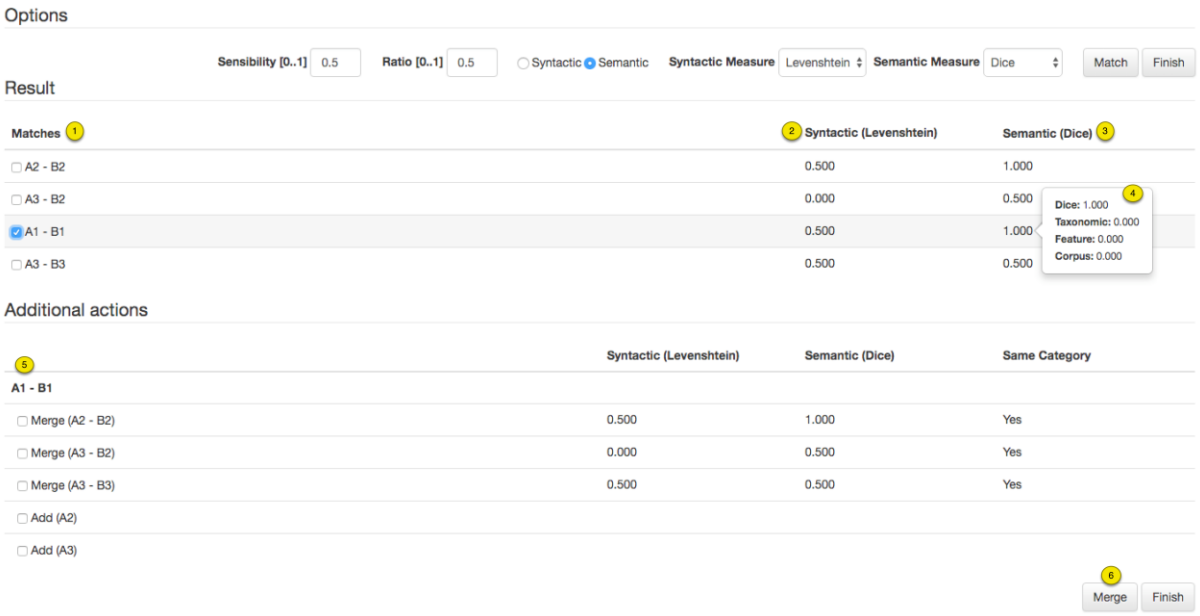


Figura 30: Resultado da ação *match*

Da Figura 30 anterior, obtém-se:

1. Lista de *matches* encontrados, aplicando o processo de similaridade semântica, com os modelos *source* e *target* selecionados;
2. Valor sintático do *match*, dado pela medida *Levenshtein*;
3. Valor semântico do *match,* dado pela medida *Dice - CRRM*;
4. *Tooltip* com o valor das restantes medidas aplicadas;
5. Lista de ações adicionais, após seleção de um *match*. É possível unir conceitos vizinhos (com indicação se possuem a mesma categoria de relação) ou adicionar conceitos do modelo *source* no modelo *target;* e
6. Efetuar *merge* dos *matches* selecionados. A seleção de *matches* está limitada pelas seguintes regras:
    a) Após seleção de 1 *match*, não é possível selecionar outro *match*, cujo valor de similaridade se altere pela junção do *match* anterior; e
    b) Não é possível selecionar um *match* cujo conceito *source* já esteja presente noutro *match* selecionado.

A operação *merge* avança para a próxima iteração, unindo os *matches* selecionados, que serão refletidos no modelo *target*. Ao finalizar o processo, quer seja por não haver mais *matches* possíveis ou por opção do especialista, o modelo *target* e o *source* passam a ser considerados um só, e o resultado da integração fica disponível no modelo final.

Por ser um processo iterativo, caso existam mais possibilidades de efetuar *merge*, o processo continua numa nova iteração, em que o especialista pode efetuar todos os procedimentos definidos anteriormente, mas, desta vez, aplicados ao resultado do *merge* da iteração imediatamente anterior. A Figura 31 e a Figura 32 seguintes, representam respetivamente, o resultado do *merge* da iteração 0 e o resultado final da integração, concluindo o processo na iteração 1.

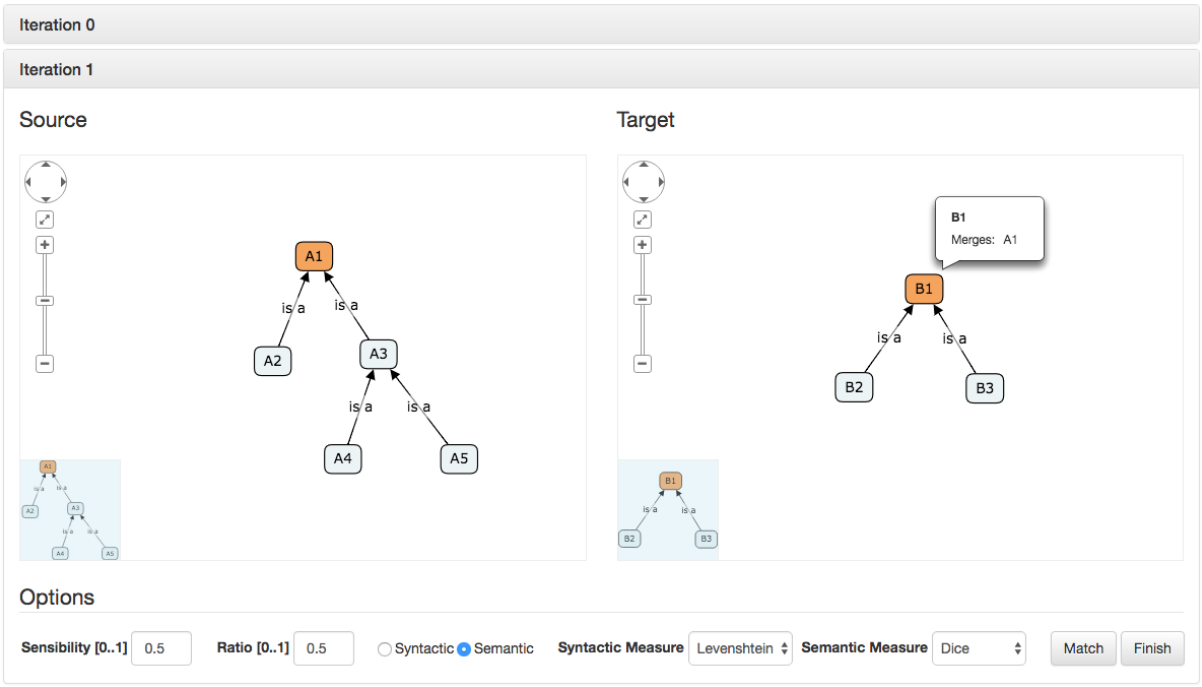


Figura 31: Resultado da ação *merge* numa iteração anterior

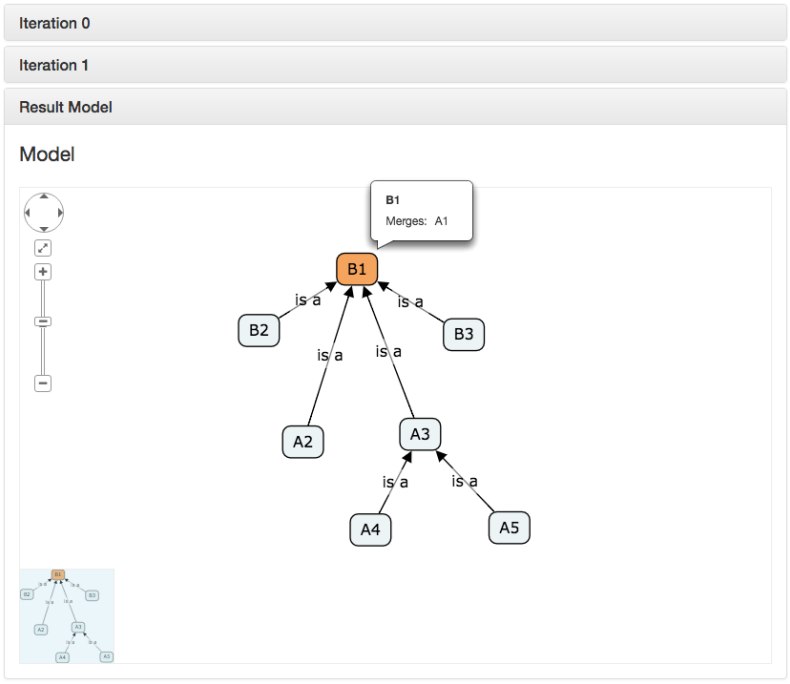


Figura 32: Resultado final da integração

Nesta última figura, é possível ver o modelo final, resultante do processo de integração. Nos conceitos que foram alvo da ação de *merge*, são indicados quais os conceitos, do modelo *source*, que foram integrados no respetivo conceito do modelo *target*.

4.4 Arquitetura da solução proposta

A solução criada, para cálculo de similaridade, foi desenvolvida para estar disponível na *web*, e assim, ser acessível por qualquer cliente. Para isso, foi implementada segundo uma arquitetura orientada a serviços, *REST Web Service*. Neste tipo de arquiteturas, a disponibilização de recursos via *URI*, permite que os clientes se abstraiam da implementação, e executem as ações pretendidas. Apenas precisam de comunicar, de acordo com a especificação de cada serviço.

Neste trabalho, foram introduzidas duas componentes. A primeira é o serviço, que implementa a abordagem proposta para o cálculo de similaridade entre modelos conceptuais. Na segunda, como complemento, foi desenvolvido um cliente para auxiliar todo o processo de integração de modelos conceptuais. A Figura 33 apresenta a arquitetura da solução final com as tecnologias utilizadas.

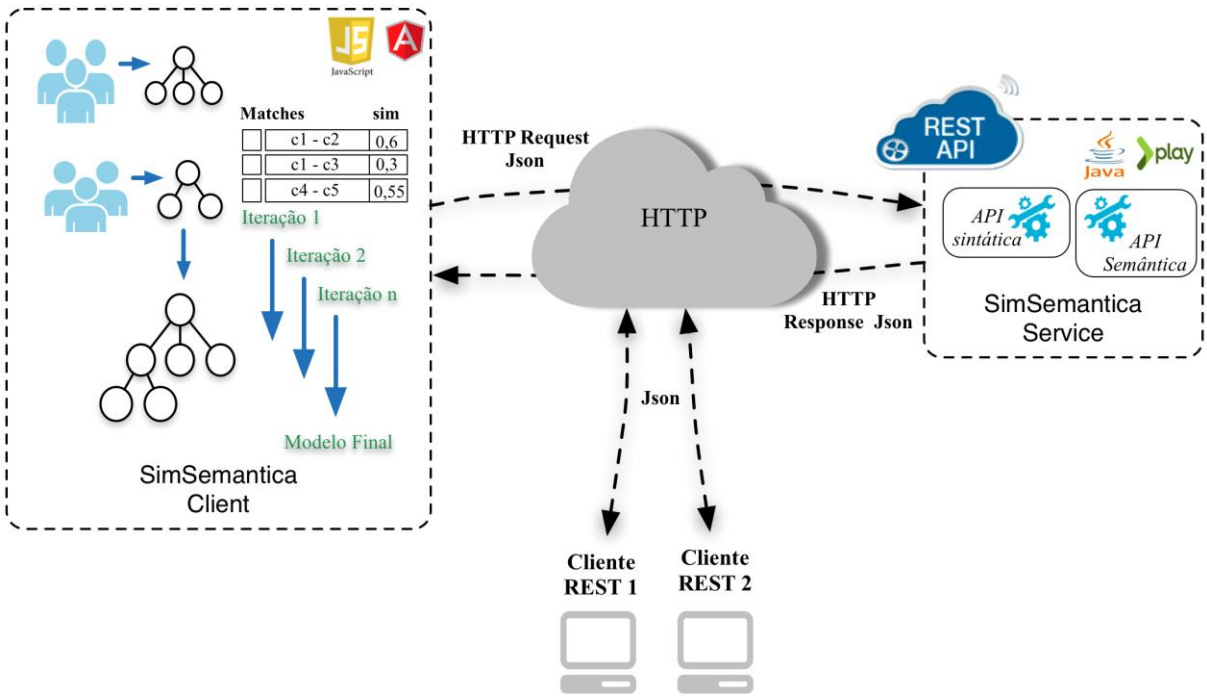


Figura 33: Arquitetura da solução proposta

O serviço, *SimSemanticaService*, foi desenvolvido em *Java*, utilizando a *framework* a *Play Framework*[25]. Esta *framework* é completamente vocacionada para a criação de *web services*, apresentando um conjunto variado de funcionalidades, que simplificam o desenvolvimento. Possui servidor *web* integrado, gestão de rotas e segue o padrão *model – view – controller* (MVC[26]).

A troca de mensagens entre os clientes e o serviço é realizada com recurso a objetos *JSON*. A especificação dos objetos de entrada e saída permite uma correta comunicação entre as partes.

[25] *Play Framework*: Framework Java para desenvolvimento de aplicações Web (https://www.playframework.com).
[26] *MVC*: Padrão de desenvolvimento, adotado na *Play Framework*, que tem como princípio a separação de uma aplicação em três camadas: modelo, vista e controlo (https://www.playframework.com/documentation/1.0/main).

No serviço implementado, residem as medidas utilizadas, que fazem uso de *API* externas para reutilizar as medidas já disponíveis. Da *API SimPack*, importa-se a medida *Levenshtein*, e da *API DKPro*, utilizam-se as medidas *Jaro*, *Cosine* e *MongeElkan*.

O cliente web, *SimSemanticaClient*, desenvolvido para ilustração do comportamento do serviço, na tarefa de integração de modelos conceptuais, é desenvolvido em *JavaScript*, utilizando a *framework AngularJS*[27].

Com o cliente, os especialistas podem visualizar todas as alterações do modelo, no decorrer das diversas iterações, e analisar os resultados das ações tomadas. Existe um conjunto de opções que podem ser configuradas (ver seção 4.3.2, página 65) para que seja possível invocar o serviço com o envolvimento dos especialistas. O processo de integração é iterativo, e em cada iteração o resultado dos *matches*, devolvidos pelo serviço, serão apresentados de forma tabelar, com o respetivo valor de similaridade sintática e semântica.

4.4.1 Serviço - SimSemanticaService

Para colocar em prática a proposta exposta anteriormente (seção 4.2, página 53), foi desenvolvido um serviço *web*, com o intuito de ser invocado a partir de qualquer cliente.

4.4.1.1 Recurso */api/match*

O recurso */api/match* é responsável por calcular a similaridade entre conceitos do modelo *source* e conceitos do modelo *target*, e devolver a lista de *matches* encontrados, que respeitem as opões definidas.

4.4.1.1.1 Especificação

Para invocar este recurso deve ser feito um pedido *REST HTTP POST* com um objeto *JSON* de entrada. Este objeto de entrada representa os modelos e as opções que serão alvo de análise pelo processo de similaridade.

A resposta deste recurso também é realizada no formato *JSON*, e contém uma lista de *matches*, encontrados durante a aplicação do processo de similaridade semântica. A Tabela 11 resume a especificação do recurso */api/match*.

[27] *AngularJS*: Framework *JavaScript*, desenvolvida pela *Google*, para a criação de clientes *web* (https://angularjs.org).

Tabela 11: Especificação do recurso */api/match*

URI	http://simsemantica.lmcosta.pt:9100/api/match
Método	*POST*
Content-type	*application/json*

Entrada	{
	  "source": {
	  "name": "string – model name",
	  "concepts": [],
	  "relations": []
	  },
	  "target": {
	  "name": "string – model name",
	  "concepts": [],
	  "relations": []
	  },
	  "options": {
	  "semantic": "true | false",
	  "sensibility": "double [0 .. 1]",
	  "ratio": "double [0 .. 1]",
	  "syntacticMeasure": "Levenshtein | Jaro | MongeElkan",
	  "semanticMeasure": "Dice | Corpus | Taxonomic | Feature"
	  },
	  "iteration": "int [0 .. n]",
	  "maps": [
	  {
	  "source": "string - source concept id",
	  "target": "string - target concept id"
	  }
	  ]
	}

Saída	[
	  {
	  "source": {
	  "id": "string - concept id",
	  "name": "string - concept name"
	  },
	  "target": {
	  "id": "string - concept id",
	  "name": "string - concept name"
	  },
	  "syntactic": {
	  "measure": "string - syntactic measure",
	  "value": "double [0 .. 1]"
	  },
	  "semantic": {
	  "measure": "string - semantic measure",
	  "value": "double [0 .. 1]"
	  }
	  }
	]

Os objetos de entrada e saída do serviço de similaridade semântica encontram-se detalhados na Tabela 12 e na Tabela 13, respetivamente.

Tabela 12: Objeto de entrada aceite pelo serviço de similaridade semântica

Propriedade	Descrição
source	Modelo conceptual de origem, composto pelo nome, lista de conceitos e lista de relações.
target	Modelo conceptual de destino, composto pelo nome, lista de conceitos e lista de relações.
options	Opções definidas pelo especialista a utilizar na análise de similaridade entre conceitos. • *Semantic*: Ativa (*true*) ou não (*false*) a fase 3 – análise semântica, do processo para cálculo de similaridade semântica. • *sensibility*: valor decimal entre 0 e 1. • *ratio*: valor decimal entre 0 e 1. • *syntacticMeasure*: nome da medida sintática a utilizar. Pode ser: *Levenshtein*, *Jaro* ou *MongeElkan* • *semanticMeasure*: nome da medida semântica a utilizar. Pode ser: *Dice*, *Corpus*, *Taxonomic* ou *Feature*
iteration	Valor inteiro que indica a iteração atual.
maps	Lista de integrações efetuadas pelo especialista até a iteração atual. Contêm referências para os pares de conceitos integrados.

Tabela 13: Objeto de saída do serviço de similaridade semântica

Propriedade	Descrição
source	Conceito do modelo de origem.
target	Conceito do modelo de destino.
syntactic	Valor decimal entre 0 e 1, com o resultado da medida sintática entre *source* e *target*.
semantic	Valor decimal entre 0 e 1, com o resultado da medida semântica entre *source* e *target*.

4.4.1.1.2 Descrição

O recurso */api/match* é responsável pelo cálculo do grau de similaridade semântica entre conceitos, seguindo a abordagem proposta neste trabalho. Em primeiro lugar, efetua-se a inicialização do modelo de dados. Posteriormente, as fases 1, 2 e 3, como apresentadas no processo de similaridade, são executadas e, no final, a lista de *matches* resultante é devolvida. A Tabela 14 apresenta, no formato de pseudo-código, os passos principais do recurso */api/match*, desde o pedido de análise de similaridade até à resposta com a lista de *matches*.

Tabela 14: Implementação geral do serviço

```
1.  // Inicializar Modelos
2.  Model source = ModelUtils.init(requestInfo.getSource());
3.  Model target = ModelUtils.init(requestInfo.getTarget());
4.  // Fase 1
5.  Phase1.run(source, target, language);
6.  // Fase 2
7.  Results phase2Results = Phase2.run(source, target, maps);
8.  phase2Results.filterPhaseResults(sensibility);
9.  results.updateMatches(phase2Results);
10. // Terminar processo se apenas for definido para fazer análise sintática
11. if (!semantic) {
12.   return results;
13. }
14. // Fase 3
15. Results phase3Results = Phase3.run(source, target, maps);
16. phase3Results.filterPhaseResults(sensibility);
17. results.updateMatches(phase2Results, phase3Results);
18. return results;
```

Inicialização

As linhas 2 e 3 da Tabela 14 são responsáveis pela implementação da atividade de preparação, definida na proposta de abordagem para o cálculo de similaridade semântica. O objeto de entrada é

convertido num modelo de dados, constituído por três classes: Modelo (*Model*), Conceito (*Concept*) e Relação (*Relation*). A Figura 34 representa, em formato de diagrama de classes, os três objetos e as respetivas relações.

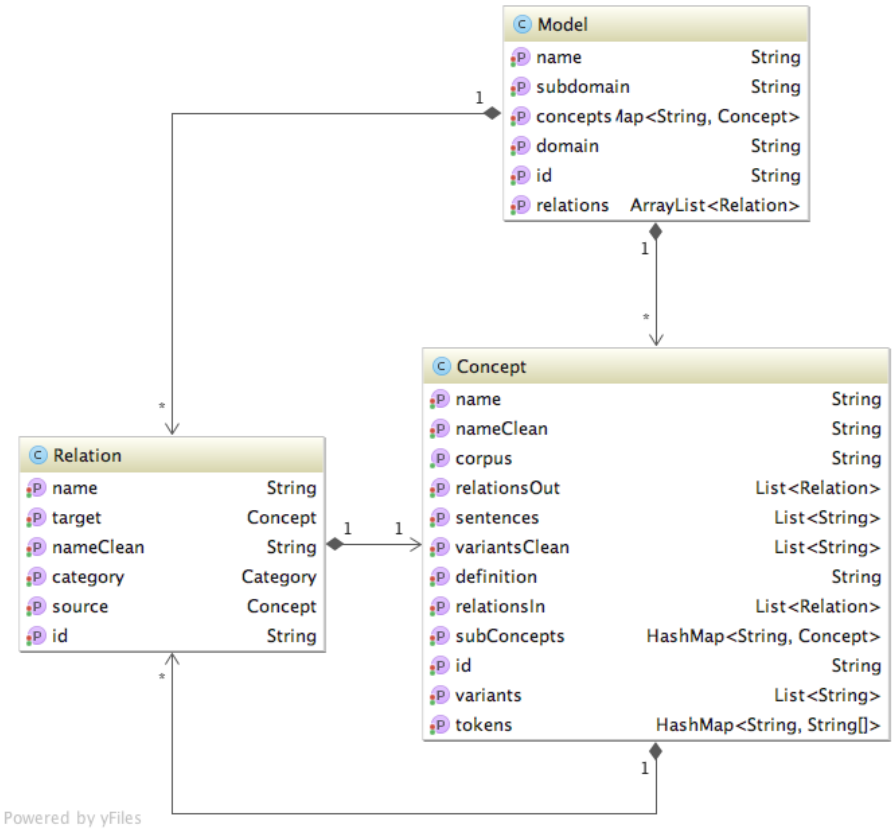


Figura 34: Diagrama classes com as entidades modelo, conceito e relação

O modelo de dados é instanciado com uma lista de conceitos e uma lista de relações. Cada conceito criado, além da informação recebida no objeto de entrada (*id, name, definition, variants* e *sentences*), contém um conjunto de novas propriedades para suportar as três fases do processo de similaridade. Essas propriedades são:

a) ***nameClean*, *variantsClean* e *tokens***: Usadas para armazenar o resultado da normalização;
b) ***subConcepts***: Armazena a referência dos sub-conceitos, caso existam relações taxonómicas envolvidas (conceitos subsumidos); e
c) ***relationsIn* e *relationsOut***: Guardam as referências para relações de entrada e saída, respetivamente. São propriedades úteis para, de uma forma rápida, obter os conceitos vizinhos, ligados de forma direta ao conceito. Como exemplo, a Figura 35, apresenta visualmente uma relação de entrada e uma relação de saída.

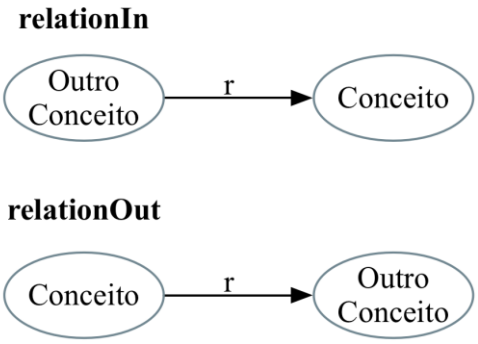


Figura 35: Representação de uma relação de entrada e de uma relação de saída

Um objeto do tipo relação vai ser instanciado com *id*, *name*, *source* e *target*. A propriedade *source* representa o conceito de origem da relação. Enquanto que, a propriedade *target*, representa a direção da relação. A propriedade *nameClean,* tal como nos conceitos, tem como finalidade armazenar o nome da relação após aplicação da fase 1 - Normalização.

Nas relações existe, ainda, a propriedade *category*. É nesta propriedade que, após aplicada a fase 1 do processo de similaridade semântica, é armazenada a categoria atribuída à relação. Esta categoria, resulta de uma ontologia de relações CRRM [2] e tem como finalidade servir de elemento preponderante na análise semântica aplicada à estrutura conceptual.

Fase 1: Normalização

Nesta primeira fase, para normalizar os nomes atribuídos a conceitos e relações, o serviço aplica algoritmos de:

- *stemming* recorrendo à API *LucenePorterStemmer*;
- *tokenizing* utilizando a API Apache OpenNLP; e
- *stop words*.

Com base no nome das relações, é realizada uma categorização, recorrendo à ontologia CRRM [2]. Por exemplo, existe uma categoria denominada de *Hierarchy* que é composta pelos nomes de relações *is-a*, *type of*, etc. Qualquer relação dos modelos em análise, com nome *is-a* ou *type of*, será categorizada como *Hierarchy*. Se existirem relações em que o nome não se enquadre em nenhuma categoria defina na ontologia, a categoria atribuída será *Default,* ou seja, categoria desconhecida para o CRRM.

Em seguida, a Figura 36, apresenta um complemento ao modelo de dados, com o diagrama de classes para Relação, Categoria e CRRM.

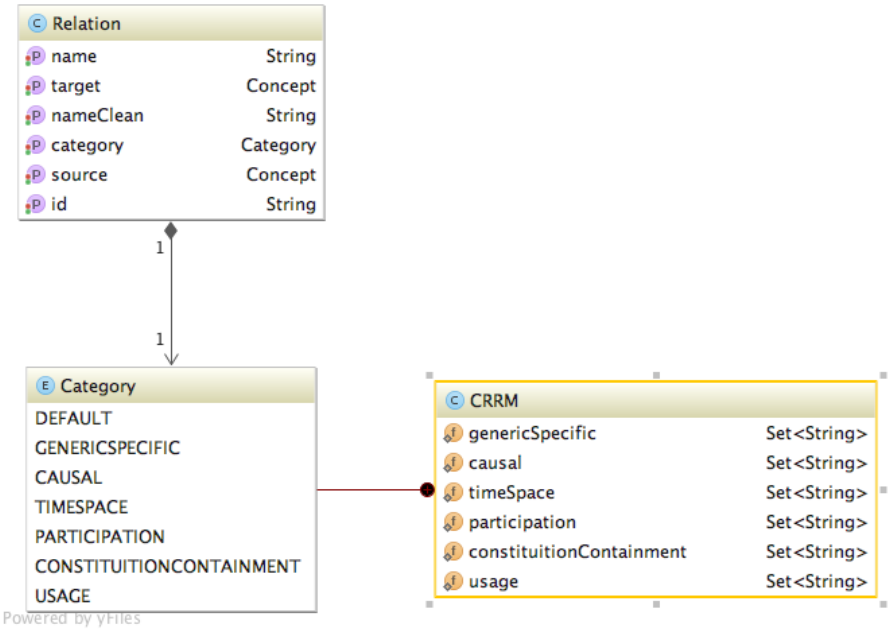


Figura 36: Diagrama de classes com as entidades relação, categoria e CRRM

Fase 2: Análise Sintática

O serviço, nesta segunda fase, calcula o valor da similaridade entre conceitos, aplicando medidas sintáticas. Tal como referido na proposta de abordagem para o cálculo de similaridade semântica (seção 4.2, página 53), as medidas usadas são: *Levenshtein*, *Jaro* e *MongeElkan*. A Tabela 15 resume a implementação do serviço para a fase 2.

Tabela 15: Excerto de código para cálculo de similaridade sintática – fase 2

```
1.  //source model concepts
2.  for (Concept sourceConcept : source.getConcepts().values()) {
3.    //target model concepts
4.    for (Concept targetConcept : target.getConcepts().values()) {
5.      if (!conceptsToIgnore.contains(sourceConcept.getId())) {
6.        Map result = phase2.getSimilarity(sourceConcept, targetConcept);
7.        results.addPhaseResult(sourceConcept, targetConcept, result);
8.      }
9.    }
10. }
```

À exceção dos conceitos do modelo *source*, que já foram integrados em iterações anteriores (conceitos armazenados na lista *conceptsToIgnore* linha 5), é efetuada uma comparação entre pares de conceitos, usando medidas sintáticas. O método *getSimilarity* (linha 6), devolve o maior valor de similaridade, resultante da comparação do nome e das variantes dos dois conceitos, aplicando as três medidas (*Levenshtein*, *Jaro* e *MongeElkan*).

A aplicação destas medidas foi realizada recorrendo a bibliotecas disponíveis para o cálculo de similaridade entre palavras. Para a medida *Levenshtein*, utilizou-se o SimPack, e para as outras duas medidas, *Jaro* e *MongeElkan*, utilizou-se o DKPro. A utilização de API's externas, com implementações

dos algoritmos das medidas testadas, permite garantir que o resultado seja válido. No final deste processo, a análise sintática está efetuada e o serviço prossegue para a próxima fase.

Fase 3: Análise Semântica

A última fase descrita no processo para cálculo de similaridade semântica, é implementada pelo serviço, utilizando medidas semânticas para obter o valor de similaridade. Até o início desta terceira fase, o valor de similaridade era apenas baseado na sintática, presente nos nomes dos conceitos. Nesta fase, com a utilização de medidas semânticas, o valor de similaridade é obtido, considerando a estrutura conceptual, as caraterísticas e o *corpus* de cada conceito.

Para cada par de conceitos válidos para comparação (tal como na fase anterior, os conceitos válidos são aqueles que ainda não foram integrados) serão aplicadas as medidas: *Dice*, *Corpus*, *Taxonomic* e *Feature*. Na Tabela 16 é apresentado o excerto de código, executado pelo serviço, para implementação da fase 3, do processo proposto para o cálculo de similaridade entre conceitos.

Tabela 16: Excerto de código para cálculo da similaridade semântica – fase 3

```
1.  Results results = new Results();
2.  Phase3 phase3 = new Phase3();
3.  //create taxonomy
4.  phase3.createTaxonomy(source, target, maps);
5.  //source model concepts
6.  for (Concept sourceConcept : source.getConcepts().values()) {
7.    //target model concepts
8.    for (Concept targetConcept : target.getConcepts().values()) {
9.      if (!conceptsToIgnore.contains(sourceConcept.getId())) {
10.       Map  result = phase3.getSimilarity(sourceConcept, targetConcept, source, target,
    maps);
11.       results.addPhaseResult(sourceConcept, targetConcept, result);
12.     }
13.   }
14. }
15. return results;
```

Em primeiro lugar, é criada uma taxonomia, com base nos modelos de entrada e na lista de conceitos integrados em iterações anteriores (lista de *maps*) pelos especialistas. Esta taxonomia vai servir como recurso para a aplicação da medida taxonómica implementada. Os passos para a criação da taxonomia são:

1º Obter todas as relações hierárquicas existentes nos dois modelos (*source* e *target*);

2º Criar pontos de ligação entre os modelos, com base na lista de *maps*;

3º Criar o grafo que representa a taxonomia; e

4º Adicionar como raiz do grafo o conceito *Root*. A medida taxonómica Wu e Palmer implica a existência do conceito *Root* para o cálculo do grau de similaridade entre dois conceitos presentes na taxonomia (ver seção 3.1.1.1.4, página 29).

A criação de uma taxonomia, gerada a partir dos modelos *source* e *target*, tem como pressupostos a existência de relações hierárquicas (1º passo para a criação da taxonomia) nos dois modelos e a

definição de uma lista de *maps* (2º passo), com os conceitos integrados em iterações anteriores. Sem estes dois pressupostos definidos, é impossível obter uma taxonomia que represente os dois modelos.

Considerando todas as relações válidas (relações taxonómicas) e os conceitos unidos pelos especialistas, pode-se fazer elos de ligação entre modelos e construir uma taxonomia única. A Figura 37 demonstra como chegar a uma taxonomia, a partir de dois modelos.

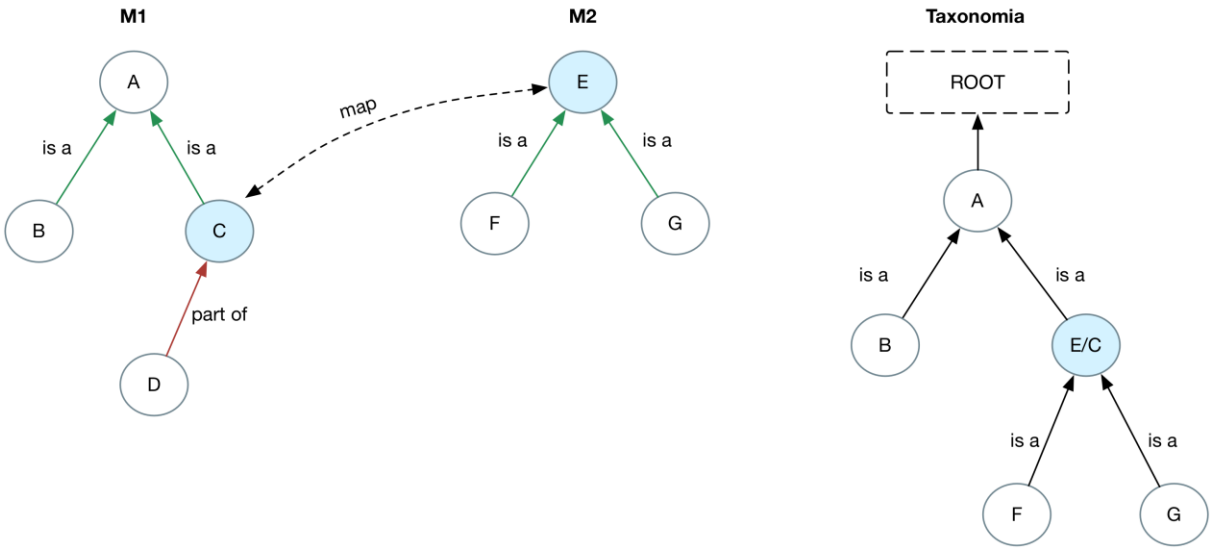


Figura 37: Criação da taxonomia a partir de dois modelos

Para os modelos *M1* e *M2*, apresentados na Figura 37, as relações válidas são:

a) ***M1*:** B *is a* A; C *is a* A; e

b) ***M2*:** F *is a* E; G *is a* E.

A relação *D part of C*, do modelo *M1*, não é considerada porque não é uma relação taxonómica. O conceito *C*, do modelo *M1*, e o conceito *E*, do modelo *M2*, foram integrados anteriormente por ação do especialista. Atualizando o *source* da relação C *is a* A para E *is a* A, chega-se ao resultado da taxonomia representada na Figura 37, demonstrada anteriormente.

A construção da taxonomia, numa estrutura em grafo, é feita recorrendo à biblioteca *Java JGraphT*[28]. Esta biblioteca permite a utilização de algoritmos para obter o menor caminho entre conceitos, tal como o *Dijkstra*[29], para facilitar a contagem de relações taxonómicas (*edges*) entre conceitos (*vertex*).

Medida Taxonómica

A aplicação da medida taxonómica tem como objetivo perceber o nível de similaridade entre conceitos, de acordo com uma taxonomia. Quanto mais afastados dois conceitos estiverem numa

[28] ***JGraphT***: http://jgrapht.org
[29] ***Dijkstra*:** algoritmo que indica o caminho mais curto entre nós de um grafo.

taxonomia, menos similares eles serão. Apresenta-se de seguida a Tabela 17, que mostra o excerto de código da implementação da medida taxonómica Wu e Palmer.

Tabela 17: Implementação da medida taxonómica

```
1.  List sourcePathToRoot = getPath(taxonomy, sourceConcept.getId(), ROOT_VERTEX);
2.  List targetPathToRoot = getPath(taxonomy, targetConcept.getId(), ROOT_VERTEX);
3.  //LCS
4.  String LCS = getLCS(taxonomy, sourcePathToRoot, targetPathToRoot);
5.  LCS = LCS != null ? LCS : ROOT_VERTEX;
6.  //Paths
7.  List sourcePathToLCS = getPath(taxonomy, sourceConcept.getId(), LCS);
8.  List targetPathToLCS = getPath(taxonomy, targetConcept.getId(), LCS);
9.  List LCSpathToRoot = getPath(taxonomy, LCS, ROOT_VERTEX);
10. //result
11. Double result = ((double) (2 * LCSpathToRoot.size()) / (double) (sourcePathToLCS.size()
     + targetPathToLCS.size() + (2 * LCSpathToRoot.size())));
```

A medida taxonómica Wu e Palmer considera as menores distâncias de cada conceito (*vertex*) para o conceito comum mais próximo - *least common subsumer* (LCS) - e de LCS para o *vertex Root*. As menores distâncias são obtidas a partir do algoritmo *Dijkstra*, disponível na biblioteca *JGraphT*.

A título demonstrativo, utilizando os modelos da Figura 37, apresentada anteriormente, e os conceitos *B*, do modelo *M1*, e *F*, do modelo *M2*, o LCS é o conceito *A*, visível na taxonomia. Num caso extremo, o conceito LCS pode ser o *Root*, por este fato é que se torna obrigatória a introdução elemento *Root* como raiz da taxonomia.

Em resumo, a medida Wu e Palmer, considera as seguintes variáveis:

a) Distância do conceito 1 para o conceito LCS;
b) Distância do conceito 2 para o conceito LCS; e
c) Distância do conceito LCS para o *Root*.

Depois de obtidas as três distâncias, aplica-se a fórmula (4), proposta por Wu e Palmer e descrita na análise de medidas (ver seção 3.1.1.1.4, página 29).

Medida de *Dice - CRRM*

A medida *Dice - CRRM* baseia-se na estrutura conceptual (CRC) para calcular o seu grau de similaridade. Considera o número de relações da mesma categoria e o número total de relações (*degree*). O grau de similaridade é obtido aplicando a instrução da Tabela 18.

Tabela 18: Excerto para o cálculo da similaridade usando *Dice - CRRM*

```
1.  Double result = ((double) (2 * commonRelations) / (double) (degSourceConcept + degTarge
    tConcept));
```

Dado dois conceitos em análise, *C1* e *C2*, as relações que ligam estes conceitos são obtidas utilizando as propriedades *relationsIn* (relações que apontam para o conceito) e *relationsOut* (relações com origem no conceito). Somando as relações de entrada e saída, obtém-se o grau de relações de cada conceito (*degree*). Para obter o total de relações comuns, somam-se todas as relações da mesma

categoria, de entrada ou de saída, que ligam os conceitos. O resultado final é dado pela medida de *Dice - CRRM*, apresentada na fase 3 da proposta (seção 4.2.4, página 59).

Medida IC (*Corpus*)

A aplicação desta medida permite obter o grau de semelhança entre dois conceitos pelo cálculo da quantidade de informação (IC) que eles partilham. As informações extraídas de cada conceito para utilização na medida baseada no *corpus* são:

a) Definição;
b) Variantes; e
c) Frases.

Depois da construção do *corpus* de cada conceito, utiliza-se a medida *Cosine* implementada em DKPro [92]. Esta medida constrói um vetor com o número de ocorrências de cada palavra no *corpus* e, quanto mais palavras dois *corpus* partilharem, maior será o grau de similaridade entre os conceitos.

Exemplo: pretende-se analisar a similaridade entre o conceito *carro* e o conceito *automóvel*. O *corpus* de cada conceito é o seguinte:

a) ***Corpus* do conceito carro:** "Veículo com quatro rodas. Meio de transporte."
b) ***Corpus* do conceito automóvel:** "Meio de transporte com quatro rodas."

Fazendo o levantamento de todas as palavras distintas dos dois *corpus*, e contando as frequências de cada palavra, em cada *corpus*, chega-se ao vetor de ocorrências exposto na Tabela 19.

Tabela 19: Vetor de ocorrências de palavras num corpus

	Veículo	Com	Quatro	Rodas	Meio	De	Transporte
Corpus Carro	1	1	1	1	1	1	1
Corpus Automóvel	0	1	1	1	1	1	1

Uma vez encontradas as frequências, pode-se calcular o grau de similaridade, usando a fórmula (21), descrita no estudo de medidas (seção 3.1.2.7, página 38). Neste exemplo, as variáveis que pertencem à fórmula mencionada apresentam os seguintes valores:

- Produto escalar dos dois vetores = 6
- Raiz quadrada do total de palavras existentes no *corpus* carro = 2,65
- Raiz quadrada do total de palavras existentes no *corpus* automóvel = 2,45

$$Sim(carro, automóvel) = \frac{6}{2{,}65 \times 2{,}45} = 0{,}92$$

Medida baseada na análise da vizinhança (conceitos vizinhos comuns que partilham relações conceptuais do mesmo tipo)

Na medida baseada na vizinhança, o grau de similaridade entre dois conceitos, é obtido verificando o número de conceitos vizinhos comuns e que partilham o mesmo tipo de relação. É

atribuído um peso equitativo a cada comparação, ou seja, é tão importante haver conceitos vizinhos comuns como relações conceptuais do mesmo tipo. Com isto, evita-se que seja atribuído um elevado grau de similaridade entre dois conceitos, que apresentem vizinhos comuns, mas ligados por relações de diferentes categorias. A Tabela 20 representa o excerto da implementação da medida desenvolvida, baseada na vizinhança dos conceitos em análise.

Tabela 20: Implementação da medida baseada na vizinhança

```
1.  Map sourceConceptNeighbors = getNeighborsConcepts(sourceConcept);
2.  Map targetConceptNeighbors = getNeighborsConcepts(targetConcept);
3.  Map commons = getCommonsNeighborsConcepts(sourceConceptNeighbors, targetConceptNeighbor
    s);
4.  Map commonsRelations = getCommonsRelations(commons);
5.  int totalNeighbors = sourceConceptNeighbors.size() + targetConceptNeighbors.size();
6.  int totalCommonNeighbors = commons.size;
7.  int totalRelations = commonsRelations.size();
8.  Double result = (((double) totalCommonNeighbors / (double) totalNeighbors) * CONCEPT_WE
    IGHT) + (((double) totalCommonRelations / (double) totalRelations) * RELATION_WEIGHT);
```

Esta medida, tal como a taxonómica, parte do pressuposto que existe a informação de que dois conceitos vizinhos são iguais. Então, conceitos vizinhos serão considerados comuns se:

a) Existir um mapeamento de conceitos (integração feita pelo especialista e refletida na lista de *maps*); ou

b) O conceito for genérico, de um conceito especifico integrado (subsunção de conceitos).

A Figura 38, apresenta um exemplo de como se procede para verificar se os conceitos vizinhos são iguais.

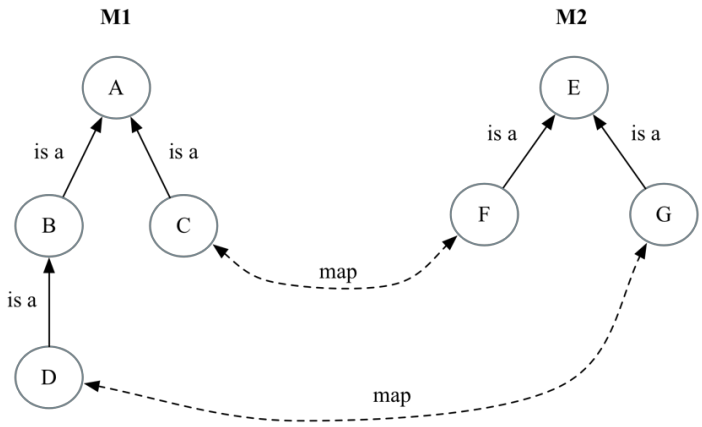


Figura 38: Comparação de conceitos na análise baseada em vizinhança

Analisando os conceitos vizinhos de *A*, do modelo *M1*, e *E*, do modelo *M2*, verifica-se que:

- *C* e *F* são o mesmo conceito (foram integrados pelo especialista numa iteração anterior); e
- O conceito *B*, de forma direta, não é comum nem a *F* nem a *G* do modelo *M2*, contudo, o conceito *B* é o conceito genérico de *D* e este foi unido com *G*. Desta forma, *A* e *E* possuem 4 vizinhos comuns (*B, C, F* e *G*), em 4 possíveis, e 4 relações comuns (*B is a A; C is a A;*

F is a E; G is a E), também em 4 possíveis. Deste modo, aplicando a fórmula (26), descrita anteriormente na seção 4.2.4, página 63, o valor de similaridade será o máximo possível, ou seja, similaridade = 1.

Resposta do recurso *`/api/match`*

Finalmente, depois de calculados os valores de similaridade entre pares de conceitos do modelo *source* e *target*, aplicando diferentes medidas, os resultados com os *matches* encontrados serão devolvidos. Para um *match* ser considerado válido, precisa apresentar um valor de similaridade maior ou igual à opção *sensibility* definida e a diferença entre o valor de similaridade sintática e semântica não pode exceder o *ratio* definido pelo especialista.

4.4.1.2 Recurso *`/api/match/selections`*

Após o especialista selecionar um *match* no cliente *web* desenvolvido, é automaticamente invocado o recurso */api/match/selections*, para perceber qual o grau de semelhança entre os conceitos vizinhos.

Neste recurso, o cálculo de similaridade segue a mesma lógica do recurso descrito anteriormente, mas apenas considera o conjunto de conceitos vizinhos dos *matches* selecionados. É acrescentada a cada *match* a indicação se as relações, dos conceitos comparados, possuem a mesma categoria. Esta indicação permite informar aos especialistas como proceder com os conceitos que estão ligados diretamente ao *match* selecionado. O especialista pode decidir se, além de integrar o *match* selecionado, quer também integrar ou adicionar conceitos vizinhos diretamente no modelo *target*.

4.4.1.2.1 Especificação

Para invocar este recurso deve ser feito um pedido *REST HTTP POST* com um objeto *JSON* de entrada. O recurso responde com um objeto *JSON*, contendo a lista de correspondências (*matches*) encontradas durante a aplicação do processo de similaridade semântica. A Tabela 21 resume a especificação do recurso */api/match/selections*.

Tabela 21: Especificação do recurso */api/match/selections*

URI	http://simsemantica.lmcosta.pt:9100/api/match/selections
Método	*POST*
Content-type	*application/json*

Entrada	{
	"source": {
	"name": "string – model name",
	"concepts": [],
	"relations": []
	},
	"target": {
	"name": "string – model name",
	"concepts": [],
	"relations": []
	},
	"options": {
	"semantic": "true | false",
	"sensibility": "double [0 .. 1]",
	"ratio": "double [0 .. 1]",
	"syntacticMeasure": "Levenshtein | Jaro | MongeElkan",
	"semanticMeasure": "Dice | Corpus | Taxonomic | Feature"
	},
	"iteration": "int [0 .. n]",
	"maps": [
	{
	"source": "string - source concept id",
	"target": "string - target concept id"
	},
	"matches": [
	{
	"source": "string - source concept id",
	"target": "string - target concept id"
	}
	]
	}

Saída	[
	{
	"source": {
	"id": " string - concept id ",
	"name": " string - concept name "
	},
	"target": {
	"id": " string - concept id ",
	"name": " string - concept name "
	},
	"matches": [],
	"add": []
	}
	]

Os objetos de entrada e saída do recurso */api/match/selections* encontram-se detalhados na Tabela 22 e na Tabela 23, respetivamente.

Tabela 22: Objeto de entrada aceite pelo recurso */api/match/selections*

Propriedade	Descrição
source	Modelo conceptual de origem, composto pelo nome, lista de conceitos e lista de relações.
target	Modelo conceptual de destino, composto pelo nome, lista de conceitos e lista de relações.
options	Opções definidas pelo especialista a utilizar na análise de similaridade entre conceitos. • *Semantic*: Ativa (*true*) ou não (*false*) a fase 3 – análise semântica, do processo para cálculo de similaridade semântica. • *sensibility*: valor decimal entre 0 e 1. • *ratio*: valor decimal entre 0 e 1. • *syntacticMeasure*: nome da medida sintática a utilizar. Pode ser: *Levenshtein*, *Jaro* ou *MongeElkan* • *semanticMeasure*: nome da medida semântica a utilizar. Pode ser: *Dice*, *Corpus*, *Taxonomic* ou *Feature*
iteration	Valor inteiro que indica a iteração atual.
maps	Lista de integrações efetuadas pelo especialista até à iteração atual. Contêm referências para os pares de conceitos integrados.
matches	Lista com os matches selecionados na iteração atual.

Tabela 23: Objeto de saída do recurso */api/match/selections*

Propriedade	Descrição
source	Conceito do modelo de origem.
target	Conceito do modelo de destino.
matches	Lista de novos *matches* resultantes da análise de similaridade aplicada aos *matches* selecionados no objeto de entrada.
add	Lista de possíveis conceitos vizinhos ao match selecionado, no objeto de entrada, que o especialista pode adicionar no modelo target.

No capítulo seguinte a abordagem proposta será avaliada com base em dois cenários.

5 AVALIAÇÃO COM BASE EM CENÁRIOS

Neste capítulo, pretende-se validar o processo e avaliar o serviço de similaridade desenvolvido neste trabalho. Para isso, foram desenhados dois cenários de aplicação e, com base nesses cenários, o serviço foi testado. A validação é feita com base na qualidade dos resultados, aplicando as medidas *precision* e *recall*, e utilizando os resultados de referência de cada cenário.

No primeiro cenário, são utilizados um *dataset* comum (conjunto de modelos disponibilizados para efeitos de teste e de avaliação das ferramentas) e os resultados de referência, disponíveis na iniciativa *Ontology Alignment Evaluation Initiative - OAEI*[30], ocorrida em 2015. No segundo cenário, pretende-se avaliar a aplicabilidade do serviço de similaridade semântica na integração de modelos conceptuais, desenvolvidos de forma colaborativa. São utilizados dois modelos conceptuais reais, resultantes de atividades de conceptualização de dois grupos existentes no projeto Forsys[31]. Neste último cenário, as medidas *precision* e *recall* são calculadas, considerando os resultados obtidos pelo serviço de similaridade semântica e pela lista de conceitos equivalentes assinalados pelos especialistas de domínio.

5.1 Medidas de avaliação

As diversas ferramentas existentes para o problema de integração de modelos possuem variadas abordagens, originando diferentes conjuntos de correspondências (*matches*) nos resultados. Aos conjuntos de correspondências, é dado o nome de alinhamento, que indicam os conceitos equivalentes entre dois modelos. Como forma de avaliar os resultados das variadas abordagens existentes é necessário comparar os *matches* de um alinhamento obtido por uma ferramenta, com os *matches* do resultado de referência [105].

Na área de *ontology matching* usam-se, preferencialmente, as medidas *precision* e *recall* para avaliar a qualidade dos alinhamentos resultantes da comparação de duas ontologias [105]. Como este trabalho também incide na comparação de dois modelos e o resultado é um conjunto de *matches*, as medidas *precision* e *recall* serão igualmente usadas para avaliação dos resultados obtidos.

Na Figura 39 é demonstrado como avaliar um alinhamento *A* (resultante de uma ferramenta de integração), utilizando um alinhamento de referência *R* (resultado proposto por especialistas).

[30] http://oaei.ontologymatching.org/2015/conference/index.html
[31] http://www.cost.eu/COST_Actions/fps/FP0804

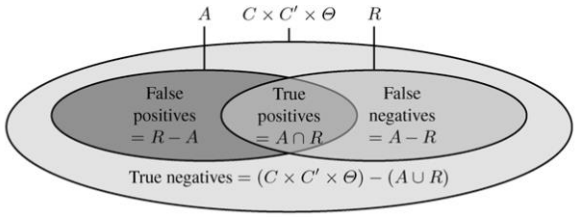


Figura 39: Relação entre dois alinhamentos no cálculo da *precision* e *recall* [105]

Da Figura 39 anterior, é importante reter três variáveis que estão na base do cálculo da *precision* e *recall*. Estas variáveis são [105]:

- *False positives (FP): Matches* no alinhamento *A* que não estão presentes no alinhamento de referência *R*, ou seja, *matches* assinalados como similares de forma errada;
- *True positives (TP): Matches* existentes, tanto no alinhamento *A*, como no Alinhamento *R*, ou seja, *matches* assinalados corretamente; e
- *False negatives: (FN): Matches* presentes no alinhamento *R* e que não estão no alinhamento *A*, ou seja, *matches* esperados que não foram assinalados.

A medida *precision* avalia o rácio entre os *matches* corretos (*TP*) e o total de *matches* assinalados (*TP* + *FP*). Dá a indicação de quantos *matches* assinalados pelas ferramentas são de facto relevantes [105]. O valor da *precision* é dado pela seguinte fórmula (27).

$$Precision = \frac{TP}{TP + FP} \tag{27}$$

Por outro lado, a medida *recall* avalia o rácio entre os matches corretos (TP) e o total de *matches* esperados (TP + FN). Esta medida dá a indicação de quantos *matches* relevantes foram de facto assinalados nos alinhamentos das ferramentas [105]. O cálculo do *recall* é dado pela fórmula (28).

$$Recall = \frac{TP}{TP + FN} \tag{28}$$

A utilização das medidas *precision* e *recall* irão permitir a avaliação dos alinhamentos resultantes da aplicação da abordagem proposta nesta dissertação para integração de modelos conceptuais.

5.2 Cenários

Para validação do processo de integração de modelos conceptuais, proposto neste trabalho, serão aplicados dois diferentes cenários. O objetivo é validar a aplicação do processo e avaliar qualitativamente os resultados obtidos, a partir das medidas *precision* e *recall*. No primeiro cenário, pretende-se efetuar um teste de desempenho, utilizando um *dataset* comum e comparando os resultados com outras ferramentas existentes. No segundo cenário, pretende-se explorar a vertente colaborativa, com diferentes configurações, implementada no processo de similaridade semântica.

5.2.1 Cenário 1: Análise comparativa com *dataset* de referência

Este primeiro cenário tem como objetivo avaliar o desemprenho do processo de similaridade semântica, proposto nesta dissertação, tendo como comparação os resultados disponíveis de outras ferramentas existentes na literatura.

Desde 2004, ocorre anualmente a iniciativa *Ontology Alignment Evaluation Initiative*[32] (OAEI), que tem como objetivo avaliar ferramentas relacionadas com a área de *Ontology Matching*. Esta iniciativa apresenta um conjunto de problemas, de vários domínios, para as ferramentas efetuarem os alinhamentos e submeterem os seus resultados.

Para este cenário, serão utilizadas as ontologias que descrevem o domínio da organização de conferências disponíveis na última edição da campanha, decorrida em 2015, e os respetivos resultados das ferramentas que participaram nesta campanha. As ontologias do domínio organização de conferências são consideradas adequadas para a tarefa de *ontology matching* porque [106]:

- O domínio é geralmente compreensível: Responsáveis pelo desenvolvimento das ontologias estão familiarizados com organização de conferências;
- São independentes: As ontologias foram desenvolvidas de forma independente e com base em diferentes recursos. Assim, as questões relativas à organização de conferências estão representadas com diferentes pontos de vista e com diferentes terminologias (heterogéneas); e
- São ricas em axiomas: Informação semântica importante para aplicação de ferramentas para correspondência de ontologias.

A edição OAEI 2015[33], para o domínio de organização de conferências, disponibiliza os seguintes recursos:

- *Conference*[34]: *Dataset* composto por 16 ontologias que descrevem o domínio da organização e conferências;
- *Reference Alignment*[35]: Alinhamentos de referência entre as ontologias. São a base para a avaliação dos resultados, obtidos pelas ferramentas de correspondência de ontologias; e
- *Conference Results*[36]: Resultados dos alinhamentos obtidos pelas ferramentas participantes.

Na Tabela 24, na Tabela 25 e na Tabela 26 encontram-se, em forma de resumo, os recursos disponibilizados na campanha OAEI 2015. O *dataset*, apresentado na Tabela 24, servirá como input do processo. Será realizada uma análise da similaridade entre cada uma das ontologias, à exceção dos

[32] http://oaei.ontologymatching.org/2015
[33] http://oaei.ontologymatching.org/2015/conference/index.html
[34] http://oaei.ontologymatching.org/2015/conference/data/conference.zip
[35] http://oaei.ontologymatching.org/2015/conference/data/reference-alignment.zip
[36] http://oaei.ontologymatching.org/2015/conference/data/conference2015-results.zip

modelos assinalados com asterisco (*). Estes modelos, apesar de pertencerem ao *dataset*, não contêm um resultado de alinhamento de referência (Tabela 25). Sem esta referência, não é possível aplicar as medidas *precision* e *recall*, e por sua vez, avaliar os resultados obtidos.

Na Tabela 26 estão as ferramentas que submeteram os seus resultados na edição OAEI 2015. Utilizando o resultado de referência, é também possível obter os valores de *precision* e *recall* de cada ferramenta em cada alinhamento. Adicionalmente, fora do contexto da iniciativa OAEI, foram acrescentados os resultados obtidos, utilizando a ferramenta S-Match (seção 3.3.5, página 47) com o mesmo *dataset* disponível. A ferramenta S-Match é direcionada para análises de similaridade entre *lightweight ontologies*, contendo semelhanças com a abordagem proposta neste trabalho e avaliada neste cenário.

No final, depois de se conhecer todos os valores de *precision* e *recall* para todas as ferramentas, é possível efetuar uma análise comparativa em relação à qualidade dos resultados.

Tabela 24: *Dataset* com ontologias do domínio organização de conferências

Ontologia	URI
Cmt	http://oaei.ontologymatching.org/2015/conference/data/cmt.owl
Cocus*	http://oaei.ontologymatching.org/2015/conference/data/cocus.owl
Conference	http://oaei.ontologymatching.org/2015/conference/data/Conference.owl
Confious*	http://oaei.ontologymatching.org/2015/conference/data/confious.owl
ConfOf	http://oaei.ontologymatching.org/2015/conference/data/confOf.owl
Crs_dr*	http://oaei.ontologymatching.org/2015/conference/data/crs_dr.owl
Edas	http://oaei.ontologymatching.org/2015/conference/data/edas.owl
Ekaw	http://oaei.ontologymatching.org/2015/conference/data/ekaw.owl
Iasted	http://oaei.ontologymatching.org/2015/conference/data/iasted.owl
Linklings*	http://oaei.ontologymatching.org/2015/conference/data/linklings.owl
MICRO*	http://oaei.ontologymatching.org/2015/conference/data/MICRO.owl
MyReview*	http://oaei.ontologymatching.org/2015/conference/data/MyReview.owl
OpenConf*	http://oaei.ontologymatching.org/2015/conference/data/OpenConf.owl
Paperdyne*	http://oaei.ontologymatching.org/2015/conference/data/paperdyne.owl
PCS*	http://oaei.ontologymatching.org/2015/conference/data/PCS.owl
Sigkdd	http://oaei.ontologymatching.org/2015/conference/data/sigkdd.owl

*Ontologias não utilizadas neste cenário por não conterem resultado de referência.

Tabela 25: Alinhamentos com resultados de referência

cmt-conference	conference-edas	confOf-sigkdd
cmt-confOf	conference-ekaw	edas-ekaw
cmt-edas	conference-iasted	edas-iasted
cmt-ekaw	conference-sigkdd	edas-sigkdd
cmt-iasted	confOf-edas	ekaw-iasted
cmt-sigkdd	confOf-ekaw	ekaw-sigkdd
conference-confOf	confOf-iasted	iasted-sigkdd

Tabela 26: Ferramentas com resultados de alinhamento

AML	JarvisOM	Mamba
COMMAND*	Lily	RSDLWB
CroMatcher	LogMap	ServOMBI
DKPAOM	LogMapC	XMAP
GMap	LogMapLite	

*Ferramenta com alinhamento inválido que impossibilita o cálculo da *precision* e *recall*.

O *dataset,* disponibilizado pela OAEI, é direcionado ao *matching* ontologias, o que impossibilita a utilização direta, no serviço desenvolvido para a implementação do processo de similaridade semântica, proposto nesta dissertação. Desta forma, foi necessária uma atividade inicial de conversão de ontologias em mapas de conceitos. Na Figura 40, é apresentada a sequência de todo o cenário, desde a conversão de ontologias em mapa de conceitos, até o cálculo de *precision* e *recall* de cada análise de similaridade semântica entre modelos.

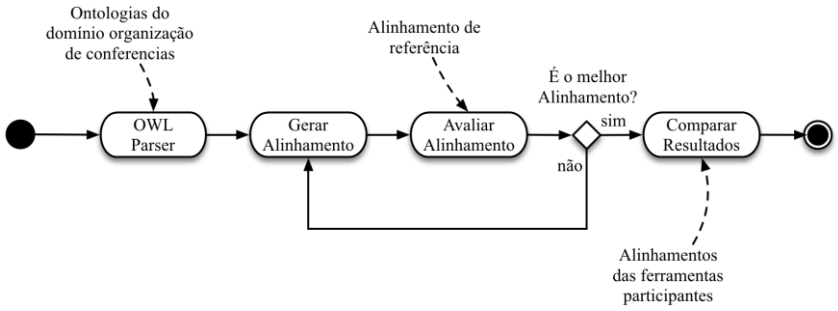


Figura 40: Visão geral da aplicação do processo de similaridade semântica utilizando o *dataset* disponível em OAEI

Na Figura 40 anterior, são descritas as principais atividades existentes neste cenário. Em resumo, as atividades são:

OWL *Parser*: Atividade para conversão de ontologias em modelos conceptuais, de modo a possibilitar a utilização do *dataset*;

Gerar Alinhamento e **Avaliar Alinhamento**: Nestas atividades, diversos alinhamentos serão gerados, com diferentes configurações, e em cada alinhamento é calculado o valor de *precision* e *recall*. O alinhamento, com melhor *precision* e *recall*, é considerado o melhor alinhamento possível de se obter com o serviço de similaridade semântica. O melhor alinhamento é o que será considerado para efeitos comparativos, com os alinhamentos das ferramentas participantes.

Comparar Resultados: Atividade que visa comparar os alinhamentos de cada ferramenta e calcular o valor de *precision* e *recall* para cada uma delas.

Para a aplicação deste cenário, que tem como objetivo avaliar o serviço desenvolvido para integração de modelos conceptuais, definiram-se os seguintes pressupostos:

- Os alinhamentos de referência contêm os *matches* esperados. São vistos como a solução dos especialistas e consideram-se 100% corretos;
- Apenas serão consideradas as *classes* (correspondem a um conceito num modelo conceptual) e as *object properties* (correspondem a relações num modelo conceptual). Todas as restantes entidades existentes nas ontologias serão descartadas, uma vez que este trabalho está focado no nível conceptual, antes da formalização de uma ontologia.
- As avaliações são efetuadas, com base no ponto anterior. Apenas considera *matches* entre *classes* (conceitos), ou seja, com o filtro M1 definido na iniciativa OAEI.

5.2.1.1 Transformação de ontologias em modelos conceptuais

A transformação de uma ontologia em mapa de conceitos é a primeira atividade de preparação para execução do cenário. Para este cenário e, uma vez que os modelos existentes não estão orientados ao *merge* de modelos conceptuais, foi estabelecido um conjunto de informações, que se pode obter de uma ontologia para construir o respetivo modelo conceptual.

Na literatura não foi encontrada qualquer ferramenta que efetuasse conversão direta de ontologias em mapas de conceitos, desta forma, foi criado um módulo para efetuar a conversão (*parse*) de um ficheiro *owl*[37]. Existem na literatura, abordagens de como transformar uma ontologia em mapa de conceitos, e é com base nos princípios propostos pelas abordagens existentes que se desenvolveu o módulo *parser* owl. Os autores em [107], [108], [109] e [110] levantam um conjunto de similaridades entre elementos das ontologias e mapas de conceitos. A Figura 41 apresenta, em termos gerais, os elementos de uma ontologia que podem ser transpostos num mapa de conceitos.

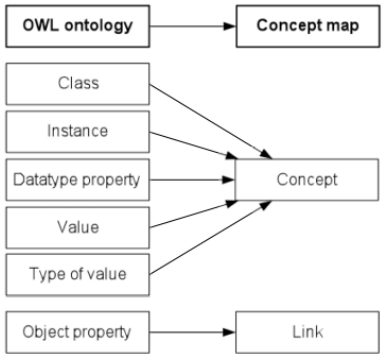


Figura 41: Correspondência de elementos de uma ontologia num mapa de conceitos [108]

A Figura 41, apresentada anteriormente, faz apenas referência aos principais elementos de uma ontologia e suas designações num mapa de conceitos. Existem, no entanto, outros elementos utilizados numa ontologia que devem ser transformados em elementos de um mapa de conceitos. Em [109] é apresentada uma tabela completa, com as correspondências indicadas entre elementos de uma ontologia e elementos de um mapa de conceitos. Contudo, no âmbito do trabalho desenvolvido nesta dissertação, pretende-se apenas trabalhar a nível conceptual, logo, o nível lógico e físico, presentes nas ontologias, não será considerado. A título de exemplo, enumeram-se alguns dos elementos ignorados:

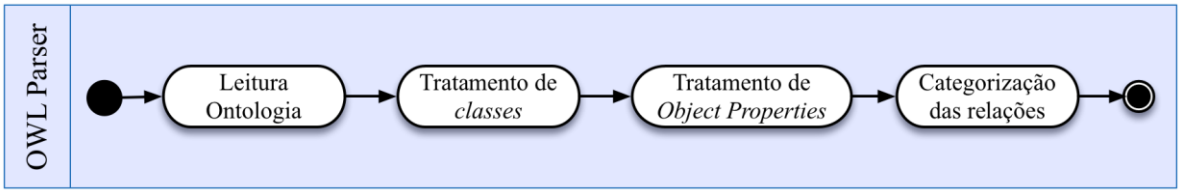
- Propriedades inversas;
- Propriedades transitivas;
- Interseções;
- Uniões;
- Instâncias; e
- Restrições

Em conjunto com as ligações obtidas das *Object properties*, foi introduzida uma nova ligação para representar uma estrutura hierárquica, entre duas classes num modelo conceptual. Deste modo, quando estiver representado numa ontologia algo como: *ClassA subClassOf ClassB*, num mapa de conceitos vai estar representado como: *ClassA is a ClassB*. A adição da relação *is a* é necessária para representar as relações entre conceitos, num modelo conceptual, que são descritos utilizando

[37] *Web Ontology Language (OWL)*: Linguagem de Web Semântica projetada para a representação de conhecimento de determinado domínio (https://www.w3.org/OWL/).

construções especificas na ontologia [108]. A Tabela 27 ilustra o fluxo, desde a entrada de uma ontologia até ao resultado da conversão em mapa de conceitos.

Tabela 27: *Workflow* do módulo *parser* owl

***Parser* OWL**	
Propósito	Converter uma ontologia num mapa de conceitos.
Pressupostos	A ontologia é válida. Considerar apenas *Classes* e *Object Properties* (Filtro M1).
Limitações	Considera os elementos a nível conceptual de uma ontologia.
Resultados	Ficheiro *.txt* com os triplos resultantes da conversão no formato de afirmações para possibilitar a importação no CMapTools. Ficheiro *.json* válido para invocar o serviço com o modelo conceptual gerado.
Recursos utilizados	Ontologias disponibilizadas no *dataset*. Apache Jena: Manipulação dos elementos owl (*queries*). Pellet: *reasoner* para permitir a inferência quando se executam as *queries*.
Fluxo	
	

A leitura da ontologia é feita com recurso à API Apache Jena[38] e com o *reasoner* Pellet[39]. Após a classificação efetuada pelo *reasoner*, no momento de leitura do owl, é possível efetuar *queries* para obter os elementos presentes na ontologia, necessários para o modelo conceptual.

As *queries* são feitas, recorrendo à procura de triplos RDF[40] com determinadas condições. Na Tabela 28 estão representadas as *queries* executadas para obter cada tipo de elemento.

Tabela 28: *Queries* para obter elementos da ontologia

Elemento	***Query***

1) Classes	`ontologyGraphModel.find(Triple.create(`
	`  Node.ANY,`
	`  RDF.Nodes.type,`
	`  OWL.Class.asNode()))`

2) Sub Classes	`ontologyGraphModel.find(Triple.create(`
	`  Node.ANY,`
	`  RDFS.Nodes.subClassOf,`
	`  Node.ANY))`

3) Object Properties	`ontologyGraphModel.find(Triple.create(`
	`  Node.ANY,`
	`  RDF.Nodes.type,`
	`  OWL.ObjectProperty.asNode()))`

[38] Apache Jena: API Java para manipulação de ontologias (https://jena.apache.org/).
[39] Pellet Reasoner: Reasoner OWL baseado em Java, utilizado em conjunto com o Apache Jena, que permite inferir sobre um conjunto afirmações ou axiomas presentes nas ontologias.
[40] https://www.w3.org/TR/2004/REC-rdf-concepts-20040210

Na Tabela 28 anterior, as *queries* realizadas consistem em:

1) Obter todos os triplos em que *Subject* seja do tipo *Class* no vocabulário *owl*;
2) Obter todos os triplos em que o *Predicate* é *subClassOf*. Isto significa que, qualquer classe, que seja subclasse de uma outra classe, será incluída na resposta; e
3) Obter todas as propriedades do tipo *Object Property*.

Um triplo é caraterizado por: *Subject – Predicate – Object* e a sua instanciação, recorrendo à API Jena, é feita utilizando o método *create*, disponível na classe *Triple*, por exemplo: `Triple.create(Subject, Predicate, Object)`.

O relacionamento entre conceitos no modelo conceptual é criado por duas formas:

1ª: Por cada triplo obtido na *query* 2), citada anteriormente, origina na seguinte estrutura conceptual:

*triple.getSubject() – **is a** – triple.getObject()*, em que,

triple.getSubject(): Corresponde ao conceito origem;

triple.getObject(): Corresponde ao conceito destino; e

***is a*:** Relação criada para definir hierárquica entre dois conceitos.

2ª: Por cada triplo obtido na *query* 3), citada anteriormente, é criada uma estrutura conceptual:

property.getDomain() – property.getName() – property.getRange(), em que,

property.getDomain(): Corresponde ao conceito origem;

property.getRange(): Corresponde ao conceito destino; e

property.getName(): Nome da relação que liga os dois conceitos;

Categorização das relações: Atividade que tem como objetivo categorizar as relações, provenientes da conversão da ontologia em modelo conceptual, e evoluir a ontologia de relações CRRM. Seguindo um conjunto de regras preposicionais da língua Inglesa (idioma utilizado no *dataset* disponível), foi levantado um conjunto de padrões, que determina as caraterísticas dos tipos de relações [111]. Considerando que *p* é uma preposição que compõe uma frase de ligação (*l*), denotando uma relação conceptual (*CR*), entre dois conceitos (*c*), então a categoria de relações é dada por um dos padrões apresentados na Tabela 29.

Tabela 29: Padrões para categorização de relações conceptuais

$if\ p = "for"\ AND\ l\ contains\ p = "for"\ then\ l \Rightarrow UsageRelation(l)$
$if\ (l = adverb + p)\ then\ l \Rightarrow UsageRelation(l)$
$if\ (l = adjective + p)\ then\ l \Rightarrow ParticipationRelation(l)$
$if\ p = "of"\ AND\ (l = p\ OR\ l\ contains\ p)\ then\ l \Rightarrow Constitution\_and\_ContainmentRelation(l)$ *Exception:* $if\ p = "of"\ AND\ (l = noun + p)\ then\ l \Rightarrow CausalRelation(l)$
$if\ (p = "at"\ OR\ p = "in"\ OR\ p = "on"\ OR\ p = "to")\ AND\ l = p$ $then\ l \Rightarrow Time\_and\_Space\_DependenceRelation(l)$
$if\ p = "by"\ AND\ (l = verb + p)\ then\ l \Rightarrow CausalRelation(l)$
$if\ (p = "of"\ OR\ p = "in"\ OR\ p = "to")\ AND\ (l = noun + p)\ then\ l \Rightarrow CausalRelation(l)$
$if\ l = "is\ a"\ then\ l \Rightarrow HierarchyRelation(l)$

A evolução da ontologia de relações CRRM é realizada aplicando os padrões da Tabela 29, apresentada anteriormente, na categorização das relações resultantes do *parser*. Terminado o processo de conversão, o resultado é armazenado (Anexo 1) para utilização posterior na invocação do serviço de similaridade semântica.

5.2.1.2 Criar e avaliar o alinhamento

Uma vez criados os modelos conceptuais, resultantes da conversão das ontologias, pode-se invocar o serviço de similaridade semântica desenvolvido. Para cada um dos modelos será feita uma análise de similaridade, tendo como objetivo comparar os resultados obtidos com os resultados de referência.

No alinhamento não existirá qualquer intervenção por parte dos especialistas, apesar do processo de similaridade semântica, proposto nesta dissertação, o permitir. Será um alinhamento automatizado, tal como as ferramentas que servirão como comparação (Tabela 26, página 88), utilizando apenas a informação recolhida do *dataset*.

O serviço de similaridade semântica deve ser invocado, segundo a sua especificação (seção 4.4.1.1, página 71). Assim, para cada par de modelos a analisar deverá ser criado o respetivo objeto que será o input do serviço. No que diz respeito ao *modelo source* e ao *modelo target*, a definição é direta, utilizando o resultado do *parser owl* de cada ontologia. Para definição das opções *sensibility*, *ratio*, *syntacticMeasure* e *semanticMeasure* foi criado um procedimento combinatório, que tem como princípio invocar o serviço, com todos os valores possíveis (valores definidos na especificação do serviço), e mediante os resultados, deverá ser armazenado o resultado que possui melhor valor de *precision* e *recall*. A Tabela 30 é demonstrativa de um objeto que se deve enviar para o serviço com as opções definidas para obter o melhor resultado.

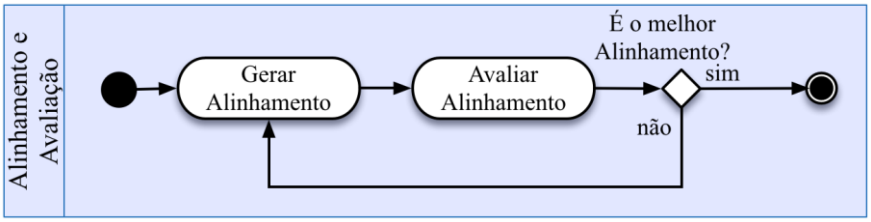
Tabela 30: Objeto de input para o calculo de similaridade semântica

```
{
  "source" : {},
  "target" : {},
  "options" : {
    "semantic" : true,
    "sensibility" : 0.71,
    "ratio" : 1.0,
    "language" : "en",
    "syntacticMeasure" : "Levenshtein",
    "semanticMeasure" : "Corpus"
  }
}
```

Na avaliação dos resultados, os valores de *precision* e *recall* serão calculados, recorrendo à biblioteca AlignAPI[41] e aos resultados de referência, disponíveis na variante *ra1* e modalidade *M1*[42]. Na Tabela 31, são apresentados os pontos principais de todo o processo de alinhamento.

[41] http://alignapi.gforge.inria.fr
[42] http://oaei.ontologymatching.org/2015/conference/eval.html#modalities

Tabela 31: *Workflow* do Alinhamento e Avaliação

Alinhamento e Avaliação	
Propósito	Efetuar alinhamento entre modelos conceptuais para a sua integração. Avaliar os resultados (*matches* de conceitos similares) com as medidas *precision* e *recall,* recorrendo aos resultados de referência.
Pressupostos	Utilizar os resultados de referência *ra1*. Considerar apenas *Classes* e *Object Properties* (Filtro M1). Processo automatizado e com base no melhor resultado obtido do serviço para cada par de modelos em análise.
Limitações	Não tem intervenção do especialista;
Resultados	Ficheiro *.json* com o objeto a utilizar para invocar o serviço de similaridade semântica, contendo: • Modelo origem; • Modelo destino; e • Opções (*sensibility, ratio, semantic, measures*).
Recursos utilizados	Modelos conceptuais convertidos das ontologias disponibilizadas no *dataset*. Serviço de similaridade semântica. *AlignAPI* para cálculo da *precision* e *recall*.
Fluxo	
	

O primeiro passo tem como finalidade obter o valor para cada opção esperada pelo serviço de similaridade semântica. Para cada medida sintática e semântica, é efetuada uma chamada ao serviço com o pior cenário (alinhamento com todos os resultados possíveis) e o valor da *sensibility* e *ratio* serão alterados até que se encontre os valores que permitem obter um alinhamento com o melhor *precision* e *recall*.

O melhor alinhamento é o que será utilizado para comparar com os resultados obtidos pelas ferramentas participantes. Como exemplo, para o alinhamento entre o modelo *cmt* e *sigkdd*, o serviço foi invocado com o objeto definido na Tabela 30, descrita anteriormente, e resultou no alinhamento apresentado na Tabela 32.

Tabela 32: Resultado do alinhamento *cmt-conference*

Alinhamento SimSemantica Filtro M1		**Alinhamento Referência (ra1) Filtro M1**	
Origem (cmt)	**Destino (sigkdd)**	**Origem (cmt)**	**Destino (sigkdd)**
Conference	Conference	ConferenceChair	General_Chair
Paper	Paper	ProgramCommittee	Program_Committee
Person	Person	ProgramCommitteeChair	Program_Chair
Document	Document	Conference	Conference
Author	Author	Paper	Paper
Review	Review	Person	Person
Reviewer	Review	Document	Document
ProgramCommittee	Program_Committee	Author	Author
ProgramCommitteeMember	Program_Committee_member	Review	Review
		ProgramCommitteeMember	Program_Committee_member
Total Correspondências: 9		**Total Correspondências: 10**	

Legenda:

Correspondência correta
Correspondência errada

Olhando para o alinhamento de referência, o número de *matches* corretos esperados é 10. No resultado do alinhamento efetuado pelo serviço de similaridade semântica, obtiveram-se os seguintes valores:

- Matches esperados: 10;
- Matches corretos (TP): 8;
- Matches incorretos (FP): 1;
- Matches em falta (FN): 10 – 8 = 2.

Aplicando as fórmulas de *precision* e *recall* (seção 5.1, página 85), tem-se:

$$precision = \frac{8}{8+1} = 0.89 \qquad recall = \frac{8}{8+2} = 0.8$$

Neste caso, *recall* < *precision* porque a quantidade de matches relevantes devolvidos erradamente (FP) é menor que a quantidade de matches relevantes não devolvidos (FN).

5.2.1.3 Análise dos resultados

Como forma de validar os resultados alcançados, utilizando o processo de similaridade proposto, serão comparados os alinhamentos obtidos pelas diversas ferramentas. Será feita uma análise comparativa entre os alinhamentos, utilizando o mesmo *dataset*, aplicando o filtro *M1* e utilizando os alinhamentos de referência *ra1*, disponíveis em OAEI.

Três casos de teste serão realizados para explorar as diferentes capacidades do processo de similaridade semântica. Os casos de teste são:

- Teste 1: Alinhamento com melhor resultado, utilizando a seleção automática da medida sintática e da medida semântica *Dice - CRRM*;

- Teste 2: Alinhamento com melhor resultado, utilizando as medidas sintáticas e semânticas automaticamente; e
- Teste 3: Alinhamento com melhor resultado, utilizando a seleção automática da medida sintática, e utilizando a medida semântica Corpus, com a introdução de informação semântica. Este teste é efetuado evoluindo o *dataset* com nova informação, pelo que não deve ser alvo de comparação com os resultados das outras ferramentas. É apenas utilizado para validar se o uso do corpus aumenta a qualidade dos resultados.

Com estes diferentes testes, pretende-se perceber a importância da utilização de diferentes parâmetros, na utilização do serviço, e o impacto que as diferentes opções têm no valor de *precision* e *recall*, tendo em conta as especificidades de cada par de modelos conceptuais em análise.

Para cada teste são verificadas quais as melhores opções a serem utilizadas para se obter o melhor alinhamento possível dado pelo serviço (seção 5.2.1.2, página 93).

A Tabela 33 apresenta os valores obtidos para os diferentes parâmetros das opções em cada um dos três testes.

Tabela 33: Cenários de teste

Teste: 1 (syntacticMeasure: Automática; semanticMeasure: Dice-CRRM)											
align	**semantic**	**sensibility**	**ratio**	**syntactic measure**	**semantic measure**	**align**	**semantic**	**sensibility**	**ratio**	**syntactic measure**	**semantic measure**
cmt-conference	sim	0.81	0.5	Levenshtein	Dice-CRRM	confOf-edas	sim	0.76	0.4	Levenshtein	Dice-CRRM
cmt-confOf	sim	0.61	0.2	MongeElkan	Dice-CRRM	confOf-ekaw	sim	0.71	0.4	Levenshtein	Dice-CRRM
cmt-edas	sim	0.86	0.5	Levenshtein	Dice-CRRM	confOf-iasted	sim	0.51	0.0	Levenshtein	Dice-CRRM
cmt-ekaw	sim	0.81	0.5	Levenshtein	Dice-CRRM	confOf-sigkdd	sim	0.76	0.5	Levenshtein	Dice-CRRM
cmt-iasted	sim	0.67	0.4	Levenshtein	Dice-CRRM	edas-ekaw	sim	0.86	0.4	Levenshtein	Dice-CRRM
cmt-sigkdd	sim	0.76	0.6	Levenshtein	Dice-CRRM	edas-iasted	sim	0.76	0.4	Levenshtein	Dice-CRRM
conference-confOf	sim	0.67	0.4	Levenshtein	Dice-CRRM	edas-sigkdd	sim	0.77	0.5	Levenshtein	Dice-CRRM
conference-edas	sim	0.67	0.3	Levenshtein	Dice-CRRM	ekaw-iasted	sim	0.73	0.3	Levenshtein	Dice-CRRM
conference-ekaw	sim	0.67	0.4	Levenshtein	Dice-CRRM	ekaw-sigkdd	sim	0.86	0.4	Levenshtein	Dice-CRRM
conference-iasted	sim	0.7	0.4	Levenshtein	Dice-CRRM	iasted-sigkdd	sim	0.76	0.3	Levenshtein	Dice-CRRM
conference-sigkdd	sim	0.76	0.4	Levenshtein	Dice-CRRM						

Teste: 2 (syntacticMeasure: Automática; semanticMeasure: Automática)											
align	**semantic**	**sensibility**	**ratio**	**syntactic measure**	**semantic measure**	**align**	**semantic**	**sensibility**	**ratio**	**syntactic measure**	**semantic measure**
cmt-conference	sim	0.61	1.0	Levenshtein	Corpus	confOf-edas	sim	0.85	0.9	Jaro	Corpus
cmt-confOf	sim	0.01	0.0	Levenshtein	Corpus	confOf-ekaw	sim	0.89	1.0	Jaro	Corpus
cmt-edas	sim	0.01	0.0	Levenshtein	Corpus	confOf-iasted	sim	0.01	0.0	Levenshtein	Corpus
cmt-ekaw	sim	0.01	0.0	Levenshtein	Corpus	confOf-sigkdd	sim	0.01	0.0	Levenshtein	Corpus
cmt-iasted	sim	0.01	0.0	Levenshtein	Corpus	edas-ekaw	sim	0.65	1.0	Levenshtein	Corpus
cmt-sigkdd	sim	0.71	1.0	Levenshtein	Corpus	edas-iasted	sim	0.86	1.0	Levenshtein	Corpus
conference-confOf	sim	0.01	0.0	Levenshtein	Corpus	edas-sigkdd	sim	0.01	0.0	Levenshtein	Corpus
conference-edas	sim	0.01	0.0	Levenshtein	Corpus	ekaw-iasted	sim	0.76	0.8	Levenshtein	Corpus
conference-ekaw	sim	0.76	1.0	Levenshtein	Corpus	ekaw-sigkdd	sim	0.01	0.0	Levenshtein	Corpus
conference-iasted	sim	0.01	0.0	Levenshtein	Corpus	iasted-sigkdd	sim	0.76	0.9	Levenshtein	Corpus
conference-sigkdd	sim	0.76	1.0	Levenshtein	Corpus						

Teste: 3 (syntacticMeasure: Automática; semanticMeasure: Corpus)											
align	**semantic**	**sensibility**	**ratio**	**syntactic measure**	**semantic measure**	**align**	**semantic**	**sensibility**	**ratio**	**syntactic measure**	**semantic measure**
cmt-conference	sim	0.61	0.7	Levenshtein	Corpus	confOf-edas	sim	0.85	0.9	Jaro	Corpus
cmt-confOf	sim	0.01	0.0	Levenshtein	Corpus	confOf-ekaw	sim	0.89	1.0	Jaro	Corpus
cmt-edas	sim	0.31	0.3	Levenshtein	Corpus	confOf-iasted	sim	0.01	0.0	Levenshtein	Corpus
cmt-ekaw	sim	0.31	0.3	Levenshtein	Corpus	confOf-sigkdd	sim	0.01	0.0	Levenshtein	Corpus
cmt-iasted	sim	0.01	0.0	Levenshtein	Corpus	edas-ekaw	sim	0.65	1.0	Levenshtein	Corpus
cmt-sigkdd	sim	0.71	1.0	Levenshtein	Corpus	edas-iasted	sim	0.86	1.0	Levenshtein	Corpus
conference-confOf	sim	0.01	0.0	Levenshtein	Corpus	edas-sigkdd	sim	0.01	0.0	Levenshtein	Corpus
conference-edas	sim	0.01	0.0	Levenshtein	Corpus	ekaw-iasted	sim	0.76	0.8	Levenshtein	Corpus
conference-ekaw	sim	0.69	0.7	Levenshtein	Corpus	ekaw-sigkdd	sim	0.01	0.0	Levenshtein	Corpus
conference-iasted	sim	0.01	0.0	Levenshtein	Corpus	iasted-sigkdd	sim	0.76	0.9	Levenshtein	Corpus
conference-sigkdd	sim	0.76	1.0	Levenshtein	Corpus						

Utilizando o serviço de similaridade semântica desenvolvido, será gerado um alinhamento por cada par de modelos. Cada alinhamento será construído com base nas opções descritas na Tabela 33, descrita anteriormente, e será calculado o valor de *precision* e *recall* (com recurso à *AlignAPI*) para cada par de modelos conceptuais analisados. Cada alinhamento criado é comparado com o alinhamento de referência e com os alinhamentos das restantes ferramentas.

Na Tabela 34 estão apresentados os resultados médios de *precision* e *recall* obtidos (*H-mean*) para cada teste efetuado, aplicando o filtro M1 e utilizando o alinhamento de referência *ra1* (ver o resultado completo dos valores de *precision* e *recall*, para cada teste, no Anexo 2).

Tabela 34: Resultados globais

Algo	**Teste 1**		**Teste 2**		**Teste 3**	
	Prec.	**Rec.**	**Prec.**	**Rec.**	**Prec.**	**Rec.**
Reference	1,00	1,00	1,00	1,00	1,00	1,00
SimSemantica	0,44	0,32	0,80	0,56	0,83	0,58
AML	0,83	0,70	0,83	0,70	0,83	0,70
CroMatcher	0,72	0,51	0,72	0,51	0,72	0,51
DKPAOM	0,84	0,59	0,84	0,59	0,84	0,59
GMap	0,76	0,71	0,76	0,71	0,76	0,71
JarvisOM	0,88	0,44	0,88	0,44	0,88	0,44
Lily	0,59	0,63	0,59	0,63	0,59	0,63
LogMap	0,83	0,54	0,83	0,54	0,83	0,54
LogMapC	0,84	0,52	0,84	0,52	0,84	0,52
LogMapLite	0,84	0,54	0,84	0,54	0,84	0,54
Mamba	0,84	0,66	0,84	0,66	0,84	0,66
RSDLWB	0,88	0,53	0,88	0,53	0,88	0,53
ServOMBI	0,56	0,44	0,56	0,44	0,56	0,44
XMAP	0,86	0,63	0,86	0,63	0,86	0,63
S-Match	0,39	0,30	0,39	0,30	0,39	0,30

Como é visível no teste 1, o processo proposto apresenta resultados de menor qualidade. Isto deve-se à forma como foram construídos os modelos. Utilizando a medida semântica *Dice – CRRM*, o resultado da similaridade está muito dependente da existência de uma completa categorização das relações. Neste caso, os modelos conceptuais gerados, a partir do *dataset* de ontologias do domínio organização de conferências, possuem uma grande heterogeneidade, resultante da construção de diferentes grupos. Este fator tem impacto na integração de modelos, como é possível verificar a partir do valor de *precision*. Além disso, as ontologias também possuem *object properties* muito especificas, dificultando uma completa categorização das relações. A falta de uma categorização completa de relações influencia o cálculo de similaridade, usando a medida *Dice - CRRM*. Neste teste 1, 44% dos *matches* assinalados pelo serviço estão corretos, cobrindo 32% do total de *matches* esperados

(assinalados no resultado de referência). Contudo, se as construções das ontologias tivessem por base uma ontologia de relações de referência, os resultados teriam sido mais interessantes.

No teste 2, utilizando o melhor alinhamento calculado, com a medida para o cálculo da similaridade sintática e semântica definida automaticamente, os resultados melhoraram consideravelmente. A quantidade de *matches* assinalados como similares, e que realmente o são, subiu para 80%, superando os valores de algumas das ferramentas existentes e apresentando valores próximos das melhores ferramentas. A quantidade de *matches* corretos que o serviço assinalou também subiu para 56% (*recall* = 0,56), o que significa que mais de metade dos matches que deveriam assinalar foram realmente assinalados pelo serviço que implementa a proposta deste trabalho.

O teste 3 foi realizado para determinar até que ponto o serviço devolve melhores resultados com informação adicional, associada aos conceitos. Deve salientar-se que este teste não pode ser comparável com os resultados do alinhamento das restantes ferramentas, porque usará o *dataset* evoluído, alterando a lógia da avaliação padrão OAEI. A evolução do *dataset* consistiu em acrescentar variantes a alguns dos conceitos existentes nos modelos. As variantes adicionadas foram definidas com base no *input* de participantes assíduos em conferências. A Tabela 35 apresenta as variantes de alguns conceitos adicionadas ao *dataset*.

Tabela 35: Variantes associadas a conceitos existentes no *dataset* utilizado

Conceito	Variante
Co-author	Contribution co-author
Document	Conference document
Late paid applicant	Late Registered Participant
Submitted contribution	Submitted Paper
Accepted contribution	Accepted Paper
Contribution	Paper

A evolução do modelo permitiu aumentar a precisão dos matches selecionados em 3% e aumentar também a quantidade de matches corretos esperados em 2%, face ao melhor resultado de alinhamento obtido pelo serviço de similaridade semântica. O valor de *precision* obtido resultou em 0,83 (83% dos matches assinalados estão corretos) e o *recall* 0,58 (58% dos matches esperados foram assinalados). A Figura 42 e a Figura 43 mostram visualmente a evolução dos valores de *precision* e *recall,* obtidos nos três testes efetuados, utilizando o serviço de similaridade semântica.

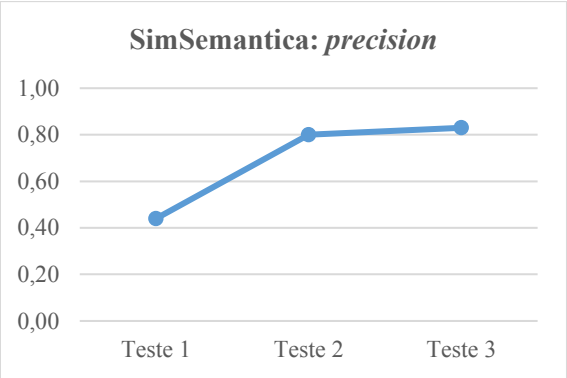


Figura 42: Evolução da precisão dos resultados ao longo dos 3 testes

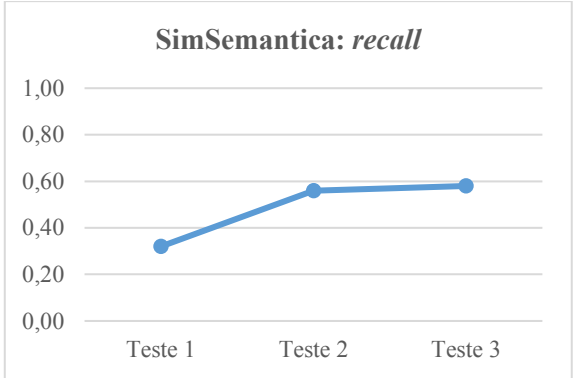


Figura 43: Evolução do *recall* dos resultados ao logo dos 3 testes

Em termos gráficos, também se apresentam na Figura 44 e na Figura 45 os valores de *precision* e *recall*, obtidos por cada ferramenta. O valor de *Reference* corresponde ao valor do alinhamento de referência, e indica o alinhamento 100% correto. A barra legendada com *SimSemantica* corresponde aos valores de *precision* e *recall* obtidos, utilizando o teste 2 (considerado o melhor alinhamento possível, obtido pelo serviço utilizando o *dataset* original).

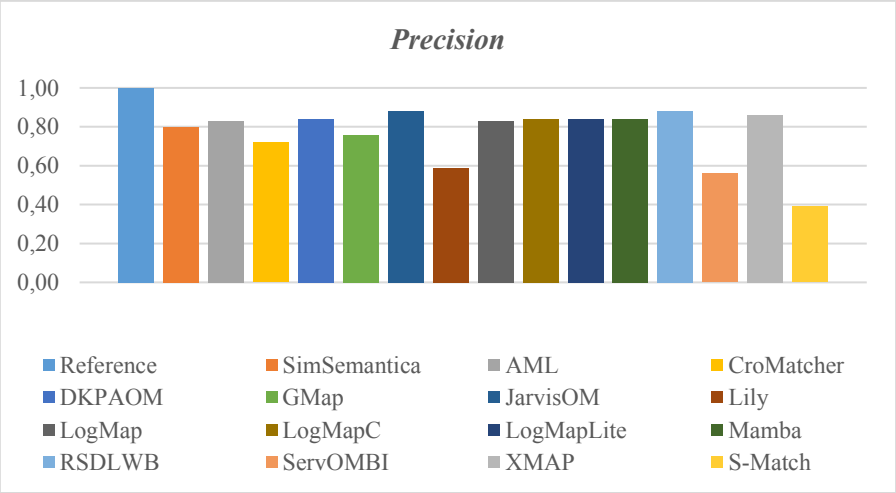


Figura 44: *Precision* obtida em cada ferramenta

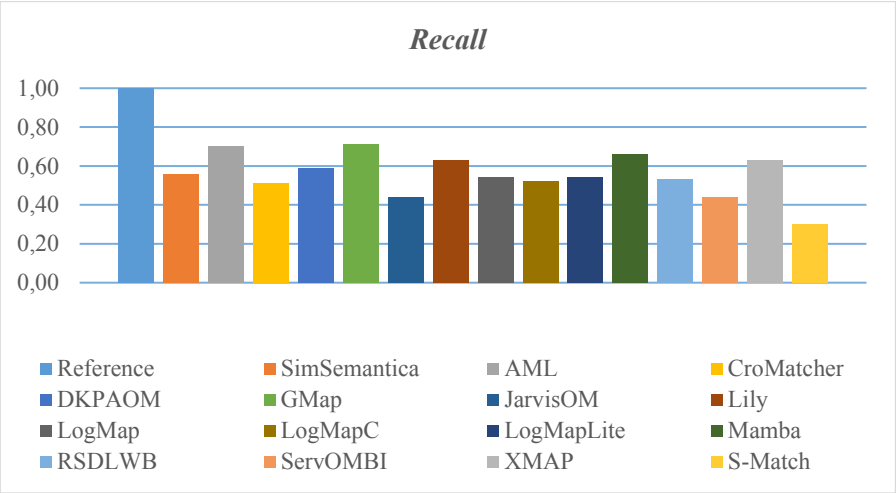


Figura 45: *Recall* obtido em cada ferramenta

Considerando o cenário apresentado e os resultados obtidos, é claramente observável a flexibilidade e utilidade do processo de análise de similaridade de modelos conceptuais, discutido nesta dissertação. Apesar de alguns resultados terem sido menos interessantes em alguns dos testes efetuados, obtiveram-se evidências inequívocas da utilidade da abordagem advogada neste documento, em cenários de colaboração entre especialistas nas atividades de desenvolvimento de conceptualizações semi-formais de domínio.

Esta abordagem também revelou aplicabilidade em cenários de avaliação do grau de similaridade de modelos formais, alcançando resultados promissores. Por outro lado, não será fácil para as ferramentas, testadas neste cenário 1, avaliar o grau de similaridade de modelos conceptuais não formalizados, fato comprovado pelos resultados obtidos com a ferramenta S-Match. Quando a análise está concentrada no nível conceptual (semi-formal), algumas das informações semânticas, presentes nas ontologias, e que ajudam no cálculo similaridade, não são consideradas. As consequências disso são os 39% de *precision* e 30% de *recall* alcançados pela ferramenta S-Match. Face aos valores obtidos pelo S-Match, a abordagem proposta neste trabalho apresentou resultados qualitativamente melhores.

5.2.2 Cenário 2: Processo colaborativo

Neste segundo cenário, para validação da abordagem proposta, serão utilizados modelos conceptuais reais, resultantes de uma conceptualização colaborativa, no projeto FOREST MANAGEMENT DECISION SUPPORT SYSTEMS - FORSYS[43]. Este projeto tem como foco a área da gestão florestal e o seu principal objetivo é fornecer uma visão agregada sobre a experiência europeia, no desenvolvimento e na aplicação de sistemas de apoio à decisão para a gestão florestal.

Uma das atividades realizadas no Forsys, foi a criação de modelos conceptuais que definissem o domínio de sistemas de apoio à decisão para a gestão florestal. Para a construção dos modelos conceptuais foram criados dois grupos privados, G1 e G2, compostos por dois especialistas de domínio.

[43] Informação do projeto FORSYS disponível em http://www.cost.eu/COST_Actions/fps/FP0804.

Numa primeira fase, cada grupo desenvolveu, de forma colaborativa, uma proposta de modelo conceptual que caraterizasse o domínio em questão. Os resultados das conceptualizações estão ilustrados na Figura 46 e na Figura 47, respetivamente, para o grupo G1 e grupo G2.

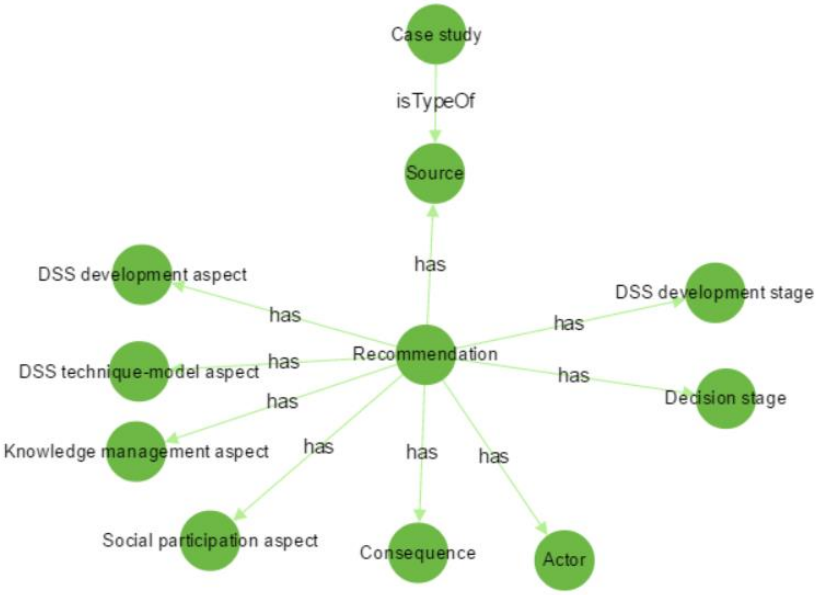


Figura 46: Modelo resultante da conceptualização do grupo G1

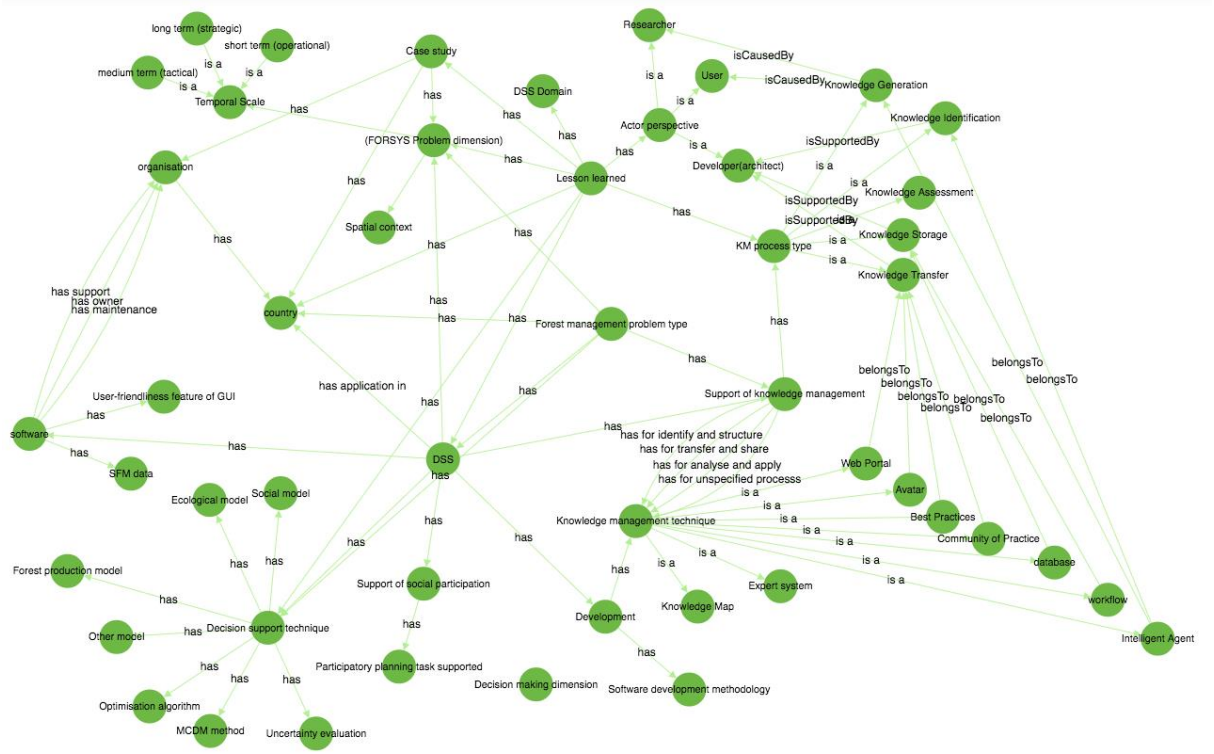


Figura 47: Modelo resultante da conceptualização do grupo G2

Após a construção dos modelos conceptuais, os grupos G1 e G2 disponibilizaram as propostas num espaço colaborativo comum. Neste espaço, existiram atividades de negociação conceptual, com o objetivo de alcançar uma única solução, com o consenso dos participantes. Durante a negociação, os especialistas assinalaram os conceitos repetidos, ou equivalentes, presentes nas duas propostas de modelos conceptuais. Na Tabela 36 é apresentada a listagem de conceitos assinalados pelos especialistas e que foram unidos, dando origem à solução final.

Tabela 36: Conceitos equivalentes assinalados pelos especialistas (resultado de referência)

Conceitos equivalentes	
Modelo do grupo G1	**Modelo do grupo G2**
Actor	Actor perspective
DSS development aspect	Development
Knowledge management aspect	Knowledge management technique
Recommendation	Lesson learned
Social participation aspect	Support of social participation
Case study	Case study
DSS technique-model aspect	Decision support technique
Decision stage	Decision making dimension

Como é possível ver pelos modelos (Figura 46 e Figura 47, descritas anteriormente) a tarefa dos especialistas para encontrar os conceitos equivalentes pode tornar-se muito morosa, especialmente, quando os modelos possuem um elevado número de conceitos (é recomendado uma conceptualização com 15 a 25 conceitos [9]). Neste cenário real, e em particular o modelo do grupo G2, já apresenta uma certa complexidade, fruto dos seus 50 conceitos, que constituem o modelo do grupo.

Assim, com a aplicação deste cenário, pretende-se verificar se, utilizando a abordagem para integração de modelos conceptuais, proposta nesta dissertação, a fase de negociação pode ser facilitada, encontrando automaticamente os conceitos similares entre os modelos em análise. Serão usadas as medidas *precision* e *recall* para uma avaliação quantitativa e qualitativamente dos resultados obtidos, utilizando o serviço de similaridade semântica, desenvolvido neste trabalho.

5.2.2.1 Resultados obtidos

Para este cenário foram criados dois casos de teste, com diferentes configurações. O primeiro caso, tem como objetivo fazer uma análise de similaridade entre modelos, utilizando apenas uma iteração e sem a intervenção do especialista. No segundo caso, pretende-se dar continuidade ao primeiro caso e introduzir o processo iterativo com o envolvimento do especialista. Em resumo, os dois casos de teste consideram:

- Teste 1: Alinhamento com melhor resultado de *precision* e *recall*, tendo em consideração a lista de conceitos equivalentes assinalados pelos especialistas (Tabela 36, citada anteriormente). A lista de equivalentes é considerada como o resultado esperado para qualquer ferramenta de análise de similaridade entre modelos; e
- Teste 2: Utilizar o processo interativo e iterativo disponibilizado pelo serviço desenvolvido como tentativa de melhorar os resultados obtidos no teste 1.

Em ambos os testes, tal como definido no cenário 1, as opções a serem utilizadas, em cada pedido de similaridade semântica, serão definidas automaticamente, seguindo o processo descrito no cenário 1 (seção 5.2.1.2, página 93). A Tabela 37 apresenta as opções usadas em cada teste, nos pedidos efetuados ao serviço de similaridade semântica.

Tabela 37: Testes e respetivas opções usadas no cenário 2

Teste: 1 (syntacticMeasure: Automática; semanticMeasure: Automática)						
align		**semantic**	**sensibility**	**ratio**	**syntactic measure**	**semantic measure**
G1-G2		sim	0.58	0.7	Levenshtein	Corpus
Teste: 2 (syntacticMeasure: Automática; semanticMeasure: Automática)						
align		**semantic**	**sensibility**	**ratio**	**syntactic measure**	**semantic measure**
G1-G2	Iteração 0	sim	0.58	0.7	Levenshtein	Corpus
	Iteração 1	sim	0.53	0.6	MongeElkan	Dice-CRRM

Teste 1

Utilizando as opções apresentadas para o teste 1, na Tabela 37, o serviço criou o alinhamento apresentado na Figura 48.

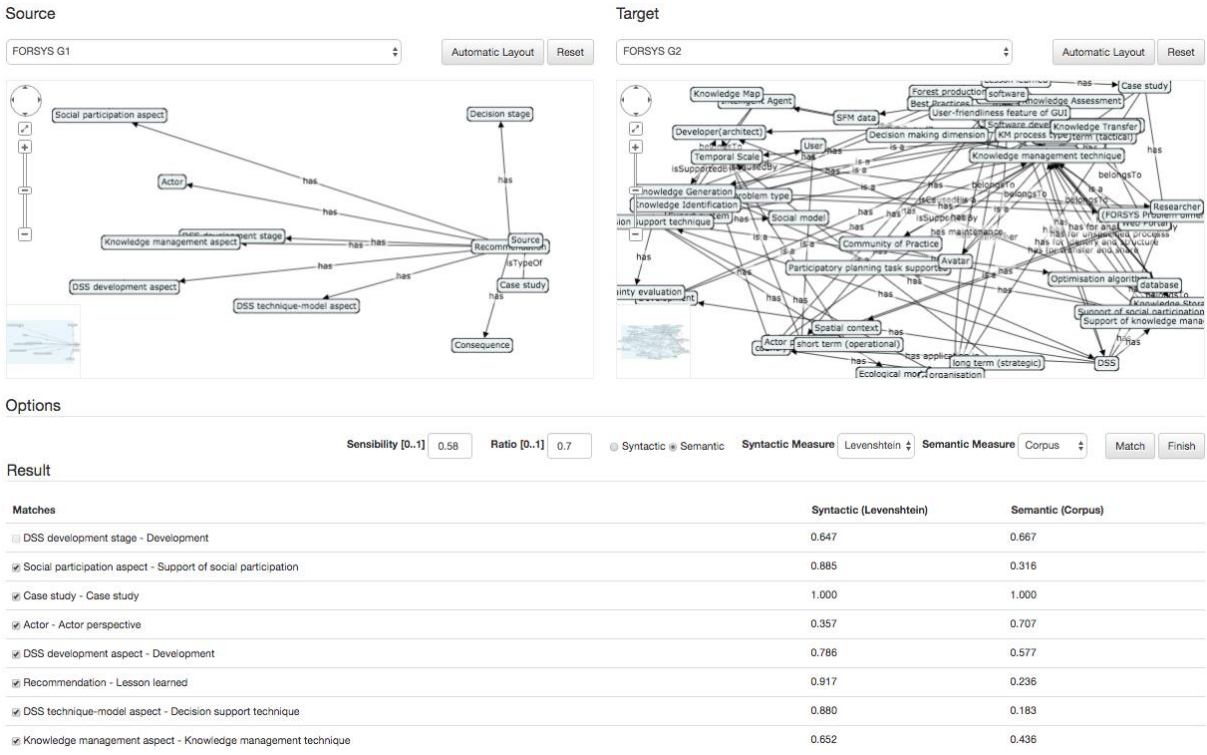


Figura 48: Resultado do alinhamento do teste 1 no cenário 2

Neste teste, os resultados obtidos resumem-se a:

- Total de matches obtidos: 8;
- Total de matches corretos: 7;
- Total de matches errados: 1; e

- Total de matches corretos não obtidos: 1 (tendo em conta os 8 matches esperados, assinalados pelos especialistas na Tabela 36, definida anteriormente).

Teste 2

Neste teste, para avaliar o processo interativo e iterativo desenvolvido para integração de modelos conceptuais, os especialistas deram continuidade ao processo, avançando para a iteração seguinte, após a ação de *merge* dos matches corretos obtidos no teste 1.

Na segunda iteração, com a utilização das novas opções apresentadas na Tabela 37, descrita anteriormente, (teste 2, iteração 1), foram obtidos os resultados apresentados na Figura 49.

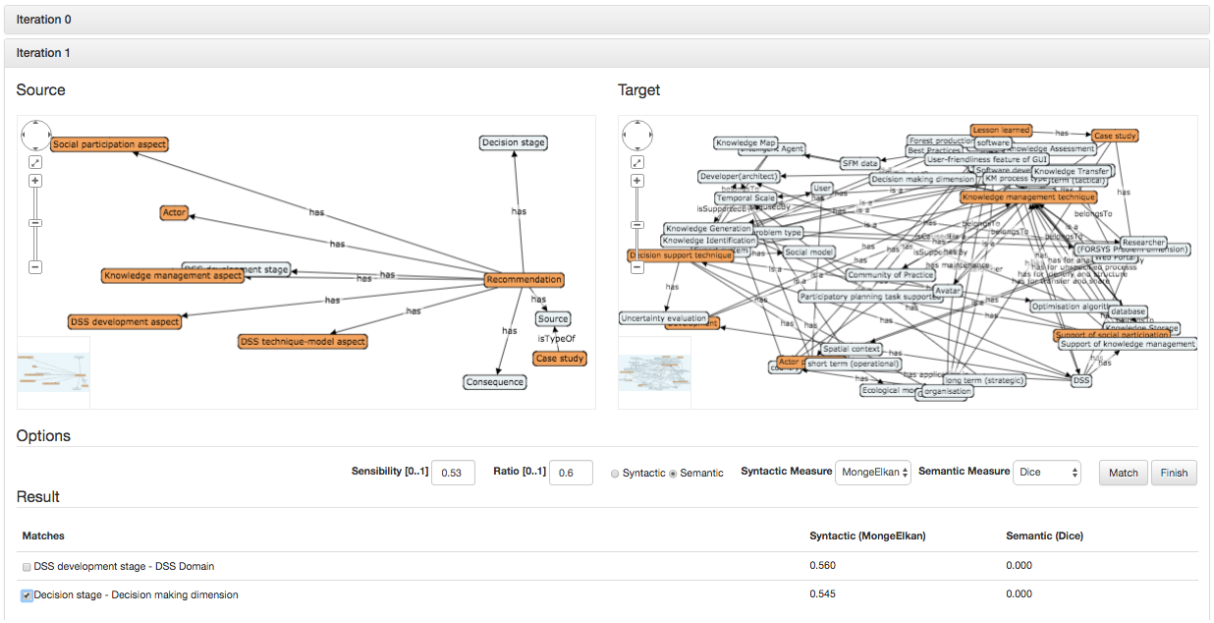


Figura 49: Resultado do alinhamento do teste 2 no cenário 2

Nesta segunda iteração, os resultados obtidos resumem-se a:

- Total de matches obtidos: 2;
- Total de matches corretos: 1;
- Total de matches errados: 1; e
- Total de matches corretos não obtidos: 0 (Dos 8 matches esperados, assinalados pelos especialistas na Tabela 36, definida anteriormente, 7 foram obtidos na primeira iteração e o último na segunda iteração).

Nesta segunda iteração, a medida semântica utilizada (Dice - CRRM) apresenta valor de similaridade igual a zero para os matches resultantes. Este valor é indiciativo de uma não semelhança entre os conceitos, contudo, a conjugação de todas as opções (medida sintática, medida semântica, sensibilidade e ratio) permitiu chegar ao alinhamento apresentado. Em muitas situações, as medidas selecionadas não funcionam só como principal condição para a devolução de *matches* corretos, mas também têm papel importante na não inclusão de *matches* errados nos resultados obtidos. A título de

exemplo, utilizando as mesmas opções desta segunda iteração, e variando apenas a medida semântica, foi obtido um maior número de *matches* errados. A Figura 50 apresenta o resultado com a alteração da medida semântica.

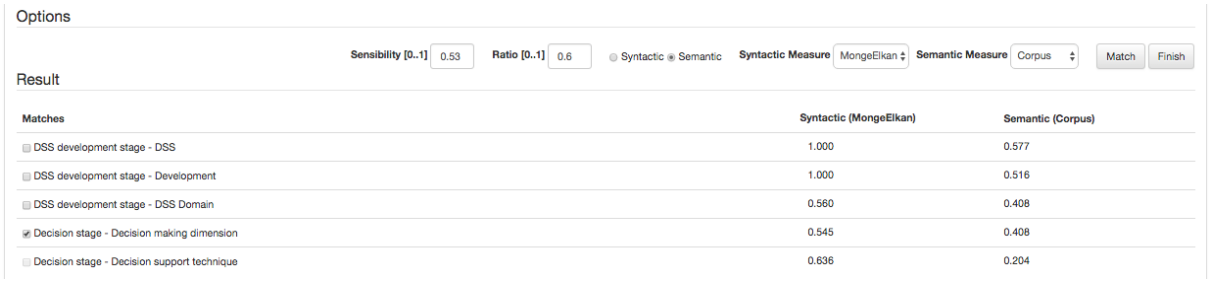


Figura 50: Exemplo demonstrativo de um resultado após alteração da medida utilizada

Tendo em conta que no final desta iteração foram obtidos na totalidade os matches esperados, o processo de integração de modelos conceptuais é dado por concluído.

Para uma análise comparativa dos resultados, os modelos do grupo G1 e G2 foram integrados, recorrendo à ferramenta *CMapTools*. Esta ferramenta (a única encontrada na literatura para comparações entre mapas de conceitos) permite fazer comparações entre mapas de conceitos, assinalando os conceitos similares. A Figura 51 ilustra o resultado da comparação dos modelos do grupo G1 e G2, obtido pelo *CMapTools*.

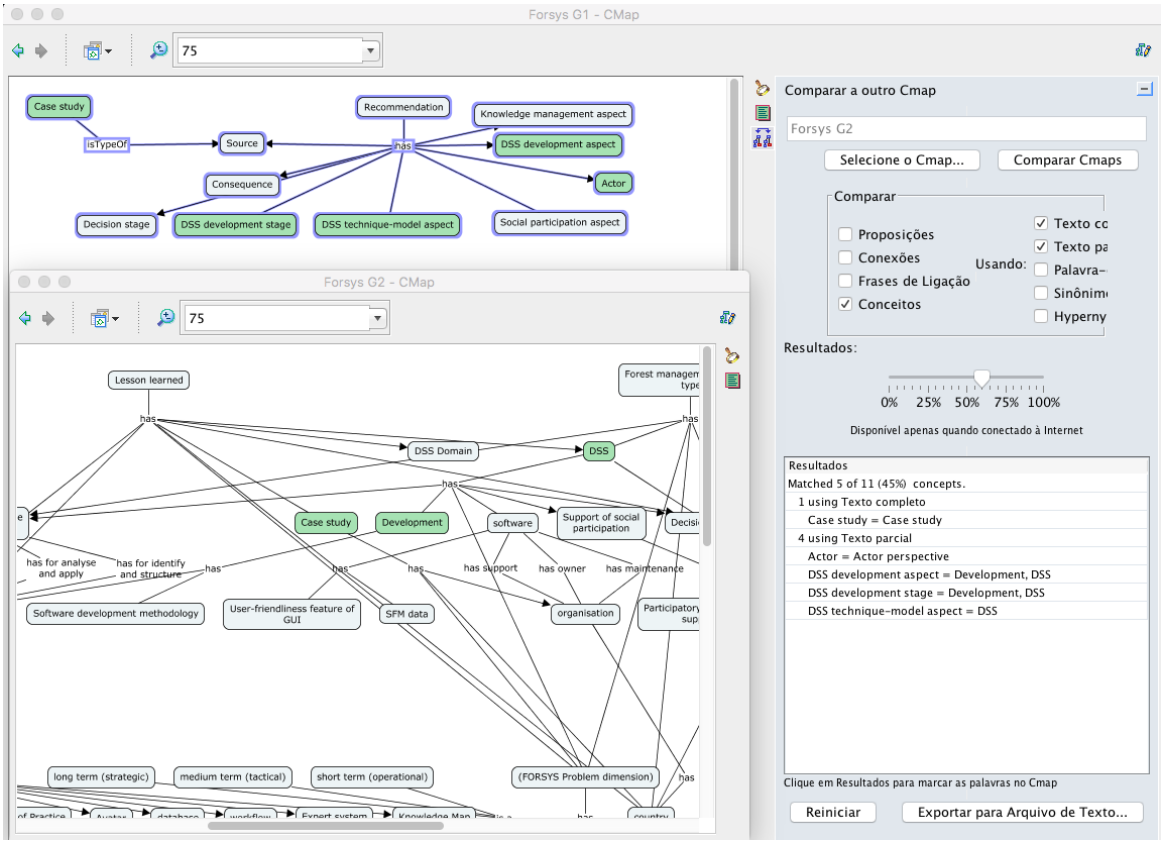


Figura 51: Integração com a ferramenta CMapTools

Resultados obtidos utilizando o CMapTools:

- Total de matches obtidos: 7;
- Total de matches corretos: 3;
- Total de matches errados: 4; e
- Total de matches corretos não obtidos: 5 (Dos 8 *matches* esperados apenas 3 *matches* foram assinalados).

A Tabela 38 seguinte apresenta o resumo dos resultados obtidos dos dois testes e da ferramenta *CMapTools*.

Tabela 38: Resultados obtidos no cenário 2

Resultado esperado	**Teste 1**	**Teste 2**	**CMapTools**
Actor = Actor perspective	✓	✓	✓
DSS development aspect = Development	✓	✓	✓
Knowledge management aspect = Knowledge management technique	✓	✓	×
Recommendation = Lesson learned	✓	✓	×
Social participation aspect = Support of social participation	✓	✓	×
Case study = Case study	✓	✓	✓
DSS technique-model aspect = Decision support technique	✓	✓	×
Decision stage = Decision making dimension	×	✓ (iteração 2)	×
Total *Matches* Corretos	7	8	3
Total *Matches* Errados (não assinalados nos resultados esperados)	1	2	4
Total de *Matches* Obtidos	8	10	7

Aplicando as medidas *precision,* fórmula (27), e *recall,* fórmula (28), citadas anteriormente, a cada um dos resultados, foram obtidos os valores apresentados na Tabela 39 e ilustrados nas Figura 52 e na Figura 53. A linha assinalada com "*Reference*" corresponde ao resultado esperado (Tabela 36, citada anteriormente).

Tabela 39: *Precision* e *recall* obtidos no teste 2

Algo	**Teste 1**		**Teste 2**	
	Prec.	**Rec.**	**Prec.**	**Rec.**
Reference	1,00	1,00	1,00	1,00
SimSemantica	0,88	0,88	0,80	1,00
Cmap	0,43	0,38	0,43	0,38

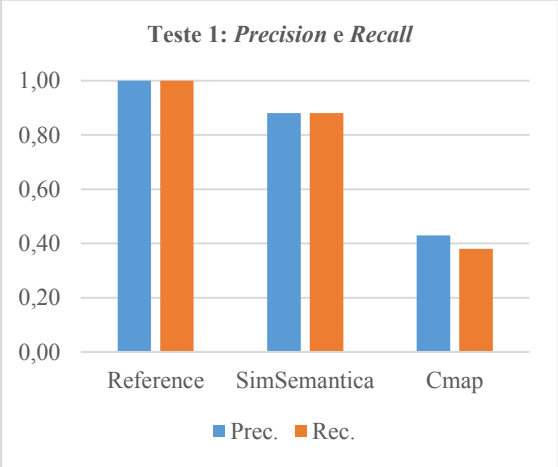


Figura 52: Gráfico comparativo dos resultados do teste 1

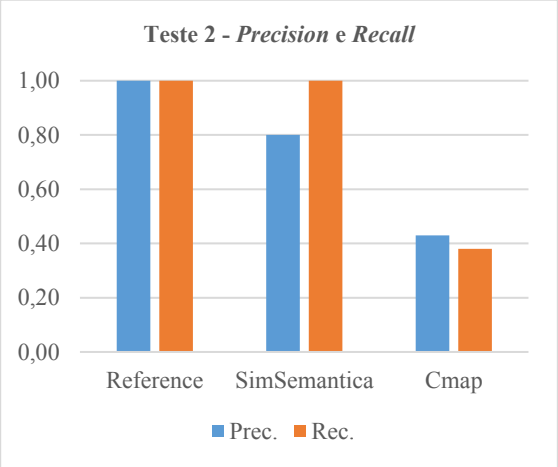


Figura 53: Gráfico comparativo dos resultados do teste 2

5.2.2.2 Análise dos resultados

Dos resultados obtidos no teste 1, foi assinalado apenas um *match* errado e não assinalado um *match* correto, traduzindo-se numa percentagem de 88% tanto para *precision* quanto para *recall*. Comparando com os 43% de *precision* e 38% de *recall*, da ferramenta *CMapTools*, o resultado do teste 1, obtido pela proposta deste trabalho, apresentou melhores valores. Tanto o teste 1 como a ferramenta *CMapTools* calcularam o grau de semelhança entre conceitos, numa única iteração e sem a ação do especialista, contudo, a análise de variantes e a análise semântica, implementadas na abordagem proposta neste trabalho, permitiram que os resultados do teste 1 fossem consideravelmente melhores.

Por sua vez, no teste 2, com a introdução da interação do especialista, foi possível obter o match que faltava, apresentando assim um valor de *recall* de 100%. Face ao teste 1, a precisão do teste 2 desceu para os 80%, após ser assinalado de forma errada um novo *match* na iteração 2. Ainda assim, mantendo um bom valor de *precision*, foi possível cobrir todos os matches esperados, contrastando com os resultados do teste 1 e da ferramenta *CMapTools*.

Mais uma vez, foram notórias as vantagens da flexibilidade existente na utilização do serviço implementado, com a possibilidade de configurar as diferentes opções e a existência da interação do especialista, ao longo das iterações, no processo de integração de modelos conceptuais. O processo iterativo permitiu obter *matches* corretos, que ainda não tinham sido assinalados, mantendo sempre bons valores de *precision* e *recall*.

5.3 Limitações

A aplicação da abordagem desenvolvida nesta dissertação, para o cálculo de similaridade semântica, revelou-se útil no sentido de guiar os especialistas para a união de modelos conceptuais. O serviço desenvolvido apresentou valores de *precision* e *recall* bastante assinaláveis, em comparação com as ferramentas testadas. Contudo, a existência de algumas limitações na abordagem proposta, impediu o alcance de resultados ainda melhores.

As limitações encontradas foram:

- Este tipo de problema, que envolve uma análise de modelos construídos, de forma livre, cria dificuldades acrescidas, pois existem várias situações não controláveis. A construção do modelo, de forma mais completa possível, é preponderante para obter melhores resultados, ou seja, aumentar o número de *matches* corretos e diminuir os falsos *matches*. Considera-se este ponto como uma limitação externa, porque depende de uma correta conceptualização. No entanto, o serviço deve estar preparado para superar a incorreta designação dos nomes que nomeia termos e relações (necessidade da normalização implementada na fase 1 do serviço).
- A envolvência dos especialistas torna-se necessária, uma vez que são eles que irão tomar a decisão final. O processo, apenas dá indicações de possibilidade de matches, com os respetivos graus de similaridade, durante as várias iterações.
- A abordagem proposta neste trabalho considera os tipos de relações na análise de similaridade entre modelos conceptuais (seção 4.2.4). A ontologia de relações CRRM contém a definição dos diferentes tipos de relações disponíveis para categorizar relações existentes nos modelos conceptuais. A dificuldade está na constante evolução dos tipos de relações, e em diferentes línguas. Deste modo, o CRRM precisa ser o mais completo possível para se poder categorizar corretamente o maior número possível de relações nos modelos conceptuais.

No capítulo seguinte serão apresentadas as conclusões e indicados os trabalhos futuros.

6 CONCLUSÕES E TRABALHO FUTURO

6.1 Conclusões

O grande problema que conduziu à realização deste trabalho reside no fato de se observar um elevado grau de esforço na tarefa de integração de modelos conceptuais, desenvolvidos de forma colaborativa. É necessária a obtenção de consenso entre os especialistas para que possam chegarem ao modelo final pretendido.

Assim, este trabalho teve como objetivo principal contribuir na tarefa de integração de modelos conceptuais, procurando agilizar e facilitar o processo de negociação conceptual. Para cumprir este objetivo, foi realizada uma análise de várias medidas de similaridade, sintáticas e semânticas, e verificou-se que a utilização destas medidas, de forma híbrida, poderia ajudar na tarefa de deteção de conceitos similares, em diferentes modelos.

A introdução e o teste de várias medidas permitiram que a abordagem apresentada respondesse a um maior número de particularidades, existentes em modelos conceptuais construídos de forma colaborativa, de múltiplos domínios e em diferentes línguas. O serviço desenvolvido, com implementação de medidas sintáticas e semânticas, apropria-se para as seguintes situações:

- **Modelos com relações hierárquicas**: melhores resultados, utilizando a medida taxonómica, pois consideras as relações hierárquicas (*is-a*)*;*
- **Modelos com informação associada a cada conceito**: a medida *corpus* é a mais indicada porque procura por coocorrências de palavras entre o *corpus* dos conceitos;
- **Modelos com vários tipos de relações**: As medidas *Dice - CRRM* e *Feature* demonstram bons resultados quando os conceitos partilham relações do mesmo tipo; e
- **Modelos simples, com regras de construção e pouca informação semântica**: o uso de medidas sintáticas adequam-se porque a construção de conceitos controlada obriga a que os nomes dos conceitos sejam muito parecidos a nível sintático, fator essencial para obter bons resultado neste tipo de medidas.

Para concluir, este trabalho apresenta as seguintes contribuições:

- Desenho e implementação de uma abordagem de análise semântica, com foco nas relações, *corpus* e taxonomia;
- Serviço *web* (*SimSemanticaService*) flexível e utilizável em vários níveis do processo de criação de modelos conceptuais;
- Abordagem que, ao contrário das existentes, considera o processo de conceptualização colaborativo e os modelos não formalizados;
- Abordagem que tem como pressuposto o apoio à decisão, no processo de integração de modelos, e não na automatização do processo;

- Parte do pressuposto de que o resultado de um alinhamento está fortemente dependente dos tipos de relações usadas na construção dos modelos conceptuais, considerando relações hierárquicas e não-hierárquicas. Este pressuposto confere também, de um outro ponto de vista, uma limitação; e
- Proposta de ferramenta cliente (*SimSemanticaClient*) para auxiliar o processo interativo de integração de modelos conceptuais.

6.2 Possibilidades de trabalho futuro

Após a apresentação das limitações existentes, no desenvolvimento desta investigação, e da análise dos resultados obtidos, sugere-se que trabalhos futuros sobre este tema levem em consideração os seguintes pontos:

- desenvolvimento de mecanismos para detetar qual a melhor medida a ser utilizada em cada iteração (mecanismo parcialmente implementado no capítulo 5.2.1.2, página 93). Envolve análise de relações (recorrendo ao CRRM) e processos de otimização de resultados;
- criação de mecanismos para validar a integridade entre as relações, depois da união de dois conceitos. Muitas vezes, a união faz com que determinados conceitos tenham todo o sentido, mas ao transpor as relações associadas ao conceito do modelo *source* para o modelo *target*, acaba por invalidar o modelo; e
- introdução de medidas baseadas em ontologias externas, em que só devem ser consultadas mediante o domínio em análise. Na abordagem desenvolvida nesta dissertação foi verificado que, na maior parte das vezes, os domínios em análise não tinham recursos externos, sendo inconsequente a aplicação destes tipos de medidas.

REFERÊNCIAS

[1] C. Pereira, "A organização da informação e conhecimento em redes colaborativas como um processo de construção social do significado: uma teoria e um método prático," Universidade do Porto, 2010.

[2] C. Sousa, "Collaborative knowledge representation processes and techniques to support domain experts in conceptual modeling," Universidade do Porto, 2015.

[3] L. Costa, C. Sousa, A. L. Soares, and C. Pereira, "ConceptME: Gestão colaborativa de modelos conceptuais," in *Conferência da Associação Portuguesa de Sistemas de Informação (Capsi)*, 2012.

[4] J.-B. Gao, B.-W. Zhang, and X.-H. Chen, "A WordNet-based semantic similarity measurement combining edge-counting and information content theory," *Eng. Appl. Artif. Intell.*, vol. 39, pp. 80–88, Mar. 2015.

[5] J. Mylopoulos, "Conceptual Modelling and Telos," *Concept. Model. Databases, Case An Integr. view Inf. Syst. Dev.*, pp. 49–68, 1992.

[6] N. Guarino, "Formal Ontology and Information Systems," *Form. Ontol. Inf. Syst.*, pp. 3–15, 1998.

[7] A. J. Cañas, "A Summary of Literature Pertaining to the Use of Concept Mapping Techniques and Technologies for Education and Performance Support," Pensacola, 2003.

[8] J. D. Novak and D. B. Gowin, *Learning How to Learn*. London: Cambrige University Press, 1984.

[9] J. D. Novak and A. J. Cañas, "The Theory Underlying Concept Maps and How to Construct and Use Them," 2008.

[10] H. Yao and L. Etzkorn, "Automated conversion between different knowledge representation formats," *Knowledge-Based Syst.*, vol. 19, no. 6, pp. 404–412, 2006.

[11] A. J. Cañas and J. D. Novak, "Re-examining the foundations for effective use of concept maps," in *Concept maps: theory, methodology, technology. Proceedings of the second international conference on concept mapping*, 2006, vol. 1, pp. 494–502.

[12] M. A. Ruiz-Primo, "Examining Concept Maps as an Assessment Tool," in *Proc. of the First Int. Conference on Concept Mapping.*, 2004.

[13] L. M. Camarinha-Matos and H. Afsarmanesh, "Collaborative networks: value creation in a knowledge society," *Prolamat´06*, no. October 2016, pp. 1–16, 2006.

[14] L. M. Carneiro, A. L. Soares, R. Patrício, A. L. Azevedo, and J. Pinho de Sousa, "Case studies on collaboration, technology and performance factors in business networks," *Int. J. Comput. Integr. Manuf.*, vol. 26, no. 1–2, pp. 101–116, 2013.

[15] T. Maria, P. Dimitris, F. Garifallos, G. Athanasios, and M. Roumeliotis, "Collaboration Learning as a Tool Supporting Value Co-creation. Evaluating Students Learning through Concept Maps," *Procedia - Soc. Behav. Sci.*, vol. 182, pp. 375–380, May 2015.

[16] U. S. Bititci, V. Martinez, P. Albores, and J. Parung, "Creating and Managing Value in Collaborative Networks," *Int. J. Phys. Distrib. Logist. Manag.*, vol. 34, no. 3/4, pp. 251–268, 2004.

[17] D. Romero, N. Galeano, and A. Molina, "Mechanisms for assessing and enhancing organisations' readiness for collaboration in collaborative networks," *Int. J. Prod. Res.*, vol. 47, no. 17, pp. 4691–4710, 2009.

[18] H. Pirkkalainen and J. M. Pawlowski, "Global social knowledge management – Understanding barriers for global workers utilizing social software," *Comput. Human Behav.*, vol. 30, pp. 637–647, Aug. 2014.

[19] Y. Duan, X. Xu, and Z. Fu, "Understanding transnational knowledge transfer," in *Proceedings of the 7th European Conference on Knowledge Management*, 2006, pp. 126–135.

[20] O. Eglene and S. S. Dawes, "Challenges and Strategies for Conducting International Public Management Research," *Adm. Soc.*, vol. 38, no. 5, pp. 596–622, Nov. 2006.

[21] Y. Liu, R. Rousseau, and R. Guns, "A layered framework to study collaboration as a form of knowledge sharing and diffusion," *J. Informetr.*, vol. 7, no. 3, pp. 651–664, Jul. 2013.

[22] C. Sousa, "Discussing and Collaborating Through Concepts : the Conceptme Approach." 2007.

[23] J. Basque and M.-C. Lavoie, “Collaborative concept mapping in education: major research trends,” *Proc. 2nd Int. Conf. Concept Mapp.*, 2006.
[24] H. Gao, M. M. Thomson, and E. Shen, “Knowledge Construction in Collaborative Concept Mapping : A Case Study,” vol. 2, no. March, pp. 1–15, 2013.
[25] A. De Moor, “Ontology-guided meaning negotiation in communities of practice,” in *Proc. of the Workshop on the Design for Large-Scale Digital Communities At the 2Nd International Conference on Communities and Technologies (C&T 2005*, 2005, vol. 2005, pp. 21--28.
[26] L. Costa, C. Pereira, and C. Sousa, “Apoio à negociação conceptual com base em processos híbridos de avaliação de similaridade semântica,” in *Capsi*, 2013.
[27] D. Druckman, “Negotiation models and applications,” in *Diplomacy Games: Formal Models and International Negotiations*, R. Avenhaus and I. W. Zartman, Eds. Springer Berlin Heidelberg, 2007, pp. 83–96.
[28] C. Pereira, C. Sousa, and A. Lucas Soares, “Supporting conceptualisation processes in collaborative networks: a case study on an R&D project,” *Int. J. Comput. Integr. Manuf.*, vol. 26, no. 11, pp. 1066–1086, Jul. 2013.
[29] C. Sousa, C. Pereira, and A. Soares, “Collaborative elicitation of conceptual representations: A corpus-based approach,” in *Advances in Intelligent Systems and Computing*, 2013, vol. 206 AISC, pp. 111–124.
[30] C. Pereira, C. Sousa, and A. L. Soares, “A socio-semantic approach to collaborative domain conceptualization,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5872 LNCS, pp. 524–533, 2009.
[31] S. Cristóvão, A. L. Soares, C. Pereira, and R. Costa, “Supporting the identification of conceptual relations in semi-formal ontology development,” in *Terminology and Knowledge Representation (Terminology and Knowledge Representation)*, 2012, pp. 26–32.
[32] A. De Moor, P. De Leenheer, and R. Meersman, “DOGMA-MESS : A Meaning Evolution Support System for Interorganizational Ontology Engineering,” in *Conceptual Structure: Inspiration and Application*, vol. 4068, 2006, pp. 189–202.
[33] G. Fauconnier and M. Turner, “Conceptual integration Networks,” *Cogn. Sci.*, vol. 22, no. 2, pp. 133–187, 1998.
[34] T. R. Gruber, “A translation approach to portable ontology specifications,” 1993.
[35] N. Noy and D. McGuinness, “Ontology development 101: A guide to creating your first ontology,” 2001.
[36] P. Lambe, *Organising Knowledge: Taxonomies, Knowledge and Organisational Effectiveness*. Chandos Publishing, 2007.
[37] G. P. Malafsky, “Knowledge Taxonomy.” TechI LLC, 2008.
[38] M.-L. FOUCHÉ, “The Role of Taxonomies in Knowledge Management,” University of South Africa, 2006.
[39] A. Pellini and H. Jones, “Knowledge taxonomies - A literature review,” London, United Kingdom, 2011.
[40] O. Serrat, “Taxonomies for Development,” *Knowledge Solutions*. p. 7, 2010.
[41] M. Pincher, “A guide to developing taxonomies for effective data management,” 2010. [Online]. Available: http://www.computerweekly.com/feature/A-guide-to-developing-taxonomies-for-effective-data-management. [Accessed: 27-Nov-2015].
[42] H. Hedden, *The Accidental Taxonomist*, vol. 6, no. 5. Information Today Inc., 2010.
[43] G. H. Alexander Budanitsky, A. Budanitsky, and G. Hirst, “Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures,” in *Workshop on Wordnet and Other Lexical Resources, Second Meeting of The North American Chapter of The Association for Computational Linguistics*, 2001.
[44] D. F. da Silva, “Estudo de Funções de Similaridade Semântica de Termos Aplicadas a um Domínio.” Recife, p. 45, 2007.
[45] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, “Introduction to WordNet: An On-line Lexical Database,” *Int. J. Lexicogr.*, vol. 3, no. 4, pp. 235–244, Dec. 1990.
[46] V. Santos, “Buscas Semânticas na identificação de similaridades entre conceitos para Integração Semântica de Informações.” Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, pp. 1–6, 2010.

[47] C. Mangold, “A survey and classification of semantic search approaches,” *Int. J. Metadata, Semant. Ontol.*, vol. 2, no. 1, p. 23, Jan. 2007.
[48] C. Pesquita, D. Faria, A. O. Falcão, P. Lord, and F. M. Couto, “Semantic Similarity in Biomedical Ontologies,” *PLoS Comput. Biol.*, vol. 5, no. 7, p. e1000443, Jul. 2009.
[49] P. H. Guzzi, M. Mina, C. Guerra, and M. Cannataro, “Semantic similarity analysis of protein data: assessment with biological features and issues,” *Brief. Bioinform.*, vol. 13, no. 5, pp. 569–585, Dec. 2011.
[50] F. M. Couto, M. J. Silva, and P. Coutinho, “Implementation of a Functional Semantic Similarity Measure between Gene-Products,” Lisboa, 2003.
[51] F. M. Couto, M. J. Silva, and P. M. Coutinho, “Semantic similarity over the gene ontology,” in *Proceedings of the 14th ACM international conference on Information and knowledge management - CIKM ’05*, 2005, p. 343.
[52] J. D. Ferreira and F. M. Couto, “Semantic similarity for automatic classification of chemical compounds.,” *PLoS Comput. Biol.*, vol. 6, no. 9, Jan. 2010.
[53] S. Köhler *et al.*, “Clinical diagnostics in human genetics with semantic similarity searches in ontologies.,” *Am. J. Hum. Genet.*, vol. 85, no. 4, pp. 457–64, Oct. 2009.
[54] D. Faria, C. Pesquita, F. M. Couto, and a. Falcão, “Proteinon: A web tool for protein semantic similarity,” *Gene*, 2007.
[55] C. Pesquita, D. Pessoa, D. Faria, and F. M. Couto, “CESSM: Collaborative Evaluation of Semantic Similarity Measures,” *JB2009 Challenges Bioinforma.*, vol. 157, p. 190, 2009.
[56] K. Janowicz, M. Raubal, and W. Kuhn, “The semantics of similarity in geographic information retrieval,” *J. Spat. Inf. Sci.*, vol. 2011, no. 2, pp. 29–57, May 2011.
[57] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web,” *Sci. Am.*, vol. 284, no. 5, pp. 34–43, May 2001.
[58] C. D’Amato, “Similarity-based Learning Methods for the Semantic Web,” Università degli Studi di Bari, 2007.
[59] F. A. Lachtim, H. C. de Souza Jr, A. M. de C. Moura, and M. C. R. Cavalcanti, “Interoperabilidade de Ontologias.” Instituto Militar de Engenharia, Rio de Janeiro, p. 38, 2008.
[60] C. H. Felicíssimo, “Interoperabilidade Semântica na Web: Uma Estratégia para o Alinhamento Taxonômico de Ontologias,” Pontifícia Universidade Católica do Rio de Janeiro, 2004.
[61] P. Raftopoulou and E. Petrakis, “Semantic similarity measures: a comparison study,” Crete, 2005.
[62] Y. Arens, C.-N. Hsu, and C. A. Knoblock, “Query processing in the SIMS information mediator,” in *Advanced Planning Technology: Technological Achievements of The Arpa/Rome Laboratory Planning Initiative*, 1996, pp. 69–69.
[63] E. Mena, V. Kashyap, A. P. Sheth, and A. Illarramendi, “OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperability between Pre-existing Ontologies,” *Conf. Coop. Inf. Syst.*, vol. 271, pp. 14–25, 1996.
[64] S. Castano and V. De Antonellis, “Global viewing of heterogeneous data sources,” *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 2, pp. 277–297, 2001.
[65] P. Resnik, “Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language,” *J. Artif. Intell. Res.*, vol. 11, pp. 95–130, May 1999.
[66] Y. Sun, “A Comparative Evaluation of String Similarity Metrics for Ontology Alignment,” *J. Inf. Comput. Sci.*, vol. 12, no. 3, pp. 957–964, 2015.
[67] A. M. B. Abdelrahman and A. Kayed, “A Survey on Semantic Similarity Measures between Concepts in Health Domain,” *Am. J. Comput. Math.*, vol. 5, no. 2, pp. 204–214, May 2015.
[68] R. Rada, H. Mili, E. Bicknell, and M. Blettner, “Development and application of a metric on semantic nets,” *IEEE Trans. Syst. Man. Cybern.*, vol. 19, no. 1, pp. 17–30, 1989.
[69] R. Richardson, A. F. Smeaton, and J. Murphy, “Using WordNet as a Knowledge Base for Measuring Semantic Similarity Between Words,” in *In Proceedings of AICS Conference*, 1994.
[70] G. Hirst and D. St-Onge, “Lexical chains as representations of context for the detection and correction of malapropisms,” *WordNet - An Electron. Lex. Database*, no. April, pp. 305–332, 1998.
[71] Z. Wu and M. Palmer, “Verbs semantics and lexical selection,” in *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics - ACL ’94*, 1994, pp. 133–138.

[72] Y. Li, Z. A. Bandar, and D. McLean, “An approach for measuring semantic similarity between words using multiple information sources,” *Knowl. Data Eng. IEEE Trans.*, vol. 15, no. 4, pp. 871–882, Jul. 2003.
[73] C. Leacock and M. Chodorow, “Combining Local Context and WordNet Similarity for Word Sense Identification,” *WordNet An Electron. Lex. database.*, vol. 49, no. JANUARY 1998, pp. 265–283, 1998.
[74] M. A. Jaro, “Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida,” *J. Am. Stat. Assoc.*, vol. 84, no. 406, pp. 414–420, 1989.
[75] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, “A Comparison of String Distance Metrics for Name-Matching Tasks,” in *Proceedings of IJCAI-03 Workshop on Information Integration*, 2003, pp. 73–78.
[76] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble, “Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation,” *Bioinformatics*, vol. 19, no. 10, pp. 1275–83, Jul. 2003.
[77] D. Lin, “Principle-based parsing without overgeneration,” in *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics - ACL ’93*, 1993, pp. 112–120.
[78] J. J. Jiang and D. W. Conrath, “Semantic similarity based on corpus statistics and lexical taxonomy,” in *Proc of 10th International Conference on Research in Computational Linguistics, ROCLING’97*, 1997.
[79] A. Tversky, “Features of similarity,” *Psychol. Rev.*, vol. 84, no. 4, pp. 327–352, 1977.
[80] M. A. Rodriguez and M. J. Egenhofer, “Determining semantic similarity among entity classes from different ontologies,” *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 2, pp. 442–456, Mar. 2003.
[81] R. Knappe, H. Bulskov, and T. Andreasen, “On Similarity Measures for Concept-based Querying,” in *Proceedings of the 10th International Fuzzy Systems Association World Congress (IFSA’03)*, 2003, pp. 400–403.
[82] L. Gravano *et al.*, “Using q-grams in a DBMS for Approximate String Processing,” *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 28–34, 2001.
[83] F. M. Gondim, “Algoritmo de Comparação de Strings para Integração de Esquemas de Dados.” Universidade Federal de Pernambuco, Recife, p. 49, 2005.
[84] E. N. Borges and C. A. Cony, “Consultas por similaridade em SGBDS comerciais: estendendo o POSTGRESQL.” Fundação Universidade Federal do Rio Grande, Rio Grande, p. 90, 2005.
[85] W. E. Winkler, “String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage,” in *Proceedings of the Section on Survey Research*, 1990, pp. 354–359.
[86] R. S. Filho, E. P. M. de Souza, A. Traina, and C. T. Jr., “Desmistificando o Conceito de Consultas por Similaridade: A Busca de Novas Aplicações na Medicina,” in *I Workshop de Informática Médica*, 2001.
[87] I. V. Levenshtein, “Binary codes capable of correcting deletions, insertions and reversals,” *Cybern. Control Theory*, vol. 10, no. 8, 1966.
[88] F. Naumann and M. Herschel, “An Introduction to Duplicate Detection,” *Synth. Lect. Data Manag.*, vol. 2, no. 1, pp. 1–87, Jan. 2010.
[89] R. E. Johnson and B. Foote, “Designing Reusable Classes,” *J. Object-Oriented Program.*, vol. 1, no. 2, pp. 22–35, 1988.
[90] A. Bernstein, E. Kaufmann, C. Kiefer, and B. Christoph, “SimPack : A Generic Java Library for Similarity Measures in Ontologies,” 2006.
[91] “ws4j - WordNet Similarity for Java.” [Online]. Available: https://code.google.com/p/ws4j/. [Accessed: 21-Nov-2015].
[92] D. Bär, T. Zesch, and I. Gurevych, “DKPro Similarity: An Open Source Framework for Text Similarity,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2013, pp. 121–126.
[93] K. Kotis, G. A. Vouros, and K. Stergiou, “Towards automatic merging of domain ontologies: The HCONE-merge approach,” *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 4, no. 1, pp. 60–79, Jan. 2006.

[94] G. Stumme and A. Maedche, “FCA-MERGE: bottom-up merging of ontologies,” *IJCAI’01 Proc. 17th Int. Jt. Conf. Artif. Intell. - Vol. 1*, pp. 225–230, Aug. 2001.
[95] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag Berlin Heidelberg, 1999.
[96] Y. Kalfoglou and M. Schorlemmer, “Ontology mapping: the state of the art,” *Knowl. Eng. Rev.*, vol. 18, no. 1, pp. 1–31, Jan. 2003.
[97] N. F. Noy and M. A. Musen, “The PROMPT suite: Interactive tools for ontology merging and mapping,” *Int. J. Hum. Comput. Stud.*, vol. 59, no. 6, pp. 983–1024, Dec. 2003.
[98] A. Maedche, B. Motik, N. Silva, and R. Volz, “MAFRA - A MApping FRAmework for Distributed Ontologies,” in *EKAW ’02 Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management*, 2002, pp. 235–250.
[99] R. Lourdusamy and G. Ganapathy, “Feature Analysis of Ontology Mediation Tools,” *J. Comput. Sci.*, vol. 4, no. 6, pp. 437–446, 2008.
[100] F. Giunchiglia, A. Autayeu, and J. Pane, “S-Match: An open source framework for matching lightweight ontologies,” *Semant. Web*, vol. 3, no. 3, pp. 307–317, 2012.
[101] C. D. Sousa, A. L. Soares, and C. S. Pereira, “Collaborative conceptualisation processes in the development of lightweight ontologies,” *VINE J. Inf. Knowl. Manag. Syst.*, vol. 46, no. 2, pp. 175–193, 2016.
[102] M. M. Hussain, “A Study of Different Ontology Matching System,” vol. 37, no. 12, pp. 10–16, 2012.
[103] P. Shvaiko and J. Euzenat, “Ontology Matching: State of the Art and Future Challenges,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. X, pp. 158–176, 2013.
[104] A. J. Cañas *et al.*, “CmapTools: A Knowledge Modeling and Sharing Environment,” *Concept Maps Theory, Methodol. Technol. Proc. First Int. Conf. Concept Mapp.*, vol. 1, no. 1984, pp. 125–135, 2004.
[105] J. Euzenat and P. Shvaiko, *Ontology matching*. Heidelberg, 2007.
[106] M. Cheatham *et al.*, “Results of the Ontology Alignment Evaluation Initiative 2015,” *Ontol. Alignment Eval. Initiat.*, no. March 2016, pp. 61–100, 2015.
[107] V. Graudina, J. Grundspenkis, and S. Milasevica, “Ontology Merging in the Context of Concept Maps,” *Appl. Comput. Syst.*, vol. 13, no. 1, pp. 29–36, 2012.
[108] V. Graudina and J. Grundspenkis, “Concept map generation from OWL ontologies,” *Proc. third Int. Conf. concept mapping, Tallinn, Est. Helsinki, Finl.*, pp. 263–270, 2008.
[109] V. Graudina, “Owl Ontology Transformation Into Concept Map,” *Comput. Sci.*, vol. 34, pp. 80–92, 2008.
[110] M. Dean, “Annotation classes: A structuring mechanism for owl ontologies,” *CEUR Workshop Proc.*, vol. 496, pp. 1–6, 2009.
[111] N. Schneider, V. Srikumar, J. D. Hwang, and M. Palmer, “A Hierarchy with, of, and for Preposition Supersenses,” in *The 9th Linguistic Annotation Workshop*, 2015, pp. 112–123.
[112] *The American Heritage Dictionary of the English Language*, 4th ed. Boston: Houghton Mifflin Company, 2000.

Anexo 1 RESULTADO DA CONVERSÃO DE ONTOLOGIAS EM MODELOS CONCEPTUAIS

Ontologia	Conceito Origem	Relação	Conceito Destino	Categoria
Cmt	Paper	acceptedBy	Administrator	CAUSAL
Cmt	Administrator	acceptPaper	Paper	
Cmt	ProgramCommitteeMember	addedBy	Administrator	CAUSAL
Cmt	Administrator	addProgramCommitteeMember	ProgramCommitteeMember	
Cmt	Reviewer	adjustBid	Bid	
Cmt	Bid	adjustedBy	Reviewer	CAUSAL
Cmt	Reviewer	assignedByAdministrator	Administrator	CAUSAL
Cmt	ExternalReviewer	assignedByReviewer	Reviewer	CAUSAL
Cmt	Paper	assignedTo	Reviewer	CAUSAL
Cmt	Reviewer	assignExternalReviewer	ExternalReviewer	
Cmt	Administrator	assignReviewer	Reviewer	
Cmt	Co-author	co-writePaper	Paper	
Cmt	Conference	detailsEnteredBy	Administrator	CAUSAL
Cmt	Administrator	enableVirtualMeeting	Conference	
Cmt	ProgramCommitteeChair	endReview	Review	
Cmt	Administrator	enterConferenceDetails	Conference	
Cmt	Administrator	enterReviewCriteria	Conference	
Cmt	Administrator	finalizePaperAssignment	Conference	
Cmt	Conference	hardcopyMailingManifestsPrintedBy	Administrator	CAUSAL
Cmt	Paper	hasAuthor	Author	CONSTITUITION
Cmt	Reviewer	hasBeenAssigned	Paper	CONSTITUITION
Cmt	Paper	hasBid	Bid	CONSTITUITION
Cmt	Paper	hasCo-author	Co-author	CONSTITUITION
Cmt	Conference	hasConferenceMember	ConferenceMember	CONSTITUITION
Cmt	Person	hasConflictOfInterest	Document	CONSTITUITION
Cmt	Paper	hasDecision	Decision	CONSTITUITION
Cmt	ProgramCommittee	hasProgramCommitteeMember	ProgramCommitteeMember	CONSTITUITION
Cmt	Paper	hasSubjectArea	SubjectArea	CONSTITUITION
Cmt	Acceptance	is a	Decision	HIERARCHY
Cmt	Administrator	is a	User	HIERARCHY
Cmt	AssociatedChair	is a	Chairman	HIERARCHY
Cmt	Author	is a	ConferenceMember	HIERARCHY
Cmt	Author	is a	User	HIERARCHY
Cmt	AuthorNotReviewer	is a	Author	HIERARCHY
Cmt	Chairman	is a	ConferenceMember	HIERARCHY
Cmt	Co-author	is a	Author	HIERARCHY
Cmt	ConferenceChair	is a	Chairman	HIERARCHY
Cmt	ConferenceMember	is a	Person	HIERARCHY
Cmt	ExternalReviewer	is a	Person	HIERARCHY
Cmt	Meta-Review	is a	Review	HIERARCHY
Cmt	Meta-Reviewer	is a	Reviewer	HIERARCHY
Cmt	Paper	is a	Document	HIERARCHY
Cmt	PaperAbstract	is a	Paper	HIERARCHY
Cmt	PaperFullVersion	is a	Paper	HIERARCHY
Cmt	ProgramCommitteeChair	is a	Chairman	HIERARCHY
Cmt	ProgramCommitteeChair	is a	ProgramCommitteeMember	HIERARCHY
Cmt	ProgramCommitteeMember	is a	ConferenceMember	HIERARCHY
Cmt	Rejection	is a	Decision	HIERARCHY
Cmt	Review	is a	Document	HIERARCHY
Cmt	Reviewer	is a	ConferenceMember	HIERARCHY
Cmt	Reviewer	is a	User	HIERARCHY
Cmt	User	is a	Person	HIERARCHY
Cmt	ConferenceMember	memberOfConference	Conference	CAUSAL
Cmt	ProgramCommitteeMember	memberOfProgramCommittee	ProgramCommittee	CAUSAL

Ontologia	Conceito Origem	Relação	Conceito Destino	Categoria
Cmt	Conference	paperAssignmentFinalizedBy	Administrator	CAUSAL
Cmt	Conference	paperAssignmentToolsRunBy	Administrator	CAUSAL
Cmt	Administrator	printHardcopyMailingManifests	Conference	
Cmt	Paper	readByMeta-Reviewer	Meta-Reviewer	CAUSAL
Cmt	Paper	readByReviewer	Reviewer	CAUSAL
Cmt	Reviewer	readPaper	Paper	
Cmt	Paper	rejectedBy	Administrator	CAUSAL
Cmt	Administrator	rejectPaper	Paper	
Cmt	Conference	reviewCriteriaEnteredBy	Administrator	CAUSAL
Cmt	Conference	reviewerBiddingStartedBy	Administrator	CAUSAL
Cmt	Administrator	runPaperAssignmentTools	Conference	
Cmt	Administrator	setMaxPapers	ProgramCommitteeMember	
Cmt	Administrator	startReviewerBidding	Conference	
Cmt	Author	submitPaper	Paper	
Cmt	Conference	virtualMeetingEnabledBy	Administrator	CAUSAL
Cmt	Author	writePaper	Paper	
Cmt	Reviewer	writeReview	Review	
Cmt	Review	writtenBy	Reviewer	CAUSAL
Conference	Important_dates	belong_to_a_conference_volume	Conference_volume	CONSTITUITION
Conference	Topic	belongs_to_a_review_reference	Review_preference	CONSTITUITION
Conference	Person	contributes	Conference_document	PARTICIPATION
Conference	Active_conference_participant	gives_presentations	Presentation	PARTICIPATION
Conference	Committee	has_a_committee_chair	Chair	CONSTITUITION
Conference	Committee	has_a_committee_co-chair	Co-chair	CONSTITUITION
Conference	Conference_volume	has_a_commtitee	Committee	CONSTITUITION
Conference	Conference_volume	has_a_program_committee	Committee	CONSTITUITION
Conference	Conference_proceedings	has_a_publisher	Publisher	CONSTITUITION
Conference	Reviewed_contribution	has_a_review	Review	CONSTITUITION
Conference	Submitted_contribution	has_a_review_expertise	Review_expertise	CONSTITUITION
Conference	Conference_volume	has_a_steering_committee	Committee	CONSTITUITION
Conference	Review_expertise	has_a_submitted_contribution	Submitted_contribution	CONSTITUITION
Conference	Conference_part	has_a_track-workshop-tutorial_chair	Track-workshop_chair	CONSTITUITION
Conference	Conference_part	has_a_track-workshop-tutorial_topic	Topic	CONSTITUITION
Conference	Conference_volume	has_an_organizing_committee	Committee	CONSTITUITION
Conference	Conference_document	has_authors	Person	CONSTITUITION
Conference	Conference_volume	has_contributions	Conference_contribution	CONSTITUITION
Conference	Conference_volume	has_important_dates	Important_dates	CONSTITUITION
Conference	Committee	has_members	Committee_member	CONSTITUITION
Conference	Conference_volume	has_parts	Conference_part	CONSTITUITION
Conference	Conference_volume	has_tracks	Conference_part	CONSTITUITION
Conference	Conference_volume	has_tutorials	Tutorial	CONSTITUITION
Conference	Conference_volume	has_workshops	Workshop	CONSTITUITION
Conference	Reviewer	invited_by	Reviewer	CAUSAL
Conference	Reviewer	invites_co-reviewers	Reviewer	USAGE
Conference	Abstract	is a	Extended_abstract	HIERARCHY
Conference	Accepted_contribution	is a	Reviewed_contribution	HIERARCHY
Conference	Active_conference_participant	is a	Conference_contributor	HIERARCHY
Conference	Active_conference_participant	is a	Conference_participant	HIERARCHY
Conference	Call_for_paper	is a	Conference_document	HIERARCHY
Conference	Call_for_participation	is a	Conference_document	HIERARCHY
Conference	Camera_ready_contribution	is a	Accepted_contribution	HIERARCHY
Conference	Chair	is a	Committee_member	HIERARCHY
Conference	Co-chair	is a	Committee_member	HIERARCHY
Conference	Committee_member	is a	Person	HIERARCHY

Ontologia	Conceito Origem	Relação	Conceito Destino	Categoria
Conference	Conference_announcement	is a	Conference_document	HIERARCHY
Conference	Conference_applicant	is a	Person	HIERARCHY
Conference	Conference_contribution	is a	Conference_document	HIERARCHY
Conference	Conference_contributor	is a	Person	HIERARCHY
Conference	Conference_participant	is a	Person	HIERARCHY
Conference	Conference_volume	is a	Conference	HIERARCHY
Conference	Conference_www	is a	Conference_document	HIERARCHY
Conference	Contribution_1th-author	is a	Regular_author	HIERARCHY
Conference	Contribution_co-author	is a	Regular_author	HIERARCHY
Conference	Early_paid_applicant	is a	Paid_applicant	HIERARCHY
Conference	Extended_abstract	is a	Regular_contribution	HIERARCHY
Conference	Information_for_participants	is a	Conference_document	HIERARCHY
Conference	Invited_speaker	is a	Conference_contributor	HIERARCHY
Conference	Invited_talk	is a	Presentation	HIERARCHY
Conference	Late_paid_applicant	is a	Paid_applicant	HIERARCHY
Conference	Organizing_committee	is a	Committee	HIERARCHY
Conference	Paid_applicant	is a	Registeered_applicant	HIERARCHY
Conference	Paper	is a	Regular_contribution	HIERARCHY
Conference	Passive_conference_participant	is a	Conference_participant	HIERARCHY
Conference	Poster	is a	Conference_contribution	HIERARCHY
Conference	Presentation	is a	Conference_contribution	HIERARCHY
Conference	Program_committee	is a	Committee	HIERARCHY
Conference	Registeered_applicant	is a	Conference_applicant	HIERARCHY
Conference	Regular_author	is a	Conference_contributor	HIERARCHY
Conference	Regular_contribution	is a	Written_contribution	HIERARCHY
Conference	Rejected_contribution	is a	Reviewed_contribution	HIERARCHY
Conference	Review	is a	Conference_document	HIERARCHY
Conference	Reviewed_contribution	is a	Submitted_contribution	HIERARCHY
Conference	Reviewer	is a	Person	HIERARCHY
Conference	Steering_committee	is a	Committee	HIERARCHY
Conference	Submitted_contribution	is a	Written_contribution	HIERARCHY
Conference	Track	is a	Conference_part	HIERARCHY
Conference	Track-workshop_chair	is a	Person	HIERARCHY
Conference	Tutorial	is a	Conference_part	HIERARCHY
Conference	Workshop	is a	Conference_part	HIERARCHY
Conference	Written_contribution	is a	Conference_contribution	HIERARCHY
Conference	Topic	is_a_topis_of_conference_parts	Conference_part	CAUSAL
Conference	Presentation	is_given_by	Active_conference_participant	CAUSAL
Conference	Conference_part	is_part_of_conference_volumes	Conference_volume	CONSTITUITION
Conference	Conference_contribution	is_submitted_at	Conference_volume	TIMESPACE
Conference	Publisher	issues	Conference_proceedings	PARTICIPATION
Conference	Review	reviews	Reviewed_contribution	PARTICIPATION
Conference	Co-chair	was_a_committe_co-chair_of	Committee	CONSTITUITION
Conference	Chair	was_a_committee_chair_of	Committee	CONSTITUITION
Conference	Committee	was_a_committee_of	Conference_volume	CONSTITUITION
Conference	Committee_member	was_a_member_of	Committee	CONSTITUITION
Conference	Program_committee	was_a_program_committee_of	Conference_volume	CONSTITUITION
Conference	Steering_committee	was_a_steering_committee_of	Conference_volume	CONSTITUITION
Conference	Track-workshop_chair	was_a_track-workshop_chair_of	Conference_part	CONSTITUITION
Conference	Organizing_committee	was_an_organizing_committee_of	Conference_volume	CONSTITUITION
confOf	Contribution	dealsWith	Topic	
confOf	Person	employedBy	Organization	CAUSAL
confOf	Member_PC	expertOn	Topic	TIMESPACE
confOf	Administrative_event	follows	Administrative_event	
confOf	Working_event	hasAdministrativeEvent	Administrative_event	CONSTITUITION
confOf	Working_event	hasTopic	Topic	CONSTITUITION
confOf	Reviewing_event	is a	Administrative_event	HIERARCHY
confOf	Social_event	is a	Event	HIERARCHY
confOf	Participant	is a	Person	HIERARCHY
confOf	Submission_event	is a	Administrative_event	HIERARCHY
confOf	Trip	is a	Social_event	HIERARCHY

Ontologia	Conceito Origem	Relação	Conceito Destino	Categoria
confOf	Banquet	is a	Social_event	HIERARCHY
confOf	Tutorial	is a	Working_event	HIERARCHY
confOf	Reception	is a	Social_event	HIERARCHY
confOf	Member_PC	is a	Person	HIERARCHY
confOf	Company	is a	Organization	HIERARCHY
confOf	Chair_PC	is a	Person	HIERARCHY
confOf	Volunteer	is a	Person	HIERARCHY
confOf	Science_Worker	is a	Person	HIERARCHY
confOf	Scholar	is a	Person	HIERARCHY
confOf	Student	is a	Participant	HIERARCHY
confOf	Registration_of_participants_event	is a	Administrative_event	HIERARCHY
confOf	Camera_Ready_event	is a	Administrative_event	HIERARCHY
confOf	University	is a	Organization	HIERARCHY
confOf	Working_event	is a	Event	HIERARCHY
confOf	Administrative_event	is a	Event	HIERARCHY
confOf	Author	is a	Person	HIERARCHY
confOf	Assistant	is a	Person	HIERARCHY
confOf	Member	is a	Participant	HIERARCHY
confOf	Conference	is a	Working_event	HIERARCHY
confOf	Reviewing_results_event	is a	Administrative_event	HIERARCHY
confOf	Poster	is a	Contribution	HIERARCHY
confOf	Short_paper	is a	Contribution	HIERARCHY
confOf	Paper	is a	Contribution	HIERARCHY
confOf	Regular	is a	Participant	HIERARCHY
confOf	Workshop	is a	Working_event	HIERARCHY
confOf	Administrator	is a	Person	HIERARCHY
confOf	Administrative_event	parallel_with	Administrative_event	TIMESPACE
confOf	Member_PC	reviewes	Contribution	
confOf	Scholar	studyAt	University	TIMESPACE
confOf	Author	writes	Contribution	
confOf	Contribution	writtenBy	Author	CAUSAL
Edas	Person	attendeeAt	ConferenceEvent	TIMESPACE
Edas	Programme	belongsToEvent	ConferenceEvent	CONSTITUITION
Edas	Call	forEvent	AcademicEvent	TIMESPACE
Edas	ConferenceEvent	hasAttendee	Person	CONSTITUITION
Edas	AcademicEvent	hasCall	Call	CONSTITUITION
Edas	Conference	hasCountry	Country	CONSTITUITION
Edas	ConferenceEvent	hasLocation	Place	CONSTITUITION
Edas	Conference	hasMember	Person	CONSTITUITION
Edas	MealEvent	hasMenu	MealMenu	CONSTITUITION
Edas	ConferenceEvent	hasProgramme	Programme	CONSTITUITION
Edas	ActivePaper	hasRating	ReviewRating	CONSTITUITION
Edas	Author	hasRelatedPaper	Paper	CONSTITUITION
Edas	Reviewer	hasReviewHistory	PersonalReviewHistory	CONSTITUITION
Edas	ComputerNetworksTopic	is a	Topic	HIERARCHY
Edas	PendingPaper	is a	Paper	HIERARCHY
Edas	ComputerNetworksMeasurementsTopic	is a	ComputerNetworksTopic	HIERARCHY
Edas	AcceptRating	is a	ReviewRating	HIERARCHY
Edas	OperatingTopicsystems	is a	Topic	HIERARCHY
Edas	CryptographyTopic	is a	Topic	HIERARCHY
Edas	PerformanceTopic	is a	Topic	HIERARCHY
Edas	WirelessCommunicationsTopic	is a	Topic	HIERARCHY
Edas	SessionChair	is a	Person	HIERARCHY
Edas	MobileComputingTopic	is a	Topic	HIERARCHY
Edas	Reception	is a	SocialEvent	HIERARCHY
Edas	Programme	is a	Document	HIERARCHY
Edas	ConferenceDinner	is a	MealEvent	HIERARCHY
Edas	NGO	is a	Organization	HIERARCHY
Edas	NonAcademicEvent	is a	ConferenceEvent	HIERARCHY
Edas	CallForReviews	is a	Call	HIERARCHY
Edas	ComputerNetworksSwitchingTopic	is a	ComputerNetworksTopic	HIERARCHY

Ontologia	Conceito Origem	Relação	Conceito Destino	Categoria
Edas	NumericalReviewQuestion	is a	ReviewQuestion	HIERARCHY
Edas	ComputerNetworksAapplicationsTopic	is a	ComputerNetworksTopic	HIERARCHY
Edas	TravelGrant	is a	Sponsorship	HIERARCHY
Edas	PowerlineTransmissionTopic	is a	Topic	HIERARCHY
Edas	IndustryOrganization	is a	Organization	HIERARCHY
Edas	AccpetIfRoomRating	is a	ReviewRating	HIERARCHY
Edas	SecurityTopic	is a	Topic	HIERARCHY
Edas	RatedPapers	is a	ActivePaper	HIERARCHY
Edas	Author	is a	Person	HIERARCHY
Edas	AcademiaOrganization	is a	Organization	HIERARCHY
Edas	PersonalReviewHistory	is a	PersonalHistory	HIERARCHY
Edas	WeekRejectRating	is a	ReviewRating	HIERARCHY
Edas	CADTopic	is a	Topic	HIERARCHY
Edas	Reviewer	is a	Person	HIERARCHY
Edas	MeetingRoomPlace	is a	Place	HIERARCHY
Edas	SingleLevelConference	is a	Conference	HIERARCHY
Edas	ComputerNetworksSensorTopic	is a	ComputerNetworksTopic	HIERARCHY
Edas	CallForPapers	is a	Call	HIERARCHY
Edas	OrganizationalMeeting	is a	AcademicEvent	HIERARCHY
Edas	CommunicationsTopic	is a	Topic	HIERARCHY
Edas	SatelliteAndSpaceCommunicationsTopic	is a	Topic	HIERARCHY
Edas	RejectedPaper	is a	Paper	HIERARCHY
Edas	TextualReviewQuestion	is a	ReviewQuestion	HIERARCHY
Edas	ActivePaper	is a	Paper	HIERARCHY
Edas	PublishedPaper	is a	Paper	HIERARCHY
Edas	TPCMember	is a	Person	HIERARCHY
Edas	PaperPresentation	is a	AcademicEvent	HIERARCHY
Edas	MedicineTopic	is a	Topic	HIERARCHY
Edas	Presenter	is a	Author	HIERARCHY
Edas	Workshop	is a	AcademicEvent	HIERARCHY
Edas	ComputerNetworksEnterpriseTopic	is a	ComputerNetworksTopic	HIERARCHY
Edas	ParallelAndDistributedComputingTopic	is a	Topic	HIERARCHY
Edas	GovernmentOrganization	is a	Organization	HIERARCHY
Edas	DiningPlace	is a	Place	HIERARCHY
Edas	CoffeeBreak	is a	BreakEvent	HIERARCHY
Edas	ComputerNetworksSecurityTopic	is a	ComputerNetworksTopic	HIERARCHY
Edas	AccommodationPlace	is a	Place	HIERARCHY
Edas	Attendee	is a	Person	HIERARCHY
Edas	AntennasTopic	is a	Topic	HIERARCHY
Edas	FreeTimeBreak	is a	BreakEvent	HIERARCHY
Edas	SocialEvent	is a	NonAcademicEvent	HIERARCHY
Edas	Paper	is a	Document	HIERARCHY
Edas	TestOnlyTopic	is a	Topic	HIERARCHY
Edas	TalkEvent	is a	AcademicEvent	HIERARCHY
Edas	MultimediaTopic	is a	Topic	HIERARCHY
Edas	ComputerNetworksOpticalTopic	is a	ComputerNetworksTopic	HIERARCHY
Edas	ConferenceVenuePlace	is a	Place	HIERARCHY
Edas	Excursion	is a	SocialEvent	HIERARCHY
Edas	WelcomeTalk	is a	TalkEvent	HIERARCHY
Edas	Review	is a	Document	HIERARCHY
Edas	ComputerArchitectureTopic	is a	Topic	HIERARCHY
Edas	AcademicEvent	is a	ConferenceEvent	HIERARCHY
Edas	BreakEvent	is a	NonAcademicEvent	HIERARCHY
Edas	AcceptedPaper	is a	Paper	HIERARCHY
Edas	TwoLevelConference	is a	Conference	HIERARCHY
Edas	ComputerNetworksManagementTopic	is a	ComputerNetworksTopic	HIERARCHY
Edas	MicroelectronicsTopic	is a	Topic	HIERARCHY
Edas	WithdrawnPaper	is a	Paper	HIERARCHY
Edas	MealEvent	is a	NonAcademicEvent	HIERARCHY
Edas	RadioCommunicationsTopic	is a	Topic	HIERARCHY
Edas	MealBreak	is a	BreakEvent	HIERARCHY

Ontologia	Conceito Origem	Relação	Conceito Destino	Categoria
Edas	SignalProcessingTopic	is a	Topic	HIERARCHY
Edas	CallForManuscripts	is a	Call	HIERARCHY
Edas	RejectRating	is a	ReviewRating	HIERARCHY
Edas	CommunicationTheoryTopic	is a	Topic	HIERARCHY
Edas	MealMenu	is a	Document	HIERARCHY
Edas	ConferenceChair	is a	Person	HIERARCHY
Edas	SlideSet	is a	Document	HIERARCHY
Edas	PersonalPublicationHistory	is a	PersonalHistory	HIERARCHY
Edas	ClosingTalk	is a	TalkEvent	HIERARCHY
Edas	Place	isLocationOf	ConferenceEvent	CONSTITUITION
Edas	Person	isMemberOf	Conference	CONSTITUITION
Edas	MealMenu	isMenuOf	MealEvent	CONSTITUITION
Edas	Organization	isProviderOf	Sponsorship	CONSTITUITION
Edas	PersonalReviewHistory	isReviewHistoryOf	Reviewer	CONSTITUITION
Edas	Paper	isWrittenBy	Author	CAUSAL
Edas	Sponsorship	providedBy	Organization	CAUSAL
Edas	AcceptedPaper	relatedToEvent	PaperPresentation	TIMESPACE
Edas	PaperPresentation	relatedToPaper	AcceptedPaper	TIMESPACE
Ekaw	Person	authorOf	Document	CONSTITUITION
Ekaw	Event	hasEvent	Event	CONSTITUITION
Ekaw	Paper	hasReview	Review	CONSTITUITION
Ekaw	Paper	hasReviewer	Possible_Reviewer	CONSTITUITION
Ekaw	Document	hasUpdatedVersion	Document	CONSTITUITION
Ekaw	Event	heldIn	Location	TIMESPACE
Ekaw	Organising_Agency	is a	Organisation	HIERARCHY
Ekaw	Submitted_Paper	is a	Paper	HIERARCHY
Ekaw	Contributed_Talk	is a	Individual_Presentation	HIERARCHY
Ekaw	Evaluated_Paper	is a	Assigned_Paper	HIERARCHY
Ekaw	Track	is a	Scientific_Event	HIERARCHY
Ekaw	SC_Member	is a	PC_Member	HIERARCHY
Ekaw	Paper	is a	Document	HIERARCHY
Ekaw	Proceedings_Publisher	is a	Organisation	HIERARCHY
Ekaw	University	is a	Academic_Institution	HIERARCHY
Ekaw	Conference_Paper	is a	Paper	HIERARCHY
Ekaw	Conference	is a	Scientific_Event	HIERARCHY
Ekaw	Conference_Proceedings	is a	Proceedings	HIERARCHY
Ekaw	Tutorial	is a	Individual_Presentation	HIERARCHY
Ekaw	Positive_Review	is a	Review	HIERARCHY
Ekaw	Multi-author_Volume	is a	Document	HIERARCHY
Ekaw	Workshop	is a	Scientific_Event	HIERARCHY
Ekaw	Invited_Talk	is a	Individual_Presentation	HIERARCHY
Ekaw	Abstract	is a	Document	HIERARCHY
Ekaw	Neutral_Review	is a	Review	HIERARCHY
Ekaw	Industrial_Paper	is a	Paper	HIERARCHY
Ekaw	Accepted_Paper	is a	Evaluated_Paper	HIERARCHY
Ekaw	Conference_Session	is a	Session	HIERARCHY
Ekaw	Late-Registered_Participant	is a	Conference_Participant	HIERARCHY
Ekaw	Assigned_Paper	is a	Submitted_Paper	HIERARCHY
Ekaw	Rejected_Paper	is a	Evaluated_Paper	HIERARCHY
Ekaw	OC_Member	is a	Conference_Participant	HIERARCHY
Ekaw	Regular_Paper	is a	Paper	HIERARCHY
Ekaw	Negative_Review	is a	Review	HIERARCHY
Ekaw	Agency_Staff_Member	is a	Person	HIERARCHY
Ekaw	Research_Institute	is a	Academic_Institution	HIERARCHY
Ekaw	Regular_Session	is a	Session	HIERARCHY
Ekaw	Poster_Paper	is a	Paper	HIERARCHY
Ekaw	Conference_Participant	is a	Person	HIERARCHY
Ekaw	Paper_Author	is a	Person	HIERARCHY
Ekaw	Academic_Institution	is a	Organisation	HIERARCHY
Ekaw	Session_Chair	is a	Conference_Participant	HIERARCHY
Ekaw	Session_Chair	is a	PC_Member	HIERARCHY

Ontologia	Conceito Origem	Relação	Conceito Destino	Categoria
Ekaw	Early-Registered_Participant	is a	Conference_Participant	HIERARCHY
Ekaw	Workshop_Paper	is a	Paper	HIERARCHY
Ekaw	Presenter	is a	Conference_Participant	HIERARCHY
Ekaw	Workshop_Chair	is a	PC_Member	HIERARCHY
Ekaw	Workshop_Chair	is a	Conference_Participant	HIERARCHY
Ekaw	Invited_Talk_Abstract	is a	Abstract	HIERARCHY
Ekaw	OC_Chair	is a	OC_Member	HIERARCHY
Ekaw	Programme_Brochure	is a	Document	HIERARCHY
Ekaw	PC_Member	is a	Possible_Reviewer	HIERARCHY
Ekaw	Tutorial_Abstract	is a	Abstract	HIERARCHY
Ekaw	Demo_Paper	is a	Paper	HIERARCHY
Ekaw	Tutorial_Chair	is a	PC_Member	HIERARCHY
Ekaw	Tutorial_Chair	is a	Conference_Participant	HIERARCHY
Ekaw	Session	is a	Scientific_Event	HIERARCHY
Ekaw	Possible_Reviewer	is a	Person	HIERARCHY
Ekaw	Demo_Session	is a	Session	HIERARCHY
Ekaw	Review	is a	Document	HIERARCHY
Ekaw	Flyer	is a	Document	HIERARCHY
Ekaw	Demo_Chair	is a	Conference_Participant	HIERARCHY
Ekaw	Camera_Ready_Paper	is a	Paper	HIERARCHY
Ekaw	Conference_Banquet	is a	Social_Event	HIERARCHY
Ekaw	Conference_Trip	is a	Social_Event	HIERARCHY
Ekaw	PC_Chair	is a	Conference_Participant	HIERARCHY
Ekaw	PC_Chair	is a	PC_Member	HIERARCHY
Ekaw	Student	is a	Person	HIERARCHY
Ekaw	Industrial_Session	is a	Conference_Session	HIERARCHY
Ekaw	Proceedings	is a	Multi-author_Volume	HIERARCHY
Ekaw	Social_Event	is a	Event	HIERARCHY
Ekaw	Poster_Session	is a	Session	HIERARCHY
Ekaw	Individual_Presentation	is a	Scientific_Event	HIERARCHY
Ekaw	Web_Site	is a	Document	HIERARCHY
Ekaw	Invited_Speaker	is a	Presenter	HIERARCHY
Ekaw	Workshop_Session	is a	Session	HIERARCHY
Ekaw	Scientific_Event	is a	Event	HIERARCHY
Ekaw	Location	locationOf	Event	CONSTITUITION
Ekaw	Event	partOfEvent	Event	CONSTITUITION
Ekaw	Possible_Reviewer	reviewerOfPaper	Paper	CONSTITUITION
Ekaw	Review	reviewOfPaper	Paper	CONSTITUITION
Ekaw	Review	reviewWrittenBy	Person	CAUSAL
Ekaw	Event	scientificallyOrganisedBy	Academic_Institution	CAUSAL
Ekaw	Academic_Institution	scientificallyOrganises	Event	
Ekaw	Event	technicallyOrganisedBy	Organisation	CAUSAL
Ekaw	Organisation	technicallyOrganises	Event	
Ekaw	Document	updatedVersionOf	Document	CONSTITUITION
Ekaw	Document	writtenBy	Person	CAUSAL
Iasted	Item	go_through	Activity	
Iasted	Sponzorship	is a	Money	HIERARCHY
Iasted	Presentation	is a	Conference_activity	HIERARCHY
Iasted	Technic_activity	is a	Conference_activity	HIERARCHY
Iasted	Author_attendee_book_registration_fee	is a	Registration_fee	HIERARCHY
Iasted	Building	is a	Place	HIERARCHY
Iasted	Conference_airport	is a	Building	HIERARCHY
Iasted	Social_program	is a	Conference_activity	HIERARCHY
Iasted	Brief_introduction_for_Session_chair	is a	Document	HIERARCHY
Iasted	Hotel_fee	is a	Fee	HIERARCHY
Iasted	Value_added_tax	is a	Tax	HIERARCHY
Iasted	Non_speaker	is a	Delegate	HIERARCHY
Iasted	Author_book_proceedings_included	is a	Author	HIERARCHY
Iasted	Double_hotel_room	is a	Hotel_room	HIERARCHY
Iasted	Memeber_registration_fee	is a	Registration_fee	HIERARCHY
Iasted	Renting	is a	Activity_before_conference	HIERARCHY

Ontologia	Conceito Origem	Relação	Conceito Destino	Categoria
Iasted	Sponsor_city	is a	City	HIERARCHY
Iasted	Van	is a	Transport_vehicle	HIERARCHY
Iasted	Modelling	is a	Research	HIERARCHY
Iasted	Video_presentation	is a	Presentation	HIERARCHY
Iasted	One_day_presenter	is a	Delegate	HIERARCHY
Iasted	Author_attendee_cd_registration_fee	is a	Registration_fee	HIERARCHY
Iasted	Review	is a	Document	HIERARCHY
Iasted	Deadline_hotel_reservation	is a	Deadline	HIERARCHY
Iasted	Simulating	is a	Research	HIERARCHY
Iasted	Nonauthor_registration_fee	is a	Registration_fee	HIERARCHY
Iasted	Student_lecturer	is a	Lecturer	HIERARCHY
Iasted	Session_room	is a	Place	HIERARCHY
Iasted	Conference_building	is a	Building	HIERARCHY
Iasted	Session	is a	Lecture	HIERARCHY
Iasted	Final_manuscript	is a	Submission	HIERARCHY
Iasted	Technical_commitee	is a	Delegate	HIERARCHY
Iasted	Publication	is a	Item	HIERARCHY
Iasted	Sponsor	is a	Person	HIERARCHY
Iasted	Student_non_speaker	is a	Non_speaker	HIERARCHY
Iasted	Conference_hall	is a	Place	HIERARCHY
Iasted	Car	is a	Transport_vehicle	HIERARCHY
Iasted	PowerPoint_presentation	is a	Presentation	HIERARCHY
Iasted	Cheque	is a	Payment_document	HIERARCHY
Iasted	Document	is a	Item	HIERARCHY
Iasted	Transport_vehicle	is a	Item	HIERARCHY
Iasted	Presenter_city	is a	City	HIERARCHY
Iasted	Receiving_manuscript	is a	Activity_before_conference	HIERARCHY
Iasted	Introduction	is a	Conference_activity	HIERARCHY
Iasted	Welcome_address	is a	Conference_activity	HIERARCHY
Iasted	Hotel_registration_form	is a	Form	HIERARCHY
Iasted	Computer	is a	Audiovisual_equipment	HIERARCHY
Iasted	Main_office	is a	Place	HIERARCHY
Iasted	Transparency	is a	Document	HIERARCHY
Iasted	Departure	is a	Activity_after_conference	HIERARCHY
Iasted	Single_hotel_room	is a	Hotel_room	HIERARCHY
Iasted	Tutorial_speaker	is a	Author	HIERARCHY
Iasted	Reviewer	is a	Speaker	HIERARCHY
Iasted	Plenary_lecture_speaker	is a	Author	HIERARCHY
Iasted	Payment_document	is a	Document	HIERARCHY
Iasted	Conference_days	is a	Time	HIERARCHY
Iasted	Deadline_for_notification_of_acceptance	is a	Deadline	HIERARCHY
Iasted	Form	is a	Document	HIERARCHY
Iasted	Author_information_form	is a	Form	HIERARCHY
Iasted	Presenter_house	is a	Building	HIERARCHY
Iasted	Conference_Hiker	is a	Delegate	HIERARCHY
Iasted	Conference_city	is a	City	HIERARCHY
Iasted	Delegate	is a	Item	HIERARCHY
Iasted	Delegate	is a	Person	HIERARCHY
Iasted	Deadline	is a	Time	HIERARCHY
Iasted	Cd_proceening	is a	Publication	HIERARCHY
Iasted	IASTED_member	is a	Delegate	HIERARCHY
Iasted	Video_cassette_player	is a	Audiovisual_equipment	HIERARCHY
Iasted	Taxi	is a	Transport_vehicle	HIERARCHY
Iasted	Card	is a	Item	HIERARCHY
Iasted	Presenter_state	is a	State	HIERARCHY
Iasted	Refusing_manuscript	is a	Activity_before_conference	HIERARCHY
Iasted	Viza	is a	Document	HIERARCHY
Iasted	Trip_day	is a	Time	HIERARCHY
Iasted	Fee	is a	Money	HIERARCHY
Iasted	IASTED_non_member	is a	Delegate	HIERARCHY
Iasted	Listener	is a	Delegate	HIERARCHY

Ontologia	Conceito Origem	Relação	Conceito Destino	Categoria
Iasted	Full_day_tour	is a	Activity_after_conference	HIERARCHY
Iasted	Coffee_break	is a	Conference_activity	HIERARCHY
Iasted	Accepting_manuscript	is a	Activity_before_conference	HIERARCHY
Iasted	Research	is a	Activity_before_conference	HIERARCHY
Iasted	Introduction_of_speaker	is a	Introduction	HIERARCHY
Iasted	Conference_hotel	is a	Building	HIERARCHY
Iasted	Conference_activity	is a	Activity	HIERARCHY
Iasted	Initial_manuscipt	is a	Submission	HIERARCHY
Iasted	Nonmember_registration_fee	is a	Registration_fee	HIERARCHY
Iasted	Record_of_attendance	is a	Document	HIERARCHY
Iasted	Author	is a	Speaker	HIERARCHY
Iasted	Submission	is a	Document	HIERARCHY
Iasted	Hotel_presenter	is a	Delegate	HIERARCHY
Iasted	Lecturer	is a	Author	HIERARCHY
Iasted	Tip	is a	Money	HIERARCHY
Iasted	Submissions_deadline	is a	Deadline	HIERARCHY
Iasted	Worker_non_speaker	is a	Non_speaker	HIERARCHY
Iasted	Overhead_projector	is a	Audiovisual_equipment	HIERARCHY
Iasted	Lecture	is a	Conference_activity	HIERARCHY
Iasted	Activity_after_conference	is a	Activity	HIERARCHY
Iasted	Book_proceeding	is a	Publication	HIERARCHY
Iasted	Credit_card	is a	Card	HIERARCHY
Iasted	LCD_projector	is a	Audiovisual_equipment	HIERARCHY
Iasted	Bank_transfer	is a	Payment_document	HIERARCHY
Iasted	Tax	is a	Money	HIERARCHY
Iasted	One_conference_day	is a	Conference_days	HIERARCHY
Iasted	Registration_fee	is a	Fee	HIERARCHY
Iasted	Trip_city	is a	City	HIERARCHY
Iasted	Departure_tax	is a	Tax	HIERARCHY
Iasted	Conference_restaurant	is a	Building	HIERARCHY
Iasted	Dinner_banquet	is a	Social_program	HIERARCHY
Iasted	Presenter_university	is a	Building	HIERARCHY
Iasted	Speaker_lecture	is a	Session	HIERARCHY
Iasted	Camera_ready_manuscript_deadline	is a	Deadline	HIERARCHY
Iasted	Invitation_letter	is a	Document	HIERARCHY
Iasted	Mailing_list	is a	Document	HIERARCHY
Iasted	Speaker	is a	Delegate	HIERARCHY
Iasted	Plenary_lecture	is a	Lecture	HIERARCHY
Iasted	Activity_before_conference	is a	Activity	HIERARCHY
Iasted	Conference_state	is a	State	HIERARCHY
Iasted	Registration_form	is a	Form	HIERARCHY
Iasted	Audiovisual_equipment	is a	Item	HIERARCHY
Iasted	Session_chair	is a	Delegate	HIERARCHY
Iasted	Sponsor_state	is a	State	HIERARCHY
Iasted	Author_cd_proceedings_included	is a	Author	HIERARCHY
Iasted	Sponsor_company_house	is a	Building	HIERARCHY
Iasted	Student_registration_fee	is a	Registration_fee	HIERARCHY
Iasted	Worker_lecturer	is a	Lecturer	HIERARCHY
Iasted	Fee_for_extra_trip	is a	Fee	HIERARCHY
Iasted	Registation_deadline	is a	Deadline	HIERARCHY
Iasted	Tutorial	is a	Lecture	HIERARCHY
Iasted	Shuttle_bus	is a	Transport_vehicle	HIERARCHY
Iasted	Registration	is a	Conference_activity	HIERARCHY
Iasted	Coctail_reception	is a	Conference_activity	HIERARCHY
Iasted	Hotel_room	is a	Place	HIERARCHY
Iasted	Place	is_equipped_by	Item	TIMESPACE
Iasted	Item	is_given_to	Person	TIMESPACE
Iasted	Activity	is_held_after	Time	TIMESPACE
Iasted	Activity	is_held_before	Time	TIMESPACE
Iasted	Item	is_made_from	Item	
Iasted	Item	is_needed_for	Person	CAUSAL
Iasted	Money	is_paid_by	Person	CAUSAL
Iasted	Money	is_paid_in	Building	TIMESPACE
Iasted	Money	is_paid_with	Item	
Iasted	Item	is_prepared_by	Person	CAUSAL
Iasted	Person	is_present	Time	
Iasted	Item	is_sent_after	Time	TIMESPACE
Iasted	Item	is_sent_before	Time	TIMESPACE
Iasted	Item	is_sent_by	Person	CAUSAL
Iasted	Item	is_signed_by	Person	CAUSAL
Iasted	Item	is_used_by	Person	CAUSAL
Iasted	Item	is_writen_by	Person	CAUSAL
Iasted	Person	need	Item	
Iasted	Person	obtain	Item	
Iasted	Person	pay	Money	
Iasted	Person	prepare	Item	
Iasted	Person	send	Item	
Iasted	Person	sign	Item	
Iasted	Person	write	Item	
Sigkdd	Person	can_stay_in	Place	TIMESPACE
Sigkdd	ACM_SIGKDD	design	Deadline	
Sigkdd	Deadline	designed_by	ACM_SIGKDD	CAUSAL
Sigkdd	ACM_SIGKDD	hold	Conference	
Sigkdd	Conference	holded_by	ACM_SIGKDD	CAUSAL
Sigkdd	Invited_Speaker	is a	Speaker	HIERARCHY
Sigkdd	Registration_fee	is a	Fee	HIERARCHY
Sigkdd	Program_Committee_member	is a	Organizator	HIERARCHY
Sigkdd	Registration_SIGMOD_Member	is a	Registration_fee	HIERARCHY
Sigkdd	Program_Committee	is a	Committee	HIERARCHY
Sigkdd	Abstract	is a	Document	HIERARCHY
Sigkdd	Speaker	is a	Person	HIERARCHY
Sigkdd	Best_Applications_Paper_Award	is a	Award	HIERARCHY
Sigkdd	Bronze_Supporter	is a	Sponzor	HIERARCHY
Sigkdd	Webmaster	is a	Organizator	HIERARCHY
Sigkdd	Best_Student_Paper_Supporter	is a	Sponzor	HIERARCHY
Sigkdd	Organizing_Committee	is a	Committee	HIERARCHY
Sigkdd	Sponzor_fee	is a	Fee	HIERARCHY
Sigkdd	Program_Chair	is a	Organizator	HIERARCHY
Sigkdd	Exhibitor	is a	Sponzor	HIERARCHY
Sigkdd	Platinum_Supporter	is a	Sponzor	HIERARCHY
Sigkdd	Listener	is a	Person	HIERARCHY
Sigkdd	Deadline_Author_notification	is a	Deadline	HIERARCHY
Sigkdd	Review	is a	Document	HIERARCHY
Sigkdd	Hotel	is a	Place	HIERARCHY
Sigkdd	Registration_Student	is a	Registration_fee	HIERARCHY
Sigkdd	Conference_hall	is a	Place	HIERARCHY
Sigkdd	Organizator	is a	Person	HIERARCHY
Sigkdd	Deadline_Paper_Submission	is a	Deadline	HIERARCHY
Sigkdd	Sponzor	is a	Person	HIERARCHY
Sigkdd	General_Chair	is a	Organizator	HIERARCHY
Sigkdd	Silver_Supporter	is a	Sponzor	HIERARCHY
Sigkdd	Best_Research_Paper_Award	is a	Award	HIERARCHY
Sigkdd	Author	is a	Speaker	HIERARCHY
Sigkdd	Registration_SIGKDD_Member	is a	Registration_fee	HIERARCHY
Sigkdd	Registration_Non-Member	is a	Registration_fee	HIERARCHY
Sigkdd	Gold_Supporter	is a	Sponzor	HIERARCHY
Sigkdd	Best_Paper_Awards_Committee	is a	Committee	HIERARCHY
Sigkdd	Main_office	is a	Place	HIERARCHY
Sigkdd	Author_of_paper	is a	Author	HIERARCHY
Sigkdd	Best_Student_Paper_Award	is a	Award	HIERARCHY
Sigkdd	Paper	is a	Document	HIERARCHY
Sigkdd	Author_of_paper_student	is a	Author	HIERARCHY

Ontologia	Conceito Origem	Relação	Conceito Destino	Categoria
Sigkdd	Deadline_Abstract_Submission	is a	Deadline	HIERARCHY
Sigkdd	Organizing_Committee_member	is a	Organizator	HIERARCHY
Sigkdd	Author	notification_until	Deadline_Author_notification	
Sigkdd	Author	obtain	Award	
Sigkdd	Person	pay	Registration_fee	
Sigkdd	Registration_fee	payed_by	Person	CAUSAL
Sigkdd	Speaker	presentation	Document	
Sigkdd	Document	presentationed_by	Speaker	CAUSAL

Ontologia	Conceito Origem	Relação	Conceito Destino	Categoria
Sigkdd	ACM_SIGKDD	search	Sponzor	
Sigkdd	Sponzor	searched_by	ACM_SIGKDD	CAUSAL
Sigkdd	Author	submit	Paper	
Sigkdd	Document	submit_until	Deadline	

Anexo 2 RESULTADOS OBTIDOS NO CENÁRIO 1

Teste 1

algo	Reference		SimSemantica		AML		CroMatcher		DKPAOM		GMap		JarvisOM		Lily		LogMap		LogMapC		LogMapLite		Mamba		RSDLWB		ServOMBI		XMAP		S-Match	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
cmt-conference	1,00	1,00	1,00	0,25	0,64	0,58	0,43	0,25	0,63	0,42	0,77	0,83	0,60	0,25	0,53	0,75	0,64	0,58	0,70	0,58	0,57	0,33	0,71	0,42	0,67	0,33	0,58	0,58	0,73	0,67	0,60	0,25
cmt-confOf	1,00	1,00	0,14	0,20	0,89	0,80	0,67	0,20	0,80	0,40	0,83	0,50	0,80	0,40	0,80	0,40	0,80	0,40	0,80	0,40	0,80	0,40	1,00	0,70	0,80	0,40	n/a	n/a	0,83	0,50	0,67	0,20
cmt-edas	1,00	1,00	0,25	0,50	0,89	1,00	0,67	0,75	1,00	1,00	1,00	1,00	1,00	0,88	0,64	0,88	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,80	1,00	1,00	1,00	0,22	0,63
cmt-ekaw	1,00	1,00	0,22	0,25	0,75	0,75	0,50	0,50	1,00	0,63	0,67	0,75	1,00	0,63	0,55	0,75	0,86	0,75	0,86	0,75	0,83	0,63	0,63	0,63	1,00	0,63	1,00	0,63	1,00	0,75	0,67	0,50
cmt-iasted	1,00	1,00	0,50	0,25	0,80	1,00	0,67	1,00	0,67	1,00	0,67	1,00	0,80	1,00	0,57	1,00	0,80	1,00	0,80	1,00	0,80	1,00	0,80	1,00	0,80	1,00	0,33	1,00	0,80	1,00	1,00	0,25
cmt-sigkdd	1,00	1,00	0,38	0,50	0,90	0,90	1,00	0,70	1,00	0,80	0,90	0,90	1,00	0,60	0,70	0,70	1,00	0,90	1,00	0,80	1,00	0,80	1,00	0,90	1,00	0,80	n/a	n/a	1,00	0,90	0,67	0,60
conference-confOf	1,00	1,00	0,67	0,55	0,82	0,82	0,86	0,55	0,82	0,82	0,77	0,91	0,88	0,64	0,67	0,73	0,75	0,82	0,82	0,82	0,88	0,64	0,90	0,82	0,88	0,64	n/a	n/a	0,80	0,73	0,75	0,27
conference-edas	1,00	1,00	1,00	0,21	0,69	0,64	0,70	0,50	0,89	0,57	0,64	0,64	0,88	0,50	0,41	0,50	0,90	0,64	0,89	0,57	0,88	0,50	0,82	0,64	0,88	0,50	n/a	n/a	0,89	0,57	0,11	0,29
conference-ekaw	1,00	1,00	0,28	0,30	0,78	0,78	0,53	0,39	0,53	0,35	0,62	0,57	0,71	0,22	0,62	0,70	0,65	0,48	0,67	0,43	0,62	0,35	0,68	0,57	0,67	0,35	0,44	0,48	0,65	0,48	0,67	0,09
conference-iasted	1,00	1,00	0,15	0,15	0,83	0,38	0,50	0,31	0,83	0,38	0,78	0,54	0,80	0,31	0,67	0,46	0,78	0,54	0,78	0,54	0,80	0,31	0,60	0,46	0,80	0,31	0,22	0,31	0,83	0,38	1,00	0,08
conference-sigkdd	1,00	1,00	0,83	0,42	0,83	0,83	0,80	0,67	0,90	0,75	0,85	0,92	0,80	0,33	0,71	0,83	0,77	0,83	0,83	0,83	0,89	0,67	0,91	0,83	0,89	0,67	n/a	n/a	0,82	0,75	0,80	0,33
confOf-edas	1,00	1,00	0,83	0,36	0,90	0,64	0,86	0,43	0,90	0,64	0,83	0,71	0,89	0,57	0,69	0,64	0,90	0,64	0,90	0,64	0,90	0,64	0,90	0,64	0,89	0,57	0,86	0,86	0,90	0,64	0,21	0,64
confOf-ekaw	1,00	1,00	0,62	0,40	0,94	0,80	0,77	0,50	0,93	0,65	0,85	0,85	0,88	0,35	0,79	0,75	0,93	0,70	0,93	0,65	0,83	0,50	0,88	0,70	0,89	0,40	0,73	0,80	0,93	0,70	1,00	0,25
confOf-iasted	1,00	1,00	1,00	0,11	0,80	0,44	0,67	0,22	0,80	0,44	0,60	0,67	1,00	0,44	0,46	0,67	1,00	0,44	1,00	0,44	1,00	0,44	0,83	0,56	1,00	0,44	0,40	0,67	0,80	0,44	1,00	0,22
confOf-sigkdd	1,00	1,00	0,25	0,33	1,00	0,83	1,00	0,67	1,00	0,67	0,80	0,67	1,00	0,67	0,17	0,17	1,00	0,67	1,00	0,67	1,00	0,67	1,00	0,67	1,00	0,67	0,83	0,83	1,00	0,67	1,00	0,17
edas-ekaw	1,00	1,00	0,60	0,32	0,79	0,58	0,60	0,47	0,73	0,58	0,65	0,68	0,71	0,26	0,67	0,63	0,73	0,58	0,71	0,53	0,69	0,47	0,77	0,53	0,75	0,47	0,74	0,74	0,79	0,58	0,44	0,37
edas-iasted	1,00	1,00	0,60	0,32	0,82	0,47	0,57	0,42	0,88	0,37	0,82	0,47	0,83	0,26	0,50	0,37	0,88	0,37	0,88	0,37	0,88	0,37	0,82	0,47	0,88	0,37	0,33	0,37	0,88	0,37	0,17	0,05
edas-sigkdd	1,00	1,00	0,63	0,45	1,00	0,64	1,00	0,73	1,00	0,64	1,00	0,64	1,00	0,64	0,63	0,45	1,00	0,64	1,00	0,64	1,00	0,64	1,00	0,64	1,00	0,64	n/a	n/a	1,00	0,64	0,33	0,55
ekaw-iasted	1,00	1,00	0,33	0,20	0,88	0,70	1,00	0,60	1,00	0,60	0,54	0,70	1,00	0,50	0,50	0,80	n/a	n/a	n/a	n/a	0,86	0,60	0,86	0,60	1,00	0,60	0,47	0,70	1,00	0,70	1,00	0,20
ekaw-sigkdd	1,00	1,00	0,29	0,18	0,80	0,73	0,75	0,55	1,00	0,64	0,89	0,73	1,00	0,45	0,50	0,45	n/a	n/a	n/a	n/a	0,88	0,64	0,82	0,82	1,00	0,64	0,80	0,73	0,88	0,64	1,00	0,55
iasted-sigkdd	1,00	1,00	0,64	0,47	0,81	0,87	0,88	0,93	0,85	0,73	0,86	0,80	0,89	0,53	0,56	0,67	n/a	n/a	n/a	n/a	0,92	0,73	0,92	0,80	0,92	0,73	n/a	n/a	0,87	0,87	1,00	0,27
H-mean	1,00	1,00	0,44	0,32	0,83	0,70	0,72	0,51	0,84	0,59	0,76	0,71	0,88	0,44	0,59	0,63	0,83	0,54	0,84	0,52	0,84	0,54	0,84	0,66	0,88	0,53	0,56	0,44	0,86	0,63	0,39	0,30

n/a: valor não disponibilizado ou não reconhecido pela AlignAPI.

Teste 2

algo	Reference		SimSemantica		AML		CroMatcher		DKPAOM		GMap		JarvisOM		Lily		LogMap		LogMapC		LogMapLite		Mamba		RSDLWB		ServOMBI		XMAP		S-Match	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
cmt-conference	1,00	1,00	0,46	0,50	0,64	0,58	0,43	0,25	0,63	0,42	0,77	0,83	0,60	0,25	0,53	0,75	0,64	0,58	0,70	0,58	0,57	0,33	0,71	0,42	0,67	0,33	0,58	0,58	0,73	0,67	0,60	0,25
cmt-confOf	1,00	1,00	0,80	0,40	0,89	0,80	0,67	0,20	0,80	0,40	0,83	0,50	0,80	0,40	0,80	0,40	0,80	0,40	0,80	0,40	0,80	0,40	1,00	0,70	0,80	0,40	n/a	n/a	0,83	0,50	0,67	0,20
cmt-edas	1,00	1,00	1,00	1,00	0,89	1,00	0,67	0,75	1,00	1,00	1,00	1,00	1,00	0,88	0,64	0,88	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,80	1,00	1,00	1,00	0,22	0,63
cmt-ekaw	1,00	1,00	1,00	0,63	0,75	0,75	0,50	0,50	1,00	0,63	0,67	0,75	1,00	0,63	0,55	0,75	0,86	0,75	0,86	0,75	0,83	0,63	0,63	0,63	1,00	0,63	1,00	0,63	1,00	0,75	0,67	0,50
cmt-iasted	1,00	1,00	0,80	1,00	0,80	1,00	0,67	1,00	0,67	1,00	0,67	1,00	0,80	1,00	0,57	1,00	0,80	1,00	0,80	1,00	0,80	1,00	0,80	1,00	0,80	1,00	0,33	1,00	0,80	1,00	1,00	0,25
cmt-sigkdd	1,00	1,00	0,89	0,80	0,90	0,90	1,00	0,70	1,00	0,80	0,90	0,90	1,00	0,60	0,70	0,70	1,00	0,90	1,00	0,80	1,00	0,80	1,00	0,90	1,00	0,80	n/a	n/a	1,00	0,90	0,67	0,60
conference-confOf	1,00	1,00	0,88	0,64	0,82	0,82	0,86	0,55	0,82	0,82	0,77	0,91	0,88	0,64	0,67	0,73	0,75	0,82	0,82	0,82	0,88	0,64	0,90	0,82	0,88	0,64	n/a	n/a	0,80	0,73	0,75	0,27
conference-edas	1,00	1,00	0,88	0,50	0,69	0,64	0,70	0,50	0,89	0,57	0,64	0,64	0,88	0,50	0,41	0,50	0,90	0,64	0,89	0,57	0,88	0,50	0,82	0,64	0,88	0,50	n/a	n/a	0,89	0,57	0,11	0,29
conference-ekaw	1,00	1,00	0,57	0,35	0,78	0,78	0,53	0,39	0,53	0,35	0,62	0,57	0,71	0,22	0,62	0,70	0,65	0,48	0,67	0,43	0,62	0,35	0,68	0,57	0,67	0,35	0,44	0,48	0,65	0,48	0,67	0,09
conference-iasted	1,00	1,00	0,80	0,31	0,83	0,38	0,50	0,31	0,83	0,38	0,78	0,54	0,80	0,31	0,67	0,46	0,78	0,54	0,78	0,54	0,80	0,31	0,60	0,46	0,80	0,31	0,22	0,31	0,83	0,38	1,00	0,08
conference-sigkdd	1,00	1,00	0,80	0,67	0,83	0,83	0,80	0,67	0,90	0,75	0,85	0,92	0,80	0,33	0,71	0,83	0,77	0,83	0,83	0,83	0,89	0,67	0,91	0,83	0,89	0,67	n/a	n/a	0,82	0,75	0,80	0,33
confOf-edas	1,00	1,00	0,82	0,64	0,90	0,64	0,86	0,43	0,90	0,64	0,83	0,71	0,89	0,57	0,69	0,64	0,90	0,64	0,90	0,64	0,90	0,64	0,90	0,64	0,89	0,57	0,86	0,86	0,90	0,64	0,21	0,64
confOf-ekaw	1,00	1,00	0,90	0,45	0,94	0,80	0,77	0,50	0,93	0,65	0,85	0,85	0,88	0,35	0,79	0,75	0,93	0,70	0,93	0,65	0,83	0,50	0,88	0,70	0,89	0,40	0,73	0,80	0,93	0,70	1,00	0,25
confOf-iasted	1,00	1,00	1,00	0,44	0,80	0,44	0,67	0,22	0,80	0,44	0,60	0,67	1,00	0,44	0,46	0,67	1,00	0,44	1,00	0,44	1,00	0,44	0,83	0,56	1,00	0,44	0,40	0,67	0,80	0,44	1,00	0,22
confOf-sigkdd	1,00	1,00	1,00	0,67	1,00	0,83	1,00	0,67	1,00	0,67	0,80	0,67	1,00	0,67	0,17	0,17	1,00	0,67	1,00	0,67	1,00	0,67	1,00	0,67	1,00	0,67	0,83	0,83	1,00	0,67	1,00	0,17
edas-ekaw	1,00	1,00	0,59	0,53	0,79	0,58	0,60	0,47	0,73	0,58	0,65	0,68	0,71	0,26	0,67	0,63	0,73	0,58	0,71	0,53	0,69	0,47	0,77	0,53	0,75	0,47	0,74	0,74	0,79	0,58	0,44	0,37
edas-iasted	1,00	1,00	0,67	0,42	0,82	0,47	0,57	0,42	0,88	0,37	0,82	0,47	0,83	0,26	0,50	0,37	0,88	0,37	0,88	0,37	0,88	0,37	0,82	0,47	0,88	0,37	0,33	0,37	0,88	0,37	0,17	0,05
edas-sigkdd	1,00	1,00	1,00	0,64	1,00	0,64	1,00	0,73	1,00	0,64	1,00	0,64	1,00	0,64	0,63	0,45	1,00	0,64	1,00	0,64	1,00	0,64	1,00	0,64	1,00	0,64	n/a	n/a	1,00	0,64	0,33	0,55
ekaw-iasted	1,00	1,00	1,00	0,60	0,88	0,70	1,00	0,60	1,00	0,60	0,54	0,70	1,00	0,50	0,50	0,80	n/a	n/a	n/a	n/a	0,86	0,60	0,86	0,60	1,00	0,60	0,47	0,70	1,00	0,70	1,00	0,20
ekaw-sigkdd	1,00	1,00	1,00	0,64	0,80	0,73	0,75	0,55	1,00	0,64	0,89	0,73	1,00	0,45	0,50	0,45	n/a	n/a	n/a	n/a	0,88	0,64	0,82	0,82	1,00	0,64	0,80	0,73	0,88	0,64	1,00	0,55
iasted-sigkdd	1,00	1,00	0,92	0,80	0,81	0,87	0,88	0,93	0,85	0,73	0,86	0,80	0,89	0,53	0,56	0,67	n/a	n/a	n/a	n/a	0,92	0,73	0,92	0,80	0,92	0,73	n/a	n/a	0,87	0,87	1,00	0,27
H-mean	1,00	1,00	0,80	0,56	0,83	0,70	0,72	0,51	0,84	0,59	0,76	0,71	0,88	0,44	0,59	0,63	0,83	0,54	0,84	0,52	0,84	0,54	0,84	0,66	0,88	0,53	0,56	0,44	0,86	0,63	0,39	0,30

n/a: valor não disponibilizado ou não reconhecido pela AlignAPI.

Teste 3

algo	Reference		SimSemantica		AML		CroMatcher		DKPAOM		GMap		JarvisOM		Lily		LogMap		LogMapC		LogMapLite		Mamba		RSDLWB		ServOMBI		XMAP		S-Match	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
cmt-conference	1,00	1,00	0,62	0,67	0,64	0,58	0,43	0,25	0,63	0,42	0,77	0,83	0,60	0,25	0,53	0,75	0,64	0,58	0,70	0,58	0,57	0,33	0,71	0,42	0,67	0,33	0,58	0,58	0,73	0,67	0,60	0,25
cmt-confOf	1,00	1,00	0,80	0,40	0,89	0,80	0,67	0,20	0,80	0,40	0,83	0,50	0,80	0,40	0,80	0,40	0,80	0,40	0,80	0,40	0,80	0,40	1,00	0,70	0,80	0,40	n/a	n/a	0,83	0,50	0,67	0,20
cmt-edas	1,00	1,00	1,00	1,00	0,89	1,00	0,67	0,75	1,00	1,00	1,00	1,00	1,00	0,88	0,64	0,88	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,80	1,00	1,00	1,00	0,22	0,63
cmt-ekaw	1,00	1,00	1,00	0,63	0,75	0,75	0,50	0,50	1,00	0,63	0,67	0,75	1,00	0,63	0,55	0,75	0,86	0,75	0,86	0,75	0,83	0,63	0,63	0,63	1,00	0,63	1,00	0,63	1,00	0,75	0,67	0,50
cmt-iasted	1,00	1,00	1,00	1,00	0,80	1,00	0,67	1,00	0,67	1,00	0,67	1,00	0,80	1,00	0,57	1,00	0,80	1,00	0,80	1,00	0,80	1,00	0,80	1,00	0,80	1,00	0,33	1,00	0,80	1,00	1,00	0,25
cmt-sigkdd	1,00	1,00	0,89	0,80	0,90	0,90	1,00	0,70	1,00	0,80	0,90	0,90	1,00	0,60	0,70	0,70	1,00	0,90	1,00	0,80	1,00	0,80	1,00	0,90	1,00	0,80	n/a	n/a	1,00	0,90	0,67	0,60
conference-confOf	1,00	1,00	0,88	0,64	0,82	0,82	0,86	0,55	0,82	0,82	0,77	0,91	0,88	0,64	0,67	0,73	0,75	0,82	0,82	0,82	0,88	0,64	0,90	0,82	0,88	0,64	n/a	n/a	0,80	0,73	0,75	0,27
conference-edas	1,00	1,00	0,88	0,50	0,69	0,64	0,70	0,50	0,89	0,57	0,64	0,64	0,88	0,50	0,41	0,50	0,90	0,64	0,89	0,57	0,88	0,50	0,82	0,64	0,88	0,50	n/a	n/a	0,89	0,57	0,11	0,29
conference-ekaw	1,00	1,00	0,77	0,43	0,78	0,78	0,53	0,39	0,53	0,35	0,62	0,57	0,71	0,22	0,62	0,70	0,65	0,48	0,67	0,43	0,62	0,35	0,68	0,57	0,67	0,35	0,44	0,48	0,65	0,48	0,67	0,09
conference-iasted	1,00	1,00	0,80	0,31	0,83	0,38	0,50	0,31	0,83	0,38	0,78	0,54	0,80	0,31	0,67	0,46	0,78	0,54	0,78	0,54	0,80	0,31	0,60	0,46	0,80	0,31	0,22	0,31	0,83	0,38	1,00	0,08
conference-sigkdd	1,00	1,00	0,80	0,67	0,83	0,83	0,80	0,67	0,90	0,75	0,85	0,92	0,80	0,33	0,71	0,83	0,77	0,83	0,83	0,83	0,89	0,67	0,91	0,83	0,89	0,67	n/a	n/a	0,82	0,75	0,80	0,33
confOf-edas	1,00	1,00	0,82	0,64	0,90	0,64	0,86	0,43	0,90	0,64	0,83	0,71	0,89	0,57	0,69	0,64	0,90	0,64	0,90	0,64	0,90	0,64	0,90	0,64	0,89	0,57	0,86	0,86	0,90	0,64	0,21	0,64
confOf-ekaw	1,00	1,00	0,90	0,45	0,94	0,80	0,77	0,50	0,93	0,65	0,85	0,85	0,88	0,35	0,79	0,75	0,93	0,70	0,93	0,65	0,83	0,50	0,88	0,70	0,89	0,40	0,73	0,80	0,93	0,70	1,00	0,25
confOf-iasted	1,00	1,00	1,00	0,44	0,80	0,44	0,67	0,22	0,80	0,44	0,60	0,67	1,00	0,44	0,46	0,67	1,00	0,44	1,00	0,44	1,00	0,44	0,83	0,56	1,00	0,44	0,40	0,67	0,80	0,44	1,00	0,22
confOf-sigkdd	1,00	1,00	1,00	0,67	1,00	0,83	1,00	0,67	1,00	0,67	0,80	0,67	1,00	0,67	0,17	0,17	1,00	0,67	1,00	0,67	1,00	0,67	1,00	0,67	1,00	0,67	0,83	0,83	1,00	0,67	1,00	0,17
edas-ekaw	1,00	1,00	0,59	0,53	0,79	0,58	0,60	0,47	0,73	0,58	0,65	0,68	0,71	0,26	0,67	0,63	0,73	0,58	0,71	0,53	0,69	0,47	0,77	0,53	0,75	0,47	0,74	0,74	0,79	0,58	0,44	0,37
edas-iasted	1,00	1,00	0,67	0,42	0,82	0,47	0,57	0,42	0,88	0,37	0,82	0,47	0,83	0,26	0,50	0,37	0,88	0,37	0,88	0,37	0,88	0,37	0,82	0,47	0,88	0,37	0,33	0,37	0,88	0,37	0,17	0,05
edas-sigkdd	1,00	1,00	1,00	0,64	1,00	0,64	1,00	0,73	1,00	0,64	1,00	0,64	1,00	0,64	0,63	0,45	1,00	0,64	1,00	0,64	1,00	0,64	1,00	0,64	1,00	0,64	n/a	n/a	1,00	0,64	0,33	0,55
ekaw-iasted	1,00	1,00	1,00	0,60	0,88	0,70	1,00	0,60	1,00	0,60	0,54	0,70	1,00	0,50	0,50	0,80	n/a	n/a	n/a	n/a	0,86	0,60	0,86	0,60	1,00	0,60	0,47	0,70	1,00	0,70	1,00	0,20
ekaw-sigkdd	1,00	1,00	1,00	0,64	0,80	0,73	0,75	0,55	1,00	0,64	0,89	0,73	1,00	0,45	0,50	0,45	n/a	n/a	n/a	n/a	0,88	0,64	0,82	0,82	1,00	0,64	0,80	0,73	0,88	0,64	1,00	0,55
iasted-sigkdd	1,00	1,00	0,92	0,80	0,81	0,87	0,88	0,93	0,85	0,73	0,86	0,80	0,89	0,53	0,56	0,67	n/a	n/a	n/a	n/a	0,92	0,73	0,92	0,80	0,92	0,73	n/a	n/a	0,87	0,87	1,00	0,27
H-mean	1,00	1,00	0,83	0,58	0,83	0,70	0,72	0,51	0,84	0,59	0,76	0,71	0,88	0,44	0,59	0,63	0,83	0,54	0,84	0,52	0,84	0,54	0,84	0,66	0,88	0,53	0,56	0,44	0,86	0,63	0,39	0,30

n/a: valor não disponibilizado ou não reconhecido pela AlignAPI.