UFAM

UNIVERSIDADE FEDERAL DO AMAZONAS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Informática

L-*MATCH*: MAPEAMENTO SEMÂNTICO ENTRE ONTOLOGIAS UTILIZANDO CLASSIFICAÇÃO SUPERVISIONADA

FABRÍCIO D'MORISON DA SILVA MARINHO

MANAUS
2008

UFAM

UNIVERSIDADE FEDERAL DO AMAZONAS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Informática

FABRÍCIO D'MORISON DA SILVA MARINHO

L-MATCH: MAPEAMENTO SEMÂNTICO ENTRE ONTOLOGIAS UTILIZANDO CLASSIFICAÇÃO SUPERVISIONADA

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Informática, área de concentração Inteligência Artificial e Recuperação de Informação.

MANAUS
2008

Todos me criticam por eu ser diferente, mas eu rio deles por serem todos iguais (...), loucos como eu vivem pouco, mas vivem como querem, portanto não me importa se não houver o amanhã, pois me deram a vida e não a eternidade!

Charles Chaplin

Acontecem tantas coisas ruins no mundo as quais não podemos evitar, mas nem por isso devemos nos fechar pra ele! Somos capazes de muito durante a vida tão curta de um humano: rir, chorar, odiar e até amar. Nada é em vão! Por isso, por mais longe que eu vá, a verdade é que sempre estarei aqui!

Fabrício D'Morison

RESUMO

Descobrir automaticamente relações semânticas entre ontologias é uma tarefa de grande importância para Integração de Informação. Entende-se por **mapeamento** ou **alinhamento semântico** o problema de estabelecer analogias na forma de **axiomas ponte** entre conceitos de ontologias distintas, o que não é abordado pela maioria dos sistemas de alinhamento devido à dificuldade de tratar heterogeneidade semântica. Adotando o formalismo OWL, este trabalho objetiva explorar o problema de forma diferenciada, uma vez que recorre a Classificação Supervisionada: o método fundamenta-se na possibilidade de aplicar restrições lógicas comparando a instanciação dos conceitos das ontologias a mapear. Inicialmente, uma indução (**propagação *bottom-up*)** generaliza os valores de interseção e similaridade calculados entre conceitos durante a etapa de classificação. Sem perder a noção de direção na qual estes valores são computados, **regras de compatibilidade** entre conceitos definem mapeamentos de **equivalência**, **mais geral**, **menos geral**, **sobreposição** e **diferença.** Finalmente, estas regras são aplicadas seguindo uma **estratégia dedutiva *top-down*** capaz de computar mapeamentos mais confiáveis. O sistema desenvolvido chama-se L-*Match* (*Learning Match*), é iterativo (pode reutilizar os mapeamentos computados) e utiliza diferentes algoritmos de classificação como sub-rotina, sendo notório o desempenho do classificador ***Naive Bayes with Shrinkage*** (NB-*Shrinkage*) que ajudou o L-*Match* a alcançar precisão e revocação acima de 80%. Uma abordagem auxiliar para seleção e desambiguação de sinônimos em vocabulários especializados foi desenvolvida com base em teoria de Grafos e da Informação, objetivando incrementar a precisão da classificação e conseqüentemente do mapeamento.

Palavras-Chave: integração de informação, ontologia, mapeamento semântico, aprendizado de máquina, Naive Bayes Shrinkage.

ABSTRACT

Automatically discovering semantic relationships between ontologies plays an important role for Information Integration. We understand as **semantic mapping** or **alignment** the problem of establish analogies as **bridge axioms** between concepts of distinct ontologies, an approach not handled by most alignment systems due to difficulties of dealing with semantic heterogeneity. Adopting the OWL formalism, this work aims to explore this problem in a different way, since it uses Supervised Classification: the method is based on the possibility of creating logical restrictions between ontological concepts comparing their instantiation. Initially, an induction (**bottom-up propagation)** generalizes the classification output composed by intersection and similarity values computed between concepts. Regarding the direction notion of these computed values, **compatibility rules** define mappings of **equivalence**, **more general**, **less general**, **overlapping** and **difference** between concepts. Finally, these rules are applied deductively following a **top-down strategy** which helps predicting more reliable mappings. The **L-Match** (Learning Match) system has been developed to be iterative (reuse its own computed mappings) and to use different classification algorithms as sub-routine. However, the **Naive Bayes with Shrinkage** classifier (NB-Shrinkage) has outperformed the others notoriously, helping L-Match to reach precision and recall higher than 80%. An auxiliary approach for synonym selection and disambiguation on specialized vocabularies has been developed backed by Graph and Information Theories intending to increase both the classification and mapping precision.

Key words: information integration, ontology, semantic mapping, machine learning, naive bayes with shrinkage.

ÍNDICE DE ILUSTRAÇÕES

ÍNDICE DE TABELAS

ÍNDICE DE EQUAÇÕES

SUMÁRIO

Capítulo 1

INTRODUÇÃO

Em anos recentes, é considerável a quantidade de sistemas de informação que têm surgido para atender às necessidades de guardar, organizar e recuperar as informações de diversos domínios de conhecimento, como Biologia, Medicina, Ecologia, Agricultura, Turismo, Comércio Eletrônico, etc. Contudo, em paralelo a este crescimento surgem problemas técnicos de organização, de redundância e de interoperabilidade semântica, pois estes sistemas guardam modelagens de conhecimento locais e próprias as quais podem pertencer a múltiplos domínios inter-relacionados. Conseqüentemente, muitas dificuldades surgem quando estes sistemas precisam interoperar e trocar informação entre si, motivando as pesquisas sobre compartilhamento e integração de informação, as quais são abordadas no presente trabalho através do uso e reuso de ontologias e dos mapeamentos entre elas.

Como tendem a se tornar cada vez maiores e mais elaborados, estes sistemas demandam maior homogeneidade em suas bases de informação. O maior e mais conhecido exemplo é a WEB. Tendo-se popularizado há pouco tempo (menos de duas décadas), a Web não pára de crescer, levando a uma grande quantidade de informações heterogêneas e redundantes que inviabilizam o processamento manual, que seria muito demorado e propenso a erro, daí a necessidade de procedimentos automáticos.

Além da automação, há a necessidade de uma representação de conhecimento que privilegie a semântica, viabilizando a organização clara, objetiva e homogênea de conhecimento mesmo em ambientes originalmente heterogêneos. Isso facilita a atuação de processos automáticos como máquinas de busca, as quais ainda hoje ignoram informação semântica ou contextual. Torna-se então atrativa a utilização de ontologias para representação formal de conhecimento, uma vez que permite modelagem de alto nível de informação (estrutura e semântica), na forma de conceitos, propriedades, restrições e instâncias (UDREA, *at al.*, 2007).

Ontologias têm grande valor para a Web, especialmente para a Web Semântica. Características ímpares da Web, como o crescimento exponencial do seu já imenso volume de dados, de informações e de usuários, levam a um desenvolvimento distribuído de muitas ontologias no intuito de amenizar (ou mesmo eliminar) a heterogeneidade de informação. Disto resultam informações multiplamente representadas, caracterizando um novo cenário de problemas onde a heterogeneidade passa a ser semântica e as pesquisas se concentram em viabilizar soluções automáticas que permitam a **comunicação** entre sistemas ao estabelecer mapeamentos semânticos entre os conceitos de suas ontologias. Esta é a motivação que torna a integração automática de ontologias cada vez mais importante e desejada.

Porém, o mapeamento entre entidades de ontologias diversas ainda é uma questão a investigar. Freqüentemente, problemas de integração são intitulados como **casamento**, **alinhamento**, **mapeamento** ou **mesclagem**. Esta Dissertação cobre apenas os três primeiros casos e os considera equivalentes: não há consenso preciso sobre diferença entre casamento, alinhamento e mapeamento, mas cada autor tem preferência por uma nomenclatura ou outra. A diferença é clara apenas quando ocorre mesclagem devido às alterações que causa nas

ontologias iniciais: o produto final é uma nova ontologia ou uma nova versão de cada ontologia original.

Nesta Dissertação, tratamos de um problema ainda mais específico e complexo: o **mapeamento semântico**. Neste caso, além das entidades (conceitos, propriedades, etc.) mais similares, deseja-se descobrir a relação semântica mais forte entre as entidades. As relações são expressas por axiomas, tornando o mapeamento mais expressivo e útil do que mapeamentos que mensuram apenas similaridade. Segundo a Lógica, axiomas são verdades auto-evidentes e que por isso não requerem prova, podendo consistir de proposições assumidas ou de regras e princípios universalmente aceitos. Além disso, o mapeamento é considerado mais elegante quando as evidências investigadas também são axiomas e, portanto, semânticas. Este é o caso de (GIUNCHIGLIA, *at al.*, 2005), onde uma taxonomia é transformada num conjunto de proposições lógicas submetidas a um raciocinador.

Por mais elegante que seja, nenhuma abordagem de mapeamento processa única e exclusivamente evidências semânticas. Neste estado-da-arte, é impossível ignorar o uso de heurísticas de similaridade num primeiro momento de mapeamento. Raciocinadores lógicos em nada ajudarão neste ponto, já que nenhum mapeamento *a priori* (fato ou regra) é conhecido **entre** as ontologias consideradas. Felizmente, raciocinadores estatísticos são bastante adequados para lidar com a informalidade da heterogeneidade inicial das evidências a investigar, levando a supostos mapeamentos sem a obrigatoriedade de utilizar conhecimento externo. Exemplos são as heurísticas de classificação supervisionada exploradas neste trabalho, que além de consolidadas como tecnologia, desempenham satisfatoriamente a tarefa de descobrir relações de pertinência desconhecidas entre instâncias e conceitos (mapeamento instância-conceito). A partir daí sim, quando algum tipo de mapeamento preliminar puder ser assumido com boa confiança (como as relações de

pertinência, por exemplo), justificar-se-á alocar esforços para aplicação de raciocínio lógico (UDREA, *at al.*, 2007)(GIUNCHIGLIA, *at al.*, 2005)(HAASE e MOTIK, 2005) na expectativa de ajustar o mapeamento inicial, reparando erros e conseqüentemente aumentando a corretude e a completude do processo como um todo.

Abordagens atuais de integração semântica costumam ser fortemente dependentes de fontes adicionais de conhecimento, que impõem dependências para o resultado final da integração. É o caso da utilização de um vocabulário global, controlado e unificador (uma ontologia, um dicionário, etc.) imposto como filtro às ontologias locais, na tentativa de criar mapeamentos que nunca sejam *ad hoc*, isto é, que nunca sejam criados diretamente entre ontologias locais (GIUNCHIGLIA, *at al.*, 2005). Estas abordagens não favorecem características de liberdade de descrição semântica específicas de autor e de domínio, nem de manutenção de ontologias e mapeamentos no decorrer do tempo. Pragmaticamente, é inviável depender sempre de um vocabulário global para mapear, dada a diversidade de domínios específicos e a necessidade de alterações evolutivas e corretivas nas ontologias. Sabe-se que estas alterações criam inconsistências entre mapeamentos já estabelecidos, logo quanto menos ontologias envolvidas no processo melhor. Portanto, para o bem da portabilidade de um método de mapeamento, deve-se diminuir a dependência de conhecimento externo, mas nem por isso desprezá-lo, utilizando-o apenas como Informação Auxiliar (RAHM e BERNSTEIN, 2001). Por esta razão, é um objetivo deste trabalho considerar exclusivamente mapeamentos *ad hoc*, que podem ser computados simplesmente através do uso de algoritmos que independem de conhecimento externo, mas que podem utilizá-lo como item opcional se disponível, sem imposição de dependências.

Investigar instâncias é útil para mapeamento. Sem dúvida, o aprendizado de relações taxonômicas pode ser visto intuitivamente como uma tarefa de classificação (CIMIANO, *at*

al., 2004). Na maioria das vezes, a classificação em ontologias (anotação de instâncias) é feita por especialistas humanos e tende a capturar precisamente a relação entre sintaxe e semântica dos conceitos. Esta classificação é atrativa por ter a propriedade de possibilitar, até certo ponto, a inferência da semântica de um conceito com base no seu comportamento sintático taxonômico (KORHONEN e BRISCOE, 2004). Mesmo não sendo suficiente para uma inferência semântica completa, a classificação é capaz de capturar generalizações de um grande número de propriedades, explícitas ou implícitas nas definições originais do autor da ontologia, sendo útil para resolver situações não cobertas por conhecimento léxico, como dicionários semânticos, os quais nunca são completos.

Entender o significado que delimita os conceitos ontológicos é um problema chave para o mapeamento e, segundo (SOROKINE, *at al.*, 2005), isso pode ser feito através de instâncias e propriedades. Vale ressaltar que, na medida do possível, não se deve contrariar o entendimento que o autor da ontologia pretendia expressar no momento que especificou cada conceito, o que acontece, por exemplo, quando se utiliza uma ontologia global impondo a visão de mundo de terceiros. Um dos melhores recursos que o autor pode usar para evidenciar sua idéia sobre cada conceito é a utilização de exemplos. Esta idéia é reforçada por (KENT, 2000) ao afirmar teoricamente que instâncias são boas evidências para relacionar conceitos que receberam interpretações dadas por comunidades diferentes. A abrangência de um conceito dada pela enumeração momentânea de instâncias pode apenas ser contrariada pela definição de uma regra ou restrição sobre o mesmo conceito. Porém, a utilização de regras e restrições (ou axiomas de domínio) como evidências para mapeamento entre ontologias ainda é uma questão em aberto (UDREA, *at al.*, 2007), como reiterado por (FüRST e TRICHET, 2005):

> *"Certas evidências, como as propriedades algébricas, equivalências e disjunções, não são suficientemente usadas pela comunidade ao desenvolver ontologias para serem consideradas como material de similaridade. Já no caso dos axiomas, que incluem as regras e as restrições, não existe pesquisa nem suporte prático suficiente".*

Vimos que aprender através de exemplos é uma boa estratégia, mas para isso é necessário um sistema capaz de classificar uma grande porção de instâncias com uma precisão aceitável. Isso é um desafio, pois é sabido que classificadores funcionam melhor conforme mais instâncias estão disponíveis para treiná-lo, sendo que a quantidade de instâncias nas ontologias não costuma ser grande. Contudo, esta situação é flexível: poucas instâncias possibilitam explorar técnicas de classificação mais eficazes as quais seriam ineficientes em bases de dados muito grandes, compensando a escassez de instâncias em muitas situações.

Em situações mais práticas, é um desperdício desenvolver uma ontologia e não utilizá-la para anotar instâncias, especialmente no domínio biológico (agricultura, ecologia, medicina, piscicultura, genética, etc.) onde as ontologias têm encontrado aceitação, investimento e pesquisa para a anotação de dados antes em bases de dados comuns sem suporte à semântica. A verdade é que o mapeamento só é útil quando feito em ontologias que são usadas na prática e os dois maiores exemplos atuais são a *Gene Ontology* (GO) e a *Protein Ontology* (PO), vastamente anotadas e utilizadas, como é esperado que ocorra com ontologias diversas no futuro. Portanto, a atual escassez de ontologias instanciadas não constitui argumento para ignorar instâncias durante mapeamentos.

Sendo assim, a falta de uma abordagem completa sobre mapeamento semântico entre ontologias e de uma abordagem satisfatória para classificação supervisionada aplicada para este fim, motivou os trabalhos desta Dissertação, cujo objetivo é investigar evidências, métricas e métodos de classificação para criar uma nova abordagem de boa qualidade para

comparação e conseqüente mapeamento entre ontologias. Um protótipo do sistema proposto foi implementado, testado e chamado de L-*Match*, do inglês *Learning Match*. O L-*Match* é orientado à instanciação de conceitos e pode ser compreendido como um operador semântico *f* que toma duas ontologias *A* e *B* e sugere a **relação semântica mais forte** entre pares de conceitos como segue:

$$f : Conceitos[A] \times Conceitos[B] \rightarrow \text{Re}\,lação$$

A principal estratégia utilizada no L-*Match* para comparação de conceitos é extensional (Seção 2.1.2.1): similar ao GLUE (ANHAI, *at al.*, 2004), as extensões investigadas são as instâncias dos conceitos as quais, caracterizadas por suas propriedades, permitem tratar o problema de mapeamento semântico como um problema de Classificação Supervisionada. Por fim, agora similar ao S-*Match* (GIUNCHIGLIA, *at al.*, 2005), a saída do mapeador é sempre uma relação ou axioma dentre **equivalência (≡)**, **mais geral (⊒)**, **menos geral (⊏)**, **sobreposto (⊓)** e **diferença (≢)**.

Para especificação de ontologias, adotamos a *Ontology Web Language* - OWL (Seção 2.1.1), cuja popularidade é crescente. Em princípio, recursos específicos de OWL não foram explorados, como raciocínio envolvendo Lógica de Descrição em OWL-DL, contudo foi inevitável que características suas influenciassem o trabalho. Taxonomias em OWL permitem bastante sobreposição (interseção) entre conceitos, pois suportam herança e instanciação múltiplas e são do tipo não-particionadas, por permitir sobreposição entre conceitos irmãos. Estas características facilitam a especificação de ontologias, mas criam um cenário onde aplicar classificação automática não é trivial, pois excessos de sobreposição dificultam a separação precisa dos conceitos, bem como impedem a utilização de algumas restrições comuns em hierarquias taxonômicas mais tradicionais, dificultando a tomada de decisão.

Felizmente, os resultados experimentais mostraram que os classificadores foram capazes de lidar com este cenário, talvez porque ontologias são bem formadas e por isso bastante adequadas para treinar algoritmos de Aprendizado de Máquina.

Assumiremos que as ontologias utilizadas no nosso estudo possuem instâncias. Sempre é interessante usar a maior quantidade de informação possível: uma vez que usamos apenas uma fonte de evidência ou apenas um algoritmo, ficamos muito dependentes deles, o que pode ser perigoso. A combinação das fontes de informação, objetivando que o mapeamento final seja semântico, de fato pode melhorar os resultados. Por isso, apesar de nos concentrarmos numa abordagem de mapeamento específica, não defendemos utilizá-la exclusivamente, mas sim investigá-la da melhor forma possível. Classificação automática tem a atrativa vantagem de ser flexível quanto à combinação de evidências e de algoritmos, por isso experimentamos seis algoritmos de classificação, bem como combinações entre eles, incluindo algoritmos clássicos como *k*NN (SHAKHNAROVICH, *at al.*, 2006), *Naive Bayes* (RISH, 2001) e SVM (BURGES, 1998), e também algoritmos menos conhecidos como *NB-Shrinkage* (Seção 2.5.2), Máxima Entropia (NIGAMY, *at al.*, 1999) e TF-IDF (JOACHIMS, 1997).

Os experimentos foram orientados a eficácia (qualidade de resposta) e não totalmente a eficiência (velocidade de resposta) e foram conduzidos especialmente sobre um par de ontologias oriundas dos domínios da Ecologia e da Agricultura. Estas ontologias possuem aplicação real, são relativamente grandes (entre 64 a 200 conceitos) e complexas, o que as torna difíceis de mapear. Os resultados experimentais do nosso protótipo são promissores, pois apresentaram valores altos e equilibrados entre precisão (corretude) e revocação (completude).

1.1 CONTRIBUIÇÃO

A principal contribuição deste trabalho é o desenvolvimento de uma abordagem para tratar mapeamento semântico e não mapeamento sintático, sendo que a abordagem é diferenciada uma vez que aplica técnicas de Aprendizagem de Máquina, no caso Classificação Supervisionada, o que atualmente é ignorado pela maioria dos trabalhos.

Uma grande diferença em relação a abordagens anteriores que utilizaram Aprendizagem de Máquina é que não aplicamos apenas um algoritmo específico. Investigamos e comparamos diferentes algoritmos de classificação, bem como combinações entre eles, em busca da mais eficaz forma de *classificar-para-mapear* semanticamente. O L-*Match* também não exige que conceitos mais genéricos sejam diretamente instanciados, propondo um método intuitivo e eficaz baseado em Redes Bayesianas para generalizar analogias entre conceitos através de uma propagação *bottom-up*, o que pode ser reutilizado em qualquer outra abordagem que calcule similaridade entre conceitos organizados em hierarquia taxonômica.

A fase de pré-processamento das ontologias é vital para as fases seguintes. Dada essa observação, desenvolvemos uma abordagem prática e rápida para identificar e desambiguar sinônimos nos rótulos de conceitos, instâncias e propriedade: assume-se que sinônimos são análogos a cliques num grafo que conecta palavras similares, que sinônimos carregam informação (veja entropia, Seção 2.4) e que vocabulários ontológicos são restritos (específicos de domínio). Conseqüentemente, a desambiguação mantém apenas sinônimos com mais informação, isto é, os mais relevantes ao domínio em questão.

1.2 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta Dissertação está organizada em 6 capítulos a começar por esta introdução. No Capítulo 2 estão os principais termos e conceitos necessários para uma boa compreensão do trabalho. No Capítulo 3 está a revisão de literatura sobre os principais trabalhos correlatos. O Capítulo 4 é o mais importante, pois é nele que a abordagem desenvolvida ao longo deste trabalho é descrita. No Capítulo 5 são apresentados os experimentos com a abordagem e os resultados obtidos são comentados. Por fim, no Capítulo 6, faz-se o fechamento do trabalho com a exposição de conclusões e sugestões de trabalhos futuros.

Capítulo 2

CONCEITOS E TERMINOLOGIA

Para este capítulo, foram selecionados, enumerados e revisados os principais conceitos necessários para a boa compreensão dos capítulos seguintes. Começando pela definição de ontologia, revisamos também aspectos principais sobre a linguagem OWL e sobre mapeamento entre ontologias com ênfase em mapeamento semântico. Não obstante, apresentamos brevemente o problema da enumeração de cliques máximos, prosseguindo diretamente para a discussão sobre combinação de evidências utilizando Redes Bayesianas e entropia. Na seqüência, apresentamos princípios básicos comuns a diferentes métodos de classificação supervisionada, finalizando o capítulo com conceitos sobre as métricas de avaliação de eficácia que aplicamos sobre os experimentos com o mapeador L-*Match*.

2.1 ONTOLOGIAS

O termo **ontologia** vem do grego *ontos+logoi* que significa "conhecimento do ser". Popularizado pela comunidade de Inteligência Artificial, o termo é originário da Filosofia e diz respeito ao estudo do que existe no mundo, ao estudo do "ser enquanto ser" e por isso compõe uma subárea da Metafísica, a parte da Filosofia preocupada com a existência.

Em Ciência da Computação, ontologias são estruturas especiais relacionadas à tecnologia que visa estruturar o conhecimento, atribuindo significado e contexto a uma vasta

gama de entidades desde conceitos e idéias abstratas até documentos e imagens concretas. Por isso, as ontologias são empregadas como **bases semânticas** ou **bases de conhecimento** contendo uma modelagem formal de um determinado domínio de conhecimento (Genética, Turismo, Religião, etc.), relacionando entidades similares e agrupando-as em classes. Ontologias podem ainda ser definidas como segue:

> *"Uma ontologia identifica classes, cada qual caracterizada por propriedades que todos os elementos desta classe compartilham e as organiza hierarquicamente. Isto também inclui importantes relações entre classes e elementos, em um domínio de conhecimento específico", (CHANDRASEKARAN,* at al.*, 1999).*

O objetivo da representação formal de conhecimento é que a especificação utilizada (uma ontologia, por exemplo) seja legível para computadores, viabilizando o processamento automático da semântica (informação contextual) que rege determinados objetos (como páginas da Web). As ontologias servem para compartilhar conhecimento entre sistemas, para classificar e aplicar inferências sobre os objetos de um domínio e para adicionar contexto nas buscas por informação, permitindo também que um sistema possa aprender conforme se agrega conhecimento na ontologia, manual ou automaticamente. Em relação a outras tecnologias, as ontologias estão em fase relativamente recente de pesquisa e seu uso é quase que exclusivamente acadêmico, porém vem ganhando mais popularidade e aplicabilidade no decorrer do tempo.

Independentemente do formalismo utilizado para descrevê-las, ontologias são compostas fundamentalmente pelas **primitivas conceituais** do domínio modelado, as quais incluem os **conceitos** e as suas **propriedades**. Os conceitos são conjuntos abstratos de instâncias identificadas pelas mesmas propriedades. As propriedades, por sua vez, podem ser **atributos** elementares ou **relacionamentos** entre conceitos. Relacionamentos também são

considerados conceitos, porém são conceitos compostos já que criam ligações específicas entre outros conceitos. Os tipos de relacionamento são muitos, sendo mais popularmente encontrados nas ontologias os relacionamentos taxonômicos (relação de herança) organizando os conceitos em hierarquias.

Ontologias podem também conter **regras lógicas** que criam restrições sobre conceitos e propriedades, o que aumenta o grau de formalidade da ontologia e permite a descrição mais fidedigna do domínio em questão. Além de conceitos, propriedades e regras, ontologias podem receber anotações de **instâncias**, também chamadas objetos, exemplos ou indivíduos. As instâncias são a manifestação, a personificação dos conceitos e propriedades concretizados no mundo real, constituindo por isso a base mais elementar da ontologia. Apesar disso, instâncias são consideradas **extensões**, pois não é obrigatório declarar explicitamente instâncias numa ontologia. Contudo, um dos objetivos da Engenharia Ontológica é justamente a classificação e anotação de instâncias, mesmo que em um primeiro momento elas não estejam explicitamente declaradas nas ontologias.

Diferentes infra-estruturas e formalismos estão disponíveis e voltados especificamente para a modelagem de ontologias. Dentre os formalismos, vale ressaltar *Lógica de Primeira Ordem* (SOWA, 2000), *Frames* (MINSKY, 1975) e *Lógica de Descrição* (BAADER, *at al.*, 2002). Quanto à infra-estrutura, a linguagem que o W3C[1] recomenda para especificação de ontologias é a *Ontology Web Language* (OWL), a qual vem se tornando um padrão cada vez mais forte, impulsionando especialmente pelo desenvolvimento da ferramenta *Protégé*[2].

[1] http://www.w3.org/
[2] http://protege.stanford.edu/

2.1.1 OWL – Ontology Web Language

Ontologias consideradas formais são especificadas em linguagens que envolvem algum tipo de lógica. Dos diferentes padrões disponíveis atualmente, a linguagem adotada no presente trabalho é a OWL, *Ontology Web Language*, pois é um padrão aberto. Isso é muito importante, pois seu uso é mais difundido e bem suportado por ferramenta gratuita e com código aberto, facilitando os trabalhos tanto dos desenvolvedores de ontologias quanto dos desenvolvedores de aplicações. Por isso, a OWL tem sido bastante utilizada para a construção de inúmeras ontologias espalhadas por repositórios Web, contrastando com outras linguagens que são utilizadas em projetos específicos, como ocorre com projeto conhecidos a exemplo do Cyc[3] e do OBO[4].

OWL em inglês significa "coruja", animal que ficou conhecido como símbolo grego da sabedoria, já que sempre acompanhava *Athena*, a deusa da sabedoria e da guerra justa. Uma das características marcantes de OWL é o fato de usar Visão de Mundo Aberto. Esta visão de mundo assume que não se sabe tudo que é possível saber sobre o mundo, por mais que se tenha uma grande base de conhecimento. Tecnicamente, se uma suposta verdade não pode ser provada apenas com o conhecimento disponível sobre o mundo, é impossível concluir que ela seja falsa e, portanto, permanece desconhecida. Em outras palavras, nenhuma verdade pode ser falsa a não ser que seja explicitamente declarada como falsa. É o contrário do que ocorre com outras linguagens, a exemplo de Prolog, que utilizam Visão de Mundo Fechado onde o conhecimento disponível sobre o mundo é suficiente para provar qualquer coisa como falso ou como verdadeiro.

[3] http://www.cyc.com/
[4] http://www.obofoundry.org/

A linguagem OWL visa aproximar ontologias e Web, contribuindo significativamente para o desenvolvimento da Web Semântica. A idéia é atribuir significado explícito aos dados publicados na rede mundial, que então se tornaria um ambiente facilitador para aplicações inteligentes e para a interoperabilidade entre as mesmas. Considerado um dos trabalhos pioneiros e ainda recentes acerca de recomendações do W3C relativas a padrões da Web Semântica, a linguagem OWL é o resultado mais atual dos trabalhos, estudos e lições aprendidas durante o desenvolvimento de tecnologias anteriores sobre modelagem d conhecimento, pois sua estrutura é implementada com base em XML, RDF e RDF-*Schema*, bem como, na qualidade de linguagem de ontologia, é considerada uma revisão e evolução das linguagens OIL, DAML e DAML+OIL.

Facilitando-se destes modelos, OWL herda a capacidade genérica para rotular conteúdo através da sintaxe XML e herda a declaração de objetos/recursos e suas relações a partir de RDF, utilizando RDF-*Schema* para agrupar os objetos em classes e propriedades organizadas em hierarquias taxonômicas. Sendo a última camada nesta pilha de modelos, OWL chega para adicionar maior riqueza e detalhamento semânticos, finalmente conferindo o nível de ontologia para o modelo. Resumindo, uma ontologia OWL é um conjunto de triplas RDF imbuídas de significado.

O *trade-off* entre **expressividade** e **decidibilidade** é abordado por OWL. Quanto mais expressiva for uma linguagem, maior a liberdade e facilidade para declarar conhecimento, em detrimento da completude e da decidibilidade alcançável por um algoritmo de raciocínio. Para isso, OWL divide-se em três sub-linguagens que variam em graus de expressividade/decidibilidade. A primeira é **OWL-*Lite***, cujo objetivo é ser mais simples e menos expressiva, sendo porém mais fácil de tratar computacionalmente, pois oferece suporte apenas a recursos básicos como taxonomias e restrições simples. A segunda linguagem é

OWL-DL a qual objetiva um melhor equilíbrio utilizando Lógica de Descrição para permitir maior expressividade ao passo que garante a completude e a decidibilidade do raciocínio. Finalmente, o máximo de expressividade e liberdade pode ser obtido apenas com **OWL-*Full***, ao custo da eliminação das garantias de completude e de decidibilidade.

Poucos sistemas consideram o uso de OWL-*Full* devido ao risco de intratabilidade. No caso dos sistemas de mapeamento entre ontologias, a maioria utiliza OWL-*Lite* e uma minoria tenta explorar OWL-DL. Apesar desta Dissertação atualmente não explorar recursos adicionais de Lógica de Descrição (como padrões de inferência), o L-*Match* aceita ontologias especificadas tanto em OWL-*Lite* como em OWL-DL.

2.1.2 Mapeamento entre Ontologias

Toda atividade de integração de informação (relacionar, comunicar, transferir, interoperar, fundir, etc.) depende de alguma rotina de mapeamento. Esta habilidade de combinar modelos automaticamente, relacionando e integrando suas estruturas, é uma área de pesquisa cada vez mais sólida sobre um problema comum a diferentes comunidades, como casamento de grafos (KLINGER e AUSTIN, 2005), Visão Computacional (PELILLO, *at al.*, 1998) e aplicações de bancos de dados (RAHM e BERNSTEIN, 2001), como gerenciamento de conhecimento científico, comércio eletrônico e *data warehousing*. No mais, a integração de ontologias tem uma tênue correlação com problemas de integração de esquemas de bancos relacionais (UDREA, *at al.*, 2007), a diferença é que o compromisso com semântica é rigorosamente maior nas ontologias.

Como (EUZENAT e VALTCHEV, 2003), classificamos as abordagens de mapeamento em função das fontes de evidência e das técnicas de comparação:

- **Comparação Terminológica –** compara rótulos de entidades. Inclui técnicas de processamento de *string* (distância de edição, prefixo/sufixo e N-Gram), técnicas baseadas em linguagem (*tokenization*, *stemming* e *stopwords*) ou em recursos lingüísticos, como *thesaurus* e dicionários taxonômicos.
- **Comparação da Estrutura Interna –** compara evidências da estrutura interna de entidades (domínio, *range*, cardinalidade, etc.).
- **Comparação da Estrutura Externa** – compara as estruturas de relacionamento entre entidades, como a árvore taxonômica e o grafo cíclico formado por outras relações. Neste item, algoritmos para grafos são comuns.
- **Comparação Extensional –** compara extensões das entidades. Extensões são entidades que, a princípio, não precisam estar declaradas nas ontologias. Exemplos são comentários e instâncias. Inclui os algoritmos de classificação automática.
- **Comparação Semântica –** compara a interpretação (modelo lógico) das entidades. Inclui técnicas de inferência lógica/simbólica sobre axiomas, como Lógica de Descrição.

O L-Match utiliza comparação terminológica, da estrutura externa e extensional. Observe que o fato de uma abordagem utilizar comparação semântica, não significa que o produto do mapeamento será necessariamente semântico (vide Seção 2.1.2.1 adiante).

Definimos genericamente um **mapeador de ontologias** como um sistema em função de sua **entrada** e **saída**. Sejam então as ontologias *A*, com *m* conceitos, e *B*, com *n* conceitos:

Figura 2-1: Entrada e Saída de um Mapeador

- Um mapeador de conceitos é uma função:

$$f: Conceitos[A] \times Conceitos[B] \rightarrow Qualificador$$

- A entrada são as duas ontologias *A* e *B*;
- A saída é o mapeamento entre cada par de conceitos como triplas (a_i, b_j, q_{ij}).

Onde a_i *e* b_j são conceitos oriundos das ontologias *A* e *B*, respectivamente, e q_{ij} é o **qualificador** do mapeamento, podendo aparecer sob a forma de uma relação axiomática, um valor de similaridade na escala [0,1], ou ambas as coisas.

2.1.2.1 Mapeamento Sintático e Semântico

Revisamos nesta Seção os mesmos conceitos sobre mapeamento sintático e mapeamento semânticos introduzidos e utilizados por (BOUQUET, *at al.*, 2003) em C-OWL e (GIUNCHIGLIA, *at al.*, 2005) no S-*Match*, conceitos os quais aplicamos no L-*Match*.

Diz-se que o **mapeamento sintático** ocorre quando o qualificador *q* é um valor de similaridade entre [0,1]: o problema consiste em encontrar os pares de conceitos mais similares, isto é, mapeamentos 1-1 (um para um). Este tipo de mapeamento é bastante

limitado, pois só permite saber "*o quão similar são os conceitos*". Por isso, o qualificador *q* ideal para mapeamentos entre ontologias recai na noção básica de **axioma ponte** de forma a indicar "*como os conceitos se relacionam*". Axiomas qualificam mapeamentos mais expressivos, imbuídos de maior riqueza semântica e permitem discriminar melhor quais as propriedades que se aplicam simultaneamente a elementos de ontologias distintas.

Entende-se por **mapeamento semântico** o problema de encontrar **o axioma ponte mais forte** (ou relação semântica mais forte) entre cada par de conceitos oriundos de ontologias distintas. Como não apenas o "*mais similar*" é importante, mapeamentos semânticos são do tipo *n-n* (muitos para muitos). Em (BOUQUET, *at al.*, 2003), (GIUNCHIGLIA, *at al.*, 2005) e nesta Dissertação são utilizados cinco axiomas ponte, ordenados do mais forte ao mais fraco: **equivalência (≡)**, **mais geral (⊐)**, **menos geral (⊏)**, **sobreposto (⊓)** e **diferença (≠)**. Estes axiomas são operadores de Lógica de Descrição e representam as relações mais básicas e possíveis de se estabelecer entre conjuntos. A **escala de força** e as relações expressas por estes axiomas são melhor compreendidas observando interseções entre dois conceitos fictícios A e B em diagramas de Vem:

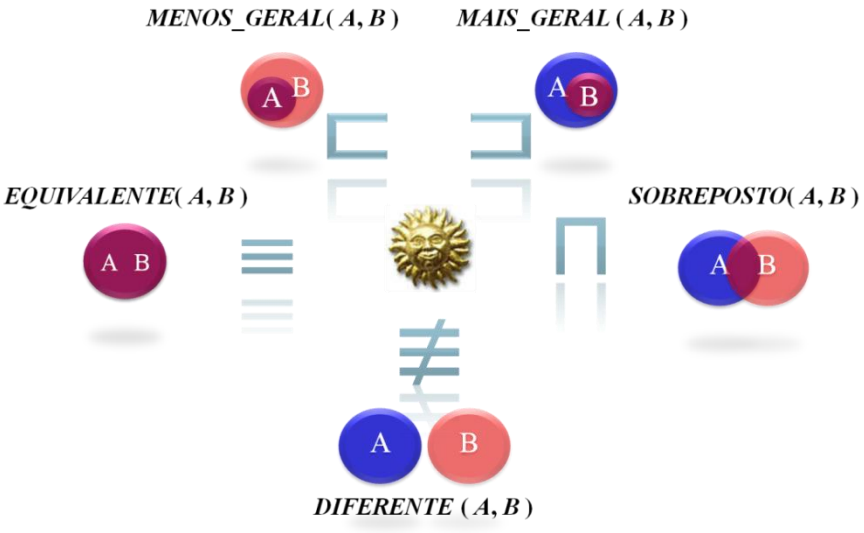


Figura 2-2: Axiomas utilizados como ponte para mapear conceitos

Quanto maior a interseção, mais forte será a relação. A ***equivalência*** é a relação mais forte por representar uma interseção total entre *A* e *B*, o que significa que suas propriedades e instâncias são as mesmas. **As relações de *mais geral* e *menos geral*** são relações inversas menos fortes que a equivalência, pois a interseção é total para apenas um dos conceitos: *A* é menos geral que *B* se toda instância de *A* pertence a *B*, porém a recíproca não é verdadeira. Estas relações de generalidade fornecem informação de herança (superclasse/subclasse), permitindo identificar quando um conceito contém ou é contido por outro. A **relação de sobreposição** é a mais fraca: é um resquício de sobreposição entre os conjuntos, onde nenhum está contido ou é equivalente ao outro. Segundo (GIUNCHIGLIA, *at al.*, 2005), nenhuma informação relevante pode ser concluída a partir da relação de sobreposição. Por fim, a **relação *diferença*** é aplicada quando os conceitos são totalmente disjuntos ($A \cap B = \varnothing$), indicando a informação de que o complemento de *A* contém *B*, e vice versa. Por ser o oposto da equivalência, a diferença também é importante.

Mapeamentos semânticos podem também ser qualificados por **axiomas complexos** formados a partir da composição entre $<\equiv, \sqsupset, \sqsubset, \sqcap, \not\equiv>$ mais a **negação($\neg$)** e eventualmente a **união($\sqcup$)**, como observado e exemplificado por (BOUQUET, *at al.*, 2003).

Ao apresentar o S-*Match*, (GIUNCHIGLIA, *at al.*, 2005) faz considerações interessantes sobre mapeamento semântico, deixando clara a diferença entre **conceito de vértice** e **conceito de rótulo**, o que normalmente causa bastante confusão e discussões desnecessárias entre especialistas: não se deve esquecer que o mapeamento é entre os conceitos atribuídos aos vértices, e não aos rótulos. Veja:

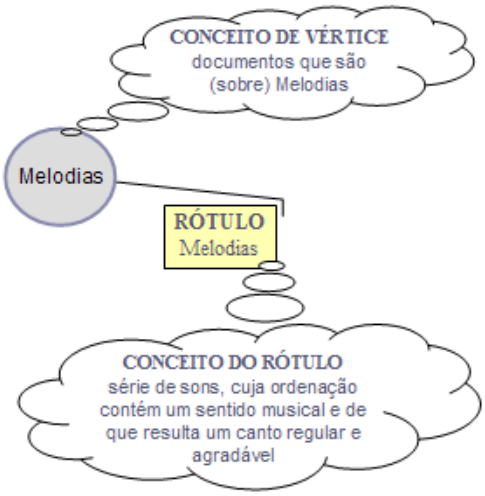


Figura 2-3: Conceito de Vértice e Conceito de Rótulo.

Neste caso, a idéia de conceito dentro da ontologia vai mais longe, pois o conceito passa a ser divisível: conceito de vértice e conceito de rótulo. No exemplo, temos um vértice rotulado como Melodia, mas o conceito do rótulo não necessariamente é igual ao conceito do vértice, apesar de que estão relacionados. Rótulos em hierarquias de classificação são usados para remeter ao real conceito do vértice que define o conjunto de documentos a serem classificados sob o vértice (ou classe). Por isso, quando falamos de *Melodias*, não necessariamente queremos definir ou referir-se exatamente a "*uma série de sons, cuja ordenação contém um sentido musical e de que resulta um canto regular e agradável*", mas sim a "*documentos que são (sobre) Melodias*". O rótulo é apenas um identificador humano, uma facilidade, uma palavra conveniente que deve ser simples e pequena o suficiente para desempenhar sua função que é tão somente terminológica. Ao ser lido por uma pessoa, o rótulo remete à semântica mais comum que ele receberia no mundo real. Porém, o conceito de vértice não é determinado pela terminologia, ele é determinado e regido pela lógica, através das propriedades e regras declaradas numa ontologia.

2.2 PROBLEMA DA ENUMERAÇÃO DE CLIQUES MÁXIMOS

Problemas de clique são clássicos na área de Otimização Combinatória, sendo muito importantes para diversas áreas, como Biologia, especialmente Genética, e Recuperação de Informação, como é o caso desta Dissertação e de trabalhos com *links* em páginas Web. Por definição, um clique é um subgrafo completo (existe aresta de todo vértice para todo vértice), veja o exemplo:

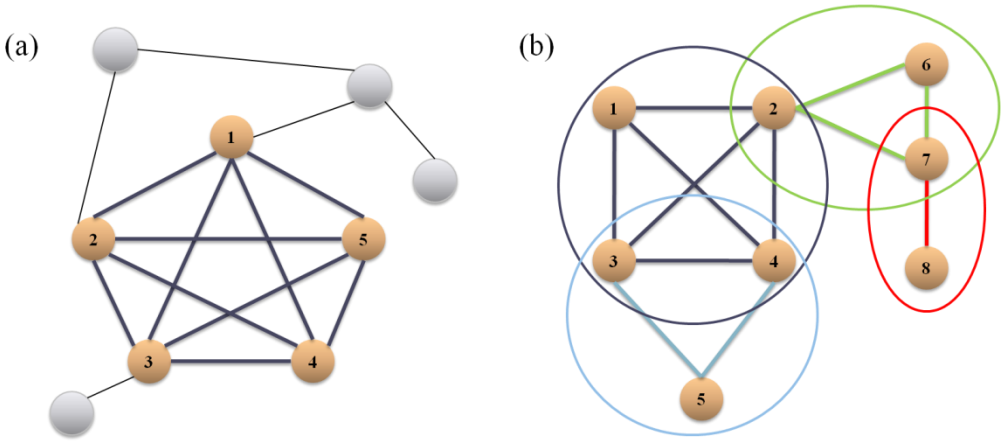


Figura 2-4: Exemplos de cliques máximos

No grafo (a) existem vários cliques menores em destaque e todos estão contidos num clique maior de tamanho cinco. Este é um **clique máximo** (DU e PARDALOS, 2007), definido como o maior clique que não está contido em outro clique. Contudo, neste trabalho estamos interessados num problema mais abrangente que consiste na **enumeração de todos os cliques máximos** (BRON e KERBOSCH, 1973) como no grafo (b) que contém quatro cliques máximos: observe que nenhum clique máximo em está contido em outro.

Por fim, o problema de encontrar um clique máximo é *NP*-completo, mas o problema de enumerá-los é *NP*-difícil. Para tal, pode-se usar estratégias heurísticas e/ou de força bruta, dependendo do tamanho do espaço de busca do problema específico.

Nesta Dissertação, o problema da enumeração de cliques máximos é utilizado no método de identificação de sinônimos (Seção 4.1.2). Seguindo a idéia de Unidades de Significado (VENANT, 2006)**,** cliques são enumerados a partir de um grafo formado por palavras similares conectadas: palavras pertencentes a um mesmo clique são consideradas sinônimas.

2.3 MODELO *NOISY-OR* DE REDES BAYESIANAS

Em Inteligência Artificial, abordagens que lidam com raciocínio podem ser divididas em duas grandes áreas que compreendem **Raciocínio Lógico** e **Raciocínio Probabilístico** (ou Estatístico). No contexto probabilístico, o emprego de Redes Bayesianas (JENSEN, 2001) constitui umas das principais técnicas de raciocínio, especialmente útil quando se quer diminuir/sintetizar o total de probabilidades em um sistema. Exemplos de aplicação incluem classificação automática, reconhecimento de voz, análise de texto em geral, processamento de imagem, suporte a decisão, descoberta de seqüências de proteínas, etc.

Também conhecidas como redes de crença, redes de opinião, redes causais e etc., as Redes Bayesianas associam Teoria das Probabilidades a Teoria dos Grafos para modelar a idéia de relacionamento probabilístico entre entidades. Tecnicamente, uma Rede Bayesiana é um modelo de árvore, isto é, um grafo acíclico direcionado (DAG) no qual os vértices representam variáveis ou entidades de qualquer natureza e os arcos representam relacionamentos entre as entidades. Os relacionamentos são condicionalmente independentes, isto é, o estado de uma entidade (causa) influencia o estado de outra entidade (conseqüência), contudo as causas são independentes umas das outras, garantindo que o grafo seja acíclico.

Apesar do poder de Raciocínio Lógico ser inegável, as Redes Bayesianas são consideradas melhores para tratar **raciocínio com incerteza**, situação na qual não se sabe tudo a respeito do domínio, ou seja, falta informação (Ignorância Teórica):

> *"A principal vantagem de raciocínio probabilístico sobre raciocínio lógico é o fato de que agentes podem tomar decisões racionais mesmo quando não existe informação suficiente para se provar que uma ação funcionará", (CHARNIAK, 1991).*

A Figura 2-5 exemplifica a combinação de evidencias usando Redes Bayesianas: se uma proposição representada por um nó *x* implica em uma proposição *y*, então uma aresta é colocada de *x* para *y* significando $P(y|x)$. O modelo em (a) representa a estrutura geral de **Independência Causal** em Redes Bayesianas (LAMMA e MELLO, 1999), uma árvore que expressa a idéia de que um conjunto de causas C_n independentes entre si influenciam um efeito comum *E* através de causas intermediárias I_m utilizando uma função determinística *Operador*. Em (b) a rede foi modelada especificamente para recuperar documentos em uma máquina de busca: d_j são os documentos, *q* é a consulta e k_i são os termos do vocabulário da coleção. Neste exemplo, os termos são evidências (causas) que, quando combinadas, representam documentos e consulta (conseqüências ou efeitos).

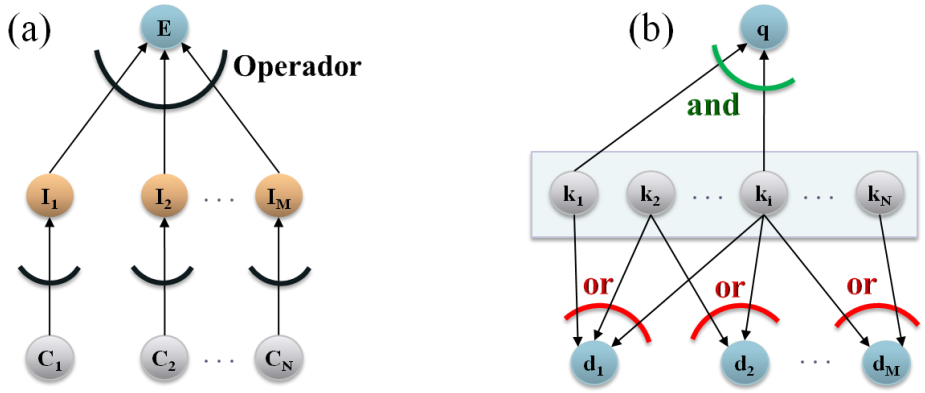


Figura 2-5: Redes Bayesianas para (a) Independência Causal e (b) Recuperação de Informação

Como em (VALE, *at al.*, 2001), o exemplo acima demonstra a utilização de dois operadores diferentes: um conjuntivo (AND) combinando evidências para a consulta, e um disjuntivo (OR) combinando evidências para os documentos. Estes operadores representam dois modelos típicos para construir Redes Bayesianas: o modelo *Noisy-Or* e o modelo *Noisy-And*. Excepcionalmente, existem ainda os modelos *Noisy-Max* e *Noisy-Min*.

Nesta Dissertação, usamos apenas o modelo *Noisy-Or*, um dos modelos mais populares para construir Redes Bayesianas em diversas situações práticas e que tem a mesma estrutura matemática de funções de combinação (LAMMA e MELLO, 1999). O *Noisy-Or* é uma adaptação probabilística equivalente ao OU da Lógica, e por isso representa uma disjunção de preposições. Seja E um efeito que depende das causas $C_1 \ldots C_n$, o *Noisy-Or* é definido genericamente pelo produtório do complemento das causas, veja:

Equação 1: $$\text{NoisyOr}(E \mid C_1 \ldots C_N) = \neg \prod_{1}^{N} P(\neg C_i) = 1 - \prod_{1}^{N} P(1 - C_i)$$

Por exemplo, se o efeito *E* dependesse das probabilidades causais *a*, *b* e *c*, a fórmula acima seria assim instanciada:

$$NoisyOr(E|a,b,c) = 1 - (1-a) * (1-b) * (1-c)$$

Na Seção 4.2, aplicamos o *Noisy-Or* nas taxonomias das ontologias para fazer propagação *bottom-up* da similaridade computada entre conceitos, equivalente a dizer que sintetizamos probabilidades em árvores de generalização. Também utilizamos o *Noisy-Or* para combinar resultados de diferentes classificadores supervisionados.

2.4 ENTROPIA

Entropia é uma métrica eficaz utilizada para medir uniformidade de distribuições de probabilidade. A teoria da entropia é bastante intuitiva, facilitada pela abstração de que é possível **medir a informação em um conjunto de resultados**, independente da semântica especifica destes resultados: quanto mais informação mais uniforme.

O termo entropia vem do grego ***εντροπία,*** mais precisamente *em + trope* que significa "em transformação": quanto maior a entropia maior a probabilidade de que um dado evento ocorra. Originalmente empregado na Termodinâmica (Física), foi agregado à moderna **Teoria da Informação** como **Entropia Informacional** desde a publicação de (SHANNON, 1948), na tentativa de modelar matematicamente a comunicação como um problema estatístico para benefício do trabalho de engenheiros elétricos.

O princípio chave da utilização de entropia é que, tendo-se à disposição várias amostras de probabilidade, deve-se certamente preferir a amostra com mais informação: este é o conceito da **Máxima Entropia**, um princípio particularmente importante para combinação de evidências em sistemas de Recuperação de Informação. Nesta área, um dos mais interessantes exemplos de aplicação é (NIGAMY, *at al.*, 1999) que desenvolveu o precursor dos classificadores de texto baseados na Máxima Entropia.

A Entropia Informacional é usada para medir a quantidade de informação contida em um conjunto de símbolos que formam uma mensagem. No entanto, a teoria de entropia não considera a semântica dos dados, mas sim o nível de caos ou desequilíbrio entre eles: quanto maior o caos, menor a quantidade de informação. A entropia informacional *H* de um conjunto de símbolos de tamanho *n* é definida como segue:

Equação 2: $H = -\sum_{1}^{n} p_i \times \log_2(p_i)$

- p_i é a probabilidade de ocorrência de cada i-ésimo símbolo dentro do conjunto de *n* símbolos, logo temos que $\sum_{1}^{n} p_i = 1.0$;
- $\log_2(p_i)$ é o grau de caoticidade;
- $máx(H) = \log_2(n)$, logo podemos normalizar *H*: $0.0 \leq \frac{H}{\log_2(n)} \leq 1.0$

Como se pode ver, o valor máximo de entropia ocorre quando a distribuição das probabilidades p_i é totalmente uniforme.

A definição de entropia é utilizada em várias partes deste trabalho. É associada ao *Noisy-Or* em duas situações (Seção 4.2): para (a) combinar classificadores e para (b) propagação *bottom-up* de similaridade nas taxonomias das ontologias. Além disso, é aplicada também para medir a informação em grupos de sinônimos (*synsets*) (Seção 4.1.2), o que depende da quantidade de palavras polissêmicas contidas no grupo: quanto mais polissemia, menos informação.

2.5 CLASSIFICAÇÃO SUPERVISIONADA

Classificação Supervisionada trata de um subconjunto de problemas de **Aprendizado de Máquina** vastamente utilizados para Recuperação de Informação (R.I.). Encontra grande aplicação na Web, como em filtros de spam e em diretórios de máquinas de busca, onde a divisão de documentos é feita por categoria, classificados taxonomicamente por assunto

(CHAKRABARTI, 2002). Bastante pesquisa sobre classificação supervisionada tem sido feita no decorrer das últimas décadas, sugerindo a adaptação das técnicas existentes para o domínio ontológico. Apesar de pouco explorado neste domínio, o uso de Aprendizado de Máquina em ontologias não é novidade, como em (ANHAI, *at al.*, 2004) que adaptou o classificador *Naive Bayes* para mapeamento entre ontologias e em (HUANG, *at al.*, 2008) que combinou o classificador SVM junto a algoritmos genéticos numa solução semi-automática para instanciar a *Gene Ontology* (GO).

Denomina-se **classificador automático** todo algoritmo capaz de atribuir categorias a instâncias sem categoria conhecida (objetos de teste), a partir de um conjunto finito de categorias conhecidas. A classificação é **supervisionada** quando utiliza uma base de conhecimento cujas categorias são previamente instanciadas com uma amostra de exemplos (objetos de treinamento). Quando isso não ocorre, a classificação é **não-supervisionada**. Em estatística, os classificadores são conhecidos como **funções discriminantes**.

Uteis por substituir o trabalho manual, as máquinas de classificação têm a difícil tarefa de identificar, grosso modo, o "tipo" ou "qualidade" de cada objeto submetido a elas, como imagens, sons, páginas *html*, e-*mails* e outros tipos de recursos, mídia e dados, utilizando-se de passos matemáticos, sintáticos e semânticos. Quando feita por especialistas humanos, a classificação é considerada mais segura. No entanto, mesmo os especialistas podem ter dúvidas e errar ao classificar um objeto, por isso não se deve esperar que mesmo a melhor máquina de classificação acerte sempre 100% da classificação dos objetos que recebe.

2.5.1 Conjuntos de Teste e de Treino

Independente de como um classificador supervisionado for implementado, ele sempre trabalha com dois conjuntos: o *conjunto de teste* e o *conjunto de treino*. Veja:

Figura 2-6: Entrada e Saída de um Classificador.

O *conjunto de teste* é um grupo de instâncias cuja classificação ainda é desconhecida, ou pelo menos, assume-se que é desconhecida no momento em que eles são submetidos ao classificador. A expressão "*testar a instância*" é o mesmo que afirmar que tal instância vai ser submetida à classificação.

O conjunto de treino é um grupo de instâncias cuja classificação é conhecida, isto é, instâncias pré-classificadas que servirão de exemplo para o classificador, formando sua base de conhecimento. Assume-se que a pré-classificação do conjunto de treino é confiável e indiscutível. Normalmente, para assegurar a confiabilidade do conjunto de treino, a pré-classificação é feita por especialistas humanos. Porém, em casos mais específicos, é aceito que a pré-classificação seja feita por procedimento automático considerado confiável e seguro.

2.5.2 O Classificador NB-*Shrinkage*

Dos diferentes algoritmos de classificação utilizados para mapeamento no L-*Match*, o *Naive Bayes with Shrinkage*, ou simplesmente NB-*Shrinkage*, foi evidentemente melhor. Por isso, esta Seção foi criada para destacar os conceitos gerais associados a este classificador e abordar especificamente a técnica de *shrinkage*.

A denominação *Naive Bayes* remete a uma família de classificadores estatísticos bem fundamentados em Probabilidade Bayesiana. Neste caso, "*naive*" significa que estes classificadores utilizam o Teorema de Bayes simplificado, assim definido em Estatística:

Equação 3: $$P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)}$$

A notação $P(A|B)$ equivale ao *Modus Pones* de lógicas discretas, como Lógica Proposicional, e significa "*o que se pode inferir sobre A, dado que tudo que se sabe é B*?". Cada parte tem nome próprio: $P(A|B)$ é a probabilidade posterior, $P(A)$ é a probabilidade anterior de A, $P(B|A)$ é a probabilidade condicional de B dado A, e $P(B)$ é utilizada como constante de normalização. No caso de classificação supervisionada, seja uma classe C descrita por um conjunto de características $F_1 \ldots F_n$, o Teorema de Bayes é reescrito:

Equação 4: $$P(C \mid F_1 \ldots F_n) = \frac{P(C)P(F_1 \ldots F_n \mid C)}{P(F_1 \ldots F_n)}$$

Por sua simplicidade e eficiência, estes classificadores são bastante utilizados em diferentes problemas (notoriamente em categorização de texto), também porque costumam

ser bastante eficientes, pois são treinados em tempo linear. Por esta razão, são considerados classificadores de bom custo benefício entre eficácia (qualidade) e eficiência (velocidade).

A forma mais simples do *Naive Bayes* não é tão robusta e sofre com alguns problemas. Muitas melhorias foram propostas, culminando no surgimento de algoritmos que associam outros princípios ao *Naive Bayes* original, como *smoothing* (JUAN e NEY, 2002), maximização de entropia (NIGAMY, *at al.*, 1999) e *shrinkage* (MCCALLUM, *at al.*, 1998).

Na Estatística, s*hrinkage* é uma técnica conhecida e eficaz usada para melhorar estimativas de probabilidades e pode ser associada paralelamente com outras técnicas de *smoothing*. Intimamente relacionada à inferência Bayesiana, a técnica de *shrinkage* foi adaptada para classificação, dando origem ao classificador NB-*Shrinkage* que incrementa significativamente a eficácia do *Naive Bayes*, como mostram experimentos de classificação de texto em (MCCALLUM, *at al.*, 1998) nos quais a acurácia do *Naive Bayes* foi aumentada em até 29% ao utilizar *shrinkage*. Por exemplo, considerando a média μ de uma distribuição Gaussiana e a variância em torno da média, basicamente o que ocorre é o seguinte:

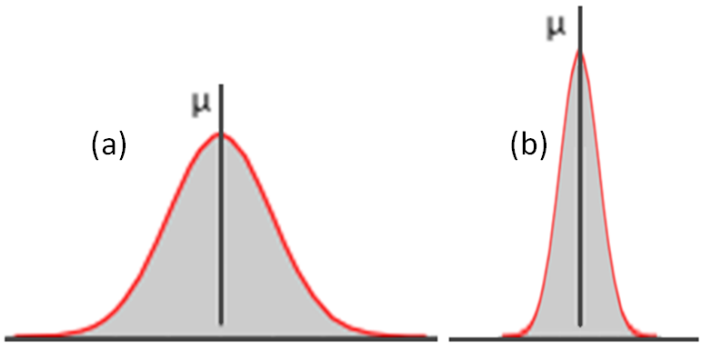


Figura 2-7: Distribuição (a) sem *shrinkage* e (b) com *shrinkage*

Com a utilização de *shrinkage*, os pontos da distribuição ficaram mais próximos da média, diminuindo a variância. Essa é a idéia chave para melhorar as estimativas.

Shrinkage é uma técnica especialmente utilizada em problemas de classificação hierárquica (MCCALLUM, *at al.*, 1998): explora os caminhos de uma hierarquia taxonômica tentando melhorar as estimativas dos descendentes em função de dados dos ancestrais através de combinação linear (CHAKRABARTI, 2002). Talvez isso explique o desempenho do NB-*Shinkage* ao ser aplicado para mapear conceitos de ontologias, naturalmente organizados em taxonomia, o que ficou claro nos experimentos desta Dissertação (Capítulo 5). Apesar da nossa abordagem não ter implementado exatamente uma classificação hierárquica e sim uma propagação hierárquica da classificação, ao que se apresenta, o NB-*Shinkage* beneficiou-se da hierarquia assim mesmo. Disso, pode-se concluir que s*hrinkage* se comporta bem ao ser aplicado em ontologias, e pode ser explorado mais profundamente em trabalhos futuros.

2.6 PRECISÃO, REVOCAÇÃO E MEDIDA-F

Esta seção trata das métricas de avaliação utilizadas neste trabalho para mensuração da eficácia ou qualidade de resposta do sistema de mapeamento. Um mapeamento $M = (Ci, Cj, Qij)$ entre conceitos Ci e Cj é classificado nos conjuntos:

Figura 2-8: Conjuntos Ideal, Computado e Acertos

O **conjunto Ideal** contém mapeamentos estabelecidos por especialistas humanos, por isso mapeamentos deste conjunto são considerados confiáveis e assumidos como verdade. O **conjunto Computado** é equivalente à saída do mapeador a ser avaliado. Da interseção entre estes conjuntos, surge um terceiro: o **conjunto Acertos,** contendo mapeamentos corretamente computados. Uma vez que mapeamentos sugeridos automaticamente podem não coincidir com a realidade, a idéia é confrontar os conjuntos Ideal e Computado e esperar que a interseção entre eles seja a maior possível: quanto maior o conjunto Acertos, melhor a qualidade da resposta do sistema.

Para quantificar a relevância de um conjunto Computado resultante de experimentos de Recuperação de Informação (R.I.), mensura-se a **corretude** e a **completude** do sistema. Para isso, utilizam-se três métricas principais variando dentro da escala [0,1]: **precisão**, **revocação** e **medida-F**, definidas a seguir:

Equação 5: $$precisão = \frac{|ideal \cap computado|}{|computado|}$$ **Equação 6:** $$revocação = \frac{|ideal \cap computado|}{|ideal|}$$

Equação 7: $$medidaF = 2 \times \frac{precisão \times revocação}{precisão + revocação}$$

A precisão é a métrica de corretude. Serve para medir a quantidade de acertos computados em relação ao total de respostas. O desejável é que a quantidade de respostas ideais supere a quantidade de respostas erradas, que funcionam como "lixo" afetando negativamente a precisão. No melhor caso da precisão, o lixo é inexistente.

A revocação é a métrica de completude ou abrangência da resposta. Serve para medir a quantidade de acertos computados em relação ao total de acertos possíveis: mesmo que venha lixo na resposta, deseja-se que todas as respostas ideais sejam encontradas, o que seria o melhor caso da revocação.

Sendo assim, para obter valores máximos de precisão e revocação, os resultados computados devem conter **apenas mapeamentos corretos** e **todos os mapeamentos corretos**, quando então os conjuntos Ideal, Computado e Acerto serão equivalentes. Infelizmente, isso dificilmente acontece, pois é comum encontrar valores de precisão e revocação desequilibrados: precisão alta e revocação baixa (ou vice-versa) não são sinais de bons resultados. Para avaliar o equilíbrio entre as duas métricas, costuma-se combiná-las numa terceira métrica: a medida-F, que é a média harmônica entre precisão e revocação.

Quando temos valores parciais de precisão, revocação e medida-F computados para vários objetos, mas queremos um valor global, precisamos pensar na **macro-avaliação** e na **micro-avaliação** (CHAKRABARTI, 2002). Seja um conjunto $X = \{x_1, \ldots, x_n\}$ composto por objetos a avaliar, a macro-precisão e a macro-revocação são as médias das avaliações parciais, sendo que a macro-medida-F é a média harmônica destas duas métricas:

Equação 8: $$macroPrecisão(X) = \frac{precisão(x_1) + \ldots + precisão(x_n)}{n}$$

Equação 9: $$macroRevocação(X) = \frac{revocação(x_1) + \ldots + revocação(x_n)}{n}$$

Por outro lado, a micro-precisão e a micro-revocação são calculadas normalmente em função do somatório de acertos, de respostas ideais e de respostas computadas para cada objeto individual, sendo a micro-medida-F a média harmônica destas duas métricas.

Equação 10: $$microPrecisão(X) = \frac{|Acerto(x_1)| + ... + |Acerto(x_n)|}{|Computado(x_1)| + ... + |Computado(x_n)|}$$

Equação 11: $$macroPrecisão(X) = \frac{|Acerto(x_1)| + ... + |Acerto(x_n)|}{|Ideal(x_1)| + ... + |Ideal(x_n)|}$$

No caso do mapeamento semântico aqui desenvolvido, utilizamos a macro-avaliação para obter a avaliação global dos mapeamentos produzidos pelo L-*Match* (Seção 5.3), em função das avaliações parciais de cada tipo de mapeamento, uma vez que existe um conjunto Ideal, um Computado e um Acerto para cada relação *R* utilizada como axioma ponte:

- ***Ideal*** é constituído pelos mapeamentos idealmente qualificados por *R*;
- ***Computado*** é constituído pelos mapeamentos automaticamente qualificados por *R*;

Espera-se que o mapeador acerte a relação *R* para cada par de conceitos, pois quando o mapeador erra, ele diminui a precisão da relação computada (pois ele insere um mapeamento errado) e diminui a revocação da relação ideal (pois ele não foi capaz de encontrá-la).

Capítulo 3

TRABALHOS RELACIONADOS

Neste capítulo, serão revisadas as idéias principais das referências que mais contribuíram com esta Dissertação, focando em trabalhos de Integração de Informação voltados a formalismos, aprendizado de máquina e mapeamento entre ontologias.

O passo inicial é organizar e formalizar os conceitos sobre mapeamento. Nesta área, (RAHM e BERNSTEIN, 2001) propõe uma taxonomia para classificar a maioria das abordagens sobre mapeamento de esquema até então desenvolvidas, numa primeira tentativa de padronizá-las. Baseado nesta taxonomia, (EUZENAT e VALTCHEV, 2003) propõe classificar estas abordagens em função da estratégia e da fonte de evidência utilizadas para comparar conceitos: comparação terminológica, comparação da estrutura interna, comparação da estrutura externa, comparação extensional e comparação semântica. Esta classificação é mais adequada para o estado da arte e foi adotada nesta Dissertação e em vários outros trabalhos como (EUZENAT e VALTCHEV, 2004) e (FüRST e TRICHET, 2005).

Em (BOUQUET, *at al.*, 2003) e (MAGNINI, *at al.*, 2003) uma nova abordagem sobre mapeamento começa a ser delineada em função do que chamam de *coordenação semântica*. Observando a falta de infra-estrutura para especificação de mapeamentos semânticos, é apresentada a linguagem *Context* OWL (C-OWL) que estende a sintaxe e a semântica da

OWL convencional para coibir certas operações e permitir a especificação de mapeamentos semânticos entre conceitos de ontologias distintas. Mais tarde, (STUCKENSCHMIDT, *at al.*, 2004) testou C-OWL durante a integração de ontologias médicas, concluindo que C-OWL é um formalismo adequado para o suporte de mapeamentos complexos.

Em C-OWL, os mapeamentos são expressos pelas relações de equivalência (≡)**,** mais geral (⊐)**,** menos geral (⊏)**,** sobreposto (⊓) e diferença (≢). Agregado à esta linguagem, é proposto um algoritmo para computar estas relações utilizando o *WordNet* e raciocinadores baseados na Satisfatibilidade (SAT): os conceitos são contextualizados pela conjunção de seus ancestrais, formando cláusulas lógicas que são validadas pelo raciocinador cuja base de conhecimento é a taxonomia do *WordNet*. Apesar desta abordagem se limitar às taxonomias de ontologias e do *WordNet*, ignorando regras e restrições (o que é comum no estado-da-arte), ela representa um grande avanço porque, pela primeira vez, permite que os mapeamentos sejam semânticos e exemplificam a utilização de raciocínio lógico.

A partir destas idéias, um grande corpo de trabalho vem sendo desenvolvido, sobrepujando várias outras abordagens que praticam apenas mapeamento sintático. A primeira experimentação envolvendo raciocinadores SAT culminou no desenvolvimento do mapeador semântico chamado *Context Match* ou Ctx-*Match* (BOUQUET, *at al.*, 2003) (MAGNINI, *at al.*, 2004), cujas avaliações incluem apenas as relações de equivalência**,** mais geral e menos geral. Apesar da abordagem elegante, os experimentos com o Ctx-*Match* foram relativamente precisos ao custo de revocação muito baixa, deixando a desejar quanto ao equilíbrio entre corretude *versus* completude. A precisão dos mapeamentos de mais e menos geral chegou a 80% e 90% com revocação de cerca de 50%. O pior caso do Ctx-*Match* é para a relação de equivalência: precisões baixas variando de 27% a 78%, com revocação

baixíssima variando de 4% a 13%. Diferente do Ctx-*Match*, o L-*Match* não se mostrou desequilibrado e nem teve mau desempenho ao computar relações de equivalência.

Apesar dos resultados ruins do Ctx-*Match*, a sua abordagem era promissora. Por isso, (GIUNCHIGLIA, *at al.*, 2005) deu andamento às idéias do Ctx-*Match* num novo mapeador semântico batizado de *Semantic Match* ou S-*Match*, o qual obteve significativa qualidade de resposta, mesmo quando comparado com outros sistemas, no caso Cupid (MADHAVAN, *at al.*, 2001), COMA (DO e RAHM, 2002) e Rondo (MELNIK, *at al.*, 2003). Isso ocorreu em grande parte devido a várias otimizações e combinações com outras estratégias de mapeamento. Só então a abordagem ganhou reconhecimento, sendo considerada uma das melhores. Apesar das metodologias de comparação entre conceitos do S-*Match* e do L-*Match* serem diferentes, o S-*Match* é uma das principais referências desta Dissertação, pois o mapeamento produzido pelos dois sistemas é semântico e qualificado pelas mesmas cinco relações (axiomas ponte).

Para investigar evidências apenas no nível de esquema (sem investigar instâncias), o S-*Match* torna-se dependente do *WordNet* que funciona globalmente impondo mapeamentos locais. O processo de mapeamento do S-*Match* pode ser dividido em duas grandes etapas: primeiramente, utiliza o *WordNet* para gerar mapeamentos iniciais que são ajustados na segunda etapa por raciocinadores SAT. A SAT é um problema NP-Completo, portanto as soluções são heurísticas e podem ser demoradas, o que prejudica o desempenho do S-*Match*. Mesmo assim, a qualidade das conclusões deste tipo de raciocínio é interessante, podendo futuramente ser agregado ao L-*Match* para fins de experimentação, criando um híbrido entre o L-*Match* e o S-*Match* onde não existe a imposição do *WordNet*.

Ontologias ainda constituem tecnologia recente, cujo uso é pouco disseminado. Uma das conseqüências disto é a atual escassez de instâncias em ontologias, o que parece ser

apenas uma situação momentânea. Mesmo assim, trabalhos como (BOUQUET, *at al.*, 2003) criticam a investigação de instâncias para mapeamento alegando esta escassez. Contudo, é mais adequado dizer que não se deve depender de apenas uma fonte de evidência. O próprio trabalho de (BOUQUET, *at al.*, 2003) tem dependências prejudiciais por ser fortemente atrelado a evidências extraídas do *WordNet*, já que não apresenta alternativas ao dicionário. O *WordNet* está disponível apenas na língua inglesa, é um conhecimento estático, incompleto e, dependendo de como é utilizado, impõe uma única visão de mundo nos mapeamentos locais, o que o torna pragmaticamente indesejável especialmente pela comunidade de Recuperação de Informação (R.I.).

Por sua vez, (MAGNINI, *at al.*, 2004) afirma que algoritmos de classificação de instâncias são inviáveis quando a natureza das instâncias não é textual, o que não é verdade. Classificação automática encontra extensa aplicação nas áreas de Recuperação de Informação em Imagem, Processamento Digital de Imagem e Visão Computacional, a exemplo de problemas relacionados a *Optical Character Recognition* (OCR), filtros de pornografia, processamento de imagens de satélite e radar, etc. Alguns trabalhos são (SHIBA, *at al.*, 2005) que avalia cinco classificadores aplicados sobre imagens de radar, (BRAGA, *at al.*, 2006) e (RIBEIRO, *at al.*, 2005) que comparam classificadores supervisionados e não-supervisionados em imagens sobre o uso de solo e à cobertura terrestre de solo, e (KUMAR e MILLER, 2006) que classifica objetos numa imagem usando informação de cor e textura. Há também classificação de recursos da Web fundamentada na investigação de *hyperlinks* associados ou não com texto e imagem, a exemplo de (CALADO, *at al.*, 2003).

Iniciativas como estas, que pregam não investigar a instanciação de conceitos, são precipitadas e negativas. Conseqüentemente, a literatura sobre mapeamento entre ontologias é pobre quanto à utilização de técnicas de Aprendizado de Máquina, como Classificação

Supervisionada, causando uma demanda de pesquisa não atendida. Chega a ser paradoxal, pois muitos dos trabalhos que dizem integrar ontologias ignorando as instâncias, como (GIUNCHIGLIA, *at al.*, 2005), experimentam suas abordagens apenas sobre bancos relacionais, catálogos e diretórios da Web, onde instâncias são fartamente disponíveis, não mostrando experimentos com ontologias de fato.

Um amplo corpo de ferramentas e trabalhos bem sucedidos (CHAKRABARTI, 2002)(CALADO, *at al.*, 2003)(YANG, *at al.*, 2002)(LIU, *at al.*, 2005), motivados por problemas da Web, tem trazido extensos benefícios a processos de Classificação Supervisionada, principalmente no que se refere a hipertexto e imagem, sugerindo fortemente a adaptação das técnicas existentes para problemas do domínio ontológico, a exemplo da anotação da *Gene Ontology* em (HUANG, *at al.*, 2008) e do mapeador GLUE (ANHAI, *at al.*, 2004).

Apesar de ser um mapeador sintático, o GLUE merece destaque em relação à nossa abordagem, uma vez utiliza Aprendizado de Máquina para computar mapeamentos. Ele é dividido em duas etapas: a primeira utiliza o classificador *Naive Bayes* para computar similaridade de *Jaccard* entre conceitos; a segunda etapa cria mapeamentos através de *relaxation labelling*, técnica utilizada no GLUE para otimizar e relaxar regras taxonômicas, entre outras. Os experimentos mostraram que a precisão do GLUE é alta, entre 70% a 90%, quando combina todas as evidências disponíveis.

Os resultados obtidos com o GLUE encorajam a utilização de Aprendizado de Máquina para mapear ontologias, porém sua abordagem é ainda bastante trivial e não explora diversas possibilidades. A eficácia do GLUE é muito dependente do classificador *Naive Bayes*, pois não foram experimentados outros algoritmos de classificação. Já a eficiência do GLUE é prejudicada, pois calcula o complemento de cada conceito a mapear, além do que o

complemento pode facilmente se tornar muito maior que o conceito original. Isso aumenta o risco de desbalanceamento entre classes, um dos piores problemas que podem ocorrer num processo de classificação supervisionada. Em seguida, o *Naive Bayes* é re-treinado e re-executado sobre cada conceito e seu complemento. Este esforço para fazer classificação binária (em duas classes) é custoso e desnecessário, sendo mais adequada e eficiente uma única etapa de classificação multiclasse (em várias classes), como o L-*Match* faz, dado que ontologias normalmente possuem várias classes/conceitos.

O GLUE não faz classificação hierárquica e nem propagação *bottom-up* da classificação e, portanto, não explora a taxonomia dos conceitos antes do *relaxation labelling*, prejudicando o mapeamento entre conceitos genéricos e abstratos. Além disso não qualifica os mapeamentos através de axiomas (GIUNCHIGLIA, *at al.*, 2005). A técnica de *relaxation labeling* é eficaz e eficiente mesmo quando o número de regras é grande, mas a forma como *relaxation labeling* converge para as soluções não é muito bem conhecida, podendo convergir pra máximos locais sem explicação. Isto impede um maior domínio sobre o algoritmo, dificultando entender por que alguns mapeamentos funcionam e outros não.

Em (ICHISE, *at al.*, 2001) e (ICHISE, *at al.*, 2003) foi proposto o mapeador HICAL que explora a sobreposição de instâncias entre duas taxonomias utilizando o método *k-statistics* (ao invés de *Jaccard* como no GLUE) para mensurar similaridade entre conceitos. Similar ao L-*Match*, o HICAL utiliza uma estratégia *top-down* na taxonomia e seus desenvolvedores previram a necessidade de alguma estratégia *bottom-up*. A classificação feita no HICAL é não-supervisionada, ignorando o conteúdo das instâncias (texto, imagens, propriedades, etc.), por isso o HICAL dependente fortemente que duas taxonomias compartilharem exatamente as mesmas instâncias e em grande quantidade, o que acontece raramente na prática.

O GLUE e o HICAL limitam-se a mapeamentos sintáticos de equivalência, que são mapeamentos 1-1 (um-para-um). Por exemplo, se a relação mais forte entre um par de conceitos é uma generalização, estes mapeadores não serão capazes de encontrá-la. As idéias preliminares do GLUE e do HICAL assemelham-se às do L-*Match*: classificar (GLUE) ou identificar (HICAL) instâncias de uma ontologia *A* nos conceitos de outra ontologia *B* e vice-versa, indicando um fluxo de informação em duas direções: $A \rightarrow B$ e $B \rightarrow A$. Esta percepção de direção é necessária para o mapeamento semântico, porém é perdida durante o cálculo de similaridade com *Jaccard* (GLUE) e *k-statistics* (HICAL), por isso GLUE e HICAL perdem informação ao calcular similaridade entre conceitos. Além disso, estes cálculos não aproveitam os valores de similaridade computados pelo algoritmo de classificação utilizado.

Outros trabalhos também utilizaram instâncias e Aprendizado de Máquina em problemas envolvendo ontologias. Na área de Aprendizado de Ontologia, (BUITELAAR, *at al.*, 2005) apresenta o estado da arte sobre aquisição automática de conhecimento na forma de conceitos, taxonomias, relações e regras, a exemplo de (SNOW, *at al.*, 2006) e (CIMIANO, *at al.*, 2004) que criam e evoluem taxonomias. Em (HUANG, *at al.*, 2008) utiliza-se instâncias de proteínas anotadas na *Gene Ontology* pra treinar o classificador SVM e um algoritmo genético. Como a *Gene Ontology* é grande e complexa, o objetivo é auxiliar especialistas que fazem anotações na ontologia, sugerindo automaticamente a localização subcelular de novas proteínas recentemente descobertas.

O L-*Match* beneficia-se das taxonomias das ontologias. Em (RESNIK, 1990) é mostrado como taxonomias podem ser usadas para resolver ambigüidades sintáticas e semânticas e em (SOROKINE, *at al.*, 2005) é apresentado um estudo teórico de como instâncias e propriedades são responsáveis por definir os limites de suas classes ou conceitos em uma hierarquia taxonômica.

Há ainda outras abordagens de Integração de Informação, como os sistemas *Anchor*-PROMPT (NOY e MUSEN, 2001), o *Chimerae* (MCGUINNESS, *at al.*, 2000), o FCA-*Merge* (STUMME e MAEDCHE, 2001), o IF-*Map* (KALFOGLOU e SCHORLEMMER, 2003) e o COMA++ (AUMUELLER, *at al.*, 2005). Há também o sistema OLA (EUZENAT, *at al.*, 2004) para mapear ontologias em OWL, mas que não exaure as possibilidades de inferência em OWL, como ocorre com o L-*Match*. Atualmente, existe tendência de adicionar etapas de raciocínio Lógico na integração de ontologias, como o ILIADS (UDREA, *at al.*, 2007) que combina inferência estatística e lógica para mesclar ontologias.

A tabela seguinte, montada seguindo critérios de (SHVAIKO e EUZENAT, 2005), resume alguns sistemas citados neste capítulo. Observe que o fato de um sistema investigar instâncias não significa que utilizou Aprendizado de Máquina, por isso os sistemas que associam instância a Aprendizado de Máquina são sinalizados com asterisco (*).

Tabela 3-1: Sistemas de Integração de Ontologias

SISTEMA		NÍVEL DE ELEMENTO		NÍVEL DE ESTRUTURA		INSTÂNCIA	ESQUEMA
		Sintaxe	**Conhecimento Externo**	**Sintaxe**	**Semântica**		
2003	**HICAL**	-	-	Classificação Não-Supervisionada, *k-Statistics*	-	√*	-
2004	**GLUE**	Baseado em Linguagem	-	*Classificador Naive Bayes*, Similaridade de *Jaccard*, *Relaxation Labeller*	-	√*	-
2004	**OLA**	Baseado em Linguagem, Técnicas de *String*	Dependente de Dicionário Global (WordNet)	*Iterative Fix-Point Computation*, *Neighborhood Matching*	-	√	√
2002 e 2005	**COMA e COMA++**	Baseado em Linguagem, Técnicas de *String*	Auxiliado por Dicionário	Estratégias Diversas	-	√	√
2004 e 2005	**Ctx-*Match* e S-*Match***	Baseado em Linguagem, Técnicas de *String*	Dependente de Dicionário Global (WordNet)	-	Propositional SAT Solvers	-	√
2008	**L-*Match***	Baseado em Linguagem, Cliques em Grafos, Entropia	Auxiliado por *Thesaurus* (WordNet)	Classificação Supervisionada Propagada *Bottom-Up*, diferentes classificadores	Sobreposição entre Conceitos, Dedução Top-Down	√*	-

Capítulo 4

L-MATCH: UTILIZANDO APRENDIZADO DE MÁQUINA PARA DESCOBRIR MAPEAMENTOS SEMÂNTICOS ENTRE CONCEITOS DE ONTOLOGIAS

A denominação L-*Match* vem do inglês *Learning Match* e foi dada ao sistema desenvolvido neste trabalho, cujo funcionamento é centrado em Classificação Supervisionada, subárea de Aprendizado de Máquina. Por estes meios, propomos uma abordagem simples que visa identificar relacionamentos até então desconhecidos entre conceitos de ontologias distintas. No Mapeamento Semântico, o problema maior não é descobrir **quais** conceitos se relacionam, mas sim **como** eles se relacionam. Podemos responder esta questão prevendo automaticamente a regra mais apropriada para descrever o relacionamento através de métodos heurísticos. Nesta Dissertação, tratamos este problema de forma relativamente simples com a ajuda de classificadores quando há disponibilidade de instâncias nos conceitos das ontologias, pois esta abordagem nos permite definir algumas regras de mapeamento, como mostraremos ao longo do presente capítulo.

Num mundo onde os sistemas se tornam mais inteligentes, o interesse em mapeamentos automáticos que sejam semânticos é relevante, uma vez que o único mapeador realmente semântico desenvolvido até hoje é o S-*Match* (GIUNCHIGLIA, *at al.*, 2005), e

antes dele todos os mapeadores eram sintáticos. Por isso o L-*Match* surge como abordagem alternativa ao S-*Match*.

Utilizamos três estratégias de comparação (Seção 2.1.2): comparação terminológica, extensional e de estrutura externa. A arquitetura do L-*Match* possui três módulos: Extração, Similaridade e Mapeamento, cada uma correspondendo a uma estratégia de comparação. Além disso, a abordagem é iterativa, pois o sistema pode reutilizar os mapeamentos computados numa execução anterior. Veja:

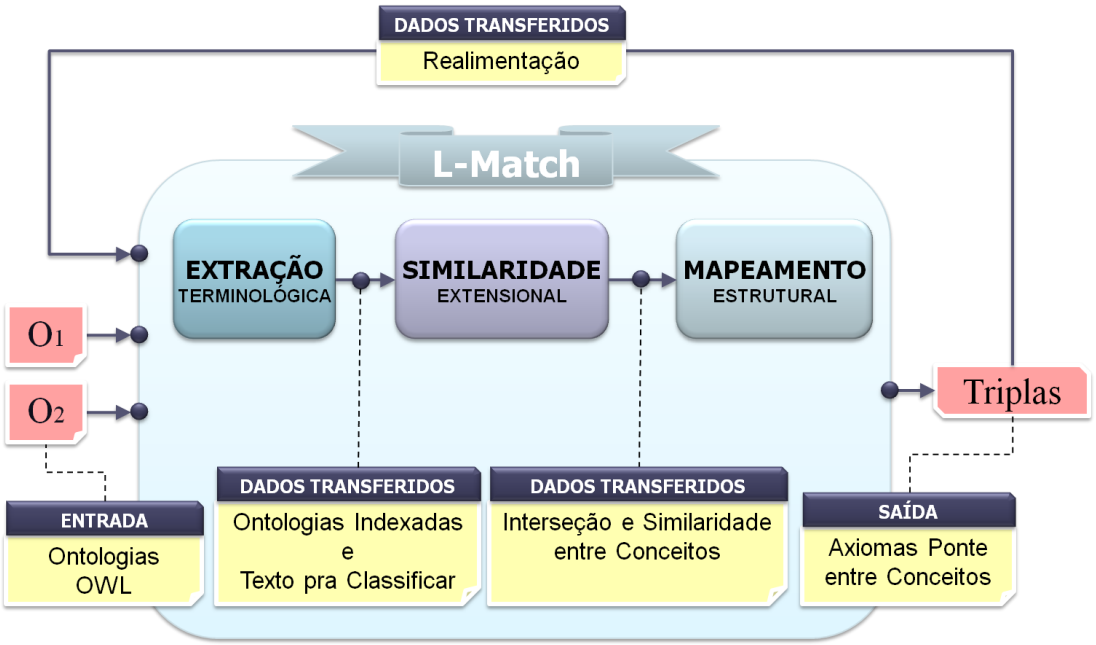


Figura 4-1: Arquitetura do L-*Match*

O L-Match recebe duas ontologias OWL, transforma suas instâncias em texto sobre o qual aplicada técnicas terminológicas; o modulo de similaridade classifica o texto de cada instância obtendo valores de similaridade e interseção entre conceitos; o módulo de mapeamento usa estes valores para sugerir axiomas pontes entre pares de conceitos (triplas). Nas seções seguintes, os três módulos acima serão mais bem detalhados.

4.1 EXTRAÇÃO

O Módulo de Extração é responsável por pré-processar as ontologias a mapear. Ontologias são formadas basicamente pelas mesmas entidades (conceitos, propriedades, instâncias e regras) e por isso nós as padronizamos num formato interno normal que independe do formato inicial (apenas OWL por enquanto), facilitando o mapeamento. As entidades são indexadas porque precisarão ser tabeladas nos módulos seguintes, mas a principal funcionalidade deste módulo é gerar texto para classificar.

4.1.1 Geração de Texto para Classificar

Instâncias em OWL (como em outros formatos de ontologia) não são documentos de texto e sim estruturas em XML/RDF. Por isso, primeiramente são renderizadas num conteúdo textual que é submetido aos classificadores de texto do módulo seguinte. Ao contrário do GLUE (DOAN, *at al.*, 2002), não consideramos conteúdo textual de páginas Web relacionadas a ontologias, consideramos apenas instâncias fornecidas com as próprias ontologias, o que nos dá certa autonomia ao gerar texto para classificar.

Ontologias fornecem informações precisas sobre suas instâncias: nome, conceitos, generalizações e propriedades com respectivos valor e *range*. Por exemplo, em uma de nossas ontologias, ***waterTemperature*** instância diretamente o conceito *TemperatureOf* e indiretamente os seus super-conceitos (*EcologicalData*, *Quantity* e *TemperatureQuantity*), relacionando-se com *celsius* (instância de *ScaleOfTemperature*) e *water* (instância de *EcologicalEntity*) por meio das propriedades *hasTemperatureScale* e *hasEntity*, respectivamente. Estas evidências ou informações sobre *waterTemperature* seriam renderizadas e discriminadas em quatro blocos dependendo de sua origem, veja:

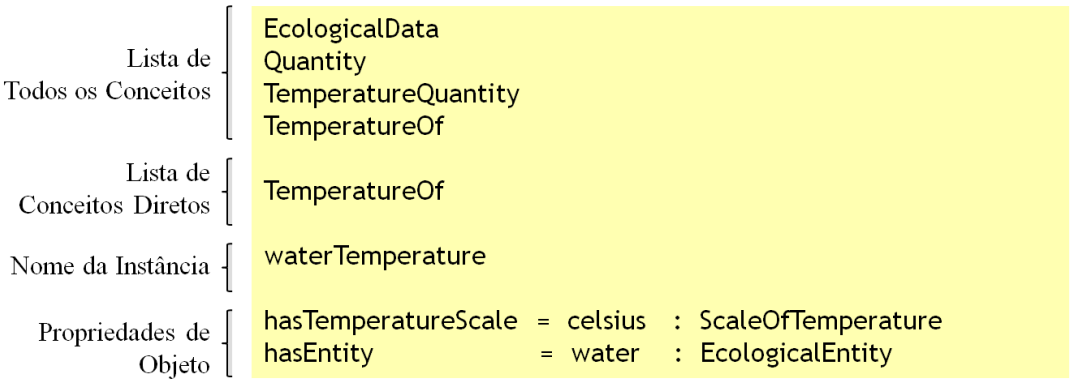


Figura 4-2: Texto básico gerado para a instância *WaterTemperature*

Os conceitos a que *waterTemperature* pertence são listados acima, seguidos pelo seu nome e por suas propriedades no formato ***nome = valor : range*.** Apesar de o texto gerado ser pequeno e preciso por ser composto por palavras chave, ele precisa ser normalizado para que haja melhor aproveitamento por parte dos classificadores: nomes são convertidos para caixa baixa, separados em *tokens* e perdem as *stopwords* (*to*, *has*, *of*, *the*, etc.), tarefas as quais são básicas em qualquer sistemas de processamento de texto. O resultado é o seguinte:

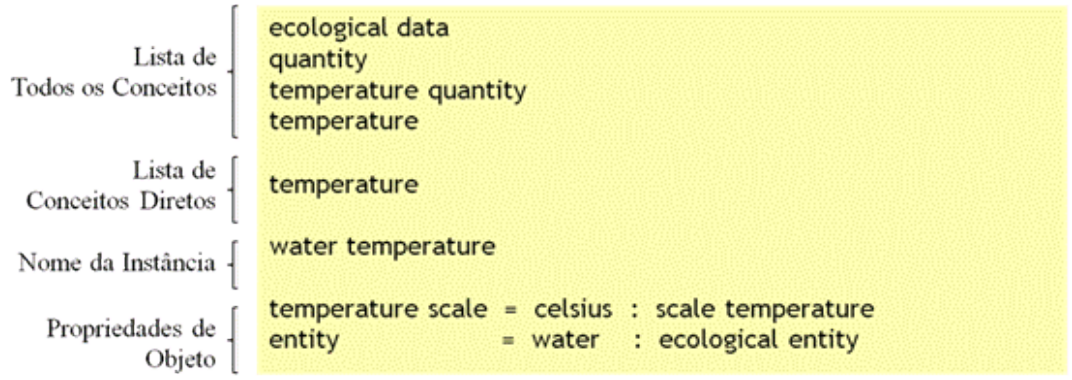


Figura 4-3: Texto tratado para a instância *WaterTemperature*

Como comentamos anteriormente, o L-Match é capaz de reaproveitar mapeamentos computados em execuções anteriores. Esta realimentação iterativa ocorre no momento em

que o texto é gerado: reutilizamos os mapeamentos recém descobertos de equivalência e de classe/subclasse para associar instâncias de uma das ontologia mapeadas com conceitos da outra ontologia e vice-versa. Levando em consideração estas novas associações, as instâncias são novamente renderizadas em texto para então re-treinar os classificadores do L-*Match*:

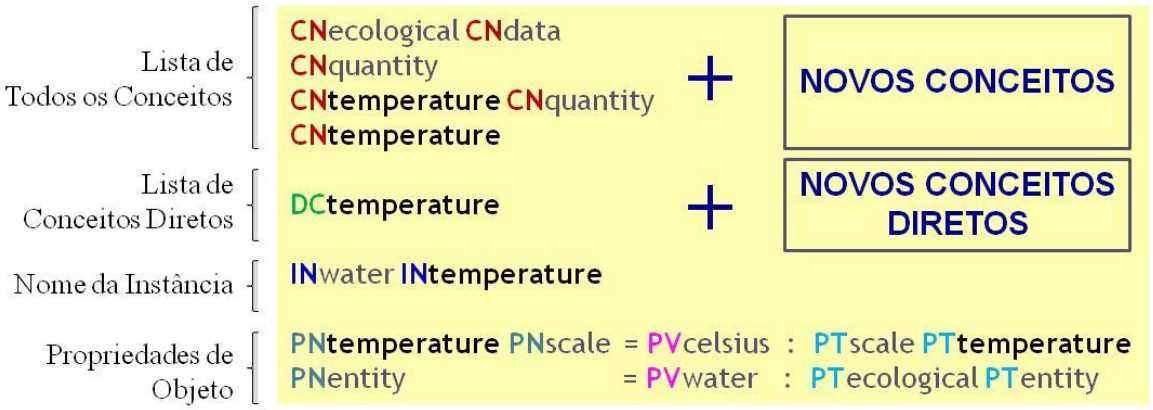


Figura 4-4: Reaproveitando mapeamentos para associar instâncias e conceitos de ontologias distintas

A vantagem desta abordagem se dá supondo que o L-*Match* é capaz de computar mapeamentos precisos, o que permitiria tornar ainda mais precisos e confiáveis os mapeamentos das iterações seguintes, o que de fato ocorre como podemos comprovar nos experimentos do próximo capítulo.

4.1.2 Qualidade do Texto

Deve-se lembrar que o resultado da extração alimentará classificadores de texto: mesmo o melhor algoritmo de classificação não funcionará satisfatoriamente se receber dados de má qualidade, o que torna o Módulo de Extração muito importante. Esta subseção descreve as estratégias de qualidade consideradas durante a extração: **escopos textuais** juntamente com **identificação** e **desambiguação de sinônimos**.

4.1.2.1 Prefixos de Escopos Textuais

Observe como a palavra *temperature* ocorre em diferentes partes do texto gerado para a instância *waterTemperature*:

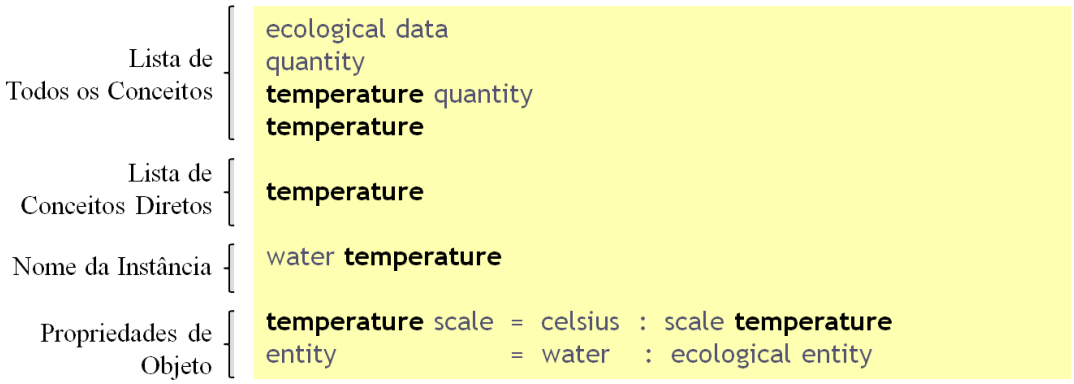


Figura 4-5: Palavra repetindo-se em diferentes partes do texto prejudica a qualidade

Cada bloco de texto tem significado diferente dependendo da origem do texto, mas esta diferença não está explícita uma vez que palavras iguais terão o mesmo efeito independente do bloco em que apareçam, causando um efeito negativo que confundirá os classificadores de texto. Por exemplo, a palavra *temperature* ocorrendo como conceito não está sendo diferenciada de *temperature* ocorrendo como *range* de propriedade: a instância *WaterTemperature* pertence aos conceitos *TemperatureQuantity* e *TemperatureOf*, mas não a *ScaleOfTemperature*.

Como classificadores de texto comparam *strings*, adicionamos diferentes prefixos às palavras originais dependendo do bloco em que aparecem, gerando novas palavras que discriminam **escopos textuais**, mesmo que originalmente fossem palavras idênticas. No exemplo a seguir, a palavra *temperature* não é mais a mesma palavra morfologicamente: CNtemperature, DCtemperature, INtemperature, PNtemperature e PTtemperature. Veja:

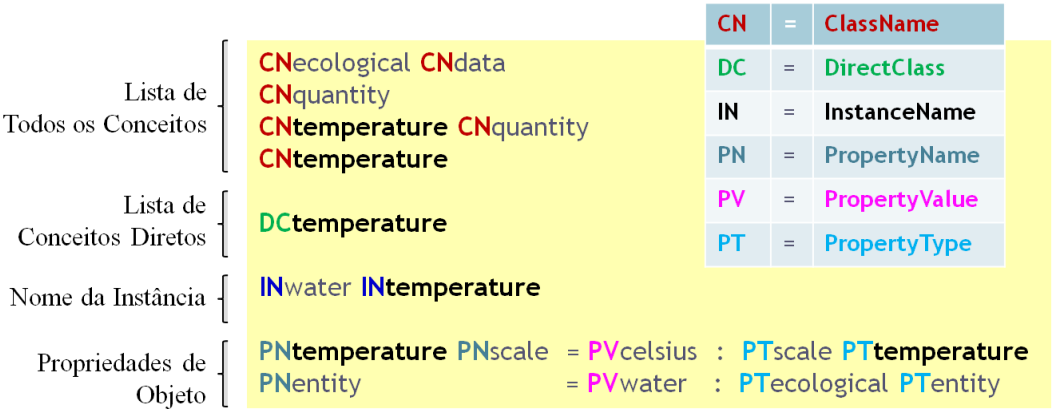


Figura 4-6: Adição de prefixos de escopo ao texto

4.1.2.2 Identificação e Desambiguação de Sinônimos

Identificar, desambiguar e agrupar palavras com significado similar num contexto ajuda a tratar o problema de rotular conceitos equivalentes com palavras diferentes. A técnica terminológica aplicada ameniza diferenças entre vocabulários das ontologias a mapear na medida em que seus domínios são correlatos, estabelecendo um terceiro vocabulário mais homogêneo que facilite a ação de classificadores de texto. As palavras são **reduzidas** morfologicamente através de *stemming* e semanticamente através de *synsets* (grupos de sinônimos) para uma **forma única**. O algoritmo de *stemming* é o *Porter Stemmer*[5] (PORTER, 1997) e o serviço de *thesaurus* é o *WordNet* (FELLBAUM, 1998). Exemplo de reduções:

1. $\boldsymbol{Base} \leftarrow \{ Base, Basic, Primary, Fundamenal \}$
2. $\boldsymbol{Dimens} \leftarrow \{Dimension, Dimensions\}$
3. $\boldsymbol{BaseDimens} \leftarrow \{ PrimaryDimension, FundamentalDimensions \}$

[5] http://tartarus.org/~martin/PorterStemmer/

No exemplo acima, a palavra ***base*** foi escolhida aleatoriamente para representar as palavras ***base***, ***basic***, ***primary*** e ***fundamental***, que foram consideradas sinônimas por nossa abordagem e, portanto, serão reduzidas para ***base***. Em seguida, as palavras ***dimension*** e ***dimensions*** foram agrupadas em torno do radical ***dimens***, o qual foi identificado pelo algoritmo de *stemming*. Palavras compostas também podem ter representantes únicos se todas as suas partes forem redutíveis a um mesmo representante: se nos deparássemos com palavras como ***PrimaryDimension*** e ***FundamentalDimensions***, poderíamos reduzi-las para ***BaseDimens***. Esta é uma observação muito simples, mas torna mais abrangente o método de amenizar diferenças entre vocabulários distintos.

Contudo, o *WordNet* nem sempre agrupa palavras num mesmo *synset* da mesma forma que um especialista agruparia em seu contexto específico. No último exemplo, as palavras *Primary* e *Fundamental* não são encontradas num mesmo *synset* do *WordNet*, mas no domínio de conhecimento do exemplo estas palavras são interpretadas como sinônimos. A alternativa é explorar os *links* de similaridade que o *WordNet* mantém entre *synsets* e esperar que *Primary* e *Fundamental* pertençam a *synsets* similares: de fato, podemos descobrir os caminhos *Fundamental→Important→Primary* e *vice-versa*. É importante que o caminho seja nos dois sentidos, o que torna a relação de similaridade mais forte, como se as palavras "concordassem" que são similares.

Ignorando os *synsets* originais do *WordNet*, geramos um novo *synset* que inclui *Fundamental* e *Primary*. Contudo, caminhar por *links* de similaridade pode levar muito longe e com isso aumentar o tamanho dos novos *synsets*. Reduzir um *synset* grande a uma única palavra pode criar generalizações, o que é um problema, pois a expectativa é que *synsets* contenham apenas palavras equivalentes. Veja um novo *synset* que se tornou genérico:

$$\boldsymbol{C} \leftarrow \{\, C, Carbon, Coulomb, Celsius, century \,\}$$

Para explorar *links* de similaridade do *WordNet* e gerar *synsets* menores e mais equivalentes, procuramos por **cliques máximos** num grafo de palavras similares. Intuitivamente, como cliques máximos são aglomerados muito fortes de palavras similares, entendemos que palavras "concordam" ser sinônimas quando formam um clique entre si. Essa é a idéia de **Unidade de Significado ou Clique** (VENANT, 2006) como a menor unidade semântica possível, aplicado em trabalhos sobre modelos geométricos de palavras organizadas em coordenadas semânticas, como é o caso do Atlas Semântico[6].

O vocabulário das ontologias é transformado num grafo $G = (V, E)$: palavras são vértices em V e duas palavras são conectadas se há caminho de ida e volta entre elas em função dos *links* de similaridade do *WordNet*. Para **enumerar todos os cliques máximos** (Seção 2.2) utilizamos um algoritmo exato como em (BRON e KERBOSCH, 1973) para resolver este problema *NP*-Difícil, pois o espaço de busca da nossa aplicação é naturalmente pequeno: *synsets* nunca são grandes, logo palavras similares formam subgrafos pequenos e isolados, sendo $|V|$ restrito ao tamanho do vocabulário das ontologias. Veja:

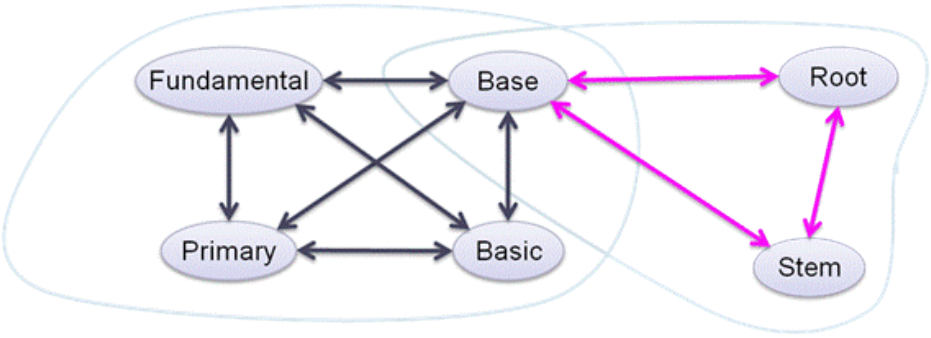


Figura 4-7: Unidades de Significado: os cliques serão os novos *synsets*

[6] http://dico.isc.cnrs.fr/en/index.html

Na Figura 4-7 temos um exemplo de **polissemia** ou **ambigüidade**: a palavra *Base* aparece em dois *synsets* distintos, logo possui dois significados. Isso contrasta severamente com a idéia de reduzir todas as palavras a uma única forma padrão, pois neste caso existem duas possibilidades de redução para *Base*. A solução é eliminar polissemia garantindo que cada palavra seja associada a um único *synset*.

O problema de descobrir o mais apropriado *synset*, significado ou sentido para uma palavra dentro de um contexto é chamado de **Desambiguação de Sentido de Palavra**. Quando a desambiguação é suave (*Soft Disambiguation*) as palavras podem continuar associadas a mais de um significado após desambiguação, mas não é isso que desejamos: queremos associar cada palavra com apenas um significado (no caso, o mais apropriado) e com isso garantir reduções únicas. Segundo (BUITELAAR e SACALEANU, 2001), num domínio restrito (como é o caso das ontologias mais comuns) muitos termos polissêmicos terão forte "preferência" por apenas um de seus possíveis significados ou *synsets*, pois este *synset* é boa evidência sobre a similaridade entre os domínios das ontologias a mapear. Por exemplo, nos nossos experimentos gostaríamos que os *synsets* da Figura 4-7: Unidades de Significado: os cliques serão os novos *synsets* fossem desambiguados como a seguir:

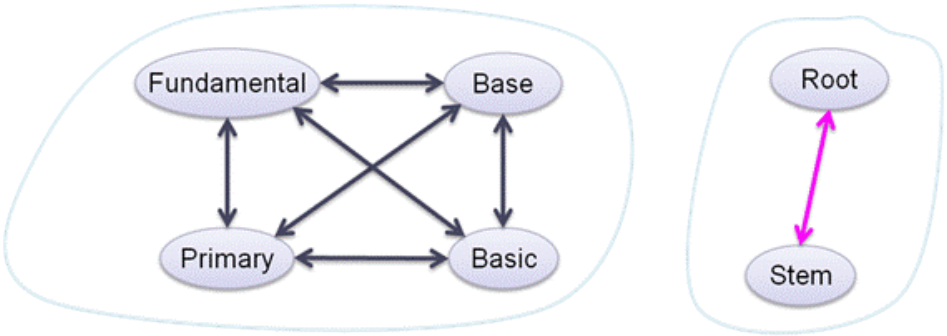


Figura 4-8: Apenas *synsets* mais informativos são mantidos, eliminando a polissemia.

Tradicionalmente, técnicas de desambiguação costumam depender de contexto taxonômico, mas em nosso caso exploramos o contexto do vocabulário como um todo, dado

pela forma como palavras similares se agrupam. Elaboramos uma técnica simples e eficaz para desambiguação que consiste em encontrar e manter apenas os **cliques** ou ***synsets* mais informativos** num domínio, punindo os menos informativos. Recorremos ao conceito de **entropia** (Seção 2.4) oriundo de Teoria da Informação: a **quantidade de informação** em cada *synset* é mensurada (como explicado mais adiante) para ranqueá-los por ordem crescente de informação. Uma estratégia gulosa iterativa penaliza os *synsets* menos informativos: percorrendo o *rank* do inicio ao fim, remove-se de cada *synset* as palavras compartilhadas com outros *synsets* (polissêmicas). O resultado é que, ao atingir o final do *rank*, todas as palavras serão monossêmicas, alguns *synsets* irrelevantes do início do *rank* deixarão de existir e as palavras tenderão a se concentrar nos *synsets* do final do *rank*, que não por coincidência são os *synsets* mais informativos e importantes.

Precisamos de uma **definição de informação**: um *synset* importante possui informação valiosa para o domínio, mas um *synset* sobreposto a muitos outros não traz informação uma vez que é prejudicado pela polissemia de suas palavras; quando mais polissemia num *synset*, menos informação ele terá, constituindo forte candidato a eliminação. A entropia ou quantidade de informação contida num *synset* é então definida em função (a) do total de palavras polissêmicas no *synset* e (b) do total de significados que cada palavra possui, isto é, total de *synsets* que se sobrepõe ao *synset* em questão.

Sejam dois vetores *A* e *B* criados para um *synset S* com *n* palavras, onde *A* é um vetor de inteiros contabilizando os *synsets* de cada palavra em *S*, e *B* é um vetor booleano que marca cada palavra como polissêmica ou monossêmica. A quantidade de informação em *S* é dada pela combinação *NoisyOr* (Seção *2.3*) da informação contida nos vetores *A* e *B*:

Equação 12: $Informacao(A,B) = NoisyOr(Entropia(A), Entropia(B))$

Perceba que quantificar informação como grandeza oposta à polissemia é intuitivo e ajuda a eliminar *synsets* que pouco ou em nada ajudarão. Análogo à definição de entropia, pode-se mensurar a importância do mesmo *synset S* como segue:

Equação 13: $FatorA(S) = 1 - contagemDePalavrasPolissemicas(S)/n$

Equação 14: $FatorB(S) = n/contagemDeSynsetsSobrepostos(S)$

Equação 15: $Fator(S) = NoisyOr(\ fatorA(S),\ fatorB(S))$

A importância máxima "1" é alcançada quando *S* não possui palavra polissêmica. O fator da Equação 13 atinge menor importância "0" quanto toda palavra em *S* é polissêmica, o fator da Equação 14 tende a "0" quanto mais significados cada palavra em *S* possuir e a Equação 15 combina as Equações 13 e 14. A importância de um *synset* é definida como**:**

Equação 16: $Importancia(S) = Informacao(S) * Fator(S)$

Todos os aspectos abordados nesta subseção contribuíram com a precisão da solução proposta neste trabalho. Escopos textuais são essenciais para boa qualidade do mapeamento, enquanto a identificação de sinônimos potencializa sistemas que processam texto, como é o caso do nosso L-*Match*, um mapeador baseado em algoritmos de classificação de texto.

4.2 COMPUTANDO SOBREPOSIÇÃO E SIMILARIDADE ENTRE CONCEITOS

O Módulo de Similaridade do L-*Match* estabelece correspondências iniciais entre conceitos de ontologias distintas através de algoritmos de classificação supervisionada, introduzindo nesta Dissertação a abordagem de Aprendizado de Máquina. Nesta seção, discutiremos como

manipular classificadores para descobrir dois tipos de correspondências entre conceitos: **sobreposição** e **similaridade**. Não pressupomos que as ontologias sejam correlatas, pois se não forem esperamos que os valores de sobreposição e similaridade sejam nulos ou irrisórios.

Inicialmente, nada se conhece sobre o que há em comum entre ontologias distintas. Queremos começar a eliminar esta ignorância classificando instâncias de uma ontologia nos conceitos de outra ontologia e vice-versa, ou seja, descobrir **mapeamentos de pertinência *instância-conceito* (∈)**, como esquematizado a seguir:

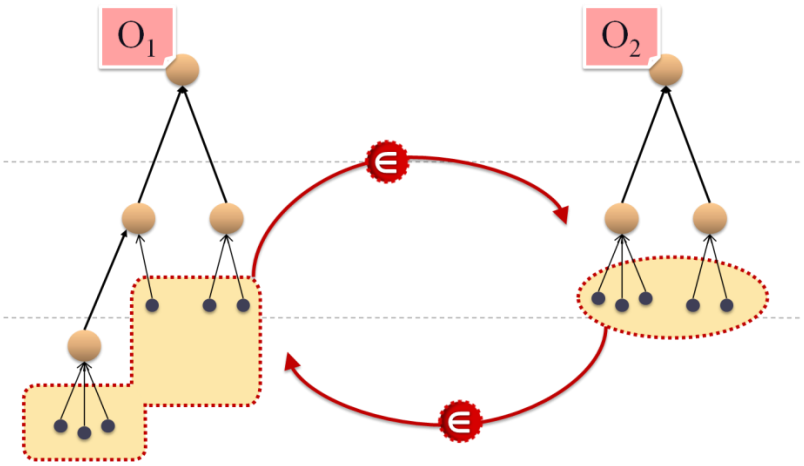


Figura 4-9: Descobrindo relações de pertinência entre ontologias

Algoritmos de classificação relacionam instâncias desconhecidas (conjunto de teste) com categorias conhecidas (conjunto de treino), com certa precisão e utilizando dados heterogêneos: precisamos então adaptar classificadores para trabalhar com heterogeneidade semântica. Recuperar relações de pertinência desconhecidas permitirá comparar conceitos de ontologias e descobrir mapeamentos semânticos também desconhecidos até então. Para isso, o Módulo de Similaridade do L-*Match* reveza as ontologias hora como treino e hora como teste, de forma que as instâncias de uma sejam classificadas nos conceitos da outra:

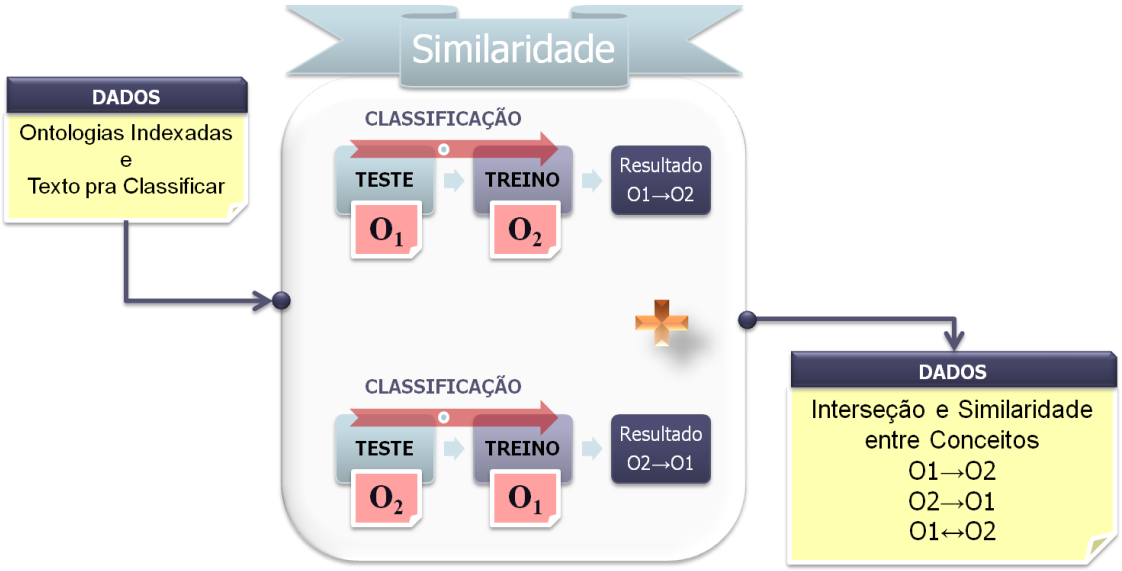


Figura 4-10: Classificação automática alternada das instâncias de um par de ontologias

Quando uma instância é submetida a um classificador, o resultado é um *rank* de conceitos ordenados por similaridade decrescente com a instância, como na Figura 4-11, onde as instâncias do conceito *BaseUnit* foram comparadas por classificadores a conceitos de outra ontologia onde *BaseUnit* não foi declarado:

TESTE	TREINO			
BaseUnit	**1°**	**2°**	**3°**	**4°**
dimensionlessUnit	0,028 UnitOfIdentity	0,026 UnitOfMoney	0,024 UnitOfTime	0,020 DUnitOfFDimension
radian	0,028 UnitOfIdentity	0,026 UnitOfMoney	0,024 UnitOfTime	0,020 DUnitOfFDimension
candela	0,028 UnitOfIdentity	0,026 UnitOfMoney	0,024 UnitOfTime	0,020 DUnitOfFDimension
mole	0,024 UnitOfIdentity	0,023 UnitOfMoney	0,022 UnitOfTime	0,018 UnitOfLength
kelvin	0,028 UnitOfIdentity	0,025 UnitOfMoney	0,024 ScaleOfTemperature	0,024 UnitOfTime
ampere	0,028 UnitOfIdentity	0,026 UnitOfMoney	0,024 UnitOfTime	0,020 DUnitOfFDimension
steradian	0,024 UnitOfIdentity	0,023 UnitOfMoney	0,022 UnitOfTime	0,018 UnitOfLength
meter	0,064 **UnitOfLength**	0,024 UnitOfIdentity	0,022 UnitOfMoney	0,021 UnitOfTime
kilogram	0,076 **UnitOfMass**	0,027 DUnitOfFDimension	0,023 UnitOfIdentity	0,021 UnitOfMoney
second	0,053 **UnitOfTime**	0,046 DUnitOfFDimension	0,024 UnitOfIdentity	0,022 UnitOfMoney
SIMILARIDADE	0,260 ⊏ UnitOfIdentity	0,259 ⊏ UnitOfTime	0,241 ⊏ UnitOfMoney	0,231 ⊏ UnitOfMass
SOBREPOSIÇÃO	7 ⊏ UnitOfIdentity	1 ⊏ UnitOfTime	1 ⊏ UnitOfMoney	1 ⊏ UnitOfMass

Figura 4-11: Ranqueamento de conceitos por similaridade com *BaseUnit* e suas instâncias

O classificador estima a similaridade entre toda instância e todo conceito de treino, mas a classificação de fato só ocorre quando se atribui pertinência entre cada instância com algum conceito: adotamos a estratégia mais simples que é classificar a instância no primeiro conceito do *rank* (o conceito mais similar), ou seja, adotamos um limiar *Top-k* onde *k*=1.

Pertinência $x \in C$ e similaridade $Sim(x, C)$ são relações ***instância-conceito***. Como queremos mapeamentos conceitos, precisamos chegar a relações ***conceito-conceito***. Para isso, podemos descrever os conceitos em função de suas instâncias: sejam dois conceitos $A = \{a_0, a_1, \dots, a_m\}$ e $B = \{b_0, b_1, \dots, b_n\}$, queremos descobrir e contabilizar possíveis $a_i \in B$ e $b_j \in A$ para então (a) estimar o **valor de sobreposição** $|\boldsymbol{A \cap B}|$ e (b) estimar a **similaridade *conceito-conceito*** $\boldsymbol{Sim(A, B)}$ a partir da combinação dos valores de similaridades *instância-conceito* $Sim(x_A, B)$ e $Sim(x_B, A)$. As combinações são feitas através de *NoisyOr* e entropia, como veremos na subseção seguinte.

Como um conjunto de teste é sempre aplicado sobre um conjunto de treino, temos a noção de direção $\boldsymbol{teste \longrightarrow treino.}$ Revezando as ontologias como teste e como treino, podemos estimar os valores $|A \cap B|$ e $Sim(A, B)$ nas direções $\boldsymbol{A \longrightarrow B}$ (contexto da ontologia do conceito *A*) e $\boldsymbol{B \longrightarrow A}$ (contexto da ontologia do conceito *B*), sendo que $\boldsymbol{A \longleftrightarrow B}$ é uma combinação que elimina a noção de direção: trabalhos sobre mapeamento sintático (como o GLUE) perdem informação uma vez que mantêm apenas valores $A \longleftrightarrow B$. Por esta e outras razões (Seção 4.3), $|A \cap B|$ e $Sim(A, B)$ são mantidos nos três sentidos.

4.2.1 Propagação *Bottom-Up* dos Valores de Sobreposição e Similaridade na Taxonomia

Os algoritmos de classificação utilizados precisam receber **conceitos concretos**, isto é, conceitos que recebem **instanciação direta** e não apenas **instanciação indireta** oriunda de subconceitos. Numa taxonomia, é comum que conceitos concretos estejam concentrados apenas em níveis mais específicos: na prática, a maioria das instâncias está nas folhas da taxonomia. Conseqüentemente, $|A \cap B|$ e $Sim(A,B)$ seriam estimados apenas em conceitos específicos, excluindo conceitos mais genéricos menos providos de instâncias diretas, mas que também precisam ser mapeados. Portanto, é necessário propagar $|A \cap B|$ e $Sim(A,B)$ de baixo para cima (*bottom-up*) na taxonomia.

Segundo a semântica taxonômica, um conceito é a união ($\sqcup$) de suas **subsunções**, ou seja, instâncias e subconceitos. Como a união equivale à operação ***OR*** ($\vee$) e $Sim(A,B)$ é uma probabilidade, utilizamos o **Modelo *NoisyOr* de Redes Bayesianas** para **propagação *bottom-up* de $\boldsymbol{Sim(A,B)}$** desde conceitos específicos até o topo genérico de uma taxonomia. Sejam os conceitos *A* e *B*, a **propagação** e a **combinação** de similaridade estão implícitas no cálculo recursivo de $Sim(A,B)$. Então, para estimar o **quanto *A* é similar a *B*** temos a definição de $Sim(A,B)$ na direção $A \longrightarrow B$ como segue:

Equação 17: $$\overset{A\to B}{Sim}(A,B) = 1 - \left(\prod_{x\in A}(1 - Sim(x,B))\right) * \left(\prod_{y\subset A}(1 - \overset{y\to B}{Sim}(y,B))\right)$$

O primeiro produtório combina os valores de similaridade estimados entre o conceito *B* e toda instância direta $x \in A$ (a instância não precisa ter sido classificada em *B*, basta o valor de similaridade); o passo recursivo está no segundo produtório que combina a similaridade estimada entre todo subconceito direto $y \subset A$ e o conceito *B* na direção $y \longrightarrow B$.

Dessa forma, os valores de similaridades *instância-conceito,* obtidos pelos classificadores, são combinados numa similaridade *conceito-conceito* que é propagada até conceitos mais genéricos. Por fim, Observe que a Equação 17 é um *NoisyOr* e isso significa que a similaridade tende a crescer conforme é propagada para cima.

Queremos agregar Teoria da Informação à definição de $Sim(A,B)$. Se por um lado a similaridade aumenta quando generalizada pela propagação *bottom-up* uma vez que a disponibilidade de instâncias indiretas é maior em conceitos genéricos, por outro lado perde-se informação uma vez que a generalização descarta propriedades de conceitos específicos. Portanto, quanto mais genérico um conceito, menos informação ele contém (RESNIK, 1990). Recorremos novamente à definição de entropia para alterar a definição de $Sim(A,B)$ no sentido $A \longrightarrow B$:

Equação 18: $\overset{A \to B}{Sim}(A,B) = NoisyOr(V) * Entropia(V)$

- Onde V é a enumeração de $\overset{x \to B}{Sim}(x,B)$, para todo $x \in A$ ou $x \subset A$.
- A entropia penaliza a similaridade conforme a informação em V diminui.

A perda de informação também ocorre devido a erros de classificação. Por exemplo, mesmo que uma instância de um conceito *A* erroneamente receba grande similaridade com um conceito *B* o qual é diferente de *A*, espera-se que o mesmo não ocorra com a maioria das instâncias de *A*. Certamente, a similaridade entre *A* e *B* será penalizada pela entropia, para não implicar, por exemplo, numa falsa equivalência entre *A* e *B*.

A Figura 4-12 esquematiza uma Rede Bayesiana propagando similaridade numa taxonomia. Nela, conceitos são representados por círculos maiores e instâncias por círculos

menores. Deseja-se então calcular a similaridade entre um conceito arbitrário *Q* com todos os conceitos da taxonomia, partindo da similaridade inicialmente estimada por um classificador entre *Q* e cada instâncias da taxonomia:

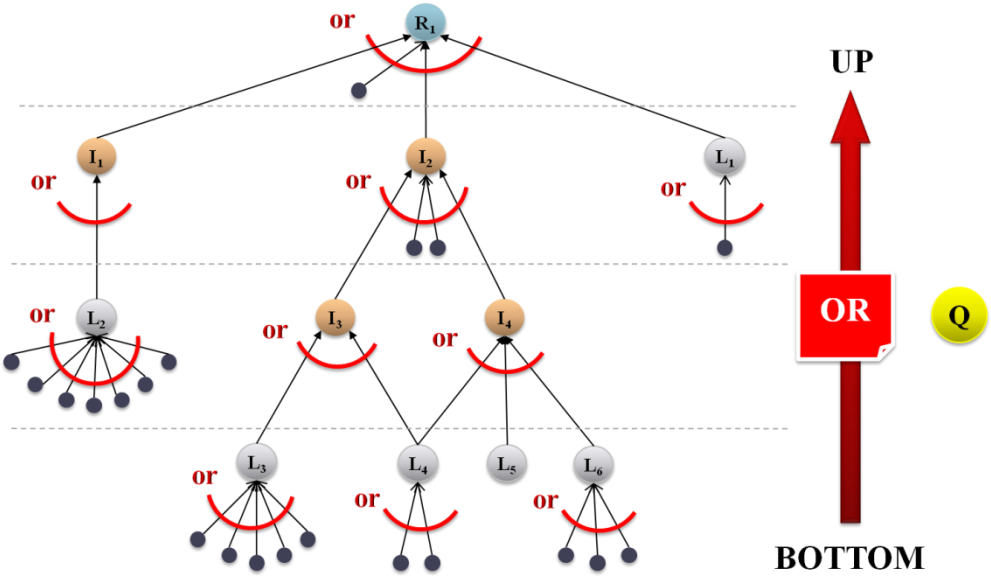


Figura 4-12: Rede Bayesiana para propagação da similaridade entre o conceito Q e os conceitos de uma taxonomia

Cada marcação *OR* em vermelho corresponde a uma chamada à Equação 18. Veja que as combinações iniciam nas instâncias, tanto que qualquer conceito não instanciado (direta ou indiretamente) acaba sendo ignorado por nossa abordagem.

Análogo ao cálculo de similaridade, o valor de sobreposição $|A \cap B|$ tem uma definição recursiva responsável pela **propagação *bottom-up* de $|A \cap B|$** que é feita através da operação **soma (+)** que, como o *NoisyOr*, também é análoga à operação de união. Portanto a definição de $|A \cap B|$ no sentido $A \longrightarrow B$ é:

Equação 19: $$\overset{A \to B}{Sobreposicao}(A,B) = \overset{A \to B}{sobreposicaoDireta}(A,B) + \sum_{y \subset A} \overset{y \to B}{Sobreposicao}(y,B)$$

A Equação 19 contabiliza todas as instâncias (diretas ou indiretas) que os conceitos *A* e *B* compartilham, segundo a resposta do algoritmo de classificação utilizado. O resultado é uma estimativa do tamanho da sobreposição ou interseção entre *A* e *B*.

Até agora, sempre apresentamos cálculos em alguma direção, isto é, de $A \longrightarrow B$ ou de $B \longrightarrow A$, pois estes cálculos serão essenciais para encontrar mapeamentos semânticos de **mais geral** e de **menos geral** no Módulo de Mapeamento (Seção 4.3). Contudo, podemos também obter cálculos de $A \leftrightarrow B$, isto é, sem direcionamento. Para isso, dizemos que $A \leftrightarrow B$ é a combinação de valores estimados nas direções $A \longrightarrow B$ e $B \longrightarrow A$. Portanto, para combinar similaridade (Equação 20) e sobreposição (Equação 21), utilizamos a média harmônica dos valores direcionados:

Equação 20: $$\overset{A\leftrightarrow B}{Sim}(A,B) = \frac{2 * \overset{A\rightarrow B}{Sim}(A,B) * \overset{B\rightarrow A}{Sim}(A,B)}{\overset{A\rightarrow B}{Sim}(A,B) + \overset{B\rightarrow A}{Sim}(A,B)}$$

Equação 21: $$\overset{A\leftrightarrow B}{|A \cap B|} = \overset{A\rightarrow B}{|A \cap B|} + \overset{B\rightarrow A}{|A \cap B|}$$

Durante o desenvolvimento deste trabalho, diferentes...

4.2.2 Combinação de Classificadores

Diferentes algoritmos de classificação foram investigados e comparados (Seção 5.3) em busca da melhor forma de *classificar para mapear*, incluindo *k*NN, *Naive Bayes*, SVM, Máxima Entropia, NB-*Shrinkage*, TF-IDF e combinações entre eles.

Ensemble ou *meta-learner* é a denominação de um classificador resultante da combinação de outros. Combinam-se as decisões dos algoritmos e não os algoritmos propriamente ditos (embutindo um no outro). Apesar de não haver garantias, espera-se que a eficácia da classificação melhore se entendermos a combinação como um processo democrático: se um classificador individual erra, espera-se que a maioria dos classificadores

não o faça. Por outro lado, o processo demora mais à medida que mais classificadores são combinados. O ideal seria que um classificador individual produzisse a melhor eficácia, mas isso nem sempre ocorre, sendo válido investigar *ensembles*.

Convencionalmente, combinação de classificadores é feita com *NoisyOr*. Como na Equação 18, preferimos combinar *NoisyOr* e entropia para penalizar respostas controversas quando classificadores discordam entre si. Então, seja x uma instância de teste, C um conceito de treino e R um vetor de respostas dadas por n classificadores, temos que:

Equação 22: $combinacaoDaClassificacao(R) = NoisyOr(R) * Entropia(R)$

4.3 COMPUTANDO MAPEAMENTOS SEMÂNTICOS ENTRE CONCEITOS

Os módulos de Extração e Similaridade fornecem satisfatoriamente valores de similaridade e sobreposição entre conceitos de ontologias distintas, valores que na seqüência são utilizados para alimentar o Módulo de Mapeamento, onde finalmente os mapeamentos são estabelecidos em sua forma axiomática e que por isso são semânticos.

4.3.1 Regras de Compatibilidade entre Conceitos

A semântica formal dos axiomas de mapeamento que queremos utilizar é dada pela compatibilidade entre a classificação de instâncias (BOUQUET, *at al.*, 2003): uma função de mapeamento será extensionalmente correta se as regras abaixo se aplicarem às instâncias de dois conceitos *A* e *B* oriundos de hierarquias taxonômicas:

$$
\begin{aligned}
A \xrightarrow{\equiv} B \quad &\Rightarrow \quad instâncias(A\downarrow) = instâncias(B\downarrow) \\
A \xrightarrow{\sqsupset} B \quad &\Rightarrow \quad instâncias(A\downarrow) \supset instâncias(B\downarrow) \\
A \xrightarrow{\sqsubset} B \quad &\Rightarrow \quad instâncias(A\downarrow) \subset instâncias(B\downarrow) \\
A \xrightarrow{\sqcap} B \quad &\Rightarrow \quad instâncias(A\downarrow) \cap instâncias(B\downarrow) \quad \neq \emptyset \\
A \xrightarrow{\not\equiv} B \quad &\Rightarrow \quad instâncias(A\downarrow) \cap instâncias(B\downarrow) \quad = \emptyset
\end{aligned}
$$

Onde o símbolo $\downarrow$ indica que as instâncias indiretas também são consideradas. Além disso, a precedência das regras é de cima pra baixo, isto é, da mais forte para a mais fraca. Isso significa que para uma regra ser aplicável é necessário que outra mais forte não seja. Estas regras podem ser traduzidas para um algoritmo:

01	***SE***	*as instâncias de A e B sao as mesmas*	***ENTÃO*** $A \equiv B$
02	***SENÃO SE***	*toda instância de A pertence a B*	***ENTÃO*** $A \sqsubset B$
03	***SENÃO SE***	*toda instância de B pertence a A*	***ENTÃO*** $A \sqsupset B$
04	***SENÃO SE***	*existe alguma instância comum a A e B*	***ENTÃO*** $A \sqcap B$
05	***SENÃO***	$A \not\equiv B$	

Esta idéia é simples, porém chave, pois justifica nossa abordagem com classificação de instâncias. Mas da forma como estão, estas regras não funcionarão se algum erro tiver ocorrido durante a classificação. Por exemplo, se dois conceitos são equivalentes não é garantido que, com a ajuda dos classificadores, seremos sempre capazes de descobrir que estes conceitos compartilham 100% de suas instâncias. Isso acontece porque algoritmos de classificação são heurísticas e é esperado que errem. Por esta razão é mais apropriado **relaxar** as regras de compatibilidade antes de usá-las, o que pode ser feito utilizando alguns limiares. Poderíamos então assumir, por exemplo, que dois conceitos são equivalentes se descobrirmos que eles compartilham pelo menos 80% das suas instâncias.

A relação de $menos\ geral$ ($\sqsubset$) foi escolhida como base para implementar outras regras: considera-se *A* menos geral que *B* quando toda instância de *A* também pertence a *B*. Esta idéia é relaxada na Equação 23:

Equação 23: $$menosGeral(A,B) = \frac{\overset{A\to B}{|A \cap B|}}{|A|} \geq T_{\max\_overlap}$$

- Onde $|A \cap B|$ é a sobreposição na direção $A \longrightarrow B$, pois inclui apenas instâncias de *A* classificadas em *B*, independente das instâncias de *B* classificadas em *A*;
- $0.5 < T_{max\_overlap} \leq 1.0$, pois menos que 50% de sobreposição não faz qualquer sentido se esperamos que $A \subset B$, ao passo que 100% de sobreposição seria teoricamente o ideal. Contudo, um bom valor de limiar deve estar, empiricamente, entre 0.75 ~ 0.90 para efeito de relaxamento.

A Equação 23 pode ser melhorada se agregarmos a ela o resultado de mais algumas observações. Assim sendo, quando um conceito *A* está contido em outro conceito *B*:

a) espera-se que $\overset{A\to B}{Sim}(A,B)$ seja 1, pois toda instância de *A* está em *B*. Podemos esperar que esta similaridade esteja próxima de 1, logo devemos estabelecer um limiar $T_{\max\_sim}$ próximo de 1:

$$regraA(A,B) = \overset{A\to B}{Sim}(A,B) \geq T_{\max\_sim}$$

b) espera-se que $\overset{B\to A}{Sim}(A,B)$ tenda a 0, pois nem toda instância de B está em A. Como este valor tende a 0 mas não é 0, podemos esperar que exista um

mínimo de sobreposição ou de similaridade na direção $B \longrightarrow A$, logo devemos estabelecer um limiar α próximo de 0:

$$regraB(A,B) = \left(\frac{|A \cap B|}{|B|}^{B \to A} \geq \alpha \right) \vee \left(Sim(A,B)^{B \to A} \geq \alpha \right)$$

Observe que a *regraB* recorre tanto ao valor de sobreposição quanto ao valor similaridade dentro de uma cláusula *OR*. Isso é útil porque algumas vezes o valor de sobreposição pode ser irrisório ou até nulo se errarmos a classificação da maioria das instâncias de *B* que também pertencem a *A*. Neste caso, resta o valor de similaridade como última alternativa.

Seguindo as observações nos itens **a** e **b**, finalmente alteramos a Equação 23 para:

Equação 24: $$menosGeral(A,B) = \left(\frac{|A \cap B|}{|A|}^{A \to B} \geq T_{\max\_overlap} \right) \wedge regraA(A,B) \wedge regraB(A,B)$$

Podemos definir $equivalência$ ($\equiv$) e $mais\ geral$ ($\sqsupset$) em função de $menos\ geral$($\sqsubset$), uma vez que $\sqsupset$ e $\sqsubset$ são inversos e $\equiv$ ocorre quando dois conceitos estão contidos um no outro:

Equação 25: $$maisGeral(A,B) = menosGeral(B,A)$$

Equação 26: $$equivalente(A,B) = menosGeral(A,B) \wedge menosGeral(B,A)$$

Não implementamos a relação de $sobreposição$ ($\sqcap$) em função de $menos\ geral$ uma vez que a relação de *sobreposição* é mais fraca. Esta relação é aplicada quando ainda resta alguma interseção/sobreposição entre os conceitos a mapear, mas esta sobreposição não caracteriza as relações de $equivalência$, nem $mais\ geral$ e nem $menos\ geral$. Na prática, a principal preocupação é diferenciar a relação de *sobreposição* da relação de *diferença* ($\not\equiv$). Em condições ideais, dois conceitos são diferentes quando não possuem qualquer instância em comum, mas para efeito de relaxamento, devemos assumir um limiar $T_{\min\_overlap}$ próximo a 0:

Equação 27: $$sobreposto(A,B) = \left(\frac{|A \cap B|}{|A|}^{A \to B} \geq T_{\min\_overlap} \right) \wedge \left(\frac{|A \cap B|}{|B|}^{B \to A} \geq T_{\min\_overlap} \right)$$

A $diferença$ ($\not\equiv$) representa a disjunção entre conjuntos e é aplicada automaticamente quando todas as relações anteriores falharem.

Com isso, finalizamos as definições das relações que propomos utilizar como axiomas ponte entre conceitos. Então, temos agora como consultar qual a relação mais apropriada para cada par de conceitos e implementar nosso algoritmo no método ***consultaRelacao***:

```
00  consultaRelacao(A, B) : Relacao
01  INICIO
02  :  SE          equivalente(A, B)    RETORNE ≡
03  :  SENAO SE    maisGeral(A, B)      RETORNE ⊐
04  :  SENAO SE    menosGeral(A, B)     RETORNE ⊏
05  :  SENAO SE    sobreposto(A, B)     RETORNE ⊓
06  :  SENAO                            RETORNE ≢
07  FIM
```

4.3.2 Comparação *Top-Down* de Conceitos

De posse do método *consultaRelacao*, o passo seguinte poderia ser comparar todos os conceitos entre si ao custo de uma complexidade quadrática. Contudo, essa abordagem não é interessante uma vez que, devido a erros oriundos da classificação das instâncias, observamos algumas relações fortes sendo estabelecidas entre conceitos distintos, claramente localizados em regiões taxonômicas também distintas.

Na tentativa de contornar os erros que naturalmente ocorrem durante a classificação, exploramos a estrutura taxonômica das ontologias para evitar erros grosseiros de mapeamento. Basicamente, a idéia é percorrer as taxonomias de cima para baixo comparando seus conceitos, ou em outras palavras, utilizar uma **estratégia dedutiva *top-down*** para mapear primeiramente conceitos mais genéricos, de maneira a traçar o contexto taxonômico (conjunção de ancestrais) de conceitos mais específicos para só então mapeá-los usando o método *consultaRelacao*. Esta estratégia tem caráter dedutivo porque, como toda dedução, envolve inferência partindo de princípios gerais, isto é, do universal para o particular.

Intuitivamente, como conceitos similares contêm subconceitos também similares, a comparação *top-down* usa as taxonomias como guias, evitando comparações desnecessárias para reduzir o espaço de busca. Outra vantagem é que a hierarquia taxonômica melhora a estimativa dos parâmetros de entrada (RESNIK, 1990) que no nosso caso são os valores de similaridade e de sobreposição. Isso acontece porque devido à propagação *bottom-up*, conceitos genéricos, outrora problemáticos por falta de instâncias diretas, passam a ter mais instâncias indiretas do que conceitos específicos. Logo, parâmetros estimados próximo ao topo taxonômico são mais confiáveis. Espera-se então que esta estratégia repare estimativas menos confiáveis nas classes específicas, evitando falsos mapeamentos fortes entre conceitos distintos de regiões taxonômicas de fato distintas.

A comparação *top-down* é simples, recursiva e foi implementada como no método $\boldsymbol{compareTopDown}$ que recebe dois conceitos *A* e *B*:

```
00  compareTopDown(A, B) : void
01  INICIO
02  ⋮  relacao = consultaRelacao(A, B)
03  ⋮  ESCOLHA(relacao)
04  ⋮  INICIO
05  ⋮  ⋮  CASO ≡ : compareTopDown(subconceitos[A], subconceitos[B]) PARE
06  ⋮  ⋮  CASO ⊐ : compareTopDown(subconceitos[A], B) PARE
07  ⋮  ⋮  CASO ⊏ : compareTopDown(A, subconceitos[B]) PARE
08  ⋮  ⋮  CASO ⊓ : compareTopDown(A, subconceitos[B])
09  ⋮  ⋮           compareTopDown(subconceitos[A], B) PARE
10  ⋮  ⋮  CASO ≢ : compareTopDown(A, subconceitos[B])
11  ⋮  ⋮           compareTopDown(subconceitos[A], B) PARE
12  ⋮  FIM
13  FIM
```

Diferente de $compareTopDown(A, B)$, as chamadas recursivas nas linhas 5 a 11 recebem vetores. Deve-se assumir estes casos como iterações implícitas que chamam novamente $compareTopDown(A, B)$, seguindo a ordenação decrescente de $Sim^{A \leftrightarrow B}(A, B)$, o que chamamos de **alinhamento horizontal**.

Dependendo da relação estimada, os argumentos da chamada recursiva são diferentes, o que faz com que o próprio caminhamento recursivo seja diferente, evitando comparações desnecessárias não demandadas pela relação a ser estimada:

- Ao estimar e assumir $A \equiv B$, deve-se comparar apenas os subconceitos entre si, evitando comparar A com $subconceitos[B]$ ou comparar B com $subconceitos[A]$, pois ao assumir $A \equiv B$ já é possível inferir o que descobriríamos com essas comparações;

- Ao estimar e assumir $A \sqsupset B$, deve-se comparar B a $subconceitos[A]$. É desnecessário comparar A com $subconceitos[B]$, pois ao assumir $A \sqsupset B$ já podemos inferir o resultado destas comparações. Analogamente, o mesmo é feito ao estimar $A \sqsubset B$;
- Ao estimar e assumir $A \sqcap B$, significa que ainda pode existir alguma relação mais específica entre subconceitos de A e B, ou mesmo entre A com $subconceitos[B]$ ou entre B com $subconceitos[A]$. A relação mais específica pode ser equivalência, mais/menos geral ou sobreposição;
- Ao estimar e assumir $A \not\equiv B$, talvez ainda exista alguma relação que, em classes mais específicas, passe nos testes de limiar durante a relaxação das regras de compatibilidade.

Seria adequado explicar pq se pode inferir isso, algo como "... ao assumir A»B os subconceitos de B seriam todos menos gerais que A. Resta prosseguir a comparação através dos subconceitos de A..."

4.3.3 Alinhamento Vertical de Equivalências

O método $compareTopDown$ apresentado pode receber melhorias. Por exemplo, evitar a recursão ao estimar $A \not\equiv B$ sendo possível saber que $A \cap B = \emptyset$. Outro exemplo é quando detectamos $A \equiv B$ no topo das taxonomias, situação em que não sabemos se a equivalência realmente ocorre no topo ou é fruto de uma equivalência que na realidade ocorre entre conceitos mais específicos, logo a equivalência precisa de um **alinhamento vertical.** Esse problema ocorre devido à possibilidade de erro durante a classificação e devido à eventual escassez ou má distribuição de instâncias concentradas em poucos conceitos mais específicos.

Por exemplo, a Figura 4-13 exibe uma situação desfavorável ao mapeamento, ilustrando conceitos localizados em duas ramificações oriundas de taxonomias distintas. As

equivalências exibidas são mapeamentos computados (e não mapeamentos ideais) e o número acima das equivalências representa o valor $|A \cap B|$ computado no Módulo de Similaridade:

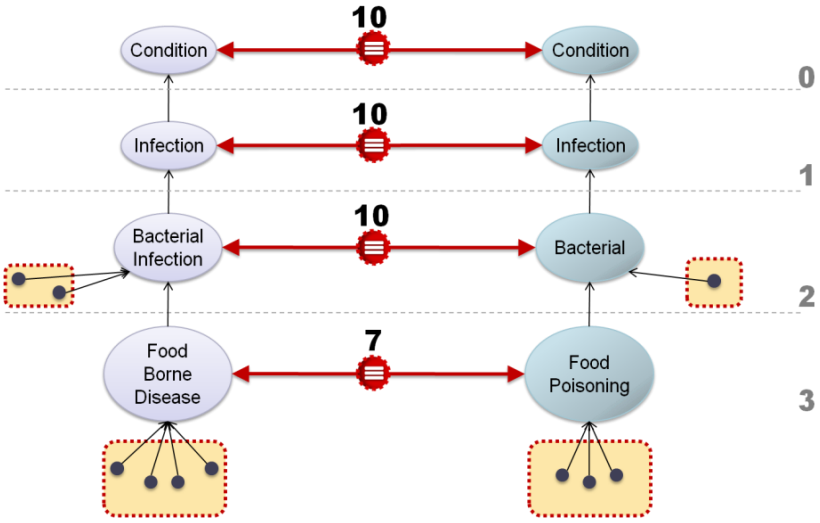


Figura 4-13: Má distribuição de instâncias numa ramificação da taxonomia

O problema começa no topo da taxonomia: as regras de compatibilidade avaliariam *Condition* e *Condition* como conceitos equivalentes (e de fato são), uma vez que compartilham 100% de suas instâncias, 10 ao todo. Este aparente acerto não passa de uma coincidência, pois a sobreposição estimada foi insuficiente, logo não deveríamos assumir esta equivalência. Acontece que *Condition* e *Condition* são **conceitos abstratos** (sem instância direta) e as 10 instâncias levadas em consideração são oriundas de conceitos mais específicos onde a equivalência é mais evidente. O ideal seria que *Condition* e *Condition* tivessem outros subconceitos instanciados ou mais instâncias diretas, ou seja, falta uma melhor distribuição de instâncias no exemplo da Figura 4-13.

Explicar por que.

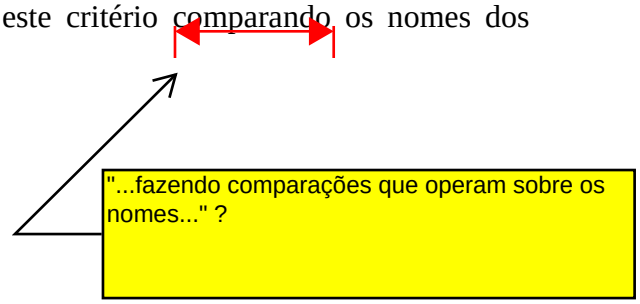
Nesta situação, precisaríamos de algum outro critério além da estimativa de $|A \cap B|$ para **alinhar verticalmente** a equivalência sugerida, isto é, descobrir se essa equivalência realmente é aplicável no nível de abstração em que foi sugerida ou se é mais apropriado deslocá-la para níveis mais específicos.

O alinhamento vertical foi implementado priorizando níveis mais específicos de uma taxonomia: se o valor de $|A \cap B|$ for igual no próximo nível mais específico, a equivalência desce um nível, exceto se alguma outra condição se opuser a isso. Por exemplo, a equivalência detectada entre o par (*Condition*,*Condition*) desceria enquanto $|A \cap B| = 10$ até chegar em (*BacterialInfection*,*Bacterial*). Com isso, perdemos possíveis equivalências em níveis mais genéricos, contudo garantimos que as equivalência só são assumidas quando realmente temos certeza.

Portanto, quando a distribuição de instâncias é ruim e recorremos ao alinhamento vertical, precisamos de outro critério que pare o alinhamento vertical e identifique equivalências em níveis mais genéricos, as quais outrora seriam perdidas. Sejam os conceitos *A*, *B* e *B' tal que B'*$\subset B$, definimos o alinhamento vertical da equivalência sugerida entre *A* e *B* como:

$$alinhamentoVertical(A,B) = \begin{cases} B & se \quad |A \cap B|^{A \leftrightarrow B} > |A \cap B'|^{A \leftrightarrow B'} \\ B' & se \quad |A \cap B|^{A \leftrightarrow B} = |A \cap B'|^{A \leftrightarrow B'} \ \wedge \ \neg\, equivalente'(A,B) \end{cases}$$

O critério $equivalente'(A,B)$ deve ser uma técnica extra não baseada nos valores de sobreposição que computamos com a ajuda dos classificadores, para determinar se *A* e *B* são equivalentes. Atualmente, o L-*Match* implementa este critério comparando os nomes dos conceitos.



Capítulo 5

EXPERIMENTOS

Neste capítulo, será discutida e apresentada a avaliação da qualidade dos resultados da abordagem de mapeamento semântico proposta no presente trabalho. Primeiramente serão apresentadas as ontologias experimentadas juntamente com os recursos computacionais utilizados, seguidos pela discussão do método de avaliação hierárquico e, finalizando o capítulo, os resultados de avaliação obtidos serão apresentados e comentados.

Apesar de ser considerada importante, a eficiência (velocidade) não é tratada rigorosamente neste trabalho, uma vez que ontologias normalmente têm poucos conceitos (menos que 1000) e também poucas instâncias. Como nosso objetivo é criar uma nova abordagem, precisamos primeiramente nos certificar da eficácia (qualidade) da abordagem e saber se ela é viável e promissora, para só depois pensar em otimizações de desempenho, se for o caso.

No contexto de mapeamento entre ontologias e esquemas, existem iniciativas[7] de avaliação que tentam trazer algum consenso para o tema, porém seus padrões são limitados, mais voltados a mapeamento sintático e não fornecem *baselines* realmente úteis, seja devido

[7] http://www.ontologymatching.org/evaluation.html

a *links* quebrados, ontologias corrompidas, ontologias triviais ou, o que mais acontece, ontologias não instanciadas que definitivamente não servem para avaliar mapeadores baseados em instâncias como o L-*Match*. Na prática, muitos trabalhos não utilizam estas iniciativas de padronização, pois ainda são precárias e não atendem a demanda. Por isso estabelecemos ~~nossa própria abordagem de avaliação e também~~ escolhemos as ontologias a experimentar, como veremos na seção a seguir.

"... estabelecemos uma abordagem diferenciada de avaliação que permite escolhermos ..."

Para a realização dos experimentos, todos os códigos foram feitos em classes implementadas em JAVA e compiladas com J2SDK 1.6. Os experimentos foram feitos em ambiente Ubuntu Linux 10.4, numa máquina HP Intel(R) Core(TM) 2 Duo T5250 1.50GHz com 2GB RAM. Utilizamos a API JENA[8] para ler ontologias em OWL, a *Rainbow*[9] como serviço de classificação e o *WordNet* 2.1[10] como serviço de *thesaurus*.

5.1 ONTOLOGIAS UTILIZADAS

Baselines são bases oficiais usadas como referência para trabalhos subseqüentes, mas quando os *baselines* demandados são ontologias instanciadas, a dificuldade para encontrá-los é imensa. Na falta destes, descreveremos a seguir as ontologias que utilizamos.

As ontologias Ecolíngua (BRILHANTE, 2004) e Apes (ATHANASIADIS, *at al.*, 2006) possuem domínios sobrepostos e foram utilizadas para conduzir nossos experimentos. A Ecolíngua é uma ontologia do domínio ecológico, originalmente implementada em Prolog e posteriormente em OWL, instanciada com exemplos do *PondSystem* (GRANT, *at al.*, 1997)

[8] http://jena.sourceforge.net/index.html
[9] http://www.cs.cmu.edu/~mccallum/bow/rainbow/
[10] http://wordnet.princeton.edu/

(FORD, 1999) (HAEFNER, 1996). Por sua vez, a Apes é uma ontologia do domínio agrícola desenvolvida pelo Projeto SEAMLESS-IF[11], relacionado a problemas de modelagem ambiental e agrícola. Este é o par de ontologias mais importante para o presente trabalho, pois contém ontologias cuidadosamente elaboradas, criadas para aplicações reais e desenvolvidas por grupos de trabalho distintos que não sabiam da existência um do outros, ou seja, as ontologias não foram criadas artificialmente para avaliar mapeamentos. Conseqüentemente, apesar de serem correlatas, Apes e Ecolíngua tiveram implementações bem diferentes o que, somado ao fato de possuírem muitos conceitos, as torna muito atraentes para avaliar o desempenho do L-*Match*.

Mesmo sendo instanciadas, Apes e Ecolíngua não possuem uma quantidade ideal de instâncias, criando um desafio ainda maior para serem mapeadas pelo L-*Match*. Os gráficos a seguir apresentam a distribuição de instâncias por classe concreta nestas ontologias:

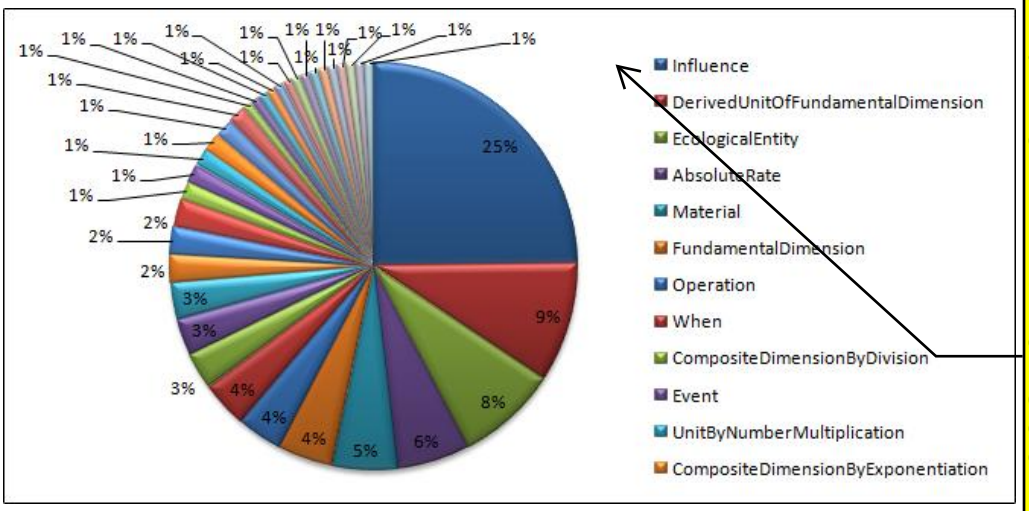
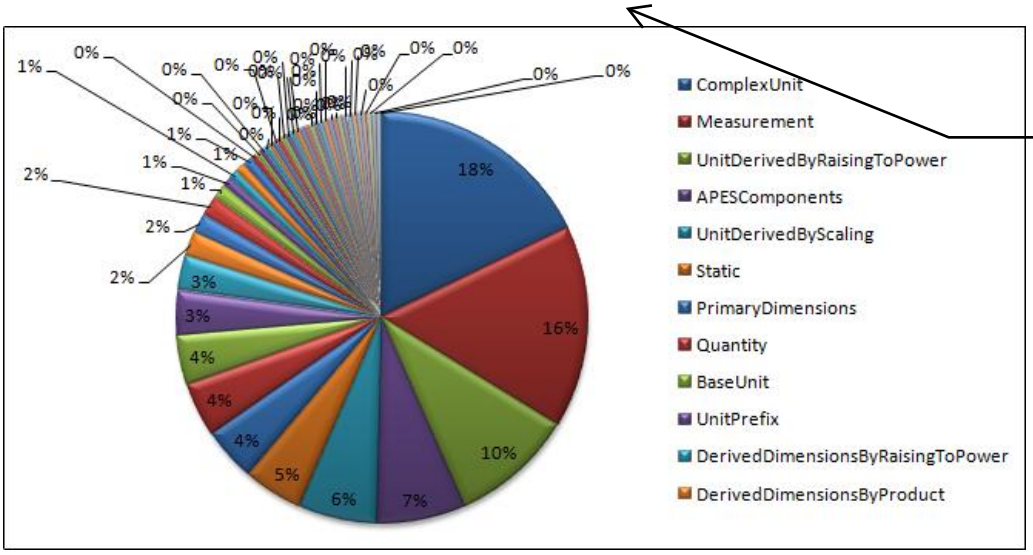


Figura 5-1: Distribuição de instâncias na Ecolingua.owl

Um primeiro problema é que há 35 valores de percentagens e apenas 12 cores nas legendas, o que impossibilita o entendimento. Possíveis alterações:
(1) retirar os 23 menores valores e redesenhar o gráfico
(2) na impossibilidade da opção anterior, tente mudar o tipo de gráfico, talvez p/ colunas 3D, p/ver se ele torna-se mais legível
(3) se for impossível melhorar isso, pelo menos assegure-se que a banca receba essas páginas impressas em cores (em laser mono é impossível diferenciar a legenda)

[11] http://www.seamless-ip.org/



Igual (na verdade, um pouco pior) à Fig.5-1

Figura 5-2: Distribuição de instâncias na Apes.owl

A distribuição não é homogênea, mas não houve problema de desbalanceamento de classes durante a classificação das instâncias. Acreditamos que isso ocorra porque o conteúdo das instâncias ontológicas é bastante preciso, permitindo separar bem as classes.

"Tabela 5-1"

Outros pares de ontologias também foram utilizados, cujas informações aparecem resumidas na tabeladas a seguir, incluindo conceitos concretos (diretamente instanciados), abstratos (instanciados indiretamente apenas) e vazios, propriedades de objeto (owl:ObjectProperty) e elementares (owl:DataTypeProperty) e dados sobre instâncias:

Tabela 5-1: Informações sobre classes, instâncias e propriedades das ontologias utilizadas

PAR	ONTOLOGIA	CONCEITOS				PROPRIEDADES		INSTÂNCIAS		
		Total	Concreto	Abstrato	Vazio	Objeto	Elementar	Total	Por conceito	Por conceito concreto
1°	Apes	157	47	25	85	68	70	258	1.64	5.48
	Ecolíngua	65	35	25	5	32	6	136	2.09	3.88
2°	Disease1	9	4	3	2	3	1	8	0.88	2.0
	Disease2	7	4	3	0	4	0	12	1.71	3.0
3°	Cornell	21	17	4	0	0	0	398	18.95	23.41
	Washington	31	25	6	0	0	0	721	23.25	28.84
4°	Russia1	162	75	32	55	61	19	239	1.47	3.18
	Russia2	151	51	39	61	59	16	157	1.03	3.07

O segundo par é composto por ontologias simples e pequenas sobre doenças, por isso tem função mais ilustrativa e foram implementadas exatamente como foram encontradas em (UDREA, *at al.*, 2007). O terceiro par e composto por uma amostra dos catálogos de cursos da Universidade de Cornell[12] (Austrália) e da Universidade de Washington[13] (Estados Unidos), idêntica à amostra utilizada pelo S-*Match*. Estes catálogos são freqüentemente utilizados como caso de teste em trabalhos de integração de ontologias, como S-*Match* e GLUE, o que permite fazer alguma comparação, mas não são ontologias originalmente, e sim esquemas formados por cursos, departamentos, colégios e escolas que funcionam como conceitos organizados hierarquicamente que podem ser instanciados pelas disciplinas semestrais. Por fim, o quarto par de ontologias é sobre a Rússia, é publicamente conhecido assim como seus mapeamentos, é conhecido por ser muito difícil de mapear e foi testado por vários sistemas durante a I3CON[14], *The Information Interpretation and Integration Conference*, em 2004.

" é "

"... mapeamentos, é considerada muito difícil de mapear, tendo sido testado ..."

5.2 AVALIAÇÃO HIERÁRQUICA DA QUALIDADE DOS MAPEAMENTOS COMPUTADOS

Problemas mais elaborados por envolver classes organizadas em hierarquia taxonômica inserem maior dificuldade na solução e na avaliação do problema, pois desafiam a capacidade computacional de fazer generalizações. Exemplos de problemas assim são mapeamento semântico e classificação hierárquica.

[12] http://www.cuinfo.cornell.edu/Academic/Courses/
[13] http://www.washington.edu/students/crscat/
[14] http://www.atl.external.lmco.com/projects/ontology/i3con.html

Apesar de não existir uma abordagem padrão para avaliar estes problemas, sabemos intuitivamente que devemos julgar os resultados computados comparando-os com resultados ideais dados por especialistas humanos. Porém, mesmo os especialistas podem não ter certeza sobre o mapeamento ideal quando a subjetividade é grande, o que exige certa flexibilidade ao estabelecê-los, permitindo que mais de um mapeamento seja aceito em casos particulares.

Baseamos nossa avaliação nas métricas de precisão, revocação e medida-F. Contudo, se utilizadas de maneira convencional num ambiente hierárquico, estas métricas não capturariam detalhes importantes e considerariam nossos resultados completamente errados: por exemplo, quando uma equivalência entre dois conceitos *A* e *B* é esperada, mas é erroneamente computada entre *A* e algum *Ancestral(B)*, não podemos dizer que o erro foi total, pois ainda é possível inferir que *A* é ancestral de todo *Descendente(B)* e que *A* descende de algum *Ancestral(B)*.

Para resolver este impasse, nos inspiramos na idéia de **escopo expandido** (DING, *at al.*, 2000) para primeiramente inferir mapeamentos implícitos para só então avaliá-los: seja *M* um mapeamento entre dois conceitos *A* e *B*, temos então que $Expansao(M)$ é o conjunto formado por *M* e por todo mapeamento inferível a partir de *M*. No nosso caso, a expansão ou inferência precisa ocorrer de maneira diferenciada dependendo do axioma utilizado no mapeamento, o que acontece seguindo as regras abaixo:

Tabela 5-2: Regras de expansão de mapeamentos

$equivalente(A, B)$	*A* e *B* compartilham os mesmos ancestrais e descendentes.
$maisGeral(A, B)$	*A* é mais geral que todo descendente de *B* *B* é menos geral que todo ancestral de *A* *A* é sobreposto a todo ancestral de *B*
$menosGeral(A, B)$	*B* é mais geral que todo descendente de *A* *A* é menos geral que todo ancestral de *B* *B* é sobreposto a todo ancestral de *A*

$sobreposto(A, B)$	*A* e todo ancestral de *A* são sobrepostos a todo ancestral de *B*. *B* e todo ancestral de *B* são sobrepostos a todo ancestral de *A*.
$diferente(A, B)$	*A* e *B* não possuem descendentes em comum.

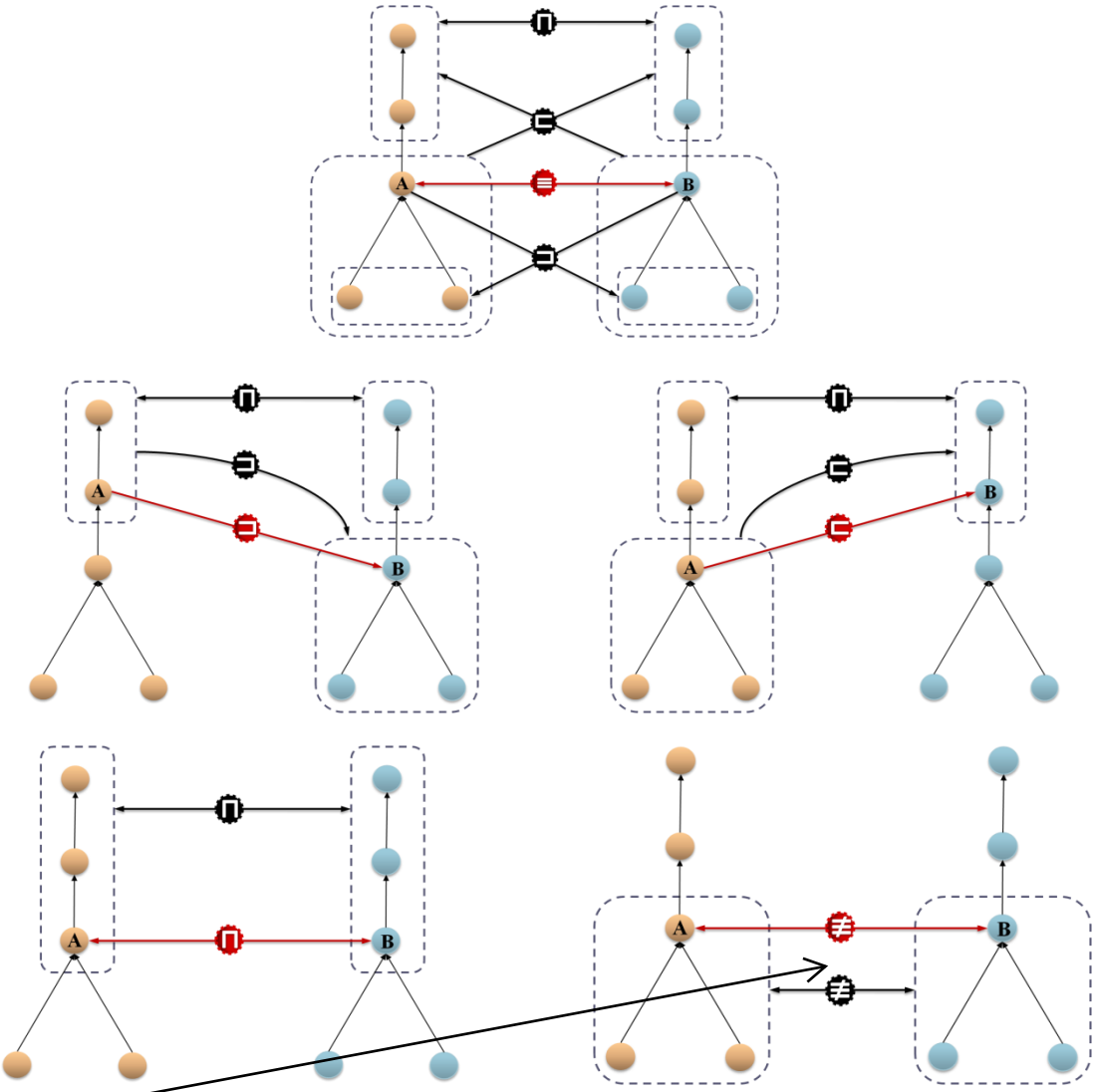


Figura 5-3: Mapeamentos pretos são inferidos a partir dos mapeamentos vermelhos

Quando impresso em laser mono, esses sinais ficam idênticos

As regras acima são implementadas numa **Tabela de Mapeamentos** que é o produto cartesiano $Conceitos(O_1) \times Conceitos(O_2)$, onde O_1 e O_2 são ontologias. Para a avaliação, são criadas duas tabelas: uma com mapeamentos ideais e outra com mapeamentos computados. Particularmente, a tabela ideal é um complemento à abordagem de mapeamento semântico discutida na Seção 4.3. As posições destas tabelas são então comparadas na

expectativa de que os mapeamentos atribuídos para ambas sejam iguais, indicando um acerto. Os acertos são então contabilizados para cada axioma de mapeamento, por isso cada axioma recebe um valor de precisão, revocação e medida-F.

O Módulo de Similaridade do L-*Match* também é avaliado com base na tabela ideal. Como os resultados deste módulo são valores de sobreposição e similaridade (e não axiomas de mapeamento), não há como utilizar a tabela computada. Contudo traçamos uma estratégia de avaliação: a idéia é observar a saída do Módulo de Similaridade e mensurar sua utilidade para o cálculo dos mapeamentos semânticos.

Após a propagação *bottom-up*, os conceitos de treino são ranqueados pela ordem decrescentes dos valores de similaridade e sobreposição com cada conceito de teste (veja a Figura 4-11). Utilizamos um **limiar relativo** para "limpar" o *rank* de similaridade, isto é, remover conceitos pouco similares; este limiar mantém no *rank* apenas conceitos com similaridade maior ou igual a *x*% da similaridade observada na primeira posição do *rank*. Para os experimentos, utilizamos sempre um limiar relativo a 70%.

É fácil avaliar o *rank* de sobreposição: basta manter conceitos com sobreposição maior que zero, o que automaticamente inclui superconceitos. Mas se um conceito é mantido no *rank* de similaridade, não significa que seus superconceitos também serão e vice-versa. Portanto, a taxonomia de treino não sofre propagação *bottom-up* antes da avaliação do *rank* de similaridade, de forma que este *rank* contenha apenas conceitos concretos (diretamente instanciados). Em seguida, o *rank* é expandido e passa a incluir outros conceitos, seguindo as regras da Tabela 5-2 em função do mapeamento ideal mais forte esperado para o conceito de teste correspondente.

As figuras seguintes exemplificam a avaliação de sobreposição e similaridade, mostrando conceitos de teste na primeira coluna, seguidos pela precisão (P), revocação (R),

medida-F (F) e pelo *rank* com classes de treino; os mapeamentos idealmente esperados estão indicados ao lado dos conceitos de treino, a cardinalidade dos conceitos de teste aparecem entre parênteses e os *ranks* de similaridade incluem apenas conceitos concretos:

Root: Taxonomy Micro-Organism	P	R	F	Rank	
|0|--Micro-Organism(0)	1,00	1,00	1,00	0,918 ⊃ Bacterial	
|1|--Bacteria(2)	1,00	1,00	1,00	1,000 ≡ Bacterial	
|1|--Virus(0)	?.??	?.??	?.??		
Root: Taxonomy Person	**P**	**R**	**F**	**Rank**	
|0|--Person(2)	1,00	1,00	1,00	1,000 ≡ Researcher	
Root: Taxonomy Condition	**P**	**R**	**F**	**Rank**	
|0|--Condition(0)	1,00	1,00	1,00	0,852 ⊃ BacterialInfection	0,842 ⊃ FoodBorneDisease
|1|--Infection(0)	1,00	1,00	1,00	0,852 ⊃ BacterialInfection	0,842 ⊃ FoodBorneDisease
|2|--ViralInfection(0)	?.??	?.??	?.??		
|2|--BacterialInfection(1)	1,00	1,00	1,00	0,918 ≡ BacterialInfection	0,907 ⊃ FoodBorneDisease
|3|--FoodPoisoning(3)	1,00	1,00	1,00	0,986 ≡ FoodBorneDisease	
Global Evaluation	**1,00**	**1,00**	**1,00**		

Figura 5-4: Avaliação dos *ranks* de similaridade para ontologias de doenças

Root: Taxonomy Micro-Organism	P	R	F	Rank			
|0|--Micro-Organism(0)	1,00	1,00	1,00	6 ⊂ Pathogen	6 ⊃ Bacterial		
|1|--Bacteria(2)	1,00	1,00	1,00	6 ⊂ Pathogen	6 ≡ Bacterial		
|1|--Virus(0)	?.??	?.??	?.??				
Root: Taxonomy Person	**P**	**R**	**F**	**Rank**			
|0|--Person(2)	1,00	1,00	1,00	4 ≡ Researcher			
Root: Taxonomy Condition	**P**	**R**	**F**	**Rank**			
|0|--Condition(0)	1,00	1,00	1,00	10 ≡ Condition	10 ⊃ Infection	10 ⊃ BacterialInfection	6 ⊃ FoodBorneDisease
|1|--Infection(0)	1,00	1,00	1,00	10 ⊂ Condition	10 ≡ Infection	10 ⊃ BacterialInfection	6 ⊃ FoodBorneDisease
|2|--ViralInfection(0)	?.??	?.??	?.??				
|2|--BacterialInfection(1)	1,00	1,00	1,00	10 ⊂ Condition	10 ⊂ Infection	10 ≡ BacterialInfection	6 ⊃ FoodBorneDisease
|3|--FoodPoisoning(3)	1,00	1,00	1,00	6 ⊂ Condition	6 ⊂ Infection	6 ⊂ BacterialInfection	5 ≡ FoodBorneDisease
Global Evaluation	**1,00**	**1,00**	**1,00**				

Figura 5-5: Avaliação dos *ranks* de sobreposição para ontologias de doenças

Sendo assim, espera-se que apenas conceitos relevantes permaneçam após estas "limpezas" nos *ranks*. A avaliação global (*Global Evaluation*) do Módulo de Similaridade não é a macro-avaliação (média das avaliações locais), e sim a micro-avaliação calculada em função do somatório de acertos, conceitos ideais e conceitos computados em cada *rank*.

5.3 RESULTADOS

Esta seção apresenta os resultados dos experimentos enfatizando a eficácia do método. Os seguintes limiares de relaxamento foram fixados para todos os pares de ontologias utilizadas:

Esse "alfa" foi mencionado ou explicado anteriormente?

Tabela 5-3: Limiares de Relaxamento

$T_{max\_overlap}$	0.85
T_{max_sim}	0.90
α	0.10
$T_{min\_overlap}$	0.30

Além destes, foi fixado um limiar relativo a 70% para avaliar os valores de similaridade. Nas subseções a seguir veremos os resultados dos experimentos com as ontologias Ecolingua e Apes e também com outras ontologias.

5.3.1 Avaliação Geral do Mapeamento Ecolíngua × Apes

"necessária" ou "indicada"

Iniciamos os experimentos sem realimentar o L-*Match*, pois queremos uma idéia inicial sobre o comportamento dos módulos de Similaridade e de Mapeamento e também dos algoritmos de classificação. A avaliação do Módulo de Similaridade é ~~valida~~ porque sua eficácia influencia diretamente a do Módulo de Mapeamento. As conclusões tiradas com Ecolingua e Apes serão então usadas para direcionar os experimentos com outros pares de ontologias.

"foram"

"Tabela 5-4"

"... por cada algoritmo ..."

A ~~tabela seguinte~~ apresenta uma avaliação geral da primeira iteração de mapeamento por algoritmo de classificação utilizado junto ao L-*Match*, a fim de compará-los. Incluímos a micro-avaliacão dos valores combinados de similaridade e sobreposição, a macro-avaliação do mapeamento e a mensuração de tempo em segundos. Em seguida é plotado um gráfico a partir da macro-avaliação do mapeamento e da mensuração de tempo:

Tabela 5-4: Avaliação Geral do Mapeamento Ecolingua×Apes

CLASSIFICADOR	*SOBREPOSIÇÃO*			*SIMILARIDADE*			*MAPEAMENTO*			TEMPO
	P	R	F	P	R	F	P	R	F	
k-NN	0.38	0.64	0.48	0.80	0.41	0.54	0.60	0.47	0.53	6.79s
Naive Bayes	**0.61**	0.76	**0.68**	0.73	0.17	0.27	0.80	0.63	0.70	19.83s
NB-Shrinkage	0.48	**0.77**	0.59	**0.90**	**0.61**	**0.73**	**0.82**	**0.76**	**0.79**	19.19s
SVM	0.38	0.75	0.50	0.62	0.53	0.57	0.40	0.44	0.42	20.22s
Máxima Entropia	0.42	0.70	0.53	0.71	0.57	0.63	0.68	0.66	0.67	20.40s
TF-IDF	0.34	0.68	0.46	0.65	0.47	0.55	0.60	0.59	0.60	**5.84s**

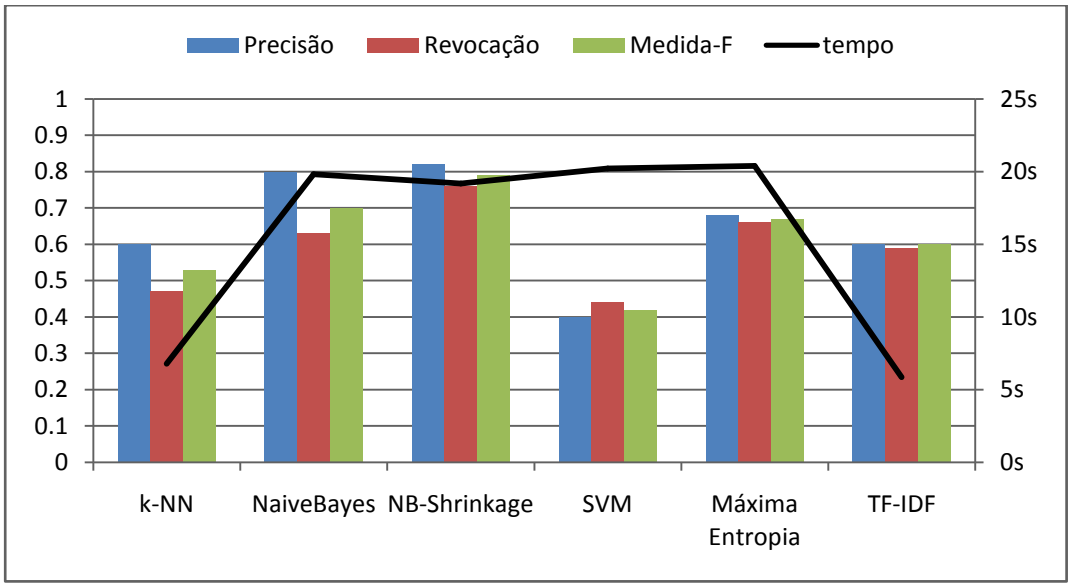


Figura 5-6: Gráfico da Avaliação por Classificador do Mapeamento Ecolingua×Apes

Em geral, os resultados foram bastante equilibrados, mas a vantagem qualitativa é do NB-*Shrinkage*, desde a avaliação de sobreposição e similaridade, consolidando-se na avaliação do mapeamento com valores próximos a 80%. Bons resultados também foram obtidos por *Naive Bayes* e Máxima Entropia, a avaliação dos mapeamentos ultrapassa 50% em quase todos os casos e a maior velocidade foi obtida por TF-IDF, ~~o que não é vantagem~~, pois mapeamentos muito melhores foram obtidos em menos de apenas 20 segundos pelo NB-*Shrinkage*.

Explicações "defensivas" são, na maioria das vezes, desnecessárias... altere este texto pra algo como: "... obtida por TF-IDF, embora mapeamentos muito melhores terem sido obtidos em menos de 20 segundos, especialmente pelo NB-Shrinkage."

"como era esperado"

Observamos também que a avaliação de sobreposição não foi tão notória quanto a de similaridade, o que não prejudicou gravemente o mapeamento e nos leva a concluir que a técnica de mapeamento guiada pela taxonomia contornou muitos dos erros de classificação, com era esperado. Dada a boa avaliação da similaridade computada entre conceitos, podemos supor também que, ignorando valores de sobreposição computados entre conceitos pouco similares, podemos futuramente melhorar a precisão dos valores de sobreposição e, conseqüentemente, do mapeamento.

5.3.2 Avaliação do Módulo de Similaridade para Ecolíngua × Apes

As tabelas seguintes exibem nas colunas $A \longrightarrow B$ e $B \longrightarrow A$ a avaliação da sobreposição e similaridade parciais (com noção de direção) entre conceitos, re-exibindo na coluna $A \longleftrightarrow B$ os valores combinados da Tabela 5-4 (sem noção de direção). Seja então o mapeamento Ecolingua× Apes:

Tabela 5-5: Avaliação da Sobreposição Ecolingua×Apes por Classificador

CLASSIFICADOR	$A \longrightarrow B$			$B \longrightarrow A$			$A \longleftrightarrow B$		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
k-NN	0.66	0.54	0.59	0.25	0.32	0.28	0.38	0.64	0.48
Naive Bayes	0.69	0.47	0.56	**0.57**	0.51	**0.54**	**0.61**	0.76	**0.68**
NB-Shrinkage	**0.81**	**0.68**	**0.74**	0.43	**0.69**	0.53	0.48	**0.77**	0.59
SVM	0.68	0.61	0.64	0.38	0.64	0.48	0.38	0.75	0.50
Máxima Entropia	0.64	0.62	0.63	0.38	0.57	0.46	0.42	0.70	0.53
TF-IDF	0.64	0.55	0.59	0.30	0.49	0.37	0.34	0.68	0.46

Tabela 5-6: Avaliação da Similaridade Ecolingua×Apes por Classificador

CLASSIFICADOR	$A \longrightarrow B$			$B \longrightarrow A$			$A \longleftrightarrow B$		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
k-NN	0.66	0.59	0.62	0.35	0.50	0.41	0.80	0.41	0.54
Naive Bayes	0.69	0.47	0.56	**0.61**	0.51	0.55	0.73	0.17	0.27
NB-Shrinkage	**0.81**	0.61	**0.70**	0.50	**0.71**	**0.58**	**0.90**	**0.61**	**0.73**
SVM	0.58	**0.66**	0.62	0.37	0.67	0.47	0.62	0.53	0.57
Máxima Entropia	0.72	0.58	0.64	0.46	0.63	0.54	0.71	0.57	0.63
TF-IDF	0.65	0.62	0.63	0.34	0.58	0.43	0.65	0.47	0.55

Os resultados nas colunas $A \longrightarrow B$ são muito melhores que aqueles da coluna $B \longrightarrow A$, tendo o NB-*Shrinkage* alcançado até 81% de precisão e cerca de 70% de medida-F. Isso significa que houve maior dificuldade de treinar na Ecolíngua no momento de classificar as instâncias da Apes, talvez porque a Ecolíngua possui menos instâncias: apenas 3.88 instâncias por conceito concreto, contra 5.48 na Apes. De qualquer forma, esta diferença é natural, pois também foi observada em outros pares de ontologias.

Os melhores valores se concentram entre *Naive Bayes* e NB-*Shrinkage*. Apesar de não ter sido unânime, a vantagem é do NB-*Shrinkage*, pois é sempre superior ou próximo ao *Naive Bayes*. Esta superioridade é mais clara na avaliação de similaridade do que na avaliação de sobreposição. Isso ocorre porque o cálculo de similaridade entre conceitos é mais abrangente, pois não se resume à primeira posição do *rank* retornado pelos classificadores para cada instância: Por outro lado, o cálculo da sobreposição espera que o conceito correto de toda instância venha na primeira posição deste *rank* (utilizamos um limiar *Top-k*=1), o que é provável mas não garantido, pois às vezes pode ocorrer em outras posições iniciais. Contudo, a investigação de valores maiores para *Top-k* não trouxeram melhoria.

Isso mostra também que, ao comparar conceitos em função de instâncias, é melhor utilizar a similaridade computada pelos algoritmos de classificação, os quais capturam nuances maiores, do que utilizar a similaridade de *Jaccard* em função dos valores absolutos de sobreposição.

5.3.3 Avaliação do Módulo de Mapeamento para Ecolíngua × Apes

Esta subseção detalha a avaliação mais importante que é a dos mapeamentos. Apresentaremos a avaliação por relação (axioma ponte) e realimentaremos o L-*Match* com

sua própria saída para mostrar que a abordagem iterativa é capaz de melhorar a qualidade do mapeamento quando associada ao NB-*Shrinkage*, classificador utilizado para guiar os experimentos devido ao seu bom desempenho, como vimos anteriormente.

Basicamente, a avaliação do mapeamento será tabelada seguindo o modelo abaixo. Os tamanhos dos conjuntos Ideal, Computado e Acerto sao exibidos por relação, bem como precisão, revocação e medida-F, que chamaremos de medidas parciais. A linha de totais sumariza os tamanhos dos conjuntos e apresenta as medidas de macro-precisão, macro-revocação e macro-medida-F, que são as médias das parciais:

Tabela 5-7: Modelo de Avaliação de Mapeamento

RELAÇÃO	IDEAL	COMPUTADO	ACERTO	PRECISÃO	REVOCAÇÃO	MEDIDA-F
relação $_1$	i_1	c_1	a_1	P_1	R_1	F_1
...	...	...	...	...	...	...
relação $_N$	i_N	c_N	a_N	P_N	R_N	F_N
Total	$\sum i_k$	$\sum c_k$	$\sum a_k$	$\mathbf{P_{MACRO}}$	$\mathbf{R_{MACRO}}$	$\mathbf{F_{MACRO}}$

Utilizamos a macro-avaliação porque, neste caso, a micro-avaliação iguala precisão, revocação e medida-F; logo, obeteríamos sempre o *accuracy*. Isso acontece porque os conjuntos Ideal e Computado tem o mesmo tamanho, isto é, $\sum i_k = \sum c_k$.

Fizemos apenas três iterações de mapeamento, pois observamos um fenômeno interessante: após um pequeno número de iterações, o mapeamento tende a convergir para um resultado que não se altera mais ou então se altera muito pouco. No caso do mapeamento Ecolingua×Apes utilizando NB-*Shrinkage*, os resultados se tornam constantes a partir da terceira iteração.

A tabela seguinte detalha a avaliação do mapeamento da 1° iteração de mapeamento entre as ontologias Ecolingua e Apes. Observe que as macro-medidas são as mesmas

apresentadas na Tabela 5-4 para o NB-*Shrinkage*. Em seguida, as medidas parciais são utilizadas para plotar o gráfico da Figura 5-7, veja:

Tabela 5-8: Avaliação da 1° Iteração de Mapeamento Ecolingua×Apes com NB-*Shrinkage*

RELAÇÃO	IDEAL	COMPUTADO	ACERTO	PRECISÃO	REVOCAÇÃO	MEDIDA-F
Diferença	9756	9801	9736	0.99	1.00	1.00
Sobreposição	38	46	18	0.39	0.47	0.43
Menos Geral	157	154	147	0.95	0.94	0.95
Mais Geral	240	193	180	0.93	0.75	0.83
Equivalência	14	11	9	0.82	0.64	0.72
Total	**10205**	**10205**	**10090**	**0.82**	**0.76**	**0.79**

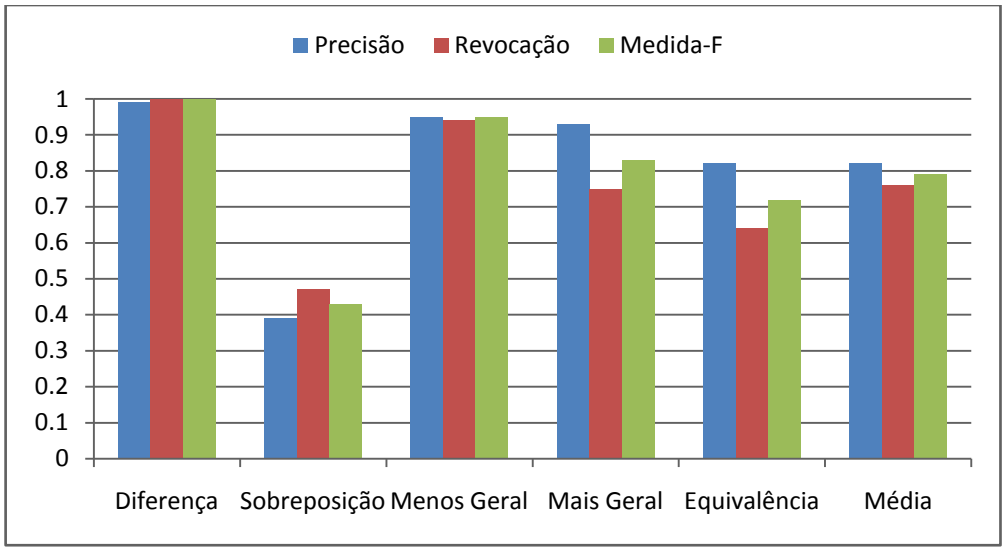


Figura 5-7: Gráfico da 1° Iteração de Mapeamento Ecolingua×Apes com NB-*Shrinkage*

Não houve problema para identificar diferenças, o que demonstra a capacidade da estratégia *Top-Down* de não associar conceitos extremamente diferentes. Contudo, muitos erros recaem sobre a sobreposição, que teve a pior avaliação: sua baixa precisão indica que relações mais fortes identificadas incorretamente acabaram sendo computadas, no mínimo, como sobreposição, por erro da abordagem ou por falta de instâncias. Porém, a sobreposição é a relação que menos importa, já que praticamente não tem utilidade.

Explicação desnecessária - simplesmente retire.

Como desejávamos, depois da diferença, os melhores valores foram obtidos para as relações mais importantes (equivalência, mais geral e menos geral), especialmente em termos de precisão e medida-F. Isso significa que tivemos sucesso ao descobrir relações de herança e também desempenhamos bem ao descobrir equivalências com alta precisão (82%) e revocação mais baixa a 64%, que pelo menos está acima de 50%.

"... que está acima de 50%, estipulado como mínimo aceitável."

De maneira geral, relações de herança também foram descobertas com boa precisão nos mapeamentos semânticos de (BOUQUET, *at al.*, 2003) e (MAGNINI, *at al.*, 2004), trabalhos que avaliaram o Ctx-*Match*. Contudo, a medida-F do Ctx-*Match* mostra grande desequilíbrio, devido à revocações muito mais baixas que precisões. Além disso, o Ctx-*Match* teve a pior avaliação para a equivalência, onde o desequilíbrio é gritante, hora com baixa precisão (33%) e revocação (4%), hora com alta precisão (78%) e baixa revocação (13%). Por sua vez, o S-*Match* não apresentou a avaliação por relação e, apesar de aparentemente interessante, suas avaliações foram feitas sobre esquemas pequenos, sem grandes diferenças estruturais e terminológicas, portanto fáceis de mapear.

"... portanto menos difíceis de mapear."

Agora apresentaremos a avaliação da estratégia iterativa do L-*Match*: certamente obtivemos bons valores para a equivalência com o L-*Match*, mas isso precisa melhorar já que esta relação é a mais importante. Para isso realimentamos o L-*Match* com seus próprios mapeamentos e avaliamos a segunda iteração de mapeamentos:

Tabela 5-9: Avaliação da 2° Iteração de Mapeamento Ecolingua×Apes com NB-*Shrinkage*

RELAÇÃO	IDEAL	COMPUTADO	ACERTO	PRECISÃO	REVOCAÇÃO	MEDIDA-F
Diferença	9756	9764	9717	1.00	1.00	1.00
Sobreposição	38	21	16	0.76	0.42	0.54
Menos Geral	157	145	137	0.94	0.87	0.91
Mais Geral	240	263	223	0.85	0.93	0.89
Equivalência	14	12	11	0.92	0.79	0.85
Total	**10205**	**10205**	**10104**	**0.89**	**0.80**	**0.84**

O total de acertos foi maior que o da iteração anterior (+14), com ganhos ocorrendo em todas as relações, com pequenas perdas na precisão de mais geral (-8%) e na revocação de menos geral (-7%) e da sobreposição (-5%). Os ganhos para equivalência são evidentes (+10% de precisão e +15% de revocação), bem como o ganho na precisão da sobreposição (+37%). Há também queda no número de falsas diferenças. Estes ganhos elevam a avaliação total para a casa dos 80%, com precisão de quase 90%. Estes resultados melhoraram ainda mais após a terceira iteração, quando então convergiram, veja:

Tabela 5-10: Avaliação da 3° Iteração de Mapeamento Ecolingua×Apes com NB-*Shrinkage*

RELAÇÃO	IDEAL	COMPUTADO	ACERTO	PRECISÃO	REVOCAÇÃO	MEDIDA-F
Diferença	9756	9762	9717	1.00	1.00	1.00
Sobreposição	38	19	16	0.84	0.42	0.56
Menos Geral	157	143	137	0.96	0.87	0.91
Mais Geral	240	266	227	0.85	0.95	0.90
Equivalência	14	15	14	0.93	1.00	0.97
Total	**10205**	**10205**	**10111**	**0.92**	**0.85**	**0.88**

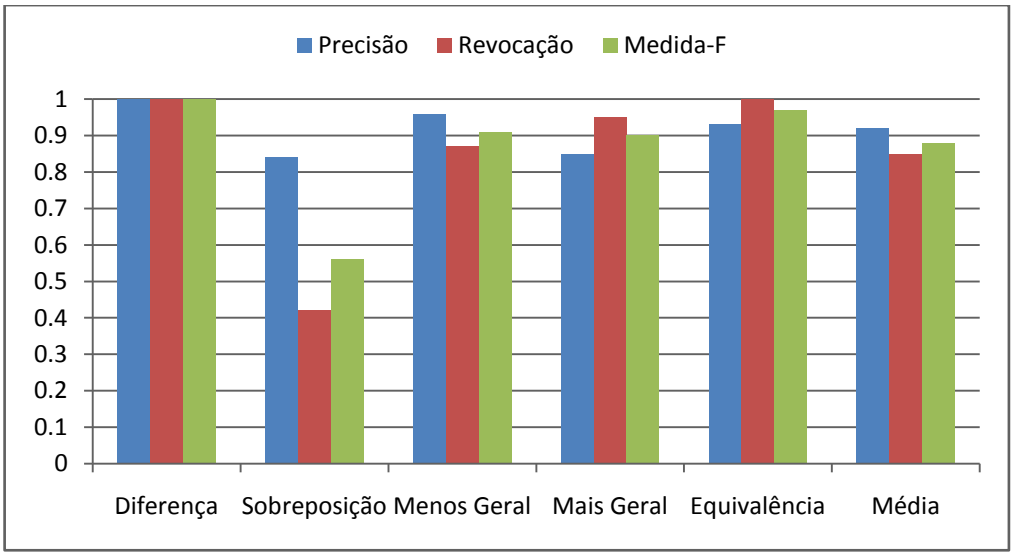


Figura 5-8: Gráfico da 3° Iteração de Mapeamento Ecolingua×Apes com NB-Shrinkage

O aumento de acertos (+5) nesta iteração não é o grande responsável pelos ganhos, mas sim a distribuição dos acertos: ganhos ocorreram em todas as relações, sem nenhuma

"A partir desses resultados, ..."

perda de valor; exceto pela revocação e medida-F da sobreposição, todas as outras medidas estão acima de 83%, algumas ultrapassando 90%. O resultado mais satisfatório foi o da equivalência, que alcançou revocação máxima (100%) e altíssima precisão (93%), o que surpreende porque Ecolíngua e Apes possuem mapeamentos bastante subjetivos.

Portanto, concluímos que a abordagem iterativa foi muito importante porque propiciou ganhos muito satisfatórios. Contudo, estes resultados foram obtidos utilizando apenas o NB-*Shrinkage*, por isso experimentamos fazer o mesmo com outros algoritmos de classificação. As duas tabelas seguintes são equivalente e apresentam resultados por classificador em cada iteração, mas a primeira apresenta a macro-avaliação e a segunda apresenta os ganhos/perdas de avaliação em relação à primeira iteração:

"Cabe destacar que os resultados apresentad os acima ..."

Acho que aqui poderia haver uma separação maior, talvez até aprofundando mais o nível de subseção, algo como "5.3.3.1 Avaliando a opção pelo SB-Shrinkage"

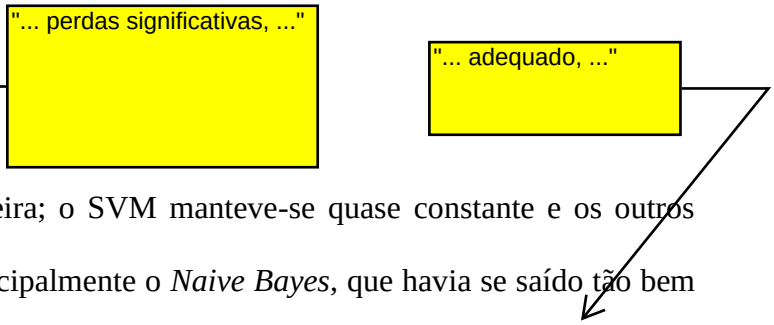
Tabela 5-11: Avaliação por Classificador do Mapeamento Iterativo Ecolingua×Apes

CLASSIFICADOR	*1° Iteração*			*2° Iteração*			*3° Iteração*		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
k-NN	0.60	0.47	0.53	0.57	0.37	0.45	0.53	0.40	0.45
Naive Bayes	0.80	0.63	0.70	0.75	0.46	0.57	0.62	0.42	0.50
NB-Shrinkage	**0.82**	**0.76**	**0.79**	**0.89**	**0.80**	**0.84**	**0.92**	**0.85**	**0.88**
SVM	0.40	0.44	0.42	0.43	0.43	0.43	0.39	0.45	0.42
Máxima Entropia	0.68	0.66	0.67	0.70	0.72	0.71	0.57	0.47	0.51
TF-IDF	0.60	0.59	0.60	0.52	0.63	0.57	0.47	0.59	0.52

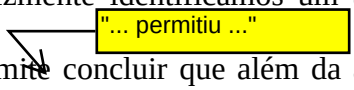
Tabela 5-12: Ganho por Classificador do Mapeamento Iterativo Ecolingua×Apes

CLASSIFICADOR	*1° Iteração*			*2° Iteração*			*3° Iteração*		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
k-NN	0.60	0.47	0.53	-3%	-10%	-8%	-7%	-7%	-8%
Naive Bayes	0.80	0.63	0.70	-5%	-17%	-13%	-18%	-21%	-20%
NB-Shrinkage	**0.82**	**0.76**	**0.79**	**+7%**	**+4%**	**+5%**	**+20%**	**+9%**	**+9%**
SVM	0.40	0.44	0.42	+3%	-1%	+1%	-1%	+1%	0%
Máxima Entropia	0.68	0.66	0.67	+2%	+5%	+4%	-11%	-19%	-16%
TF-IDF	0.60	0.59	0.60	-8%	**+4%**	-3%	-13%	0%	-8%

Infelizmente, ganhos não são garantidos quando utilizamos outro classificador individual senão o NB-*Shrinkage*: o Máxima Entropia até obteve ganhos na segunda iteração,



mas logo teve grandes perdas na terceira; o SVM manteve-se quase constante e os outros algoritmos obtiveram ~~boas perdas~~, principalmente o *Naive Bayes*, que havia se saído tão bem na primeira iteração. Felizmente identificamos um algoritmo de classificação ~~ideal~~, o NB-*Shrinkage*, o que nos permite concluir que além da abordagem iterativa de mapeamento, o algoritmo de classificação utilizado também é importante.



Repetimos estes experimentos com algumas combinações (ensembles) do NB-*Shrinkage* com outros classificadores que, na primeira iteração, deram mostra que poderiam funcionar:

Tabela 5-13: Avaliação por Ensemble do Mapeamento Iterativo Ecolingua×Apes

CLASSIFICADOR	*1° Iteração*			*2° Iteração*			*3° Iteração*		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
NB-Shrinkage	0.82	0.76	**0.79**	**0.89**	0.80	**0.84**	**0.92**	**0.85**	**0.88**
NB-Shrinkage+k-NN	0.83	0.75	**0.79**	0.86	0.74	0.80	0.84	0.73	0.78
NB-Shrinkage+NaiveBayes	**0.85**	0.74	**0.79**	0.86	0.80	0.83	0.88	0.78	0.83
NB-Shrinkage+SVM	0.71	**0.78**	0.75	0.79	**0.85**	0.82	0.71	0.76	0.73
NB-Shrinkage+MaxEnt	0.75	0.76	0.76	0.79	0.78	0.78	0.72	0.70	0.71
NB-Shrinkage+TF-IDF	0.73	0.73	0.73	0.76	0.80	0.78	0.79	0.84	0.82
k-NN+Naive Bayes	0.82	0.68	0.74	0.83	0.60	0.69	0.81	0.46	0.58

Tabela 5-14: Ganho por Ensemble do Mapeamento Iterativo Ecolingua×Apes

CLASSIFICADOR	*1° Iteração*			*2° Iteração*			*3° Iteração*		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
NB-Shrinkage	0.82	0.76	**0.79**	+7%	+4%	+5%	**+20%**	+9%	**+9%**
NB-Shrinkage+k-NN	0.83	0.75	**0.79**	+3%	-1%	+1%	+1%	-2%	-1%
NB-Shrinkage+NaiveBayes	**0.85**	0.74	**0.79**	+1%	+6%	+4%	+3%	+4%	+4%
NB-Shrinkage+SVM	0.71	**0.78**	0.75	**+8%**	**+7%**	**+7%**	0%	-2%	-2%
NB-Shrinkage+MaxEnt	0.75	0.76	0.76	+4%	+2%	+2%	-2%	-6%	-5%
NB-Shrinkage+TF-IDF	0.73	0.73	0.73	+3%	**+7%**	+5%	+6%	**+11%**	**+9%**
k-NN+Naive Bayes	0.82	0.68	0.74	+1%	-8%	-5%	-1%	-22%	-16%

Nenhuma das combinações encontrou resultados melhores do que o NB-*Shrinkage* utilizado individualmente, porém as quedas de valor durante as iterações foram menores, exceto na última combinação, a qual não envolveu o NB-*Shrinkage*. Apesar dos ganhos na segunda

iteração, as combinações com Máxima Entropia e SVM logo entraram em declínio; por outro lado as combinações com *Naive Bayes* e principalmente com TF-IDF também obtiveram ganhos consecutivos.

"... consecutivos."

5.3.4 Avaliação do Módulo de Mapeamento para Outros Pares de Ontologias

"Após concluirmos que o NB-Shrinkage é, de fato, o classificador mais adequado, experimentamos outras ontologias no L-Match. Iniciamos ..."

~~Experimentaremos outras ontologias no L-*Match* baseado apenas no NB-*Shrinkage*, pois concluímos que é o mais adequado~~. Iniciamos mapeando com duas iterações as ontologias sobre doenças, que são fáceis de mapear e se beneficiaram bastante do alinhamento vertical:

Tabela 5-15: Avaliação da 1° Iteração de Mapeamento Disease 1 e 2 com NB-Shrinkage

RELAÇÃO	IDEAL	COMPUTADO	ACERTO	PRECISÃO	REVOCAÇÃO	MEDIDA-F
Diferença	39	38	38	1.00	0.97	0.99
Sobreposição	0	0	0	1.00	1.00	1.00
Menos Geral	11	14	11	0.79	1.00	0.88
Mais Geral	7	6	6	1.00	0.86	0.92
Equivalência	6	5	5	1.00	0.83	0.91
Total	**63**	**63**	**60**	**0.96**	**0.93**	**0.94**

Tabela 5-16: Avaliação da 2° Iteração de Mapeamento Disease 1 e 2 com NB-*Shrinkage*

RELAÇÃO	IDEAL	COMPUTADO	ACERTO	PRECISÃO	REVOCAÇÃO	MEDIDA-F
Diferença	39	40	39	0.98	1.00	0.99
Sobreposição	0	1	0	0.00	1.00	0.00
Menos Geral	11	9	9	1.00	0.82	0.90
Mais Geral	7	7	7	1.00	1.00	1.00
Equivalência	6	6	6	1.00	1.00	1.00
Total	**63**	**63**	**61**	**0.80**	**0.96**	**0.87**

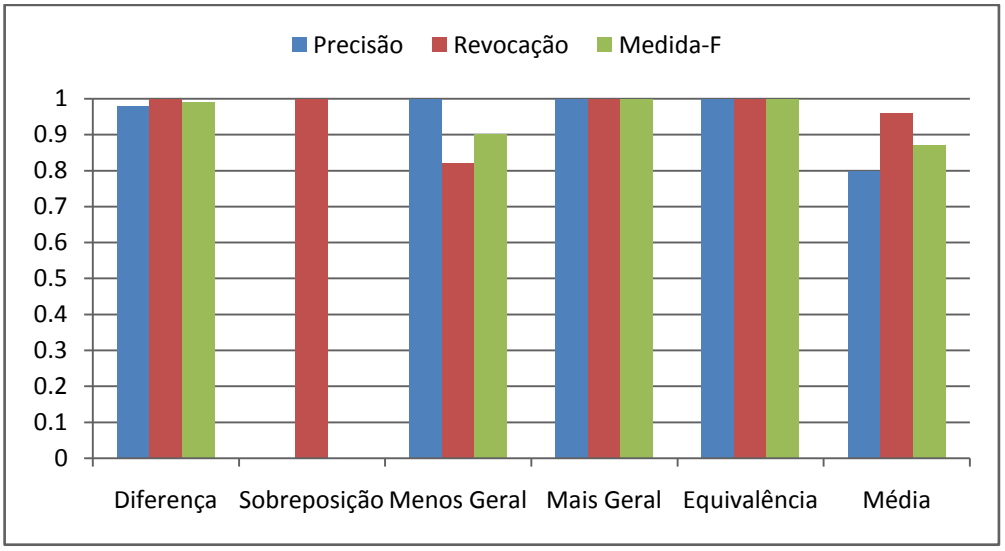


Figura 5-9: Gráfico da 2° Iteração de Mapeamento Disease1×Disease2 com NB-*Shrinkage*

A primeira iteração teve resultados muito bons, mas outros ainda melhores foram alcançados novamente na segunda iteração: equivalência e mais geral alcançaram avaliação máxima e a medida-F de menos geral aumentou. As quedas na macro-avaliação são enganosas: um único falso-positivo zerou a precisão da sobreposição, influenciando exageradamente a macro-precisão, pois a média é uma função estatística tendenciosa a altos e baixos valores. ~~De qualquer forma, a sobreposição é a relação menos importante.~~

"Além disso, a sobreposição é a relação que agrega menos informação."

"... bastante afetada por ..."

O próximo par de ontologias a mapear corresponde à amostra retirada dos catálogos de cursos das universidades de Cornell e de Washington. Os resultados foram interessantes uma vez que todas as avaliações atingiram valor máximo logo na primeira iteração, veja:

Tabela 5-17: Avaliação da 1° Iteração de Mapeamento Cornell×Washington com NB-*Shrinkage*

RELAÇÃO	IDEAL	COMPUTADO	ACERTO	PRECISÃO	REVOCAÇÃO	MEDIDA-F
Diferença	506	506	506	1.00	1.00	1.00
Sobreposição	1	1	1	1.00	1.00	1.00
Menos Geral	49	49	49	1.00	1.00	1.00
Mais Geral	74	74	74	1.00	1.00	1.00
Equivalência	21	21	21	1.00	1.00	1.00
Total	**651**	**651**	**651**	**1.00**	**1.00**	**1.00**

"... expressivos, ..."

Contudo, estes resultados não são tão ~~notórios~~, pois estes catálogos são parecidos. De qualquer forma, este mesmo experimento foi realizado pelo S-*Match* que obteve o mesmo resultado máximo, portanto o L-*Match* não deixou a desejar. O GLUE também realizou diferentes experimentos com estes catálogos e, apesar de ter alcançado alguns resultados acima de 90%, não obteve valores máximos como o S-*Match* e o L-*Match*.

Por último, mapeamos as ontologias sobre a Rússia. Neste caso, a abordagem iterativa não ajudou, pois os resultados convergiram logo na primeira iteração:

Tabela 5-18: Avaliação da 1° Iteração de Mapeamento Rússia 1 e 2 com NB-*Shrinkage*

RELAÇÃO	IDEAL	COMPUTADO	ACERTO	PRECISÃO	REVOCAÇÃO	MEDIDA-F
Diferença	22908	22719	22246	0.98	0.97	0.98
Sobreposição	101	108	21	0.19	0.21	0.20
Menos Geral	952	827	532	0.64	0.56	0.60
Mais Geral	460	769	389	0.51	0.85	0.63
Equivalência	41	39	23	0.59	0.56	0.58
Total	**24462**	**24462**	**23211**	**0.58**	**0.63**	**0.60**

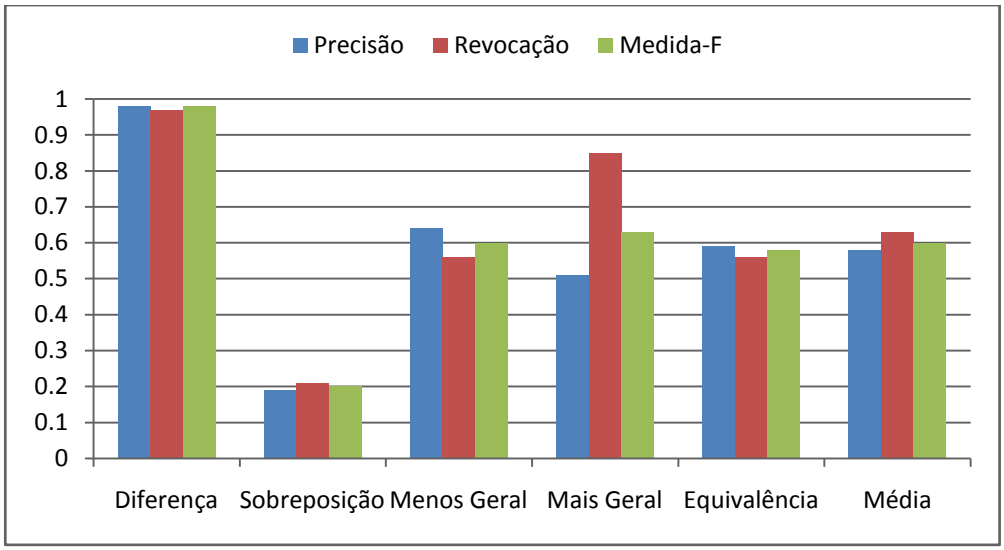


Figura 5-10: Gráfico da 1° Iteração de Mapeamento Rússia 1 e 2 com NB-*Shrinkage*

"... expressivos ..."

Apesar de não serem ~~notórios~~ como os anteriores, estes resultados foram muito bons, dado que superam em cerca de 20% os experimentos realizados durante a I3CON em 2004[15], que aplicou diferentes mapeadores sobre este mesmo par de ontologias, considerando os mesmos mapeamentos ideais, mas que ~~malmente~~ atingiram 40% na avaliação comparativa de media-f, contra 60% alcançado pelo L-*Match.* Além disso, estas ontologias sobre Rússia são conhecidas por serem realmente difíceis de mapear, pois apresentam muitas diferenças, principalmente estruturais.

"... cujos melhores desempenhos ..."

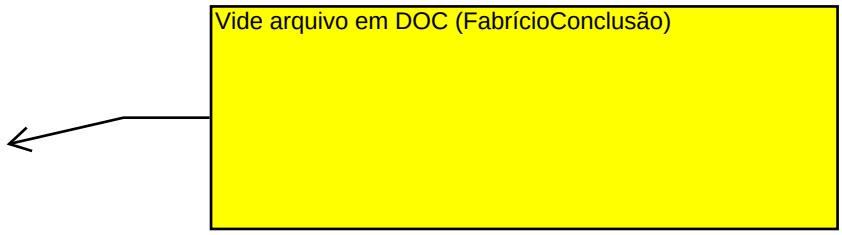
Combinando NB-*Shrinkage* com *Naive Bayes* no L-*Match*, conseguimos resultados de mapeamento um pouco melhores para Rússia 1 e 2, mas a diferença foi pouca, o que não ~~compensa~~ já que praticamente o mesmo pode ser obtido executando apenas um classificador:

"... compensaria mudanças de abordagem ..."

Tabela 5-19: Avaliação da 1° Iteração de Mapeamento Rússia 1 e 2 com NB-*Shrinkage+NaiveBayes*

RELAÇÃO	IDEAL	COMPUTADO	ACERTO	PRECISÃO	REVOCAÇÃO	MEDIDA-F
Diferença	22908	22719	22246	0.98	0.97	0.97
Sobreposição	101	108	21	0.23	0.26	0.24
Menos Geral	952	827	532	0.74	0.56	0.64
Mais Geral	460	769	389	0.45	0.86	0.59
Equivalência	41	39	23	0.60	0.61	0.60
Total	**24462**	**24462**	**23211**	**0.60**	**0.65**	**0.62**

[15] http://www.atl.external.lmco.com/projects/ontology/papers/I3CON-Results.pdf



Capítulo 6

CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho pesquisou a utilização de Aprendizado de Máquina para Integração de Informação em bases de dados heterogêneas, no caso, mapeamento semântico entre conceitos de ontologias distintas. Foram investigadas diferentes técnicas de classificação supervisionada adaptadas para criar uma nova abordagem de mapeamento semântico, cuja qualidade foi avaliada por meio de uma implementação preliminar chamada L-*Match* que explora conteúdo textual gerado a partir das propriedades das instâncias declaradas pelos autores das ontologias.

Dos bons resultados obtidos durante a avaliação dos experimentos, concluímos que a abordagem funciona muito bem, é bastante promissora e competitiva, que explorar instâncias no mapeamento é bom, que o NB-*Shrinkage* é o algoritmo de classificação que melhor funciona dentro da nossa abordagem, talvez pela intimidade entre *shrinkage* e hierarquias taxonômicas, e que a abordagem de mapeamento iterativo que reutiliza mapeamentos computados pelo próprio L-*Match* pode incrementar muito a qualidade dos resultados. Contudo, infelizmente não foi possível fazer avaliação comparativa com o S-*Match*, que é o único mapeador semântico além do L-*Match*, pois seus criadores não o disponibilizam para terceiros.

6.1 TRABALHOS FUTUROS

Apesar do bom desempenho na avaliação dos experimentos, o L-*Match* é apenas a peça inicial de uma solução mais completa, pois muitas melhorias futuras são aplicáveis para prover ganhos de precisão e de velocidade. Identificamos a seguir novas oportunidades espalhadas pelos três módulos do L-*Match*.

Melhorias no Módulo de Extração são bem vindas, pois dele dependem todos os outros módulos. Este é o momento no qual dados são extraídos de suas fontes e transformados (renderizados em texto, limpos, indexados, normalizados, etc.) numa informação que deve ser útil, precisa e confiável. Trabalhos futuros vão desde incrementos mais simples, como expansão de siglas e técnicas de casamento de *strings*, passando pela experimentação de outras técnicas de desambiguação e identificação de sinônimos, até a extensão da portabilidade do L-*Match* para trabalhar com diferentes formatos de ontologias. É interessante também Investigar a correlação entre palavras que modificam seus significados mutuamente (CHAKRAVARTHY, 1995), dependendo de suas classes gramaticais: por exemplo, na Ecolíngua os conceitos *Temperature* e *OfTemperature* são rotulados um por substantivo e outro por locução adjetiva, mas ao excluir a *stopword "of"* perdemos esta informação de diferença, levando a falsos positivos que comprometem a qualidade do texto.

Para o Módulo de Similaridade podemos adaptar o NB-*Shrinkage* para fazer classificação hierárquica e experimentar APIs de classificação mais atualizadas do que a *Rainbow*, como Weka[16] e LibSVM[17]. Podemos também aumentar o número de instâncias utilizando anotação automática ou, por outro lado, diminuir este número eliminando

[16] http://www.cs.waikato.ac.nz/ml/weka/
[17] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

instâncias ruins contendo ruídos que atrapalham a classificação (*noisy exemplar pruning*), mantendo apenas instâncias com classificação confiável. Por fim, será interessante podar os valores de sobreposição entre conceitos pouco similares.

Nó Módulo de Mapeamento é ainda necessário observar melhor o comportamento da estratégia comparativa *TopDown* afim de otimizá-la, estender o L-*Match* para mapear propriedades (relacionamentos e atributos), usar técnicas melhores para comparação de rótulos durante o alinhamento vertical, refinar a abordagem iterativa controlando o conhecimento que é reutilizado (ou não) e, por fim, associar o L-*Match* com outras técnicas de mapeamento como, por exemplo, submeter os mapeamentos do L-*Match* ao raciocínio proposicional implementado no S-*Match*, que é o real diferencial deste mapeador. Podemos também tentar delimitar conceitos através de suas propriedades para então compará-los, similar ao que o L-*Match* faz atualmente utilizando instâncias.

REFERÊNCIAS

ANHAI, Doan, *at al.* 2004. Ontology Matching: A Machine Learning Approach. In: *Handbook on Ontologies*. s.l. : Springer-Verlag, 2004, p. 385–404.

ATHANASIADIS, Ioannis, *at al.* 2006. Enriching Software Model Interfaces Using Ontology-Based Tools. In: *The iEMSs Third Biannual Meeting "Summit on Environmental Modelling and Software".* July 2006.

AUMUELLER, David, *at al.* 2005. Schema and Ontology Matching with COMA++. In: *ACM SIGMOD International Conference on Management of Data.* June 2005, pp. 906–908.

BAADER, Franz; HORROCKS, Ian e SATTLER, Ulrike. 2002. Description Logics as Ontology Languages for the Semantic Web. In: *The International Workshop on Ontologies.* 2002.

BOUQUET, Paolo, *at al.* 2003. A SAT-Based Algorithm for Context Matching. In: *CONTEXT2003.* 2003.

BOUQUET, Paolo, *at al.* 2003. C-OWL: Contextualizing Ontologies. In: *The 2nd International Semantic Web Conference (ISWC-2003).* 2003, Vol. 2870, pp. 164-179.

BOUQUET, Paolo; SERAFINI, Luciano e ZANOBINI, Stefano. 2003. Semantic Coordination: a New Approach and an Application. In: *The 2nd International Semantic Web Conference (ISWC'03).* October 2003.

BRAGA, Alessandra, *at al.* 2006. Comparação entre as Classificações Híbrida eSupervisionada no Mapeamento do Uso do Solo Usando Imagens de Alta Resolução. In: *Congresso Brasileiro de Cadastro Técnico Multifinalitário.* October 2006.

BRILHANTE, Virgínia. 2004. An Ontology for Quantities in Ecology. In: *The 17th Brazilian Symposium on Artificial Intelligence.* 2004, pp. 144-153.

BRON, Coen e KERBOSCH, Joep. 1973. Algorithm 457: Finding All Cliques of an Undirected Graph. In: *Communications of the ACM.* 1973, Vol. 16.

BUITELAAR, Paul; CIMIANO, Philipp e MAGNINI, Bernardo. 2005. *Ontology Learning from Text: Methods, Evaluation and Applications.* s.l. : IOS Press, 2005. Vol. 123 Frontiers in Artificial Intelligence.

BUITELAAR, Paul e SACALEANU, Bogdan. 2001. Ranking and Selecting Synsets by Domain Relevance. In: *WordNet and Other Lexical Resources: Applications, Extensions and Customizations. NAACL 2001 Workshop.* June 2001.

BURGES, Christopher. 1998. A Tutorial on Support Vector Machines for Pattern Recognition. In: *Data Mining and Knowledge Discovery.* 1998, pp. 121 - 167.

CALADO, Pável, *at al.* 2003. Combining Link-Based and Content-Based Methods for Web Document Classification. In: *Conference on Information and Knowledge Management (CIKM03).* 2003, pp. 394-401.

CHAKRABARTI, Soumen. 2002. *Mining the Web: Discovering Knowledge from Hypertext Data.* Bombay, India : Morgan-Kaufmann Publishers, 2002. ISBN 1-55860-754-4.

CHAKRAVARTHY, Anil. 1995. Sense Disambiguation Using Semantic Relations and Adjacency Information. In: *Meeting of the Association for Computational Linguistics.* 1995, pp. 293-295.

CHANDRASEKARAN, B.; JOSEPHSON, John e BENJAMINS, V.. 1999. What Are Ontologies, and Why Do We Need Them?. In: *IEEE Intelligent Systems.* January 1999, Vol. 14.

CHARNIAK, Eugene. 1991. Bayesians Networks without Tears. In: *IA Magazine.* 1991, Vol. 12, pp. 50-63.

CIMIANO, Philipp, *at al.* 2004. Learning Taxonomic Relations from Heterogeneous Evidence. In: *The Ontology Learning and Population Workshop, 16th European Conference on Artificial Intelligence (ECAI04).* 2004.

DING, Junyan; GRAVANO, Luis e SHIVAKUMAR, Narayanan. 2000. Computing Geographical Scopes of Web Resources. In: *26th International Conference on Very Large Databases (VLDB'2000).* 2000.

DO, Hong-Hai e RAHM, Erhard. 2002. COMA - A System for Flexible Combination of Schema Matching Approaches. In: *VLDB.* 2002, pp. 610-621.

DOAN, AnHai, *at al.* 2002. Learning to Map Between Ontologies on the Semantic Web. In: *The 11th International World Wide Web Conference.* May 2002.

DU, Ding-Zhu e PARDALOS, Panos. 2007. *Handbook of Combinatorial Optimization.* s.l. : Springer-Verlag, 2007. ISBN:0792359240.

EUZENAT, Jerome, *at al.* 2004. Ontology Alignment with OLA. In: *The International Semantic Web Conference Workshop on Evaluation of Ontology-Based Tools (EON'04).* 2004, pp. 59–68.

EUZENAT, Jérôme e VALTCHEV, Petko. 2004. An Algorithm and an Implementation of Semantic Matching. In: *The First European Semantic Web Symposium.* 2004.

---, 2003. An Integrative Proximity Measure for Ontology Alignment. In: *Semantic Integration Workshop, Second International Semantic Web Conference (ISWC-03).* 2003.

---, 2004. Similarity-based Ontology Alignment in OWL-Lite. In: *European Conference on Artificial Intelligence (ECAI-04).* 2004, pp. 333–337.

FELLBAUM, Christiane. 1998. WordNet - An Electronic Lexical Database. In: *The MIT Press.* 1998.

FORD, Andrew. 1999. *Modeling the Environment: An Introduction To System Dynamics Modeling Of Environmental Systems.* Washington, USA : Island Press, 1999. ISBN 1559636017.

FüRST, Frédéric e TRICHET, Francky. 2005. Axiom-based ontology matching: a method and a experiment. In: *Relatório Técnico N° 05-02, Laboratório de Informática de Nantes-Atrantique (LINA).* Março 2005.

---, 2005. Axiom-Based Ontology Matching. In: *The 3rd International Conference on Knowledge Capture.* October 2005, pp. 195-196.

GHIDINI, Chiara e GIUNCHIGLIA, Fausto. 2001. Local Models Semantics, or Contextual Reasoning = Locality + Compatibility. In: *Artificial Intelligence.* April 2001, 2, Vol. 127, pp. 221-259.

GIUNCHIGLIA, Fausto; SHVAIKO, Pavel e YATSKEVICH, Mikalai. 2005. S-Match: an Algorithm and an Implementation of Semantic Matching. In: *Semantic Interoperability and Integration.* 2005.

GRANT, William; PEDERSEN, Ellen e MARíN, Sandra. 1997. *Ecology and Natural Resource Management: Systems Analysis and Simulation.* s.l. : John Wiley & Sons, 1997. ISBN 0471137863.

GRUBER, Thomas. 1995. Toward Principles for the Design of Ontologies used for Knowledge Sharing. In: *The International Journal of Human-Computer Studies.* 1995, Vol. 43, pp. 907-928.

GUARINO, Nicola e GIARETTA, Pierdaniele. 1995. Ontologies and Knowledge Bases Towards a Terminological Clarification. In: *The 2nd International Conference on Building and Sharing Very Large-Scale Knowledge Bases.* 1995, pp. 25-32.

HAASE, Peter e MOTIK, Boris. 2005. A Mapping System for the Integration of OWL-DL Ontologies. In: *The 1st International Workshop on Interoperability of Heterogeneous information Systems.* 2005, pp. 9-16.

HAEFNER, James. 1996. *Modeling Biological Systems: Principles and Applications* . s.l. : Springer, 1996. ISBN 0412042010.

HUANG, Wen-Lin, *at al.* 2008. ProLoc-GO: Utilizing Informative Gene Ontology Terms for Sequence-Based Prediction of Protein Subcellular Localization. In: *BMC Bioinformatics.* 2008.

I., Bomze, *at al.* 1999. The Maximum Clique Problem. In: *Handbook of Combinatorial Optimization.* 1999.

ICHISE, Ryutaro; TAKEDA, Hiedeaki e HONIDEN, Shinichi. 2003. Integrating Multiple Internet Directories by Instance-based Learning. In: *The Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03).* 2003, pp. 22–28.

---, 2001. Rule Induction for Concept Hierarchy Alignment. In: *The Workshop on Ontology Learning at IJCAI.* 2001.

JENSEN, Finn. 2001. *Bayesian Networks and Decision Graphs.* s.l. : Springer-Verlag, 2001. ISBN 0387952594.

JOACHIMS, Thorsten. 1997. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In: *The 14th International Conference on Machine Learning.* 1997.

JUAN, Alfons e NEY, Hermann. 2002. Reversing and Smoothing the Multinomial Naive Bayes Text Classifer. In: *The 2nd International Workshop on Pattern Recognition in Information Systems (PRIS'02).* 2002, pp. 200-212.

KALFOGLOU, Yannis e SCHORLEMMER, Marco. 2003. If-Map: an Ontology Mapping Method. In: *Journal on Data Semantics.* October 2003, pp. 98–127.

---, 2003. Ontology Mapping: the State of the Art. In: *The Knowledge Engineering Review.* January 2003, Vol. 18, pp. 1 - 31.

KENT, Robert. 2000. The Information Flow Foundation for Conceptual Knowledge Organization. In: *In Proceedings of the 6th International Conference of the International Society for Knowledge Organization (ISKO'00).* 2000.

---, 2000. The Information Flow Foundation for Conceptual Knowledge Organization. In: *The 6th International Conference of the International Society for Knowledge Organization (ISKO).* July 2000, pp. 10-13.

KLINGER, Stefan e AUSTIN, Jim. 2005. Chemical Similarity Searching Using a Neural Graph Matcher. In: *The European Symposium on Artificial Neural Networks (ESANN'05).* April 2005.

KORHONEN, Anna e BRISCOE, Ted. 2004. Extended Lexical-Semantic Classification of English Verbs. In: *Workshop on Computational Lexical Semantics.* May 2004, pp. 38–45.

KUMAR, Mukesh e MILLER, Douglas. 2006. A Non- Parametric Classification Strategy for Remotely Sensed Images using both Spectral and Textural Information. In: *The 24th IASTED International Multi-Conference Signal Processing, Pattern Recognition and Applications.* February 2006, pp. 81-89.

LACHER, Martin e GROH, Georg. 2001. Facilitating the Exchange of Explicit Knowledge Through Ontology Mappings. In: *The First International FLAIRS Conference.* May 2001.

LAMMA, Evelina e MELLO, Paola. 1999. *AI*IA 99: Advances in Artificial Intelligence: 6th Congress of the Italian Association for Artificial Intelligence.* Bologna, Italy : Springer-Verlag, 1999. ISBN:3540673504.

LIU, Tie-Yan*, at al.* 2005. Support Vector Machines Classification with Very Large Scale Taxonomy. In: *SIGKDD Explorations, Special Issue on Text Mining and Natural Language Processing.* 2005, 1, Vol. 7, pp. 36-43.

MADHAVAN, Jayant; BERNSTEIN, Philip e RAHM, Erhard. 2001. Generic Schema Matching with Cupid. In: *VLDB.* 2001, pp. 49-58.

MAGNINI, Bernardo; SPERANZA, Manuela e GIRARDI, Christian. 2004. A Semantic-Based Approach to Interoperability of Classification Hierarchies: Evaluation of Linguistic Techniques. In: *COLING-2004.* August 2004.

MAGNINI, Bernardo; SERANI, Luciano e SPERANZA, Manuela. 2003. Making Explicit the Semantics Hidden in Schema Models. In: *The Workshop on HLT form Semantic Web and Web Services at ISWC'03.* 2003.

MCCALLUM, Andrew*, at al.* 1998. Improving Text Classification by Shrinkage in a Hierarchy of Classes. In: *The 15th International Conference on Machine Learning (ICML'98).* 1998, pp. 359-367.

MCGUINNESS, Deborah*, at al.* 2000. An Environment for Merging and Testing Large Ontologies. In: *The 7th International Conference on Principles of Knowledge Representation and Reasoning.* 2000, pp. 483–493.

MELNIK, Sergey; RAHM, Erhard e BERNSTEIN, Philip. 2003. Rondo: A Programming Platform for Generic Model Management. In: *SIGMOD.* 2003, pp. 193–204.

MILLER, George. 1995. Wordnet: A Lexical Database for English. In: *Communications of the ACM.* 1995, pp. 39-41.

MINSKY, Marvin. 1975. A Framework for Representing Knowledge. In: *Reprinted in The Psychology of Computer Vision.* 1975, pp. 221–280.

NIGAMY, Kamal; LAFFERTY, John e MCCALLUM, Andrew. 1999. Using Maximum Entropy for Text Classification. In: *IJCAI-99 Workshop on Machine Learning for Information.* 1999, pp. 61-67.

NOY, Natalia e MUSEN, Mark. 2001. Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In: *Workshop on Ontologies and Information Sharing at the Seventeenth International JointConference on Artificial Intelligence (IJCAI'2001).* 2001.

PELILLO, Marcello; SIDDIQI, Kaleem e ZUCKER, Steven. 1998. Matching Hierarchical Structures Using Association Graphs. In: *The European Conference on Computer Vision (ECCV'98).* 1998, pp. 3-16.

PORTER, Martin. 1997. An Algorithm for Suffix Stripping. In: *Reprinted in Readings in Information Retrieval*. San Francisco : Morgan Kaufmann Publishers Inc., 1997, p. 313-316.

QU, Yuzhong; HU, Wei e CHENG, Gong. 2006. Constructing Virtual Documents for Ontology Matching. In: *The 15th International Conference on World Wide Web (WWW '06).* May 2006.

RAHM, Erhard e BERNSTEIN, Philip. 2001. A Survey of Approaches to Automatic Schema Matching. In: *VLDB Journal.* December 2001, Vol. 10, pp. 334–350.

RESNIK, Philip. 1990. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. In: *Journal of Artificial Intelligence Research.* 1990, pp. 95–130.

RIBEIRO, Regiane; SOARES, Vicente e VIEIRA, Carlos. 2005. Avaliação de Métodos de Classificação de Imagens IKONOS para o Mapeamento da Cobertura Terrestre. In: *Anais XII Simpósio Brasileiro de Sencoriamento Remoto.* april 2005, pp. 4277-4283.

RISH, Irina. 2001. An Empirical Study of the Naive Bayes Classifier. In: *International Joint Conferences on Artificial Intelligence, Workshop on Empirical Methods in Artificial Intelligence.* 2001.

SHAKHNAROVICH, Gregory; DARRELL, Trevor e INDYK, Piotr. 2006. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice.* s.l. : The MIT Press, 2006. ISBN 0-262-19547-X.

SHANNON, Claude. 1948. A Mathematical Theory of Communication. In: *Bell System Technical Journal.* July and October 1948, Vol. 27, pp. 379-423 and 623-656.

SHIBA, Marcelo, *at al.* 2005. Classificação de Imagens de Sensoriamento Remoto pela Aprendizagem por Árvore de Decisão: uma Avaliação de Desempenho. In: *Anais XII Simpósio Brasileiro de Sensoriamento Remoto.* April 2005, pp. 4319-4326.

SHVAIKO, Pavel e EUZENAT, Jerome. 2005. Tutorial on Schema and Ontology Matching. In: *The 2nd European Semantic Web Conference.* 2005.

SNOW, Rion; JURAFSKY, Daniel e NG, Andrew. 2006. Semantic Taxonomy Induction from Heterogenous Evidence. In: *The 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL.* 2006, pp. 801 - 808 .

SOROKINE, Alexandre; BITTNER, Thomas e RENSCHLER, Chris. 2005. Ontological Investigation of Ecosystem Hierarchies and Formal Theory for Multiscale Ecosystem Classifications. In: *Geoinformatica.* 2005, Vol. 10, pp. 313-335.

SOWA, John. 2000. *Knowledge Representation - Logical, Philosophical and Computational Foundations, 1st Edition.* s.l. : Brooks Cole Publishing Co., Pacific Grove, CA, 2000. 0534949657.

STUCKENSCHMIDT, Heiner, *at al.* 2004. Using C-OWL for the Alignment and Merging of Medical Ontologies. In: *KR-MED workshop at KR.* 2004.

STUMME, Gerd e MAEDCHE, Alexander. 2001. FCA-MERGE: Bottom-Up Merging of Ontologies. In: *The 17th International Joint Conference on Artificial Intelligence (IJCAI'2001).* 2001, pp. 225–234.

UDREA, Octavian e GETOOR, Lise. 2007. Combining Statistical and Logical Inference for Ontology Alignment. In: *Workshop on Semantic Web for Collaborative Knowledge Acquisition at the 20th International Joint Conference on Artificial Intelligence.* 2007.

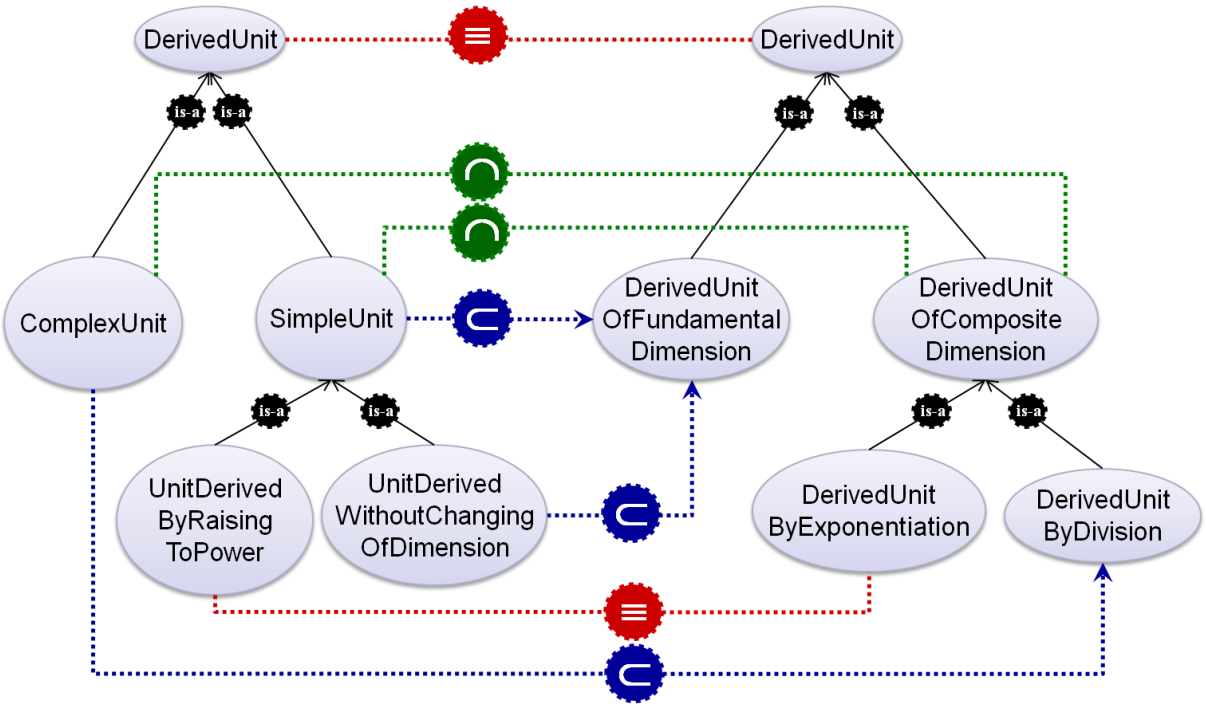
UDREA, Octavian; GETOOR, Lise e MILLER, Renée. 2007. Leveraging Data and Structure in Ontology Integration. In: *The 2007 ACM SIGMOD International Conference on Management of Data.* 2007, pp. 449 - 460.

VALE, Rodrigo, *at al.* 2001. Recuperação de Informação em Coleções Médicas Utilizando Categorização Automática de Documentos. In: *The XVI Simpósio Brasileiro de Banco de Dados.* October 2001, pp. 243-258.

VENANT, Fabienne. 2006. A Geometric Approach to Meaning Computation: Automatic Disambiguation of French Adjectives. In: *International Joint Conference.* October 2006.

YANG, Yiming; SLATTERY, Se e GHANI, Rayid. 2002. A Study of Approaches to Hypertext Categorization. In: *Journal of Intelligent Information Systems.* March 2002, 2, Vol. 18.

//// RASCUNHO



É importante que a semântica não seja ferida por mapeamentos errôneos, o que em princípio poderia ser garantido se especialistas humanos criassem os mapeamentos. Entretanto, manualmente este seria um trabalho braçal, árduo e propenso a erros de lógica, portanto soluções automáticas e precisas são inevitáveis.

An exciting and potentially far-reaching development in computer science is the invention and application of methods of machine learning. These enable a computer program to automatically analyse a large body of data and decide what information is most relevant. This crystallised information can then be used to automatically make predictions or to help people make decisions faster and more accurately.

The semantics introduced in De_nition 4 can be viewed as an instance of the compatibility relation between contexts as de_ned in Local Models Semantics [8, 5]. Indeed, suppose we take a set of documents D as the domain of interpretation of the local models of two contexts c_1 and c_2, and each concept as a unary predicate. If we see the documents associated to a concept as the interpretation of a predicate in a local model, then the relation we discover between concepts of di_erent contexts can be viewed as a compatibility constraint between the local models of the two concepts. For example, if the algorithm returns an equivalence between the concepts k_1 and k_2 in the contexts c_1 and c_2, then it can be interpreted as the following constraint: if a local model of c_1 associates a document d to k_1, then any compatible model of c_2 must associate d to k_2 (and vice versa); analogously for the other relations.

De_nition 4. A mapping function M from H_s to H_t is extensionally correct with respect to two hierarchical classi_cations $_s$ and $_t$ of the same set of doc-

uments D in H_s and H_t, respectively, if the following conditions hold for any k_s 2 K_s and k_t 2 K_t:

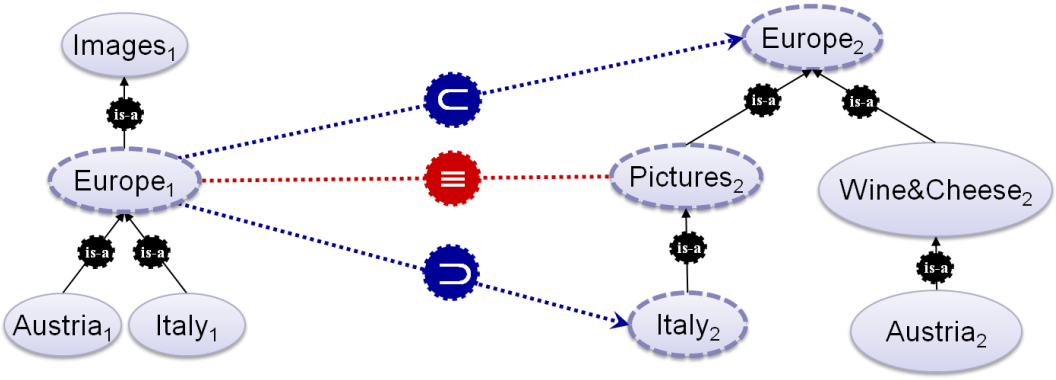
the directionality of information flow (BOUQUET*, at al.*, 2003)

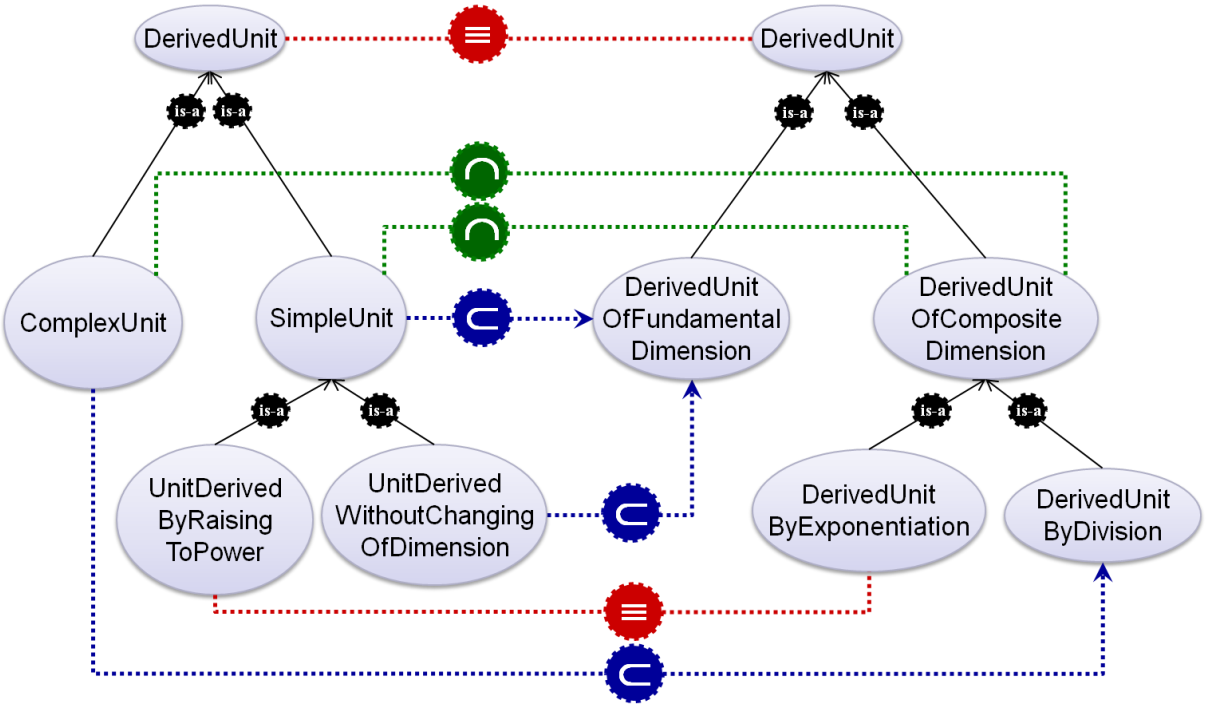
(LACHER e GROH, 2001) também utilizou the *Bow* toolkit para mapear conceitos de ontologias.
(RESNIK, 1990) shows how concept hierarchies
can also be used for the resolution of syntactic and semantic
ambiguities.

Veja um exemplo simples de mapeamento semântico dado por (GIUNCHIGLIA, *at al.*, 2005):

Basicamente, ontologias são formadas por conceitos que tipicamente estão organizados numa hierarquia taxonômica e para cada conceito são atribuídas algumas instâncias

teoricamente que as instâncias dos conceitos ontológicos são uma boa evidência para relacionar conceitos que receberam interpretações dadas por comunidades diferentes





Ontologias são compostas primariamente por conceitos freqüentemente numa hierarquia taxonômica e que por vezes recebem algumas instâncias: para nossa abordagem, que se completa nesta seção, tanto a quantidade de instanciações quanto de conceitos instanciados é importante, uma vez que a eficácia da classificação e do método que apresentaremos a seguir é influenciada, dentre outras coisas, pela quantidade de instâncias. Felizmente, ao contrário do que se temia, a quantidade reduzida de instâncias nas ontologias não foi problema para o mapeamento devido ao bom desempenho dos classificadores.

Por exemplo, nas ontologias Ecolingua e Apes, como o conceito *Ecolingua.EcologicalData.Quantity* aparece num contexto específico (dados ecológicos), é teoricamente menos geral que *Apes.SeamCore.Quantity* (dados de mensuração), contudo ambos os conceitos têm a mesma utilidade nas ontologias e, portanto, podem também ser considerados equivalentes sem problema.

Nossa motivação inicial era o SVM, pois esperávamos que sobrepujasse os outros algoritmos, como comumente ocorre. Mas os experimentos nos surpreenderam com o péssimo desempenho do SVM, enquanto os melhores resultados vieram de onde menos esperávamos: do NB-*Shrinkage*, que inicialmente nem ao menos havia sido cogitado.

Podemos mais uma vez diferenciar L-*Match* e S-*Match*, pois o primeiro raciocina sobre a enumeração (estatisticamente e localmente) enquanto o segundo raciocina sobre propriedades taxonômicas (simbolicamente e globalmente) para chegar num objetivo comum: relacionar conceitos através de *equivalência* (≡)**,** *mais geral (⊐)***,** *menos geral (⊏)***,** *sobreposto* (⊓) e *diferença* (≢). Teoricamente, concordamos que este problema remete à relação de compatibilidade entre contextos locais definida pela **Semântica de Modelos Locais** (GHIDINI e GIUNCHIGLIA, 2001) e que inspirou o S-*Match*. Na prática, isso é Teoria de Conjuntos utilizada para regrar interpretações de contextos locais, tornando-os compatíveis.

Os axiomas $\sqsupset$ e $\sqsubset$ são direcionados, assimétricos e inversos o que remete à **direcionalidade do Fluxo da Informação** (BOUQUET, *at al.*, 2003), área intimamente ligada à semântica da classificação de instâncias (KENT, 2000)**.** Logo, deve-se guardar a **ontologia origem** e a **ontologia destino** ao estimar similaridade e sobreposição, mantendo-os nas direções $A \longrightarrow B$, $B \longrightarrow A$ e $B \leftrightarrow A$.