UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Caio Stein D'Agostini

Captura e Gerência de Informações de Contexto Semântico para Recuperação de Informação

dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

Prof. Renato Fileto, Dr.
Orientador

Florianópolis, Dezembro de 2009

Captura e Gerência de Informações de Contexto Semântico para Recuperação de Informação

Caio Stein D'Agostini

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, área de concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Prof. Mario Antônio Ribeiro Dantas, Dr.
Coordenador

Banca Examinadora

Prof. Renato Fileto, Dr.
Orientador

Profa. Agma Juci Machado Traina, Dra.

Profa. Christiane Gresse von Wangenheim, Dra.

Prof. José Leomar Tedesco, Dr.

Agradecimentos

Agradeço à família, aos amigos, professores, funcionários e colegas, que incentivaram, orientaram e ajudaram. Em especial ao Renato Besen e Karina Fasolin que ajudaram a implementar o sistema desenvolvido, e aos usuários que realizaram os experimentos.

A realização deste trabalho recebeu apoio financeiro da Capes, CNPq (48139212007-6) e da FEESC.

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

A utilização de descrições semânticas do conteúdo disponível para busca é uma tentativa de melhorar a qualidade de sistemas de recuperação de informação. Uma maneira de se definir descrições semânticas é através de ontologias e bases de conhecimento, que são estruturas objetivas e formais capazes de descrever porções do conhecimento.

Computadores podem facilmente se beneficiar dessas estruturas, porém, usuários humanos, menos formais e menos objetivos, têm compreensões individuais sobre uma mesma informação, talvez incompatíveis com uma ontologia ou base de conhecimento. Essa compreensão é influenciada pelo meio, pelo conhecimento prévio do usuário e pela sua intensão, todos fatores caracterizantes do contexto semântico do usuário, que influencia como cada pessoa referencia e interpreta informações.

Este trabalho apresenta como acompanhar o comportamento do usuário ao longo das interações com um sistema de busca e capturar informações sobre o seu contexto semântico, para mapear a visão do usuário às descrições formais das ontologias e bases de conhecimento. O modelo formal, usado para representar a informação de contexto semântico de cada usuário, permite, através de algoritmos para captura e uso dessas informação nas interações com o sistema, minimizar a interação do usuário com grandes massas de descrições formais. Isto torna o sistema mais agradável e auxilia o sistema a capturar prováveis interesses do usuário.

A abordagem proposta foi implementada em um sistema de buscas semânticas denominado Praestro, o qual foi utilizado para a execução de alguns experimentos preliminares visando iniciar a avaliação dos possíveis benefícios da abordagem.

Abstract

The use of semantic descriptions of the content to be searched for is an attempt towards improving the quality of information retrieval systems. One way for defining semantic descriptions is through ontologies and knowledge bases, which are objective and formal structures capable of describing portions of knowledge.

Computers can easily benefit from those structures, however, human users, which are less formal and less objective, have individual comprehensions regarding same information, which maybe incompatible with an ontology or knowledge base. This comprehension is influencied by the environment, the previous knowlege from the user and by his intentions, all of them which are factors that characterize the user's semantic context, which affects how each person references and interprets information.

This work presents how to follow the user's behavior during his interactions with the search system and how to capture information about his semantic context and how to map the user's view to the formal descriptions from the ontology and knowledge base. The formal model used to represent the context information from each user allows, through the algorithms for capturing and using this information in the interactions with the system, to minimize the interaction of the user with large volume of formal descriptions. This turns the system more pleasent and also helping the system to capture likely user's interests.

The proposed approach was implemented in a semantic search system called Praesto, which was used to perform some experiments in order to begin the assessment of the possible benefits of the approach.

Capítulo 1

Introdução

A busca por palavras-chave é funcionalidade básica e essencial para diversos sistemas de recuperação de informação, dos quais o usuário espera obter auxílio ao trabalhar com grandes massas de dados. A princípio, os mecanismos de busca por palavras-chave realizam buscas estritamente léxicas e sintáticas. Eles retornam os recursos (documentos de texto, imagens, etc) cujos nomes ou conteúdo tenham correspondência com a palavra-chave buscada. Porém, este processo é falho, por desconsiderar diferenças entre os vocabulários de diferentes usuários [Tirri 2003] e ou figuras semânticas de linguagem, como sinonímia, homonímia, meronímia (relação semântica de inclusão entre parte e todo) e metonímia (emprego de um termo no lugar de outro).

Mangold [Mangold 2007] considera que as expectativas existentes em relação ao futuro, mais especificamente ao futuro da *Web*, dependem do sucesso de tecnologias que considerem aspectos semânticos. Essa dependência acontece devido à disponibilidade e facilidade de acesso à informação, em grandes quantidades. Muitas vezes, a conseqüência dessa facilidade de acesso é negativa para o usuário, que pode acabar por dissociar os dados da razão a qual eles foram coletados. Essa dissociação é causada pela sobrecarga do usuário, que passa a não saber como gerenciar ou mesmo entender os dados [Fry 2007], como por exemplo diversos resultados retornados para uma pesquisa por conteúdo associado a uma palavra-chave.

Um sistema de busca semântico é um sistema que tem a capacidade de entender

o conteúdo dos recursos além do nível sintático, também em nível semântico [Mangold 2007]. Para alcançar este objetivo, é feito uso de recursos como ontologias e bases de conhecimento, expressos, por exemplo, como conjuntos de triplas *RDF* (*Resource Description Framework*). Estes recursos fornecem ao sistema informações sobre os diferentes possíveis usos e interpretações do conteúdo armazenadas.

Entretanto, apesar desses recursos permitirem ao sistema identificar possíveis usos das palavras buscadas pelos usuários, muitos destes fornecem ao sistema informações generalizadas. Isso significa que desconsideram a compreensão individual que cada pessoa tem sobre uma informação e conseqüentemente a relevância da informação em relação às necessidades e interesses de um único usuário. A relevância de uma determinada informação para um usuário é influenciada por fatores externos, tais como conhecimento prévio sobre o assunto, aspectos culturais, influência de outras fontes de informação, etc. Porém, as ontologias e bases de conhecimento se comprometem com o conhecimento compartilhado por diversos usuários, sendo aplicadas a populações inteiras e não a indivíduos [Tirri 2003], o que impossibilita que fatores individuais sejam considerados.

A conseqüência dessa generalização é o distanciamento dos sistemas de busca, que herdam a generalidade e objetividade das ontologias e das bases de conhecimento, da natureza subjetiva e individual do usuário. Essa distância pode prejudicar a qualidade da interação do usuário com o sistema [Winkler 1999], *i.e.*, o usuário pode não estar familiarizado com a forma que um vocabulário é utilizado pelo sistema para descrever o conteúdo armazenado, podendo também não estar disposto a dispender o tempo necessário para se familiarizar. Uma forma de tratar este problema é através da personalização dos sistemas de busca.

1.1 Exemplos Motivadores

Aqui são apresentados dois exemplos utilizados ao longo do trabalho para ilustrar os problemas relacionados às buscas por palavras-chave e também às soluções propostas. Esses exemplos são focados na busca de instâncias, ou seja, não são realizadas buscas por conceitos (e.g.: uma busca é feita pelo nome de uma cidade e não pela palavra ‘cidade’).

Essa limitação se deve à necessidade de concentrar os esforços de desenvolvimento do trabalho na captura e uso do contexto semântico. Limitações como estas são impostas entre outros fatores, pela indisponilidade de fontes de dados preparadas para a realização de testes e experimentos de um sistema que leve em consideração a semântica e peculiaridades de cada usuário, que transcendem o escopo deste trabalho.conforme descrito em detalhes no Capítulo 4, que apresenta a implementação do sistema, o planejamento e a realização de experimentos.

Exemplo 1:

Dois usuários, ‘A’ e ‘B’ têm atividades bastante diferentes. Enquanto ‘A’ é jornalista esportivo, ‘B’ trabalha com prestação de serviços para prefeituras.

‘A’ tem interesse pelos resultados de partidas de futebol e ‘B’ tem interesse em licitações abertas por prefeituras.

‘A’ não tem interesse nos resultados retornados por ‘B’, nem vice-versa. O sistema deve interpretar a intenção de cada um por ‘São Paulo’ e apresentar os resultados de acordo.

Exemplo 2:

Um usuário, sem muitos conhecimentos geográficos, fica sabendo que há um território da França (a Guiana Francesa) que faz fronteira com o Brasil. O usuário pode pesquisar por Brasil. Mas Brasil é, além do nome do país, nome de filmes e músicas, dentre outras possibilidades. Já França pode trazer muitos resultados relativos a diferentes momentos históricos, por exemplo, resultados associados à França na idade média, antes mesmo da ocupação da América.

1.2 Objetivos

O objetivo deste trabalho é desenvolver um processo para auxiliar a recuperação de informação pela captura e utilização do contexto semântico de cada usuário, em sistemas

que organizam o conteúdo com base em descrições e definições formais fornecidas por uma ontologia.

Espera-se que o processo de busca proposto, quando implementado junto a um sistema de busca, auxilie o usuário através da desambiguação de palavras ambíguas e da identificação e reutilização de correlações entre diferentes denotações de palavras-chave presentes ao longo de diversas interações do usuário com o sistema.

1.2.1 Objetivos Específicos

A fim de se obter os resultados esperados para este trabalho, é necessária a realização de diversos objetivos específicos, listados na seqüência.

- Definição de um modelo capaz de representar o conhecimento, tanto o armazenado no sistema como também o conhecimento do usuário, de acordo com o seu contexto semântico.
- Integração do modelo para representação do conhecimento relativo ao contexto semântico do usuário à arquitetura de um sistema de buscas semânticas.
- Desenvolvimento de algoritmos para a realização das buscas semânticas guiadas por informações extraídas do contexto semântico do usuário.
- Validação do trabalho através da realização de experimentos com usuários.

1.2.2 Especificidades

As especificidades referentes às delimitações de escopo do trabalho dizem respeito principalmente às capacidades do sistema de busca. O desenvolvimento do trabalho foca no desenvolvimento de mecanismos para a personalização de sistemas de busca semânticos, não no desenvolvimento do sistema completo. Por isso, são desconsideradas questões relativas ao processamento de palavras-chave usadas nas consultas (e.g.: uso de letras maísculas ou minúsculas, acentuação, busca aproximada por palavras semelhantes à palavra-chave, etc).

Outra característica do trabalho apresentado é sua utilização exclusiva de busca por instâncias, não por conceitos. As alterações necessárias para permitir buscas por conceitos envolvem o tratamento de questões relacionadas à anotação semântica do conteúdo, que estão fora do escopo deste trabalho.

1.3 Justificativa

Apesar de ser uma alternativa aparentemente bastante promissora quando aplicada à recuperação de informação, a personalização não é uma solução viável se ela depender somente do esforço do usuário. Sistemas que exigem que os usuários se adeqüem a construções prévias, sejam elas ontologias ou modelos de perfis, são conseqüência do desenvolvimento sem preocupação com a capacidade do sistema de observar o usuário e como este reage a cada interação com o sistema. A consequência deste tipo de desenvolvimento é que os usuários normalmente têm dificuldades de definir o que precisam ao criar perfis [Tirri 2003].

Como a capacidade de observação faz parte do processo de comunicação das pessoas [Degler and Lewis 2004], espera-se que integrá-la ao processo de recuperação de informação melhore a qualidade da interação com o usuário.

1.4 Metodologia

Esta seção apresenta a metodologia utilizada para o desenvolvimento deste trabalho. As etapas aqui apresentadas estão na ordenadas da forma como foram desenvolvidas, não correspondendo necessariamente com a organização do conteúdo no decorrer deste trabalho.

Revisão Bibliográfica Relacionada à Representação do Conhecimento

Em um primeiro momento, o foco da pesquisa foi verificar as possíveis limitações referentes ao uso de ontologias e bases de conhecimento para a representação de conhe-

cimento, para sistemas de informação voltados para usuários humanos.

Foram estudadas teorias sobre como as pessoas organizam o seu conhecimento e como essa organização reflete na compreensão que cada pessoa tem, individualmente, a respeito de uma informação armazenada por outra pessoa ou por ela mesma, porém em outro momento, em um contexto diferente. Com base nestas teorias, buscou-se por alternativas para a representação do conhecimento dos usuários, de maneira individual.

Análise de Requisitos

Após estudar como as pessoas representam e armazenam conhecimento, foi feita uma análise de requisitos não funcionais para um sistema de busca. Nesta análise, procurou-se identificar características desejáveis para um sistema de busca capaz de trabalhar com o conhecimento do usuário de forma personalizada.

De maneira resumida, espera-se que o sistema seja capaz de representar ao mesmo tempo informações formalmente definidas, generalizadas e compartilhadas entre diversos usuários, e informações individuais que não necessariamente são compreendidas por outros usuários. Entretanto, estes dois tipos de informações são conflitantes, sendo necessária a sua separação em diferentes componentes do sistema, sem demandar esforço extra do usuário para isto.

Desenvolvimento de um Modelo Formal para Representação do Conhecimento

Dado que já existem diversos modelos e ferramentas para representação e manipulação de informações formais e compartilhadas, como através de ontologias, foi necessário encontrar um modelo formal para representar a porção individual do conhecimento de cada usuário.

Nesta etapa foram pesquisados formas flexíveis para representação de informação, de forma a permitir a personalização do sistema conforme o conhecimento individual e dinâmico do usuário, mas ao mesmo tempo permitindo o mapeamento desta informação para a informação contida na representação formal e objetiva na forma de ontologia.

Revisão Bibliográfica sobre Sistemas Adaptativos e sobre Comportamento Emergente

Um elemento importante na comunicação e processo de aprendizado das pessoas é o *feedback* de informações, que permite que as partes se comunicando, através de pequenos ajustes, aprendam gradualmente a inferir a intensão da outra parte, criando uma comunicação efetiva.

Como o objetivo deste trabalho é o desenvolvimento de um sistema capaz de aprender a intensão do usuário de maneira o mais transparente possível, esta etapa dedicou-se ao estudo de heurísticas utilizadas para criação de sistemas adaptativos.

Implementação do Buscador

Havendo estudado o processo de representação e armazenamento do conhecimento utilizado pelas pessoas, assim como heurísticas para reproduzir este tipo de comportamento, foi realizada a implementação do sistema.

Nesta etapa foi implementado o buscador desenvolvido e foram definidas as interfaces com as quais, posteriormente, o sistema pode ser adaptado para ser utilizado em sistemas de busca específicos ou em conjuntos de dados específicos.

Obtenção e Preparação de um Conjunto de Dados

Antes da realização de experimentos para testar e validar o sistema implementado foi necessária a obtenção de dados adequados às necessidades do sistema. Nesta etapa foram buscadas fontes de dados ou ferramentas capazes de preparar fontes de dados, de forma a fornecer ao sistema um conjunto de documentos anotados semanticamente, de acordo com as definições fornecidas por uma ontologia utilizada pelo sistema de busca semântico.

Na falta de ferramentas capazes de atender às necessidades deste trabalho, o conjunto de dados foi buscado e preparado com base em documentos da Wikipedia, indo além do escopo originalmente definido para o trabalho, no qual era esperada a disponibilidade

prévia dos mesmos.

Realização de Experimentos e Análise dos Resultados

O trabalho foi concluído com a realização de experimentos com usuários. Os experimentos consistiram na realização de séries de buscas com um grupo de usuários dispostos a colaborar com a realização dos experimentos.

Após a realização dos experimentos, os dados coletados foram analisados, procurando-se observar indicativos e alguma melhoria na qualidade dos resultados obtidos pelos usuários. Após a análise, foi verificada um aumento gradual, ao longo de diversas interações do usuário com o sistema, entre o que cada usuário considerou como sendo relevante e o que o sistema retornou para o usuário com sendo um resultado relevante.

1.5 Organização do Trabalho

O Capítulo 2 apresenta a fundamentação teórica que motivou o trabalho e que também é necessária para sua compreensão. Os dois temas abordados são a organização do conhecimento por parte das pessoas e a apresentação de sistemas emergentes e adaptativos.

O Capítulo 3 descreve a arquitetura do sistema de busca semântica sendo desenvolvido, ao qual são aplicadas as contribuições deste trabalho. A arquitetura possui três níveis de informação, organizadas em contexto semântico, definições compartilhadas através de uma base de conhecimento e finalmente o conteúdo armazenado anotado semanticamente.

Na seqüência, o trabalho segue com a apresentação da solução proposta para representação e armazenamento de informações sobre o contexto semântico do usuário, assim como algoritmos relacionados ao processo de busca e observação do comportamento dos mesmos. A implementação destes algoritmos, assim como a descrição dos experimentos realizados para a validação deste trabalho e posterior análise dos dados coletados são descritos no Capítulo 4. Nesse capítulo é discutido o processo de obtenção de dados para a realização de experimentos. Alguns trabalhos relacionados são descritos no Capítulo

5, fornecendo informação extra para o leitor sobre outras iniciativas para utilização de informação de contexto dos usuários.

Finalmente, o Capítulo 6 apresenta a conclusão do trabalho, assim como algumas propostas para continuidade da pesquisa.

Capítulo 2

Fundamentos

Para poder se compreender a motivação deste trabalho e os passos adotados na busca dos objetivos esperados para sua conclusão, é necessario conhecer algumas teorias sobre como se dá a organização do conhecimento na mente humana e em sistemas de informação baseados em ontologias, e o impacto destas organizações no funcionamento de serviços de armazenagem de informação.

2.1 Organização do Conhecimento e Contexto

A utilização de ontologias é uma forma de se fornecer descrições formais e consensuais para a descrição do conteúdo armazenado. As ontologias podem ser utilizadas para a geração de anotações de conteúdos, descrevendo os conteúdos por meio de metadados descritos pela mesma. Por isso, pode-se considerar este processo de anotação semântica como um sendo baseado em definições da ontologia.

Os registros de conteúdo são definidos em um momento no tempo (são *timebound*), mas os metadados que descrevem o conteúdo são descritos durante o processo de anotação do conteúdo de acordo com a ontologia, em um momento posterior. Porém, os metadados devem permanecer compreensíveis para um observador em outro momento [Hurley 1995]. Todavia, isto nem sempre acontece. Ainda que algumas circunstâncias envolvidas na criação de um registro (como quem o criou, quando o criou, como o criou e porque o

criou) sejam contemporâneas à criação do registro, elas são históricas e sua interpretação depende de uma referência.

Isso significa que os metadados têm o significado pretendido pelo autor no momento em que são criados, de acordo com o ponto de vista do autor, mas a mesma interpretação não pode ser garantida em outro momento ou sob a perpectiva de outro usuário. A compreensão é afetada ao longo do tempo por variações no contexto do usuário acessando o registro.

O contexto, quando analisado no escopo de gerenciamento de conhecimento e sistemas de buscas, pode ser visto como um conjunto de elementos envolvendo uma entidade considerada relevante em um domínio de interesse, em uma situação específica, durante um intervalo de tempo [Souza et al. 2008] . Outra definição semelhante é fornecida por [Mani and Sundaram 2007], que definem o contexto como sendo o conjunto de informações sob atenção e que influencia a troca de mensagens entre duas entidades se comunicando.

De maneira simplificada, essas definições de contexto indicam que os significados das palavras mudam ao longo do tempo e também de pessoa para pessoa. O fator causador dessas mudanças pode ser, inclusive, informações obtidas no decorrer de uma pesquisa do usuário. Porém, muitos sistemas de busca assumem que as necessidades das pessoas são estáticas, o que torna o sistema incapaz de se adaptar às mudanças de interesse [Baeza-Yates et al. 1999]. Uma das causas é justamente o fato que essas mudanças não se refletem nos metadados utilizados para descrever o conteúdo armazenado. Assim sendo, contexto, no escopo deste trabalho, deve ser entendido como o contexto referente à semântica envolvida na comunicação entre usuário e o sistema de busca, ou seja, deve ser compreendido como **contexto semântico**.

2.1.1 Contexto Semântico: Separando Visões de Definições

A importância de se explorar as informações sobre o contexto dos usuários em serviços de informação pode ser ilustrada através da *Web*, mais precisamente, através de sua implantação, que pode ser dividida em uma série de estágios, cada um com diferen-

tes limitações. Essa divisão é feita por [Naeve 2005], que divide a implantação da *Web* semântica em três momentos:

- `Isolamento semântico` (*Semantic isolation*)
- `Coexistência semântica` (*Semantic coexistence*)
- `Colaboração semântica` (*Semantic collaboration*)

O primeiro estágio é caracterizado pela dificuldade de acesso e utilização dos metadados que descrevem as informações armazenadas na *Web*. Para um usuário ter acesso a esses metadados, ele deve saber das suas existências e como utilizá-los, o que nem sempre acontece. Por exemplo, uma ontologia utilizada como base para um processo de anotação de documentos pode diferir da visão que o usuário tem sobre o domínio em questão.

No estágio seguinte, de coexistência, os diferentes repositórios de informações publicam na *Web* informações sobre para quais conteúdos podem fornecer respostas. Os metadados podem ficar disponíveis publicamente para consulta, permitindo que motores de busca na *Web* localizem repositórios que utilizam metadados com os mesmos nomes. Conseqüentemente, exime-se o usuário da necessidade de conhecer onde os metadados estão armazenados. Contudo, para que o usuário possa se beneficiar plenamente da descrição do conteúdo provida pelos metadados, ele necessita conhecer a ontologia que os descreve.

É possível, porém, que o vocabulário utilizado pelo usuário não possua uma interseção com a descrição fornecida pela ontologia usada pelo sistema, ou que as palavras conhecidas e utilizados pelo usuário denotem elementos do domínio diferentes dos descritos na ontologia. Nestas situações a ontologia não cumpre adequadamente a sua função de prover uma descrição formal e consensual para os conhecimentos dos usuários. E, como não é provável que um usuário não venha a se familiar com todas as ontologias que ele utiliza na *Web*, é preciso que seja estabelecido um mapeamento entre a visão do usuário e as ontologias utilizadas pelos sistemas.

Através da realização desse mapeamento se estabelece uma colaboração semântica, na qual diversas entidades podem colaborar entre si, mesmo que elas possuam visões po-

tencialmente conflitantes sobre um mesmo tema. Naeve [Naeve 2005] utiliza uma arquitetura estruturada para descrever o conhecimento e o papel do contexto na interpretação da informação, chamada de *Knowledge Manifold*, descrita na Definição 1.

Definição 1: *Knowledge Manifold*

Uma Knowledge Manifold possui três componentes:

`conceito`,

`interior` do conceito,

e o `exterior` do conceito.

Dado um conceito `C`,

o `exterior de C` é chamado de `contexto de C`

e o `interior de C` é chamado de `conteúdo de C`.

De acordo com esta definição, ao conhecer o contexto em que cada usuário se encontra ao se referir a um conceito, é possível inferir possíveis intenções do usuário. E, inversamente, conhecendo o conteúdo (interior do conceito) de interesse do usuário e o presente contexto (exterior do conceito), também é possível tentar inferir quais são os conceitos que o usuário considera relevantes.

O trabalho de Winkler [Winkler 1999] apresenta uma visão semelhante sobre a organização do conhecimento, enfatizando a necessidade de separar a descrição dos domínios (conceitos da *Knowledge Manifold*) das visões que os usuários têm dos mesmos (contexto da *Knowledge Manifold*). Os autores argumentam que, mesmo que todos os usuários estejam familiarizados com as mesmas definições, ou seja, conheçam as mesmas coisas que compõem um domínio, a visão de cada usuário a respeito do domínio é diferente. Os autores apontam a necessidade de se criar um mapeamento entre as visões dos diferentes usuários (seus contextos) e as descrições formais do conhecimento (as ontologias).

Contudo, somente conhecer a importância do contexto de cada usuário não basta. É necessário também saber como capturá-lo e como representá-lo. Uma representação adequada é necessária para que a informação de contexto possa ser utilizada pelo sistema. Já o problema da captura da informação de contexto consiste na sua obtenção com o mínimo

necessário de esforço por parte do usuário. Para guiar a escolha de possíveis soluções para estes problemas, alguns requisitos foram levantados, com base em características buscadas ou trabalhadas por diversos trabalhos relacionados que também se propõem a utilizar informação de contexto do usuário:

- `Individualidade:` A representação do contexto de um usuário deve ser independente dos contextos de outros usuários.
- `Flexibilidade:` A representação do contexto do usuário deve evoluir conforme as suas intenções e percepções do domínio sendo tratado evoluem com o decorrer das interações com o sistema.
- `Transparência:` O objetivo da utilização de informações sobre o contexto do usuário é esconder, sempre que possível, a estrutura utilizada para descrever formalmente o conhecimento, no caso, a ontologia e a base de conhecimento utilizada pelo sistema. Desta forma, o usuário pode se concentrar na tarefa sendo realizada, que é a busca de informações e a interpretação das respostas, sem que seja exigido do usuário esforço extra para captura e manutenção do contexto capturado.

2.1.2 Representação do Contexto Semântico

O contexto do usuário pode ser representado de diversas maneiras. Dependendo do objetivo específico dos sistemas e da forma como eles são construídos, uma solução adequada para um sistema pode não ser a melhor para outro. A seguir algumas abordagens sao apresentadas para ilustrar essas diferenças.

- `Regiões (de documentos XML ou de ontologias):` Os trabalhos [Aleman-Meza et al. 2003] [Graupmann et al. 2005] representam o contexto como regiões, respectivamente, de uma ontologia ou de um documento XML. Quando o usuário fornece informações como palavras-chave o sistema cria caminhos entre os elementos semânticos correspondentes aos dados referentes às palavras fornecidas. Um caminho que interseciona uma região é considerado semanticamente relacionado ao contexto representado pela região.

A proposta apresentada em [Aleman-Meza et al. 2003] utiliza regiões ontológicas definidas por usuários especialistas. Como as regiões são definidas diretamente na ontologia, a qual é compartilhada por todos os usuários de sistemas que a utilizem, os contextos não são individuais. Na aplicação apresentada pelo autor, as regiões são definidas em uma ontologia relacionada à segurança, a qual é utilizada por sistemas de segurança para identificação de pessoas que representem possíveis ameaças. Os usuários são agentes de segurança, que devem se comportar conforme regras pré-definidas por terceiros e não de acordo com critérios pessoais.

O sistema de busca apresentado em [Graupmann et al. 2005] utiliza regiões definidas dinamicamente, criadas com base nos parâmetros de busca passados pelo usuário. Regiões circulares são definidas agrupando elementos dos documentos XML vizinhos aos elementos que representam o assunto sendo pesquisado. Desta maneira, as regiões são criadas dinamicamente, de acordo com a busca de cada usuário. Porém, para aproveitar todo o potencial do sistema, o usuário deve ter algum nível de conhecimento sobre como estas regiões são criadas, de forma a saber como formular a consulta.

- `Perfis Ontológicos:` Os trabalhos descritos em [Challam et al. 2007] [Sieg et al. 2007] [Vallet et al.] designam pesos aos termos e relações em uma ontologia de domínio, baseados na relevância que cada um possui para os interesses do usuário. Esta solução permite que seja feita a personalização da representação do contexto, já que os pesos associados a cada usuário podem ser armazenados individualmente. Porém, este método ainda considera somente relações já existentes na ontologia, não permitindo que o usuário insira novas informações na representação do contexto além do quanto cada termo ou relação já existente são relevantes.

- `Grafos:` O último tipo de representação de contexto é na forma de grafos. Os vértices do grafo representam assuntos em que o usuário tem interesse, enquanto as arestas representam as relações entre esses temas. Esta abordagem é utilizada em diversos trabalhos, como [Winkler 1999] [Michlmayr et al. 2007] [Mani and Sundaram 2007] [Huang et al. 2002] [Leake et al. 2005].

Assim como acontece com o uso de perfis ontológicos, o sistema pode criar uma representação individual, porém na forma de um grafo, para o contexto de cada usuário, contudo, o grafo pode ter sua estrutura editada quando necessário. Isto permite uma personalização ainda maior, já que o grafo não tem a necessidade de se manter de acordo com uma estrutura mantida por terceiros, no caso, a ontologia utilizada pelo sistema.

2.1.3 Captura do Contexto Semântico

Além do problema de como se representar o contexto, há o problema de como capturar as informações sobre o contexto do usuário. Na seqüência, são apresentadas algumas propostas organizadas em três grupos de abordagens.

- `Questionamento aos Usuários:` O sistema pode questionar o usuário diretamente para obter as informações de que necessita. Esta opção é personalizável, já que cada usuário fornece informações a seu respeito, porém é altamente invasiva.

  Outra solução é encarregar um usuário especialista da tarefa de fornecer informações ao sistema, como em [Aleman-Meza et al. 2003]. A desvantagem é que ao encarregar um terceito elemento, como um usuário especialista, de fornecer a informação referente a todos os usuários, o sistema se torna incapaz de operar com uma coleção diversa de usuários com interesses diferentes.

- `Análise do Conteúdo Disponível:` Dada uma palavra-chave, o processo descrito em [Leake et al. 2005] analisa a freqüência e proximidade das palavras nos resultados retornados para a palavra em diversos sistemas de busca. Baseado nesses resultados o sistema calcula indicadores de quanto cada palavra é relevante para a palavra-chave fornecida, podendo assim avaliar quão relevante é cada resultado, com base nas palavras em seu conteúdo. Porém, essa solução captura um único contexto, que é referente aos autores dos conteúdos retornados nos momentos de suas criações, não o contexto de um usuário específico no momento que este solicita a busca.

- `Monitoramento do Retorno do Usuário (`*`Feedback`*`):` Os trabalhos descritos em [Sieg et al. 2007], [Michlmayr et al. 2007] e [Mani and Sundaram 2007] extraem informações sobre o contexto da interação entre o usuário e o sistema. Ações de seleção ou *downloading* de conteúdo realizadas pelo usuário são consideradas como indicativos de que os respectivos conteúdos retornados pelo sistema estão alinhados com seus interesses. Deste modo, estes conteúdos são considerados relevantes dado o atual contexto do usuário. Esta é uma solução transparente para o usuário, no sentido que ele não percebe a coleta de informações. Além disso é uma solução que funciona tanto para grupos como para usuários individuais e que pode ser mantida atualizada ao longo do tempo (através de novas observações da interação usuário-sistema).

2.2 Sistemas Emergentes, Adaptativos e *Feedback*

Emergência, dentre outras possíveis definições, é a propriedade de um sistema em que padrões maiores podem emergir de ações locais descoordenadas [Johnson 2001]. Este tipo de comportamento está presente na natureza (organização de colônias de amebas, de células formando órgãos ou insetos sociais como formigas), na sociedade (organização da cidade em bairros não planejados, organizados por classe social, atividade comercial, etc, ou na forma como o mercado se auto-regula equilibrando oferta e demanda) e também em *softwares* (como redes neurais). Quando, além do surgimento de padrões maiores, os padrões se adaptam às mudanças no meio, os sistemas também são adaptativos.

Um tipo de sistema emergente e adaptativo que segue regras simples e é capaz de gerar sistemas extremamente adaptáveis é o sistema adotado por formigas. Este sistema muitas vezes é adaptado para uso em *softwares*.

A adaptação desses sistemas é feita através de algoritmos de formiga. Os algoritmos de formigas (*ant algorithms*) são uma categoria de algoritmos que adotam heurísticas que usam *feedback*. As heurísticas são baseadas no comportamento que alguns insetos sociais, como formigas ou cupins, apresentam durante a busca por alimento [Dorigo et al. 1999, Shtovba 2005, Panait and Luke 2004]. A vantagem obtida na utilização deste tipo de

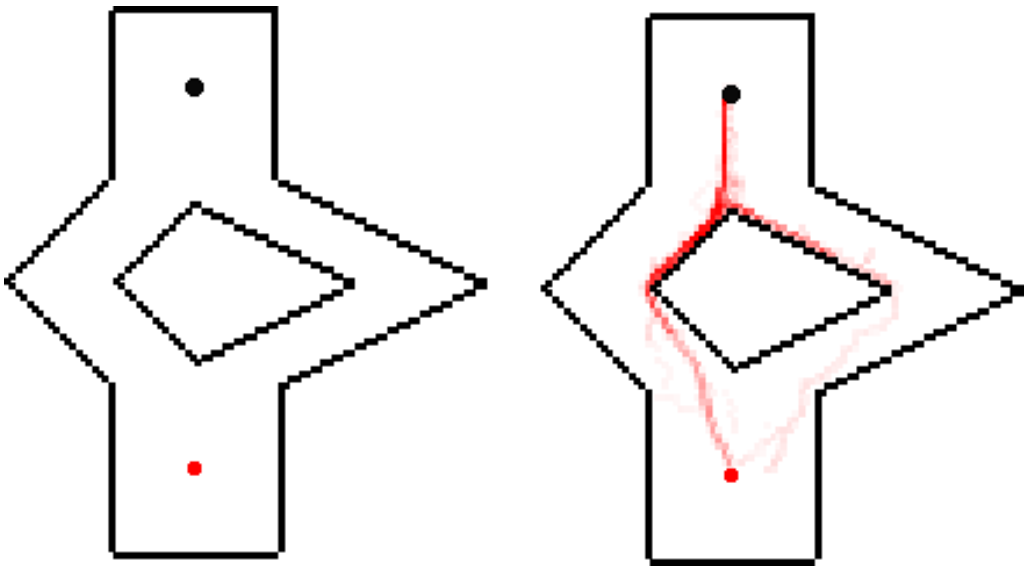

Figura 2.1: Comunicação estigmérgica através da alteração do ambiente por meio de rastros de ferormônios. O ponto superior representa uma colônia e o inferior uma fonte de comida. Os traços retos são obstáculos e traços curvos são os rastros de ferormônios marcando os caminhos (quanto mais escuro, mais curto).

algoritmo provém de sua capacidade de buscar colaborativamente soluções para problemas, reaproveitando resultados parciais obtidos por outros elementos do sistema que executam o mesmo algoritmo. Esta colaboração ocorre de maneira semelhante ao processo adotado por esses insetos, onde diversos indivíduos com capacidade limitada contornam suas limitações aproveitando as informações já obtidas por outros indivíduos.

Esses insetos adotam um mecanismo de comunicação mediado através de rastros de ferormônio deixados nos caminhos que percorrem. A este tipo de comunicação é dado o nome de estigmergia (*stigmergy*). Este tipo de comunicação é caracterizado por ocorrer através de modificações no estado do ambiente, onde informações só podem ser acessadas localmente, por meio dos rastros de ferormônio [Dorigo et al. 1999], conforme ilustrado na Figura 2.1. A figura foi gerada com auxílio de um simulador do comportamento de busca de alimento por formigas [1].

A presença dos rastros, assim como suas intensidades, indicam o potencial para pre-

[1] *http://www.rennard.org/alife/english/antsgb.html*, acessado em 14 de junho de 2009

sença de comida no fim de um caminho, quando percorrido de forma orientada, partindo-se de um ponto conhecido, como um formigueiro. Quando indivíduos buscando por alimento encontram comida, eles retorna para a colônia, deixando rastros de ferormônio por onde passam. Como os insetos retornando de fontes de comida mais próximas à colônia retornam antes que seus semelhantes, os caminhos para essas fontes são marcados antes que os caminhos para fontes mais distantes. Isto permite que seja feita uma **avaliação implícita** [Dorigo et al. 1999] da qualidade da fonte de alimento, mais especificamente, a sua distância da colônia, apesar da capacidade limitada que um único inseto possui para processar informações.

Após retornarem à colônia, quando saem para buscar alimento novamente, os indivíduos que percorreram caminhos mais longos nas suas buscas anteriores podem perceber rastros de ferormônio mais intensos que os rastros marcando os caminhos que haviam percorrido anteriormente. Estes indivíduos têm uma grande probabilidade de passarem a seguí-los também. Através deste comportamento, chamado de **autocatálise**, os caminhos utilizados com mais freqüencia têm maior chance de serem percorridos por novos indivíduos, que também acabam por aumentar a sua freqüência de utilização [Dorigo et al. 1999].

Dessa maneira, com o decorrer do tempo, mais indivíduos passam a procurar por comida seguindo caminhos que anteriormente levaram a bons resultados. Contudo, a utilização somente da avaliação implícita dos resultados e da autocatálise levaria todos indivíduos da colônia a se concentrarem nas mesmas regiões, limitando a adaptabilidade da colônia, i.e.; impedindo que descubram um caminho até uma nova fonte de alimento mais próxima que a fonte sendo consumida. Esta situação não acontece porque há uma pequena probabilidade de cada indivíduo ignorar os rastros de ferormônio. Esta flexibilidade permite que a colônia encontre alternativas para uma rota interrompida ou uma fonte de alimento exaurida [Dorigo et al. 1999], garantindo sua capacidade de adaptação à possíveis mudanças.

Os algoritmos baseados no comportamento de insetos sociais podem ser implementados utilizando como guia a meta-heurística *ACO (Ant Colony Optimization)* [Dorigo et al. 1999]. Ela fornece um algoritmo base onde os componentes e a coordenação dos

mesmos são organizados de maneira semelhante à organização das formigas. Cabe aos desenvolvedores do sistema adaptarem heurísticas de forma a se adequarem à estrutura da meta-heurística *ACO*, de acordo com as especifidades do problema sendo tratado.

2.2.1 Meta-Heurística *ACO (Ant Colony Optimization)*

ACO (Ant Colony Optimization) é o nome atribuído por [Dorigo et al. 1999] para uma meta-heurística para desenvolvimento de algoritmos de formiga. Os algoritmos desenvolvidos de acordo com esta meta-heurística implementam heurísticas que mimetizam o comportamento estigmérgico das colônias de formigas. Estes algoritmos utilizam colônias de formigas artificiais para procurar coletivamente por resultados para um problema, de forma semelhante ao processo estigmérgico que colônias usam para buscar comida. Para clareza da explicação, no restante deste trabalho o termo 'formiga' é associado às formigas virtuais, artificiais.

O processo de busca realizado pelas formigas é influenciado pela presença e intensidade dos rastros de ferormônios. Os ferormônios são variáveis de estado associadas a diferentes elementos da representação do problema, como um caminho em um grafo onde as arestas são marcadas com pesos que representam o grau de importância das mesmas. As buscas são realizadas percorrendo esta representação, guiadas pelos ferormônios, os quais servem como meio para a comunicação indireta entre as formigas. Os rastros marcam os caminhos já percorridos por diferentes possíveis soluções para o problema, permitindo que as formigas compartilhem seus resultados [Dorigo et al. 1999], associando diferentes fontes de recursos capazes de atenderem a um mesmo tipo de busca [Greer et al. 2007].

Cada formiga do sistema possui uma visão limitada do problema, restrita aos elementos em sua vizinhança, e.g., em um grafo onde os nós representam elementos do problema, uma formiga vê somente os elementos representados pelos nós vizinhos ao nó por ela visitado. Deste modo, uma única formiga é capaz de encontrar soluções, mesmo que de pouca qualidade. Ao realizar a busca em somente algumas fontes de informação, que conhecidamente respondem à busca, o sistema também reduz a quantidade de buscas

a serem realizadas [Greer et al. 2007].

A qualidade dos resultados é conseguida através dos mecanismos autocatalícos e da avaliação implícita. As melhores fontes de resultados, marcadas mais intensamente pelo ferormônio, são reaproveitadas como guias por outras formigas. As fontes que geram resultados com pouca qualidade são gradativamente abandonadas. Desta coordenação indireta entre os indivíduos da colônia e através do reaproveitamento de iterações prévias do sistema, emergem, gradativamente, resultados com melhor qualidade, conforme a colônia se ajusta ao problema sendo tratado [Dorigo et al. 1999].

Algoritmo 1: Algoritmo *ACO (Ant Colony Optimization)*. Os condicionais *booleano* são utilizados devido às três opções possíveis para colocação dos ferormônios.

```
1  início ACO
2      início Geração e Atividades
3          enquanto estado atual ≠ estado alvo faça
4              formiga = colônia.cria_formiga();
5              formiga.lê_rastros_de_ferormônio();
6              estado_atual = formiga.computa_próximo_estado();
7              deixa_rastros_passo_a_passo(booleano);
8          fim
9           deixa_rastros_após_avaliar_resultados(booleano);
10         formiga.morre();
11     fim
12     início Evaporação
13         evapora_ferormônio;
14     fim
15     início Atividades do Daemon
16         daemon.avalia_resultados();
17         daemon.deixa_rastros(booleano);
18     fim
19 fim
```

A fim de tornar os algoritmos mais eficientes, a colocação do ferormônio pelas formigas pode ser feita em três momentos diferentes. O primeiro é a cada novo passo percorrido na busca, o seguinte é somente após a obtenção e avaliação dos resultados. A

terceira possibilidade depende do uso de um *daemon*. O *daemon*, que não tem correspondente nas colônias reais, tem uma visão global do comportamento das formigas e pode coletar informações úteis para tomada de decisões quanto à colocação do ferormônio. Ele pode realizar otimizações dos resultados, reforçar os rastros de ferormônios deixados pelas formigas, ou mesmo desconsiderá-los, aplicando o ferormônio de acordo com estratégias específicas para o problema.

Todas essas possíveis abordagens para colocação do ferormônio, assim como a coordenação entre as formigas e com o *daemon*, são apresentados no Algoritmo 1, que é dividido em três fases [Dorigo et al. 1999]:

- `Geração das formigas e atividade` (linhas 2-11).
- `Evaporação do ferormônio` (linhas 12 - 14).
- `Atividades do` *`daemon`* (linhas 15-18) .

A fase de evaporação dos ferormônios é essencial para que o sistema possa se adaptar às novas condições do problema. Nesta fase, rastros de ferormônio que não são reforçados têm suas intensidades atenuadas, refletindo a possível diminuição de suas relevâncias para a solução do problema [Dorigo et al. 1999]. Isto evita que recursos sejam dispendidos para percorrer caminhos que contribuiriam com resultados de pouca qualidade. Nenhuma definição de critérios para avaliar a qualidade é apresentada porque essa definição depende do problema abordado por cada aplicação.

Capítulo 3

Praesto

Este capítulo apresenta um sistema de busca semântica chamado Praesto, cuja arquitetura proposta serve como base para o desenvolvimento de diferentes projetos. Praesto significa ‘ajudar’ ou ‘prover’ [1], em referência ao objetivo do trabalho de fornecer uma ferramenta de busca capaz de auxiliar o usuário.

O escopo deste trabalho cobre dois focos. O primeiro é a coleta e o gerenciamento de informações de contexto sobre os usuários. O segundo é a utilização dessas informações no processo de busca. O contexto de cada usuário pode fornecer ao sistema informações capazes de auxiliar na interpretação correta das suas intenções das buscas, sem que seja necessária a construção explícita de perfis de usuários. Porém, este processo também depende do uso de ontologias para a descrição formal de conteúdos compartilhados e para anotação semântica de recursos, como documentos de texto.

As questões relativas à geração da ontologia e base de conhecimento associada utilizada pelo sistema, a sua manutenção, o processo de anotação semântica utilizado pelo sistema ou possíveis mecanismos de inferência aplicados sobre a ontologia ou base de conhecimento durante a busca não estão compreendidos no escopo deste trabalho. Considera-se que estes recursos estão disponíveis na forma de ferramentas capazes de atender às demandas por funcionalidades e que podem ser acessadas através de interfaces definidas na arquitetura proposta para o sistema.

[1] http://en.wiktionary.org/wiki/praesto

As seções que seguem descrevem o sistema proposto. São apresentados requisitos não funcionais do sistema e, posteriormente, a organização conceitual proposta para o sistema, em forma de camadas. Além da organização conceitual, também são apresentados os componentes que implementam as funcionalidades do sistema e suas colocações dentro da organização conceitual previamente apresentada.

Após apresentar a organização do sistema proposto, o seu funcionamento é ilustrado, de maneira simplificada, de forma a prover uma visão de como os diversos elementos do sistema operam em conjunto. Finalmente, nas duas últimas seções deste capítulo, são apresentados detalhes da representação do contexto do usuário e de seu gerenciamento e utilização no processo de busca e apresentação dos resultados das mesmas.

3.1 Requisitos do Sistema

Nesta seção é apresentada uma breve descrição de requisitos não funcionais definidos para o sistema de busca semântica planejado. Estes requisitos são importantes, pois influenciam a organização do sistema, tanto conceitual, na forma de camadas, assim como a organização de seus componentes.

O objetivo esperado do sistema desenvolvido é um sistema de recuperação de informação capaz de aprender as intensões dos usuários, individualmente, baseado em informações sobre o contexto semântico do usuário.

Como uma das motivações do trabalho é a incompatibilidade entre o formalismo e objetividade das ontologias e a subjetividade dos usuários, o primeiro requisito é garantir a individualidade do usuário, com um sistema personalizado.

O segundo requisito é esconder a ontologia, ou qualquer outra forma de descrição generalizada e compartilhada do conhecimento, do usuário final. Considerando que o contexto do usuário é dinâmico, é necessario que o sistema seja flexível o suficiente para acompanhar as mudanças de interesse do usuário ao longo do tempo.

Com base nestes requisitos, uma organização do sistema em camadas foi proposta, organizando os componentes do sistema com base nos tipos de informação manipulada pelo sistema. As informações individuais de cada usuário são mantidas separadas das

informações compartilhadas por todos os usuários, neste caso, a ontologia e base de conhecimento utilizados para descrever o conteúdo, assim como também o conteúdo propriamente dito.

3.2 Organização Conceitual: Camadas

Esta seção apresenta a organização conceitual do Praesto projetada de forma a atender aos requisitos identificados para o sistema. Os requisitos resultaram em uma solução em que é feito o mapeamento da visão do usuário a respeito de assuntos pesquisados para a visão utilizada pelo sistema na descrição do conteúdo armazenado.

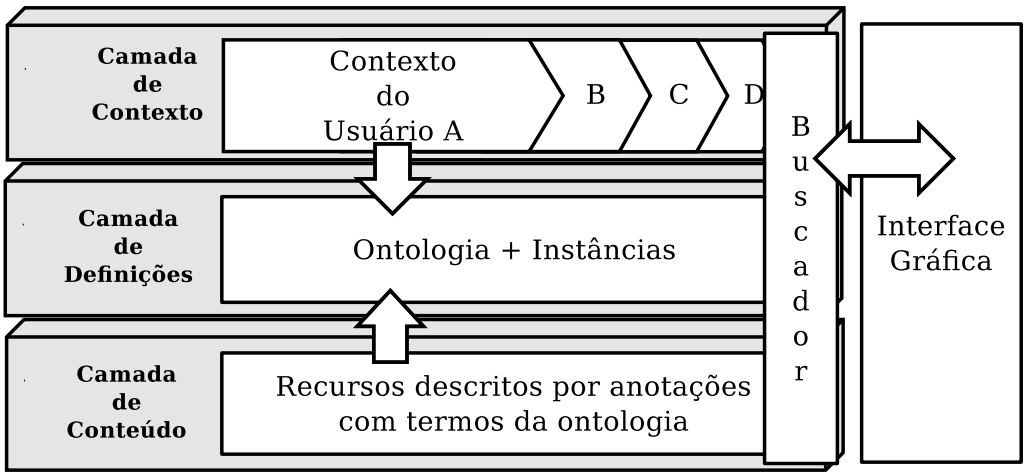

Figura 3.1: Arquitetura do sistema proposto. Escopo deste trabalho corresponde à Camada de Contexto e ao Buscador.

A Figura 3.1 ilustra a estrutura do Praesto, dividia em camadas, cada uma responsável por um tipo de informação. Juntamente com as camadas, são apresentados os componentes do sistema que as integram.

3.2.1 Camada de Conteúdo

Esta camada corresponde ao repositório de conteúdo do sistema. O conteúdo armazenado é tratado como um conjunto de recursos, anotados semanticamente de acordo com

os termos, sejam eles conceitos ou instâncias destes conceitos, disponíveis na ontologia da camada de definições. As anotações são utilizadas para descrever semanticamente os conteúdos armazenados, como também para indexá-los [Kahan et al. 2002].

A inserção de novos conteúdos no repositório é supervisionada por usuários especialistas capacitados. Estes usuários têm conhecimento sobre o domínio em questão e sobre a organização da ontologia utilizada para descrever este conhecimento. O processo de anotação semântica do conteúdo de acordo com a ontologia não é abordado no escopo deste trabalho.

3.2.2 Camada de Definições

A segunda camada do sistema corresponde a uma ontologia e base de conhecimento capazes de descrever formalmente diferentes assuntos (conceitos e instâncias destes conceitos) pertencentes a um domínio assim como as relações de que participam. A Figura 3.2 mostra uma parte de uma ontologia que descreve conceitos conhecidos pelo sistema. Na Figura 3.3 são mostrados alguns exemplos de instâncias destes conceitos.

```
<owl:Class rdf:about="http://dbpedia.org/ontology/PopulatedPlace">
<rdfs:label xml:lang="en">Populated Place</rdfs:label>
<rdfs:subClassOf rdf:resource="http://dbpedia.org/ontology/Place"/>
</owl:Class>

<owl:Class rdf:about="http://dbpedia.org/ontology/Place">
<rdfs:label xml:lang="en">Place</rdfs:label>
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>

<owl:Class rdf:about="http://dbpedia.org/ontology/Country">
<rdfs:label xml:lang="en">Country</rdfs:label>
<rdfs:subClassOf rdf:resource="http://dbpedia.org/ontology/PopulatedPlace"/>
</owl:Class>
```

Figura 3.2: Exemplo de parte da ontologia utilizada no sistema.

Essa ontologia deve ser escolhida ou ao menos conhecida pelos usuários especia-

```
<http://dbpedia.org/resource/%24100_Bill_Y%27all>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
 <http://dbpedia.org/ontology/MusicalWork> .

<http://dbpedia.org/resource/%2425_Million_Dollar_Hoax>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/ontology/TelevisionShow> .

<http://dbpedia.org/resource/%2435K_O.B.O.>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/ontology/Resource> .
```

Figura 3.3: Exemplo de parte da base de conhecimento utilizada no sistema.

listas responsáveis pelo gerenciamento do conteúdo. Sua finalidade é servir de base para a anotação semântica dos conteúdos armazenados no repositório. Ela também pode servir para realização de raciocínio (*reasoning*) por parte do sistema de busca, porém este assunto foge ao escopo deste trabalho.

3.2.3 Camada de Contexto

A camada de contexto é responsável por estender a descrição fornecida pela ontologia de forma a incorporar elementos subjetivos do conhecimento de cada usuário, que não podem ser representados pela ontologia.

A representação do contexto do usuário é feita na forma de um grafo. A construção deste grafo é gradual, ao longo de diversas interações do usuário com o sistema. Detalhes de como é feita a representação do contexto do usuário, assim como de que maneira esta representação auxilia no processo de busca são apresentadas no Capítulo 3.4.

3.3 Componentes do Sistema

Nesta seção são apresentados os dois principais componentes do sistema, o mecanismo de busca e a interface gráfica. O desenvolvimento da interface gráfica não é contemplado no escopo deste trabalho, mas informações a respeito podem ser obtidas

em [Fasolin et al. 2009].

3.3.1 Buscador

O mecanismo de busca utiliza as três camadas do Praesto. As informações sobre o contexto do usuário são usadas para para dois propósitos. O primeiro é para, se necessário, desambiguar palavras-chaves; o segundo é para expandir semanticamente as buscas, incluindo outros termos além dos relacionados diretamente às palavras-chave passadas pelo usuário, através da navegação pelo no grafo representando o contexto do usuário.

As definições fornecidas pela ontologia e base de conhecimento são utilizadas para representar o conhecimento compartilhado pelos usuários da forma como ele é armazenado e descrito pelo sistema. Já o conteúdo armazenado é acessado para recuperação dos resultados das buscas e informações sobre anotações semânticas.

3.3.2 Interface Gráfica

O desenvolvimento da interface gráfica pode contribuir bastante na satisfação do usuário. De acordo com [Koshman 2006], interfaces são desenvolvidas seguindo a hipótese que sistemas de visualização da informação são atrativos para pessoas porque eles exploram a eficiência das habilidades de processamento visual humanas.

A fim de explorar essa hipótese, a interface gráfica fornece, além dos habituais campos para as palavras-chave e listagem dos resultados, a visualização do contexto do usuário e a organização dos resultados de forma categorizada, de acordo com a organização dos conceitos fornecida pela ontologia para o domínio em questão. A interface gráfica é mostrada da Figura 3.4.

3.4 Funcionamento do Sistema

O Praesto funciona seguindo o fluxograma da Figura 3.5. Dada uma palavra-chave, o sistema verifica se há indicações na representação do contexto do usuário para inferir a sua intensão. Caso não houver, o sistema acessa a base de conhecimento e, se necessário,

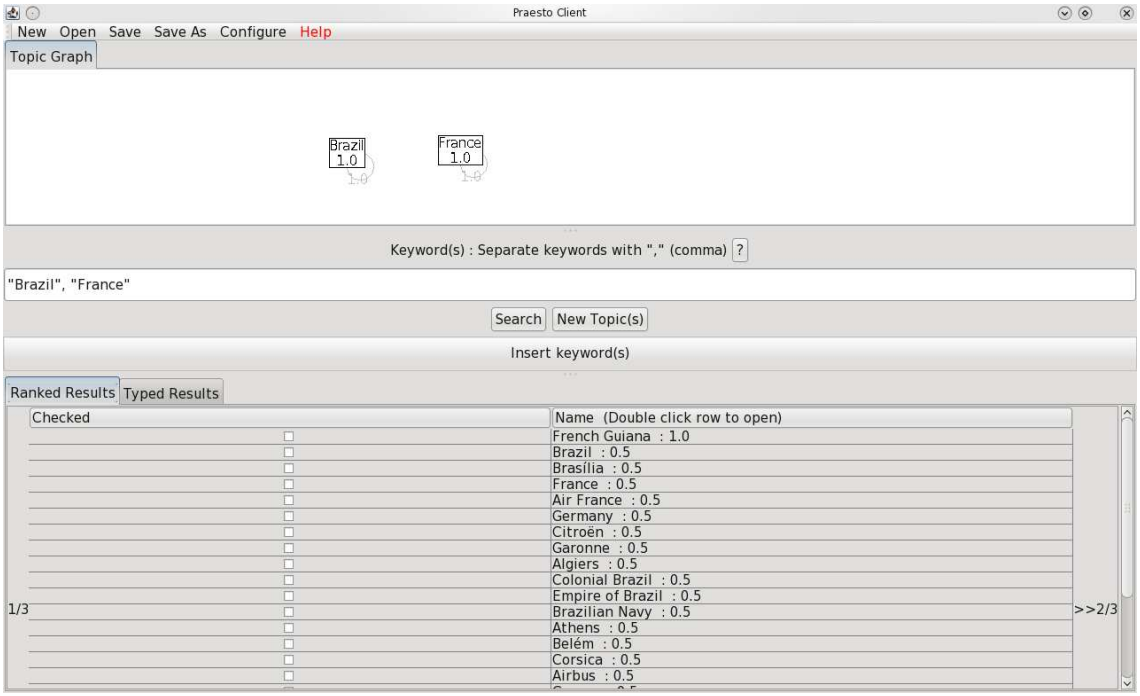

Figura 3.4: Interface Gráfica.

pede para o usuário desambiguar entre os possíveis usos para a palavra-chave descritos na base de conhecimento.

As definições associadas às palavras indicados pelo contexto ou desambiguados pelo usuário são utilizadas pelo sistema para recuperar o conteúdo, que é apresentado ao usuário. Informações sobre os resultados aprovados pelo usuário são retornadas ao sistema e usadas para atualizar as informações sobre o contexto do usuário.

As próximas seções descrevem em detalhes o modelo proposto para representar as informações de contexto do usuário e também fornecem uma descrição mais detalhada das diferentes etapas do processo ilustrado na Figura 3.5.

Este capítulo apresenta a proposta elaborada para a representação do contexto do usuário, relativo à semântica adotada durante a interação com o sistema. São apresentados seu uso na realização das buscas e também a captura das informações de contexto. Os algoritmos desenvolvidos são apresentados, descrevendo como as informações sobre o contexto são coletadas e posteriormente utilizadas para auxiliar o processo de busca, classificação e ordenação dos resultados.

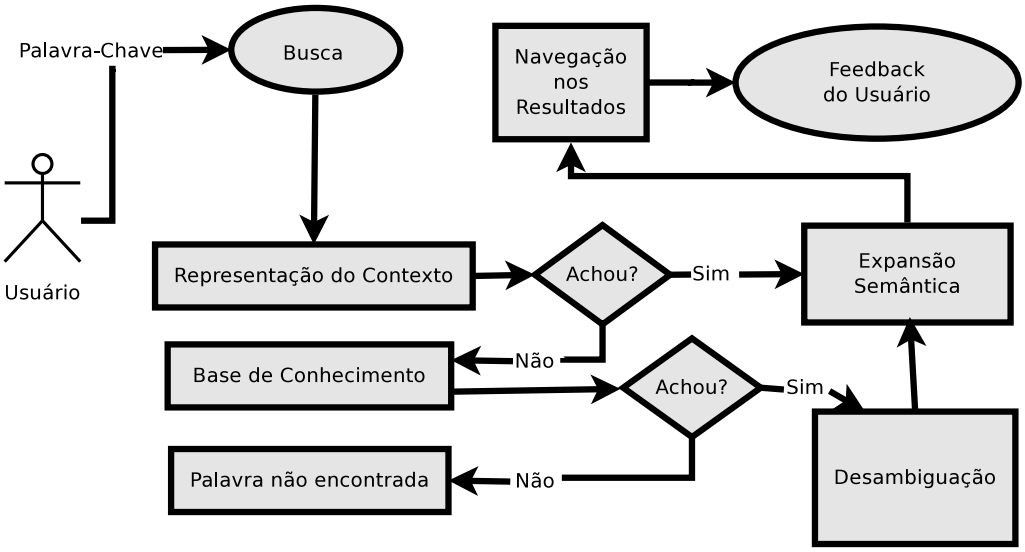

Figura 3.5: Fluxograma do funcionamento do Praesto.

3.5 Representação do Conhecimento Segundo o Usuário

A solução escolhida para representar o contexto do usuário, mais especificamente, sua percepção sobre a relevância de assuntos descritos na ontologia utilizada pelo sistema para a realização de buscas semânticas, é a utilização de um grafo $G(T, A)$ com pesos. A Definição 2 apresenta uma descrição formal deste grafo.

Definição 2: Grafo de Tópicos do Contexto do Usuário

O grafo de tópicos representando o contexto de um usuário é denotado por $G(T, A)$, onde:

T é um conjunto de tópicos representando assuntos possivelmente relevantes para o usuário

A é um conjunto de associações representando relações entre os tópicos de T

Um tópico é uma tripla $(nome, termo, peso)$ que associa a utilização da palavra $nome$ pelo usuário ao conceito ou instância ($termo$) de uma ontologia ou base de conhecimento BC que o sistema utiliza durante a realização de buscas semânticas, para descrever o domínio. O $nome$ deve pertencer ao conjunto V, representando o conjunto

de palavras utilizado para dar nome aos termos em BC. A Definição 3 descreve um tópico e também como os $nomes$ e os $termos$ podem representar relações de sinonímia e homonímia.

Definição 3: Tópico

Um tópico $t \in T$ é uma tripla

$$t = (nome_t, termo_t, peso_t)$$

Onde

$nome_t \in V$

$termo_t \in BC$

$peso_t \in [0, 1]$

O conjunto T de tópicos deve satisfazer as seguintes condições

$$\forall t_i \in T, \neg(\exists t_j \in T | nome_{t_i} = nome_{t_j} \wedge termo_{t_i} = termo_{t_j})$$

$$\forall\, nome \in V, \sum_{t \in h(nome)} peso_{t_i} = 1$$

Onde $h : V \rightarrow 2^T \,|\, h(nome) = \{t \in T \,|\, nome_t = nome\}$.

Os $peso$s dos tópicos são utilizados como indicativos do interesse do usuário em situações de homonímia. O valor 0 no $peso$ de um tópico indica provável ausência total de interesse por parte do usuário no tópico, enquanto 1 indica interesse exclusivo naquele tópico em detrimento dos outros tópicos homônimos. A soma do $peso$ de todos os tópicos com um mesmo $nome$ é sempre igual a 1.

O interesse do usuário por um tópico pode estar associado ao interesse por outro tópico. Este interesse pode ser indireto, fruto da utilização de diferentes figuras de linguagem (e.g. sinônimos), ou da co-ocorrência freqüente dos tópicos nas interações do usuário com o sistema. Uma associação entre dois tópicos $origem$ e $destino$ indica que estes tópicos possuem uma relação orientada no sentido de $origem$ para $destino$, independente da relação existir ou não na BC. A Definição 4 sintetiza a descrição das associações.

Definição 4: Associação

Uma associação $a \in A$ é uma tripla

$a = (origem_a, destino_a, peso_a)$

Onde

$origem_a, destino_a \in T$

$peso_a \in [0, 1]$

O conjunto A de associações deve satisfazer a seguinte condição

$\forall\, t \in T, \sum_{a \in o(t)} peso_a = 1$

Onde $o : T \rightarrow 2^A \mid o(t) = \{a \in A \mid origem_a = t\}$.

O $peso$ de uma associação é utilizado para indicar o grau de relevância que o usuário atribui à relação entre dois tópicos que ela representa. Tal como ocorre com os tópicos, o valor 0 indica provável ausência de qualquer relevância, por parte do usuário, para a associação entre os tópicos $origem$ e $destino$, enquanto o valor 1 indica uma associação possivelmente relevante, em comparação com as outras associações com a mesma $origem$. Os pesos das associações são calculados em função dos seus tópicos de $origem$, de forma que a soma total do $peso$ de todas associações partindo de um mesmo tópico também é sempre igual a 1.

O cálculo dos $pesos$ dos tópicos e das associações depende de duas funções, $h : V \rightarrow 2^T$ (homôninos) e $o : T \rightarrow 2^A$ (origem). Estas funções mapeiam, respectivamente, uma palavra a um conjunto de tópicos homônimos e um tópico ao conjunto de associações que partem do mesmo. Os valores dos $pesos$ destes tópicos ou associações são normalizados no intervalo $[0, 1]$, para facilitar a comparação entre tópicos homônimos e entre associações com mesma origem. Essa normalização também auxilia o usuário a entender como a heurística de busca adotada é influênciada pelo seu grafo.

A relação entre os tópicos, associações e como eles são mapeados para o conjunto de $termos$ usados pelo sistema é mostrada na Figura 3.6. Na figura, os hexágonos representam os tópicos, que são identificados por palavras e referenciam $termos$ (as instâncias representados como elipses e os conceitos como retângulos). No exemplo da figura é possível ver que, apesar de haver mais de um tópico com o mesmo nome (’São Paulo’), cada um referencia um termo diferente.

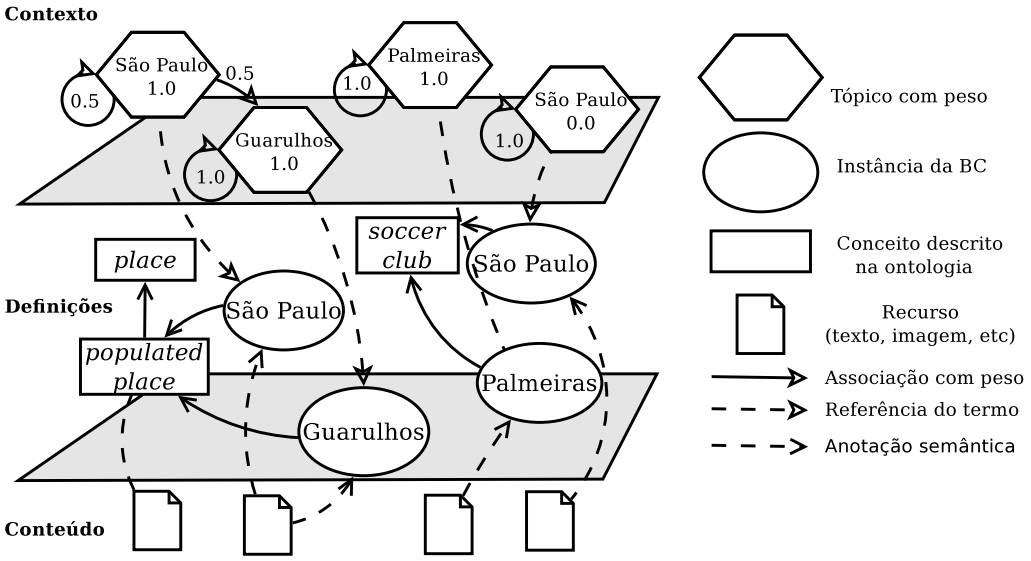

Figura 3.6: Exemplo de um possível grafo de tópicos para o usuário A do Exemplo 1 (interessado em cidades). O grafo esconde a BC do usuário.

É importante destacar que, apesar da possível existência de associações cuja $origem$ e $destino$ se referem ao mesmo tópico, estas associações não são percorridas durante as buscas. O seu propósito é manter a consistência com a definição de que a soma dos $pesos$ de todas associações partindo de uma mesma origem deve ser igual a 1. Caso não houvesse este tipo de associação, o grafo do usuário poderia representar erroneamente os interesses do usuário, já que não representaria a possibilidade do usuário, ao ter interesse por um tópico t_{origem}, não ter interesse pelos tópicos alcançaveis a partir de t_{origem}. Isto pode ser verificado na Figura 3.6 no tópico referente a 'São Paulo' que possui uma associação para ele mesmo e outra para o tópico 'Guarulhos'.

3.6 O Processo de Busca

Esta seção apresenta o algoritmo de busca desenvolvido. Este algoritmo adapta a meta-heurística *ACO* ao problema de buscas por palavras-chave, a fim de compor o mecanismo de busca dirigido pelo contexto semântico do usuário, que é representado através do grafo de tópicos. O grafo de tópicos corresponde à representação dos rastros de

ferormônio da heurística *ACO*. As correspondências entre diversos elementos envolvidos na solução (a colônia, formigas, rastros de ferormônio, etc) e os elementos da arquitetura proposta para o sistema de busca são apresentadas a seguir.

- `Formiga:` Cada formiga do sistema corresponde a uma excecução de uma travessia de um percurso no grafo de tópicos, utilizado para expandir semanticamente as palavras-chave buscadas. A qualidade dos percursos percorridos no grafo é indicada pela intensidade dos rastros de ferormônios.

- `Ferormônio:` Os rastros de ferormônios correspondem aos $pesos$ dos tópicos e das associações nos caminhos percorridos. Estes rastros permitem que os caminhos percorridos pelas formigas partindo do formigueiro sejam persistidos para reuso futuro, por outras formigas.

- `Entrada/Saída do Formigueiro:` O formigueiro corresponde ao local de onde as formigas partem em busca de comida, podendo possuir diversas entradas e saídas. No Praesto, dado uma busca do usuário por uma palavra-chave, as entradas do formigueiro correspondem ao conjunto de tópicos cujas ocorrências na ontologia correspondem lexicamente às palavras chaves buscadas pelo usuário e cujos pesos indicam suas relevâncias. Ou seja, cada um destes tópicos corresponde a uma entrada do formigueiro adequada para a busca (adequação avaliada pela correspondencia do tópico à palavra-chave e pelo peso do tópico), do qual as formigas podem iniciar buscas levando até uma fonte de comida.

- `Fonte de comida:` As fontes de comida correspondem aos $termos$ da ontologia ou da base de conhecimento referenciados pelos tópicos considerados relevantes para a busca, que são os tópicos percorridos durante as travessias no grafo com propósito de expandir semanticamente a busca. Ao vistar um termo da ontologia (indiretamente, através de um tópico que o referencia), as expansões retornam a comida ali armazenada, que corresponde ao conteúdo armazenado anotado semanticamente por este termo.

- `Comida:` A comida corresponde aos recursos anotados semanticamente, através de ligações com um ou mais termos da ontologia. A qualidade dos recursos (comida) é avaliada pelo valor dos pesos no caminho percorrido pelas expansões (a intensidade dos rastros de ferormônios deixados nos caminhos percorridos pelas formigas). Como a indicação sobre a qualidade do recurso depende do usuário e da sua visão global de todos resultados retornados (diferente de cada formiga que tem uma visão local restrita à sua vizinhança), a responsabilidade da colocação dos ferormônios é atribuída ao *daemon*.

- *`Daemon:`* O *daemon* corresponde aos mecanismos para interação do usuário com o sistema, incluindo a coleta do *feedback* do usuário. Ele possui uma visão global de todos elementos envolvidos no processo de busca e gerenciamento da informação de contexto do usuário. O *daemon* é responsável por observar os resultados escolhidos pelo usuário, analisar suas anotações semânticas e realizar a marcação dos rastros de ferormônio, através da atualização dos pesos das associações e dos tópicos.

A utilização de um *daemon* é necessária porque os resultados são recuperados através de diversos caminhos percorridos no grafo de tópicos do usuário a fim de se expandir a busca. Os resultados retornados através de diferentes caminhos são avaliados coletiva e simultaneamente. A avaliação da qualidade dos resultados é feita com base nas indicações fornecidas pelo usuário, ao selecionar os resultados de seu interesse. Cabe ao *daemon* organizar os resultados coletados pelas diversas formigas e apresentá-los para o julgamento pelo usuário. Esta organização inclui conciliar diferentes resultados retornados na forma de entradas únicas na lista de resultados (evitar que um mesmo resultado seja apresentado ao usuário múltiplas vezes, caso seja recuperado por meio de múltiplos caminhos percorridos). O *daemon* também analisa as freqüências e co-ocorrências entre termos utilizados para anotar os recursos selecionados pelo usuário. Os resultados são utilizados para determinar os incrementos nos pesos dos tópicos e associações do grafo de tópicos.

A coordenação entre esses diferentes componentes do sistema também mantém correspondência com a organização original proposta pelos algoritmos *ACO*, sendo realizada

Algoritmo 2: Organização das diferentes fases do algoritmo

```
   Entrada: PC[], δ ;                                   // palavras-chave e índice de atenuação
   Dados: G(T,A), Ont, repositório
1  início Algoritmo Busca Contextual Estigmérica
2      resultados = busca(PC[]);
3      atenuação(índice δ) ;                                                  // δ ∈ [0, 1]
4      manutenção(coleta_feedback(resultados));
5  fim
```

em três etapas. As etapas são busca, atenuação dos $peso$s e manutenção do grafo de tópicos do usuário, conforme apresentado no Algoritmo 2. Estas etapas correspondem, respectivamente, às etapas de geração de formigas e atividades, de evaporação do ferormônio e de ações do *daemon*. As diferentes etapas são apresentadas individualmente e em detalhes nas próximas seções.

Para facilitar a compreenção do funcionamento do sistema, a descrição de cada etapa do processo de busca é ilustrada com imagens capturadas da implementação do Praesto, durante diferentes estados do sistema, na execução cenário de busca descrito no Exemplo 2, mostrado novamente a seguir.

- Um usuário, sem muitos conhecimentos geográficos, fica sabendo que há um território da França (a Guiana Francesa) que faz fronteira com o Brasil. O usuário pode pesquisar por Brasil. Mas Brasil é, além do nome do país, nome de filmes e músicas. Já França pode trazer muitos resultados relativos a diferentes momentos históricos, por exemplo, resultados associados à França na idade média, antes mesmo da ocupação da América.

Devido às limitações de escopo do projeto e ao conjunto de dados utilizados, as palavras-chave das figuras referentes ao exemplo não se referem a conceitos, somente a instâncias. Também, todo o conteúdo, incluindo as palavras-chave, está em inglês. Detalhes sobre a implementação são descritos no Capítulo 4.

A etapa de busca é a primeira realizada pelo sistema. O algoritmo da etapa de busca recebe como entrada a consulta do usuário, na forma de um conjunto de palavras-chaves. O processo de busca descrito neste trabalho considera que qualquer tratamento da consulta, por exemplo, remoção de *stop-words* ou identificação de palavras compostas, já

Figura 3.7: Usuário fornece as palavras-chave.

foram realizados.

Seguindo o Exemplo 2, considere que o usuário busca pelas palavras-chave ‘Brazil’ e ‘France’ (Figura 3.7), já que o objetivo da sua busca está relacionado a estes dois países.

O Algoritmo 3 descreve a etapa de busca a partir do recebimento das palavras-chave fornecidas pelo usuário. Dada uma palavra-chave, o sistema busca, no grafo de tópicos representando o contexto do usuário, tópicos correspondentes às palavras-chave. Isto significa que, dada uma palavra-chave $palavra$, é necessário que o grafo de tópicos possua no mínimo um tópico cujo $nome = palavra$ (conforme o teste realizado na linha 5 do Algoritmo 3) para que o Praesto possa usar o grafo de tópicos do usuário para processar a busca por *palavra*.

Quando essa condição não é atendida, o sistema busca correspondências presentes entre $palavra$ e os termos na BC, mais especificamente, nas palavras utilizadas para rotulá-los (linha 6 do Algoritmo 3). Porém, é provável que nem todo o termo cujo rótulo corresponde à $palavra$ seja do interesse do usuário. Para verificar suas relevâncias, o sistema deve questionar o usuário quanto à sua intenção ao utilizar $palavra$. Isto é necessário por não haver informações de contexto do usuário relacionadas ao seu uso (não há tópicos com $nome = palavra$).

Desambiguação

Este processo de desambiguação é mostrado na Figura 3.8. Seguindo o Exemplo 2, o usuário seleciona as denotações para ‘Brazil’ e ‘France’ referentes aos países atuais, dentre outras possibilidades, inclusive opções referentes a outros períodos históricos passados, como ilustrado na Figura 3.8.

O sistema realiza essa desambiguação apresentando ao usuário todos os termos ro-

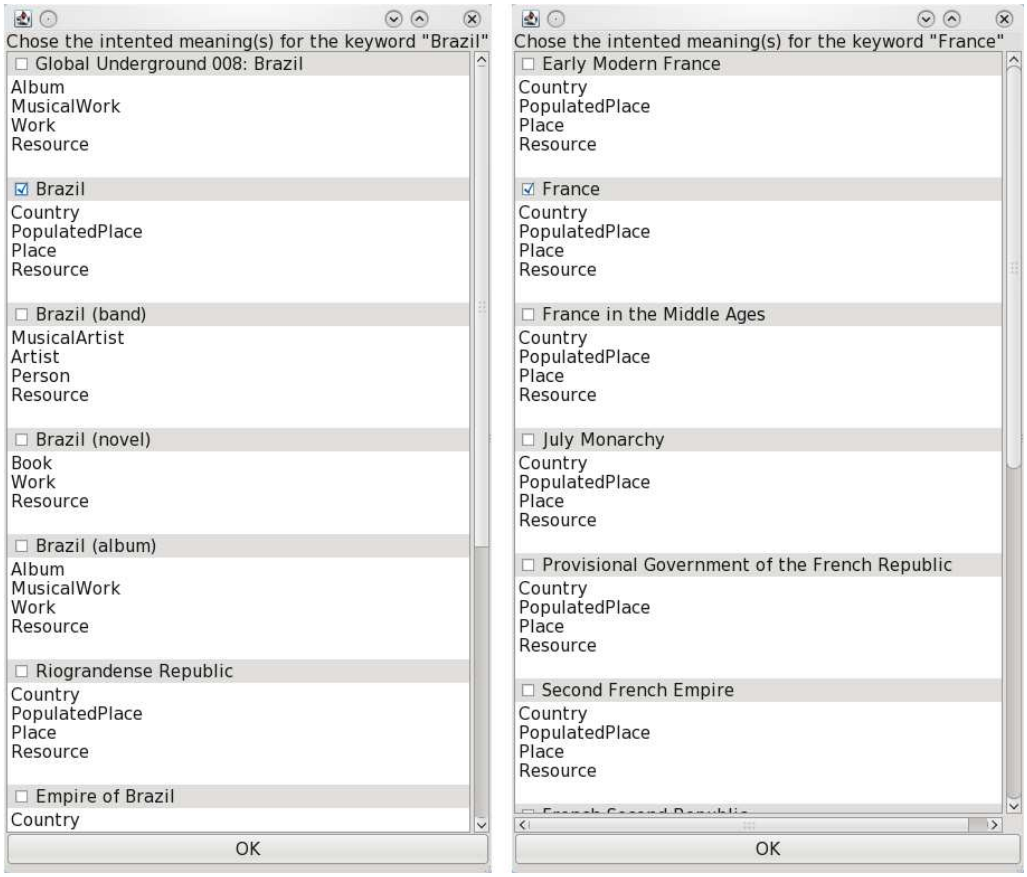

Figura 3.8: Desambiguação das palavras-chave: O usuário indica denotações que correspondem à sua intenção de busca.

Algoritmo 3: Etapa de busca

```
1  início Busca(PC[])
2      tópicos_semente[] = ∅ ;
3      Visitados[] = ∅ ;
4      para cada palavra em PC[] faça
5          se ¬(G.tem_tópico_chamado(palavra)) então
6              denotações[].adiciona( Ont.termos_rotulados_com(palavra));
7              possíveisIntensões[].adiciona(consulta_o_usuário(denotações[]));
8              G.cria_tópicos_para(possíveisIntensões[]);
9          fim
10         tópicos_semente[].adiciona(G.tópicos_homônimos_a(palavra));
11     fim
12     para cada tópico em tópicos[] faça
13         Visitados[].adiciona(expande (tópico));                    // segue ferormônio/explora
14     fim
15     para expansão em Visitados[] faça
16         Resultados[] = ∅;
17         Resultados[].adiciona(recuperação(expansão));
18     fim
19     retorna Resultados[];
20 fim
```

tulados com $palavra$, para que ele indique as que considera relevantes (linha 7 do Algoritmo 3). Para cada termo indicado pelo usuário, o sistema cria um novo tópico, como ilustrado na Figura 3.9. Este novo tópico tem $nome = palavra$ e o seu $termo$ é uma referência ao termo em BC (linha 8) rotulado com $palavra$ que foi escolhido pelo usuário. O $peso$ atribuído ao novo tópico é $peso = 1/quantidade$, onde $quantidade$ é a quantidade de tópicos criados na desambiguação de $palavra$.

Tendo questionado o usuário e criado os novos tópicos, a condição de existência de tópicos correspondentes à palavra-chave passa a ser atendida e o sistema pode prosseguir a busca. Existe a possibilidade de não existir nenhuma correspondência com $palavra$, seja nos tópicos do grafo ou nos termos da ontologia. O tratamento deste caso não é contemplado neste trabalho, já que ele é resultado da ausência de uma ou mais palavras na ontologia, a qual não é o foco deste trabalho. Possíveis soluções podem envolver o armazenamento dessas palavras para sua inserção na ontologia.

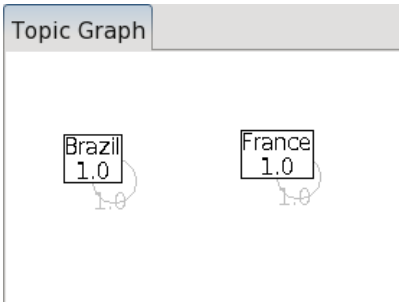

Figura 3.9: Tópicos criados para as denotações escolhidas para as palavras-chave.

Expansão Semântica

Após identificar os tópicos correspondentes às palavras-chaves, o sistema inicia o processo de expansão semântica destes tópicos (linha 13 do Algoritmo 3). Dado um conjunto de palavras-chave PC usado para especificar uma busca, para cada palavra-chave $palavra_i \in PC$ o sistema seleciona tópicos sementes, de onde terá inicio uma expansão semântica, com base no contexto do usuário. A Definição 5 descreve a relação entre o conjunto de palavras-chave e os conjuntos de tópicos sementes. As informações referentes ao processo de expansão semântica são armazenadas em estruturas de dados chamadas $expansões$, conforme a Definição 6.

Definição 5: Sementes

Seja PC um conjunto de palavras-chave especificando uma busca do usuário.

Para cada $palavra_i \in PC$, $\exists$ um conjunto $Sem_{palavra_i}$ de tópicos sementes

$$Sem_{palavra_i} = \{ t \in T \mid nome_t = palavra_i \wedge peso_t < peso\_mínimo\_do\_tópico\_semente \}.$$

Definição 6: Expansão (estrutura de dados, referenciada como expansãoED nos algoritmos)

Seja exp uma expansão:

$exp = (origem\_exp, destino\_exp, relevância\_exp)$ onde:

$semente$ é um tópico semente $t_{semente_i}$

$origem\_exp$ é uma outra $expansão$

$destino\_exp$ é o um tópico visitado no processo de expansão semântica

$relevância\_exp = relevância_{destino\_exp} \in [0, 1]$.

Após obter os conjunto de tópicos sementes para todas as palavras-chave, o sistema inicia o processo de expansão semântica, uma palavra-chave de cada vez. Então, dada uma palavra-chave $palavra$ e o seu respectivo conjunto $Sem_{palavra}$ de tópicos sementes, cada tópico $t_{semente_i} \in Sem_{palavra}$ é expandido semanticamente.

Nesse processo de expansão, o grafo de tópicos é percorrido com uma busca em profundidade com aprofundamento iterativo (BPAI), onde a busca somente prossegue à uma profundidade maior após visitar todos tópicos na profundidade atual, o que resulta na mesma ordem de visitas de busca em largura. Esta busca BPAI segue as associações partindo de $t_{semente_i}$ e cada tópico visitado recebe um valor $relevância \in [0, 1]$, que mede a relevância inferida pelo sistema para o $termo$ do tópico, em relação à palavra-chave $palavra$ usada para especificar a busca do usuário.

O processo de atribuição das $relevâncias$ aos tópicos percorridos e o aprofundamento da BPAI dependem de duas etapas distintas, que estão associadas respectivamente à desambiguação e à expansão semântica. A primeira é o cálculo do valor máximo para $relevância$ dos tópicos percorridos, o segundo é o seu cálculo propriamente dito.

O valor máximo para a $relevância$ indica o valor máximo que pode ser atribuído à $relevância$ de um tópico. Este limite depende da certeza do sistema quanto ao interesse do usuário por uma denotação para $palavra$ em detrimento de outra. Nenhum tópico visitado na busca partindo de $t_{semente_i}$ pode receber uma $relevância$ maior que $peso_{t_{semente_i}}$.

O Praesto calcula o valor $relev_{t_{percorrido}}$, referente à $relevância$ de cada tópico percorrido a partir de $t_{semente_i}$ multiplicando o $peso$ das associações percorridas de $t_{semente_i}$ até $t_{percorrido}$. O processo de expansão semântica é descrito no Algoritmo 4.

O Algoritmo 4 descreve o processo de expansão semântica a partir de um tópico semente $tópico\_semente$ usado como ponto de partida. Este algoritmo realiza a busca com aprofundamento iterativo. Todas as informações referentes ao processo de expansão que

Algoritmo 4: BPAI - Busca em Profundidade com Aprofundamento Iterativo

```
1  início expande(tópico_semente)
2      se tópico_semente.peso > peso_mínimo_do_tópico_semente então
3          exp = nova expansãoED(tópico_semente, tópico_semente, ∅, tópico_semente.peso);
4          Por_Expandir[].adiciona(exp);
5          Tópicos_Visitados[tópico_semente] = exp ;        // indice = t ∈ T e valor = expansão|destino_exp = t
6          temporario[] = ∅ ;
7          profundidade = 0;
8          enquanto profundidade < profundidade_máxima_da_busca faça
9              enquanto ¬(Por_Expandir[] = ∅) faça
10                 exp_atual = por_expandir[].remove_primeiro;
11                 tpc_atual = exp_atual.destino_exp;
12                 se tpc_atual.relevância ≥ relevância_mínima_do_tópico então
13                     se (Tópicos_Visitados[tpc_atual] ≠ ∅ ∧ tpc_atual.relevância ≤
                       Tópicos_Visitados[tpc_atual].relevância_exp) então
14                         exp_atual = Tópicos_Visitados[tpc_atual];
15                     fim
16                     temporário[] = temporário[] + expande_largura(exp_atual);
17                     Tópicos_Visitados[tpc_atual] = exp_atual;
18                 fim
19                 Por_Expandir[] = Por_Expandir[] + temp[];
20                 profundidade = profundidade + 1;
21             fim
22         fim
23     fim
24     retorna tópicos_visitados[];
25 fim
```

serão necessárias em etapas futuras são armazenadas como estruturas de dado $expansões$. Na linha 2 o algoritmo cria uma $expansão\ exp \mid origem\_exp_{exp} = tópico\_semente$, $destino\_exp = tópico\_semente$, $relevância\_exp = relevância_{tópico\_semente}$. O tópico origem e destino de exp são os mesmos porque, até este momento, nenhuma associação foi percorrida. O valor de $relevância\_exp_{exp}$ é o valor de $relevância_{tópico\_semente}$ pelo mesmo motivo.

O valor da $relevância$ de cada tópico visitado é calculado pelo Algoritmo 5, que também realiza a navegação no grafo. Este algoritmo recebe um tópico, passado na forma de uma $expansão\ exp\_atual$. O algoritmo verifica quais outros tópicos podem ser alcançados a partir de $destino\_exp_{exp\_atual}$ e cria uma nova $expansâo$ para cada um destes tópicos, sendo que $origem\_exp = exp\_atual$ e $destino\_exp$ é o tópico sendo visitado. Assim as $expansôes$ formam uma estrutura de dados recursiva pela qual pode se fazer o caminho inverso ao percorrido durante uma expansão semântica, o que será necessário em etapas futuras.

Algoritmo 5: Expansão

```
1  início expande_largura(exp_atual)
2      novas_expansões[] = ∅;
3      tópico = exp_atual.destino_exp;
4      para cada associação em tópico.associações_partindo faça
5          se associação.peso ≥ peso_mínimo_da_associação então
6              tópico_destino = associação.destino;
7              tópico_destino.relevância = tópico.relevância * associação.peso;
8              nova_exp = expansão(tópico, tópico_destino, tópico_destino.relevância);
9              novas_expansões[].adiciona(nova_exp) ;
10         fim
11     fim
12     retorna novas_expansões[]
13 fim
```

No caso de um mesmo tópico ser visitado, para uma mesma palavra-chave, mais de uma vez, é considerada somente a expansão semântica que atribuiu maior $relevância$ ao tópico. Esta verificação é realizada no Algoritmo 4, utilizando a coleção $Tópicos\_Visitados$ que associa um tópico à $associaão$ com valor $relevância\_exp$ a percorrê-lo. Essa verificação evita que a expansão semântica passe a percorrer o grafo formando ciclos, prejudi-

cando a eficiência do processo de busca.

O processo de expansão semântica da busca depende de algumas condições, que são impostas por valores de *thresholds*. Estes valores são:

- `peso_mínimo_do_tópico_semente`: A busca somente é iniciada para um determinado tópico semente $t_{semente}$ se $peso_{t_{semente_i}}$ for maior que um valor mínimo. Isto evita que o sistema dispenda tempo e recursos processando uma busca que conhecidamente vai resultar em tópicos com valor de $relevância$ reduzidos.

- `peso_mínimo_da_associação:` A busca somente percorre uma associações se seu $peso$ for maior que um valor mínimo, pelo mesmo motivo do item anterior.

- `profundidade_máxima_da_busca:` A busca não se aprofunda além de uma profundidade máxima de $t_{semente_i}$. Desta maneira pode-se restringir o espaço de busca, dependendo do tamanho do grafo de tópicos, de acordo com as necessidades do usuário.

- `relevância_mínima_de_um_tópico:` Quando um tópico recebe $relevância$ com valor abaixo de um valor mínimo, a busca não é aprofundada a partir deste tópico, pelo mesmo motivo dos dois primeiros itens.

A Figura 3.10 e a Figura 3.11 mostram os caminhos percorridos durante a busca pelas palavras-chave, porém em dois momentos diferentes (duas interações diferentes de todo processo de busca). Na Figura 3.10 o grafo do usuário possui somente os tópicos criados logo após a desambiguação das palavras-chave, assim, os caminhos percorridos pelas expansões partem dos tópicos origem (correspondentes à cada palavra-chave) em direção a eles mesmos.

Tendo completado a expansão semântica de todos os tópicos sementes, é obtido um conjunto $Visitados$ de tópicos visitados, juntamente com suas respectivas $relevâncias$, os quais são utilizados pelo sistema para recuperar o conteúdo considerado como relevante para o usuário.

Figura 3.10: Grafo de tópicos rescém criado, com associações percorridas durante expansão semântica.

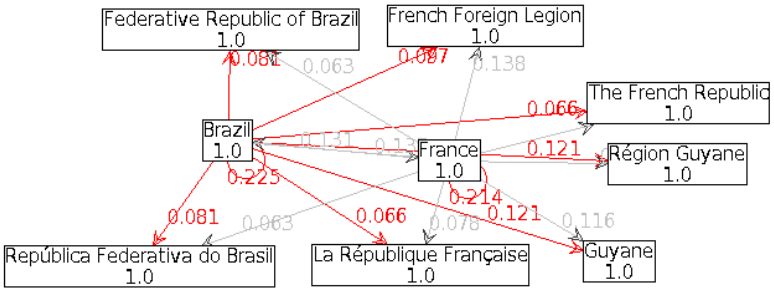

Figura 3.11: Grafo de tópicos com associações percorridas durante expansão semântica.

Recuperação do Conteúdo e *Ranking* dos Resultados

Para cada tópico $t \in Visitados$ o sistema retorna, da camada de conteúdo do Praesto, todos os conteúdo anotados semanticamente por $termo_t$. O valor $relevância_t$ é usado para ordenar os conteúdos retornados no $ranking$ apresentado ao usuário.

Contudo, um mesmo conteúdo pode ser anotado por vários termos da base de conhecimento, os quais podem ser referidos por diversos tópicos, diversas vezes, já que um mesmo tópico pode ser percorrido por mais de um processo de expansão. Por isso, cada conteúdo recuperado é associado, via sua *URI* a um $item$ (Definição 7) retornado como resultado. O $item$ também armazena as estruturas de dados $expansões$ que levaram ao conteúdo ao qual ele se refere. Os resultados são apresentados ao usuário através do ordenamento dos diversos $items$, que são organizados em um *ranking* de acordo com os seus valores $ranking\_normalizado$. A recuperação dos conteúdos no repositório, a criação dos $itens$ e o seu ordenamento são realizados pelo Algoritmo 6.

Definição 7: Item (item de resultado)

Seja $item$ um item

$item = (conteúdo, ranking\_normalizado, Expansõoes)$ onde:

$conteúdo$ é o conteúdo recuperado como resultado, identificado na camada de conteúdo do Praesto por uma URI $uri\_conteúdo$

$ranking\_normalizado$ é um valor $\in (0, 1]$ usado para ordenar o item junto a outros itens durante a apresentação dos resultados ao usuário.

$Expansões$ é um conjunto que armazena as estruturas de dados referentes a todas as expansões semânticas que resultaram na recuperação do $conteúdo$ do item.

Algoritmo 6: Recuperação

```
1  início Recuperação
2      itens[] = ∅;
3      total = 0; máximo = 0;  para cada expansão em expansões[] faça
4          tópico = expansão.ultimo_tópico();
5          termo = tópico.termo();
6          conteúdos[] = repositório.retorna_conteudos_descritos_por(termo);
7          para cada conteudo em conteudos[] faça
8              itens[conteúdo].peso = itens[conteúdo].peso + expansão.relevância_exp;
9              itens[conteúdo].adiciona_expansão(expansão);
10         fim
11         total = total + expansão.relevância_exp;
12     fim
13     para cada item em itens[] faça
14         item.ranking_não_normalizado = item.peso/ total;
15         máximo = máximo_entre(máximo, item.ranking_não_normalizado);
16     fim
17     para cada item em itens[] faça
18         item.ranking_normalizado = item.normaliza_ranking(máximo);
19     fim
20     retorna itens[];
21 fim
```

O $item$ acumula a $relevância$ de todas $expansões$ que levaram até $conteúdo_{item}$. Esse valor é dividido pelo valor acumulado de todas $expansões$ criadas no processo de busca. Depois, o valor calculado é normalizado de forma que o resultado calculado como sendo o mais relevante para o usuário tenha um valor $ranking\_normalizado$ igual a 1.

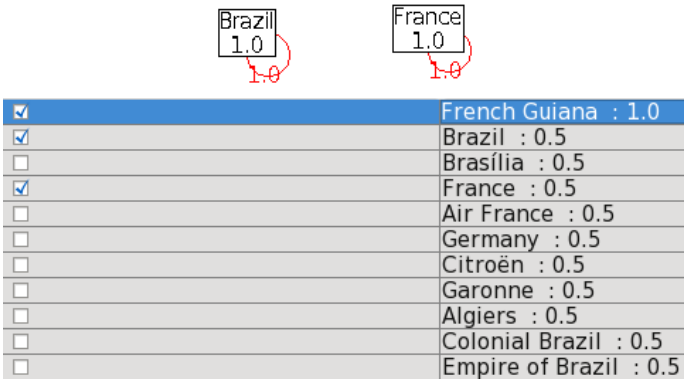

Figura 3.12: Resultados de uma busca utilizando quantidades reduzidas de informações de contexto do usuário.

Na Figura 3.12, dentre os resultados retornados, ‘French Guiana’ foi retornada tanto pela busca por ‘Brazil’ como pela busca por ‘France’ e por isso, no *ranking* dos resultados, é apresentada antes que os outros resultados, que correspondem a somente uma das palavras-chave.

Na Figura 3.13 é mostrada uma busca pelas mesmas palavras-chave, porém guiada por um grafo de tópicos com mais informações sobre o contexto do usuário. De acordo com a imagem, o resultado ‘French Guiana’ foi retornado por $expansões$ que também percorreram os tópicos ‘Brazil’ e ‘France’, correspondentes às palavras-chave, porém foi retornado pelos $termos$ referenciados por outros tópicos, ‘Federative Republic of Brazil’, ‘Republica Federativa do Brasil’, ‘French Foreign Legion’, ‘The French Republic’, ‘La République Française’, ‘Guyane’ e ‘Région Guyane’, todos alcançados via associações originadas no tópico ‘Brazil’.

Dentre os resultados listados na Figura 3.13 o sistema atribuiu uma posição melhor no ranking justamente ao objetivo do usuário, ‘French Guiana’, assim como aos resultados relacionados com ‘Brazil’ e ‘France’ referentes à atualidade (em detrimento de resultados relacionados ao Brasil Colonial (’Colonial Brazil’) ou Brasil Império (’Empire of Brazil’)). Resultados como ‘Chamber of Deputies of Brazil’ e ‘Brazilian Army’, que têm relação com ‘Brazil’ mas não com o objetivo da busca do usuário foram retornadas, porém

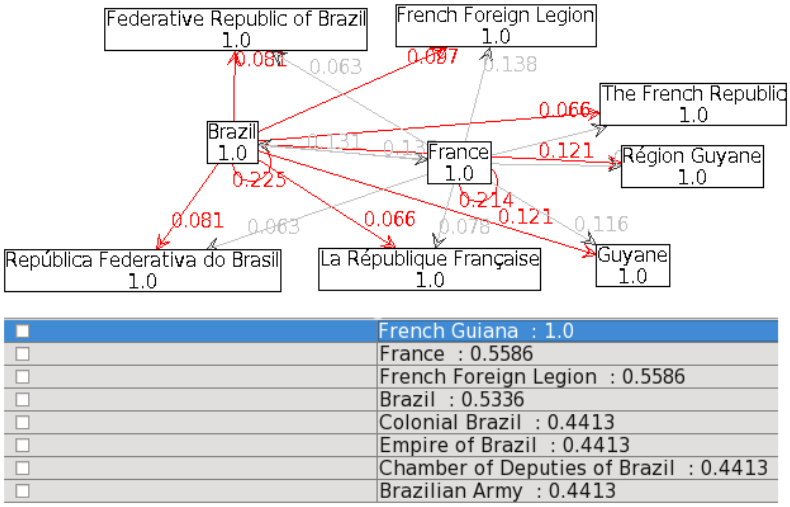

Figura 3.13: Resultado de uma busca utilizando grafo com informações de contexto coletadas de interações prévias.

classificadas como menos importantes quanto à sua provável relevância para o usuário.

3.6.1 Atenuação - Evaporação do Ferormônio

Com a passagem do tempo, ao longo de diversas interações entre o usuário e o sistema, os interesses dos usuários mudam. Alguns tópicos antes importantes para alguma atividade realizada pelo usuário podem deixar de ser relevantes e novas prioridades podem surgir. Esta evolução dos interesses, e conseqüente depreciação de alguns tópicos, deve ser reproduzida na representação do contexto utilizada pelo sistema.

Algoritmo 7: Etapa de atenuação dos pesos

```
1 início atenuação(δ)
2     para cada tópico em T faça
3         tópico.peso = tópico.peso ∗ δ;
4     fim
5     para cada associação em A faça
6         associação.peso = associação.peso ∗ δ;
7     fim
8 fim
```

Essa evolução não é importante somente a longo prazo, mas também durante a duração da interação do usuário com o sistema, no decorrer de uma consulta. Conforme resul-

tados são apresentados ao usuário, seu conhecimento sobre o assunto buscado é acrescido de novas informações presentes nesses resultados, o que pode alterar sua percepção sobre o tema em questão ou influenciar seus interesses.

O efeito da passagem do tempo no contexto do usuário é simulado a cada vez que o sistema apresenta os resultados ao usuário e este fornece um *feedback* em relação a que resultados o agradaram. Toda vez que isto ocorre, os $pesos$ de todos os tópicos e associações no grafo do usuário são atenuados, conforme descrito no Algoritmo 7. A intensidade da atenuação depende do parâmetro δ, no intervalo $[0, 1]$, sendo que 0 implica na ausência de qualquer memória passada e 1 implica na persistência dos interesses passados dos usuários para sempre.

3.6.2 Manutenção - Ações do Daemon

Durante a etapa de busca, o sistema tem uma visão local, restrita a cada tópico semente onde uma expansão semântica começou e aos tópicos ligados a estes por meio de associações. Como um mesmo recurso do repositório pode ser recuperado por mais de uma expansão semântica, é necessária uma visão global do sistema durante a realização da etapa de manutenção.

Nesta etapa o sistema trabalha com os $itens$ retornados pelo sistema, mais especificamente, os marcados como relevantes pelo usuário. Dado um conjunto $Selecionados$ de $itens$ marcados pelo usuário como sendo resultados relevantes, o sistema percorre todos $itens\ item_i \in Selecionados$ e calcula o valor $cardinalidade$ para cada $termo$ utilizado para anotar os $conteúdos$ destes resultados. A Definição 8 define o valor $cardinalidade$ como a quantidade de vezes que um mesmo $termo$ é usado para anotar $conteúdos$ considerados como sendo relevantes.

Definição 8: Cardinalidade de um Termo

$cardinalidade_{termo} = |Ocorrências_{termo}|$ onde $Ocorrências_{termo} = \{item_i \in Selecionados \mid conteúdo_{item_i}$ é anotada semânticamente por $termo\}$

O valor $cardinalidade$, juntamente com um parâmetro $incremento$, são utilizados para calcular o valor do reforço aplicado aos $pesos$ dos tópicos e associações do grafo de tópicos do usuário. Conforme o Algoritmo 8, para cada $item$ $item_i$ em $Selecionados$, o sistema percorre todos os caminhos das expansões semânticas que o retornaram, porém ao contrário. Este caminho é facilmente percorrido graças às estruturas de dados $expansões$ armazenadas junto a cada $item$ (linha 17 do Algoritmo 8). Durante este percurso são reforçados os $pesos$ de cada associação $associação\_percorrida$ percorrida, de forma que $peso\_novo_{associação\_percorrida} = peso_{associação\_percorrida} + incremento$, reforçando assim as associações que contribuíram para retornar resultados marcados como relevantes.

Quando a travessia inversa de uma $expansão$ chega ao tópico que originou a expansão semântica em questão, o tópico inicial $tópico\_semente$ (no Algoritmo 8, linha 12 como tpc_origem) também é reforçado, de forma que $peso\_novo_{tópico\_semente} = peso_{tópico\_semente} + incremento$. Este processo reforça o peso dos tópicos que, quando usados como tópicos semente para o processo de expansão semântica, foram percorridos pelo sistema e foram responsáveis pela recuperação de $conteúdos$ relevantes para usuário.

Essas primeiras alterações nos $pesos$ dos tópicos e associações do grafo de tópicos servem para reforçar as partes da representação do contexto que demonstraram corresponder aos interesses do usuário. Além desta etapa de reforço, também é realizado o aprendizado de novas informações.

Caso $|Selecionados| > 1$, ou seja, caso o usuário tenha selecionado mais de um $item$ como sendo relevante, o Praesto reforça tópicos que correspondem a $termos$ em comum nas anotações semânticas dos vários resultados relevantes. Para cada tópico t percorrido nas expansões que retornaram o conteúdo dos $itens$ em $Selecionados$, o sistema obtém um conjunto $Rotulos_{termo_t}$. Este conjunto contém uma lista de palavras que, de acordo com a BC, podem ser usadas para nomear $termo_t$. Para cada $rótulo \in Rotulos_{termo_t}$ o sistema cria ou recupera, se ja existir, o tópico $t_{rótulo} = (rótulo, termo_t)$. Se o tópico $t_{rótulo}$ já existir, $peso_{t_{rótulo}}$ é incrementado com $incremento$ / $cardinalidade_{termo}$. Assim, tópicos referentes a $termos$ que anotam mais resultados relevantes recebem um refoço maior que tópicos que retornam menos resultados relevantes.

Algoritmo 8: Etapa de manutenção

```
1  início manutenção(Selecionados[])
2      Cardinalidade[], termos_usados[] = ∅;
3      para cada item em Selecionados[] faça
4          para cada termo em item.conteúdo.é_anotado_por faça
5              Cardinalidade[termo] =+ 1;
6              termos_usados[].adiciona(termo);
7          fim
8      fim
9      para cada item em Selecionados[] faça
10         mapa_palavras_chave[] = ∅;
11         para cada expansão em item.expansões() faça
12             tpc_origem = expansão.origem();
13             mapa_palavras_chave[].adiciona(tpc_origem);
14             tpc_origem.peso =+ incremento;
15             para cada termo em termos_usados[] faça
16                 rótulos[].adiciona(Ont.possíveis_rótulos_para(termo));
17                 para cada associação em expansão.faz_caminho_inverso() faça
18                     associação.peso = + incremento;
19                 fim
20                 para cada rótulo em rótulos[] faça
21                     tpc_destino = G.tópico(rótulo, termo);
22                     tpc_destino.peso =+ incremento/Cardinalidade[tpc_origem.termo ];
23                     assoc_início_fim = G.associação(expansão.origem(), tpc_destino);
24                     assoc_início_fim.peso =+ incremento/Cardinalidade[tpc_origem.termo ];
25                 fim
26             fim
27         fim
28         para cada tópico_origem em mapa_palavras_chave[] faça
29             para cada tópico_destino em mapa_palavras_chave[] faça
30                 se tpc_origem ≠ tpc_destino então
31                     trecho = G.associação(tpc_origem, tpc_destino);
32                     trecho.peso = + incremento;
33                 fim
34             fim
35         fim
36     fim
37     G.normaliza_pesos();
38     G.remove_tópicos_pouco_usados();
39 fim
```

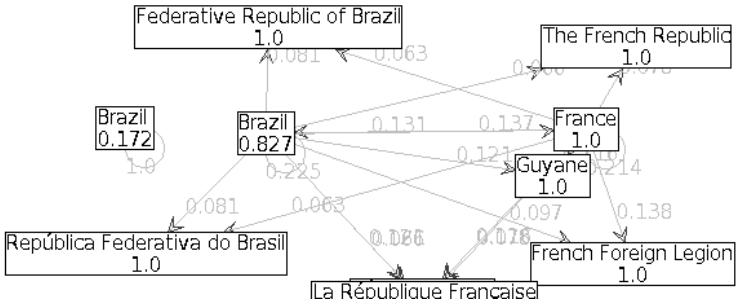

Figura 3.14: Grafo com tópicos homônimos ('Brazil').

O sistema também cria ou reforça a associação $(t, t_{rótulo})$, também com o valor $incremento\ /\ cardinalidade_{termo}$. Caso $item_i$ tenha sido recuperado devido a expansões semânticas com origem em mais de um tópico semente, o sistema também reforça (criando-as quando necessário) as associações entre estes tópicos com o valor $incremento$.

O valor do incremento é calculado desta forma para que a representatividade de cada $termo$ em relação às anotações semânticas dos resultados relevantes seja repassada ao grafo. Um $termo$ que anota o $contedo$ de todos $itens$ retornados como resultados marcados como relevantes tem $incremento = 1$, enquanto que um $termo$ que anota somente a enézima parte dos resultados marcados como relevantes tem $incremento = 1/n$.

Durante a criação dos novos tópicos e novas associações também são inseridos novos tópicos sinônimos para os $termos$ de interesse do usuário. Os $nomes$ destes novos tópicos podem vir a serem usados pelo usuário em buscas futuras, podendo ser, por exemplo, aprendidos pelo usuário em interações anteriores. Isso evita a necessidade da realização da etapa de desambiguação em interações futuras em que, dentre as palavras-chave, constem estes nomes.

Ao final dessas etapas, os $pesos$ dos tópicos e as associações os não são mais consistentes com as condições estabelecidas para os mesmos nas definições dos tópicos (soma dos $pesos$ dos tópicos homônimos ser igual a 1) e das associações (soma dos $pesos$ das associações originadas em um mesmo tópico ser igual a 1). Para tornar os $pesos$ consistentes com as regras, eles são normalizados, conforme a linha 37 do Algoritmo 8.

Como a normalização sempre retornará a soma dos $pesos$ os tópicos homônimos a 1, sempre restará ao menos um tópico correspondente à cada palavra-chave que o usuário já utilizou, mesmo que não a tenha utilizado ao longo de muitas interações com o sistema. Uma maneira de se resolver isso é contar a quantidades de interações em seqüência em cada tópico não é utilizado (seja nas expansões semânticas ou seja tendo seu peso reforçado na etapa de manutenção). Ao atingir um determinado limite imposto para este contador, estes tópicos são removidos, juntamente com as associações chegando ou partindo dos mesmos.

A Figura 3.14 mostra diversos tópicos e associações criados com base nas informações extraídas dos resultados 'French Guiana', 'France' e 'Brazil', marcados como relevantes pelo usuário. O sistema reforçou os tópicos 'France' e 'Brazil', já que estes foram utilizados para retornar os resultados marcados como relevantes. A Figura 3.14 mostra o resultado do reforço no tópico utilizado, caso na primeira parte do exemplo, referente à desambiguação, o usuário também tivesse escolhido outro uso para 'Brazil', porém nenhuma relação com qualquer dos resultado selecionados como relevantes. Ao reforçar o tópico 'Brazil' correspondente ao país e não reforçar o outro tópico homônimo, os pesos passariam a indicar que o tópico referente ao país como sendo mais representativo dos interesses do usuário.

De volta à Figura 3.12, dentre os novos tópicos, o sistema criou tópicos com $nomes$ diferentes porém se referindo aos mesmos $termos$, como 'Federative Republique of Brazil' e 'República Federativa do Brasil', ou 'La République Francaise' e 'The French Republic'. Isso provê ao gráfo conhecimento sobre sinônimos que podem vir a ser usados pelo usuário em interações futuras, conforme o usuário adquire novos conhecimentos ao utilizar o sistema.

Capítulo 4

Experimentos

Este capítulo apresenta a implementação das camadas do Praesto de acordo com conjunto de dados obtido para realização de experimentos. Também apresenta a posterior realização dos experimentos. Primeiro é apresentada a preparação do conjunto de dados utilizados. Depois é apresentada a divisão das 3 camadas (contexto, definições e conteúdo) em uma aplicação cliente-servidor, para possibilitar a distribuição do Praesto aos usuários. No final do capítulo são apresentados alguns experimentos e a avaliação dos dados coletados.

4.1 Obtenção e Preparação dos Dados

Para realizar experimentos é necessário fornecer ao sistema dados referentes às três camadas; contexto, definições e conteúdo. As informações de contexto são obtidas durante a execução do Praesto, porém as definições (conceitos e instâncias fornecidos pela ontologia e pela base de conhecimento), assim como o conteúdo armazenado, anotado de acordo com essas definições, devem ser previamente disponibilizados.

Não só as definições e o conteúdo precisam ser fornecidos, mas também as anotações semânticas deste conteúdo com as instâncias da base de conhecimentos devem estar disponíveis. Como o processo de anotação semântica não faz parte do escopo deste trabalho, surgiram duas alternativas.

- Encontrar um conjunto de documentos (sejam eles notícias, imagens, textos, etc) previamente anotados semanticamente de acordo com instâncias de uma base de conhecimento. Além disso é necessário ter acesso à base de conhecimento e também à ontologia da qual ela deriva. A disponibilidade de somente dois dos três itens inviabiliza a execução do Praesto.

- Encontrar um sistema capaz de realizar a anotação semântica do conteúdo de acordo com uma ontologia e base de conhecimento. Neste caso os três items podem ter origens diferentes, facilitando suas obtenções. Contudo, para assegurar a qualidade do processo de anotação é necessário conhecer o conjunto de documentos, para garantir a compatibilidade entre o domínio do conteúdo e da ontologia.

A opção de anotação semântica realizada por um sistema foi tentada, utilizando-se o KIM [Kiryakov et al. 2004]. O KIM possui uma ontologia própria chamda Proton e uma base de conhecimento de entidades nomeadas, que se refere a empresas, pessoas, eventos, etc. Essas entidades nomeadas são categorizados de acordo com a Proton. Dentre as funcionalidades oferecidas pelo KIM está a anotação automática de um conjunto de documentos de texto de acordo com este conjunto de entidades nomeadas.

A ferramenta foi testada com conjuntos de dados como o 20 News Group[1] e a Reuters-21578[2], porém o próprio KIM não foi capaz de realizar buscas sobre os conjuntos anotados. Para melhorar a qualidade das anotações seria necessário editar a ontologia Proton e a base de conhecimento, porém, devido a dificuldades de acesso a uma documentação clara do sistema, e pelo fato de criação ou edição de ontologias não fazer parte do trabalho, optou-se por pela primeira alternativa, utilizando um grande conjunto de documentos já previamente anotados semanticamente, disponibilizado juntamente da ontologia e base de conhecimento.

[1] `http://people.csail.mit.edu/jrennie/20Newsgroups/`, acessado em 23 de outubro de 2009

[2] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`, acessado em 23 de outubro de 2009

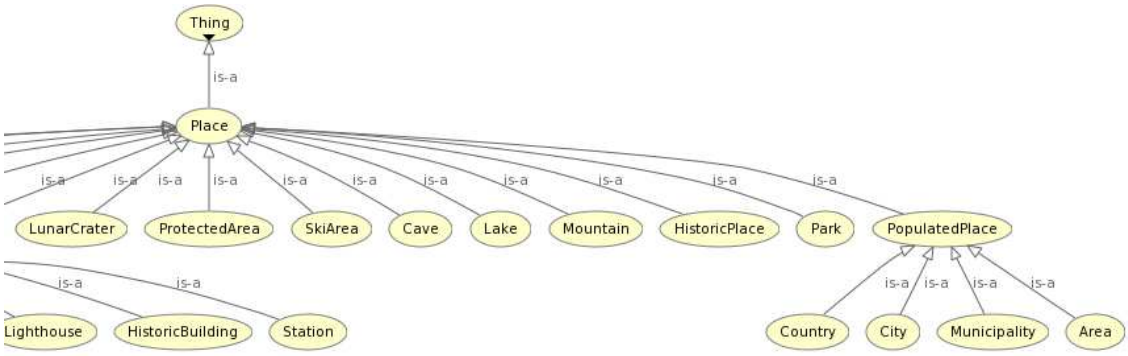

Figura 4.1: Exemplo de parte da DBPedia

4.1.1 A Ontologia e as Instâncias

A solução escolhida para suprir as camadas de conteúdo e de definição foi usar conjuntos de dados a DBpedia [dbp] e da Wikipedia. DBpedia é uma iniciativa para extração de informação estruturada da Wikipedia e para a sua publicação na *Web*. A sua base de conhecimento conta com 882 mil instâncias, sendo no mínimo 214 mil pessoas, 248 mil lugares, 193 mil trabalhos, 90 mil espécies, 76 mil organizações, 23 mil prédios, além de outras instâncias associadas a outros conceitos. Estas instâncias estão organizadas em um total de 274 milhões de triplas *RDF*.

Essas instâncias são classificadas de acordo com a *DBpedia Ontology*, uma ontologia criada com base nas informações presentes nas *infoboxes* da Wikipedia. Uma *infobox* é uma caixa na lateral das páginas da Wikipedia que contém informações apresentadas de maneira padronizada em função do tipo de conteúdo da páginas. A Figura 4.2 mostra a *infobox* associada ao verbete referente à própria Wikipedia[3]. A Figura 4.1 ilustra graficamente a relação entre algumas classes da DBPedia. Para a versão utilizada, DBPedia 3.2, somente um subconjunto de todos tipos de *infobox* foi mapeado para a ontologia, que juntamente com a base de conhecimento, é disponibilizada na forma de três arquivos:

- Um arquivo *OWL* que contém os possíveis tipos das instâncias, organizados em classes. São exemplos de classes lugar (*place*), pessoa (*person*) ou esporte (*sport*), dentre um total de 170 classes.

[3] `http://en.wikipedia.org/wiki/Wikipedia`

URL	www.wikipedia.org
Slogan	The free encyclopedia that anyone can edit.
Commercial?	No
Type of site	Online encyclopedia
Registration	Optional
Available language(s)	236 active editions (267 in total)[1]
Owner	Wikimedia Foundation
Created by	Jimmy Wales, Larry Sanger[2]
Launched	January 15, 2001 (8 years ago)
Alexa rank	#7[3]
Current status	work-in-progress[4]

Figura 4.2: Exemplo de *infobox*: Verbete da própria Wikipedia

- Um arquivo de triplas que faz o mapeamento de tipo entre as instâncias da base de conhecimento e as classes da ontologia, através de triplas no formato:

  `<instância> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <classe>.`

- Um arquivo de triplas que define propriedades das instâncias e seus respectivos valores. Dentre estas propriedades estão `http://xmlns.com/foaf/0.1/name` e `http://dbpedia.org/ontology/nativeName`, que definem nomes para as instâncias. Essas são as propriedades utilizadas para buscar por correspondências com as palavras-chave das buscas.

4.1.2 Os Documentos e as Anotações Semânticas

As instâncias da base de conhecimento da DBPedia correspondem, cada uma, a uma e somente uma página da Wikipedia. Como cada página da Wikipedia está associada a uma instância, não há uma descrição semântica da página com qualidade suficiente para o sistema de busca. Contudo, cada página da Wikipedia possui *links*, armazenados em triplas *RDF* `<origem> <http://dbpedia.org/ property/wikilink>`

`<destino>`, formando *hyper*-textos.

Devido a essa organização do conteúdo cada documento na Wikipedia é considerado como um *hyper*-texto formado por uma página central e um conjunto de outras páginas acessíveis por *links*. As anotações semânticas são extraídas dos *links*. Como nas páginas da Wikipedia podem existir *links* para outras páginas também da Wikipedia(como *links* para outras páginas com mesmo nome, porém com conteúdo diferente, com propósito de desambiguação), só são considerados, para fins de anotação semântica, os pares de páginas em que há *links* nos dois sentidos.

4.2 Implementação

A implementação do Praesto, cuja arquitetura proposta é organizada em um sistema composto de 3 camadas (contexto, definições e conteúdo), foi dividida em 2 componentes, Cliente e Servidor, como ilustrado na Figura 4.3. O Cliente corresponde às partes do sistema responsáveis pela adaptação do sistema ao contexto de cada usuário, enquanto o resto do sistema é implementado no Servidor, conforme descrito na seqüência:

4.2.1 Servidor

O Servidor foi construído através da implementação de serviços capazes de realizarem o acesso aos bancos de dados utilizados para armazenar a ontologia, a base de conhecimento e as descrições dos documentos. Estes serviços implementam interfaces de aplicações definidas de forma a suprir os dados consumidos durante o processo de busca e os dados são persistidos em bases de dados relacionais.

Os dados utilizados são fornecidos em formato de arquivos de tripla constituídas por campos *subject*, *property*, *object*, as quais, através do uso de bibliotecas de terceiros, são persistidas nas bases de dados. Estes dados são gerenciados por dois componentes do Servidor:

- `Definições:` Este componente é responsável pelo armazenamento da ontologia e da base de conhecimento. É acessado pelo Cliente para obter informações

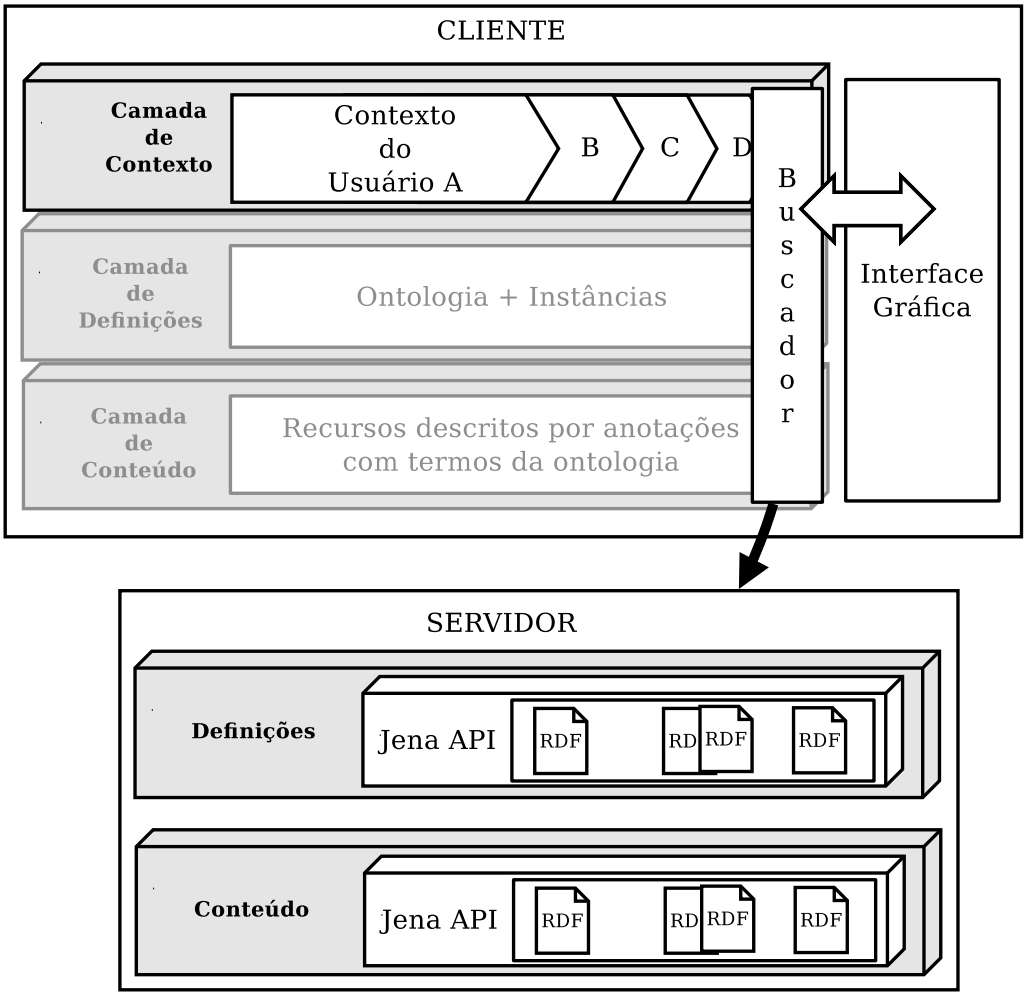

Figura 4.3: Diagrama do sistema dividido em Cliente e Servidor.

referentes a definições, como:

Quais são as instâncias cujo nome correspondem a uma palavra-chave?

Quais são os nomes utilizados para nomear uma instância específica?

A qual classe pertence uma determinada instância?

- `Repositório:` É o componente responsável pelo armazenamento das descrições dos documentos (como título, anotações e *link* para o documento em si, neste caso, páginas da *Web*). É acessado pelo Cliente para obter informações referentes ao conteúdo armazenado, como:

  Quais são os documentos anotados semanticamente com uma instância específica?

  Quais são as anotações semânticas usadas para descrever um determinado documento?

  Qual é o título de um documento armazenado?

  Quais são os *links* para as páginas da Wikipedia que compõem um documento?

Tanto `Repositório` quanto `Definições` utilizam a *API* fornecida pela biblioteca Jena[4], implementada em Java. Por praticidade, todo o restante do Servidor também é implementado em Java. Outras opções foram consideradas, como Soprano[5] e Redland[6], porém Jena ofereceu maior facilidade de acesso à documentação completa e de qualidade.

Todas as triplas *RDF* são armazenadas em bases de dados relacionais, que são gerenciadas automaticamente pelo próprio Jena. Isto não é simplesmente uma facilidade oferecida pela biblioteca, mas uma condição para garantir o funcionamento esperado das aplicações desenvolvidas.

[4] `http://jena.sourceforge.net`
[5] `http://soprano.sourceforge.net`
[6] `http://librdf.org`

4.2.2 Cliente

Os componentes do lado cliente do Praesto implementam os algoritmos apresentados neste trabalho e também provêm a visualização do grafo do usuário e também dos resultados das buscas. O Cliente é dividido em três componentes principais, que assim como no Servidor, são implementados em Java:

- `Contexto:` É o componente responsável por armazenar e organizar, na forma de um grafo de tópicos, as informações sobre o contexto semântico de cada usuário.

- `Buscador:` É o componente que faz acesso ao Servidor e realiza os processos de desambiguação, busca e extração de informações de contexto dos resultados selecionados. O acesso ao Servidor é realizado por meio de chamadas remotas (*RMI*). Para a realização dos experimentos, os parâmetros usados para o processo de busca estão configurados com os seguintes valores:

  `Peso mínimo do tópico semente:` Zero (0). O peso mínimo para o tópico semente não está sendo utilizado.

  `Peso mínimo da associação:` Está sendo utilizado um valor dinâmico. Esse valor é calculado como sendo a média dos pesos das associações partindo do tópico sendo expandido, menos três desvios padrões. Essa solução foi escolhida pois o peso das associações é inversamente proporcional a sua quantidade. Como as páginas da Wikipedia não têm uma quantidade uniforme de *links* (que são usados como anotações, que por sua vez influem na criação das associações), essa solução se ajusta ao estado atual do grafo. O valor de três desvios padrões foi escolhido empiricamente, com base em testes realizados com intensão de escolher um valor que não resultasse na exclusão de resultados potencialmente relevantes.

  `Profundidade máxima da busca:` Três (3). Este valor foi escolhido porque, no processo de manutenção do grafo, tópicos relevantes ao tópico correspondentes à palavra-chave passam a ser associados ao tópico da palavra-chave. A relevância do caminho percorrido diretamente entre os tópicos tende a ser menor que a dos caminhos percorrendo vários tópicos intermediários, assim sendo, uma

profundidade maior só aumenta o tempo de processamento, sem causar alterações significativas nos resultados.

`Relevância mínima de um tópico:` Zero (0). As expansões prosseguem até atingirem a profundidade máxima.

- `Interface Gráfica:` A interface gráfica permite a visualização dos resultados retornados pelas buscas, a visualização do gráfico de tópicos do usuário e fornece o acesso ao conteúdo dos documentos. A interface também é responsável por fornecer uma interação intuitiva entre o usuário e o sistema, porém esta questão foge do escopo deste trabalho. Mais informações sobre a interface podem ser encontradas em [Fasolin et al. 2009]

4.3 Considerações sobre os Experimentos

Antes de apresentar os experimentos propostos, é necessário reforçar a dificuldade em se obter dados para a realização dos experimentos. Mais precisamente, a dificuldade em se obter um conjunto de documentos, uma ontologia e base de conhecimento associada e as anotações semânticas destes mesmos documentos de acordo com esta base de conhecimento. Também é importante destacar que a possível alternativa da utilização de uma ferramenta de anotação semântica [Kiryakov et al. 2004] para gerar as anotações de documentos de acordo com uma base de conhecimento, não gerou resultados aproveitáveis, devido às dificuldades de uso e incompatibilidade da ontologia e base de conhecimento com o conjunto de dados.

Também existe uma incompatibilidade entre os objetivos deste trabalho e avaliações dos resultados com a utilização de *benchmarks*. Essa incompatibilidade tem origem em dois fatores. O primeiro tem a ver com a motivação do Praesto, de trabalhar com aspectos semânticos relativos ao contexto semântico individual de cada usuário. Mais precisamente, o Praesto procura inferir a intensão do usuário referente a uma palavra-chave com base nas informações de contexto semântico disponíveis. Para saber se o Praesto é capaz de cumprir esta tarefa, é necessária a realização de testes com pessoas ou a simulação

destas, o que depende da simulação do seu processo cognitivo.

Uma teoria sobre como acontecem os processos cognitivos humanos é o Computacionalismo [Ibáñez and Cosmelli 2008] [Dietrich 2000]. De acordo com essa teoria, a cognição é a execução de funções Turing-computáveis, definidas sobre diversas representações de entidades [Dietrich 2000]. Isso significa que poderia se analisar a mente sobre um modelo baseado em uma metáfora de um computador; a mente é um computador que, baseado em regras, processa símbolos, sendo conseqüentemente lógica, racional e isolada (não sofre influência de outros elementos) [Ibáñez and Cosmelli 2008]. Porém, além dessa teoria, há quem considere também outras idéias, como intensionalidade, intersubjetividade (processos cognitivos sobra uma informação dependem de aspectos subjetivos sobre informações) e ecologia da mente, esta última que diz que a mente é um processo que acontece em um contexto, em situações específicas associadas ao ambiente, [Ibáñez and Cosmelli 2008], complementando o Computacionalismo.

Então, de acordo com essas idéias, não se deve ver a mente humana como somente processos de um sistema isolado. Apesar de serem apenas teorias, não existem modelos comprovados de como exatamente funciona a mente humana para invalidá-las, havendo ainda a necessidade da realização de experimentos empíricos e construção de modelos formais com relação às ciências cognitivas [Ibáñez and Cosmelli 2008]. Por isso, não há como garantir que sistemas ou *benchmarks* que deveriam representar o processo cognitivo de um usuário realmente cumpram suas tarefas.

O segundo fator tem relação com o uso da heurística *ACO* utilizada, adapativa e baseada em *feedback*. Os sistemas adaptativos desenvolvidos para interagirem com usuários desenvolvem modelos dos estados mentais dos usuários de forma *botton-up*, ou seja, partem do usuário, sem garantias de que será baseado em características compartilhadas por outros usuários, resultando em sistemas que não são capazes de pensar, mas têm uma idéia sobre o que cada usuário está pensando [Johnson 2001]. Como o sistema se desenvolve de maneira *botton-up*, baseado em características subjetivas, individuais de cada usuário, não é possível prever seu comportamento, o que significa que não é possível avaliar o desempenho do sistema com base em *benchmarks*, por serem construídos com base em um funcionamento esperado, baseado em informações previamente conhecidas.

Ao se testar o funcionamento do sistema com um *benchmark*, este fará somente o mesmo papel de um usuário qualquer, no caso, dos usuários idealizadores do *benchmark*. No caso do *benchmark* ter sido criado com base nas experiências de diversos usuários, o que é mais provável, estará se atribuindo ao sistema uma tarefa para a qual não foi projetado, que é se adaptar a padrões originários de aspectos comuns a diversos usuários.

4.3.1 Solução para os Experimentos

Não há como garantir que o conjunto de consultas de um *benchmark* interessem a um usuário, ou que os resultados que interessam ao usuário correspondam ao conjunto de respostas previsto pelo *benchmark*, então foram usadas outras alternativas.

Pode-se consultar o conjunto de triplas *RDF* com consultas *SPARQL* que buscam por todos os documentos que são anotados semanticamente por instâncias cujos nomes correspondem às palavras-chave. Os resultados serão sempre os mesmos, contanto que o conjunto de dados continue o mesmo. Esses resultados incluem todas as possíveis denotações para as palavras-chave e nada além disto. Ao invés de comparar todos os dados obtidos pelos experimentos com os usuários com as respostas desse tipo de consulta, é possível comparar essas respostas com os resultados referentes à cada usuário.

Em cada uma dessas comparações pode-se comparar se, dentre os resultados selecionados como relevantes pelo usuário, há resultados que não estão listados entre os retornados pelas consultas *SPARQL*. A presença deste tipo de resultado indica que o Praesto foi capaz de aprender informações sobre o contexto do usuário que levaram a resultados que não são retornados apenas pelas palavras-chave, através de buscas léxicas, contribuindo para o aumento da cobertura. Ainda assim, não há como, com essas comparações, medir se o Praesto deixou de retornar resultados que, se disponíveis, o usuário consideraria como relevantes.

A avaliação da precisão é difícil de ser realizada. A expansão semântica no grafo de tópicos expande as buscas de forma a retornar mais resultados, porém o sistema não elimina resultados considerados como sendo pouco relevantes, somente os apresenta por último, segundo as relevâncias calculadas. Por isso a precisão não pode ser avaliada

através da relação entra quantidade de resultados relevantes e a quantidade de resultados selecionados, mas sim verificando se os resultados inferidos como sendo relevantes pelo sistema também são considerados como sendo relevantes pelo usuário.

Então, em vez de fornecer valores precisos de medidas de precisão e cobertura das buscas, esses experimentos procuram mostrar que, de maneira pouco invasiva ao usuário, o sistema é capaz de armazenar a desambiguação de palavras-chaves e a utilização de figuras de linguagem semânticas por parte do usuário. Avaliações mais precisas dependem de séries mais extensas de experimentos com um conjunto maior de usuários, condição necessária para se compreender com precisão o funcionamento do sistema.

4.4 O Experimento

O experimento realizado consiste na execução de buscas por palavras-chave que são repetidas após o usuário selecionar os resultados que ele considera relevantes. O objetivo deste experimento é procurar, nos dados coletados durante a interação dos usuário com o sistema, por indicações de que o processo de busca proposto e implementado no Praesto é capaz de prover algum benefício ao usuário que o utiliza, mais precisamente, que o sistema seja capaz de retornar resultados que atendam aos interesses de cada usuário.

Previamente à realização dos experimentos, todos os usuários receberam pessoalmente instruções de como deveriam proceder para a realização dos experimentos. Essas instruções incluíram também explicações sobre o funcionamento da interface gráfica da aplicação, para garantir que todos os usuários realizando os experimentos conhecessem e soubessem acessar todas funcionalidades necessárias.

O experimento consiste de três passos, dois dos quais realizados repetidamente, conforme enumerados a seguir:

1. Escrever palavra(s)-chave e buscar denotações disponíveis, repetindo o processo até criar no mínimo cinco tópicos de interesse do usuário.

2. Especificar buscas selecionando como palavras-chave os nomes dos tópicos disponíveis no grafo de tópicos. Após avaliar os resultados, selecionar os resultados

que corresponderem às intensões do usuário com a busca, ou seja, os resultados considerados relevantes.

Repetir a busca duas vezes e, ao final de cada vez, também selecionar os resultados considerados relevantes.

3. Repetir o item 2 até ter realizado três buscas diferentes, cada uma com duas repetições.

Durante a execução do experimento, uma janela mostra ao usuário as instruções que devem ser seguidas. O cliente também interrompe as repetições assim que o número necessário é atingido e orienta o usuário sobre qual é o próximo passo a ser tomado.

Um problema em potencial para a realização dos experimentos é o conjunto de dados utilizados. Algumas instâncias da base de conhecimento da DBPedia não possuem todas triplas *RDF* referentes a todas as propriedades utilizadas para realização das buscas. Um exemplo é a instância referente ao país 'China', que não possui a palavra China como valor das propriedades `name` ou `nativeName`, utilizadas para comparação com as palavras-chave. Conseqüentemente, um usuário buscando por informações sobre esse país através de uma consulta por 'China' não encontrará o conteúdo desejado, ainda que encontre outras instâncias homônimas.

A fim de contornar este problema, o processo de busca só tem início após o usuário criar 5 tópicos no seu grafo de tópicos. Esses tópicos são criados fazendo o sistema mostrar as denotações disponíveis para as palavras-chave, na camada de definições, assim como acontece no processo de desambiguação.

Todos os dados coletados sobre os experimentos dos usuários são armazenados junto ao servidor do Praesto. Os dados coletados permitem visualizar e, se necessário, refazer todas as interações dos usuários com o sistema.

4.5 Análise dos Resultados

Os experimentos realizados tiveram a participação de 8 usuários, totalizando 24 buscas. Cada busca foi repetida 2 vezes, totalizando 72 interações ($usuários * (buscas *$

$(1 + repeticões)))$) dos usuários com o Praesto. Uma interação é considerada como o conjunto de atividades esperado do usuário ao utilizar o sistema, neste caso, realizar uma busca, observar os resultados e selecionar os resultados relevantes. Cada interação do usuário com o sistema resulta em uma iteração do processo de busca do Praesto.

Do total de 72 interações, o experimento de três usuários (usuários 2, 4, 5) não foram feitos de acordo com as orientações e forem descartados, restando 45 interações. Das 45, o registro da última interação de um usuário (usuário 6) foi corrompido, porém os dados deste usuário foram mantidos, sendo utilizados somente em algumas das análises. Todos dados mantidos estão disponíveis no Apêndice A.

Os usuários tiveram acesso somente às buscas realizadas pelo Praesto, apresentados ordenados de acordo com o *ranking* calculado pelo sistema. Os resultados retornados de pesquisas realizadas com SPARQL (*SPARQL Protocol and RDF Query Language*) foram obtidos durante análise dos dados, para comparação entre a quantidade de dados retornados pelo Praesto em relação a buscas léxico sintáticas (via SPARQL).

O SPARQL foi usado como uma referência por ser uma ferramenta capaz de buscar entre os dados na forma de triplas RDF, retornando resultados recuperados por buscas léxicas e sintaticas. Os resultados retornados para uma palavra chave em particular se referem a todas as possiveis denotações para a palavra descritas no sistema, já que o sistema não busca nada além da palavra chave em questão. Assim, é possível comparar estes resultados aos resultados retornados pelo Praesto; uma desambiguação eficiente deve diminuir o número de resultados retornados, enquanto uma expansão semântica bem feita deve, ao menos em alguns casos, retornar resultados relevantes ao usuário, além dos retornados pelo SPARQL.

Foram analisadas as seguintes medidas:

- Quantidade de Resultados Selecionados `(Sel.)`: Mede a quantidade de resultados retornados pelo Praesto que foram selecionados pelo usuário como relevantes.

- Quantidade de Resultados Retornados via Praesto `(Ret. Praesto)`: Mede a quantidade de resultados retornados pelo Praesto, independentemente da avaliação do usuário sobre os mesmos.

- Quantidade de Resultados Retornados via SPARQL `(Ret. SPARQL):` Mede a quantidade de resultados retornados por buscas léxicas realizadas com SPARQL para o conjunto de palavras-chave usado pelo usuário.

- Relevância Mínima dos Resultados Selecionados `(R. Mínima Sel.):` Mede a menor relevância (calculada pelo Praesto) dentre todos os resultados selecionados pelo usuário.

- Relevância Média dos Resultados Selecionados `(R. Média Sel.):` Mede a relevância média dentre todos os resultados selecionados pelo usuário.

- Relevância Média dos Resultados Retornados `(R. Média Ret.):` Mede a relevância média dentre todos os resultados retornados pelo Praesto.

- Relação entre Relevância Média Retornada e Relevância Média Selecionada `(R. Média Sel. / R. Média Ret.:` Mede a relação ente a media de relevância dos resultados selecionados e a média de relevância dos resultados retornados.

Essas medidas foram tomadas organizadas em seqüências de 3 interações para cada busca (1 interação inicial da busca e 2 repetições posteriores). Também foram tomadas em seqüências de 8 interações. Foram usadas 8 interações para não excluir os registros do usuário 6, que não possuem a nona e última interação. O objetivo esperado da análise destas medidas é mensurar o quanto o sistema acumula informação sobre o contexto ontológico dos usuários, representadas pelo grafo de tópicos, ao longo das interações, e quanto essas informações contribuem para que o sistema corresponda com os interesses do usuário.

4.5.1 Seqüências de 3 Interações para as mesmas Palavras-Chave

São considerados os dados de todos os usuários que realizaram todas as buscas com todas as repetições corretamente (todos usuários menos usuário 6). Considera-se cada busca e repetições associadas como não sendo relacionadas às outras. Os valores avaliados são medidos calculando-se a média de cada valor para cada busca (interação 1),

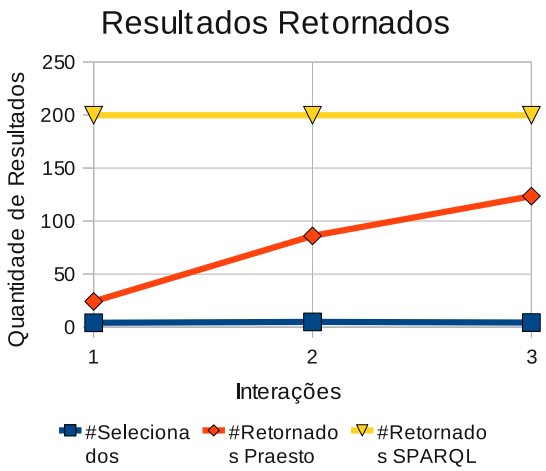

Figura 4.4: Medidas referentes às médias do número de resultados retornados pelo Praesto, SPARQL e selecionados nas seqüências de 3 interações

para a primeira repetição (interação 2) e para a segunda repetição (interação 3). Os dados referentes a esta análise estão na Tabela 4.1.

Tabela 4.1: Relevâncias ao longo de três iterações do usuário com sistema

Interação	Sel.	Ret.	Ret. SPARQL.	R. Min. Sel.	R. Média Sel.	R. Média Ret.	R. Média Sel. / R. Média Ret.
1	4,08	24,25	199,83	0,95	0,97	0,96	1,02
2	4,92	86	199,83	0,53	0,77	0,58	1,34
3	4,33	123,5	199,83	0,45	0,61	0,53	1,17

O gráfico da Figura 4.4 mostra as médias das quantidades de resultados retornados pelo Praesto para a busca do usuário, assim como a quantidade retornada por buscas léxicas com o SPARQL e, dentre os resultados trazidos pelo Praesto, quantos foram selecionados pelo usuário.

O gráfico mostra uma grande redução na quantidade de resultados retornados pelo Praesto, em relação à quantidade de resultados retornados por consultas SPARQL. Esta redução se deve ao processo de desambiguação realizado pelo Praesto, no momento que

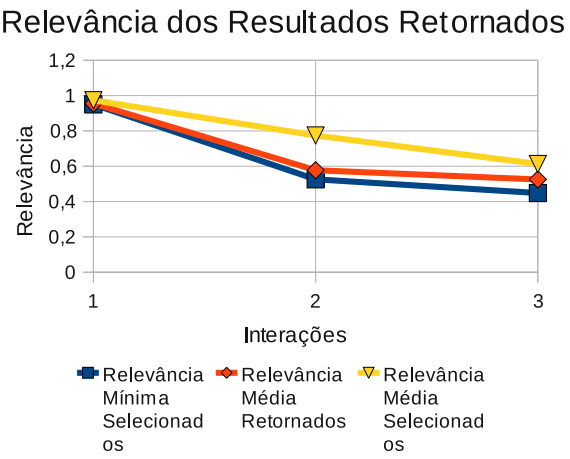

Figura 4.5: Medidas referentes às médias da relevância calculada para os resultados retornados pelo Praesto, SPARQL e selecionados nas seqüências de 3 interações

o usuário busca por uma palavra chave não existente no gráfico de tópicos. O aumento na quantidade de resultados retornados pelo Praesto ao longo das iterações seguintes se deve à adição de novos tópicos no gráfo de tópicos, provenientes da etapa de manutenção do grafo com base no *feedback* fornecido pelo usuário.

O gráfico da Figura 4.5 mostra a média dos valores das relevâncias dos resultados, para os resultados selecionados pelo usuário e para os retornados pelo Praesto. Em uma primeira análise, parece que o Praesto retorna resultados menos precisos a cada nova busca, ou seja, a qualidade dos resultados diminuiu com o passar do tempo.

Ainda assim, é possível ver que a relevância média dos resultados retornados pelo Praesto e selecionados pelo usuário como relevantes é maior que a relevância média dos resultados retornados. Isso significa que os usuários tendem a considerar como sendo mais relevantes (conseqüentemente marcando para *feeback*) os resultados que o Praesto inferiu como sendo mais relevantes (com relevância acima da média).

É possível verificar que a queda da relevância acontece ao mesmo tempo em que há um aumento no número de resultados retornados (Figura 4.4). Como com o passar das buscas o sistema cria novos tópicos no grafo do usuário e os resultados são retornados em função dos tópicos percorridos na expansão, quanto mais tópicos, é provável que

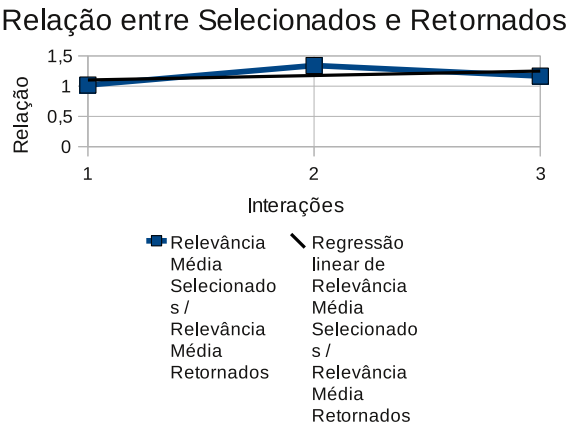

Figura 4.6: Relação entre a relevância média dos resultados selecionados e a dos resultados recuperados nas seqüências de 3 interações

o sistema retorne mais resultados. Como nem todos resultados devem ser relevantes, muitos recebem uma relevância reduzida, o que acaba refletindo na redução da relevância no decorrer do experimento, porém, os resultados relevantes para o usuário, continuam sendo inferidos como relevantes pelo Praesto.

Isso pode ser verificado na Figura 4.6. O gráfico da Figura 4.6 mostra a relação entra o valor médio da relevância dos resultados selecionados e dos resultados retornados. Apesar da queda no valor médio da relevância dos resultados, a relevância média dos resultados selecionados pelo usuário tende a aumentar, se analisada em relação ao valor médio. Isto indica que, apesar do Praesto retornar uma quantidade crescente de resultados (mesmo assim menor do que uma busca SPARQL), o sistema não apresenta todos como sendo igualmente relevantes.

Essa observação é importante, pois mostra que avaliar somente a quantidade de resultados retornados, sem observar outros dados, não garante uma análise precisa da qualidade do sistema. Um possível problema deste tipo de avaliação neste experimento é que a quantidade de documentos disponíveis no repositório relativos a cada assunto não é uniforme, e por isso, a avaliação do sistema baseado na quantidade de resultados é imprecisa, já que poucos resultados podem ser retornados por causa da baixa quantidade de documentos, não da imprecisão do sistema.

4.5.2 Seqüências de 8 Interações para as mesmas Palavras-Chave

São consideradas somente as oito primeiras interações, para poder incluir as interações do usuário que teve o registro da última interação corrompido. Desta forma, as seis primeiras interações se referem às duas primeiras buscas e suas duas repetições e as duas últimas interações correspondem à última buscas e somente uma repetição.

As análises são feitas com buscas com provável variação do contexto ao longo das interações e sem provável variação. Esta divisão foi feita por que, em alguns dos experimentos realizados, não é possível ter certeza de que os usuários mantiveram o foco das buscas em um mesmo assunto, porém, o objetivo do Praesto é justamente aprender através de repetições de buscas relacionadas a um mesmo contexto.

Com Possível Variação do Contexto

Foram utilizados somente os dados de usuários (usuários 1, 3, 6, 7 e 8) que realizaram todo o experimento de acordo com as orientações, porém são analisadas somente oito interações para não excluir o usuário 6, cujo registro da última interação foi corrompido. Dentre as buscas realizadas pelos usuários, não é possível saber quais usuários mantiveram a mesma intenção de busca (conseqüentemente o contexto ao qual as palavras-chave se referiam) ao longo de todas as 3 buscas diferentes. Essa informação é importante pois pode indicar que o Praesto precisa aprender sobre 3 diferentes contextos do usuário durante o curto período dos experimentos e não somente 1 contexto. Os dados referentes a esta análise estão na Tabela 4.2.

Sem Variação do Contexto

Foi utilizado somente os dados de usuários (usuários 1 e 6) que repetiram as palavras-chave de buscas anteriores e selecionaram resultados também retornados em buscas posteriores, indicando que, provavelmente, todas as buscas se referiam a um mesmo contexto (i.e.: o usuário 6 buscou sobre grupos musicais durante todas as 3 buscas). A ausência destes indícios não significa que não houve variação do contexto, mas também não indica o contrário.

Tabela 4.2: Relevâncias ao longo de oito iterações do usuário com sistema

Interação	Sel.	Ret.	Ret. SPARQL.	R. Min. Sel.	R. Média Sel.	R. Média Ret.	R. Média Sel. / R. Média Ret.
1	3,6	27,2	354,2	1	1	1	1
2	4,2	92,8	354,2	0,56	0,85	0,6	1,41
3	4	141,4	354,2	0,36	0,62	0,59	1,05
4	5,6	27,6	364	0,9	0,94	0,91	1,04
5	6,8	97	364	0,45	0,66	0,47	1,41
6	5,4	146,4	364	0,44	0,52	0,42	1,25
7	4	65,4	357,6	0,53	0,66	0,52	1,26
8	5,6	102,4	357,6	0,43	0,6	0,41	1,44

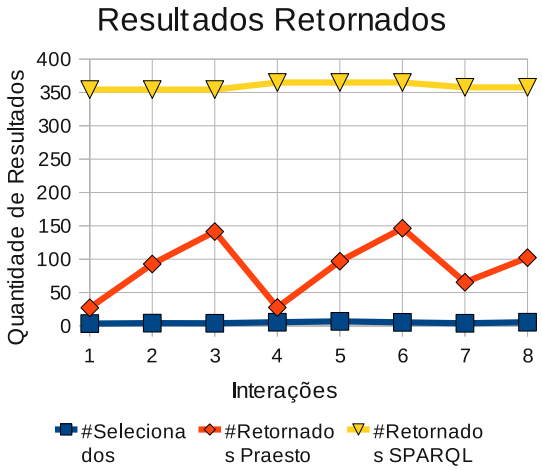

Figura 4.7: Medidas referentes às médias do número de resultados retornados pelo Praesto, SPARQL e selecionados nas seqüências de 8 interações

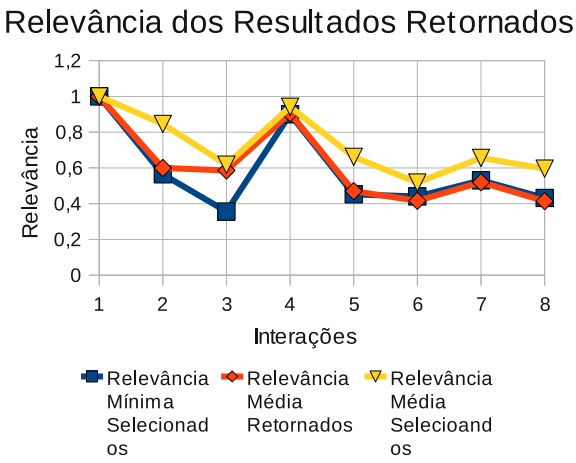

Figura 4.8: Medidas referentes às médias da relevância calculada para os resultados retornados pelo Praesto, SPARQL e selecionados nas seqüências de 8 interações

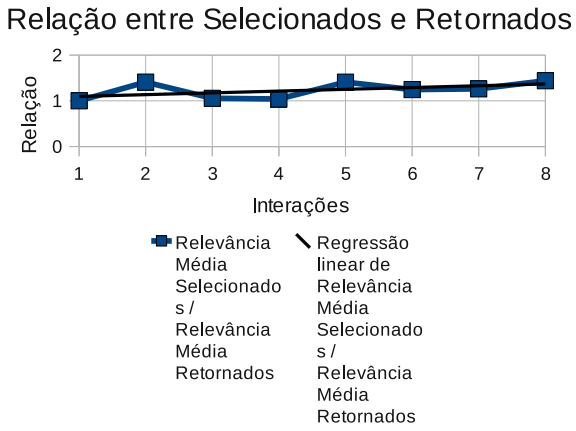

Figura 4.9: Relação entre a relevância média dos resultados selecionados e a dos resultados recuperados nas seqüências de 3 interações

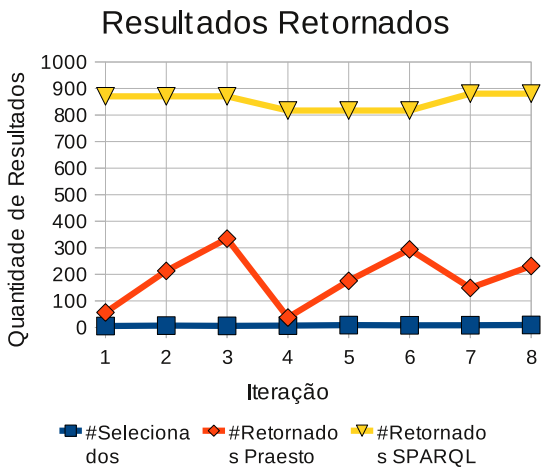

Figura 4.10: Medidas referentes às médias do número de resultados retornados pelo Praesto, SPARQL e selecionados nas seqüências de 8 interações sem variação do contexto

Nestes casos, tópicos de buscas anteriores tem maior probabilidade de contribuírem em buscas posteriores. Os dados referentes a essa análise estão na Tabela 4.3.

Tabela 4.3: Relevâncias ao longo de oito iterações do usuário com sistema, com buscas relacionadas a um mesmo tipo de informação

Interação	Sel.	Ret.	Ret. SPARQL.	R. Min. Sel.	R. Média Sel.	R. Média Ret.	R. Média Sel. / R. Média Ret.
1	5,5	56,5	870,5	1	1	1	1
2	7	213	870,5	0,38	0,75	0,35	2,17
3	6	334,55	870,5	0,35	0,77	0,31	2,52
4	7	37,5	817	0,75	0,86	0,77	1,12
5	9	175,5	817	0,25	0,67	0,15	4,43
6	8	294	817	0,58	0,75	0,1	7,89
7	8,5	149	880,5	0,33	0,64	0,3	2,13
8	9,5	231,5	880,5	0,19	0,57	0,14	4,19

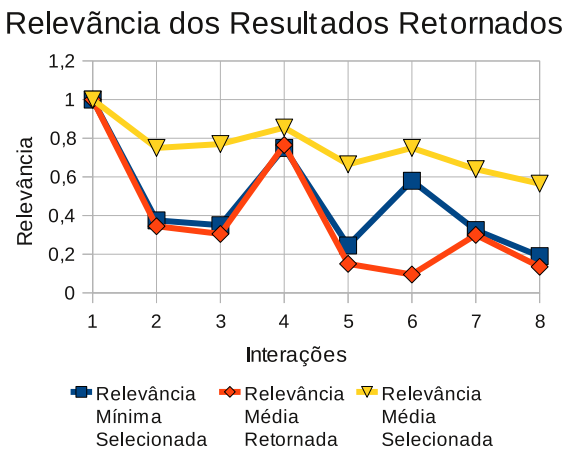

Figura 4.11: Medidas referentes às médias da relevância calculada para os resultados retornados pelo Praesto, SPARQL e selecionados nas seqüências de 8 interações sem variação do contexto

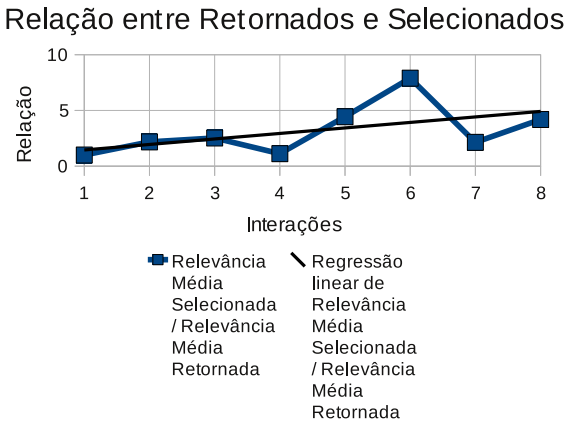

Figura 4.12: Relação entre a relevância média dos resultados selecionados e a dos resultados recuperados nas seqüências de 3 interações sem variação do contexto

Análise dos resultados

Como é possível ver nos Gráficos 4.4, 4.10 e 4.7, apesar de haver um crescimento na quantidade de resultados retornados pelo Praesto, e até mesmo, em alguns momentos, essa quantidade ser maior que o número retornado pelas buscas com SPARQL, esta quantidade tende a ser menor que a retornada pelo SPARQL.

Como o Praesto só retorna um resultado que, de acordo com a inferência do sistema, tem alguma relevância ao usuário, mesmo que mínima, e há um aumento na quantidade de resultados retornados, há uma tendência de que o valor médio da relevância calculada para os resultados diminua. Contudo, a relevância média dos resultados selecionados pelo usuário tem uma redução mais lenta, sendo que, os resultados menos relevantes selecionados pelo usuário tendem a ser mais relevantes que a relevância média, conforme indicado pelos Gráficos 4.5, 4.11 e 4.8.

A relevância alta indica que o Praesto inferiu um resultado como relevante. Um resultado selecionado indica que o usuário considera o resultado como relevante. O fato dos resultados com relevância alta serem considerados relevantes pelo usuário indica que parte dos resultados inferidos como relevantes realmente são relevantes. O sistema poderia então retornar todos resultados com uma alta relevância, porém isto refletiria em um aumento da relevância média dos resultados retornados, a ponto de alcançar a relevância média dos resultados selecionados. Entretando, os Gráficos 4.6, 4.12 e 4.9 mostram que a distância entre esses valores tende a aumentar e não reduzir com o passar das interações.

Isso pode ser verificado nas Figuras 4.13, 4.14 e 4.15, que mostram parte da interface gráfica com os resultados para 3 interações de uma mesma busca do usuário 1. Apesar do aumento na quantidade de resultados retornados, de 4 páginas, para 16 e finalmente para 26, sendo que cada página comporta até 25 resultados, os resultados relevantes continuam a subir posições no *ranking* dos resultados.

Também é interessante mostrar um comportamento visível nos dados que atesta a incompatibilidade dos experimentos com a utilização de *benchmarks*. Nas Figuras 4.16, 4.17 e 4.18, referentes ao usuário 8, é possível ver a mudança no interesse do usuário ao longo das interações. Essa situação, onde o usuário deixa de marcar um resultado já

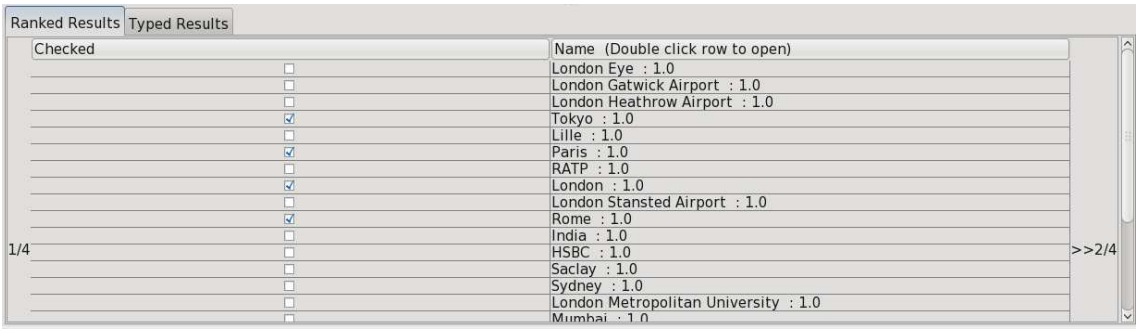

Figura 4.13: Reorganização dos resultados no ranking: primeira interação

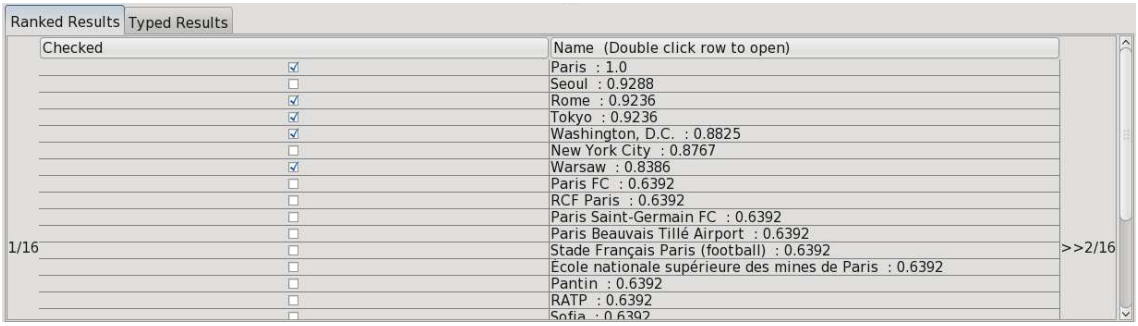

Figura 4.14: Reorganização dos resultados no ranking: segunda interação

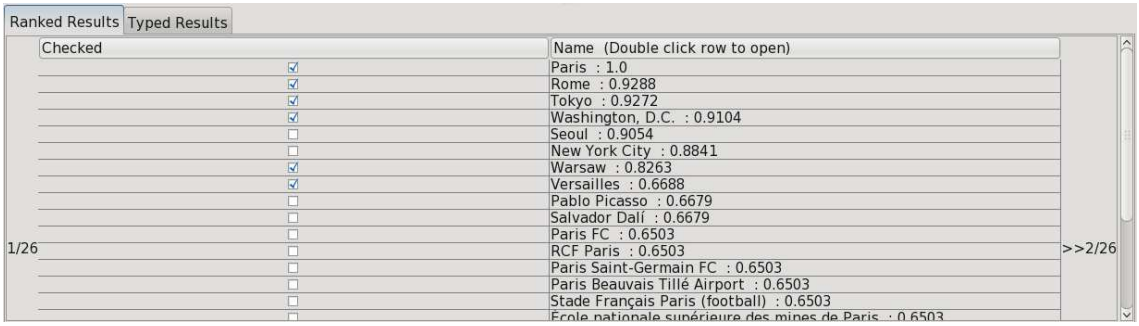

Figura 4.15: Reorganização dos resultados no ranking: terceira interação

Figura 4.16: Primeira interação com 2 resultados selecionados

Figura 4.17: Segunda interação com 1 resultado selecionado

Figura 4.18: Terceira interação com nenhum resultados selecionado

marcado em interações anteriores aconteceu também com outros usuários e atesta como que o comportamento do usuário é imprevisível. Neste exemplo em particular, uma possível explicação para o comportamento pode ser que o usuário ficou descontente com a quantidade de resultados relacionados às palavras-buscadas e vai, gradualmente, perdendo interesse na busca. Este interesse pode vir a ser redirecionado a outros resultados, mesmo que não diretamente relacionados às palavras-chave buscadas.

Para concluir a análise dos dados, a Tabela 4.4 mostra a quantidade de resultados selecionados pelo usuário, retornados pelo Praesto mas não pelo SPARQL (exemplo de busca léxico sintática, sem utilização de informações semânticas).

Tabela 4.4: Resultados Selecionados não Retornados pelo SPARQL (busca somente léxico sintática)

Usuário	Interações	Quantidade de Resultados
1	2, 3, 5	2, 1, 2
4	5, 6	1, 1
6	9	2
7	2, 3, 8, 9	1, 3, 1, 2

Capítulo 5

Trabalhos Relacionados

Este capítulo apresenta alguns sistemas de buscas, escolhidos porque, de acordo com a descrição fornecida por seus autores, utilizam informação de contexto para melhorar algum aspecto de sistemas de buscas.

A avaliação dos trabalhos é baseada na reação dos usuários ao sistema, seu funcionamento e os resultados obtidos para as buscas. Dentre os trabalhos com resultados de experimentos publicados, cada trabalho teve experimentos realizados com conjuntos distintos de dados os quais foram buscados por diferentes conjuntos de usuários, sob procedimentos diferentes. Por isto não é possível fazer uma comparação direta e objetiva entre os trabalhos, já que os resultados dependem de avaliações subjetivas dos usuários, realizadas sobre conjuntos de dados diferentes em condições também diferentes.

[Graupmann et al. 2005]

O trabalho em [Graupmann et al. 2005] apresenta um sistema de busca em esfera para documentos *XML* heterogêneos e para dados na *Web*. Os resultados são apresentados classificados com base em estatísticas de recuperação de informação e também com base em relações ontológicas estatisticamente quantificadas.

Documentos *HTML*, como páginas da *Web*, ou documentos *PDF* são convertidos para documentos *XML* e são opcionalmente anotados semanticamente por meio de ferramentas linguísticas de anotação, mais precisamente utilizando a ferramente *ANNIE*, que

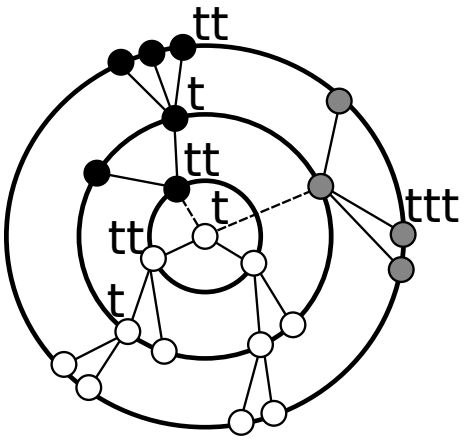

Figura 5.1: Busca esférica por ‘t’ em 3 documentos *XML* [Graupmann et al. 2005].

faz parte do *GATE* (*General Architecture for Text Engineering*). Durante esta transformação, marcações semanticamente significantes são adicionadas além das marcações de *layout* presentes nos documentos *HTML* padrões.

Uma linguagem de consulta foi desenvolvida, que inclui busca por palavras-chave ciente de conceitos (*concept-aware*) e classificação dos resultados influenciada pelo contexto (*context-aware ranking*). O contexto é explorado através de expressões condicionais e da noção de esferas.

A esfera (busca esférica) se refere ao fato que a relevância de um elemento no documento *XML* sendo buscado depende também dos elementos ao seu redor, fazendo com que a busca aconteça em círculos na árvore de elementos, como ilustrado na Figura 5.1. Quanto mais distante do centro do círculo menor a relevância do elemento. As consultas podem ser agrupadas e os grupos são avaliados separadamente. Condições de junção permitem expressar propriedades comuns entre diferentes grupos, permtindo consultas mais elaboradas.

A desambiguação das palavras-chave pode ser feita explicitamente pelo usuário, declarando uma condição para a palavra-chave que especifica o conceito ao qual ela pertence. Em [Graupmann et al. 2005], os autores destacam que, ainda que simples, a linguagem de consulta desenvolvida é elaborada demais para o típico usuário final. Para este tipo de usuário o sistema conta com o auxílio de uma interface gráfica para formular

as consultas.

Quando a experimentos, os autores destacam a dificuldade em se definir um *benchmark* para este tipo de aplicação:

> "Deixem-nos primeiramente comentar sobre as dificuldades de se definir um *benchmark* significativo para este novo tipo de sistema." [Graupmann et al. 2005]

Por isso a avaliação do sistema foi realizada através de experimentos empíricos realizados com usuários. Os usuários definiram um conjunto de consultas, das quais algumas foram selecionadas. Estas consultas foram realizadas sobre as páginas da Wikipedia e os resultados foram comparados com consultas equivalentes realizadas pelo Google, restringindo a busca também à Wikipedia.

[Winkler 1999]

O trabalho descrito em `[Winkler 1999]` foca na identificação do problema de como gerenciar o conhecimento compartilhado e individual e fornece uma possível solução, porém não descreve sua implementação. O trabalho aponta a necessidade de personalização do espaço de trabalho do usuário mantendo a interoperabilidade entre bases de dados mantidas por instalações individuais, quando essas são utilizadas em grupo.

Isso significa que as aplicações *desktop* precisam apresentar a informação de uma forma consensual - de acordo com os autores, semanticamente interoperável - o que é conflitante com a personalização. Porém, esta personalização pode ser obtida através da apresentação do universo de recursos de informação disponíveis para o usuário através de um Mapa de Tópicos.

Um Mapa de Tópicos é um grafo de tópicos em que um tópico associa uma palavra a uma descrição formal fornecida por uma ontologia. A ontologia fornece a interoperabilidade semântica enquanto os tópicos ligados por arestas oferecem a personalização da visão da ontologia.

O trabalho é apresentado como um componente a ser implementado para uma plataforma semântica chamada de *IRIS*.

[Michlmayr et al. 2007]

Os autores de [Michlmayr et al. 2007] apresentam o problema de apresentar a informação desejada de uma maneira que ela faça sentido para quem busca como um grande desafio para usuários de sistemas de informação. Destacam que uma alternativa comumente usada, a criação de perfis, exige esforço do usuário que tem que mantê-lo sempre atualizado, porém pode ser em alguns casos, criado dinamicamente, explorando a popularidade de sistemas de marcação (*tagging*) colaborativos.

Serviços de marcação como o del.icio.us [1] fornecem interfaces para anotar marcadores (*bookmarks*) com palavras-chave, sem restrição de quais podem ser as palavras. Essas informações refletem a mudança de interesses do usuário ao longo do tempo e como elas estão publicamente disponíveis, representam uma grande fonte de metadados.

O perfil do usuário é criado a primeira vez identificando *tags* que foram utilizadas ao mesmo tempo pelo usuário. Esta informação é armazenada na forma de um grafo onde os vértices são as *tags* e as arestas representam a sua co-ocorrência, como ilustrado na Figura 5.2. Pesos nas arestas refletem a quantidade de vezes que elas são utilizadas em conjunto; quanto mais vezes maior o peso.

Quando o usuário registra um novo *bookmark* as *tags* utilizadas para descrevê-lo são passadas para o grafo, atualizando-o. As arestas que já existem no grafo tem seus pesos periodicamente reduzidos, o que permite a evolução do grafo representando o contexto do usuário, esquecendo *tags* que não são mais utilizadas pelo usuário.

Os experimentos realizados com usuários revelaram que a utilização da co-relação juntamente com o decréscimo gradual do peso das arestas resultou em uma evolução do perfil que é mais intensa em períodos de maior atividade do usuário e que equilibra a relevância de *tags* muito usadas com *tags* rescentemente usadas. Contudo, estes resultados dependem de uma escolha cuidadosa da velocidade com que o peso das arestas são atenuados.

[1] http://del.icio.us

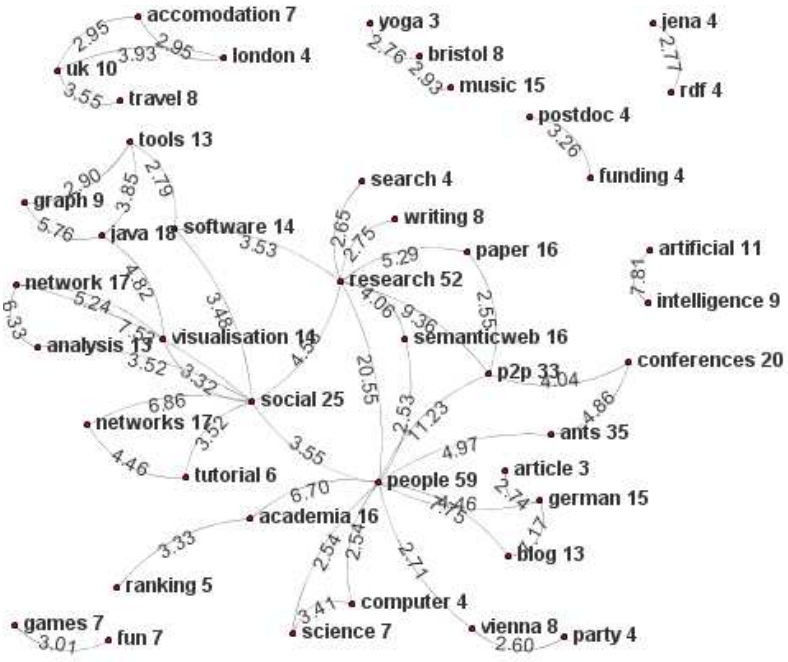

Figura 5.2: Grafo representando perfil do usuário (Imagem de `http://wit.tuwien.ac.at/people/michlmayr/addatag/`, 21/06/2009).

[Aleman-Meza et al. 2003]

A proposta apresentada em `[Aleman-Meza et al. 2003]` têm o objetivo de deixar à mostra informações potencialmente interessantes ao usuário, mas que quando buscadas através de buscas por palavras-chave, retornam entre diversos documentos sem relevância ou inserida dentro de conjuntos de dados aparentemente sem relação. Neste trabalho, as associações semânticas entre diferentes conteúdos são determinadas com base em caminhos estabelecidos nos grafos *RDF* utilizados para descrever metadados.

Duas entidades são consideradas semanticamente conectadas se existe um caminho entre elas. Já grupos de entidades são considerados semanticamente similares se as propriedades nos caminhos ligando as entidades de um grupo são sub-propriedades de caminhos ligando as entidades do outro grupo. Entidades que sejam conectadas ou similares são consideradas semanticamente associadas.

A princípio, é esperado que uma busca semântica no grafo *RDF*, conforme os critérios anteriores, possa retornar um número elevado de caminhos com, conseqüentemente

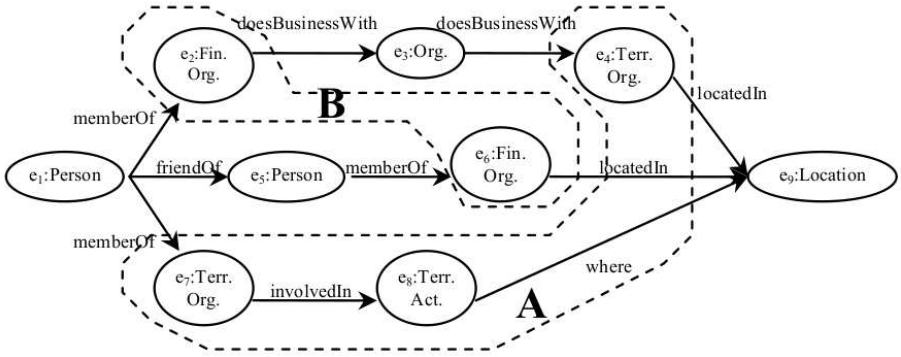

Figura 5.3: Diferentes contextos (A e B) explicitamente definidos por regiões [Aleman-Meza et al. 2003].

um número elevado de entidades consideradas de interesse do usuário. Muitas das quais seriam, posteriormente, desconsideradas pelo usuário. Para evitar esta provável situação as associações são filtradas, com base no contexto em que as buscas são realizadas e os resultados são utilizados.

Os contextos são definidos explicitamente por usuários especialistas, na forma de regiões da ontologia utilizada pelo sistema, como mostrado na Figura 5.3. O exemplo da figura mostra duas regiões em uma parte de uma ontologia utilizada para um sistema de informação relacionado às atividades de segurança. Quando os caminhos são processados no grafo *RDF*, quanto do caminho percorrido está inserido ou próximo de cada região é considerado para avaliar a relevância daquele caminho em relação ao contexto de cada região.

O sistema apresentado pelos autores é utilizado para realização de tarefas em que os usuários, no caso agentes de segurança, devem seguir regulamentos e procedimentos bem definidos, iguais para todos os usuários. Porém, a necessidade de intervenção de usuários especialistas e criação do contexto diretamente na ontologia limita a eficiência da proposta para personalização das buscas, baseada em contextos individuais.

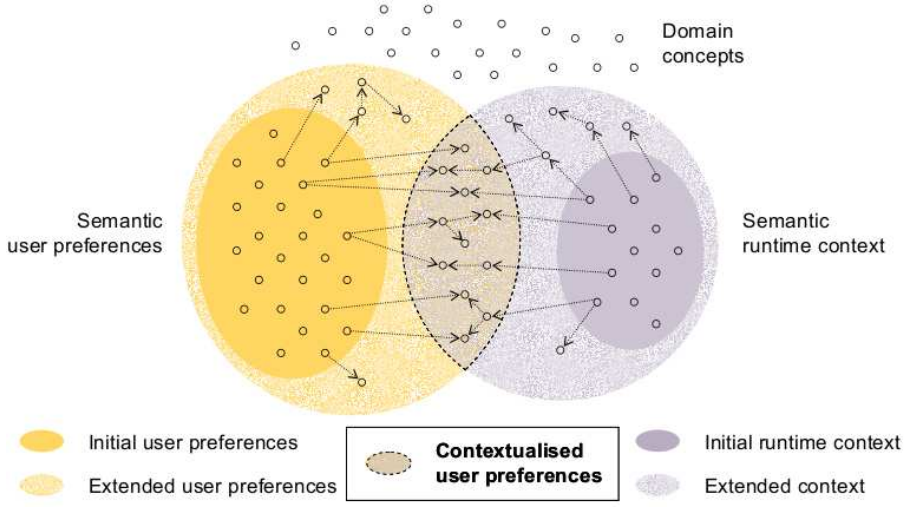

Figura 5.4: Ativação contextual das preferências semânticas do usuário [Vallet et al.].

[Vallet et al.]

Na proposta apresentada em [Vallet et al.], os autores definem contexto como sendo o conjunto de conceitos envolvidos nas interações do usuário com o sistema durante buscas. O contexto é representado como um vetor de pesos de conceitos no intervalo $[0, 1]$ que armazena o número de acessos a um conceito durante uma sessão. O peso é calculado através da combinação cumulativa de número de utilizações do conceito em buscas sucessivas, de forma que os pesos dos conceitos são atenuados com o tempo, garantindo uma representação do contexto relevante para cada momento.

As buscas são realizadas através da expansão do contexto, conforme ilustrado na Figura 5.4. Dado um conjunto de conceitos representando as preferências do usuário, as associações destes conceitos na ontologia são percorridas, atenuando o peso inicial (calculado com base no número de vezes que o conceito foi utilizado pelo usuário) conforme se afasta do conceito onde a busca se originou. Esta expansão persiste até que o peso atenuado seja menor que um valor de corte definido no sistema ou até atingir zero. Os conceitos percorridos na expansão são então organizados em um vetor representando o contexto e são usados para a recuperação dos documentos.

A relevância dos resultados de acordo com o contexto do usuário é calculada pela

sua similaridade com o vetor de contexto calculado para o usuário, ou mais precisamente, pela similaridade entre o conjunto de conceitos usados para anotar o documento com o vetor de contexto. A recuperação pode ser configurada para atribuir diferentes graus de importância às informações de contexto em relação a um mecanismo de busca convencional. Desta maneira é possível configurar se os resultados são trazidos em sua totalidade classificados em função do contexto, se desconsideram o contexto, ou se os resultados são retornados pela combinação de uma busca convencional com a busca utilizando contexto.

O sistema apresentado utiliza representações individuais dos contextos do usuário, geradas com base no seu comportamento passado com o sistema, porém exige do usuário conhecimento sobre o conteúdo descrito na ontologia. Como reflexo desta condição, o sistema também não considera o vocabulário do usuário; são consideradas as descrições e relações dos conceitos descritos na ontologia, mas não o uso do vocabulário, ou seja, a maneira como que cada usuário se refere a estes conceitos.

[Mani and Sundaram 2007]

O trabalho [Mani and Sundaram 2007] define contexto como sendo “um conjunto finito e dinâmico de informações multi-sensoriais e condições inter-relacionadas que influenciam a troca de mensagens entre duas entidades se comunicando”. Este conjunto de informações é o conjunto que pode ser estimado pelo receptor da mensagem baseado na origem e nas condições que influenciam a mensagem. Segundo os autores, quando duas entidades (como pesssoas) se comunicam, cada um mantém uma estimativa do conhecimento e contexto da outra entidade e do ambiente e a exatidão desta estimativa influencia a efetividade da comunicação.

O trabalho apresentado foca na compreenção da relação entre contexto e conhecimento e no desenvolvimento de uma representação do conhecimento baseada em grafos, capaz de modelar o contexto do usuário como um subconjunto do conhecimento. Os resultados do trabalho são utilizados em uma aplicação de navegação por uma base de dados de fotografias.

A Figura 5.5 mostra um conjunto de grafos utilizados para representar o conheci-

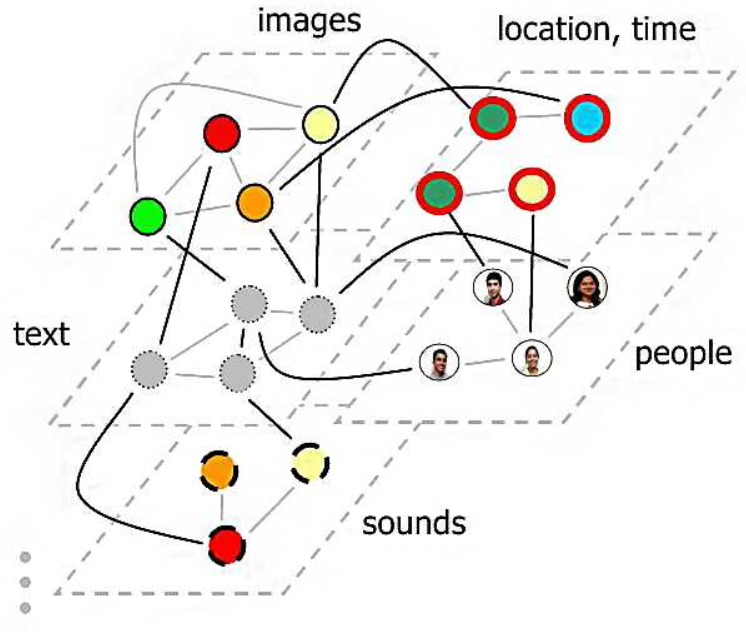

Figura 5.5: Representação do conhecimento multi-sensorial de usuários [Mani and Sundaram 2007].

mento do usuário, organizado em diferentes planos, cada um relativo a um tipo de conhecimento (conhecimento multi-sensorial). Cada nó do grafo representa uma instância de um conceito e as arestas representam o relacionamento entre as instâncias. Os pesos das arestas representam a similariade entre os nós.

As consultas do usuário são representadas como subconjuntos formados pelos nós e das arestas que estão sob atenção do usuário no momento. O grau de atenção do usuário é modelado através de pesos e a soma de todos os pesos de todos conceitos do conhecimento de um usuário é constante em todo momento. O valor constante é justificado pela suposição de que a capacidade de memória de curta duração e conseqüentemente a atenção do usuário, em um intervalo de curta duração, são também constantes. O conhecimento do usuário é inicializado através de informações fornecidas pelo usuário, como por exemplo sua profissão, intereses e algumas imagens do usuário. A partir destas informações outros conceitos são extraídos de representações de conhecimento utilizadas pelo sistema.

Inicialmente todos elementos da representação do conhecimento do usuário possuem o mesmo peso. Conforme cada elemento é utilizado, ou novos são criados, quando necessário, os seus pesos e os pesos dos vizinhos são alterados. Novos fatos sobre o conhecimento do usuário são extraídos de sua interação com o sistema, por exemplo através dos termos consultados.

O funcionamento do sistema apresentado é baseado no estudo e compreenção do processo como ocorre a comunicação humana, por isso é capaz de fornecer representações individuais do conhecimento do usuário, conseqüentemente de seus contextos, já que o contexto não é trabalhado diretamente sobre a ontologia.

[Challam et al. 2007]

Em [Challam et al. 2007] os autores alegam que a maioria dos sistemas de busca não têm preocupação com as necessidades de informação do usuário em um momento específico, considerando que as intenções do usuário mudam ao longo do tempo. Os autores propõem um sistema de busca personalizado baseado em perfis dos contextos dos usuários.

Os perfis são criados baseados na ontologia utilizada pelo sistema. O perfil construído é um perfil de curta duração, ou seja, captura as intenções do usuário no dado momento, através do monitoramento de apliacações como *browser*, editores de texto, mensageiro instantâneo, etc. Estas informações, após serem organizadas pelo sistema, são utilizadas para reordenar os resultados de busca trazidos pelo Google.

Dentre as limitações do sistema está o monitoramento contínuo do comportamento do usuário. Outra questão é a duração do contexto, que é imediato, excluindo informações de interações anteriores com o sistema.

[Sieg et al. 2007]

O trabalho descrito em [Sieg et al. 2007] aponta o contexto do usuário como um fator chave para a personalização do acesso à informação. Baseado nisto, os autores apresentam uma abordagem para personalizar buscas que envolve a construção de modelos

de contexto do usuário. Estes modelos são construídos como perfis ontológicos imbuídos de valores de interesse, calculados implicitamente, que são aplicados aos termos da ontologia.

Os pesos representam o grau de interesse do usuário e são ajustados por um algoritmo de *spreading activation* para navegar pela ontologia (neste caso, uma cópia da ontologia para cada usuário) e manter os valores de acordo com o comportamento do usuário.

O perfil ontológico é utilizado para reclassificar os resultados de buscas, de acordo com os valores atribuídos aos termos da ontologia. Conforme o usuário interage com os resultados apresentados, o perfil é atualizado com base nos termos usados para anotar os documentos, também através de um algoritmo de *spreading activation*.

5.1 Comparativo

Esta seção apresenta uma avaliação dos diferentes trabalhos, organizada na forma da Tabela 5.1, com base nas informações fornecidas pelos próprios autores/desenvolvedores, quando disponíveis. Com base no funcionamento descrito dos sistemas e experimentos relatados os trabalhos são avaliados quanto a alguns aspectos definidos com base nos requisitos do Praesto. Os aspectos avaliados são os seguintes:

- Representação da porção objetiva do conhecimento `(ROC):` indica se o trabalho em questão representa a porção objetiva do conhecimento (baseada em estruturas como ontologias ou bases de conhecimentos).

- Representação da porção subjetiva do conhecimento `(RSC):` indica se o trabalho em questão representa a informação da forma como ela é percebida ou ao menos de forma aproximada de como ela é percebida pelo usuário. Uma interrogação (’?’) é utilizada para indicar que o trabalho atende de forma limitada a este requisito.

- Transparência na Descrição do Conhecimento `(TDC):` este requisito indica se o trabalho em questão procura manter transparente ao usuário a descrição do conhecimento.

Tabela 5.1: Tabela Comparativa de Trabalhos Relacionados

Trabalho	ROC.	RSC.	TDC.	Map. Voc.Def.	Situação
Graupmann et al. [Graupmann et al. 2005]	✓				Impl.
Park, Cheyer [Winkler 1999]	✓	✓	✓	✓	Idea.
Michlmayr et al [Michlmayr et al. 2007]		✓	✓		Impl.
Aleman-Meza et al [Aleman-Meza et al. 2003]	✓		✓		Impl.
Mani, Sundaram [Mani and Sundaram 2007]	✓	✓	✓		Impl.
Challam et al [Challam et al. 2007]	✓	?	✓		Impl.
Vallet et al [Vallet et al.]	✓	?	✓		Impl.
Sieg et al [Sieg et al. 2007]	✓	?	✓		P.Impl.
Praesto	✓	✓	✓	✓	Impl.

- `Situação:` indica se a solução proposta está implementada (Impl.), somente parcialmente implementada (P.Impl.) ou se a solução proposta ainda não foi implementada ou não apresenta indicações de que tenha sido implementada (Idea.).

- Mapemanto do Vocabulário para Definições `(Map. Voc.Def.):` indica se o sistema mapeia o vocabulário utilizado pelo usuário para o vocabulário utilizado na camada de definições, resultando em um mapeamento entre o vocabulário do usuário e os termos da ontologia/base de conhecimento.

Capítulo 6

Conclusão e Trabalhos Futuros

Um dos objetivos definidos no início deste trabalho foi a elaboração de um modelo capaz de capturar e representar o contexto semântico do usuário e utilizá-lo para recuperar resultados relevantes para os interesses do usuário. Este modelo, construído na forma de um grafo de tópicos associado a uma base de conhecimento, assim como os algoritmos necessários para a sua manutenção e utilização em um sistema de busca semântica foram planejados e implementados na forma um sistema de busca, o Praestro [1], que encontra-se funcional. Portanto, foram plenamente cumpridos os três primeiros objetivos do trabalho: (i) desenvolver um modelo para representação de contexto semântico; (ii) integrá-lo a um sistema de busca; (iii) desenvolver algoritmos para utilização de informação do contexto semântico do usuário armazenada neste modelo. As realizações incluíram também o preparo de todo um conjunto de dados, que inclui uma ontologia, uma base de conhecimento compatível com esta ontologia e um conjunto de documentos anotados semanticamente com tal base de conhecimento.

A dificuldade para se realizar experimentos capazes de avaliar os resultados do sistema de busca quanto aos interesses do usuário prejudicou o último objetivo, a validação do trabalho através da realização de experimentos. Não foi possível avaliar os benefícios do sistema proposto de acordo com métricas utilizadas usualmente na área de recuperação de informação, como precisão e revocação. Porém, foi possível verificar uma crescente

[1] disponível em `www.lisa.inf.ufsc.br`, intermitentemente em função de manutenção e atualizações

correspondência entre os resultados com maior relevância atribuída pelo sistema e os resultados selecionados pelo usuário, indicando que o Praesto é capaz de inferir as intenções do usuário em certos casos.

Diversos trabalhos foram publicados como resultados parciais desta dissertação, com diferentes focos: a proposta de captura e uso de informação de contexto semântico em sistemas de busca [D'Agostini and Fileto 2008] [D'Agostini et al. 2007], a estrutura do grafo de tópicos para representar informação de contexto semântico [D'Agostini et al. 2008], a estratégia para manutenção e uso do grafo de tópicos no sistema de buscas [D'Agostini and Fileto 2009] e a interface do Praestro [Fasolin et al. 2009]. Infelizmente, a dificuldade de viabilizar experimentos que demonstrem os benefícios da abordagem proposta tem postergado a submissão de um artigo completo sobre a abordagem em a um periódico.

Trabalhos Futuros

Os trabalhos futuros incluem:

1. Avaliar possíveis níveis de flexibilização do acoplamento entre os grafos de tópicos, que representam o conhecimento específico de cada usuário, e a base de conhecimento, que representa o conhecimento coletivo. Por exemplo, permitir tópicos que não tenham correspondência com conceitos ou instâncias da base de conhecimento ou cujos rótulos sejam diferentes dos nomes utilizados na base de conhecimento.

2. Analisar informação de contexto de conjuntos de usuários, de modo a identificar características relevantes para traçar o perfil semântico desses usuários e auxiliar na recuperação de informação de usuários com perfis semelhantes.

3. Aperfeiçoar a implementação do Praestro, considerando outras abordagens e ferramentas para diversos aspectos de sua funcionalidade e o acoplamento dos seus módulos.

4. Pesquisar conjuntos de dados, cenários e critérios apropriados para medir os benefícios de sistemas utilizando informação de contexto semântico para apoiar o

processamento de buscas. Os conjuntos de dados devem incluir conteúdo a ser pesquisado e base de conhecimento para apoiar a recuperação do conteúdo. Os cenários podem incluir consultas pré-especificadas com denotações bem definidas ou permitir ao usuário pesquisar livremente o conteúdo e apontar os resultados que contemplam seus interesses. Os critérios de análise de desempenho devem permitir avaliar possíveis benefícios do uso do contexto semântico na recuperação de informação, tal como uma possível diminuição do tempo total para a obtenção de resultados procurados, a medida que o sistema acumula informação de contexto.

5. Efetuar experimentos mais extensos e aprofundados para medir os benefícios da coleta e uso de informação de contexto semântico na recuperação de informação, considerando o comportamento de grupos de usuários com diferentes perfis, em diferentes domínios de aplicação, ao longo de um número expressivo de interações do sistema, que permitam capturar uma quantidade considerável de informação de contexto semântico e combinando diferentes possibilidades de valores para diversos parâmetros.

6. Analisar questões de projeto de interfaces-homem máquina que possam influir na usabilidade e na comunicação do sistema com o usuário, de modo a permitir ao sistema capturar adequadamente as preferências e intenções do usuário a cada busca. Dentre as várias possibilidades levantadas para os componentes de interface, podem ser avaliadas quais resultam em uma melhor experiência de interação com o sistema. Por exemplo, pode ser adequado, ao menos em certas circunstâncias, permitir a um usuário navegar no seu mapa de tópicos e estipular buscas via seleção de tópicos.

Referências Bibliográficas

[dbp] DBPedia: http://wiki.dbpedia.org - acessado em 21 de junho de 2009.

[Aleman-Meza et al. 2003] Aleman-Meza, B., Halaschek, C., IB, A., and Sheth, A. (2003). Context-Aware Semantic Association Ranking, In First International Workshop on Semantic Web and Databases. Berlin, Germany.

[Baeza-Yates et al. 1999] Baeza-Yates, R., Ribeiro-Neto, B., et al. (1999). *Modern information retrieval*. Addison-Wesley Harlow, England.

[Challam et al. 2007] Challam, V., Gauch, S., and Chandramouli, A. (2007). Contextual search using ontology-based user profiles. *Conference RIAO2007*. Pittsburg PA, U.S.A.

[D'Agostini et al. 2007] D'Agostini, C., Fileto, R., Dantas, M. A. R., and Gauthier, F. O. (2007). Inferring user's intentions through context. *Sessão de Posters do Simpósio Brasileiro de Bancos de Dados (SBBD)*, pages 7–10. João Pessoa, PB, Brasil.

[D'Agostini et al. 2008] D'Agostini, C., Fileto, R., Dantas, M. A. R., and Gauthier, F. O. (2008). Contextual semantic search - capturing and using the user's context to direct semantic search. *In 10th International Conference on Enterprise Information Systems*. Barcelona, Spain.

[D'Agostini and Fileto 2009] D'Agostini, C. S. and Fileto, R. (2009). Proceedings of the 21st international conference on software engineering & knowledge engineering (seke'2009) july 1-3, 2009. In *SEKE*. Knowledge Systems Institute Graduate School. Boston, Massachusetts, USA,.

[Degler and Lewis 2004] Degler, D. and Lewis, R. (2004). Maintaining ontology implementations: The value of listening. In *Extreme Markup Languages 2004: Proceedings*. IDEAlliance.

[Dietrich 2000] Dietrich, E. (2000). Cognitive Science and the Mechanistic Forces of Darkness, or Why the Computational Science of Mind Suffers the Slings and Arrows of Outrageous Fortune. *Techné: eJournal of the Society for Philosophy and Technology*.

[Dorigo et al. 1999] Dorigo, M., Caro, G., and Gambardella, L. (1999). Ant Algorithms for Discrete Optimization. *Artificial Life*, 5(2):137–172.

[D'Agostini and Fileto 2008] D'Agostini, C. S. and Fileto, R. (2008). Capturing and managing the user context for improving information retrieval. In *VII Workshop de Teses e Dissertações em Bancos de Dados (SBBD)*, pages 31–36. Campinas, SP, Brasil.

[Fasolin et al. 2009] Fasolin, K., D'Agostini, C. S., Fileto, R., and Besen, R. (2009). Praesto - a system for contextual semantic search. *Simpósio Brasileiro de Banco de Dados*, page 25.

[Fry 2007] Fry, B. (2007). *Visualizing Data*. O'Reilly Media, Inc.

[Graupmann et al. 2005] Graupmann, J., Schenkel, R., and Weikum, G. (2005). The SphereSearch engine for unified ranked retrieval of heterogeneous XML and web documents. *Proceedings of the 31st international conference on Very large data bases*, pages 529–540.

[Greer et al. 2007] Greer, K., Baumgarten, M., Mulvenna, M., Nugent, C., and Curran, K. (2007). Knowledge-Based Reasoning Through Stigmergic Linking. *LECTURE NOTES IN COMPUTER SCIENCE*, 4725:240.

[Huang et al. 2002] Huang, W., Prie, Y., Champin, P., and Mille, A. (2002). Semantic context representation of resources using annotation graph. In *Proceedings of the Eighth International Workshop on Multimedia Information Systems 2002*.

[Hurley 1995] Hurley, C. (1995). Ambient Functions-Abandoned Children to Zoos. *ARCHIVARIA*, pages 21–39.

[Ibáñez and Cosmelli 2008] Ibáñez, A. and Cosmelli, D. (2008). Moving beyond computational cognitivism: Understanding intentionality, intersubjectivity and ecology of mind. *Integrative Psychological and Behavioral Science*, 42(2):129–136.

[Johnson 2001] Johnson, S. (2001). *Emergence: (The Connectec Lives of Ants, Brains, Cities and Softwares)*. Scribner.

[Kahan et al. 2002] Kahan, J., Koivunen, M., Prud'Hommeaux, E., and Swick, R. (2002). Annotea: an open RDF infrastructure for shared Web annotations. *Computer Networks*, 39(5):589–608.

[Kiryakov et al. 2004] Kiryakov, A., Popov, B., Terziev, I., Manov, D., and Ognyanoff, D. (2004). Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2(1):49–79.

[Koshman 2006] Koshman, S. (2006). Visualization-based information retrieval on the Web. *Library and Information Science Research*, 28(2):192–207.

[Leake et al. 2005] Leake, D., Maguitman, A., and Reichherzer, T. (2005). Exploiting rich context: An incremental approach to context-based web search. *International and Interdisciplinary Conference on Modeling and Using Context, CONTEXT*, 5:254–267.

[Mangold 2007] Mangold, C. (2007). A survey and classification of semantic search approaches. *International Journal of Metadata, Semantics and Ontologies*, 2(1):23–34.

[Mani and Sundaram 2007] Mani, A. and Sundaram, H. (2007). Modeling user context with applications to media retrieval. *Multimedia Systems*, 12(4):339–353.

[Michlmayr et al. 2007] Michlmayr, E., Cayzer, S., and Shabajee, P. (2007). Tech report: Hpl-2007-72: Adaptive user profiles for enterprise information access. Technical report.

[Naeve 2005] Naeve, A. (2005). The Human Semantic Web-Shifting from Knowledge Push to Knowledge Pull. *International Journal of Semantic Web and Information Systems*, 1(3):1–30.

[Panait and Luke 2004] Panait, L. and Luke, S. (2004). Learning ant foraging behaviors. *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9)*.

[Shtovba 2005] Shtovba, S. (2005). Ant Algorithms: Theory and Applications. *Programming and Computer Software*, 31(4):167–178.

[Sieg et al. 2007] Sieg, A., Mobasher, B., and Burke, R. (2007). Ontological user profiles for personalized web search. *In 5th Workshop on Intelligent Techniques for Web Personalization, Vancouver, Canada, July*.

[Souza et al. 2008] Souza, D., Belian, R., Salgado, A. C., and Tedesco, P. (2008). Towards a context ontology to enhance data integration processes. In *VLDB 08*. ACM.

[Tirri 2003] Tirri, H. (2003). Search in vain: Challenges for internet search. *Computer*, 36(1):115–116.

[Vallet et al.] Vallet, D., Fernández, M., Castells, P., Mylonas, P., and Avrithis, Y. Personalized Information Retrieval in Context. *3rd International Workshop on Modeling and Retrieval of Context (MRC 2006) at the 21st National Conference on Artificial Intelligence (AAAI 2006)*.

[Winkler 1999] Winkler, W. (1999). The state of record linkage and current research problems. http://citeseer.ist.psu.edu/article/winkler99state.html.

Apêndice A

Usuário	Interação	Palavras-chave	#Selecionados	#RetornadosPraesto	#RetornadosSPARQL	Ret.Praesto – Ret.Sparql
1	1º	Paris, London, Moscow	7	89	1700	
	2º	Paris, London, Moscow	10	384	1700	2
	3º	Paris, London, Moscow	8	627	1700	1
	4º	Großdeutsches Reich, Greater German Reich, Germany	7	22	197	
	5º	Großdeutsches Reich, Greater German Reich, Germany	9	197	197	2
	6º	Großdeutsches Reich, Greater German Reich, Germany	7	365	197	
	7º	Poland, Republic of Poland	8	65	207	
	8º	Poland, Republic of Poland	8	230	207	
	9º	Poland, Republic of Poland	8	230	207	
3	1º	David Gilmour	1	12	16	
	2º	David Gilmour	1	12	16	
	3º	David Gilmour	1	12	16	
	4º	Jimmy Page	6	15	16	
	5º	Jimmy Page	7	70	16	1
	6º	Jimmy Page	7	73	16	1
	7º	Kiko Loureiro	1	1	1	
	8º	Kiko Loureiro	1	1	1	
	9º	kiko Loureiro	1	1	1	
6	1º	Nirvana, Aerosmith	4	24	41	
	2º	Nirvana, Aerosmith	4	42	41	
	3º	Nirvana, Aerosmith	4	42	41	
	4º	Queen, The Who	7	53	1437	
	5º	Queen, The Who	9	154	1437	
	6º	Queen, The Who	9	223	1437	
	7º	The Who, Queen, The Rolling Stones, Ramones, The Killers, U2	9	233	1554	
	8º	The Who, Queen, The Rolling Stones, Ramones, The Killers, U2	11	233	1554	2
7	1º	Florianópolis	4	7	7	
	2º	Florianópolis	5	22	7	1
	3º	Florianópolis	7	22	7	3
	4º	Rio de Janeiro	4	23	26	
	5º	Rio de Janeiro	7	23	26	
	6º	Rio de Janeiro	4	29	26	
	7º	Samba	3	5	47	
	8º	Samba	5	25	47	1
	9º	Samba	6	25	47	2
8	1º	Campo Grande	2	4	7	
	2º	Campo Grande	1	4	7	
	3º	Campo Grande	0	4	7	
	4º	Brazil	4	25	148	
	5º	Brazil	2	41	148	
	6º	Brazil	0	42	148	
	7º	Rio de Janeiro	2	23	26	
	8º	Rio de Janeiro	3	23	26	
	9º	Rio de Janeiro	3	52	26	

Figura A.1: Palavras-chave, quantidade de resultados selecionados, retornados pelo SPARQL e quantidade de resultados retornados pelo Praesto mas não pelo SPARQL.

Usuário	Interação	Relev Média Selec	Relev Máxima Selec	Relev Minima	Relev Média	Relev Máxima
1	1º	1	1	1	1	1
	2º	0,66	1	0,02	0,19	1
	3º	0,74	1	0,01	0,12	1
	4º	1	1	1	1	1
	5º	0,69	1	0,01	0,12	1
	6º	0,84	1	0,01	0,07	1
	7º	0,68	1	0,2	0,46	1
	8º	0,63	1	0,01	0,13	1
	9º	0,63	1	0,01	0,13	1
3	1º	1	1	1	1	1
	2º	1	1	1	1	1
	3º	1	1	1	1	1
	4º	1	1	1	1	1
	5º	0,79	1	0,03	0,2	1
	6º	0,81	1	0,03	0,19	1
	7º	1	1	1	1	1
	8º	1	1	1	1	1
	9º	1	1	1	1	1
6	1º	1	1	1	1	1
	2º	0,84	1	0,16	0,5	1
	3º	0,8	1	0,2	0,49	1
	4º	0,71	1	0,5	0,53	1
	5º	0,64	1	0,24	0,18	1
	6º	0,66	1	0	0,12	1
	7º	0,6	1	0	0,14	1
	8º	0,5	1	0	0,14	1
7	1º	1	1	1	1	1
	2º	0,73	1	0,07	0,31	1
	3º	0,55	1	0,09	0,32	1
	4º	1	1	1	1	1
	5º	0,86	1	0,77	0,8	1
	6º	0,9	1	0,02	0,66	1
	7º	1	1	1	1	1
	8º	0,75	1	0,2	0,33	1
	9º	0,68	1	0,29	0,39	1
8	1º	1	1	1	1	1
	2º	1	1	1	1	1
	3º			1	1	1
	4º	1	1	1	1	1
	5º	0,92	1	0,07	0,46	1
	6º			0	0,42	1
	7º	1	1	1	1	1
	8º	0,85	1	0,78	0,8	1
	9º	0,82	0,83	0,08	0,38	1

Figura A.2: Relevâncias mínimas, médias e máximos, dos resultados selecionados e trazidos pelo Praesto.

Glossário

ACO	*Ant Colony Optimization*
OWL	*Web Ontology Language*
RDF	*Resource Description Framework*
RMI	*Remote Methode Invocation*
SPARQL	*SPARQL Protocol and RDF Query Language*
XML	*Extensible Markup Language*